ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SOCIAL SCIENCES

ESSAYS ON ESTIMATION METHODS

Ph.D. THESIS

Yasin KÜTÜK

Department of Economics

Economics Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SOCIAL SCIENCES

ESSAYS ON ESTIMATION METHODS

Ph.D. THESIS

Yasin KÜTÜK
(412132005)

Department of Economics

Economics Programme

Thesis Advisor: Prof. Dr. Bülent GÜLOĞLU

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ SOSYAL BİLİMLER ENSTİTÜSÜ

TAHMİN YÖNTEMLERİ ÜZERİNE MAKALELER

DOKTORA TEZİ

Yasin KÜTÜK
(412132005)

İktisat Anabilim Dalı

İktisat Programı

Tez Danışmanı: Prof. Dr. Bülent GÜLOĞLU

HAZİRAN 2019

Yasin KÜTÜK, a Ph.D. student of ITU Graduate School of Social Sciences in Economics (412132005), successfully defended the thesis entitled "ESSAYS ON ESTIMATION METHODS", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Bülent GÜLOĞLU** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Bülent GÜLOĞLU** İstanbul Technical University	.....
	Prof. Dr. Alp ÜSTÜNDAĞ İstanbul Technical University	.....
	Prof. Dr. Nilgün ÇİL İstanbul University	.....
	Prof. Dr. Filiz KARAMAN Yıldız Technical University	.....
	Assoc. Prof. Dr. Resul AYDEMİR İstanbul Technical University	.....

Date of Submission : **17.04.2019**
Date of Defense : **17.06.2019**

to the people who have adopted transcendental principles of morality...

FOREWORD

In this dissertation, there are three seminal but experimental studies, each of which requires great effort. In the first study, the probabilities of transition of the individuals to join active labor force are examined and a prediction model is presented which is suitable for development of future studies. In the second study, the savings rates of households and individuals were modeled with high prediction success so that macro policies could be accurately established. In the third study, the written texts of the Central Bank that shapes economic policies were transformed into numerical semantic models and modeled with a human-like prediction success.

However, the main subject that is examined in the second scene[1] of "Camera Obscura" is to give new methods for the economics science, by using state-of-the-art methods which provide progress in the field of artificial intelligence, to help provide stronger policy tools that enable high predictability. The main reason for this diligence is to be able to help change, slightly at the least, all known practices, to be a vanguard for new and groundbreaking works, to completely eradicate prejudices so that we might see the realities of today becoming obsolete in time.

I want to express my endless gratitude to my esteemed adviser Prof. Dr. Bülent GÜLOGLU, who was able to take these bold steps with me, who saved me from a deep loneliness, and whose tutelage I would like to be under for a lifetime, without forgetting the help and support of all people, and the previous chair of my department, Prof. Dr. Fuat ERDAL and the present chair, Prof. Dr. Sencer ECER, to my committee Prof. Dr. Alp ÜSTÜNDAĞ for his close interest my topics, Prof. Dr. Nilgün ÇİL for her valuable comments, Prof. Dr. Filiz KARAMAN for her corrections and care to my dissertation and Assoc. Prof. Dr. Resul AYDEMİR who carefully listens to all the people in front of him and deems them so valuable.

From my workplace at Altınbaş University [2] where I feel as part of a family, I would like to express my thanks to the honourable Prof.Dr. Emre ALKİN for his endless wholehearted support, and the valuable colleagues for their support of my efforts. I would like to express my deepest respects to Prof. Dr. İpek İLKKARACAN AJAS, who will never give up on women's human rights, and Prof. Dr. Ümit ŞENESEN who brought together a number of respectable academics under his umbrella, for their support during my Ph.D. and M.A. .

I would also like to thank Doç. Dr. Mine AKSU, together with whom we accomplished valuable works at Sabancı University, my dear friend and colleague Mehtap ALDOGAN, my dearest Yasemin KÜTÜK and my beloved family for their support and belief in me, the greatest band of the universe, RADIOHEAD.

And finally, to the open source supporters who have the greatest grasp of this dissertation,

[1] The first one is my M.A. thesis.

[2] formerly: İstanbul Kemerburgaz University.

- the Free Software Foundation[3],
- the Python Software Foundation[4],
- the R Foundation[5],
- creators of LaTeX[6], the most powerful tool that makes this dissertation possible to write w/ hard and complex equations, graphs mostly done in LaTeXenvironment, tabulations and other things,
- the Apache Software Foundation[7],
- all contributors of Stackoverflow [8]https://stackoverflow.com/,
- Andrew Ng's Coursera for providing and giving me scholarship to pursue Machine Learning and Deep Learning Specialization courses [9]https://www.coursera.org/specializations/deep-learning,
- the Terminal (Ctrl+T) and the greatest "sudo",
- the Linux Foundation[10], very comfortable home of the above I stated,

  i would like to thank so much for supporting you all to produce all the codes that I wrote.

June 2019

Yasin KÜTÜK
Research Assistant

[3]https://www.fsf.org/
[4]https://www.python.org/psf/
[5]https://www.r-project.org/foundation/
[6]https://www.latex-project.org/
[7]http://www.apache.org/foundation/
[8]https://stackoverflow.com/
[9]https://www.coursera.org/specializations/deep-learning
[10]https://www.linuxfoundation.org

TABLE OF CONTENTS

Page

FOREWORD ..... **x**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xv**
SYMBOLS ..... **xix**
LIST OF TABLES ..... **xxi**
LIST OF FIGURES ..... **xxiii**
SUMMARY ..... **xxv**
ÖZET ..... **xxvii**
1. INTRODUCTION **1**
1.1 The Importance of Prediction Power in Economics 1
2. METHODOLOGIES of ESTIMATION METHODS **13**
2.1 Machine Learning 14
2.1.1 Types of machine learning 15
2.1.2 Supervised learning 15
2.1.2.1 Regression 15
2.1.2.2 Classification and logistic regression 18
2.1.3 Unsupervised learning 25
2.1.3.1 Clustering 26
2.1.3.2 Dimensionality reduction 27
2.1.4 Semi-supervised learning 30
2.1.5 Reinforcement learning 30
2.1.6 Federated learning 32
2.1.7 Transfer learning 32
2.1.8 Ensemble learning 32
2.1.8.1 Random forest 33
2.1.8.2 Extreme gradient boosting 34
2.2 Machine Learning Algorithms 34
2.2.1 Gradient descent 34
2.2.1.1 Gradient descent implementation 36
2.2.2 Feature scaling 38
2.2.3 Regularization 39
2.2.3.1 L^2 Regularization: Ridge regression 39
2.2.3.2 L^1 Regularization: Lasso regression 40
2.3 Bias-Variance Tradeoff 40
2.4 Neural Networks 43
2.4.1 Single layer perceptron 45
2.4.2 Multilayer perceptron 45

2.4.3 Neural networks: Learning 48
2.4.3.1 Backpropagation algorithm 48
2.4.3.2 Random initialization 50
2.4.3.3 Architecture design 50
2.4.3.4 Feed forwards 50
2.4.3.5 Weight averaging 51
2.4.4 Convolutional neural networks 52
2.4.5 Sequence modeling 53
2.4.5.1 Recurrent networks 53
2.5 Activation Functions 59
2.6 Monotonic Activation Functions 59
2.6.1 Linear function 60
2.6.1.1 Identity Function 60
2.6.1.2 Piecewise linear function 61
2.6.2 Threshold (unit heaviside, binary, step) function 62
2.6.3 Sigmoid function 62
2.6.3.1 Bipolar sigmoid function 63
2.6.4 Rectified linear unit (ReLU) 64
2.6.4.1 Leaky rectified linear unit (LReLU) 65
2.6.4.2 Parametric rectified linear unit (PReLU) 65
2.6.4.3 Randomized rectified linear unit (RReLU) 66
2.6.5 Exponential linear unit (ELU) 66
2.6.5.1 Scaled exponential linear unit (SELU) 67
2.6.6 Softmax function 67
2.6.7 Odd activation (signum, sign) function 68
2.6.8 Maxout function 68
2.6.9 Softsign function 69
2.6.10 Elliott function 70
2.6.11 Hyperbolic tangent (Tanh) function 71
2.6.11.1 Arc tangent function 71
2.6.11.2 Lecun's hyperbolic tangent function 72
2.6.12 Complementary log-log function 72
2.6.13 Softplus function 73
2.6.14 Bent identity function 74
2.6.15 Soft exponential function 75
2.7 Periodic Activation Functions 76
2.7.1 Sinusoidals 76
2.7.1.1 Sine wave function 77
2.7.1.2 Cardinal sine function (Sinc) 77
2.7.1.3 Fourier transform (FT, DFT/FFT) 78
2.7.1.4 Discrete-time dimensional fourier transform (DTFT, Shannon-Nyquist) 80
2.7.1.5 Short time fourier transform (Gabor, STFT) 80
2.7.1.6 Wavelet transform 81
2.7.2 Non-sinusoidals 83
2.7.2.1 Gaussian (normal) distribution function 83

2.7.2.2 Square wave function 84
2.7.2.3 Triangle wave function 85
2.7.2.4 Sawtooth wave function 86
2.7.2.5 S-shaped rectified linear unit (SReLU) 86
2.7.2.6 Adaptive piecewise linear unit (APLU) 88
2.8 Bias Unit 89
2.9 Econometrics 89
2.9.1 Logistic regression 90
2.9.2 Quantile regression (QREG) 93
2.9.3 Theoretical background of quantile regression 93
2.9.3.1 Quantiles 94
2.9.3.2 Estimation of quantile regression 96
2.9.4 Goodness of fit measure in QREG 102
3. PREDICTION OF TRANSITION PROBABILITIES FROM UNEMPLOYMENT TO EMPLOYMENT FOR TURKEY VIA MACHINE LEARNING AND ECONOMETRICS: A COMPARATIVE STUDY 105
3.1 Introduction 105
3.2 Previous Researches 109
3.3 Turkish Studies 123
3.4 Data 124
3.5 Research Design 126
3.6 Results 129
3.7 Concluding Remarks 133
3.8 Limitations & Further Studies 135
4. PREDICTING SAVING RATES AND PROVIDING PRESCRIPTIVE STATISTICS AT THE LEVEL OF INDIVIDUALS AND HOUSEHOLDS FOR TURKEY: AN IMPLEMENTATION OF DEEP NEURAL NETWORKS 137
4.1 Introduction 137
4.2 Background 139
4.3 Turkish Studies 157
4.4 Data 159
4.5 Research Plan and Analysis of Results 161
4.5.1 Feature selection (FS) process 165
4.6 Concluding Remarks 173
4.7 Limitations & Further Studies 174
5. SEMANTIC ANALYSIS OF THE CENTRAL BANK OF THE REPUBLIC OF TURKEY COMMUNICATION REPORTS AND FORECASTING MODEL WITH LSTM 177
5.1 Introduction 177
5.2 Past Researchs 179
5.2.1 Statistical natural language processing (SNLP) 180
5.2.2 SNLP and central bank communications 182
5.3 Data 185
5.4 Research Design and Procedures 187

5.5 Results 193
5.6 Concluding Remarks 204
5.7 Limitations & Further Studies 205
6. CONCLUSION 207
APPENDICES 243
Appendix A : Glossary 245
Appendix B : Chapter 3 Appendices 251
3.1 R Codes Used to Merge HLFSs 288
3.2 R Codes for Variable Generation 330
3.3 R Codes for Analysis 332
3.4 Python Codes for Feature (Variable) Engineering 339
3.5 Python Codes for Classifiers in Machine Learning 346
Appendix C : Chapter 4 Appendices 359
3.1 R Codes for Converter of HBSs to CSV 497
3.2 R Codes Used to Merge IL Part of HBSs 500
3.3 R Codes Used to Merge HL Part of HBSs 548
3.4 R Codes Used to Merge IL with HL 560
3.5 R Codes for Descriptive Statistics 563
3.6 Python Codes for QREG 575
3.7 Python Codes for Deep Learning 582
Appendix D : Chapter 5 Appendices 597
4.1 Python Web Crawler for SMPCM Links 617
4.2 Python Web Crawler for SMPCM PDFs 620
4.3 Python PDF to Text Extractor 622
4.4 Python Word Cloud Generator 625
4.5 Python Word Counter 628
4.6 Python SMPCM Item Reader 630
4.7 Python N-Gram Extractor 632
4.8 Python Sentiment Neuron 636
4.9 Python LSTM Network Model Implementation in Keras 643
CURRICULUM VITAE 648

ABBREVIATIONS

ACC : Accuracy
AI : Artificial Intelligence
AID : Automatic Interaction Detection
ANN : Artificial Neural Network
AODE : Averaged One-Dependence Estimators
APC : Average Propensity to Consume
APLU : Adaptive Piecewise Linear Unit
AVG : Average
BAX : Baxter International Inc. Common Stock
BBN : Bayesian Belief Network
BFGS : Broyden–Fletcher–Goldfarb–Shanno Algorithm
BGD : Batch Gradient Descent
BLUE : Best Linear Unbiased Estimator
BN : Bayesian Network
CART : Classification And Regression Trees
CB : Central Bank
CBRT : Central Bank of Republic of Turkey
CC : Correlation Coefficient
c.d.f. : Cumulative Density Function
CHAID : Chi-squared Automatic Interaction Detection
CNN : Convolutional Neural Networks
Con : Contra
CPU : Central Processing Unit
CSV : Comma Seperated Values
CUDA : Compute Unified Device Architecture
CWT : Continuous Wavelet Transform
DFT : Discrete Fourier Transform
DL : Deep Learning
DNN : Deep Neural Network
DRNN : Deep Regression Neural Network
DQRNN : Deep Quantile Regression Neural Network
DTCWT : Discrete Time Continuous Wavelet Transform
DTFT : Discrete Time Fourier Transform
DTWT : Discrete Time Wavelet Transform
DWT : Discrete Wavelet Transform
ECB : European Central Bank
ELU : Exponential Linear Unit
Ext4 : Fourth Extended Filesystem
FAI : Factors Affecting Inflation
FED : Federal Reserve (US Central Bank)
FFT : Fast Fourier Transform
FNN : Feed Forward Neural Networks

FS : Feature Selection
FSS : Financial Stability Survey
FT : Fourier Transform
GB : Gigabyte
GD : Gradient Descent
GLM : Generalized Linear Model
GPU : Graphical Processing Unit
GS : Gale-Shapley Algorithm
HBS : Household Budget Survey
HH : Household
HL : Household Level
HLFS : Household Labor Force Survey
ID3 : Iterative Dichotomiser v3
ID4.5 : Iterative Dichotomiser v4.5
IL : Individual Level
ILO : International Labor Organization
InD : Inflation Developments
kNN : K-Nearest Neighbor
L-BFGS : Limited-Memory Broyden–Fletcher–Goldfarb–Shanno Algorithm
LASSO : Least Absolute Shrinkage and Selection Operator
LCH : Life Cycle Income Hypothesis
LDA : Latent Dirichlet Analysis
LReLU : Leaky Rectifier Linear Unit
LSA : Latent Semantic Analysis
LSTM : Long Short-Term Memory
LVQ : Learning Vector Organization
LWL : Locally Weighted Learning
MCMC : Markov Chain Monte Carlo
ML : Machine Learning
MLE : Maximum Likelihood Estimation
MNIST : Modified National Institute of Standards and Technology Database
MPC : Marginal Propensity to Consume
MPR : Monetary Policy and Risks
NA : Not Available
NaN : Not A Number
NB : Naive Bayes
NIMP : National Intern Matching Program
NLP : Natural Language Processing
NN : Neural Network
NRMP : National Resident Matching Program
OECD : Organisation for Economic Cooperation and Development
OLS : Ordinary Least Squares
OS : Operating System
PBSCC : Point BiSerial Correlation Coefficient
PCA : Principal Component Analysis
PIH : Permanent Income Hypothesis
PReLU : Piecewise Rectifier Linear Unit
QREG : Quantile Regression

RAM : Random Access Memory
ReLU : Rectifier Linear Unit
RIH : Random Walk Income Hypothesis
RMSE : Root Mean Squared Error
RNN : Recurrent Neural Networks
ROC : Receiever Operating Characteristics
ROL : Rank Order List
RPL : Resource Preference List
RReLU : Randomized Rectifier Linear Unit
RSS : Residual Sum of Squares
SELU : Scaled Exponential Linear Unit
SGD : Stochastic Gradient Descent
SLS : Stage Least Square
SMPCM : Summary of Monetary Policy Committee Meeting
SNARC : Stochastic Neural Analog Reinforcement Calculator
SNLP : Statistical Natural Language Processing
SOM : Self Operating Map
SReLU : S-Shaped Rectifier Linear Unit
SSV : Sentiment Score Vector
STFT : Short Time Fourier Transform
SVC : Support Vector Classifier
SVM : Support Vector Machine
TL : Turkish Lira
TN : True Negative
TP : True Positive
TPU : Tensor Processing Unit
TURKSTAT : Turkish Statistical Institute
UK : United Kingdom
UPL : User's Preference List
UR : Unemployment Ratio
US : United States
VAR : Vector Autoregression
WWII : World War II
YTL : Yeni (New) Turkish Lira

SYMBOLS

arg	**:** Argument
C	**:** Continuous
cos	**:** Cosine
$\mathscr{D}$	**:** Particular value of θ
e	**:** The base of the natural logarithm, mathematical constant
exp	**:** Exponential
inf	**:** Infimum
∞	**:** Infinity
J	**:** Jacobian Matrix
α	**:** Learning Rate
$\mathscr{L}$	**:** Likelihood
log	**:** Logarithm
max	**:** Maximization
min	**:** Minimization
ln	**:** Natural Logarithm
μ	**:** Population Mean
σ	**:** Population Standard Deviation
Pr	**:** Probability
λ	**:** Regularization
sin	**:** Sine
tan	**:** Tangent
tr	**:** Trace
T	**:** Transpose of Matrix
$:=$	**:** Updating rule
$w/$	**:** With
w/o	**:** Without
$w.r.t$	**:** With Respect To

LIST OF TABLES

Page

Table 2.1 : Supervised Learning. 15
Table 3.1 : Unemployment rates all over the World.*..... 106
Table 3.2 : Tabulation of HLFS (2004-2016)..... 126
Table 3.3 : Machine Learning Algorithms. 127
Table 3.4 : Best Classifiers Results. 131
Table 4.1 : Saving Motives Review. 143
Table 4.2 : Performance indicators for QREG, DRNN, DQRNN w/o FS. 168
Table 4.3 : Performance indicators for QREG, DRNN, DQRNN w/ FS. 169
Table 5.1 : Descriptive Statistics for Sentence-based Sentiments of SMPCMs Data. 203
Table 5.2 : LSTM Model Structure. 203
Table A.1 : Types of Errors (Confusion Matrix)..... 249
Table B.1 : Descriptive Statistics of Quantitative Features. 251
Table B.2 : Descriptive Statistics of Qualitative Features 251
Table B.3 : Classifier Results..... 253
Table B.4 : Impurity Measurement Metrics..... 256
Table B.5 : Logistic Regression. 257
Table B.6 : Household Labor Force Survey 265
Table C.1 : Descriptive Statistics for Quantitative Variables. 359
Table C.2 : Descriptive Statistics of Qualitative Features. 363
Table C.3 : Budget Survey - HH Questionnaire. 369
Table C.4 : Budget Survey - Individual Questionnaire..... 377
Table C.5 : Quantile Regression Results - 10th Quantile. 390
Table C.6 : Quantile Regression Results w/ FS - 10th Quantile. 399
Table C.7 : Quantile Regression Results - 20th Quantile. 405
Table C.8 : Quantile Regression Results w/ FS - 20th Quantile. 413
Table C.9 : Quantile Regression Results - 40th Quantile. 419
Table C.10 : Quantile Regression Results w/ FS - 40th Quantile. 427
Table C.11 : Quantile Regression Results - 80th Quantile. 433
Table C.12 : Quantile Regression Results w/ FS - 80th Quantile. 441
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 447
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 448
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 449
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 450
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 451
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 452

Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 453
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 454
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 455
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 456
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 457
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 458
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 459
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 460
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 461
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 462
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 463
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 464
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 465
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 466
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 467
Table C.13 : Point BiSerial Correlation for Qualitative Variables..... 468
Table C.14 : Pearson Correlation for Quantitative Variables..... 469
Table C.15 : Percentage of Missing Values. 492
Table C.16 : Highly Correlated Variables Above rho>.80..... 494
Table C.17 : Feature Importance based on Training Gradient Boosting Models..... 496
Table D.1 : InD Chapters N-grams of SMPCR - 1-grams. 599
Table D.2 : FAI Chapters N-grams of SMPCR - 1-grams. 601
Table D.3 : MPR Chapters N-grams of SMPCR - 1-grams..... 603
Table D.4 : InD Chapters N-grams of SMPCR - 2-grams. 605
Table D.5 : FAI Chapters N-grams of SMPCR - 2-grams. 607
Table D.6 : MPR Chapters N-grams of SMPCR - 2-grams..... 609
Table D.7 : InD Chapters N-grams of SMPCR - 3-grams. 611
Table D.8 : FAI Chapters N-grams of SMPCR - 3-grams. 613
Table D.9 : MPR Chapters N-grams of SMPCR - 3-grams..... 615

LIST OF FIGURES

Page

Figure 2.1 : A Decision Tree of a Game. 24
Figure 2.2 : Reinforcement Learning 31
Figure 2.3 : Regression Solution. 35
Figure 2.4 : Gradient Descent. 36
Figure 2.5 : Gradient Descent Towards Local Minima 38
Figure 2.6 : Bias-Variance Tradeoff Representation 41
Figure 2.7 : Single Layer Artificial Neural Network. 45
Figure 2.8 : Multi Layer (Deep) Artificial Neural Network. 46
Figure 2.9 : Architecture Design of a Neural Network. 51
Figure 2.11 : Weight averaging procedure 52
Figure 2.12 : RNN Architecture within Space and Time. 54
Figure 2.13 : Backward Pass in RNN Architecture. 55
Figure 2.14 : Monotonic Functions. 59
Figure 2.15 : Linear Function. 61
Figure 2.16 : Piecewise Linear Function. 62
Figure 2.17 : Threshold (Unit Heaviside / Binary / Step) Function. 63
Figure 2.18 : Sigmoid Function. 64
Figure 2.19 : Bipolar Sigmoid Function. 65
Figure 2.20 : Rectified Linear Unit (ReLU) Family. 66
Figure 2.21 : Exponential Linear Unit (ELU). 67
Figure 2.22 : Odd Activation (Signum / Sign) Function. 69
Figure 2.23 : Softsign, Hyperbolic Tangent and Elliot Function. 70
Figure 2.24 : Arc Tangent Function. 72
Figure 2.25 : Complementary Log-Log Function. 73
Figure 2.26 : Softplus Function. 74
Figure 2.27 : Bent Identity Function. 75
Figure 2.28 : Soft Exponential Function. 76
Figure 2.29 : Sine and Cosine Wave Functions. 78
Figure 2.30 : Sinc Function. 79
Figure 2.31 : Fourier Transform for Continuous Series. 80
Figure 2.32 : Fourier and Its Variants by Domain. 83
Figure 2.33 : Periodic Gaussian Function. 84
Figure 2.34 : Square Wave Function 85
Figure 2.35 : Triangle Wave Function. 86
Figure 2.36 : Sawtooth Wave Function 87
Figure 2.37 : Logistic Function. 91
Figure 2.38 : Graphical Illustration of the τ^{th} Quantile. 94
Figure 2.39 : Quantile Regression ρ Function. 98

Figure 3.1 : Unemployment Rates in Turkey. 107
Figure 3.2 : Quadratic Experience and Earnings. 110
Figure 3.3 : Solution Diagram of Gale-Shapley Algorithm. 119
Figure 4.1 : Household Savings Over Years. 161
Figure 4.2 : Missing Value Fractions. 165
Figure 4.3 : Collinearity of Variables [Top 26]. 167
Figure 4.4 : Variable Importance Factor Loadings [Top 32]. 168
Figure 4.5 : Deep Neural Networks Architecture. 170
Figure 4.6 : Deep Quantile Regression Results w/o FS. 171
Figure 4.7 : Deep Quantile Regression Results w/ FS. 172
Figure 5.1 : An LSTM Unit. 191
Figure 5.2 : Inside of an LSTM Unit. 192
Figure 5.3 : Word Cloud for 'Inflation Developments' Chapter in SMPCMs. 194
Figure 5.4 : Word Cloud for 'Factors Affecting Inflation' Chapter in SMPCMs. 194
Figure 5.5 : Maximum Number of Items in SMPCMs by No. 195
Figure 5.6 : Word Cloud for 'Monetary Policy and Risks'Chapter in SMPCMs. 196
Figure 5.7 : Number of Words counted in SMPCMs by No. 197
Figure 5.8 : Number of Different Words used in SMPCMs by No. 199
Figure 5.9 : Sentiment Scores of "Inflation Developments" Chapter in SMPCMs. 201
Figure 5.10 : Sentiment Scores of "Factors Affecting Inflation" Chapter in SMPCMs. 201
Figure 5.11 : Sentiment Scores of "Monetary Policy and Risks" Chapter in SMPCMs. 202
Figure 5.12 : Total Sentiment Scores of SMPCMs. 202
Figure A.1 : Precision vs Recall. 246
Figure A.1 : Collinearity of Variables [Full Heatmap] 493

ESSAYS ON ESTIMATION METHODS

SUMMARY

In this study, new estimation methods have been investigated as the title of the dissertation suggests. These methods, at the time of writing this dissertation, are classified in Artificial Intelligence. Three articles have been completed for this purpose. These articles are not only new in terms of estimation methods, but also have many innovations in terms of data sets that are created and used. In addition, the methods used are also compared with econometric methods.

The methods to be used in all the articles have been examined in detail in the first chapter. This chapter consists of two main sections. In the first section, the methodologies used in this study but mostly by artificial intelligence are examined. In the second section, econometric methodology to be used for comparison are examined.

The first article is about estimating the likelihood of individuals transitioning from unemployment to employment. For this purpose, Household Labor Force Surveys in between 2004-2016 are utilized by matching the questions and the answers semantically which are changed year by year. On the side of machine learning, since the problem is described as classification, estimation methods and their algorithms used for this purpose selected in accordance with classification. Again, classification is examined mostly in binary choice models, logistic regression is the most appropriate econometric methodology. The results indicate that XGBoost and Random Forest classifiers in ML are slightly better than logistic regression.

Second article tries to establish a forecasting, as well as prescriptive, model for savings ratio for both individuals and households. Here, Household Budget Surveys at both individual and household level are matched and merged starting from 2002 to 2016. Since most features can be used for estimation, the data is turned into high dimensional and relatively bigger one. Due to the distribution of saving ratio is normal and skewed to the lower levels, quantile regression is thought as the best method to deal with that issue. Later, deep neural networks which has a quantile loss function, normal regression loss function and a quantile regression are run to generate a model. However, as the results indicate, DQRNN is superior to DRNN and normal equation which is also costly, time-consuming and hard to estimate.

In the last article, a very experimental attempt is applied to Summaries of Monetary Policy Committee Reports issued in 2006 - 2018 in order to understand quantitatively the Central Bank of the Republic of Turkey in a very standardized way for each economic agent. In order to do it, a sentiment analysis is conducted and a predictive Recurrent Neural Network model specifically called Long Short-Term Memory is provided to predict semantics of each report with high accuracy.

All in all, these articles provided in this dissertation, in which both methodologies and the data need more effort are presented for use by economic agents, institutions, researchers and academicians with their open sourced codes and methods. The data in the first and the second articles are limited to use by TURKSTAT only and are not published, however, the third article is published. All materials can be found in this website, dissertation.yasinkutuk.com, for at least 3 years.

Finally, the author is responsible for all errors in this thesis, in addition, any opinion reflects only the author's position entirely.

TAHMİN YÖNTEMLERİ ÜZERİNE MAKALELER

ÖZET

Bu çalışma içerisinde, tez adının da gösterdiği gibi, yeni tahmin yöntemleri ve uygulamalarına yönelik çalışmalar gerçekleştirilmektedir. Burada bahsedilen yeni yöntemlerin hepsi, tezin yazıldığı dönemde oldukça gelişen yapay zeka alanında yaygın olarak kullanılmakta, otonom robotlar, ses işleme, görüntü tanıma, otomatik karar verme, doğal dil işleme, algoritmik finans, robotik ve büyük veri uygulamalarının temel yapı taşlarını oluşturmaktadır. Tabi bu çalışma sadece yöntembilimsel olarak değil, verilerinin elde edilmesi, işlenmesi ve bütünleştirilmesi anlamında da oldukça yeni yöntemler sunmaktadır. Bu doğrultuda büyük veri, yüksek boyutlu veriler, metin verisi ve bunların derlenerek karmaşık[11] yapılardan düzenli[12] veriler elde edilmesi ve bunların sayısal bir dile dönüştürülmesi gibi pek çok yenilikler ihtiva etmektedir.

Bu doktora tezi, bir giriş olarak algılanabilecek ve kullanılan tüm tahmin yöntemlerini içeren bir yöntembilim bölümü ve bunu takip eden üç uygulamalı makale ile tümüyle dört kısımdan oluşmaktadır. Yöntembilim bölümü kapsamlı bir sözlük ile birlikte ve kullanılan yöntemlerin matematiksel olarak detaylı anlatımını ve fonksiyonel düzeyde de görselleştirmelerini dahi içermektedir. Bu anlamda sonraki çalışmalar için bir kılavuz olarak kullanılabilmesi amaçlanmıştır.

Giriş olarak kabul edilebilinecek ilk bölüm iki kısımdan oluşmaktadır: tümüyle, sonraki üç makalede uygulama imkanı bulmuş, yapay öğrenme yöntemlerinin bulunduğu ilk kısmı, ikinci aşamada ekonometrik yöntemler takip etmektedir. Kaynakça anlamında daha fazla referansın yer aldığı ekonometrik kısmı görece kısa tutulmakla birlikte, yapay öğrenme kısmı, tam bunun aksine, daha detaylı olarak sunulmuştur. Yapay öğrenme yöntemlerinin anlatıldığı ilk kısımda, önce, yapay öğrenme algoritmaları kısaca anlatılmış, bir yapay öğrenme algoritmasının temel elementleri olarak kabul edilebilinecek maliyet fonksiyonu, eğim algoritması, regresyon ve sınıflama problemleri, sinir ağları, öğrenme algoritması, ileri ve geri besleme yöntemleri, katlamalı ve yinelemeli sinir ağları, etkinleştirme fonksiyonları detaylıca bir şekilde incelenmektedir. Buna ek olarak, ilk iki makalede kullanılacak ekonometrik yöntemleri tanıtmak amacıyla da ikinci kısım eklenerek lojistik ve kantil regresyon yöntemleri detaylı olarak anlatılmaktadır.

Yukarıdaki yöntembilimin uygulandığı üç makalenin yer aldığı bölümlerde ise ilkin işsizlikten istihdama geçiş olasılıklarının tahmini ekonometrik ve temel sınıflandırma yapay öğrenme yöntemleri işlenmiş, ardından tasarruf oranları,

[11] Messy data.
[12] Tidy data.

ilk makaleye benzer şekilde ekonometrik ve derin öğrenme yöntemleri ile tahminlenmiş, son olarak Türkiye Cumhuriyeti Merkez Bankası (TCMB) tarafından yayımlanan Para Politikası Raporları kullanılarak doğal dil işleme yöntemleri yinelemeli sinir ağları ile incelenmiştir.

İşsizlikten istihdama geçişin incelendiği ilk makalede, bireylerin bazı temel karakteristik, eğitimsel ve deneyimsel verileri, öncelikle 2004-2016 yılları arasında gerçekleştirilmiş Türkiye İstatistik Kurumu'nca (TÜİK) yayınlanan Hanehalkı İşgücü Anketleri (HİA) derlenerek elde edilmiştir. Ancak bu derleme, soruların ve yanıtların yıllar boyunca farklılıklar göstermesi nedeniyle hem anlamsal hem de mantıksal olarak eşleştirilmesini zorunlu kıldığından, el yordamıyla bir birleştirme gerçekleştirilmiş ve bu kodlanarak yedi milyon civarındaki bireysel anket verisi bir büyük veri çatısı altında birleştirilmiştir. Bunlardan bireylerin bulundukları cari dönemdeki istihdam bilgisi ile ilgili en ufak bir ipucu içermeyen veriler ayrıştırılmış, sadece geçmiş dönemde işsiz olarak kabul edilen bireylerle veri seti sınırlandırılarak 203891 kişinin verisi işlenmiştir. HİA'nın gerçekleştirildiği döneme göre tam bir yıl öncesinde, ILO tanımı gereği işsiz olarak kabul edilen bireylerin, o dönemde istihdama geçme veya işsiz olarak kalma olasılıkları tahmin edilmiştir. Yapay öğrenme algoritmaları için kullanılan eğitim veriseti %90 ile sınırlandırılmış ve ayıklanırken yıl bazında katmanlı rassal K-Fold algoritması ile veriler seçilmiştir; test verisi, kalan kısım ile belirlenmiştir. Kullanılan yapay öğrenmenin sınıflama algoritmaları karşısında ekonometrik olarak olasılık tahmini yapabilecek lojistik regresyon tahminleri de gerçekleştirilmiştir. %67 oranında başarılı tahminler üreten sınıflama algoritmalarından XGBoost ve Rassal Orman, ancak %63 başarım sağlayan lojistik regresyon yöntemini geçerek, bu probemi en iyi çözümleyen yöntemler olmuşlardır. Bu sayede, çalışmada sunulan ve ek olarak verilmiş, bireylere dair temel 38 özelliğe dair bilgi edinmiş herhangi bir politika yapıcı, ekonomik aktör veya araştırmacı, işsiz olan bir bireyin bir sonraki dönem istihdama geçebilme olasılığını, %67 oranıyla tahminleyebilecektir.

Son dönemlerde, gelişmekte olan ülkelerin pek çoğunda görüldüğü gibi, Türkiye'nin de en önemli sorunlarından biri olan tasarruf oranları, ikinci çalışma içerisinde incelenmektedir. 2002-2016 yılları arasında TÜİK tarafından yayınlanan Hanehalkı Bütçe Anketleri (HBA), soru ve cevapları dikkate alınarak büyük bir veriseti oluşturulmuştur. Eşleştirme bireysel bazda gerçekleştirilirken, hanehalkına ait veriler bu bireylerin satırlarında tekrarlatılmıştır. HBA verileri çok fazla soru içerdiğinden, yıl bazında da soru ve cevaplarda değişikliklere gidildiğinden, mümkün olan en azami şekilde tüm veriler korunmaya çalışılmış ve yüksek boyutlu büyük bir veri, 629372 satırda 247 öznitelikle elde edilmiştir. Bu özniteliklerden bazıları, yüksek korelasyon içerdiğinden dolayı çıkarılarak ve bazı yeni değişkenler türetilerek son olarak 371613 satırda 203 öznitelikle sonuçlandırılmıştır. Bu büyüklükte verileri işlemek için kullanılacak herhangi bir ekonometrik yöntem, varyans-kovaryans matrisinin tersinin alınacağı aşamada kitlenebilmesi muhtemel olmasına rağmen, karşılaştırma yapabilmek için, kantil regresyon yöntemiyle tahmin edilmiştir. Alternatif olarak, bu büyüklükte ve bu kadar fazla öznitelik içermesine rağmen hem daha kısa sürede hem de çok yüksek tahmin yüzdesi ile kantil regresyon maliyet fonksiyonuna dayalı derin

öğrenme metodu geliştirilerek dört katmalı bir sinir ağı modeli kurulmuştur. Derin öğrenme uygulaması için ise, ilk makaledekine benzer bir yöntemle eğitim veriseti %90 olarak belirlenmiş, yıl bazında katmanlı rassal K-Fold algoritması ile seçilmiş, kalanı test veriseti olarak kullanılmıştır. Toplamda dört adet olarak belirlenen (.10, .20, .40 ve .80) kantiller için ayrı ayrı koşturulan kantil regresyon maliyet fonksiyonuna dayalı derin öğrenme algoritması 10.000 adetlik küçük demetlerle yaklaşık 2 saat 34 dk.'da eğitilebilirken, normal denklem yöntemiyle hesaplanan kantil regresyon yaklaşık 20 saatte sonuç üretebilmiştir. Ancak sonuçlara bakıldığında ise derin öğrenme algoritması %93 oranında başarım sağlayabilirken, %15'i geçemeyen normal denklem kantil regresyonu, katlanılan maliyete uygun bir başarımın çok altında kalmaktadır. Tüm modeller ayrıca değişken seçimi ve mühendisliği yapılarak tekrar edilmiş ancak yüksek başarım elde edilememiştir. Kantil regresyonuna dayalı derin öğrenme yöntemi, hem gösterdiği yüksek başarım, hem kısa zamanda uygulanabilme açısından bundan sonraki tasarruf fonksiyonlarını tahmin etmek isteyen politika yapıcılarına, araştırmacı ve akademisyenlere iyi bir araç olarak sunulabilmektedir.

Bu tez çalışmasının gerçekleştirildiği döneme kadar, oldukça kısıtlı sayıda kalmış araştırma konularından birisi de, ekonomi alanında metin işlemeye dayalı yöntemlerin uygulama alanı bulduğu makale sayısının son derece kısıtlı kalması, dahası, Türkiye'de bu alanda bilinen bir çalışmanın bulunmamasıdır. Dolayısıyla literatürde hem bu açığı kapatacak, hem de istatistiksel doğal dil işleme yöntemlerinin kullanılabileceği, verileri işlemede oldukça etkili bir tahmin yöntemi olduğu konusunda bir önayak olabilmek amacıyla, Türkiye Cumhuriyeti Merkez Bankası (TCMB) tarafından, genel olarak aylık yayımlanan Para Politikası Kurulu (PPK) Kararları Özet Metinleri, İngilizce olarak incelenmiştir. Bu metinler, PPK toplantılarının gerçekleştirildiği günden sonraki beş işgünü içerisinde yayımlanmakta (2011 yılından itibaren), toplantılar ise aksi bir ekonomik gelişme olmadığı sürece bulunulan ayın 4. Perşembe günleri, saat 14:00'da gerçekleştirilmektedir. toplantılara dair tam metinler yerine, üç temel başlık altında bu özet raporlar TCMB'nin sitesinde PDF olarak sunulmaktadır. Bu üç başlık, genel olarak, "Enflasyon Gelişmeleri" (1), "Enflasyonu Etkileyen Unsurlar" (2) ve "Para Politikası ve Riskler" (3) olarak sıralanmaktadır. Birinci başlık altında geçmiş dönem enflasyonda meydana gelen değişiklikler incelenirken, ikinci başlıkta genel olarak geçmiş dönemdeki değişiklikleri etkileyen faktörler detaylıca incelenmektedir. Son başlıkta ise, bunlardan farklı olarak makro konjonktürde meydana gelen gelişmeler incelenmektedir.

Teknik olarak hazırlanan bu raporların genel olarak bir sistematik ierisinde hazırlandığı, başlıklardaki ufak değişiklikler haricinde ciddi bir değişikliğin olmadığı, her bir raporun içerisinde yer alan paragrafların konu bütünlüğünü değiştirmeyecek şekilde maddeler halinde verildiği, maddelerde işlenen cümlelerin maddenin konu bütünlüğünden çıkmadığı, üç başlık altındaki maddelerin ise o başlığa son derece sadık kalınarak hazırlandığı görülmektedir. TCMB hem politika belirleyici rolü, hem yazılı veya sözel her aksiyonuyla sinyal verici, hem de itibari değeriyle ülkenin bir prestiji olması dolayısıyla, bu raporlardaki özen korunmuş, şeffaflık gereği kamuoyu ile paylaşılmıştır.

Bu doğrultuda 20 Temmuz 2006'da 2006-39 No ile yayımlanmış ilk özet rapor ile 14 Haziran 2018'de yayımlanmış 2018-24 No'lu son özet rapor arasında kalan toplamda 129 PPK özet raporu, yazılan bir derleyici ile PDF ortamında TCMB web sayfasından çekilmiş, düz metin[13] dosyalarına dönüştürülmüştür. Ardından veriler, başlıklar ve numaralar altında listelenerek, en çok kullanılan kelimeler 1-gram, 2-gram ve 3-gram altında çıkarılarak birer zaman serisi olarak çıkarılmış, ayrıca 1-gramlar kelime bulutu olarak her üç başlık için resimlendirilmiştir. Buna göre (1). başlıkta fiyatlar, (2). başlıkta talep, (3). başlıkta ise enflasyon en çok kullanılan kelimelerdir. Ardından her PPK özetinin numaraları ve kullanılan kelime sayıları çıkarılmış, iş döngülerinin gerçekleştiği dönemlerde TCMB'nin daha yoğun kelime kullanımına gittiği ve raporlarda daha fazla madde listelediği gösterilmiştir. Ardından istatistiksel doğal dil işleme yöntemleri ile WordNet veritabanı kullanılarak her bir cümleye anlamsal duygu analizi yapılmış, başlık veya rapor bazında kümülatif indeks değerleri oluşturulmuştur. Bu anlamda yine belirtilen iş döngüleri dönemlerinde TCMB'nin daha pesimist bir tonda konuştuğu ve piyasaya seslendiği görülmüştür. Bu indekslemeden sonra, veritabanı cümle boyutuna indirgenerek, zaman serisi yaklaşımıyla raporlar birbirine bağlanmış ve Uzun Kısa-Süreli Bellek adı verilen yinelemeli sinir ağı modeli kurularak %93 doğruluk payı ile gelecek dönemdeki raporların anlam skorlarının tahmin edilebilmesine yönelik sağlam bir model oluşturulmuştur. Bu model ile, piyasaya verilen tonun kestirilmesinde son derece kullanışlı olmakla birlikte, anlamsal analiz ile TCMB'nin PPK kararlarına yönelik standart bir anlam skorunun elde edilebilmesi sağlanmıştır.

Nihayetinde, gerek uygulama gerekse kendi verisetini oluşturma zorlukları açısından oldukça önemli yenilikler içeren ve başaran bu tezin tüm aşamaları ekonomi disiplini içerisindeki tüm ilgililere, piyasa yapıcılarına ve politika geliştiricilerine sunulmak amacıyla açık kaynak olarak gerçekleştirilmiştir. İlk iki makalede yer alan verisetlerinin yalnızca TÜİK tarafından sunulabiliyor olması nedeniyle, sadece üçüncü çalışmanın veriseti, kullanılan tüm kodlar, en az üç yıl süresince dissertation.yasinkutuk.com adresinde bulunabilecektir.

En son olarak, bu tezdeki tüm hatalardan yazarın kendisi sorumludur, herhangi bir görüş, tamamen yazarın tutumunu içermektedir.

[13] Plain text.

1. INTRODUCTION

1.1 The Importance of Prediction Power in Economics

Irving Fisher, one of America's greatest economists, predicted that equities in the US Economy had reached a "permanently high plateau" in October 1929, however, stock market collapsed in two weeks and was not able to recover at least 25 years .

Nobel Laureate Samuelson inferenced from macroeconomic performances of the Soviet economy and asserted that it can be regarded a proof where socialism command may function and thrive. Passing one year was a clear answer for that claim since Soviet economy was collapsed.

A chief strategist from Goldman Sachs made prediction which is a resemblance of Fisher's and suggested that S&P 500 would grow at least 14% in one year in December 2007. One year later, it felt approximately 36%. Margin of error is about 50%.

The above examples can be expanded with a number of similar examples. In the economics literature, researchers made similar predictions in their all empirical studies. Moreover, over the last 50 years, a large majority of finance columnists have only written their estimates. The accuracy of ongoing estimates has perhaps never been controlled.

Economics has always been obliged to offer a policy proposal. Therefore, one of the most important activities is to offer an available recipe. This recipe is sometimes used to indicate useful tools, and sometimes not to do. Sometimes the prescription is so evident that a mutual commitment between policy tools is mathematically determinable. To be more concrete, there is a mutual relationship between causes and results. The most powerful area of this relationship can be called "positive science". Friedman (1953) argues that the ultimate goal of positive science is to produce predictions. Friedman,

[0]Of course, it does not cause not to believe in well-known Fisher Equation.

who also sees economics as a science in its entirety, claims methodically as predictive and suggests a concept called predictivist thesis. Predictivist thesis makes a tentative acceptance or rejection decision about the hypotheses proposed. While Friedman likens economics to natural sciences, he states that economics should be defined as positive economics. Positive economics [1] establishes hypotheses and tests them with appropriate methods. The result is a tentative acceptance or rejection. According to this, hypotheses are turned into theory and the power of the theory is strongly correlated with its predictive power.

It was first mentioned by Hicks (1984) that economics was on the edge of science and on the edge of history. Hicks (1984) completely denies that natural sciences and economics are in convergence; it suggests that there may be rather than a science, but a discipline. Hicks, therefore, defines economics as quasi-science. Economics, like an experimental environment with controlled conditions, lacks the principles of physics, such as reproducibility, producing the same results. Moreover, any science related to human activity lacks the production of such things. Economics, in particular, produces theories based on an unrealistic precondition such as ceteris paribus. Prediction is used completely different from natural sciences. Also, there are two types of predictions which may be conditional or unconditional, the first one is prediction of what will happen, the other is that of what will probably occur in the future. So, the second led to born econometrics the field mostly uses theory of probability. However, by means of econometrics, the discipline of economics can generate certain causal relationships.

According to Hicks, the events in the past can be modeled in the future, with stochastic models. Stochasticity can be regarded as an effect of human activity in economic problems. The reasons for the big inflation in the 1970s are a good example in this sense. There is no known example in the past; it has not been seen before. Moreover, it is a human-centered event that is caused by

[1] Hutchison (2013) makes a clear distinction between positive economics which can be called applied economics and policy economics which may be called political economy. This distinction was clarified later by Buchanan (1987) and was re-emphasized by Uygur (2006).

many economic decision makers who are unaware of each other. It will not make sense to explain it only by a single variable. There are numerous factors that explain this single variable. Econometrics may help to explain the effect of money supply on inflation, but the job of it does not finish, the main causes of fluctuation in money supply should also be explained.

One of the most deep-rooted criticisms of Friedman (1953) was realized by Sen (1986). He also defined -maybe the first in economics- prescriptive statistics or prescription concept which will be used frequently in machine learning literature. Prescription is an important tool for economics. For this, pure prediction is not enough and should be taken a step further. The good and the bad has to be evaluated and assesed by economics, actually political economics which is a older version of economics and related to mainly prescription.

Sen (1986) argues that, for prediction and its accuracy, economics will face more difficulties than physics when searching for solutions to its own problems. Therefore, it suggests that ontological elements are incomplete except for methodological and epistemological factors. And then, Sen (1986) bases his arguments to make economic predictions on Marshall who said before "the two economic events are not exactly similar". Therefore, it may not only use predictive analyses, in that sense Sraffa (1962) provides non-predictive economic tools. Because of the relationship between human behavior and social, political, biological and psychological factors, it is more difficult to make economic forecasts. As a way of overcoming these difficulties, Sen (1986) produces a different solution from Hicks, suggesting that some basic concepts (equilibrium, maximization and rationality) have to be revised. It cannot be said that it is not possible to provide the theories formed by these concepts with prediction. Apart from this, as suggested by Friedman (1953), prediction is at the center of economics, but it is a difficult issue to solve. Rather than predicition, economics do not provide any satisfactory option in comparison with other fields.

In the same way, Simon (1989) describes a fork in a way for the status of prediction. Its main purpose is prediction as the area in which economics and

economic theories will be tested. Simon, in general, rejects this statement because the main purpose of a branch of science is not just prediction. It would be very weak. He is, therefore, very cautious against the hypothesis put forward by Friedman. It does not completely oppose, but it is not right that a discipline prioritizes economics, only prediction. Secondly, Simon (1990) is cautious that economics as a science sees prediction as a test tool. Understanding the mechanisms within economic theories is more important than predictability, according to Simon. However, these mechanisms may not always provide a good estimation tool.

Simon (1990) goes a step further and tries to define economics as a whole. Economics is actually a complex system which requires modelling to understand the behaviour and the dynamics inside of it. Modeling in economics, according to him, is a basic system of understanding a complex system with simple principles. Unlike complex systems, there is chaos which is quite difficult to solve. Economics is a complex system in this sense, but we do not have a clear knowledge of whether it is a chaos. Complex systems can produce simple randomness while chaos which is a statistical condition presents complete uncertainity and unpredictability more than complex systems. Chaos theory suggests that prediction is now, in the future, and lifelong impossible. Economics is a bit chaotic in its essence and requires understanding.

Simon (1990) continues by expressing the definitive distinction between prescription and prediction. Simon explains “prediction” with the example of the Club of Rome report. In there, certain sociological and demographic predictions were made on false premises. Almost all mechanical designs contain an indispensable prediction. However, prediction has two undeniable parts which are theoretical understanding to design a mechanism and phenomena that can be extrapolated.

Prescription is a policy-making tool. It is used to understand what will occur as a result of an action. Prescription establishes a network of relationships between factors. It can be deduced that an event occurring in a factor can affect the elements in the entire network. Economics can therefore try to be

prescriptive as well as predictive. If a good modeling can be done, prescriptive statistics can work very well. This helps prediction.

So far, in economic theory, it is seen that a characteristic definition of prediction cannot be made exactly and clearly. Therefore, a generally accepted methodology has not been developed. In general, testable implications that belong to future, testable implications which are unknown at the time and only testable implacitions are the three definitions which are a bit distinct to another according to González (1998). Friedman mostly understands as testable implications which are unknown at the time. Simon and Sen supports testable implications that belong to future. Hicks understands only as testable implications which is closer to the scientific one.

The distinction between prescription and prediction is clearly made by Simon (1989, 1990). As Sen (1986) argues, presription is one of the most important activities of economics, along with prediction. Where prediction is inadequate, prescription should enter the circuit. Prescription is a recipe for what should be done and what should not be done. Sometimes it can be supported by prediction but it cannot be rebutted. As Simon argues, it is the way to understand a mechanism, transform chaos into a complex system. At least complex systems can be modeled. When complex systems are modeled, they can be used as a cross-policy. If prescription will not be produced and prediction will not be done, econometrics will not do much good. Removing large models from a small sample will cause us to produce tentative results. This is just like modeling a part of a chaotic structure. Chaos is always temporary because it contains pure randomness.

By the 1990s, a clear distinction between prediction and prescription had been made. Methodologically, prediction power was seen as an important but insufficient criterion. Economic methodology was considered a prescriptive tool and was surely supported by a predictive study. By adopting this methodology, it was used as a basis in many economics articles. Besides, significant developments have been made to increase the prediction power. In these developments which have been going on since the 1970s, the place of econometrics is quite large and undeniable. Besides, White (1992) may be

considered as the first attempt in econometric history to empower predictions by using artificial neural networks instead of use of econometrics in economic methodology. However, at this stage, the most intractable problem for predictive modeling is the disconnection between prediction and causality.

Prediction and causation in empirical studies in economics literature empowered with econometric models are mixed with together conceptually. The variables carefully selected to strengthen the power of prediction power are considered explicitly interrelated to dependent variable. Kleinberg et al. (2015) asks a fairly simple question about this false thought: a researcher working rain is investigating the dances for the rain and the possibility of using umbrellas because of the rain. Rain dances require a causality analysis for the rain. However, carrying an umbrella depends only on the estimation of the probability of rain, the causality may not be used. As methodologically accepted, all prescriptive studies supported by prediction and policy recommendation regard dependent and independent variables used in econometric studies as related. However, this is not the case. Opening an umbrella does not increase the likelihood of rain. Similarly, rain-dance and rain cannot be interconnected by causality. Therefore, prescription actually may create results that are suitable for policymakers to produce false policies. Especially in the case of high-dimensional data, researcher or policy recommender may be able to offer more false policy recommendations.

A second problem is the economics damage caused by the margin of error. Prescription, which is not supported by a strong prediction, can produce faulty results in the short or long term. For example, Rockoff et al. (2011) have shown how to achieve efficiency by employing the right teacher is chools. In a similar way, Kleinberg et al. (2017) claim jurisdictional decisions could be effective; they intended to decrease the cost of crime to judge system. Chandler et al. (2011) contribute to the development of highly predictive and preventive social policies in order to prevent frequent school massacres in the United States. Therefore, increasing the predictive power not only makes prescription strong but also contributes to the increase of social policy.

A third problem is that it may be wrong to produce theory from prediction and prescription. A prediction is made from a certain number of datasets according to the economic method developed above. Therefore, the prescription produced is only possible with the size and temporal limit of this dataset. However, the predictive success can be measured by testing the generated model in other data. In an analysis that does not include econometric and machine learning, Giannakouros and Chen (2015) produce solutions in a randomly extracted dataset through the dataset; the performance of the remaining data is recommended to measure. Athey and Imbens (2015) likewise suggest to divide the dataset into two appropriate parts, produce estimates and models from the first part, and to test the performance of model on the other part. This results in better performance.

Linearity can be counted as fourth problem in economics in which the problems have complexity, which is called economic complexity. Economic complexity can be defined as the chunk of certain facts or behaviors which are not determined well by economic theory. In quantitative terms, economic complexity is hidden in interactions between economic, social or other kind of variables, discontinuities in economic problems due to structural changes, multilayered possible solutions in observed data and so on (Delorme et al., 2010). In econometric terms, they are generally hidden in error term which may be quite large than explained. Then, economic complexity can be resulted from dependency in context, path or structure. On the other hand, econometrics can help to analyze these problems by adding non-linearity for the solution, where linearity does not work. However, added non-linearities should be exogeneous rather than endogeneous, that is, most of non-linearities can be captured by manually adding variables which are their squares, square roots, geometric average of two variables etc. Again, (the number of) possible combination of these transformations, non-linearities and interactions between independent varibles are limited for human mind. Neural networks in machine learning models, especially deeper networks rather than shallow ones, can easily capture non-linearities simultaneously with modeling explicitly and endegonously (without human intervention) these

non-linearities with its all possible combinations. Therefore, the difficulties in the interpretation of the model are overcome and the predictive power is increased by the algorithm against economic complexity in the problem.

The fifth problem is the difficulty of modeling. Many models in econometrics are made from an existing sample. Estimates and predictions are valid within this model and the range of data being processed. However, a good model should also be strong outside its own data. Therefore, a model can be considered as successful if it can produce strong predictions within another dataset, except that it is processed. Many econometric models are actually quite backward. Egami et al. (2018) show how spliting works well even causal relationship should be generated in accordance with what Giannakouros and Chen (2015) propose data splitting in adaptive regression to solve numerical solution for economic problems. However, according to Ziliak and McCloskey (2008), econometric inference is more academic rigourous than other modeling tools even the data is so predetermined, small and narrow.

Looking at the final problem, it is seen that econometrics may be methodologically insufficient in some cases. However, a more important criticism comes from the statistical front which is a base for econometric methods. Ziliak and McCloskey (2008) evaluating statistical significance in detail, which is a very impressive book, consists of five main parts: the myth of statistical significance, significance in economics, practices in other fields, history of significance and solution. According to Ziliak and McCloskey (2008), statistical significance that is a measure of sampling variance is commonly misused. In the first part, Ziliak and McCloskey (2008) state that most scientists treat statistical significance as scientific significance and statistical insignificance as scientific insignificance. As a whole, the Fisherian thinking about statistical significance as scientific significance is elaborated and criticized. Science takes into account loss functions, Type II errors, rather than the chance of finding a difference, Type I error. This is ,frankly, thinking against Popper (2005)'s falsification principle. For economics, the situation is worse than it is thought. After examination of 19 viewpoints

determined by American Economic Review for reviewing process, many of criterias are related to the condition for random sampling, reproducible experiments or stochasticity. However, nearly all data do not satisfy these conditions. This problem is not only an epistemological one, it is also against to the principle of science which aims to get results by chance. Researchers eliminate the logic of producing the results for change from the very beginning in their studies. At this point, Ziliak and McCloskey (2008) refer Haavelmo (1944) that asserts most of economic phenomena have stochastic characteristics instead of determined characteristics, which will be repeated by Simon (1990) after him. They claims that The difference between deterministic and stochastic phenomena is not as important as epistemological differences between the stochastic and indeterminate phenomena. Stochastic and uncertain phenomena have nothing in common. In conclusion, economics should give up to violate the standards of statistical persuasion [2].

Considering the six problems mentioned above and the inaccuracies of statistical methods, it is seen that economics should be more open about new methods. Varian (2014) states that machine learning and big data can produce very efficient, predictive and accurate results in solving economic problems and in generating economic or social policies. Kleinberg et al. (2017) and here [3] emphasize that machine learning provides such useful tools to generate appropriate policies in social sciences. Varian (2014) argues that ML models are far behind in terms of prediction, but this problem has been completely overcome. It has been suggested that the development of predictive models and the expected results of the policies that can be applied and the causality relationships can be revealed more clearly. Therefore, the causation between causality and prediction is closer to elimination. If data are sufficiently large (big), the third problem mentioned above will make it possible to produce economic theory from practice (Athey, 2017). Athey and Imbens (2017) recommend machine learning tools that make causality more explicit. In this way, the most suitable covariates will be selected and the prediction power

[2]For the solution, firts, "The economics profession should set meaningful standards of economic significance", second statistical textbooks and courses should be revised

[3]https://hbr.org/2016/12/a-guide-to-solving-social-problems-with-machine-learning

will be increased. Belloni et al. (2012) and Chernozhukov et al. (2017, 2018) indicate that it becomes more important to select appropriate covariates in a high dimensional environment to increase the power of prediction with more suitable features. Athey and Imbens (2017) continues that the credibility of economic policies will be enhanced by empowering predictive methodology with machine learning. Besides, Mullainathan and Spiess (2017) claim that machine learning tools maintain different solutions and help to uncover patterns from a complex structure which is not identified in a great detail.

Indeed, much earlier, as Mitchell (2006) suggests, machine learning tools can be a good candidate to solve many economic problems. Mitchell (2006) argues that by designing a system that learns how pareto optimal distribution can be achieved, self-interested individuals can be modeled as a set and that this system can always learn. There are also experimental studies which have succeeded in this direction in the economy. For example, Glaeser et al. (2018) have estimated income for New York City by using and converting image data in Google Streets Maps by using image recognition algorithms, a highly revolutionary article for economics. Moritz and Zimmermann (2016) offer an alternative method to past-return-based prediction of future returns. Galindo and Tamayo (2000) put forward an outperformed machine learning classifier instead of Probit algorithm to predict succesfully financial risks. Dubé and Misra (2017) focuse on third-degree price discrimination for firms which are implemented on their price determination, and looks for the implication of this on welfare. Giannone et al. (2018) compare predictions in macroeconomics, microeconomics and finance, conludes that economic data do not provide enough information to identify relationship between variables if there exist large data to analyze. Björkegren and Grissen (2017) catch the behavioral signatures in mobile phone data predict loan default to reduce the effect of a possible damage to the economy of misused loans. Apart from these predictive papers, Igami (2017) discusses Deep Blue's calibrated value function, which works a powerful superforecaster, can change some econometric assumptions made by conditional choice probability estimation, which will be used to correct models used in many economic forecasting

problems. Begenau et al. (2018) compare firms that use big data/machine learning and conventional methods in economic context, then concludes that using better prediction improving investors' forecasts and reduces equity uncertainty, reducing the firm's cost of capital, will put firms forward, which dominate the market than smaller firms or other firms that are not using these technologies.

There are also criticisms against machine learning algorithms. The first criticism is that the field is very new. Serious steps have been taken in this direction. Recently, it can be clearly observed that the number of machine learning applications has increased significantly. The total number of packages issued in different programming languages (regardless of the programming language) has tripled according to Kdnuggets [4]. The second important criticism is that the machine learning algorithms are not so few in human intervention as they are believed. Moreover, this is very decisive. This is quite accurate and rather a criticism. However, a good modeler can also optimize their hyperparameters. Thirdly, there is no good machine learning methodology in time series and panel data models. Time series problems have been dealt with solutions thanks to recurrent neural network and temporal convolutional neural network applications. However, as it is claimed, the panel data methodology is still lack of suitable solutions in comparison with econometrics.

As a result, not only economics but many disciplines are intertwined. Perhaps the most interdisciplinary studies on this issue are seen in the economic literature. Economics that evolves into inclusive development should seek to include a methodology that is now more predictive, well-structured prescriptive and drawing accurately causal relationships. In the last eight years, it has been seen that important breakthroughs that affect the masses have been obtained as a result of these scientific studies. The German national team was the champion of the FIFA World Cup in Brazil in 2014 and the winners of the British League in the 2016 Premier League thanks to prescriptive statatistics done by deep learning. Similarly, the work done on

[4]Full report is available here :https://www.kdnuggets.com/2018/02/top-20-python-ai-machine-learning-open-source-projects.html

social media in Obama and Trump elections is an effective result of machine learning. The market of superforecasters and predictors is currently US$ 8.8 billion in 2016, but in 2025 it will reach US$ 18.2 billion. There is no doubt that the number of studies that increase the predictive power in the economy will increase continuously. Athey (2018) predicts that this effect will occur in a much shorter time than expected. This study aims to shorten this period for economists. Machine learning is a new and powerful tool that can be used in conjunction with econometrics to increase predictive power, enable social policies to be more efficient and cost-effective, and lead to the creation of prescriptive methodology, as well as to reveal causal relationships that are difficult to define (Agrawal et al., 2018).

In this dissertation, it will be introduced a fairly new machine learning estimation methods and the econometric estimation methods for deriving new solutions in economics. Then, two experimental articles will be presented in comparison with econometric models to machine learning models, and later, one article that only consists of deep learning model which cannot be implemented in econometrics will finish this dissertation. All these three articles which also consider prediction power as an influential argument are written to open a way in economics.

2. METHODOLOGIES of ESTIMATION METHODS

In recent years, it can be easily observed that statistics has been renewing itself by means of recent advances in both the processing and the data. Processing technologies are not restricted to the CPU itself but encompasses recent developments in other technologies such as GPU, TPU and other unclassified processors as singular or parallel processing units. The data has also been growing day by day[1] with the emergence and widespread use of smartphone technologies which has accelerated, firstly, data creation and then collection and its storage compared to other resources. These two components of the technology provide a great, convenient basis for the birth of Zeitgeist [2]: Artifical Intelligence (AI hereafter).

AI actually has a long history which can be traced back to the Mechanical Turk (Standage, 2002), a chess machine, controlled by a human who is hidden in the machine playing against other humans as an intelligent and fake-automated machine thinking moves. The breakthrough in AI was made by Alan Turing when trying to break and translate the codes used by axis forces in the Second World War, which made AI an academic discipline in the late 1950's. This breakthrough was based on categorical and logical implementation of codes, symbolic reasoning and mathematical expressions to imitate a human brain. The first neural network implementation called SNARC was carried out during those years McCorduck (1979). Russell et al. (1995) assert that AI is achieved by changing the way of thinking. It has been achieved by making computers more intelligent and learnable than just codes, which feeds computers consistently from the environment or feeds them with data created or provided by human. So, the utopia of aiming to create a machine thinking, acting and feeling like a human has become a reality thanks

[1] https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#7a16b1917b1e

[2] As frequently stated or referred, the concept of zeitgeist is attributed to Hegel (1861), but he used only "the spirit of the age" which is originated as "der Geist seiner Zeit" in German.

to the changes in the way of thinking; switching from creating categorical or logical algorithms that shape problem solving behaviour of computers to computers using information-based learning. Hence, the information, namely data, became the source of learning.

The human brain learns the information which is provided by itself or by the environment. In order to make machines intelligent, the first thing which should be done is to mimic the algorithm of learning. Previously it was questioned if machines, after the problem was declared to them, can solve the problem if the problem exists and has been defined for the machine. The word is the key concept of the algorithms before the 21st century. After that, the learning could be built on whether the condition of "if" exists, as human brains do. So then, the machines are programmed to learn "if" statements' through their own capabilities, and the main condition for that is to feed into the machines the information, the data and implicit programming. This process as a whole is called machine learning. Computers can learn from the data, they can make their own "if statements" by generating equation(s), establishing a relationship between informations, seperating others etc. Machine learning is a discipline of learning from data in order to make predictions using them.

2.1 Machine Learning

The best and general definition of what machine learning may come from Mitchell (1997) explaining that "machine learning is a discipline of computer algorithms that improve automatically through experience". In the word experience, there is a hidden and a deep meaning that cannot be explained by stating the key word, the data. As Einstein once said "learning is experience, everything else is just information" as if he wanted to endorse the empiricist school (Zalta et al., 2003). Machine learning algorithms are binding these two sides in Einstein's quote by building a bridge to turn information into experience as emphasized in Mitchell (1997).

There are some concepts to know in order to understand machine learning comprehensively, a short Glossary in Appendix B has the basic definitions of them.

Table 2.1 : Supervised Learning.

Row	Income level	Age	Experience	Hourly Wage
1	Low	45	11	11.9
2	Low	26	4	14.5
3	High	61	30	37.2
4	Middle	35	9	19.6

Although it is hard to seperate into categories, machine learning can be examined in two ways. The first one is to clustering the types of machine learning, the second is grouping the main algorithms mostly used.

2.1.1 Types of machine learning

Types of machine learning are classified according to the data whether it has a label or not. So, the data intended to be used will play a great role in selecting a method or defining the problem into one of these methods. According to labeling, there are four types of machine learning problems: supervised, unsupervised, semi-supervised and reinforcement learning.

2.1.2 Supervised learning

The simplest and the most easy to understand type of machine learning can be supervised learning. Here, the data used as input is labeled (James et al., 2013). In Table 2.1, some individuals are labeled according to their income level as can be seen. So, in order to make a prediction from the data by using the variables or features, their labels play a significant role.

A machine learning algorithm should be prepared for the learning process that makes a prediction and corrects these predictions during this learning process. The training keeps iteration until the model achieves a desired level of accuracy on the training, validation or test data.

2.1.2.1 Regression

Linear regression may be used in linear or non-linear problems where the desired output (label) is a continuous or discrete variable. As the denotation provided here[3], the basic problem to be solved is, given a training set (x, y),

[3]Denotations -mostly as used by Andrew Ng :

* m : number of training examples
* x : “input" variables, or features

where $x = (x_1, \ldots, x_j)$ is a vector, and represents the (j) input features (independent variables) and y represents the output variable (dependent variable). It is intended to predict new ys given new xs.

For instance, let's say for some x, it is being tried to solve for y. There exists a dataset of x values and corresponding y results. In order to find the optimal model for this data, it is desired to predict the cost for prediction models and find the model with the least cost.

According to the data and the problem, seperation for the solution by using regression is needed:

Linear Regression: If regressand/response/dependent variable or output feature shows a linear relationship between independent/explanatory variables or features, then the best solution for this problem is linear regression. In here, SSE function $(y - XB)^2$ should be a convex function that can be represented as linear. If not, the solution for this kind of problem is different from here.

Non-linear Regression: SSE function may show non-convex properties. That is, the continuous response variable shows curvilinear relationship between at least one independent variable. These problems may contain multiple local minima. Due to nonlinearity between variables/features, the local minima can be achieved by using numerical optimization algorithm to determine coefficients. The most common non-linear regression forms are power, weibull and fourier equations.

Analytic Solution of Regression (Normal Equation) Analytic solution of regression, also called the normal equation method, is a way to solve for θ analytically, with no need for iteration and no need to choose α and scale

---

* y : "output" variable, or "target" variable
* $(x^{(i)}, y^{(i)})$: i-th training sample ($i \leq m$)
* n : number of features
* x_j : j-th feature ($j \leq n$, but remember to include x_0)
* θ : regression parameters
* h : hypothesis, (a mapping from x to y). E.g. $h_\theta(x) = h(x) = \theta_0 + \theta_1 x + \ldots$
* Note that $x_0 = 1$ (So that it is obtained as $h_\theta(x) = \vec{\theta}^T x = \theta_0 x_0 + \theta_1 x_1 + \ldots$)
* J : cost function to minimize. Regression is to minimize $J(\vec{\theta}) = \frac{1}{2m} \sum_{\forall i} \left(h_{\vec{\theta}}(x^{(i)}) - y^{(i)} \right)^2$

features. However, this would be slow if n is large, because it would be needed to compute the inverse[4] of a $n \times n$ matrix: $\left(X^T X\right)^{-1} \sim O(n^3)$. The idea behind the ML solution by using gradient descent is that it is hard to find $\left(X^T X\right)$ when the data is a relatively big one. So ML solution is classified as a computational solution of gradient descent.

Given a training set, the design matrix X is an $m \times (n+1)$ matrix (including the intercept term), where each row corresponds to one training example, and each column would correspond to a feature. Let y be the $m(\times 1)$-dimensional vector corresponding to the target values. It is defined as θ to be the $(1\times)(n+1)$-dimensional parameter vector.

Given $h_\theta(x^{(i)}) = (x^{(i)})^T\theta$, the m rows of $X\theta - y$ would be $[h_\theta(x^{(i)}) - y^{(i)}]$. Notice then that

$$J(\theta) = \frac{1}{2}||X\theta - y||^2 = \frac{1}{2}(X\theta - y)^T(X\theta - y). \tag{2.1}$$

Taking the derivative:

$$\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta} &= \nabla_\theta \frac{1}{2}(X\theta - y)^T(X\theta - y) && (2.2)\\
&= \frac{1}{2}\nabla_\theta(\theta^T X^T X\theta - y^T X\theta - \theta^T X^T y + y^T y) && (2.3)\\
&= \frac{1}{2}\nabla_\theta \mathrm{tr}(\theta^T X^T X\theta - y^T X\theta - \theta^T X^T y) \quad [5] && (2.4)\\
&= \frac{1}{2}\nabla_\theta(\mathrm{tr}\theta^T X^T X\theta - 2\mathrm{tr}(y^T X\theta)) \quad [6] && (2.5)\\
&= \frac{1}{2}(X^T X\theta + X^T X\theta - 2X^T y) \quad [7] && (2.6)\\
&= X^T X\theta - X^T y. && (2.7)
\end{aligned}$$

Thus, setting the derivative to zero, it is analytically found that

$$X^T X\theta = X^T y \tag{2.8}$$

$$\theta = (X^T X)^{-1} X^T y \tag{2.9}$$

[4]Note that what if $\left(X^T X\right)$ is non-invertible? One possibility is that there exist redundant (usually linearly dependent) features, and another possibility is that there are too many features ($m \leq n$); first solution coming to mind is to delete some features, or to use regularization.

[5]Trace of a real number is itself / $\nabla_\theta y^T y = 0$

[6]$trA = \mathrm{tr}A^T$

[7]$\nabla_A trABA^T C = B^T A^T C^T + BA^T C$

The equation in 2.8 finds regression coefficients (Greene, 2003) that minimize residuals (errors) of the equation and satisfies the best linear unbiased estimator (Puntanen and Styan, 1989) [8].

2.1.2.2 Classification and logistic regression

Another problem that needs to be solved is the classification problem in which the data has a labeled/output variable that are not continuous, but a discrete, more specifically binary/dichotomous. In such cases, linear methods will perform standard model prediction. This model is actually the possibility of the realization of the dichotomous variable. However, in linear models, this probability increases or decreases linearly with the dependent variable. Even the total probability may exceed 1. Since it is impossible to estimate, specifically the probability of occurence of something by using linear methods, the most appropriate method for them is logistic classifiers. This is so similar in econometrics the discipline used to estimate the probability of occurence as a discrete choice by using generalized linear methods family. In short, logistic classifiers are;

* used to solve classification problems (desired output: classifier variable)
* the most proper methods since Linear regression not advised for classification. First, it is sensitive to outliers/skewed data sets, second, probability is not linear and cannot exceed 1.

  Logistic function is also regarded as a sigmoid function: $g(z) = \frac{1}{1+e^{-z}}$

  - used to solve classification problems (desired output: classifier variable)
  - the most proper methods since Linear regression not advised for classification. First, it is sensitive to outliers/skewed data sets; second, probability is not linear and cannot excess 1.

    Logistic function is also regarded as sigmoid function: $g(z) = \frac{1}{1+e^{-z}}$
  - Hypothesis: $h_\theta(x)$ = estimated probability that $y = 1$ on input x, or $h_\theta(x) = P(y = 1|x;\theta)$

[8] That property is known as BLUE.

- Predict $y = 1$ if $h_\theta(x) \geq 0.5$, or since $h_\theta(x) = g(\theta^T x) \implies 1 \geq \theta^T x \geq 0$
- Linear regression cost function cannot be used as $g(x)$ is non-linear and the cost function will be a non-convex function.
- Thus, the following cost function is used:

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

- Intuition: if $y = 1$, penalize more as $h_\theta(x) \to 0$, and if $y = 0$, penalize more as $h_\theta(x) \to 1$
- Because y can only take on values of 0 or 1, it can be written:
- $\text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$
- $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}\left(h_\theta(x^{(i)}), y^{(i)}\right) =$
  $-\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right) \right]$
- $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$
- Decision boundary “separates" variable space into two decision regions. Can be linear or non-linear.
- Advanced Optimization
- There are other optimization algorithms besides gradient descent.
- E.g. Conjugate gradient; BFGS; L-BFGS.
- Advantages: No need to choose α, and often faster than gradient descent.
- Drawbacks: More complex, need more time, may need GPU.
- For multi-class classification,
- One-vs-all: For each class i, train a logistic regression classifier $h_\theta^{(i)}(x)$ to predict probability that $y = i$.
- On new input x, to make a prediction, pick i that maximizes the classifier $h_\theta^{(i)}(x)$

Instance-based Learning Machine learning models which are instance-based generally create similarity (or dissimilarity) measurements to calculate representative power of new instances when they

append to the data. This type of learning is also called as memory-based learning since it redraws the hypothesis to make a new distance or similarity index by recalling old instances. It is mostly used in recommendation systems. Certain types of instance-based machine learning algorithms are:

- k-Nearest Neighbor (kNN),
- Learning vector quantization (LVQ),
- Self-organizing map (SOM),
- Locally weighted learning (LWL),
- Radial basis function network.

Bayesian Learning There are just two laws of probability needed to know. The first is the *sum rule,* also called the *rule of total probability.* Suppose X is some event (an event is simply a collection of things that could be true, for example "the total value shown on two dice is less than six"), and suppose $\{Y_i\}_{i=1}^{N}$ are some mutually exclusive and exhaustive events ("for example, Y_i could be the event "the first rolled die has the value i," for $i \in \{1,2,\ldots,6\}$). Then:

$$\Pr(X) = \sum_{i=1}^{N} \Pr(X, Y_i), \tag{2.10}$$

so the total probability of X is the sum of the probabilities of X occurring alongside each of the scenarios described by the $\{Y_i\}$.

The second law of probability to know is the *product rule:*

$$\Pr(X, Y) = \Pr(Y \mid X) \Pr(X). \tag{2.11}$$

Here the conditional $\Pr(Y \mid X)$, read "the probability of Y given X," is the probability of event Y when it is restricted only to cases where X is true. This is easy to see from a Venn diagram. By manipulating the product rule, it is arrived at *Bayes' theorem:*

$$\Pr(Y \mid X) = \frac{\Pr(X \mid Y) \Pr(Y)}{\Pr(X)} \tag{2.12}$$

$$= \frac{\Pr(X \mid Y) \Pr(Y)}{\sum \Pr(X \mid Y) \Pr(Y)}. \tag{2.13}$$

One possible way to interpret this result is as follows. Suppose one is interested in the truth of Y. She/he begins with a prior belief about Y, $\Pr(Y)$. He/she then learns that X is true. The formula above is used to update our belief about Y by conditioning on this new information, giving $\Pr(Y \mid X)$. Bayes' theorem therefore gives a probabilistically consistent way to update one's beliefs given new information. In this context, the value $\Pr(Y)$ is called a *prior probability*, as it represents our beliefs prior to observing X. The output, $\Pr(Y \mid X)$ is called the *posterior probability.*

The process of updating beliefs in this way is often called *probability inversion,* because after observing X, it is first calculated $\Pr(X \mid Y)$ ("how likely was it to observe X if Y were true?"), then inverted using Bayes' theorem to give the desired probability ("how likely is Y now that I've seen X?"). Many classical paradoxes arise because these probabilities can be quite different from each other, depending on the prior $\Pr(Y)$.

What is the prior probability that the outcome of rolling two dice is less than six? What if I told you that the value of the first die was a five, does that change anything?

The above results are only valid for discrete random variables X. Although this can be useful in machine learning (for example, if these variables correspond to different hypotheses one wishes to compare), in practice, one will often be interested in continuous variables, such as θ above. Thankfully, the above formulas are perfectly valid for continuous random variables x and θ, where it can be replaced with probabilities Pr with probability density functions p and replace sums with integrals:

$$p(x) = \int p(x,y)\,\mathrm{d}y; \tag{2.14}$$

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{\int p(x \mid y)p(y)\,\mathrm{d}y}. \tag{2.15}$$

One will always assume that probability density functions exist, because the cases where people don't are not typically encountered in machine learning. It also saves anyone from having to state "when the density exists" all the time.

Returning to the survey example, one might begin with a prior belief about the value of θ, represented by a prior probability distribution $p(\theta)$. To couple the observations $\mathscr{D}$ to the value of interest, it is constructed such a probabilistic model $\Pr(\mathscr{D} \mid \theta)$, which describes how likely one would see a particular survey result $\mathscr{D}$ given a particular value of θ. Note that this model could have any form and one is free to make it as complicated.

Finally, one uses these to compute the posterior probability of θ given the survey results, $p(\theta \mid \mathscr{D})$. This posterior distribution encapsulates the entire belief about θ! He/she can use it to answer various questions he might be about θ.

The Bayesian Method There are four main steps to the Bayesian approach to probabilistic inference:[9]

- **Likelihood.** First, the likelihood (or *model*), $p(\mathscr{D} \mid \theta)$ should be constructed . This serves to describe the mechanism generating our observations $\mathscr{D}$ given a particular value of the parameter of interest θ.

- **Prior.** Next, the individual's prior beliefs about the parameters θ is summarized , which is encoded via a probability distribution $p(\theta)$.

- **Posterior.** Given some observations $\mathscr{D}$, we obtain the posterior distribution $p(\theta \mid \mathscr{D})$ using Bayes' theorem.

- **Inference.** Now, it is time to use the posterior distribution to draw further conclusions as required.

Here are some examples of Bayesian learning:

- Naïve Bayes (NB),

- Gaussian Naive Bayes,

- Multinomial Naïve Bayes,

[9]These are summarized from Tony O'Hagan and Jonathan Forster's eloquent introduction in *Kendall's Advanced Theory of Statistics Volume 2B.*

· Averaged One-Dependence Estimators (AODE),

· Bayesian Belief Network (BBN),

· Bayesian Network (BN).

Naïve Bayes

$$\Pr(Y \mid X) = \frac{\Pr(X \mid Y)\Pr(Y)}{\Pr(X)} \tag{2.16}$$

Naïve Bayes uses the formula given above Equation 2.16 to discriminate between classes, which is the first algorithm coming to mind in Bayesian implementation in ML. It is easy to implement a Naïve Bayes classifier on a dataset. However,
Con: Usually outperformed by other algorithms (e.g. SVMs),
Con: Assumption of independence.

Laplace Smoothing However, one problem with the afore-mentioned algorithm is that it reacts poorly to novel stimuli. When presented with a novel x that carries no information about y, $p(x|y=1)$ and $p(x|y)=0$ are both 0.
If, for example, it is always be seen $y=1$ for m observations of x. With the $(N+1)$-th observation of x, one would expect that the best estimate of $P(y=1)$ is:

$$\begin{aligned} P(y=1) &= \frac{P(y=1)}{P(y=1)+P(y=0)} \\ &= \frac{m}{m+0} = 1 \end{aligned} \tag{2.17}$$

This might not be the most optimal, because it does not take into account the possibility that on the next observation, y might equal 0 [10]. Laplace smoothing “adds" one to each value:

$$P(y=1) = \frac{P(y=1)+1}{[P(y=1)+1]+[P(y=0)+1]} \quad \text{with Laplace smoothing} \tag{2.18}$$

$$= \frac{(m+1)}{(m+1)+(0+1)} = \frac{m+1}{m+2}$$

[10] Laplace originally used this to predict whether the sun would rise tomorrow.

More broadly, for a multinomial variable x that takes on one of k values, given m observations, the maximum likelihood estimate (MLE) (of it taking on a value $j \leq k$) without and with Laplace smoothing is:

$$\phi_j = \frac{\sum_{i=1}^{m} \mathbb{1}\{x^{(i)} = j\}}{m} \qquad \text{Without Smoothing} \tag{2.19}$$

$$\phi_j = \frac{\sum_{i=1}^{m} \mathbb{1}\{x^{(i)} = j\} + 1}{m + k} \qquad \text{With Smoothing.} \tag{2.20}$$

Multivariate Bernoulli Event Model

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} \left(\mathbb{1}\{y^{(i)} = 0\} \sum_{j=1}^{n} \mathbb{1}\{x^{(i)} = k\} \right)}{\sum_{i=1}^{m} \mathbb{1}\{y^{(i)} = 0\}} \tag{2.21}$$

Above is a representation of semantic clustering of documents where,
LHS: Prob of seeing word k in negative examples.
Numerator: # of occurrences of word k in negative examples.
Denominator: Total # of words in negative examples.

Decision Tree Learning In game theory, there are two types of game representation, first is a table, second is extensive form. Extensive form games can be classified as decision trees as represented in Figure 2.1. It simply shows actions made by actors.

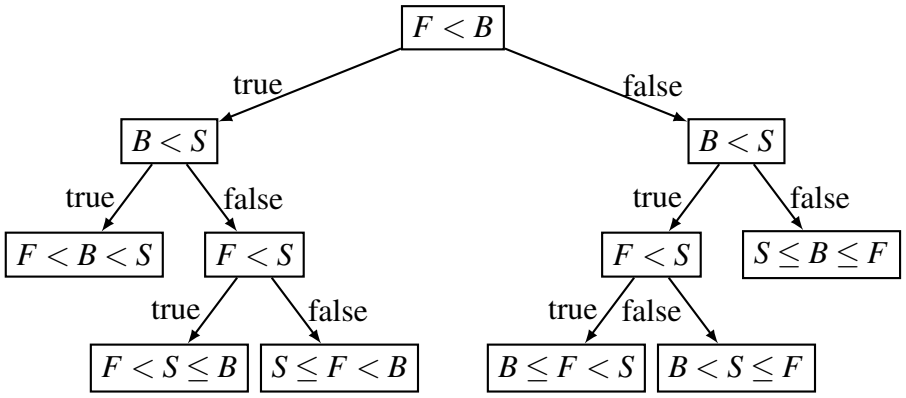


Figure 2.1 : A Decision Tree of a Game.

F:Forward, B:Backward, S:Stay

In machine learning literature, decision tree learning is based on the type of predicted output which can be a dichotomous [11] or a continuous

[11] Or a class variable which can be subsetted as a dichotomous.

feature. This may turn the problem into classifier trees or regression trees (CART). But, as a subdiscipline, decision tree algorithms are based on CART, AID, CHAID, ID3 and ID 4.5 [12]. All of these types of algorithms used for decision tree learning split the datasets into smaller parts in the learning process and then the labeling process starts again with a learning parameter. The observations which are split into wrong tree branches (subsets) are the residuals for the algorithm. An instance of decision tree algorithm is given below:

Algorithm 1: Tree Growing `treeGrowing`

Input : Training set X, attribute set A, target feature y
Output: Decision tree

Create a new tree T with a single root node.
if `stoppingCriterion(`X`)` **then**
    Mark T as a leaf with the most common value in X as a label.
else
    $\forall a_i \in A$ find a that obtain the best `splittingCriterion(`a_i, X, y`)`.
    Label n with a.
    for *each outcome* v_i *of* a **do**
        Set subtree $t_i = \sigma_{a=v_i} X, A, y$
    .
    Connect the root node of n_t to the subtree t_i with an edge that is labelled as.
return `treePruning(`S, T, y`)`

2.1.3 Unsupervised learning

In this type of machine learning problem, the data has no output or label (Friedman et al., 2001). The first aim is to regenerate the data structure or redraw the distribution that is targeted. Since there are no correct labels or outputs to estimate the accuracy of the machine learning model, it is unrealistically hard to make sure of its prediction power. In Table 2.1, suppose that the labels in the feature "Income level" are unknown, this turns it into the problem that should be solved with unsupervised learning. The first thing to do in order to generate income level groups,

[12] CART is classification and regression trees. AID is automatic interaction detection performing stepwise splitting in the tree. CHAID is Chi-Squared AID based on χ^2 distribution. ID3 is Iterative Dichotomiser 3 creating multiway tree. ID 4.5 is enhanced ID3 by yielding dynamic redifiner of discrete features

is to use other features to remap the clusters, but first, the association between other features could be examined to make clustering easy.

2.1.3.1 Clustering

It is said that this is the first unsupervised learning algorithm. In unsupervised learning problems, there only exists an input data $x^{(i)}$ with no labels, and it is intended for the algorithm to find some structure in the data. A clustering algorithm such as the k-means algorithm attempts to group the data into k “clusters" which have some similarity. Some examples include: Market Segmentation, Social Network Analysis, Organizing Computer Clusters, and Astronomical Data Analysis.

First, it could be done to choose how many clusters are wanted, k[13]. This could be done by “eyeballing" the data, or from *a priori* hypotheses about how many clusters there should be. k-means clustering algorithm consists of two parts. First, it is initialized with the cluster centroids $\mu_1, \ldots, \mu_k \in \mathbb{R}^n$ randomly. One way is to set the cluster centroids to be equal to a random subset of the training examples. Note that k-means can get stuck in local optima, so one solution is to try multiple times with different random initializations.

Next, iterative steps are used to first assigning each example to the cluster centroid closest to it (usually the 2-norm), and then updating each cluster centroid to be the means of all the points assigned to it[14]. In pseudo-code, this would be:

$$\text{For every } i, \text{ set } c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2 \tag{2.22}$$

$$\text{For each } j, \text{ set } \mu_j := \frac{\sum_i^m \mathbb{1}\{c^{(i)} = j\} x^{(i)}}{\sum_i^m \mathbb{1}\{c^{(i)} = j\}} \tag{2.23}$$

$$\tag{2.24}$$

Let define the **distortion function** (akin to a cost function):

$$J(c, \mu) = \sum_i^m ||x^{(i)} - \mu_j||^2 \tag{2.25}$$

[13] It should go without much saying that the number of clusters k should be less than the number of training examples m.

[14] If a cluster has no points, one way to deal with this is to eliminate that cluster.

The inner loop on k-means minimizes J (with respect to c holding μ fixed, and with respect to μ holding c fixed), and hence is equivalent to coordinate descent, so k-means is guaranteed to converge. As usual, in order to evaluate the convergence, J can be plotted as a function of the number of iterations and a “cutoff" defined after which we declare the algorithm to have converged. Note that, similar to previously discussed cost functions, J is a non-convex function however, so the algorithm might not converge to the global minimum.

Going back to choosing the number of clusters, one way is using the “elbow method": J could be plotted against k; J will decrease rapidly until some point, and then decrease more slowly afterwards, and a good choice would be that point at which is starts to decrease more slowly (“elbow"). Unfortunately, most of the time there might not be a clear or unambiguous “elbow".

Another way is to evaluate k-means based on a metric for how well it performs for that later purpose. E.g. if it is being studied for optimal T-shirt sizing, say $k = 3$ (S,M,L), vs $k = 5$ (XS,S,M,L,XL), one way to choose k is to balance how well your shirts will fit (higher k) vs. the additional cost of making more sizes.

2.1.3.2 Dimensionality reduction

Dimensionality reduction or dimension reduction mainly aims to reduce the set of features used in machine learning to provide the best suitable set to increase accuracy of prediction of output. It can be divided into two parts: feature extraction and feature selection.

In feature selection, there are many ways to select features which are the most relevant ones to predict output. The first and most common way is to create a variance matrix of all features with output variable and then to choose the highest variances. Second way is to look at the correlation matrix as done in econometrics.

In feature extraction, the dimensions are enhanced by creating feature space which can be a linear combination of them or nonlinear

derivatives of features to subset fewer dimensions. Another approach for multidimensional data is to convert them into tensor[15] data.

Principal Component Analysis Principal Component Analysis (PCA) is another dimensionality reduction algorithm that can be used to find structure in our data. The main aim is to find a surface onto which the projection errors are minimized. This surface is a lower dimensional subspace spanned by the **principal components** of the data. These principal components are the directions (linear combinations of the input dimensions) along which the projections of the data onto that axis have the maximum variance. The main component along which the data varies is called the principal axis of variation.
The PCA algorithm[16] basically tries to maximize the variance of the projection of the data x along a vector u.

$$\max_{u:||u||_2=1} \frac{1}{m}\sum_{i}^{m}(x^{(i)^T}u)^2 = \max_{u:||u||_2=1} \frac{1}{m}\sum_{i}^{m}(u^T x^{(i)})(x^{(i)^T}u) \tag{2.26}$$

$$= \max_{u:||u||_2=1} u^T \left(\frac{1}{m}\sum_{i}^{m} x^{(i)}x^{(i)^T}\right) u \tag{2.27}$$

This means that u is the principal eigenvector[17] of $\Sigma = \frac{1}{m}\sum_{i}^{m} x^{(i)}x^{(i)^T}$, the covariance matrix of the data (if it has zero mean), subject to the constraint $||u||_2 = 1$. To see that u is the principal eigenvector of Σ, it can be written with the Lagrangian for the optimization problem:

[15] See Appendix B.

[16] Note that there is an additional pre-processing, normalization step that is needed for PCA. (Mean normalization and feature scaling)

$$\mu = \frac{1}{m}\sum_{i}^{m} x^{(i)}$$
$$\sigma^2 = \frac{1}{m}\sum_{i}^{m}(x^{(i)} - \mu)^2$$
$$x \leftarrow \frac{x-\mu}{\sigma}$$

[17] Recall that $Au = \lambda u$ means that u is an eigenvector of A with eigenvalue λ. The eigenvector with the largest eigenvalue is the principal eigenvector.

$$\mathscr{L}(u,\lambda) = u^T\Sigma u - \lambda(u^T u - 1) \tag{2.28}$$

$$\frac{\partial \mathscr{L}}{\partial u} = \Sigma u - \lambda u = 0 \implies \Sigma u = \lambda u \tag{2.29}$$

This can be repeated to get the second, third, etc. principal components. If it is intended to project the data onto a d-dimensional subspace ($d < n$), then we can choose $u_1, \ldots, u_d$ to be the top d eigenvectors of Σ. The new representation would be $y^{(i)} = \{u_1^T x^{(i)}, \ldots, u_d^T x^{(i)}\} \in \mathbb{R}^d$[18].

Some uses of PCA include data compression by getting rid of redundant or correlated dimensions, or enabling visualization of high dimensional data by reducing a high dimensional data set to 2-3 dimensions.

Another use of PCA is to preprocess data before using a supervised learning algorithm[19]. Reducing the number of features would reduce memory storage and reduce the time taken to run supervised learning algorithms. This does **not** prevent overfitting. Dimensionality reduction is not a good way to do this – regularization should be used instead. Also, the algorithm should be run without PCA first, and then only if there is a reason to (e.g. the algorithm is running too slowly), add PCA.

If noise needs reduction, one way to choose the number of principal components is to pre-define them according to some theory that you might have. Another way to choose is to set a threshold for the variance in the data that needs to be explained. For example, if it is defined that the total variation and averaged squared projection errors to be:

$$\text{Total variation in the data} = \frac{1}{m}\sum_{i=1}^{m} ||x^{(i)}||^2 \tag{2.30}$$

$$\text{Averaged squared projection error} = \frac{1}{m}\sum_{i=1}^{m} ||x^{(i)} - y^{(i)}||^2 \tag{2.31}$$

where ($y^{(i)}$ is the reduced representation), then d can be choosen, the number of principal components, so that some threshold (say 95%) of

[18] An easy way to do this is to compute the eigenvectors of matrix Σ, say in Matlab: $[U,S,V] =$ svd(Sigma). The columns of U are the vectors we want. U is an $n \times n$ matrix, where the i-th column is the i-th principal component. Hence, we can just take the first d columns. $U_{\text{reduce}} : n \times d$, $y = U_{\text{reduce}}^T X$, where $y \in \mathbb{R}^d$, our reduced dimension representation.

[19] An important tip: the dimensionality reduction mapping should be defined by running PCA only on training set. The mapping can be applied as well to the cross-validation and test sets.

the variance is retained[20], i.e.

$$\frac{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}-y^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}||^2} \leq 0.05 \tag{2.32}$$

2.1.4 Semi-supervised learning

Semi-supervised data, as can be understood from its name, covers a mix of labeled and unlabeled data (Friedman et al., 2001; Haykin, 2009). In general, these mixed data for semi-supervised machine learning consist of mostly unlabeled ones when comparing its proportion to the amount of labeled data. It may be done in the nature of the data, which can be deal with in the semi-supervised learning. But somehow, labels of the data might be collected after other features which have been completed. In order to save time, with decreasing the cost of the data of labels which are missing, semi-supervised learning will do it with a remarkable success.

2.1.5 Reinforcement learning

In economic terms, reinforcement learning needs a data shaped among the environment, players, and moves which can be rewards or punishments (Haykin, 2009) according to the set of strategies, all of the concepts recall game theory. The setting of reinforcement learning is not the only part similar to game theory, but the goal is also quite similar to game theory, the actions played must get maximized rewards at the end of the play, that is, the data generating process.

The data being created dynamically can be exemplified by a chess game. As drawn a chess in Figure 2.2, the white player is the first mover. After that, black player plays f6 against white's e4 move. Cumulative rewards can be maximized by minimizing the pieces lost to other player or maximizing the pieces kept to win the game. If the pieces has no equal points affecting the rewards, second problem should be turned into minimization of cumulative points of pieces. Ng et al. (2006) design

[20]From the Matlab command $[U,S,V]$ = svd(Sigma), it can be obtained S, a diagonal matrix of elements S_{ii} from $i=1$ to $i=n$. For a given d, the fraction of variance left is simply: $\frac{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}-y^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m}||x^{(i)}||^2} = 1-\frac{\sum_{i=1}^{k}S_{ii}}{\sum_{i=1}^{n}S_{ii}}$

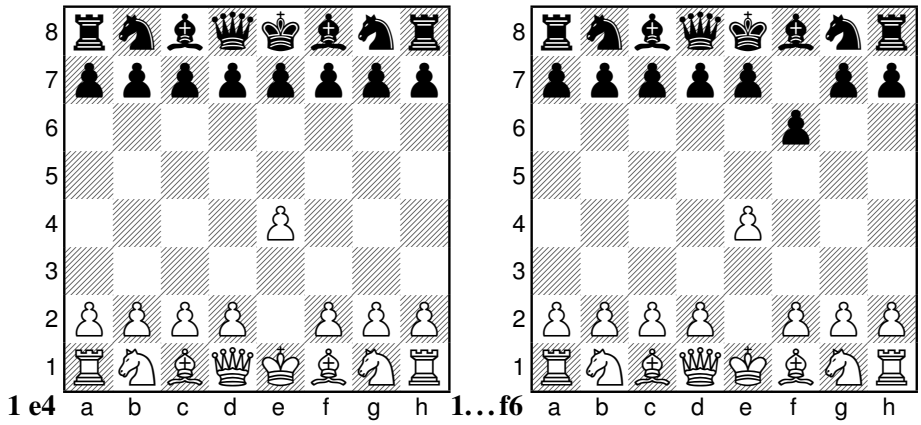


Figure 2.2 : Reinforcement Learning

a good and innovative implementation of reinforcement learning as an example.

2.1.6 Federated learning

Learning methods studied previously are closely related to the type of data. Federated learning, however, relies on the shape of data. As McMahan et al. (2016) proposed, the data can be stored into small pieces and decentralized, however, the learning updates can be done by local computers so that make the machine learning model is as correct as possible. The first attempts come from Google which collects the data from smartphones to make smarter than its initial settings by using android os and trains the models at the center, after a while, it sends the trained model by updating initial settings of the OS.

2.1.7 Transfer learning

Like federated learning, transfer learning is a newcomer in the field of machine learning. A model can be trained by using data which has certain attributes (features, variables). When the model reaches a desired accuracy, it is accepted as learned weights and adjusted parameters as the learning process is finalized. A new data which is not the continuation of the old one, however, the new one provides similar settings, collection methods and especially attributes compared to older one. Finally, the learning algorithm can be transferred into the new data with small adjustments which shortens the period rather than starting a new learning process. Dai et al. (2007) assert labeling tasks can be the most convenient jobs for being a subject of transfer learning that boosts learning and are less costly than unsupervised learning.

2.1.8 Ensemble learning

Ensemble learning is mostly suitable for the problems classified in supervised machine learning, where there must be at least two or more complex machine learning algorithms to estimate one single hypothesis. However, ensemble learning can give multiple outputs which may be mixed to increase accuracy of prediction. In machine learning literature,

there are two main common ensemble methods, bagging and boosting (Dietterich, 2000). The last follower is stacking.

Bagging: The first method for ensemble is bagging which stands for bootstrap aggregation. It is a way to reduce the variance of an estimate by averaging all estimates. Bagging, at first, generates bootstrap sampling to create subsets for training base learners. Then, bagging votes for classification and weighting for regression.

Boosting: The second methods is called boosting. Boosting transforms weak learning algorithms to stronger ones. Sequentially estimated weak learners are weighted. Then, a majority vote classification or weighted sum rule are run to produce the final estimate.

Stacking: The other method is stacking which is abbrevated from stacked generalization. There is a meta learner that learns the output of a sample from the outputs of the base learners in stacking. The base models are trained according to whole training set. Later, meta learner is trained on the outputs of base model as independent variables (features). The algorithm then summarizes stacking.

2.1.8.1 Random forest

The Random Forest algorithm as first proposed by Breiman (2001) is a so-called ensemble method. This means that the model consists of many smaller models, but predictions and other quantities of interest are obtained by combining the outputs of all the smaller models. There are many ensemble methods that consist of various sub-models. The sub-models for Random Forests are classification and regression trees (CART). Breiman (2001) proposed bagging to decrease the variance of fitted values from CART. This innovation is used to reduce the risk of overfitting. Later Meinshausen (2006) makes a generalized model for random forest which is actually a special case of quantile random forest.

2.1.8.2 Extreme gradient boosting

Extreme Gradient Boost (abbrevated as XGBoost) is a framework that optimizes gradient descent algoritm in an ensemble way proposed by Chen and Guestrin (2016). It uses trees to generate a scalable boosting system that generates state-of-the-art results. For the nodes of leaves in decision trees, XGBoost assigns negative or positive values. Later, it establishes a weak learner in initial step. Then it starts to learn the data by assuming the data is seperated with a random guess.

2.2 Machine Learning Algorithms

As studied in the Chapter 2.1.1, the tasks for machine learning taking into account only algorithms they used, can be separated into its methods adapted in the algorithm. Most and widely used algorithms are regression, instance-based ones, decision trees, Bayesian methods, clustering methods, ensemble and neural networks.

2.2.1 Gradient descent

This is done this by first forming a hypothesis for the prediction model given below as a simple linear regression:

$$h_\theta(x) = \vec{\theta}^T x = \theta_0 x_0 + \theta_1 x_1 + \varepsilon. \tag{2.33}$$

The hypothesis (regression equation) links the predicted value of y to the independent variables x. The *Gradient Descent* (GD) method tries to minimize the gradient of the error of this hypothesis [21] by evaluating Equation 2.47.

$$h_\theta(x) = \theta_0 + \sum_{j=1}^{n} \theta_j x_j = \theta^T x \tag{2.34}$$

$$\text{GD = derivative of } (h-y) : J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \tag{2.35}$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\vec{\theta}) \tag{2.36}$$

[21] In econometrics, this is the ordinary least squares model (OLS) that aims minimization of RSS.

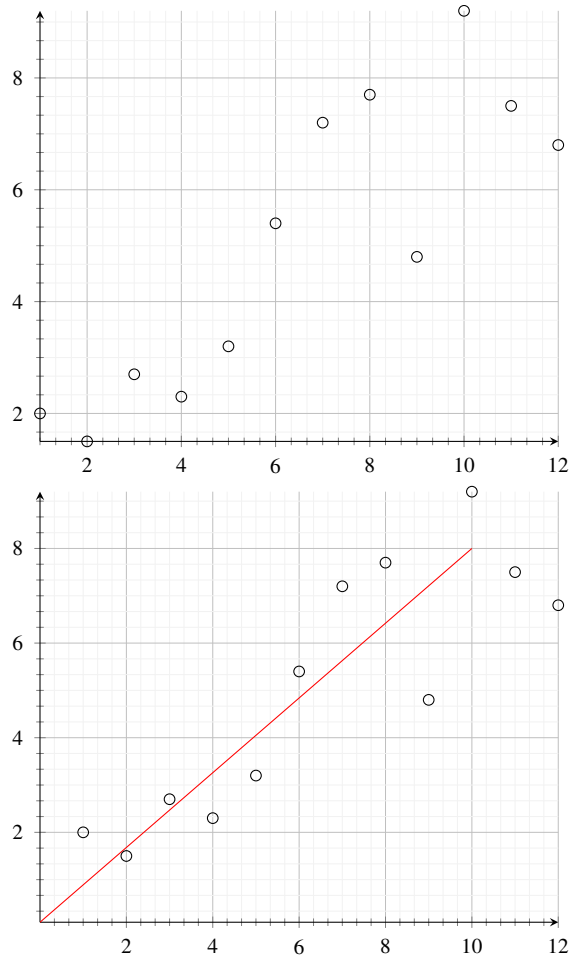


Figure 2.3 : Regression Solution.

(1) Sample dataset for a regression supervised learning problem.
(2) Sample dataset with prediction model in red.

$$= \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)} \tag{2.37}$$

One way it could visualize this is that the graph of J against θ would look like a convex, or "bowl" shaped function, and it is desired to minimize the value of J with respect to θ, i.e. local minima could be obtained by finding the minimum of the bowl as shown in Figure 2.4. The α in Equation (2.36) is called “the learning rate” that determines the convergence of the gradient descent to local minima. If the learning rate is too small, the convergence in gradient descent will be too slow. On the other hand, if the learning rate is too large, then gradient descent

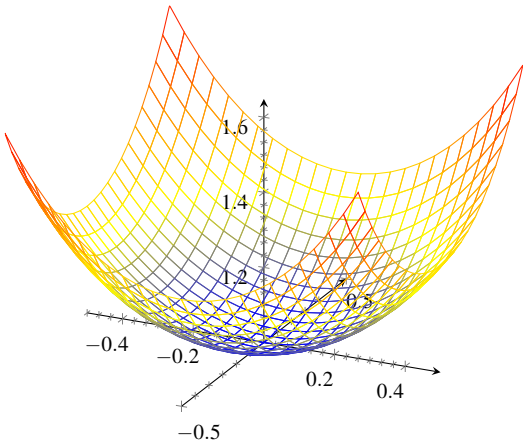


Figure 2.4 : Gradient Descent.

Plot of parameter values for cost function J(θ_0,θ_1) on resulting cost.

might overshoot the minimum, eventually failing to converge, or even diverging [22].

2.2.1.1 Gradient descent implementation

A test for convergence is easily done by plotting $J(\theta)$ as a function of number of iterations, and declare convergence if $J(\theta)$ decreases, less than 10^{-3} in one iteration [23].

One suggestion to choosing α is to vary α logarithmically in steps of 1/3. E.g. $\alpha = 0.01, 0.03, 0.1, 0.3, 1\ldots$, and testing the convergence.

Note that when doing the updating in Equation.2.36, all the θ parameters must be updated simultaneously. This makes sense because the second term in the update equation depends on all the rest of the θs, and once you update θ_i to θ_i', the update equation for $\theta_{j\neq i}$ will depend on (the new) θ_i' rather than (the old) θ_i [24].

There are two ways to implement this algorithm. **Batch gradient descent** (BGD) computes the update equation for the entire training set (i.e. all pairs of x and y), and then updates the parameters θ. **Stochastic gradient descent** (SGD), or incremental gradient descent, on the other hand, computes the update equation for one training example (i.e. one

[22] Note that as the gradient descent algorithm approaches a local minimum, the second term in the update equation will automatically be smaller and hence gradient descent will automatically take smaller steps, so there is no need to decrease α over time.

[23] It might be helpful too to see the plot, to see whether the learning rate is appropriate, etc.

[24] Doing the updating with vectorization automatically takes care of this.

pair of x and y) at a time and then updates θ. (This is useful if data is coming in one at a time.)

Algorithm 2: Gradient descent method:

Begin with some θ

repeat

1. $\Delta x := -\nabla f(x)$.
2. Change until find minimum $J(\theta)$ as Figure 2.5
3. *Update.* $\theta := \theta + t\Delta\theta$

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\vec{\theta}) = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$

until stopping criterion is satisfied.

Algorithm 2 provides some benefits to solve the problem while it undesirably emits certain drawbacks:

- Con: Need to choose an optimal α that will not overshoot or take too much time.
- Con: Need backpropagation if the weights are so arbitrarily chosen.
- Con: Computational complexities.
- Con: Vanishing.
- Con: May result in no converge.
- Con: Needs many iterations.
- Pro: Works well even if m is large (10^6).
- Pro: Works well even if θ is large.
- Pro: Variations of x such as its $\sqrt{\cdot}$, *polynomials* etc. can be easily inserted.

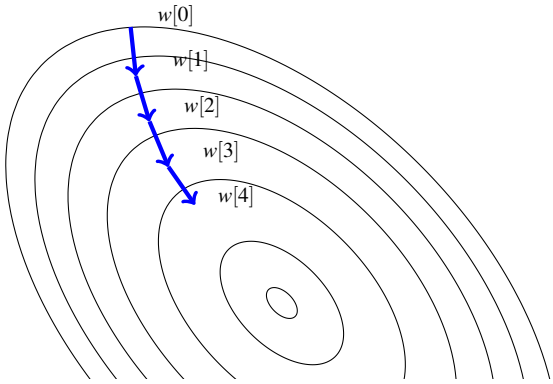


Figure 2.5 : Gradient Descent Towards Local Minima.

2.2.2 Feature scaling

The idea behind **feature scaling** is that the many variables x might have vastly different scales. E.g. x_1 could be on the order of 10^6 while x_2 could be on the order of 10^2 in their respective units. In this case, the algorithm might take much longer to reach convergence. One way to achieve feature scaling is to standardize every feature to be within $-3 \leq x_i \leq 3$. The process starts with by subtracting each feature from its mean and dividing by standard deviation likewise in below:

$$\text{Feature scaling:} \quad x_i' \rightarrow \frac{(x_i - \mu_i)}{\sigma_i} \tag{2.38}$$

where, μ[25] is the population mean[26], and σ is the population standard deviation[27]. In the literature of statistics, feature scaling is called as z-score transformation or standardization, z-value or normal scores. It scales the variables according to get a normal distribution.

Another way of **feature (re)scaling** is called as normalization as following:

$$\text{Feature rescaling:} \quad x_i' \rightarrow \frac{x_i - min(x)}{max(x) - min(x)} \tag{2.39}$$

The formula transforms x_i' in between [-1,+1] or [0,1] according to the type of the data.

Last way of **feature scaling** is mean normalization as below:

$$\text{Feature rescaling:} \quad x_i' \rightarrow \frac{x_i - min(x)}{max(x) - min(x)} \tag{2.40}$$

[25] Greek letter notation denotes for population while latin ones denote for sample.
[26] Sample mean is $\bar{X}$.
[27] Sample std. is s.

2.2.3 Regularization

Regularization is a method to prevent some features (variables) to affect output and new outputs fed by adding a penalty to overcome overfitting problem. There are two main ways to do this in machine learning literature, one is L^2 that is called Ridge, second one is L^1 called Lasso. Ridge uses restriction reducing the size of the coefficients but Ridge shrinks to zero continuously while LASSO does variable selection by setting selected variables to 0.

2.2.3.1 L^2 Regularization: Ridge regression

Ridge or Tikhonov (1943) regularization in machine learning is the first and commonly used method for penalization of parameter(s). Actually it is a well known technique in econometrics to prevent multicollinearity arising higher variance in estimation. In Equation 2.41, the second term initialized with λ is the penalty for θ.

$$\hat{\theta}_{ridge} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \theta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \theta_j^2 \tag{2.41}$$

First introduced into statistical literature by Hoerl and Kennard (1970), it can deal with non-invertible **(X'X)** matrices where it can be generalized in Equation (2.43) which represents a special case of an orthonormal design matrix. λ is a penalty term that is added to the diagonal elements of $(\boldsymbol{X'X})$ and $\boldsymbol{I}$ is an (nxn) matrix [28]

$$\hat{\theta}_{ridge} = \underset{\theta}{\operatorname{argmin}} (y - x^T \theta)^2 + \lambda \theta^2 \tag{2.42}$$

$$\hat{\theta}_{j,ridge} = \frac{\sum_{i=1}^{n} (x_{ij} - \bar{x_j})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_{ij} - \bar{x_j})^2} \frac{1}{1+\lambda} = \frac{\hat{\theta}_{j,ols}}{1+\lambda} \tag{2.43}$$

$$\hat{\theta}_{ridge} = (X'X + \lambda I)^{-1} X'y \tag{2.44}$$

[28] When λ=0, the solution provides OLS.

2.2.3.2 L^1 Regularization: Lasso regression

Lasso was first introduced by Tibshirani (1996) in order to increase the performance of a regression by using the most appropriate feature selection. In order to do that, lasso does the following:

$$\hat{\theta}_{lasso} = \underset{\theta}{\operatorname{argmin}}(\frac{1}{N}\sum_{i=1}^{n}(y_i - x_i^T\theta)^2) \;\; s.t. ||\theta_j||_1 \leq t \tag{2.45}$$

where $||\theta||_1 = \sum_{j=1}^{p}|\theta_j|$, and $p :=$ the number of features, t is the number of free parameters in regularization. Additionally,$||\theta||_1$ is called as the unit-norm of the vector θ and $t \geq 0$.

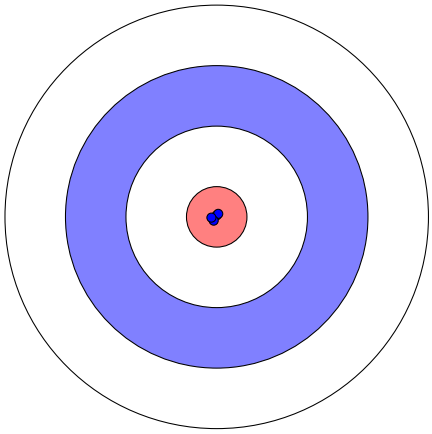
$$\hat{\theta}_{lasso} = \underset{\theta}{\operatorname{argmin}}(\frac{1}{N})\sum(y_i - x_i^T\theta)^2 + \lambda|\theta| \tag{2.46}$$

Equation (2.45) and (2.46) lead to a minimization similar to Equation (2.41) from Ridge, except the penalization to θ is linear instead of quadratic which is beneficial for penalization of some parameters by providing the properties of sparsity meaning less parameterization.

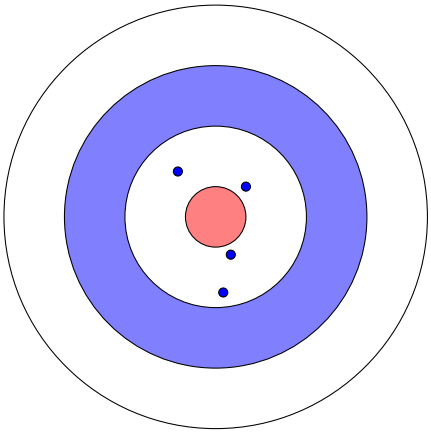
2.3 Bias-Variance Tradeoff

The tradeoff in between bias and variance is the well-known dilemma in supervised machine learning. It implicitly states that a machine learning algorithm whose data has its own labels, ought to optimally balance high accuracy by decreasing its scattering and low scattering by increasing accuracy. In general, while linear and parametric methods have a high bias but a low variance, non-linear and non-parametric algorithms often have a low bias but a high variance. Graphical definition of this tradeoff is shown in Figure 2.6. A well-known example is the Fukushima case (Stacey, 2015) that resulted in a nuclear accident triggered by a tsunami in Okuma [29] since engineers were using historical data dating back to 1600's predicted an overfitting linear regression that can be classified as a spurious one like in a time-series regression (Gujarati, 2009).

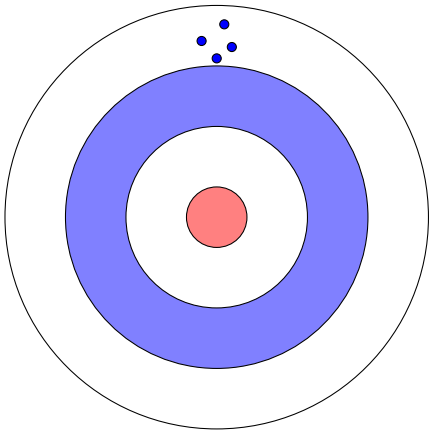
[29] http://www.greenpeace.org/international/en/campaigns/nuclear/safety/accidents/Fukushima-nuclear-disaster/



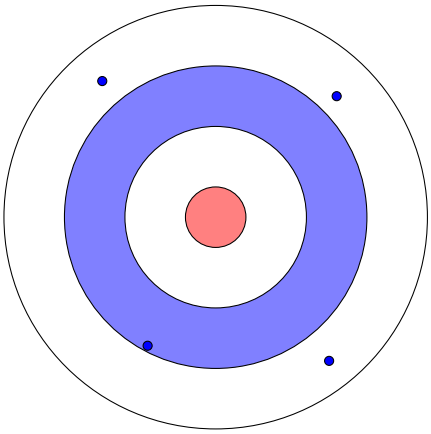
((a)) Low Variance & Low Bias



((b)) High Variance & Low Bias



((c)) Low Variance & High Bias



((d)) High Variance & High Bias

Figure 2.6 : Bias-Variance Tradeoff Representation.

The MSE: In statistics, the MSE (mean squared errors) is generally used to calculate the performance of an estimator. Since precision, bias and accuracy are the most important concepts in statistics during a statistical estimation, the MSE is frequently used in this process. It is only the average of squares of the errors. In here, error can be defined as the difference between the estimator and the estimated. It sometimes called as loss. Errors or loss are resulted from the randomness, ommited variable bias, irrelevant variable, misspecification etc. The second moment of the errors about to origin constitutes two main components, the first one is the bias, the second is the variance. For an unbiased estimator, the MSE equals the variance of the estimator since the bias is zero.

Bias-based errors: The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which it is being tried to predict. Of course the analyst only has one model so talking about expected or average prediction values might seem a little strange. However, imagine that the analyst -data scientist- could repeat the whole model building process more than once: each time he gathers new data and runs a new analysis creating a new model. Due to the randomness in the underlying data sets, the resulting models will have a range of predictions. Bias measures how far off in general these models' predictions are from the correct value.

Variance-based errors: The error due to variance is taken as the variability of a model prediction for a given data point. Again, imagine you can repeat the entire model building process multiple times. The variance is how much the predictions for a given point vary between different realizations of the model.

Mathematically, Hastie et al. (2009) show the tradeoff that the covariates are X to calculate label or outputs denotes as Y as, $Y = f(X) + \varepsilon$ in where $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon)$ as Greene (2003) states it as an assumption in linear

regression. So, the predicted $\hat{Y}$ maintains the information to calculate the followings:

$$MSE(x) = E\left[(\theta - E(\hat{\theta})^2\right]. \tag{2.47}$$

Errors calculated, however, can be decomposed into bias and variance components such as:

$$MSE(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2, \tag{2.48}$$

literally as followings:

$$MSE(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}. \tag{2.49}$$

Irreducible error, in econometrics, may be named as "structural error" emerging through ommited variable(s), irrelevant feature use or the errors can be completed by the God [30] but not a human.

2.4 Neural Networks

Neural networks are one of the most popular methods used in research and applications according to Jain et al. (1996). It is used in many applications such as optical character recognition (the automation of the postal service) and credit card authentication. It allows an easy way to compute non-linear classification problems, because essentially at the heart of the algorithm, one lets it compute its own parameters and weights [31].

Neural networks were inspired by biology, notably by brain plasticity, which is the ability of different parts of the brain to learn new functions, such as a classic neural re-wiring experiment where the auditory/somatosensory cortex can learn to "see" (receive and process visual input). The brain's amazing ability to adapt different regions of

[30] Prof. Dr. M.Ü.Ş. quotes that "there must be an equation which has $R^2 = 1$ and is known by God that is unknown.".

[31] The problem is that most of the time, it's a non-convex optimization problem (so gradient descent is not guaranteed to work). Also, it is computationally expensive to train a neural network. SVMs are more reliable. That being said, SVMs have their own pros and cons too.

the brain to learn the functions of another part suggests a “one learning algorithm" hypothesis, rather than sense-specific algorithms. In this sense, a neuron can be modeled as a computing unit, with dendrites as “input" wires and the axon as an “output" wire.

In the neural network model, a neuron is a single unit with an activation function (i.e. sigmoid/logistic). A network consists of different layers of neurons. The first layer, which directly receives the input, is called the **input layer**. The first layer then feeds into the second layer, etc, until the final layer, which eventually computes the estimation, is called the **output layer**. The rest of the layers in between are called the **hidden layers**.

Let it define $a_i^{(j)}$ as the *activation* of unit i in layer j, i.e. $a_i^{(j)} = 1$ if unit i in layer j fires. (Similar to logistic and linear regression, it can be added a constant term, a_0, also called a **bias** term). Let $\theta^{(j)}$ be the matrix of weights that control the mapping from layer j to layer $j+1$. If the network has s_j units in layer j and s_{j+1} units in layer $j+1$, then $\theta^{(j)}$ will be of dimension $s_{j+1} \times (s_j + 1)$ because of the bias term.

Passing information through the neural network from input layer to output layer, called **forward propagation**, is achieved via:

$$a_k^{(2)} = g\left(\theta_{k,0}^{(1)} x_0 + \theta_{k,1}^{(1)} x_1 + \ldots\right) \tag{2.50}$$

$$= g\left(\theta_{k,0}^{(1)} a_0^{(1)} + \theta_{k,1}^{(1)} a_1^{(1)} + \ldots\right) = g\left(\theta^{(1)} a^{(1)}\right) \tag{2.51}$$

$$\tag{2.52}$$

or more generally,

$$= g\left(z^{(j+1)}\right) \quad ; \quad z^{(j+1)} = \left(\theta^{(j)} a^{(j)}\right) \tag{2.53}$$

Simple examples of neural networks used to calculate boolean functions:

- AND $h_\theta(x) = g(-30 + 20x_1 + 20x_2)$
- OR $h_\theta(x) = g(-10 + 20x_1 + 20x_2)$
- NOT. $h_\theta(x) = g(10 - 20x_1)$ (put a large negative weight to negate)

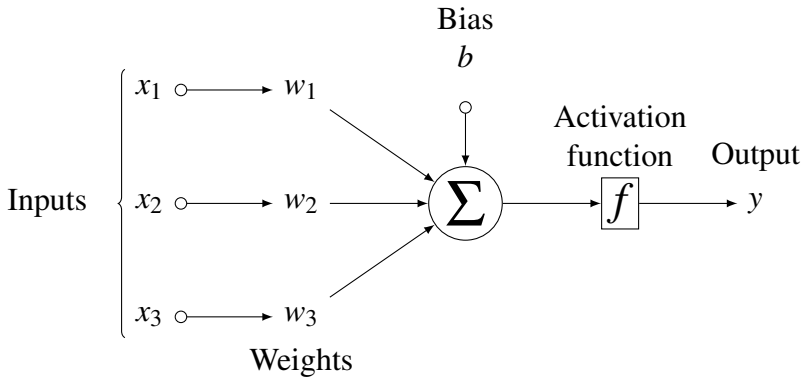


Figure 2.7 : Single Layer Artificial Neural Network.

- (NOT x_1) AND (NOT x_2). $h_\theta(x) = g(10 - 20x_1 - 20x_2)$
- x_1 XNOR x_2: $a_1 = x_1$ AND x_2, $a_2 =$ (NOT x_1) AND (NOT x_2), $h_\theta(x) = a_1$ OR a_2

Note that, similar to logistic regression, neural networks can similarly be adapted to multi class classification. One way is similar to the one-vs-all algorithm: but now it can be specified that output to be a vector. For example, you can write your algorithm such that the output $h_\theta(x) = [1000\ldots]$ if first class, $[0100\ldots]$ if second class etc., and that training set would just be $(x^{(i)}, y^{(i)})$ where $y^{(i)}$ is $[1000\ldots]$ or $[0100\ldots]$ etc. depending on class.

2.4.1 Single layer perceptron

If an ANN structure has one layer that contains neurons fed by weighted inputs, it is called a **single layer artificial neurons**. A simple architecture of single layered neurons is shown in Figure 2.7. As stated in the comparison of machine learning and econometrics below, any econometric problem can be thought of as a one layered machine learning problem without considering its iterative process.

2.4.2 Multilayer perceptron

Multilayared perceptron model is similar to single layer neural networks except for its hidden layer. If the number of hidden layers is at least 1, any shallow or wide machine learning architecture turns out to be deep learning (DL) architecture as drawn in Figure 2.8.

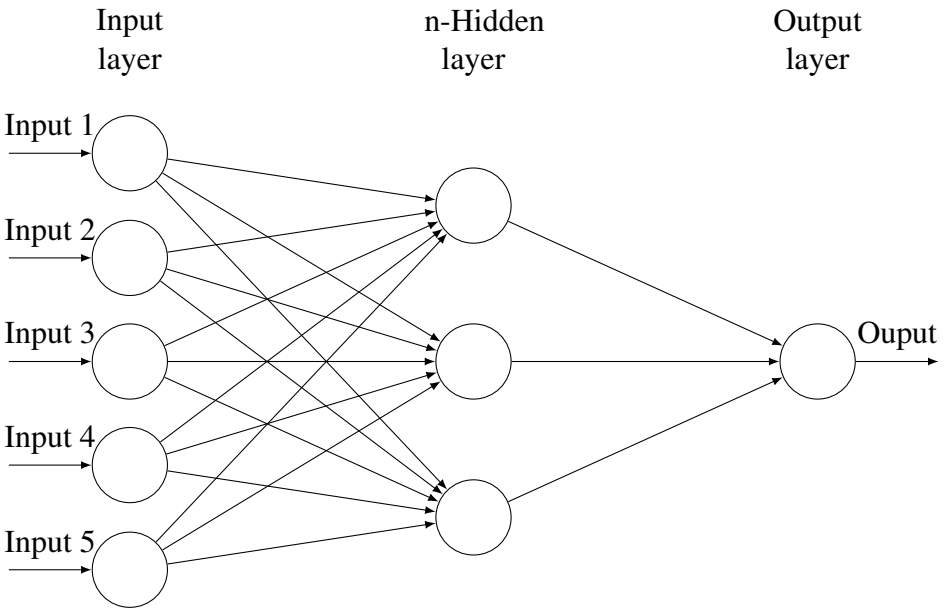


Figure 2.8 : Multi Layer (Deep) Artificial Neural Network.

In DL, there are no feedback connections in which outputs of the model are fed back into itself. The model is associated with a directed acyclic graph describing how the functions are composed together. These networks require initializing all weights to small random values. To apply gradient-based learning, a cost function and output representation must be chosen. Most modern NN are trained using maximum likelihood. This means that the cost function is simply the negative log-likelihood:

$$J(\boldsymbol{\theta}) = -\mathbb{E}_{x,y\sim\hat{y}_{data}} \log p_{model}(\boldsymbol{y}|\boldsymbol{x}) \tag{2.54}$$

Most hidden units can be described as accepting a vector of inputs ($\boldsymbol{x}$), computing an affine transformation ($\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$) and then applying an element-wise nonlinear function ($g(\boldsymbol{z})$).

Most NN are organized into groups of units called layers. Most NN architectures arrange these layers in a chain structure, with each layer being a function of the layer that preceded it. In these architectures the main considerations are choosing the depth of the network and the width of each layer. Deeper networks generalize better (most of the time).

Universal Approximation Theorem: a feedforward NN with a linear output layer and at least one hidden layer with any "squashing" activation function can approximate any Borel measurable function from one finite-dimensional space to another with any desired non-zero amount of error, provided that the network is given enough hidden units. The derivatives of the feedforward network can also approximate the derivatives of the function arbitrarily well.

Many architectures build a main layers chain but then add extra architectural features to it, such as skip connections going from layer i to layer $(i+2)$ or higher. These skip connections make it easier for the gradient to flow from output layers to layers nearer the input.

During training, forward propagation can continue onward until it produces a scalar cost $(J(\boldsymbol{\theta}))$. The back-propagation algorithm allows the information from the cost to then flow backward through the network in order to compute the gradient.

Back-propagation is the method for computing the gradient, while another algorithm, such as stochastic gradient descent, is used to perform learning using this gradient. In learning algorithms, the gradient that is most often required is the gradient of the cost function w.r.t. the parameters: $(\nabla_{\theta} J(\boldsymbol{\theta}))$.

The chain rule of calculus is used to compute the derivatives of functions formed by composing other functions whose derivatives are known. Backprop computes the chain rule, with a specific order of operations that is highly efficient.

Suppose that $(\boldsymbol{x} \in \mathbb{R}^m, \boldsymbol{y} \in \mathbb{R}^n, g)$ maps from $(\mathbb{R}^m)$ to $(\mathbb{R}^n)$, and f maps from $(\mathbb{R}^n)$ to $(\mathbb{R})$. If $(\boldsymbol{y} = g(\boldsymbol{x}))$ and $(z = f(\boldsymbol{y}))$, then $[\frac{\partial z}{\partial x_i} = \sum_i \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i}]$ In vector notation, this may be equivalently written as $[\nabla_{\boldsymbol{x}} z = \left(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right)^{\top} \nabla_{\boldsymbol{y}} z,]$ where $(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})$ is the $(n \times m)$ Jacobian matrix of g.

So, the gradient of a variable $(\boldsymbol{x})$ can be obtained by multiplying a Jacobian matrix $(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})$ by a gradient $(\nabla_{\boldsymbol{y}} z)$. The backprop algorithm

consists of performing such a Jacobian-gradient product for each operation in the graph.

2.4.3 Neural networks: Learning

Let L be the total number of layers in the network, and each layer l has s_l number of units (not counting bias units).

The cost function is given by (remember to exclude the bias terms for the regularization terms):

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right)_k + \left(1-y_k^{(i)}\right)\log\left(1-\left(h_\theta\left(x^{(i)}\right)\right)_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\theta_{ji}^{l}\right)^2 \tag{2.55}$$

For binary classification, $S_L = 1$.

For multi-class classification with K classes, $S_L = K$. $h_\theta(x) \in \mathbb{R}^K$, and $(h_\theta(x))_i \equiv i$-th output.

2.4.3.1 Backpropagation algorithm

Backpropagation algorithm is as follows:

- Need: $J(\theta)$ and $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta)$
- Recall forward propagation steps, given (x,y) in a N-Layer network:
- $a^{(1)} = x$
- $z^{(2)} = \theta^{(1)} a^{(1)}$
- $a^{(2)} = g\left(z^{(2)}\right)$ (add the bias term, $a_0^{(2)}$)
- $z^{(3)} = \theta^{(2)} a^{(2)}$
- $a^{(3)} = g\left(z^{(3)}\right)$ (add the bias term, $a_0^{(3)}$)
- $z^{(4)} = \theta^{(3)} a^{(3)}$
- $a^{(4)} = h_\theta(x) = g\left(z^{(4)}\right)$
- Intuition: $\delta_j^{(l)} \equiv$ "error" of node j in layer l

· For each output unit, for layer $L = 4$ as above

· $\delta_j^{(4)} = a_j^{(4)} - y_j = (h_\theta(x))_j - y_j$ for each element j. Or (vectorized):

· $\delta^{(4)} = a^{(4)} - y = h_\theta(x) - y$

· $\delta^{(3)} = \left(\theta^{(3)}\right)^T \delta^{(4)} .* g'\left(z^{(3)}\right)$ where $g'\left(z^{(3)}\right) = a^{(3)} .* (1 - a^{(3)})$

· $\delta^{(2)} = \left(\theta^{(2)}\right)^T \delta^{(3)} .* g'\left(z^{(2)}\right)$

· $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = a_j^{(l)} \delta_i^{(l+1)}$

· Given a training set (x, y), set accumulator $\Delta_{ij}^{(l)} = 0$

· For $i = 1$ to m; set $a^{(1)} = x^{(i)}$; perform forward propagation to compute $a^{(l)}$ for $l = 2 \ldots L$

· Using $y^{(i)}$, compute $\delta^{(L)} = a^{(L)} - y^{(i)}$, backpropagate to compute $\delta^{(l)}$s.

· $\Delta_{ij}^{(l)} += a_j^{(l)} \delta_i^{(l+1)}$ or $\Delta^{(l)} += \delta^{(l+1)} (a^{(l)})^T$ (Note, skip δ corresponding to bias unit)

· Outside for loop, calculate $D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)} + \frac{\lambda}{m} \theta_{ij}^{(l)}$ if $j \neq 0$, and $D_{ij}^{(l)} = \frac{1}{m} \Delta_{ij}^{(l)}$ if $j = 0$

· $\frac{\partial}{\partial \theta_{ij}^{(l)}} J(\theta) = D_{ij}^{(l)}$

· **Gradient Checking**

· When testing code, compute gradient numerically and compare with that given by back propagation.

· But when actually using code (e.g. learning), turn off gradient checking. It's VERY slow.

· $\frac{\partial}{\partial \theta_i} J(\theta) \approx \frac{1}{2\varepsilon} \left[J(\theta_1, \ldots \theta_i + \varepsilon, \ldots \theta_n) - J(\theta_1, \ldots \theta_i - \varepsilon, \ldots \theta_n) \right]$

2.4.3.2 Random initialization

If it is initialized the starting parameters to be all zero, then every unit in the neural network will be identical. Thus, it would be needed random initialization in order to break the symmetry. Initialize each $\Theta_{ij}^{(l)}$ to a random variable in $[-\varepsilon, \varepsilon]$[32] in Python.

2.4.3.3 Architecture design

First, it would be needed to choose a particular network architecture (how many layers, how many units...). Usually, the dimension of the input (data) and output (e.g. K classes) are decided by problem. A reasonable default is having 1 hidden layer. If one chooses to implement more than 1 hidden layer, it's a good rule of thumb to have the same number of hidden units in every layer. Usually, the more units you have in each layer, the better, but it gets more expensive to compute.

Start for the architecture in Figure2.9 to run should be as follows:

- Random Initialization of weights,
- Forward Prop,
- Compute Cost,
- Back prop (Check with gradient checking, then turn off),
- Optimize (e.g. using gradient descent) cost function.

2.4.3.4 Feed forwards

A FNN is formed by one input layer, one (shallow network) or more (deep network, hence the name deep learning) hidden layers and one output layer. Each layer of the network (except the output one) is connected to a following layer. This connectivity is central to the FNN structure and has two main features in its simplest form: a weight averaging feature and an activation feature. It will be reviewed these features extensively.

Notations are as follows:

In the following, it will be called,

- N the number of layers (not counting the input) in the Neural Network.

[32] e.g. to initialize a 10x11 theta matrix: Theta1 = rand(10,11)*(2*init_epsilon) - init_epsilon

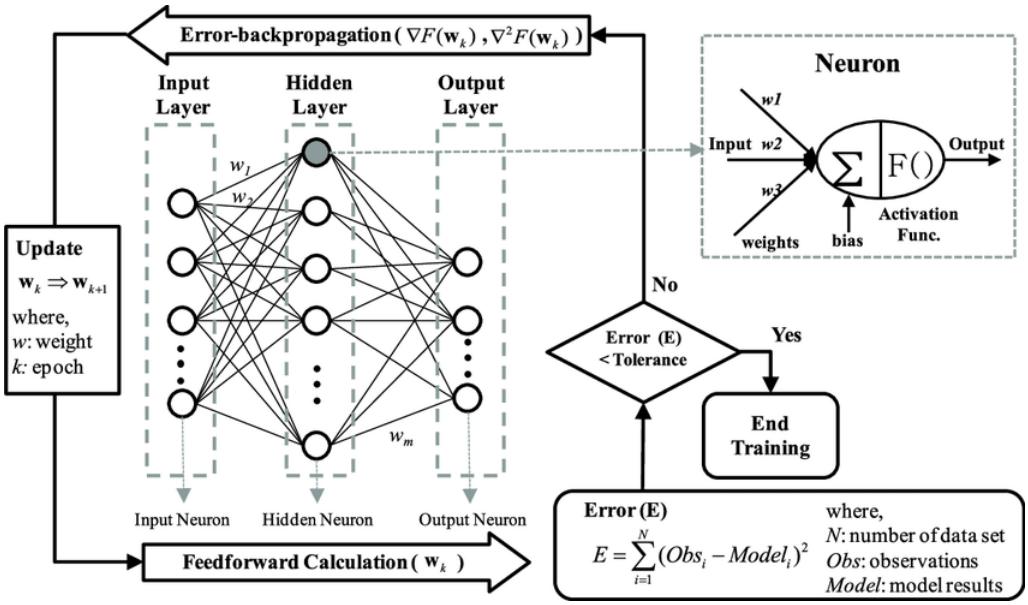


Figure 2.9 : Architecture Design of a Neural Network.

- T_{train} the number of training examples in the training set.
- T_{mb} the number of training examples in a mini-batch (see Glossary).
- $t \in [\![0, T_{\text{mb}} - 1]\!]$ the mini-batch training instance index.
- $\nu \in [\![0, N]\!]$ the number of layers in the FNN.
- F_ν the number of neurons in the ν'th layer.
- $X_f^{(t)} = h_f^{(0)(t)}$ where $f \in [\![0, F_0 - 1]\!]$ the input variables.
- $y_f^{(t)}$ where $f \in [0, F_N - 1]$ the output variables (to be predicted).
- $\hat{y}_f^{(t)}$ where $f \in [0, F_N - 1]$ the output of the network.
- $\Theta_f^{(\nu) f'}$ for $f \in [0, F_\nu - 1]$, $f' \in [0, F_{\nu+1} - 1]$ and $\nu \in [0, N-1]$ the weights matrices
- A bias term can be included. In practice, it would be seen when talking about the batch-normalization procedure that we can omit it.

2.4.3.5 Weight averaging

One of the two main components of a FNN is a weight averaging procedure, which amounts to averaging the previous layer with some weight matrix to obtain the next layer. This is illustrated in Figure 2.11.

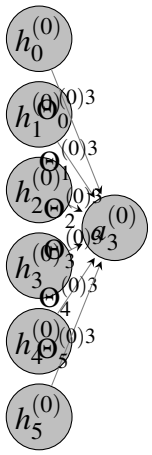


Figure 2.11 : Weight averaging procedure.

[a]For simplicity of notations, the index referencing the training set has not been indicated. Shallow architectures use only one hidden layer. Deep learning amounts to take several hidden layers, usually containing the same number of hidden neurons. This number should be on the ballpark of the average of the number of input and output variables.

Formally, the weight averaging procedure reads:

$$a_f^{(t)(\nu)} = \sum_{f'=0}^{F_\nu - 1 + \varepsilon} \Theta_{f'}^{(\nu)f} h_{f'}^{(t)(\nu)} \ , \tag{2.56}$$

where $\nu \in [\![0, N-1]\!]$, $t \in [\![0, T_{\text{mb}} - 1]\!]$ and $f \in [\![0, F_{\nu+1} - 1]\!]$. The ε is here to include or exclude a bias term. In practice, as batch-normalization is being used, one can safely omit it ($\varepsilon = 0$ in all the following).

2.4.4 Convolutional neural networks

Convolutional Neural Networks (abrrevated as CNN in ML literature) implement the idea of signal convolution to neural networks. It actually separates image data into the fragments in many ways on neurons, filters and then applies locally on subregions of the images rather than of the whole image (Lecun et al., 1998). The filters derived from these pre-processes are convolved on input images, then, it is used for creating new features, feature maps, intermediate maps.

The original input image is considered as a single feature map, $f(x,y)$. Generally the input to a convolutional layer is a set of n_f feature maps, $f_i(x,y)$, on which the n_j filters of size $(p \times q)$ $g_k(x,y)$ (also usually represented as a 4d-tensor W of weights) are convolved, thus creating n_g new feature maps

$h_j(x,y)$:

$$h_j(x,y) = \sum_{u=0}^{p} \sum_{v=0}^{q} \sum_{i=0}^{n_f} f_i(x+u, y+v) g_j(u,v) \tag{2.57}$$

A bias is then added on each individual filter map, which is then bounded by an activation function (here, the hyperbolic tangent, *tanh*is used). One gets that the output of a layer in terms of feature maps is

$$H_i(x,y) = \tanh(h_i(x,y) + b_i) = \tanh(W * x + b). \tag{2.58}$$

Finally, maxpooling is applied to the filter maps, where each $(s \times t)$ block of the maps are reduced to their maximum value. For $s = t = 2$, it is essentially scaling the image down by a factor of two, but keeping the max instead of taking, say, the average value. This gives the model some level of translation invariance, on top of the ability to detect highly local features.

After having created a feature map representation of the input, it is flattened to the vector form, and feed it to a two layer feedforward neural network, with a *tanh* hidden layer and a softmax output layer. The softmax function is as follows:

$$s(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.59}$$

2.4.5 Sequence modeling

In general, sequence modeling differs from other types of machine learning models since the output estimated in a neural network feeds other perceptrons sequentially according to the architecture of the system. As a result, sequence modeling is the most appropriate model for time series problems in econometrics.

2.4.5.1 Recurrent networks

Recurrent neural networks (RNN) are networks in which connections between units form a directed loop. It is allowed to exhibit dynamic temporal behavior with RNN. Unlike feedforward neural networks, RNNs can use their input memory to process inputs. This attribute makes RNNs a useful method of handwriting recognition and speech recognition.

People don't learn new meanings for a word they've just learned. They relate to the word they have just learned by starting from similar pre-existing words.

However, traditional artificial neural networks do not have this meaning attribute found in humans, and this is their greatest deficiency. For example, if you want to classify activities by looking at all the frames in the video, traditional neural networks will not be able to classify because they cannot make sense between people like frames.

Recurrent Neural Network (Recurrent Neural Network), by creating a loop, will provide the use of historical information and thus can make classification by making meaning between frames.

In traditional artificial neural networks, the results from the cells do not come back to them as input. In RNN, the result from the cell comes back to it as input. If RNN is opened, an architecture as shown in the following figure appears. In the time frame, the same cell repeats itself more than once. Thus, meaning can be established between frames.

Forward pass in a RNN-LSTM In Figure 2.12, the RNN architecture is presented in a schematic way;

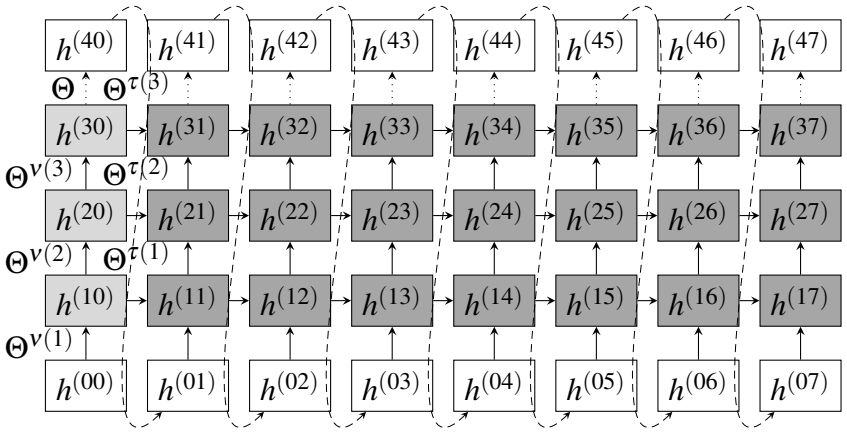


Figure 2.12 : RNN Architecture within Space and Time.

In the above example, the time dimension is of size 8 while the "spatial" one is of size 4.

The real novelty of this type of neural network is the fact that it is being used to predict a time series encoded in the very architecture of the network.

RNN have first been introduced mostly to predict the next words in a sentence (classification task), hence the notion of ordering in time of the prediction. But this kind of network architecture can also be applied to regression problems. Among other things one can think of are stock price evolution or temperature forecasting. In contrast to the preceding neural networks that were introduced, where it is defined (denoting ν as in previous chapters the layer index in the spatial direction)

$$\begin{aligned} a_f^{(t)(\nu)} &= \text{Weight Averaging } \left(h_f^{(t)(\nu)}\right) , \\ h_f^{(t)(\nu+1)} &= \text{Activation function } \left(a_f^{(t)(\nu)}\right) , \end{aligned} \tag{2.60}$$

she/he now has the hidden layers that are indexed by both a "spatial" and a "temporal" index (with T being the network dimension in this new direction), and the general philosophy of the RNN is (now the a is usually characterized by a c for cell state, this denotation, trivial for the basic RNN architecture will make more sense when it is covered in LSTM networks) to run a dynamic averaging as follows:

$$\begin{aligned} c_f^{(t)(\nu\tau)} &= \text{Weight Averaging } \left(h_f^{(t)(\nu\tau-1)}, h_f^{(t)(\nu-1\tau)}\right) , \\ h_f^{(t)(\nu\tau)} &= \text{Activation function } \left(c_f^{(t)(\nu\tau)}\right) , \end{aligned} \tag{2.61}$$

in where τ is the horizontal dimension.

Backward pass in a RNN-LSTM The backward pass in a RNN-LSTM has to respect a certain time order, as illustrated in the following Figure 2.13:

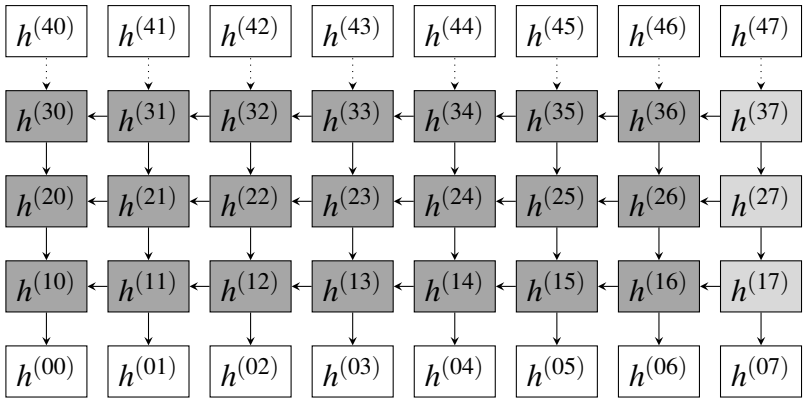


Figure 2.13 : Backward Pass in RNN Architecture.

With this in mind, the implementation of a RNN and its advanced cousin, the Long Short Term Memory (LSTM)-RNN will be discussed in detail.

Input layer In a RNN-LSTM, the input layer is recursively defined as

$$h_f^{(t)(0\tau+1)} = \left(\tilde{h}_f^{(t)(0\tau)}, h_f^{(t)(N-1\tau)}\right) . \tag{2.62}$$

where $\tilde{h}_f^{(t)(0\tau)}$ is $h_f^{(t)(0\tau)}$ with the first time column removed.

Output layer The output layer of a RNN-LSTM reads

$$h_f^{(t)(N\tau)} = o\left(\sum_{f'=0}^{F_{N-1}-1} \Theta_{f'}^{f} h_f^{(t)(N-1\tau)}\right) , \tag{2.63}$$

where the output function o is as for FNN's and Convolutional Neural Network's (CNN), which is either the identity (regression task) or the cross-entropy function (classification task).

Loss function The loss function for a regression task reads

$$J(\Theta) = \frac{1}{2T_{\text{mb}}} \sum_{t=0}^{T_{\text{mb}}-1} \sum_{\tau=0}^{T-1} \sum_{f=0}^{F_N-1} \left(h_f^{(t)(N\tau)} - y_f^{(t)(\tau)}\right)^2 . \tag{2.64}$$

and for a classification task

$$J(\Theta) = -\frac{1}{T_{\text{mb}}} \sum_{t=0}^{T_{\text{mb}}-1} \sum_{\tau=0}^{T-1} \sum_{c=0}^{C-1} \delta^{c}_{y_c^{(t)(\tau)}} \ln\left(h_f^{(t)(N\tau)}\right) . \tag{2.65}$$

Forward pass in a RNN In a RNN, the update rules read for the first time slice (spatial layer at the extreme left of Figure 2.12)

$$h_f^{(t)(\nu\tau)} = \tanh\left(\sum_{f'=0}^{F_{\nu-1}-1} \Theta_{f'}^{\nu(\nu)f} h_{f'}^{(t)(\nu-1\tau)}\right) , \tag{2.66}$$

and for the other ones

$$h_f^{(t)(\nu\tau)} = \tanh\left(\sum_{f'=0}^{F_{\nu-1}-1} \Theta_{f'}^{\nu(\nu)f} h_{f'}^{(t)(\nu-1\tau)} + \sum_{f'=0}^{F_{\nu}-1} \Theta_{f'}^{\tau(\nu)f} h_{f'}^{(t)(\nu\tau-1)}\right) . \tag{2.67}$$

Backpropagation in a RNN The backpropagation philosophy will remain unchanged : find the error rate updates, from which one can deduce the weight updates. But as for the hidden layers, the δ now has both a spatial and a temporal component. It will thus be hard to compute,

$$\delta_f^{(t)(\nu\tau)} = \frac{\delta}{\delta h_f^{(t)(\nu+1\tau)}} J(\Theta) \,, \tag{2.68}$$

to deduce

$$\Delta_{f'}^{\Theta \text{index} f} = \frac{\delta}{\delta \Delta_{f'}^{\Theta \text{index} f}} J(\Theta) \,, \tag{2.69}$$

where the index can either be nothing (weights of the ouput layers), $\nu(\nu)$ (weights between two spatially connected layers) or $\tau(\nu)$ (weights between two temporally connected layers). First, it is easy to compute for the MSE loss function,

$$\delta_f^{(t)(N-1\tau)} = \frac{1}{T_{\text{mb}}} \left(h_f^{(t)(N\tau)} - y_f^{(t)(\tau)} \right) \,, \tag{2.70}$$

and for the cross entropy loss function,

$$\delta_f^{(t)(N-1)} = \frac{1}{T_{\text{mb}}} \left(h_f^{(t)(N\tau)} - \delta^f_{y^{(t)(\tau)}} \right) \,. \tag{2.71}$$

Calling,

$$\mathscr{T}_f^{(t)(\nu\tau)} = 1 - \left(h_f^{(t)(\nu\tau)} \right)^2 \,, \tag{2.72}$$

and

$$\mathscr{H}_{ff'}^{(t')(\nu\tau)_a} = \mathscr{T}_{f'}^{(t')(\nu+1\tau)} \Theta_f^{a(\nu+1)f'} \,, \tag{2.73}$$

it is shown that (if $\tau+1$ exists, otherwise the second term is absent)

$$\delta_f^{(t)(\nu-1\tau)} = \sum_{t'=0}^{T_{\text{mb}}} J_f^{(tt')(\nu\tau)} \sum_{\varepsilon=0}^{1} \sum_{f'=0}^{F_{\nu+1-\varepsilon}-1} \mathscr{H}_{ff'}^{(t')(\nu-\varepsilon\tau+\varepsilon)_{b\varepsilon}} \delta_{f'}^{(t')(\nu-\varepsilon\tau+\varepsilon)} \,. \tag{2.74}$$

where $b_0 = \nu$ and $b_1 = \tau$.

Weight and coefficient updates in a RNN To complete the backpropagation algorithm, it is needed

$$\Delta_{f'}^{\nu(\nu)f}\,, \qquad \Delta_{f'}^{\tau(\nu)f}\,, \qquad \Delta_{f'}^{f}\,, \qquad \Delta_{f}^{\beta(\nu\tau)}\,, \qquad \Delta_{f}^{\gamma(\nu\tau)}\,. \tag{2.75}$$

it is shown below that,

$$\Delta_{f'}^{\nu(\nu-)f} = \sum_{\tau=0}^{T-1}\sum_{t=0}^{T_{\mathrm{mb}}-1} \mathscr{T}_{f}^{(t)(\nu\tau)}\delta_{f}^{(t)(\nu-1\tau)}h_{f'}^{(t)(\nu-1\tau)}\,, \tag{2.76}$$

$$\Delta_{f'}^{\tau(\nu)f} = \sum_{\tau=1}^{T-1}\sum_{t=0}^{T_{\mathrm{mb}}-1} \mathscr{T}_{f}^{(t)(\nu\tau)}\delta_{f}^{(t)(\nu-1\tau)}h_{f'}^{(t)(\nu\tau-1)}\,, \tag{2.77}$$

$$\Delta_{f'}^{f} = \sum_{\tau=0}^{T-1}\sum_{t=0}^{T_{\mathrm{mb}}-1} h_{f'}^{(t)(N-1\tau)}\delta_{f}^{(t)(N-1\tau)}\,, \tag{2.78}$$

$$\Delta_{f}^{\beta(\nu\tau)} = \sum_{t=0}^{T_{\mathrm{mb}}-1}\sum_{\varepsilon=0}^{1}\sum_{f'=0}^{F_{\nu+1-\varepsilon}-1} \mathscr{H}_{ff'}^{(t')(\nu-\varepsilon\tau+\varepsilon)_{b\varepsilon}}\delta_{f'}^{(t')(\nu-\varepsilon\tau+\varepsilon)}\,, \tag{2.79}$$

$$\Delta_{f}^{\gamma(\nu\tau)} = \sum_{t=0}^{T_{\mathrm{mb}}-1}\tilde{h}_{f}^{(t)(\nu\tau)}\sum_{\varepsilon=0}^{1}\sum_{f'=0}^{F_{\nu+1-\varepsilon}-1} \mathscr{H}_{ff'}^{(t')(\nu-\varepsilon\tau+\varepsilon)_{b\varepsilon}}\delta_{f'}^{(t')(\nu-\varepsilon\tau+\varepsilon)}\,. \tag{2.80}$$

RNN and LSTM/GRU All RNNs have feedback loops in the recurrent layer. This lets them maintain information in 'memory' over time. But, it can be difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies. This is because the gradient of the loss function decays exponentially with time (called the vanishing gradient problem). LSTM networks are a type of RNN that uses special units in addition to standard units. LSTM units include a 'memory cell' that can maintain information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten. This architecture lets them learn longer-term dependencies. GRUs are similar to LSTMs, but use a simplified structure. They also use a set of gates to control the flow of information, but they don't use separate memory cells, and they use fewer gates.

In a Long Short Term Memory Neural Network, the state of a given unit is not directly determined by its left and bottom neighbours. Instead, a cell state is updated for each hidden unit, and the output of this unit is a probe of the cell state. This formulation might seem puzzling at first, but it is philosophically similar to the ResNet approach. Instead of trying to fit an

input with a complicated function, it is tried to fit tiny variation of the input, hence allowing the gradient to flow in a smoother manner in the network.

2.5 Activation Functions

Artificial neural networks are indispensable to the activation functions. There are three main categories in ANN literature: Monotonic activation functions which are the most common functions used in ANN is first one and is explained Chapter 2.6 in detail. Periodic activation functions are the second group of activation functions, they are less known functions which are hard to compute to meet special needs of the architecture of ANN, but may perform well especially in Adaptive Artificial Networks. They are described in Chapter 2.7. The last part of activation function is actually neither an activator nor a function, nevertheless, it is an essential part of activation function, it is called bias term that is mentioned in Chapter 2.8.

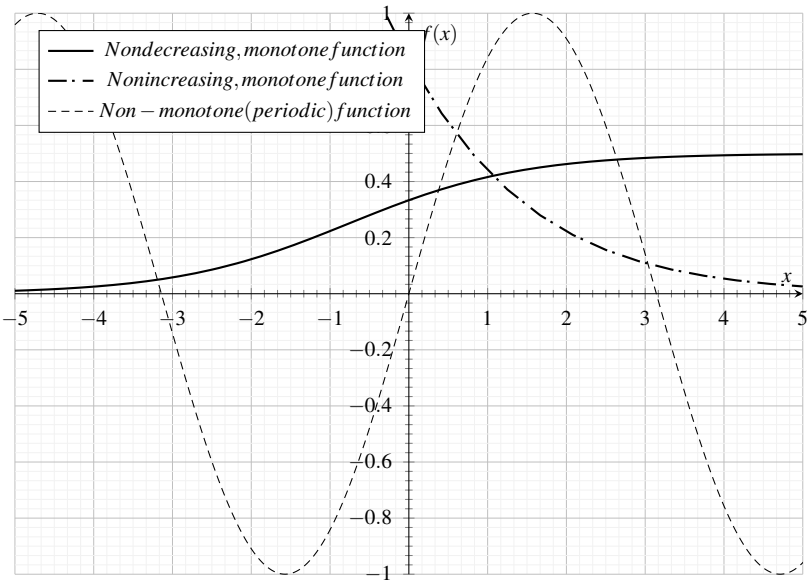


Figure 2.14 : Monotonic Functions.

2.6 Monotonic Activation Functions

In mathematics, monotonicity of any arbitrary function describes its nondecreasing or nonincreasing feature (Royden and Fitzpatrick, 1988). Visually, in Figure 2.14, thick with dotted function gradually decreases (nonincreasing) while thick function mostly increases (nondecreasing).

Conceptually, if a function satistifies that if $\forall x, y \in \mathbb{R}$ and $x \geq y$, then $f(x) \geq$

$f(y)$ is valid for a certain bound without changing the order, which implies nondecreasingness of the function f. Conversely, $\forall x, y \in \mathbb{R}$ and $x \leq y$, so the relationship such as $f(x) \leq f(y)$ is preserved for previously determined bound without changing the order, that is the functions which are nonincreasing are at least weakly monotonic.

2.6.1 Linear function

If an input is thought to be shaped according to weights determined either endogeneously or exogenously (that feed neurons as proportionally determined by hand) to neuron, linear function will be the first considered.
The main drawbacks for this function are two with high priority: First one is its derivative of it which is 1 and the results in that gradient descent are full of 1s and make it a constant gradient that indicates there is no actual relationship with x and gradient descent. Second one, the errors coming from prediction should be corrected with a constant by backpropagation showing there is no dependency in the changes in input (that is $\delta(x)$). The general formula can be stated as following with a small restriction:

$$f(x) = \alpha x, \tag{2.81}$$

where $\alpha \neq 1$, in the range of (-∞, ∞), continuous in C^{∞}, monotonic for itself and first derivative.

If the first parameter, $\alpha = 1$, it simply turns out to be an identity function which is explained in detail in Section 2.6.1.1. Both of these situations are drawn in the Figure 2.15.

2.6.1.1 Identity Function

As stated in Section 2.6.1, if α = 1, it becomes an identity function. At first sight, as stated in Section 2.6.1, an identity function may seem useless since it does not perform any transformation of inputs to neuron. But, actually, it transforms itself because the neuron uses it with no weights. So, an identity function can be called as replicatior or duplicatior for summation of weights ($\Sigma_j w_{kj} x_j$, namely summing junction (Haykin, 2001)) with no weights (Rice, 1953). The general formula given below is also represented in Figure 2.15:

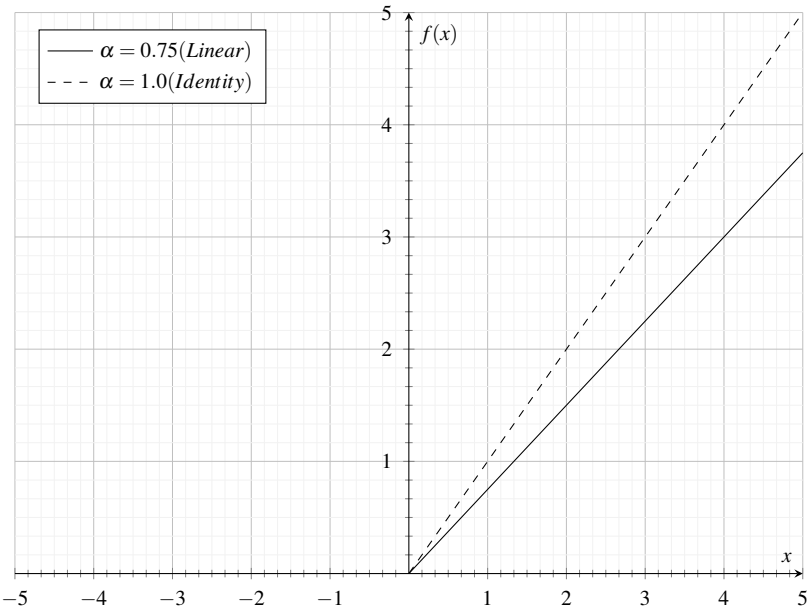


Figure 2.15 : Linear Function.

2.6.1.2 Piecewise linear function

Piecewise linear function in Equation (2.82) restricts the inputs between $\alpha_{\min}$ and $\alpha_{\max}$ in order to produce the values in between 0 and 1 as outputs predicted (Zeng et al., 2010) by dropping other values of inputs where it gives 0 if it is lower than minimum threshold or it produces 1 if it is higher than maximum as drawn in Figure 2.16.

$$f(x) = \begin{cases} 0 & \text{if } x < \alpha_{\min} \\ mx + b & \text{if } \alpha_{\min} \leq x \leq \alpha_{\max} \\ 1 & \text{if } x > \alpha_{\max} \end{cases} \tag{2.82}$$

where,

$$m = \frac{1}{\alpha_{\max} - \alpha_{\min}} \tag{2.83}$$

and

$$b = -m\alpha_{\min} = 1 - m\alpha_{\max}, \tag{2.84}$$

in the range of (-∞,∞), continuous in C^{∞}, monotonic for itself and first derivative.

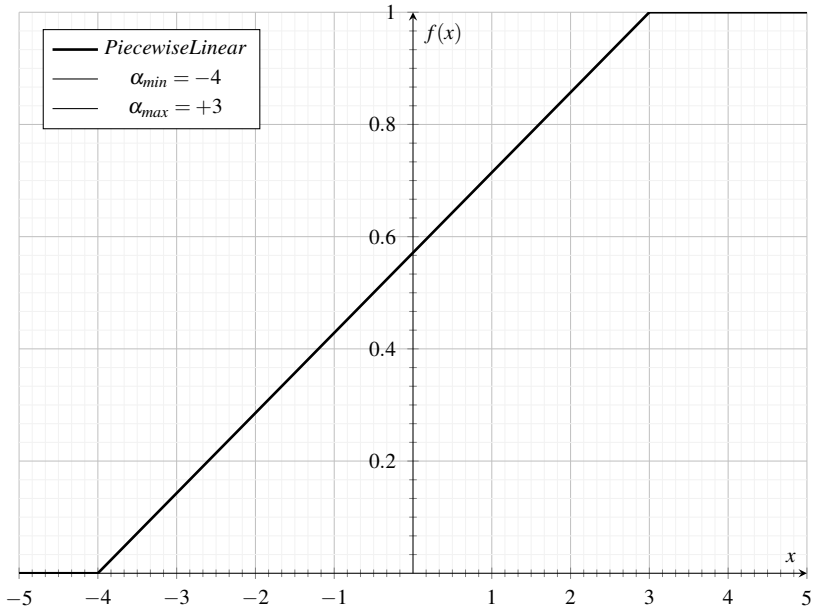


Figure 2.16 : Piecewise Linear Function.

2.6.2 Threshold (unit heaviside, binary, step) function

Well known threshold function (Batres-Estrada, 2015) has three other synonyms as unit Heaviside (Cox, 1992; Osher and Fedkiw, 2003), binary and step function (2.85). In econometrics literature, it is better known these names, it is similar to a dummy variable which can be used alone or interactively with weights as its estimated coefficient. The usefullness of a threshold function comes from its filtering power of inputs and feeding neuron like a bias term; econometrically, initially it helps to change constant term, consequently, it changes ŷ predicted. Following is the threshold function and is represented in Figure 2.85:

$$f(x) = \begin{cases} 1, & \text{if} : x \geq 0, \\ 0, & \text{if} : x < 0. \end{cases} \tag{2.85}$$

in the range of $(0,1)$, continuous in C^{-1}, monotonic for itself.

2.6.3 Sigmoid function

The sigmoid function is one of the most (Batres-Estrada, 2015; Friedman et al., 2001) common types of activation functions used in neural networks, especially in solving classification problems. It is generally used for linear and

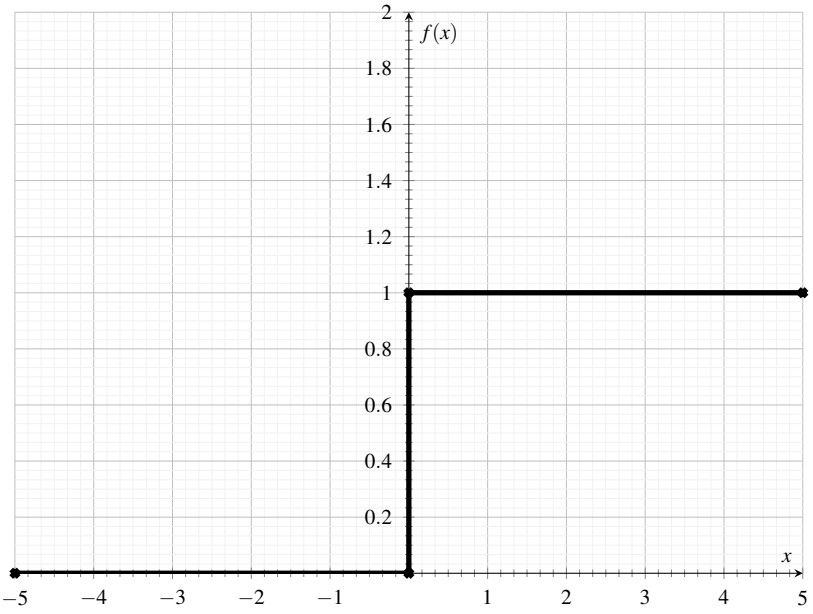


Figure 2.17 : Threshold (Unit Heaviside / Binary / Step) Function.

nonlinear transformation of inputs in order to feed other neurons as balancer. The general formula stated below is represented in Figure 2.18:

$$f(x) = \frac{1}{1+e^{-\alpha x}}, \tag{2.86}$$

in the range of $(0,+1\ )$, continuous in C^{∞}, monotonic for itself.

The sigmoid function (2.86) was first invented by Verhulst (1838) and used for estimating population growth (Verhulst, 1977)[33] as a logistic function or logistic curve. In econometric terms, if the α parameter[34] equals one, it is called a logistic function (Greene, 2003) and used for calculating probabilities when the dependent variable is a dichotomous one. Since it can also be classified in antisymmetric functions comparing origins, backpropagation process may converge faster in a training process (Haykin, 2001; LeCun et al., 2012).

2.6.3.1 Bipolar sigmoid function

Bipolar sigmoid function is a sigmoid function with a small difference, it produces values in the range of (-1,+1). So it cannot be used for probabilistic estimations since probability cannot be lower than zero. Panicker and Babu

[33]Reprinted edition.
[34]It is called steepness level of sigmoidals.

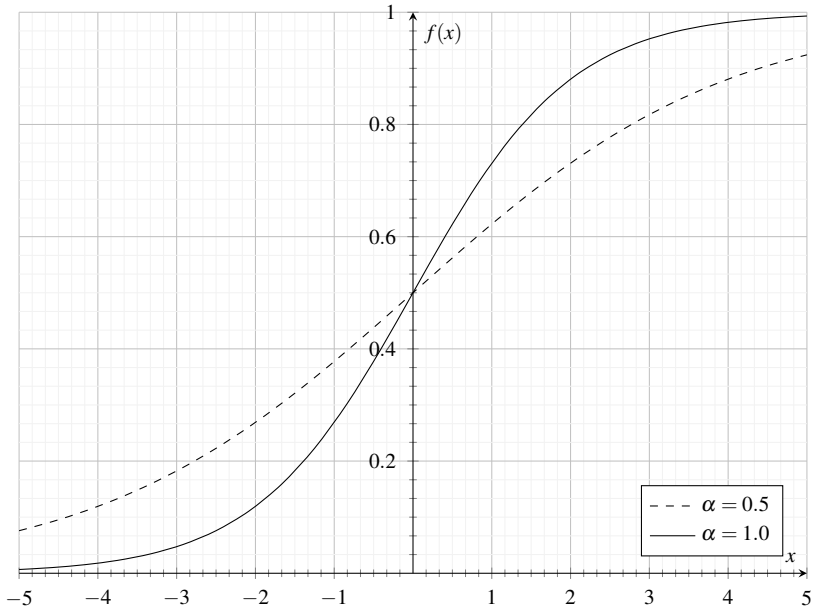


Figure 2.18 : Sigmoid Function.

(2012) find that a bipolar sigmoid is more efficient than other sigmoidals. The general formula is shown in Figure 2.19 and is given below Equation 2.87:

$$f(x) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}}, \tag{2.87}$$

where α is the smoothness parameter of a bipolar sigmoid that an activation function can be defined in the range of $(-1, +1)$, continuous in C^{∞}, monotonic for itself.

2.6.4 Rectified linear unit (ReLU)

Via the abbreviated and capitalized name, ReLU is a recent newcomer called in the existing literature as Ramp function and advanced function (Nair and Hinton, 2010) that provides numerous replications from the sigmoidal function (Batres-Estrada, 2015) with the same learned weights and biases determined in the training process. The formula is given blow as shown in Figure 2.18:

$$f(x) = \begin{cases} \alpha x, & \text{if} : x < 0, \\ x, & \text{if} : x \geq 0. \end{cases} \tag{2.88}$$

where $\alpha = 0$, in the range of $[0, +\infty)$, continuous in C^0, monotonic for itself and its derivative.

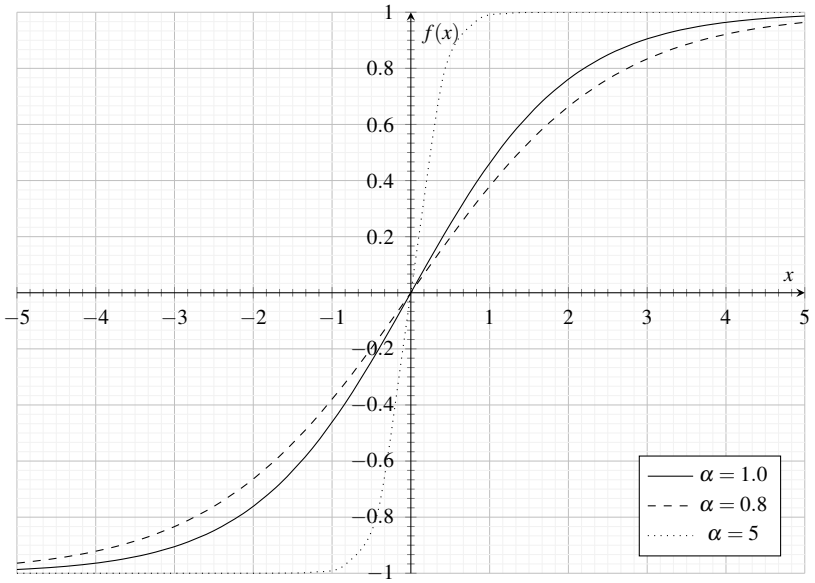


Figure 2.19 : Bipolar Sigmoid Function.

As stated in Equation (2.88), representation of RELU does not cover an α parameter, this is put there to easily identify other family members of RELU. Figure 2.20 displays only the approximation of RELU, as it was stated peviously, it replicates a large number of sigmoids. This is why it is different from a linear function, the nature of RELU is nonlinear. If the neural network is so complex, having tens or hundreds of neurons, most of them may be affected by other activation functions such as tanhs, softpluses, sinusoids, sigmoids and Gaussians. RELU was invented to burst the speed of training (Nair and Hinton, 2010) and make it easier to generalize properties of most of the linear models (Goodfellow et al., 2016).

2.6.4.1 Leaky rectified linear unit (LReLU)

The first member of the ReLU family of functions is the Leaky Rectified Linear Unit (LReLU) which is simply defined as Equation (2.88) when the parameter $\alpha = 0.01$, in the range of $(-\infty, +\infty)$, continuous in C^0, monotonic for itself and its derivative (Maas et al., 2013). It is shown in Figure 2.20.

2.6.4.2 Parametric rectified linear unit (PReLU)

The second member of ReLU is the Parametric Rectified Linear Unit (PReLU) which can be represented as Equation (2.88) without a restriction on α

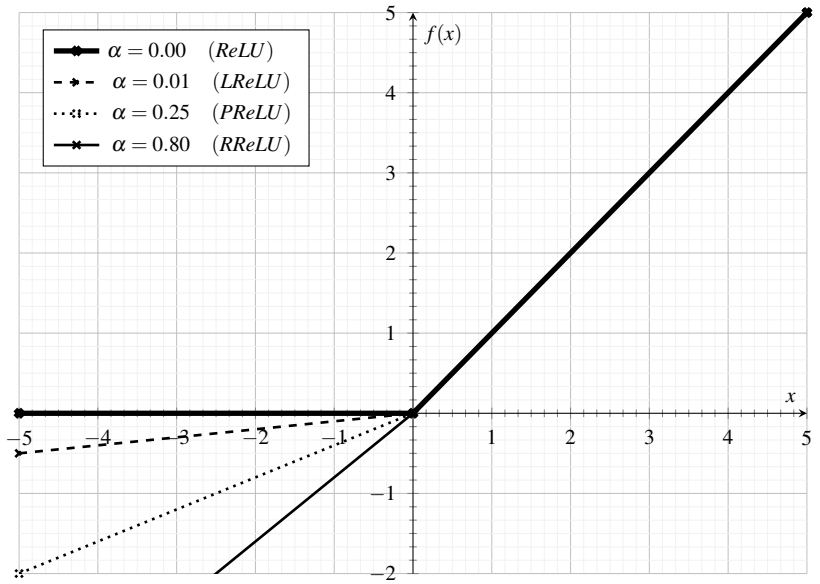


Figure 2.20 : Rectified Linear Unit (ReLU) Family.

parameter, where in the range of $(-\infty, +\infty)$, continuous in C^0, monotonic for itself, if and only if $\alpha \geq 0$, and its derivative. It is shown in Figure 2.20.

2.6.4.3 Randomized rectified linear unit (RReLU)

As the name of this member of the ReLU family explains its function, Randomized Rectified Linear Unit (RReLU) randomly selects an α parameter from $(-\infty, +\infty)$ as declared in Equation (2.88) without any restriction (Xu et al., 2015). The main flaw of RReLU is that the difference between PReLU and RRelu cannot be identified when a backpropagation process in input (that is $\delta(x)$). It is defined where in the range of $(-\infty, +\infty)$, continuous in C^0, monotonic for itself, if and only iff $\alpha \geq 0$, and its derivative. It is drawn in Figure 2.20.

2.6.5 Exponential linear unit (ELU)

Exponential Linear Unit (ELU) is another variant of ReLU and its family but the second statement presented in Equation (2.89) is totally disparate from ReLU and its family (Nair and Hinton, 2010). It is proven that ELU is one step ahead of ReLUs explained previously (Clevert et al., 2015; Trottier et al., 2016). It converges the mean of activation functions close to zero, the process decreases process time required to train a dataset containing particularly computer visions tasks.

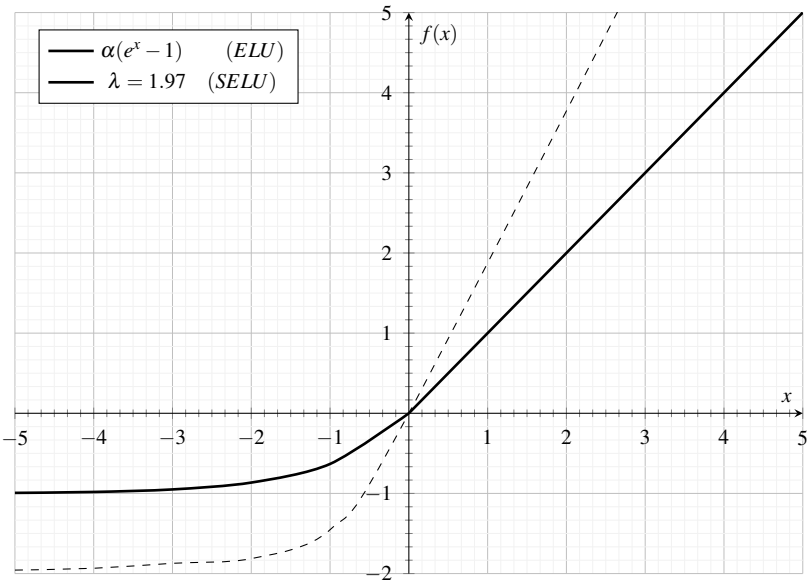


Figure 2.21 : Exponential Linear Unit (ELU).

$$f(x) = \begin{cases} \alpha(e^x - 1), & \text{if} : x < 0, \\ x, & \text{if} : x \geq 0. \end{cases} \tag{2.89}$$

In the range of $(-\alpha, +\infty)$, continuous in C^1 if $\alpha = 1$ otherwise in C^0 , monotonic for itself, if and only iff $\alpha \geq 0$, and its derivative iff $\alpha \in [0,1]$. As whown in Figure 2.21 and stated above in Eq. (2.89), ELU's limit of converging -∞ will be -1. That is, negative values of x doesn't affect it.

2.6.5.1 Scaled exponential linear unit (SELU)

Scaled Exponential Linear Unit (SELU) is derived from ELU with a λ that may be determined exogenously by hand or endogenously by the speed of self-normalization of neuron (Klambauer et al., 2017). It permits to run recursively normalization of a neuron itself and to deal with the vanishing gradient descent (Clevert et al., 2015). SELU is shown in Figure (2.89) and is stated below as Equation (2.90):

$$f(x) = \lambda \begin{cases} \alpha(e^x - 1), & \text{if} : x < 0, \\ x, & \text{if} : x \geq 0. \end{cases} \tag{2.90}$$

2.6.6 Softmax function

As previously explained, a dichotomous dependent variable that needs to calculate probabilities should be estimated with sigmoid/logistic (2.86) functions. If there are multiple, but discrete choices, in a dependent variable,

it is basically estimated with using a multinomial logistic function which is frankly a generalization of sigmoid functions (Friedman et al., 2001). In machine learning literature, most of the sigmoidals are generalized into softmax function in order to calculate multiple outcomes for each choice as a vector of probabilities. Well known function in machine learning literature is given below:

$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}, \tag{2.91}$$

in the range of $(0, +1)$, continuous in C^{∞}.

Softmax function is not only popular in artificial neural networks, but also frequently used in Naïve Bayes ,linear discriminant analysis and multinomial logistic regression. The only and tiny difference, statistically it generates a significant differentiation, is that these multiple discrete choices must be mutually exclusive classes, there should not be any dependency between the classes processed with the softmax function.

2.6.7 Odd activation (signum, sign) function

As formulated below, the Odd activation function is as follows Equation 2.92 drawn in Figure 2.22:

$$f(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \tag{2.92}$$

where $\alpha = 0$, in the range of $(-\infty, +\infty)$, continuous in C^0, monotonic for itself and its derivative.

2.6.8 Maxout function

Hinton et al. (2012) are looking for a way to reduce training time of models consisting of a few ensemble models and finds the technique called Dropout while it is also examined that estimated predictions should be harmonized by averaging them. Goodfellow et al. (2013) offer a new feed-forward artificial neural network model including a recently invented activation function called maxout where the activation function is the maximum of the inputs fed. It is

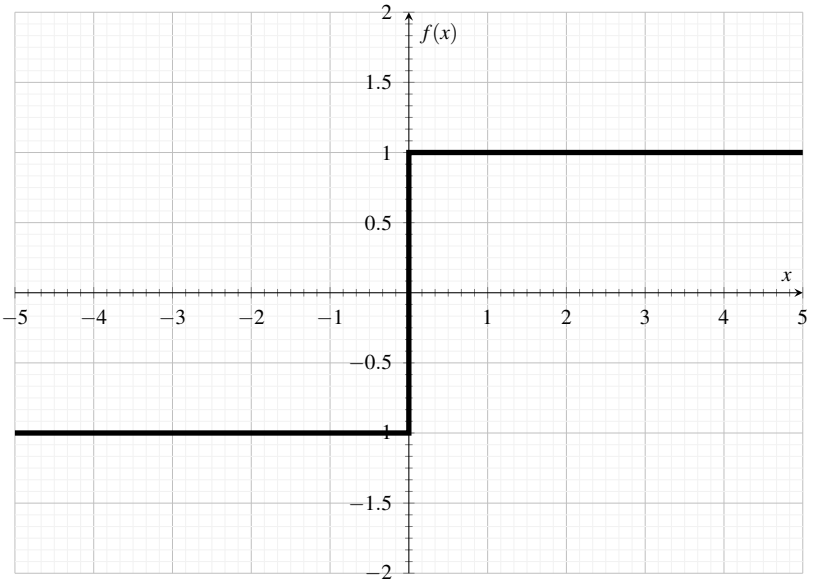


Figure 2.22 : Odd Activation (Signum / Sign) Function.

so useful with Dropout (Goodfellow et al., 2013). The form of this activation function is given below:

$$f(\vec{x}) = \max_i x_i, \tag{2.93}$$

where in the range of (-∞, +∞), continuous in C^0.

Goodfellow et al. (2013) provide that when the number of epochs exceeds approximately 30, maxout maintains significant improvements in the large cross validation set. Also, model averaging on MNIST classification using maxout function beats other rectifiers.

2.6.9 Softsign function

Some functions are useful when they play role by balancing two or more functions and may be better tools. The softsign, which is also sigmoidal, is such a function and looks like a smoothed version of both tanh in Equation2.96 (with high weight) and sigmoid function and a bit odd activation function (with less weight). It also approximates identity function with its derivate at origin (Aghdam and Heravi, 2017). In some instances, it may be considered an alternative to tanh function as it does not saturate as easily as hard clipped

functions. The function is stated in Equation (2.94) below and visualized in Figure 2.23:

$$f(x) = \frac{x}{1+|x|}, \tag{2.94}$$

in the range of $[-1,+1]$, continuous in C^{∞}, monotonic for itself.

The most significant property of the softsign function is the saturation level which will be higher as $|x|$ increases, saturation level in comparison with tanh is a bit softer compared to other functions. This property makes it compute at ease.

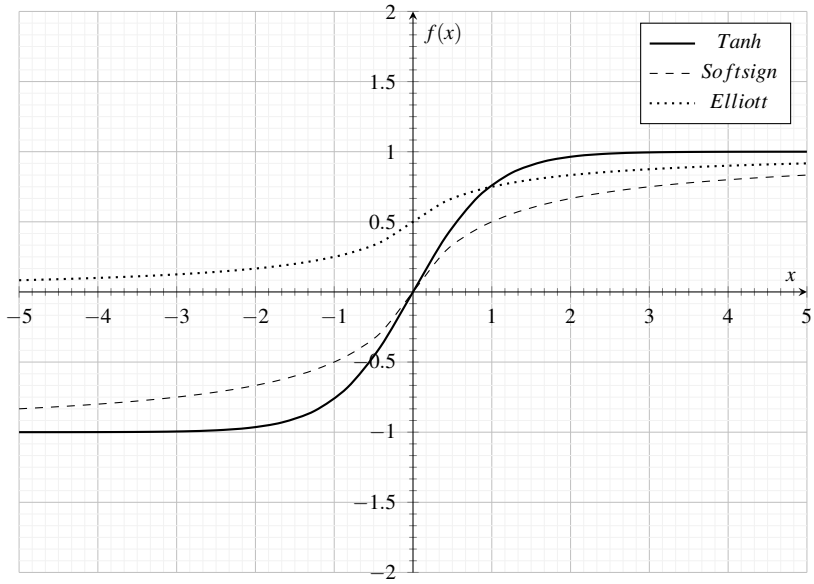


Figure 2.23 : Softsign, Hyperbolic Tangent and Elliot Function.

2.6.10 Elliott function

Elliott function (Elliott, 1993) looks like another sigmoidal which produces results between 0 and +1 for output processed with Elliott. It performs very-well and does not suffer from neuron vanishing problems according to applications done in Matlab (Ploskas and Samaras, 2016). Its formula is given below Equation (2.95) and drawn in the Figure 2.23:

$$f(x) = \frac{0.5(x)}{1+|x|} + 0.5, \tag{2.95}$$

in the range of $[0,+1]$, continuous in C^{∞}, monotonic for itself.

2.6.11 Hyperbolic tangent (Tanh) function

When Sauri (1774) first introduces the hyperbolic tangent function, which is another sigmoidal, in short "tanh". It becomes another well-known function in machine learning literature. Although it comes from $\frac{sinh(x)}{cosh(x)}$ and the shape of it resembles sigmoid and softsign functions a bit, it actually is a rescaling of a sigmoid function. Tanh and its rescaling procedure are given in Equation (2.96, 2.97,2.98) and is illustrated in Figure 2.23:

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1, \tag{2.96}$$

where in the range of $(-1,+1)$, continuous in C^{∞}, monotonic for itself. Which can also be derived from a sigmoid function;

$$tanh\,(x) = 2 \cdot \sigma\,(2x) - 1, \tag{2.97}$$

where $\sigma(x)$ is,

$$\sigma(x) = \frac{e^{x}}{1+e^{x}}. \tag{2.98}$$

The most valuable contribution of tanh and sigmoid for machine learning literature is that they can map any real number determined in the range of $(-\infty,+\infty)$ to a number in between [-1,+1] with tanh or [0,1] with sigmoid function respectively. These procedures are called normalization which makes internal covariates accurate predictors by generating batch normalization so as to normalize each vector having zero mean and unit variance. LeCun et al. (2012) also point out that batch normalization is beneficial for backpropagation procedures.

2.6.11.1 Arc tangent function

When the upper and lower thresholds do not satisfy the conditions offered by tanh, sigmoid and softsign, another sigmoidal, the arc tangent function, “arctan”, can be used, which offers half of π symmetrically to the origin as thresholds of f(x) (Aghdam and Heravi, 2017). It can be used as a normalizer

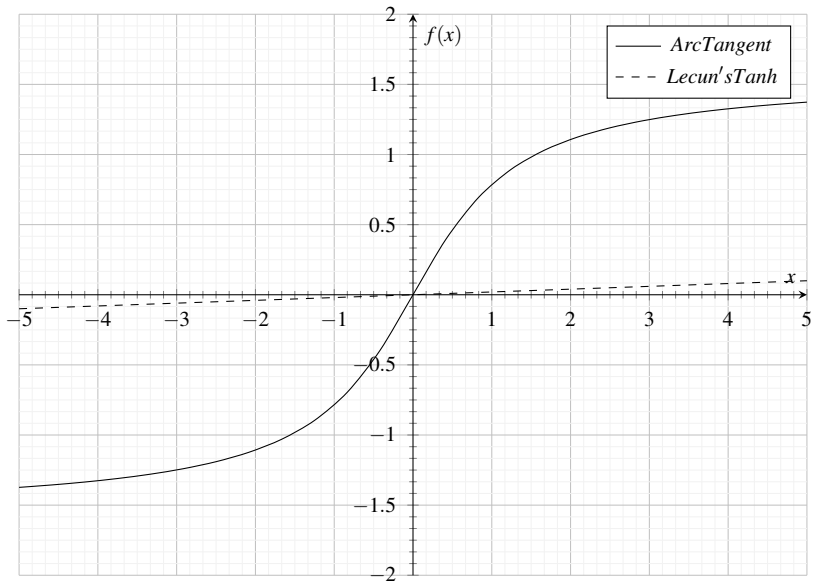


Figure 2.24 : Arc Tangent Function.

as other sigmoidals. The shape of it is shown in Figure 2.24 and given in Equation (2.99):

$$f(x) = \tan^{-1}(x), \tag{2.99}$$

in the range of $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, continuous in C^{∞}, monotonic for itself.

2.6.11.2 Lecun's hyperbolic tangent function

An experimental activation function from Lecun can be used as a normalizer again as with other sigmoidals. The shape of it is shown in Figure 2.24 and given in Equation (2.99):

$$f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right), \tag{2.100}$$

in the range of $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, continuous in C^{∞}, monotonic for itself.

2.6.12 Complementary log-log function

Complementary log-log function which is the inverse of the c.d.f. of the extreme value (or log-Weibull) distribution and is used to compute hazard ratios is quite similar to the sigmoid family. It again provides the values between 0 and 1, but the results taken from an activation function as output can be interpreted as the hazard effect of reverse extreme value

errors. Gomes and Ludermir (2008) point out that it outperforms logit and tanh activation functions according to mean squared errors of multi layer perceptron networks. Cloglog, for short, is given in Figure 2.25 and is given in Equation (2.101).

$$f(x) = 1 - \exp\big(-\exp(x)\big), \tag{2.101}$$

in the range of $(0, +1\ )$, continuous in C^{∞}, monotonic for itself.

In general, the inverse cloglog version is commonly used as an activation function which solves classification problems.

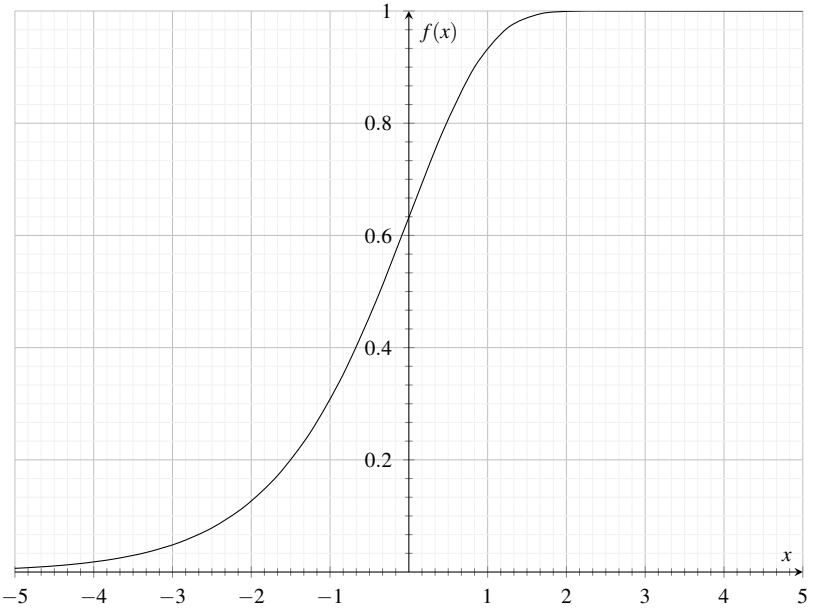


Figure 2.25 : Complementary Log-Log Function.

2.6.13 Softplus function

Since Glorot et al. (2011) emphasizes the significance of softplus function and Zheng et al. (2015) proves the improvements in deep neural networks (deep learning) by using softpluses, it is used a new rectifier rather than using $x- > max(0,x)$, that is ReLU, by smoothing its negative values. This smooth rectifier is given below in Equation (2.102) and its graph is presented in Figure 2.26:

$$f(x) = \ln(1 + e^{x}), \tag{2.102}$$

in the range of $(0, +\infty)$, continuous in C^{∞}, monotonic for itself and its derivative.

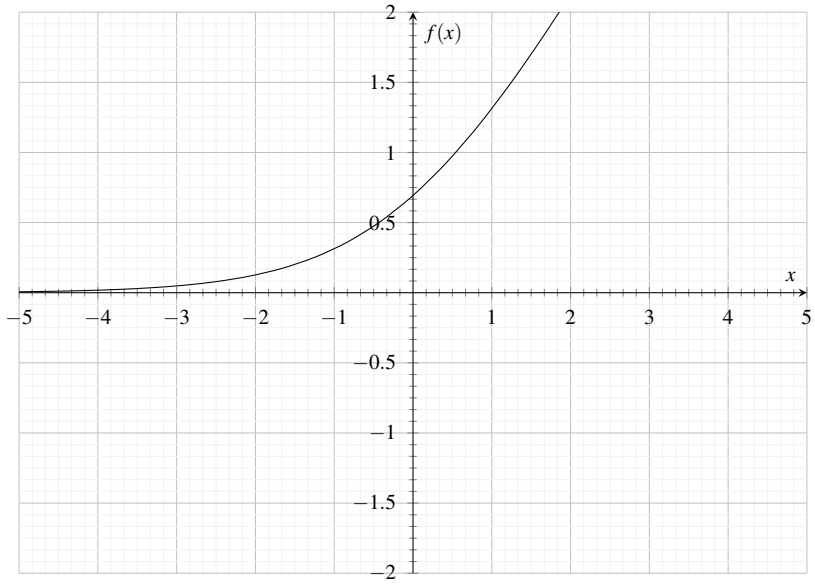


Figure 2.26 : Softplus Function.

2.6.14 Bent identity function

After they were first defined in 1960s, Bent functions were published for the first time by Rothaus (1976). At the same time, Bent identifiers were being used for cryptography in USSR by V. A. Eliseev and O. P. Stepchenkov (Tokareva, 2015). Bent functions can be classified in the Boolean function family (Çeşmelioğlu et al., 2016; Savickỳ, 1994). Bent identity function is given below in Equation (2.103) and is visualized in Figure 2.27:

$$f(x) = \frac{\sqrt{x^2+1}-1}{2} + x, \tag{2.103}$$

in the range of $(-\infty, +\infty)$, continuous in C^{∞}, monotonic for itself and its derivative.

Bent identity is an interesting function, it smoothly translates negative values of inputs to higher values and positive values of inputs to lower values. Nowadays, at the time of writing this dissertation, it is used for solving ethereum cripto-money problems if nonlinearity is needed to solve problems and cannot be solved by using other rectifiers.

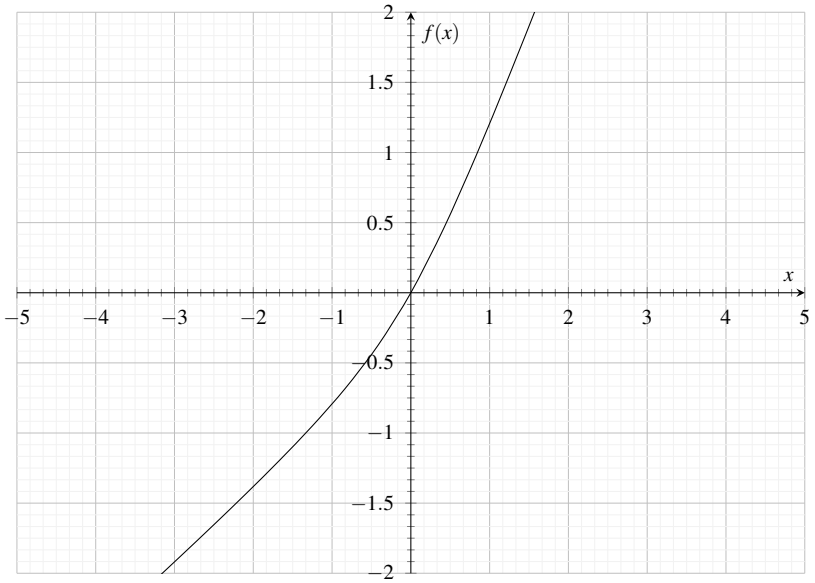


Figure 2.27 : Bent Identity Function.

2.6.15 Soft exponential function

Soft exponential function as given in Equation (2.104) is recently proposed by Godfrey and Gashler (2015) who assert that the main property of SoftE is well-suited activation function for neural networks without empirical evidences. Soft exponential functions according to some α levels are drawn in Figure 2.28:

$$f(\alpha,x) = \begin{cases} -\frac{\ln(1-\alpha(x+\alpha))}{\alpha} & \text{for} \quad \alpha < 0 \\ x & \text{for} \quad \alpha = 0 \\ \frac{e^{\alpha x}-1}{\alpha} + \alpha & \text{for} \quad \alpha > 0 \end{cases}, \tag{2.104}$$

in the range of $(-\infty,+\infty)$, continuous in C^{∞}, monotonic for itself and its derivative.

Godfrey and Gashler (2015) present SoftE as a function that harmonizes the characteristics of logarithmic, linear and exponential functions since its multiplication can be done in logarithmic space and that is easily computable, differentiable and parameterized in the needs of neural network structure. Another strong property is that SoftE does not suffer from vanishing problem when training period is run.

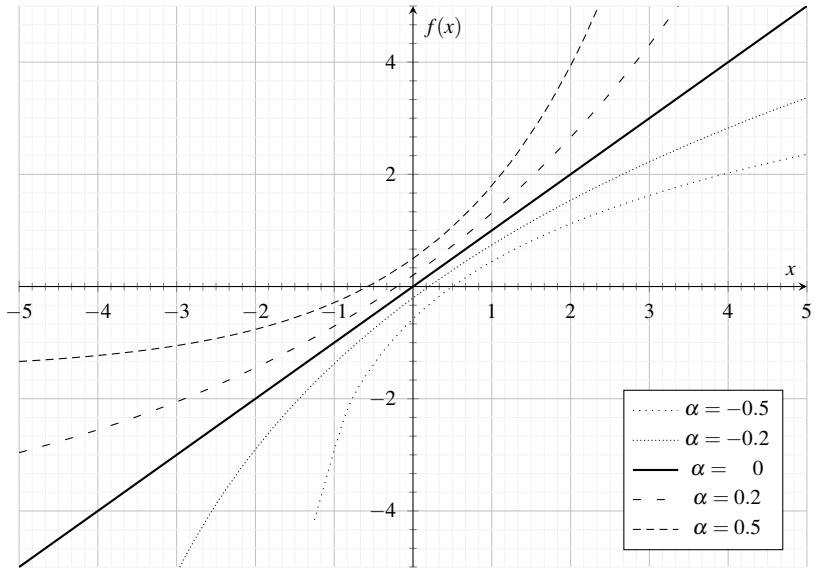


Figure 2.28 : Soft Exponential Function.

2.7 Periodic Activation Functions

Activation functions covered so far are well suited for the neural networks that have a stable architecture that is well known and frequently used. These type of ANNs include a certain activation layer which is not varied by other neurons. Another architecture type of ANNs is named as adaptive artificial neural networks which consist of at least one additive neurons that affect an activation layer producing one or more outputs simultaneously. These classes of architectures need periodic activation functions where their derivatives result in oscillated, unstable outputs. This is only because adaptive neural networks contain both the weights of inputs and the learning rate of network (Widrow and Lehr, 1993). It may also change errors by adding recursively to the model.

2.7.1 Sinusoidals

Basic differentiation of a sinusoidal function compared to other types of functions comes from the existence of a repetition of oscillations in a certain period that can be modeled with a sine function and its derivatives. It does not matter whethet these oscillations represent the same amplitude or not.

2.7.1.1 Sine wave function

Sine wave[35] is a geometric form of wave oscillating periodically with a certain amplitude looking like S-shaped wave around the x-axis in the range of -1 and +1 (Parascandolo et al., 2016). It is called and abbreviated as sine given in Equation 2.105 which results in its first derivative as cosine, but first derivative of cosine is minus sine.

$$f(x,t) = A\sin(kx \pm \omega t + \varphi) + D \tag{2.105}$$

in the range of $(-\infty, +\infty)$, continuous in C^{∞}, t is the oscillation period, (-) sign is for right moves, (+) sign is for left moves.

The main and most common use of sine wave functions as illustrated in Figure 2.29 is signal processing. Since sine or cosine are non-quasiconvex functions and therefore non-monotonic, it is useful if an increase or decrease in the correlation or dependency between activation and input oscillate the activation function in the learning process. This may easily cause the learning process to derail if the number of epochs increases or learning rate is higher than its optimal level; in addition to this, this can cause a decrase the speed rate of learning, gradient vanishing or converging local minima -rather than global - problem as Lapedes and Farber (1987) stated. In recent studies, it is shown that well-known periodic activation functions such as sine may easily suffer from high ram swelling issue, however, sine performs better than other monotonic functions in Recurrent Neural Networks (RNN) while predicting in short term (Alquezar and Sanfeliu, 1994; Sopena and Alquezar, 1994). Another property of sine and cosine is the averages of them are zero, which is useful property especially for statisticians.

2.7.1.2 Cardinal sine function (Sinc)

Cardinal sine function, abbrevated as sinc, is the candidate for the weirdest activation function among both monotonic and periodic ones. Şelariu et al. (2014) state the concept of cardinality as "a number equal to a number of finite aggregate" and classify it in centric functions because it is centered on

[35]The first song of Mogwai's "Rock Action" album that is revolutionary in post-rock genre.

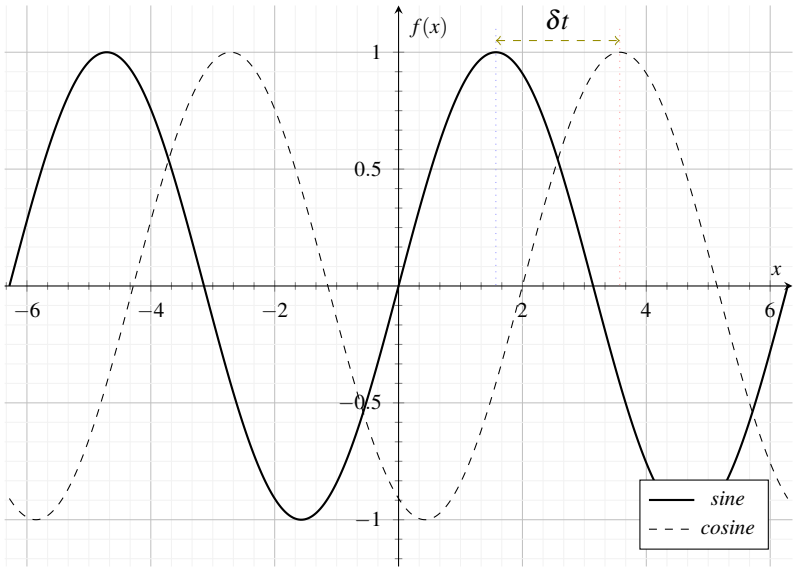


Figure 2.29 : Sine and Cosine Wave Functions.

the x-axis while waving around it, like a Mexican hat. In general, as given in Equation (2.106), it is normalized by using π as Equation (2.107), both of which them are shown in Figure 2.30.

$$f(x) = \frac{\sin(x)}{x} \tag{2.106}$$

and normalized version of Equation (2.106),

$$f(x) = \frac{\sin(\pi * x)}{\pi * x} \tag{2.107}$$

in the range of $(-\infty, +\infty)$, continuous in C^{∞}.

The most interesting point for the normalization of the sinc activation function is that it can result in a Fourier transform of a rectangular function that may be used for getting 1 for determined interval of x values like rectifier units as stated in Chapter 2.6.4.

2.7.1.3 Fourier transform (FT, DFT/FFT)

There are three concepts that must be explained respectively. The first one is fourier analysis, that area is a way to represent functions as an exact or approximated version by means of summing certain trigonometric functions. Secondly, the Fourier series in the Fourier analysis is generally related to making a representation by simply adding sine waves to each other. Fourier transformation, however, by using fourier series, converts a series to an

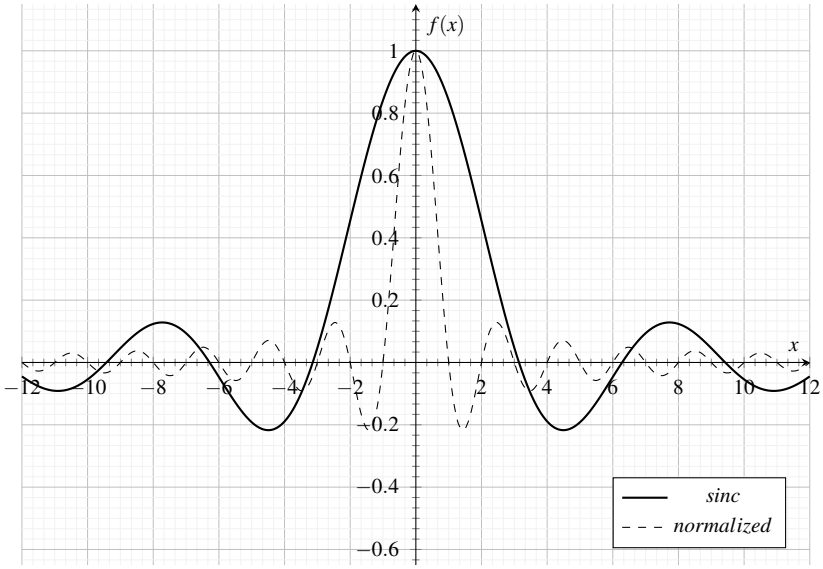


Figure 2.30 : Sinc Function.

orthogonal one into new space with its linear form (Bracewell and Bracewell, 1986). This transformation also helps state differential equations as linearized. A continuous fourier transformation is described in Equation (2.108) and (2.109) and is visualized in Figure 2.31.

$$\hat{f}(k) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x k}\, dx, \tag{2.108}$$

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(k)\, e^{2\pi i k x}\, dk, \tag{2.109}$$

in the range of $(-\infty, +\infty)$.

Fourier transformation has some properties which are linearity, frequency and time shifting, time scaling, which are useful when they are processed as activation functions. Additionally, the discrete version (DFT) which is not described and shown can be used for image processing, sound waves if the data are finite or the processing of them are relatively trivial because it helps to transform it easily by metamorphosing into a fast fourier transform (FFT) (Wickerhauser, 1994).

Meaning of fourier transform literally is the sum over all time of the inputs of f(x) multiplied by an exponential and the result is the Fourier coefficients $\hat{f}(k)$, when these coefficients are multiplied by a sinusoid of appropriate frequency k, they yield the constituent sinusoidal component of the original inputs.

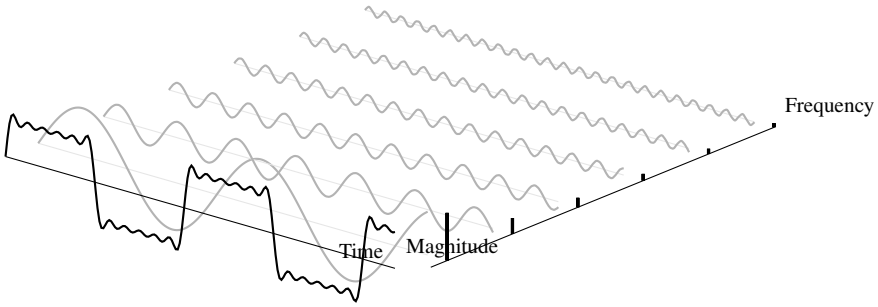


Figure 2.31 : Fourier Transform for Continuous Series.

For the relationship between Equation (2.108) and (2.109) should be denoted as $f(x) \mapsto \hat{f}(k)$, meaning "Fourier transform" of these two equations mapped, and $\hat{f}(k) \mapsto f(x)$, meaning "Inverse fourier transform" of the equations mapped. This mapping procedure is abbrevated with capital letters such as FT and IFD.

2.7.1.4 Discrete-time dimensional fourier transform (DTFT, Shannon-Nyquist)

Discrete-time dimensional Fourier transform (DTFT) having time domain where y-axis contains ampitudes of inputs (signals) is another version of Fourier Transform. There are two types of DTFT, which are periodic and aperiodic according to Memon et al. (2014), but in general, aperiodic ones are relatively hard to model and convert into a serie. DTFT is given in Equation (2.110) and (2.111)

$$X_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-i\omega n}. \tag{2.110}$$

$$X_{1/T}(f) = X_{2\pi}(2\pi f T) \overset{\text{def}}{=} \sum_{n=-\infty}^{\infty} \underbrace{T \cdot x(nT)}_{x[n]}\, e^{-i2\pi f T n} \overset{\text{Poisson f.}}{=} \sum_{k=-\infty}^{\infty} X\left(f - k/T\right), \tag{2.111}$$

in the range of $(-\infty, +\infty)$.

2.7.1.5 Short time fourier transform (Gabor, STFT)

Short time or short-term fourier transform (STFT), in general, divides whole time into small parts which are equal, afterwards, for each part, fourier transforms are calculated in order to determine sinusoidal frequencies and

are respectively bound to see all local changes of each part. The concept of short-term indicates each part or phases that were seperated into, so it can be called a mix of time dimensional and fourier transform.

Kalayci and Ozdamar (1995) regard it as a very beginning version of wavelets for an activation function and assert that it has some drawbacks such as Heisenberg's Uncertainty Principle stated as inequality: $time * bandwidth = \Delta f \Delta t \geq \frac{1}{4\pi}$. It indicates neither times nor bandwith may be jointly small. In general, STFT is used to analyse audio signals especially those which have harmonic patterns (Wu and Liu, 2008). It also fails when the data are nonstationary which causes arising asymmetries between phases or parts.

STFT has two sub-categories according to its time dimension, it can be discrete or continuous. Discrete version of STFT is given below in Equation (2.112, where the data must be split into frames :

$$\mathbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \tag{2.112}$$

In continuous case, a window function recalled as Gaussian window that centered around zero and expected value of that is zero for multiplying it for short-terms is needed again with an R parameter. R parameter is a weight for time axis as given in Equation (2.113).

$$\sum_{n=-\infty}^{\infty} x(n)w(n-mR)e^{-j\omega n} \tag{2.113}$$

The dependency between discrete and continous STFT is given below:

$$\mathrm{DSTFT}_{\omega}(x \cdot \textsc{Shift}_{mR}(w)), \tag{2.114}$$

in the range of $(-\infty, +\infty)$.

In Equation (2.114), shift in time axis is the R parameter that can be determined exogenously or endogenously.

2.7.1.6 Wavelet transform

Wavelet is seen in the compression of images with JPEG2000 format, thanks to decreasing the lengths of inputs of images to half of it, so size of images

could be decreased. Basically, when Fourier says frequencies of inputs, wavelet indicates both frequencies and time, in econometric terms, wavelet is actually a panel data version of the fourier series. Differences between all fourier variants and wavelets are shown in Figure 2.32.

According to Hippenstiel (2001), wavelets have one dimensions for time and two dimension for transform. If both inputs and outputs are continuous, then it is called as continuous wavelet transform (CWT), the most common transform of wavelets (Ombao et al., 2016) used as an activation function (Pati and Krishnaprasad, 1993). If the outputs are discrete while inputs are continuous, it is called as discrete wavelet transform (DWT). The functions for them are given below respectively:

$$X_w(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\,\psi^* \left(\frac{t-b}{a} \right) dt \tag{2.115}$$

in the range of $(-\infty, +\infty)$.

The wavelet coefficients for continuous series of transformation are computed as following:

$$c_{jk} = \left[W_\psi f \right] \left(2^{-j}, k2^{-j} \right), \tag{2.116}$$

in the range of $(-\infty, +\infty)$.

Furthermore, DWT is as given below:

$$\psi(x) = \sum_{k=-\infty}^{\infty} (-1)^k a_{N-1-k}\, \psi(2x-k), \tag{2.117}$$

in the range of $(-\infty, +\infty)$.

In addition to these, there are two other types of wavelet transforms (Hippenstiel, 2001) which are not described in detail: If inputs are given with time detail with discrete form, it is called as discrete-time wavelet transform (DTWT) but outputs of transform are also discrete. Besides, if outputs transformed are continuous, then it is called as discrete-time continuous wavelet transform (DTCWT). Mostly, continuous cases are harder to implement than discrete ones, the problems faced may be turned into discrete case and then be solved. It is less costly in the implementation period for both processing and computing, which are the most common problems in

machine learning if they are used as an activation function that is vulnerable and sensible to the vanishing gradient problem that is not desired.

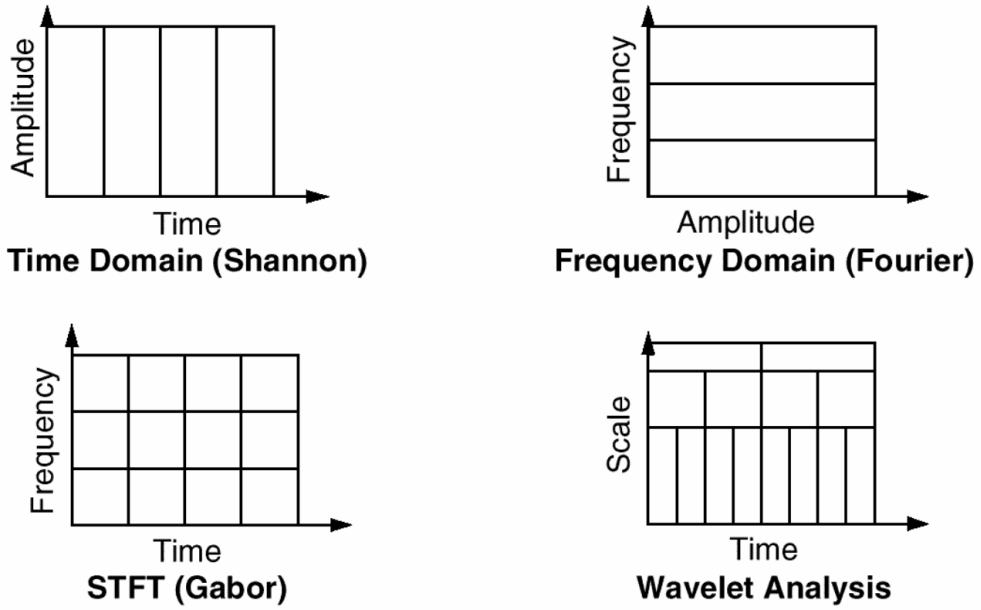


Figure 2.32 : Fourier and Its Variants by Domain.

2.7.2 Non-sinusoidals

As it can be understood from the title, in this section certain activation functions that does not include any sinusoidal function are covered. The functions classified into Non-Sinusoidal that can be periodic or not, it is not a must but these functions can be expanded into periodic.

2.7.2.1 Gaussian (normal) distribution function

The nature of Gaussian which might be the oldest function in the history of statistics (Stigler, 1986) comes from its nonperiodicity, since it produces a reference distribution at once. It does not need to explain its reputation in statistics. But it is required that periodicity, a property that can be applied to most non-monotonic functions, can be introduced by adding a periodic pattern to most functions. The nature provides that the properties of everything living or existing in it are normally distributed if these properties are eventually found when time dimension increases. Thank to Laplace, the central limit theorem proves that assertion when the number of observations increases (Bishop, 2006). So, if a distribution which has less observations about the information tends to turn into Gaussian distribution if N increases. Lin

et al. (2004) firstly use Gaussian as a regularization parameter of estimation. The figure of periodic Gaussian is shown in Figure 2.33 and are given in Equations(2.118 and 2.119):

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2.118}$$

in the range of $(-\infty, +\infty)$ and continuous in C^{∞} .

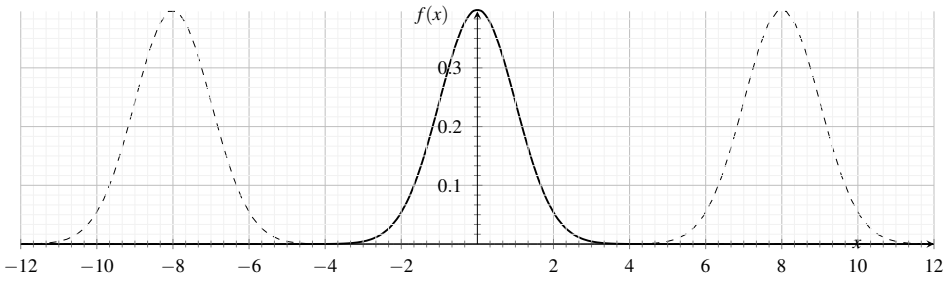


Figure 2.33 : Periodic Gaussian Function.

The periodicity of Gaussian or normal distribution function can be given by implementing the following equation in the constrained space:

$$p(x) = f\left(\left(\left(x + \frac{N}{2}\right) \bmod N\right) - \frac{N}{2}\right) \tag{2.119}$$

On the other hand, by definition, wrapped normal distribution is another kind of normal distribution which wraps around a unit circle and is widely used in directional statistics, another periodic version of normal distribution function is called as Von Mises distribution function that approximates a wrapped one.

2.7.2.2 Square wave function

Square wave can be thought of as an expanded version of the Heaviside step function by implementing its property of periodicity. It is also an application of binary transmission of a wave from a base level to the desired level. Effects of electric guitars, such as overdrive and distortion effects, are examples of a square wave function. Orthogonality of waves are required according to Thompson et al. (2001) for square waves. The function that produces square waves is given below in Equations (2.120) and (2.121) representing

the resemblance to the Heaviside unit function if it is denoted as u(t)) and is illustrated in Figure 2.34.

$$x(t) = \text{sgn}(\sin[t]),\ v(t) = \text{sgn}(\cos[t]) \tag{2.120}$$

in the range of $(-\infty, +\infty)$ and continuous in C^{∞} .

$$x(t) = \sum_{n=-\infty}^{+\infty} \sqcap(t - nT) = \sum_{n=-\infty}^{+\infty} \left(u\left[t - nT + \frac{1}{2}\right] - u\left[t - nT - \frac{1}{2}\right]\right), \tag{2.121}$$

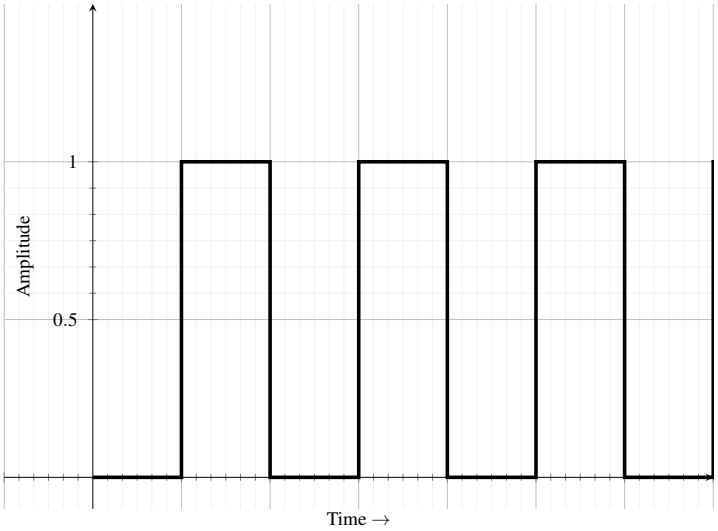


Figure 2.34 : Square Wave Function.

2.7.2.3 Triangle wave function

The shape of this non-sinusoidal function gives it its name as it draws small triangles around the x-axis (Tansel et al., 1991). As Foresee and Hagan (1997) used a triangle wave function to estimate Gauss-Newton approximation to Bayesian regularization (GNBR) algorithm for a three real world problem with application of a univariate regression, time series estimation and an artificial chaotic series model regardless of its contents. They generate the lowest costs by using triangle waves for artifical neural networks including a regularization that improves the results leading to significant decrease in errors. The functional form of the triangle wave function is stated in Equation (2.122) and is represented in Figure 2.35.

$$x(t) = \frac{2}{a}\left(t - a\left\lfloor \frac{t}{a} + \frac{1}{2}\right\rfloor\right)(-1)^{\left\lfloor \frac{t}{a} + \frac{1}{2}\right\rfloor} \tag{2.122}$$

in the range of $(-\infty, +\infty)$ and continuous in C^{∞}, where a is the amplitude of triangle wave.

The integral of a square wave is equivalent to a triangle wave as follows:

$$\int sgn(\sin(x))\, dx \tag{2.123}$$

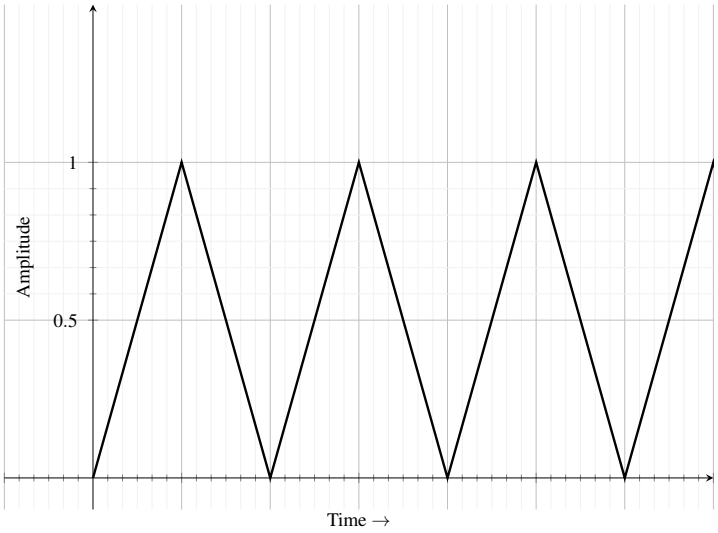


Figure 2.35 : Triangle Wave Function.

2.7.2.4 Sawtooth wave function

Sawtooth wave is quite similar to a triangle wave function as it can be seen in Figure 2.36. Bose (2007) generates a 1-5-3 neural network model that converges and fits the data properly, that is, it minimizes the errors by using sawtooth waves for power electronics and motor drives. Wang et al. (2017) use it in the cellular neural network to identify the patterns of HIV carriers by elaborating the data from images of liver tissues. Sawtooth waves can be drawn as Figure 2.36. In the Equation (2.124), the absolute value of a sawtooth wave function may result in a triangle wave as follows:

$$x(t) = 2\left(\frac{t}{a} - \left\lfloor \frac{t}{a} + \frac{1}{2} \right\rfloor\right) \tag{2.124}$$

in the range of $(-\infty, +\infty)$ and continuous in C^{∞}.

2.7.2.5 S-shaped rectified linear unit (SReLU)

As introduced first by Jin et al. (2016a), it is designed to learn both convex and non-convex functions by imitating some functions in physics. Jin et al.

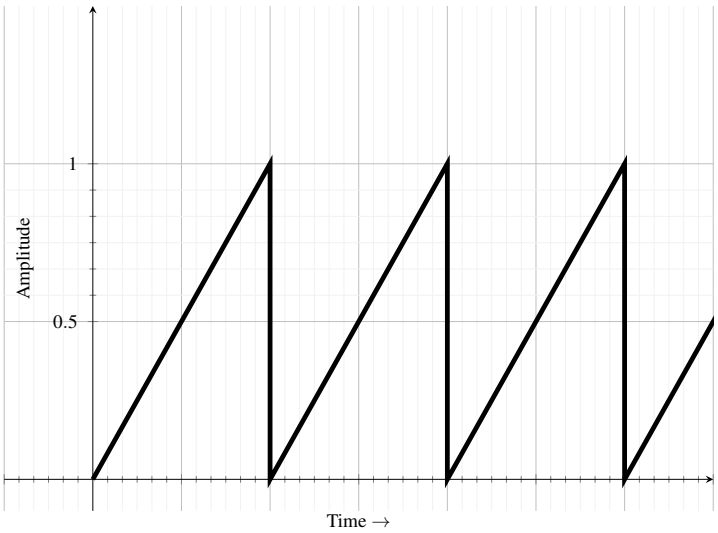


Figure 2.36 : Sawtooth Wave Function.

(2016a) states that it contains three linear functions which are reshaped with four parameters leading to two knuckles (t_l and t_r for Equation (2.125)). Since the initial parameters are unknown at the first stage, Jin et al. (2016b) and Jin et al. (2016a) propose it can be manually and exogeneously determined. But if it is set to too large or too small, SReLu may not work well. To deal with this problem, they propose:

* Initialize each o_i to be $(t_i, 0, a_i)$ in all layers where t_i is any positive real number and $a_i \in (0, 1)$, and it is freezed the update of the parameters of SReLU during the initial several training epochs.
* SReLU is degenerated into a conventional LReLU at the beginning of the training. Then upon the end of the freezing phase, it is set t_i^r to be the largest k_{th} value of each SReLU's input from all training data, i.e

$$t_i^r = supp(X_i, k);$$

where $supp(X_i, k)$ calculates the k th largest value from the set X, and X_i represents all the input values of an individual SReLU.

SReLu is explained with Equation (2.125) given below:

$$f_{t_l,a_l,t_r,a_r}(x) = \begin{cases} t_l + a_l(x - t_l) & \text{for } x \leq t_l \\ x & \text{for } t_l < x < t_r \\ t_r + a_r(x - t_r) & \text{for } x \geq t_r \end{cases} \tag{2.125}$$

in the range of $(-\infty,+\infty)$ and continuous in C^0. The parameters t_l, a_l, t_r, a_r which are learnable parameters are predetermined or endogeneously determined and can be defined in real numbers.

The first drawback, as stated previously, is that it is too hard to determine its initial parameters. Second drawback is its differentiability except its two points serving three piecewise linear functions to merge, so at these points, it is impossible to differentiate SReLu. Besides, Jin et al. (2016a) found that SReLu performs so well than other rectifiers when solving deep neural networks.

2.7.2.6 Adaptive piecewise linear unit (APLU)

While the term adaptive mainly describes the non-monotonicity of this activation function, APLU has two terms, the first one includes a ReLU, but the second term plays a role as a regularization parameter in gradient descent. Agostinelli et al. (2014) propose a new kind of activation unit that learns from neurons independently by using gradient descent, which is the most distinctive property rather than other rectifier units. Since, practically, any piecewise-linear function that must be continuous can be denoted as APLU by accepting the learning parameter a_i in Equation (2.126) as zero, Agostinelli et al. (2014) assert ApLU is a generalization of piecewise-linear function. Due to the regularization parameter, APLU can have more than one hinge, this property may turn it into a periodic function which cannot satisfy monotonicity conditions. APLU is given in Equation (2.126:

$$f(x) = \max(0,x) + \sum_{s=1}^{S} a_i^s \max(0, -x + b_i^s) \tag{2.126}$$

in the range of $(-\infty,+\infty)$.

APLU can learn from each neuron as their weights, so each iteration in a machine learning or deep learning process (the number of epochs) gradually shapes and provides a neuron-specific adaptive piecewise activation function. Agostinelli et al. (2014) assert that APLU significantly increases success of prediction in CIFAR-10 and CIFAR-100 experiments. But, since there is no consistency for an activation function for each neuron, it may lead to increased

fragility of the complete system if there exists at least one failure in a neuron. In the next step, it will trigger to produce bias for other neurons.

2.8 Bias Unit

A bias unit is an “extra” neuron added to each pre-output layer with a value of 1. Bias units are not connected to any previous layers and in this sense do not represent a true “activity”.

The bias units are characterized by the text “+1”. As it can be seen, a bias unit is just appended to the start/end of the input and each hidden layer, and is not influenced by the values in the previous layer. In other words, these neurons do not have any incoming connections.

2.9 Econometrics

There is no differentiation in between econometrics and machine learning as claimed. Any econometric application can be revised or established as a machine learning problem. The key concept here is to determine the number of layers, unfortunately, conversion from an econometric solution or approach to machine learning requires determining the number of layers as one would expect. Apart from two stage (2SLS) and three stage (3SLS) least squares, most econometric applications can be considered as a one-layered perceptron. However, if it comes to interpretation of a model and the effect of explanatory variables (features), econometrics is still a powerful tool to maintain this with its estimated coefficients. An answer from machine learning literature is given with weights which are reciprocal to coefficients and are learnable parameters to adjust themselves in order to increase estimation power.

Since there will be no use of gradient descent in econometrics, it is impossible to state a learning parameter which affects the whole process itself. Due to the non-existence of a learning process, the solution must be given by another process that named as normal equation or analytical solution methods as stated in the chapter named Analytical solution of regression. The machine learning solution results from hardness to solve the inverse of variance-covariance

matrix, $X'X\theta$, analytical solution cannot do its job when the data are relatively larger than is usual.

So, if the learning process is disqualified from the whole process of estimation, most of the econometric methods can be thought of as machine learning problems. The literature of econometrics is relatively older than machine learning literature, so some problems such as stationarity of variables, heterogeneity of panels, non-parametric and Bayesian methods and forecasting etc. may be resolved better with econometrics than with ML. In the near future, these two disciplines will converge faster than expected.

In the following subchapters, the basics of some econometrics applications used in the study are given in order to compare them with machine learning applications. The first one is the estimation procedure of discrete/binary choice models which can be estimated with logistic regression. As previously stated in theoretical explanations of machine learning literature, logistic regression is slightly different to get exact solution. Second approach is quantile regression which is a derivative of median regression to estimate the data which contains outliers, biases.

2.9.1 Logistic regression

In linear models, it is possible to get exact solution of normal equation where the dependent variable is a continuous, or more general quantitative data. Explanatory variables which are taken as unchanged or transformed into logarithm, squared or other kind of transformation show their effects on dependent variable. So, the linear models solve the following properties:

$$E(Y|X) = \beta'X \quad and \quad Var(Y|X) \propto I. \tag{2.127}$$

However, if the dependent variable is a discrete choice or qualitative data (Maddala, 1983), the effects of explanatory variables may estimate lower or higher far from outside the possible domain of dependent variable. A well-known instance for that are yes/no answers which are coded as binary in the restricted domain. The general solution of these kind of problems is generalized linear models (GLM) which requires the following property:

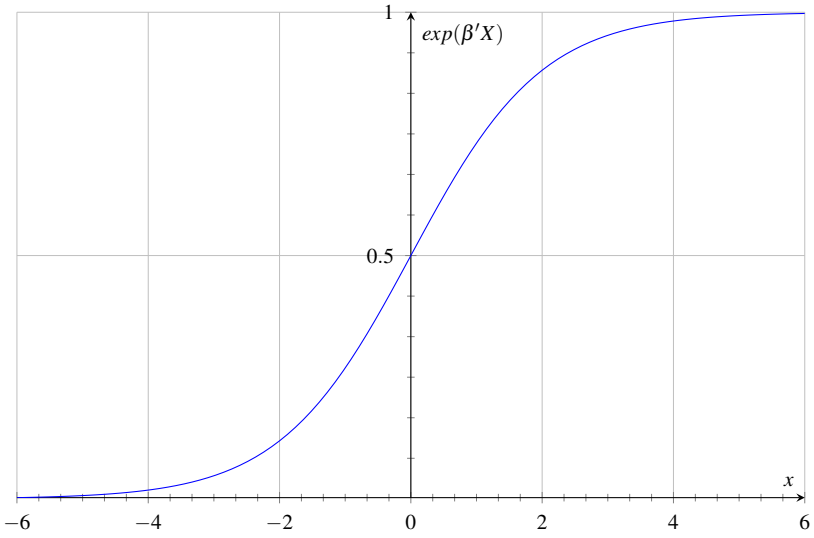


Figure 2.37 : Logistic Function.

* If Y lies in a restricted domain (e.g. $Y = 0, 1$), parameterizing $E(Y|X)$ as a linear function of X violates the domain restriction.

So, GLMs have to push the $\hat{Y}$ in restricted domain. In general, GLMs are a class of nonlinear regression models that can be used in certain cases where linear models are not appropriate to restrict estimated $\hat{Y}$ in the domain.

Logistic regression is a specific type of GLM. It is developed that logistic regression from first principals before discussing GLM's ingeneral. Logistic regression is used for binary outcome data, where $Y = 0$ or $Y = 1$, which are the domains as previously stated. It is defined by the probability mass function (Greene, 2003) as;

$$P(Y = 1|X = x) = \frac{\exp(\beta' x)}{1 + \exp(\beta' x)} = \frac{1}{1 + \exp(-\beta' x)}, \tag{2.128}$$

which implies that

$$P(Y = 0|X = x) = 1 - P(Y = 1|X = x) = \frac{1}{1 + \exp(\beta' x)}, \tag{2.129}$$

where $x_0 \equiv 1$ so β_0 is the intercept. Below, Figure 2.37 shows $P(Y = 1|X)$ and $P(Y = 0|X)$, plotted as functions of $\beta' X$:

In order to become more concrete, domain for $\hat{Y}$ can be functionalized with logit function as,

$$\text{logit}(x) = \log(x/(1 - x)), \tag{2.130}$$

maps the unit interval onto the real line Wooldridge (2010). The inverse logit function, which is also called as expit function is as follows:

$$\text{expit}(x) = \text{logit}^{-1}(x) = \frac{\exp(x)}{1+\exp(x)}, \tag{2.131}$$

where the function maps the real line onto the unit interval.

In logistic regression, the logit function is used to map the linear predictor $\beta' X$ to a probability. Therefore, the linear predictor in logistic regression is the conditional log odds which can be solved as follows:

$$\log\left[\frac{P(Y=1|X)}{P(Y=0|X)}\right] = \beta' X. \tag{2.132}$$

Thus one way to interpret a logistic regression model is that a one unit increase in X_j results in a change of β_j in the conditional log odds. Besides, a one unit increase in X_j results in a multiplicative change of $\exp(\beta_j)$ in the conditional odds.

Estimation of logit function is made with log-likelihood that requires the parameters of a statistical model using observed data according to frequentist approach. Assuming independent cases, the log-likelihood for logistic regression is

$$L(\beta|Y,X)\& = \&\log\prod_i \frac{\exp(Y_i\cdot\beta' X_i)}{1+\exp(\beta' X_i)} \tag{2.133}$$

$$= \sum_{i:Y_i=1} \beta' X_i - \sum_i \log(1+\exp(\beta' X_i)). \tag{2.134}$$

This likelihood is for the conditional distribution of Y given X. As in linear regression, we do not model the marginal distribution of X.

Logistic regression models are usually fit using maximum likelihood estimation (MLE). This means that the parametric likelihood above is maximized as a function of β. The gradient of the log-likelihood function, also called as score function, is as follows:

$$G(\beta|Y,X) = \frac{\partial}{\partial\beta}L(\beta|Y,X) = \sum_{i:Y_i=1} X_i - \sum_i \frac{\exp(\beta' X_i)}{1+\exp(\beta' X_i)} X_i. \tag{2.135}$$

The Hessian of the log-likelihood is;

$$H(\beta|Y,X) = \frac{\partial^2}{\partial\beta\beta'}L(\beta|Y,X) = -\sum_i \frac{\exp(\beta' X_i)}{(1+\exp(\beta' X_i))^2} X_i X_i'. \qquad (2.136)$$

The Hessian is strictly a negative definite as long as the design matrix has independent columns. Therefore $L(\beta|Y,X)$ is a concave function of β, so has a unique maximizer, and hence the MLE is unique.

From the general theory about the MLE, the Fisher information;

$$I(\beta) = -[EH(\beta|Y,X)|X]^{-1}, \qquad (2.137)$$

is the asymptotic sampling covariance matrix of the MLE $\hat{\beta}$. Since $H(\beta|Y,X)$ does not depend on Y, $I(\beta) = -H(\beta|Y,X)^{-1}$.

Since $\hat{\beta}$ is an MLE for a regular problem, it is consistent, asymptotically unbiased, and asymptotically normal if the model is correctly specified.

2.9.2 Quantile regression (QREG)

In a statistical regression which is called the normal equation approach, the desired estimate of $y|x$ is not always given by a conditional mean although it is the most commonly used method. While the conditional mean is useful in measuring the relationship between variables around the center of the data, it is often required to understand the relationship between variables at positions other than the center of the data. Quantile regression serves as a method which provides the capabilities to understand the relationship between variables at a given quantile.

Initially a background on quantile regression is explained. This section looks at the theory behind quantile regression, its advantages and disadvantages over more popular regression methods such as ordinary least squares (OLS) and the implementation of quantile regression under a frequentist approach, a Bayesian approach and a linear mixed effects model approach.

2.9.3 Theoretical background of quantile regression

The objective of regression analysis is to establish a relationship between a response variable, Y, and the predictor variables, $\{x_1,\ldots,x_p\}$. In real

world applications, Y cannot be calculated perfectly from the X variables. For modeling purposes it is formulated Y, for a fixed value of each x_i as a random variable. It is generally preceded by summarising the relationship of the response variable for fixed values of the predictors using measures of centrality, specifically the mean, median and mode.

Quantile regression uses the median as its central tendency and is the method of interest in this study. In this chapter the theory behind quantile regression is outlined , focusing attention on the frequentist approach to quantile regression, Bayesian quantile regression and linear quantiles for mixed models. Finally, it is examined that the computational complexity and implementation required for each of the quantile regression methods outlined above in Python programming language.

2.9.3.1 Quantiles

A quantile τ of the dependent variable Y is defined so that $100\tau\%$ of the population have values less than the τ^{th} quantile and $100(1-\tau)\%$ of the population have values greater than the τ^{th} quantile (see Figure 2.38).

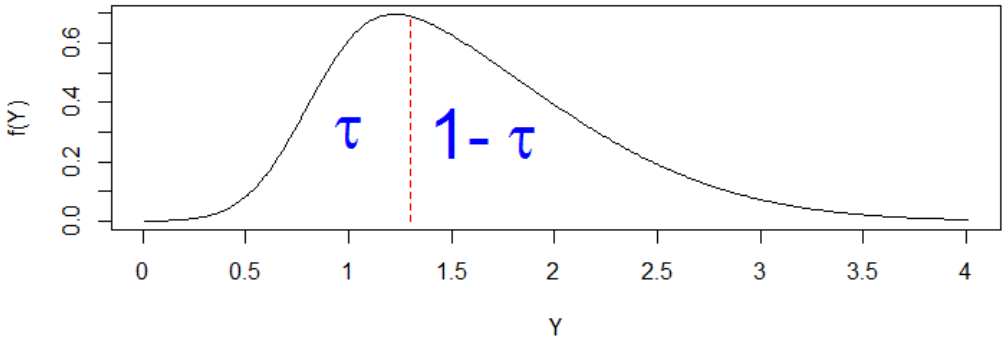


Figure 2.38 : Graphical Illustration of the τ^{th} Quantile.

The median is defined so that 50% of the population have a value above this value and 50% of the population have a value below this value. The median can therefore be interpreted as the 0.5^{th} quantile. The quantile or percentile refers to the general case of this.

More formally, let Y be a continuous real valued random variable, it may be characterised by its distribution function as

$$F_Y(y) = \mathbb{P}(Y \leq y), \tag{2.138}$$

while for any $0 < \tau < 1$,

$$Q(\tau) = \inf\{y : F_Y(y) \geq \tau\}, \tag{2.139}$$

is called the τ^{th} quantile of Y. When estimating quantiles, it is desired to determine the value of y in the sample data corresponding to a given probability τ. The τ^{th} quantile in a sample of data refers to the probability of τ for a value y, so that

$$F_Y(y_\tau) = \tau. \tag{2.140}$$

Another form of expressing the τ^{th} quantile mathematically is

$$y_\tau = F_Y^{-1}(\tau). \tag{2.141}$$

y_τ is such that it constitutes the inverse of the function $F_Y(\tau)$ for a probability τ.

If the distribution function $F_Y(y)$ is monotonically increasing, quantiles are well defined for every $\tau \in (0,1)$. However, if a distribution function $F_Y(y)$ is not strictly monotonically increasing, there are some τ's for which a unique quantile can not be defined. In the latter case one must use the smallest value that y can take for a given probability τ. In both cases the problem can be defined mathematically as seeking the value of y satisfying

$$y_\tau = F_Y^{-1}(\tau) = \inf\{y : F_Y(y) \geq \tau\}. \tag{2.142}$$

Therefore, y_τ is equal to the inverse of the function $F_Y(\tau)$ which in turn is equal to the infimum of y so that the distribution function $F_Y(y)$ is greater or equal to a given probability τ which in turn is the τ^{th} quantile.

2.9.3.2 Estimation of quantile regression

Quantile regression is a statistical technique used to estimate conditional quantile functions. Classic linear regression methods are based on minimising sum-of-squared residuals and can be used to estimate models for conditional mean functions. Quantile regression methods however offer a way of estimating models for the conditional median function and all other conditional quantile functions. As quantile regression can estimate the entire family of conditional quantile functions, it provides a much more powerful statistical analysis of relationships among random variables (Fitzenberger et al., 2013). The need for something beyond linear regression was first advocated by Koenker and Bassett Jr (1978).

What the regression curve does is give a grand summary for the averages of the distributions corresponding to the set of x's. This dissertation could go further and compute several different regression curves corresponding to the various percentage points of the distributions and thus get a more complete picture of the set. Ordinarily this is not done, and so regression often gives a rather incomplete picture. Just as the mean gives an incomplete picture of a single distribution, so the regression curve gives a correspondingly incomplete picture for a set of distributions (Koenker and Bassett Jr, 1978).

Features that characterise quantile regression and distinguish it from other regression methods are the following:

* Quantile regression can characterise the entire conditional distribution of Y through different values of τ;
* heteroscedasticity can be detected;
* median regression estimators can be more efficient then mean regression estimators if heteroscedasticity is detected;
* the minimisation problem as illustrated in equation 2.146 can be solved efficiently by linear programming methods, making estimation easy;
* quantiles are robust in regards to outliers.

The technical details in the remainder of this section will help to explain how to implement quantile regression contrasting three fitting methods. Quantile regression can be seen as one statistical method which can be used to complete the regression picture (Koenker and Hallock, 2001).

Frequentist Approach Quantile regression transforms a conditional distribution function in to a conditional quantile function by slicing it in to segments. These segments describe the cumulative distribution of a conditional variable Y given the explanatory variables x_i with the use of quantiles as defined in equation (2.142). For a dependant variable Y given the explanatory variable $X = x$ and fixed τ, $0 < \tau < 1$, the conditional quantile function is defined as the τ^{th} quantile $Q_{Y|X}(\tau|x)$ of the conditional distribution function $F_{Y|X}(y|x)$. For the estimation of the location of the conditional distribution function, the conditional median $Q_{Y|X}(0.5|x)$ can be used as an alternative to the conditional mean.

In ordinary least squares, modeling a conditional distribution function of a random sample $(y_1, \ldots, y_n)$ with a parametric function $\mu(x_i, \beta)$ where x_i represents the independent variables, β the corresponding estimates and μ the conditional mean, one addresses the minimisation problem

$$\min_{\beta \in \Re} \sum_{i=1}^{n} (y_i - \mu(x_i, \beta))^2. \tag{2.143}$$

Here, it is therefore obtained the conditional expectation function $\mathbb{E}[Y|x_i]$. The conditional expectation function is the best predictor of Y given x_i in the sense that it solves a minimum mean squared error prediction problem. It can be simply evaluated as

$$\mathbb{E}[Y|x_i] = \underset{m(x_i)}{\operatorname{argmin}} \mathbb{E}[(Y - m(x_i))^2], \tag{2.144}$$

where $m(x_i)$ is any function of x_i. This equation is minimised at $m(x_i) = \mathbb{E}[Y|x_i]$. While the approach is similar for quantile regression the central feature now becomes ρ_τ, which acts as a check function. Define $\rho_\tau(x)$ by

$$[\rho_\tau(x) = \begin{cases} \tau * x, & \text{if } x \geq 0 \\ (\tau - 1) * x, & \text{if } x < 0 \end{cases}]. \tag{2.145}$$

The check function, $\rho_\tau(x)$, ensures that:

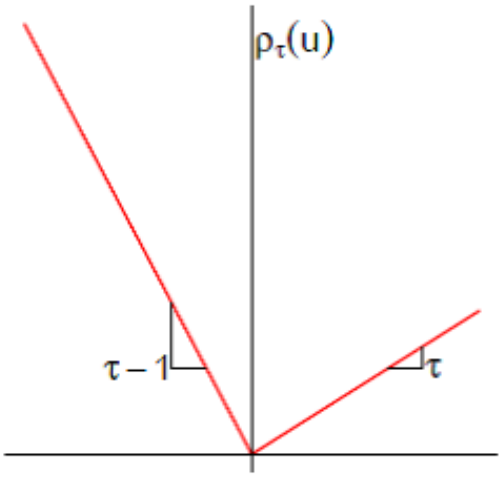


Figure 2.39 : Quantile Regression ρ Function.

* all ρ_τ are positive;
* the scale is according to the probability τ.

In quantile regression, the τ^{th} *sample* quantile may be found by solving:

$$\min_{\xi \in \Re} \sum_{i=1}^{n} \rho_\tau (y_i - \xi) \tag{2.146}$$

While it is more common to define the sample quantiles in terms of the order statistics, $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n))}$, which results in a sorted arrangement of the original sample. Their formulation as a minimisation problem has the advantage that it yields a natural generalisation of the quantiles to the regression context. The loss function for machine learning has quite a resemblance to Equation 2.146 according to Sheppard (2012).

The idea of estimating the unconditional mean is simply

$$\hat{\mu} = \underset{\mu \in \Re}{\operatorname{argmin}} \sum (y_i - \mu)^2. \tag{2.147}$$

This estimation can be extended to the estimation of the linear conditional mean function $E(Y|X = x) = x'\beta$ by solving

$$\hat{\beta} = \underset{\beta \in \Re^p}{\operatorname{argmin}} \sum (y_i - x'\beta)^2. \tag{2.148}$$

Similarly, the linear conditional quantile function, $Q_Y(\tau|X=x) = x_i'\beta(\tau)$, can be estimated by solving the minimiser

$$\hat{\beta} = \underset{\beta \in \Re^p}{\operatorname{argmin}} \sum \rho_\tau(y_i - x'\beta). \tag{2.149}$$

In contrast to ordinary least squares, the minimisation in quantile regression is done for each subset defined by ρ_τ. The τ^{th} quantile is estimated with the parametric function $\xi(x_i, \beta)$. The requirement for the square in the unconditional mean case is to ensure each calculated term is positive, the check function ρ_τ ensures this in the quantile regression case and so the square of difference is not required.

Bayesian Approach Unlike the frequentist approach to statistics, the Bayesian approach provides people with the entire posterior distribution of the parameter of interest. Additionally, it allows for uncertainty of a parameter to be taken into account when making a prediction. Irrespective of the true distribution of the data, Bayesian inference for quantile regression produces the likelihood function based on the asymmetric Laplace distribution.

A random variable ω follows the asymmetric Laplace distribution if its probability density function is given by

$$f_\tau(\omega; \mu, \sigma) = \frac{\tau(1-\tau)}{\sigma} \exp\left\{-\rho_\tau\left(\frac{\omega - \mu}{\sigma}\right)\right\}, \tag{2.150}$$

where $0 < \tau < 1$, μ is the location parameter, σ is the scale parameter and $\rho_\tau(u)$ is the loss function defined as

$$[\rho_\tau(x) = \begin{cases} \tau * x, & \text{if } x \geq 0 \\ (\tau - 1) * x, & \text{if } x < 0, \end{cases}], \tag{2.151}$$

or in a simpler form,

$$\rho_\tau(u) = \frac{|u| + (2\tau - 1)u}{2}. \tag{2.152}$$

Using general modelling techniques like ordinary least squares the estimates of the regression parameters β are computed by assuming that

* conditional on x, the random variables Y_i, are mutually independent with distributions $f(y; \mu_i)$ specified by the values of $\mu_i = E[Y_i|x_i]$;

* for some known link function g, $g(u_i) = x_i'\beta$.

However, in this dissertation it is interested in the conditional quantile, $q_\tau(y_i|x_i)$, in contrast to the conditional mean, $E[Y_i|x_i]$. Simple assumptions can be made so that regardless of the distribution of the data it is possible to solve for the quantiles in the framework of the general linear model. Assuming the following makes this possible,

* $f(y;\mu_i)$ is asymmetric Laplace;
* $g(\mu_i) = x_i'\beta(\tau) = q_\tau(y_i|x_i)$ for any $0 < \tau < 1$.

An issue with using Bayesian statistics is the requirement of a conjugate prior distribution for quantile regression formulation. While this is generally not known Markov Chain Monte Carlo (MCMC) methods can extract posterior distributions of the unknown parameters which allows the use of any prior distribution. While giving us the marginal and joint posterior distributions of all the unknown parameters, the Bayesian approach also provides us with a very practical way of including parameter uncertainty in predictive inferences. Given the observations, $y = (y_1, \ldots, y_n)$, the posterior distribution of β, $\pi(\beta|y)$ is given by

$$\pi(\beta|y) \propto L(y|\beta)p(\beta), \tag{2.153}$$

where $p(\beta)$ is the prior distribution of β and $L(y|\beta)$ is the likelihood function written as

$$L(y|\beta) = \tau^n(1-\tau)^n \exp\left\{-\sum_i \rho_\tau(y_i - x_i'\beta)\right\} \tag{2.154}$$

which is using equation 2.152 with a location parameter $\mu_i = x_i'\beta$.

The optimum strategy is to choose β so that the resulting joint posterior distribution will be proper. It can be shown that the best choice for the prior of β is for it to be improper uniform.

Linear Mixed Models Approach In statistics it is sometimes necessary to take into account the correlation of observations which belong to the same unit or cluster of the data being analyzed. Mixed effects models represent an efficient, flexible and popular way of analyzing this complex data. The

modelling technique attempts to model and estimate the variability between clusters by using cluster specific random effects. The fact that mixed models can estimate the between cluster variability is a significant advantage over standard modeling techniques as they can provide conditional inferences. In mixed models, both fixed and random effects are assumed to be location-shift effects.

The general idea of linear mixed models approach for quantile regression came from Marco Geraci and Matteo Bottai's asymmetric Laplace approach[36].

A continuous random variable $\omega \in \Re$ is said to follow an asymmetric Laplace distribution with parameters (μ, σ, τ), $\omega \sim AL(\mu, \sigma, \tau)$, if its density can be expressed as

$$p(\omega|\mu,\sigma,\tau) = \frac{\tau(1-\tau)}{\sigma} \exp\left\{-\frac{1}{\sigma}\rho_\tau(\omega-\mu)\right\} \tag{2.155}$$

where $-\infty < \mu < \infty$ is the location parameter, $\sigma > 0$ is the scale parameter, $0 < \tau < 1$ is the skew parameter and $\rho_\tau(\upsilon)$ is

$$[\rho_\tau(x) = \begin{cases} \tau * x, & \text{if } x \geq 0 \\ (\tau-1) * x, & \text{if } x < 0 \end{cases}] \tag{2.156}$$

This is the general loss function which is used in each method of quantile regression described. In here, the parameter μ is of great interest as this is the τ^{th} quantile of ω, that is that $Pr(\omega \leq \mu) = \tau$.

If the random variable ω is comprised of n independent $\omega_i's$ with common skew (τ) and scale (σ) parameters and different location (μ), then $\omega_i \sim AL(\mu_i, \sigma, \tau)$ for $i = 1, \ldots, n$. This leads to a simplified expression for $\omega's$ density function

$$p(\omega|\mu,\sigma,\tau) = \sigma_n(\tau) \exp\left\{-\frac{1}{\sigma}\rho_\tau(\omega-\mu)\right\} \tag{2.157}$$

[36] A generalisation of this model was further developed by Geraci and Bottai (2014) and is the foundation of the `lqmm R"` package which will be discussed later.

where $\sigma_n(\tau) = \frac{\tau^n(1-\tau)^n}{\sigma^n}$ and $\rho_\tau(y-\mu) = \sum_{i=1}^n \rho_\tau(\omega_i - \mu_i)$ [37].

In particular, it is assumed the following regression function,

$$Q_{y|u}(\tau|x,u) = X\beta^{(\tau)} + u, \tag{2.158}$$

where (y,X) represents the longitudinal data, u a vector of subject-specific random effects and $Q_{y|u}$ denotes the inverse of the unknown distribution $F_{y|u}$. The τ^{th} regression quantile of $y|u$ was then estimated under the convenient assumption $y|u \sim AL(X\beta^{(\tau)} + u, \sigma^{(\tau)}, \tau)$, where the τ-dependent parameters $\beta^{(\tau)}$ and $\sigma^{(\tau)}$ have a frequentist interpretation. There exists a link between the L_1 norm regression problem and the asymmetric Laplace based estimation of the coefficients $\beta^{(\tau)}$ which I will not get into here.

2.9.4 Goodness of fit measure in QREG

Quantile regression penalizes the error like in Regularization. Since then, it is impossible to find a goodness of fit measure for quantile regression likewise R^2, $PresudoR^2$ or etc. However Koenker and Machado (1999) propose a new measure described as R^1 which can be locally calculated for each quantile. R^1 is defined as;

*

$$V(\tau) = \min_\beta \sum \rho_\tau(y_i - x'\beta) \tag{2.159}$$

* Let $\hat{\beta}(\tau)$ and $\tilde{\beta}(\tau)$ be the coefficient estimates for the full model, and a restricted model, and let $\hat{V}$ and $\tilde{V}$ be the corresponding V terms, so R^1 for certain quantile is,

$$R^1(\tau) = 1 - \frac{\hat{V}}{\tilde{V}} \tag{2.160}$$

In light of the restricted model, only estimating a constant for instance, and unrestricted model, Koenker and Machado (1999) define how to compute R^1 as an R code [38] as following:

[37] (**?**) proposed a random-intercepts quantile regression model for longitudinal data using the asymmetric Laplace to model the τ^{th} conditional quantile of a continuous response variable.

[38] https://stat.ethz.ch/pipermail/r-help/2006-August/110386.html

```
rho <- function(u,tau=.5)u*(tau - (u < 0))
V <- sum(rho(f$resid, f$tau))
```

where quantiles are defined as tau and is 0.50, that is for median.

3. PREDICTION OF TRANSITION PROBABILITIES FROM UNEMPLOYMENT TO EMPLOYMENT FOR TURKEY VIA MACHINE LEARNING AND ECONOMETRICS: A COMPARATIVE STUDY

3.1 Introduction

> *If we can not change what's going on, what's the benefit if it has not arrived yet?*
> (The captive: The fugitive)
>
> Proust (1957)

Not only in strong, rich and wealthier economies, but in all economies, more or less, unemployment is a phenomenon that exists not only in theoretical textbooks and articles, but also in everyday life or society. As ILO determines, "the definition of unemployment covers people who are: out of work, want a job, have actively sought work in the previous four weeks and are available to start work within the next fortnight; or out of work and have accepted a job that they are waiting to start in the next fortnight"[1].

This definition includes only those who are actively seeking jobs, but the duration of active job search is not determined. For this reason, this active job search may be short or long. Important factors determining this period vary from the economic environment involved to the characteristic and experiential characteristics of the person.

Economic environments vary from country to country. While some countries that may be export-oriented and can provide current account surpluses can create job opportunities in line with growth rates due to rich economic environment, some countries based on labor-intensive sectors can provide less employment opportunities due not to chronically reaching high-value-added technology intensive sectors. From the perspective of labor economics, the significant point is that how an economy can deal with unemployment

[1] Accessed: 10.11.2017, http://www.ilo.org/ilostat-files/Documents/description_UR_EN.pdf

with successful policies within its economic environment. History of unemployment says most of countries have not been passed in that issue. Historically, there has been a marked increase in unemployment rates, throughout the countries. This increase in unemployment rate is quite obvious in Table 3.1:

Table 3.1 : Unemployment rates all over the World.*

Country	1950-73	1974-83	1984-93	1994-98
Belgium	3.0	8.2	8.8	9.7
Finland	1.7	4.7	6.9	14.2
France	2.0	5.7	10.0	12.1
Germany	2.5	4.1	6.2	9.0
Italy	5.5	7.2	9.3	11.9
Netherlands	2.2	7.3	7.3	5.9
Norway	1.9	2.1	4.1	4.6
Sweden	1.8	2.3	3.4	9.2
United Kingdom	2.8	7.0	9.7	8.0
Ireland	n.a.	8.8	15.6	11.2
Spain	2.9	9.1	19.4	21.8
Average	2.6	6.0	9.2	10.7
Australia	2.1	5.9	8.5	8.6
Canada	4.7	8.1	9.7	9.4
United States	4.6	7.4	6.7	5.3
Average	3.8	7.1	8.3	7.8
Japan	1.6	2.1	2.3	3.4

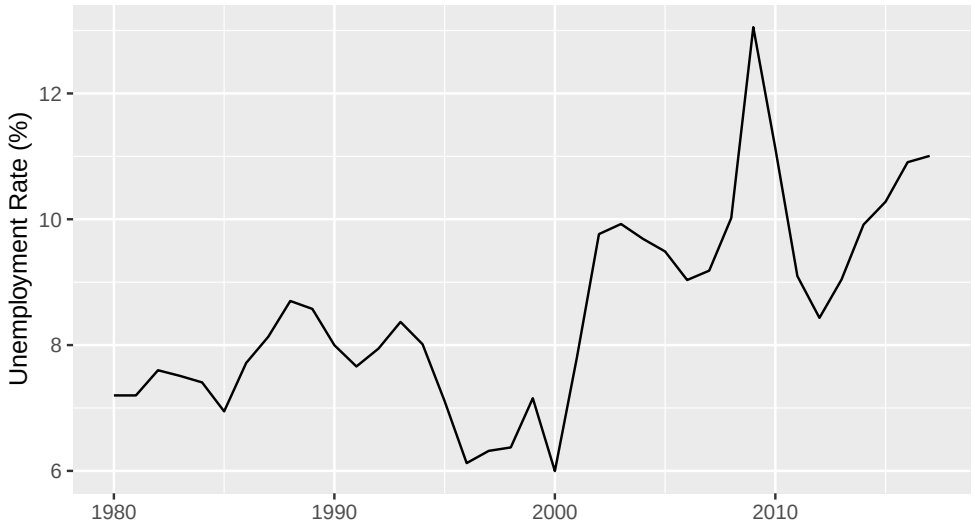
– 1950-83 from Maddison et al. (1995) and updates from OECD, Labour Force Statistics [2]accessed: 15.11.2017 [3].

There are many arguments for these fluctuations in history, but the reasons for all of them depend on several factors. Historically, despite the fact that the population has nearly doubled in all of the above countries, in all economies, except Japan, Italy and Netherlands, the unemployment ratio has tripled. When the average of each period is taken into account, the apparent increase is more pronounced. Moreover, all these countries, starting from the 1950s, have advanced to developed economies. Unemployment in developing countries presents worse scenarios than economically developed countries.

The world's 17th economy, at the moment this study was carried out [4], Turkey is one of these developing economies. Having achieved significant growth rates for the last 15 years, Turkey did not present the same success in creating

[4]https://www.worldatlas.com/articles/the-world-s-25-biggest-economies.html

employment opportunities when the trend of unemployment rates is compared with others. After 2000, having experienced two crises, one in 2001 is national, the latter in 2008-2009 was international, unemployment rates has doubled regardless of getting back prior to 2000s' level. These increases have not been reverted back as these jumps can be easily seen in Figure 3.1:



Accessed from World Development Indicators by World Bank.

Figure 3.1 : Unemployment Rates in Turkey.

The main causes of unemployment in Turkey result from the inability to achieve the desired level of industrialization, the quality remained limited due to the excess weight of the share of rural communities, inability to access the level of investment required. These reasons can quickly increase the unemployment of young people and carry structural problems. One of the major differences between developed countries and Turkey; while the level of education is high in the unemployed, the level of education in the underemployed is low in Turkey. Another major difference, in contrast to developed countries, is the unemployment rate in Turkey is higher among educated people; the cause of this is that the jobs are far from being satisfactory and unsuitable for individuals looking for work for the first time.

The employment participation rate in the OECD is around 70 percent. In Turkey, due mainly to the low participation of the female population, it is much lower; but it has begun to increase in recent years and has risen

from 57% to 58% between 2016 and 2017. This one point increase in the participation rate resulted in an increase of 1,047,000 people in the labor supply in 2017: an increase rate of more than 3.5%. However, the number of those who find jobs in this total is 926,000: an increase of less than 3.5 percent. The small gap for the whole of 2017 means an increase in the average of unemployed people by 3.7%. Interestingly, the share of the unemployed in total labor supply also rose within one year: 11.1% to 11.2%.

In order to deal with unemployment, the private sector in employment, which was initiated in 2017, became the forefront of employing the public sector to support the social security premium that should be paid to the employer. In addition to this, the tax that will be deducted from the salary of the employee will be paid by the state. According to the explanation given at the beginning, the employer who wants to benefit from the support of "one sibling" will pay the first month's tax, premium and salary of the new worker, according to the new incentives to be introduced in the coming period. Every two months from the second month, the expenses will be covered by the Unemployment Fund. Measures put into effect from the beginning of 2017 seem to be to provide temporary fixes by transferring private sector funds from the public sector rather than solving the problem permanently. The payment of the bill to the unemployment fund, which the public sector will undertake, brings many new debates.

The state of the economy is very important in the course of an individual's employment or transition to it. As mentioned before, the other most important factors are all about personal abilities. Moreover, in economies where unemployment is high, individual developments gain more importance because of the increased competition. In this sense, individuals attach importance to their personal development, receive more training, establish and expand links with their business networks, and acquire new skills. Turkey's labor market which is competitive in this perspective is a market where individual talents come to the fore. In that sense, numerous universities have opened to increase human capital, the number of individual vocational

courses supported and subsidized by the government has been increased and the quality of pre-university education has been raised.

This study, at this point, examines individual skills, ingenuities and characteristics for the transition of individuals into employment. However, here it is necessary for individuals to be unemployed in accordance with the above ILO definition in the previous period, to move employment in the next period or to keep their situation. This is called as transition which can be calculated with probability. Since the calculation of this transitivity is important for policy makers or institutions, previous studies are not enough because both the estimation and prediction power are relatively low and the datasets are limited. As the second decade of the millennium brings renovated, strengthened and enriched tools to humankind in many areas, the most powerful one is data science. This study uses new estimation methods to increase predictability and forecastability by using machine learning techniques which use all the blessings of data science. Later, these results are compared to econometric ones.

In the following sections, first, a review of literature that covers job finding in view of the three frameworks in labor market, then some key figures are shown for Turkey. Later, the methods used in this study are presented with their data as well, and the results follow. Finally, the conclusion will summarize the whole study and provide some basic insight to the makers. Limitations and further studies conclude this study.

3.2 Previous Researches

As an economic agent, individuals must acquire traits such as education, skills, judgment, development in intelligence, training, capabilities, experience and specialties to monetize them so that they can participate in the process of production to earn their livehood. The stock of these traits is called as human capital which can be innate or acquired (Acemoglu and Autor, 2011; Bowles et al., 2001). As first proposed by Schultz (1961), and established as a key concept in labor economics by Becker (1964), the revolutionary developments originate in the University of Chicago. Becker

(1964) first proposes this concept to emphasize it as a separator to distinguish one individual from another by stating the core characteristics as labor. This separator is established on a kind of capital that develops, grows with investment in schooling, medical care, training and improvement in appreciation of literature, which produces mankind itself. Immediately afterwards, Mincer (1974) offers a broad perspective on the human capital theory, in which the fragments leading to human capital accumulation were quantified. He established his paper on education in general and its effects on a human's life. Since there is no significant investment other than education, it affects earnings per unit time worked, determines labor force participation, shapes future length of time worked or the duration of unemployment.

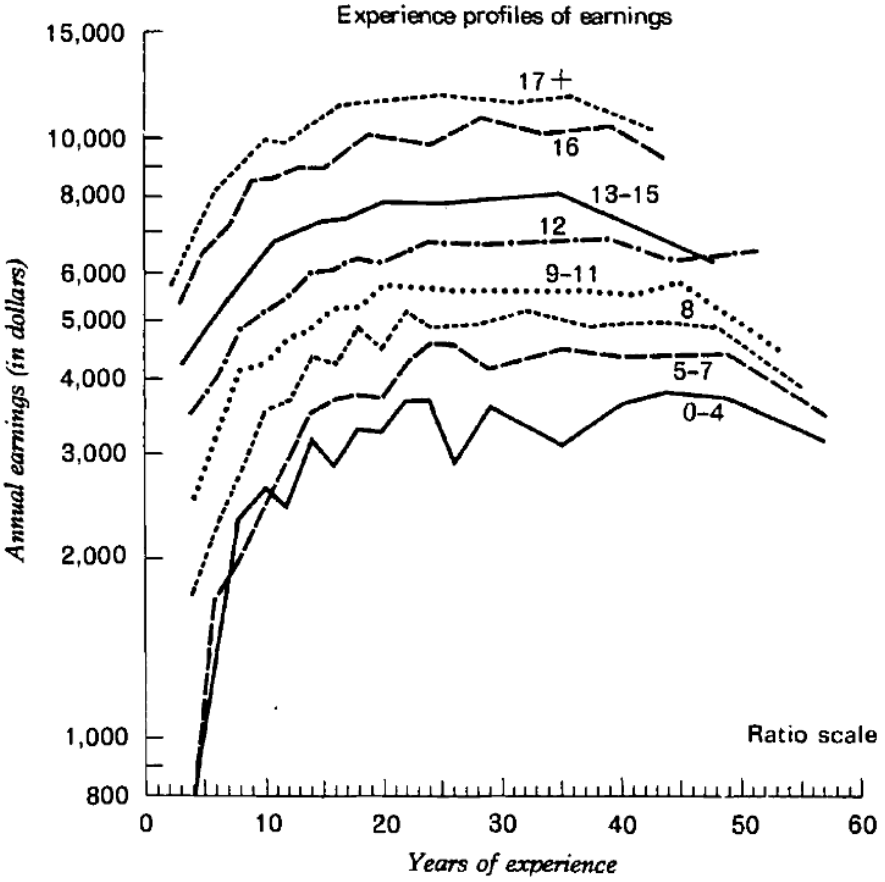


Figure 3.2 : Quadratic Experience and Earnings.

Mincer (1958) and Mincer (1974) discover an asymmetric relationship in between weekly (or hourly) earnings and experience (and also education) for non-farm labor, he offers a logarithmic parabola and a Gompertz curve for non-farm labor in the US, where the formula actually dates back to his

unpublished Ph.D. thesis done in 1957 as in Figure 3.2. Afterwards, in labor economics, the formula becomes a breakthrough for studying dynamics in labor earnings, it is called as Mincer Equation(s) (Equation 3.1 and 3.2 by citemincer1974schooling) established on experience, its square and years of schooling of an individual. Two forms of Mincer Equations are given below:

$$lnE_{s,t} = lnE_0 + r_s s + r_p k_0 t + \frac{r_p k_0}{2T} t^2 \tag{3.1}$$

$$lnE_{s,t} = lnE_0 + r_s s + \frac{r_p k_0}{\beta}(1 - e^{\beta t}) \tag{3.2}$$

in which, $lnE_{s,t}$ is gross annual earning of labor with s years of schooling. t is the schooling experience r_p and r_s denote for return of post school and schooling, k_0 is the ratio of income and investment start of working experience, β is negative coefficient of labor's ratio of income and investment made annually.

Later, a book on human capital making significant contributions to the definitions after Becker was written by Layard and Schultz (1971) on the basis of Schultz (1961) that in the book he proposed the term of "human capital" first. Columbus of human capital, some economists owe it to Shultz's works, due to his clear explanation of earnings differentiation in skilled people who have invested in themselves by taking general training earlier than unskilled people. He also invented the effect of elasticity of substitution of college graduation and others, the higher elasticity explicitly indicates higher rate of return for individuals. Schultz (1961) proposed some breakthrough ideas by asserting the role of discrimination, education inequality depending on geographies, their burdening role on economic growth and finally putting forth the significant effect of internal (and external) migration on healthy economic growth. Migration is a way to convey the knowledge and the labor in farm to other areas as supported by Weisbrod (1966).

In the job market, all workers compete with each other. In a complete information and perfect competitive labor market, labor earns only marginal product according to classical theory. Again, a rational worker can invest as much as the cost of investment in his/her human capital (Becker, 1964)

when the marginal investment of his/her human capital is positive. While it is positive, individual invests. Investment in human capital may be split into two main categories as offered: general and specific (Becker, 1964). General human capital arises if there is no firm-specific differentiation among workers and firms are not responsible for and do not tend to pay it to worker (Acemoglu and Shimer, 1999). Specific training is under the control of firms in order to increase workers' productivity while it helps to boost economic growth in a country by reducing quit rates and turnovers, increasing labor efficiency, (Donaldson and Eaton, 1976; Higashi, 2002; Parsons, 1972). In the nature of specific investment, it is not certain that it accumulates during employment (Bougheas and Georgellis, 2004) leading to an increase in productivity and uncompetitive specializations. Wasmer (2006) and Cingano (2003) conclude specific investment in human capital and propose that firm-specific skills provide higher returns rather than others.

Empirical results show that the return on education is always positive, but researchers do not reach a common result on the subject of its amount. Since Equation 3.1 and 3.2 are established on a logarithmic scale, the coefficients of return on education reflect the amount of increase on annual or hourly wages at a percentage level. Dearden (1998), Ashenfelter et al. (1999) and Psacharopoulos (1994) find the effect of education around 5%, Mincer (1974), Krueger and Ashenfelter (1992) and Becker (1964) reach the effect of schooling on earnings around 11% level. In macroeconomics,Acemoglu (2002) conducts a study in less-developed regions and finds evidence that return on education has a multiplier effect on less developed countries. Blundell et al. (1999), Mincer and Ofek (1982) draw attention to the fact that gender differentiation on rate of return of education is in favor of male workers.

Human capital theory also explains job finding under uncertainty. Education, experience, gender, training may have a significant effect on job search and finding on the side of labor supply, and then it affects the behaviour of firms when they are hiring, on the side of labor demand. At that point, there are some criticisms in labor economics literature of human capital. The first

criticism is of splitting the effects of education into two parts: the human capital and signaling. Spence (1973) distinguishes the factor of signaling when estimating hiring and that, in incomplete information and mismatch in contracts, some factors also help to produce market imperfections leading to differentiation in wage structure. Under uncertainty, hiring is a lottery according to Spence (1973). In job market, firms or employers cannot estimate the effectiveness of labor, inner abilities and other factors which are not observable by looking the education, experience and so on. In addition, this information will not become completely clear after hiring until labor has been observed for a while. Spence (1973, 1978) offers classification of the factors based on whether their types are alterable or unalterable. The characteristics of a labor which are gender, age, race, birth date are not alterable, which can be regarded as indices while some other characteristics that may be education, school, training, experience can be alterable factors which can be changed by investing and bearing the cost of it in her/his life of labor, the factors are called as signals. Spence (1978) indicates that the job finding probability is shaped as a conditional probability problem given the information about indices and signals. Cohn et al. (1987), Groot and Oosterbeek (1994) and Kroch and Sjoblom (1994) show that human capital theory generally explains the earnings, job finding probability while Acemoglu and Autor (2011) support signaling theory when unobservable factors mostly surpass the observable ones. Weiss (1995, 1971) defends implicitly signaling theory when the abilities, learning capabilities play a significant role in increasing the productivity of a labor where human capital theory cannot explain and quantify themselves in order to add as explanatory.

Second main framework that can be used for analyzing job finding probability is the search theory first defined and revolutionary examined by Stigler (1961) at the time of the human capital theory which was born in the Chicago School. Everything in life can be seen as a source of information that should be taken in. A decentralized economy may not be established in a country, where all the prices are known by sellers and buyers. But in a centralized economy, which is a fact in current economic environment, it is impossible to know

at a certain time the price of something (Stigler, 1961) at every point where those goods are being sold. This ignorance can be estimated by the dispersion of prices, the variance of prices can be the candidate for these, statistically. Stigler (1961) says the reduction of this type of ignorance is costly, decrease of the cost passes through to increase the information which is also costly to get, collect, store and share. As Reynolds (1951) mentioned about the information emerged in wage and non-wage terms of employment cannot be known by others who are not in the firm or firm-related areas, the information of job opportunities and the potentiality of labor to fit vacancies are relatively hidden type of informations. Stigler (1962) strongly emphasizing that complete knowledge is seldom possessed and dispersion in wages is a pure fact in labor economics, he states searching for a job is costly and varies according to characteristics of occupation. Axell (1977) also indicates dispersion of wages may result from zero search cost that the time a huge number of unemployed labor seeks a job at the same time [5]. Besides, search behavior strategy should be non-sequential in order to minimize duration of unemployment and to decrease search cost. Search cost does not only exist for labor, but also it exists for employers who have to increase wage levels in order to decrease exit or turnover rates and to increase job attractiveness for potential employees.

Mortensen (1970) establishes a search model under imperfect information existing for supply side of labor. The skills and wage are unknown parameters at a certain time of job search activity. A certain period of time is required for only one job search. During job search, unemployed labor bases his/her decision on the acceptable wage level which is a function of skill level that can be estimated with the number of job vacancies offered (offers are sequentially distributed) to unemployed labor is on the equilibrium of unemployment level which is derived by Phillips curve and can be explained also by unemployment compensation and the discount rate determining present value of future income. The equilibrium of unemployment is formulated as the ratio of labor force which enters employment per period divided by the sum of this

[5]The existence of wage dispersion is proved by Burdett and Judd (1983) by assuming the number of offers are random rather than sequential or one offer at a time, later than human capital and search theory.

proportion and the proportion of unemployed who find employment during a certain period for the demand side of labor market. For supply side, wage inflation depending on expected price level and the parameters of real wage and employment level determines Phillips curve, a simultaneous equation model. At that time, a response comes from McCall (1970) who offers perfect knowledge about the wage and skills (that are distributed statistically), by the way, the search theory is established on only job seekers, supply side of labor market (Stokey, 1989). The cost of search depends on transportation costs, the opportunity costs of the alternatives forgone and physical efforts spent on job search. McCall (1970) also describes the reservation wage as the minimum wage the job seeker accepts. The optimal strategy is to accept any wage that is higher than reservation wage and is also to reject wage lower than reservation wage if labor is neutral or averse to risk. In addition to these, McCall (1970) proves this strategy dominates Stigler's non-sequential search strategy even if the first offered wage is higher than expected wage.

At the time of human capital theory and search theory, some criticisms were made by Rothschild and Diamond. Rothschild (1978) criticizes the articles stated above because of considering specifically one-side of labor market, labor demand from firm or labor supply from individual workers. Optimization of any strategy should consider both sides of market actors, due to this, partial-partial equilibrium cannot be an optimal solution of labor market. On the other side, second criticism comes from Diamond (1971) proposing the Diamond paradox which is a renowned theory in economic history. Emphasizing the wage-setting power of firms, this paradox implies that the cost of search is relatively smaller than expected and it leads to a raise in prices of goods from competitive equilibrium to monopsony level. If all unemployed workers optimize their search strategy, firms in market offer the same wages to them which generates monopsony wage. These two consequences of search strategy add more power to firms in the market.

Lucas and Prescott (1974) look at the search theory from a different perspective, macroeconomics. Previously, Thornton and Phelps (1972) make an analogy between unemployment and search behaviour in a fictionary

island where labor seeks a way to get higher wage income. In order to get this, they decide to migrate to another area which allows competitive wage levels. The migration, however, is not free, the labor should bear some migration expenditures or information costs which is the cost of getting new information on the wage income level of another island. In the light of Phelps' analogy, Lucas and Prescott (1974) characterize labor flows in view of "rational expectations" indicating future events have a probability distribution of itself and the agent expects its mean value of this distribution even for future events. Search process can be recovered after a successful transition to a job, hence, market will be cleared. They also examine search process within the framework of general equilibrium. Mortensen (1970) also provides a reply to island example of Thornton and Phelps (1972) by changing the strategy of each firm is uncertain and is based on attracting skilled workers to their jobs in order to increase their efficiency and production possibility frontiers, then his model provides similar results to Lucas and Prescott (1974).

In the frameworks of Diamond (1981, 1982b) and Mortensen (1982), the search theory highlights other factors that allow the emergence of matching theory in the late 1980s, in the presence of increasing returns to scale in technology, search and recruitment models provide multiple equilibria due to several reasons. But, the most important reason is regarded as trade-frictions that result from non-unique unemployment rates. This separation from the search theory, that the reason for unemployment is accepted as relatively lower wage than either expected wage or reservation wage [6], which forgoes other macroeconomic factors lead to emerge matching theory. In very basic terms, matching theory solves the unemployment problem by looking at mismatching of firms and labor (Diamond, 1982a; Mortensen, 1982). Trade frictions and its application on unemployment made a significant breakthrough in search theory on the basis of Diamond's and Mortensen's studies and they were boosted by sequential researches written by Pissarides (1986, 1985, 1987, 1988) by means of externalities emerging with trade frictions. Pissarides (1988) clarifies the differences between reservation-wage

[6]Later, the literature is splitted into two parts, and the first part of that related with minimum wage will be classified as reservation-wage approach in search theory.

approach (which is stated in previous parts) and trade frictions approach, later, he focuses on the factors emerging from trade frictions and concludes that there aretwo externalities, one is making a contract in a short time with a small probability, the other is the dependency of job contracts which is a function of number of traders in the market. Additionally, since cycles are in the nature of unemployment (Pissarides, 1987) in order to show temporary shocks in the economy, as examined in the UK case (Pissarides, 1986), trade frictions explain that the fluctuation in unemployment flows arose from stock of unemployment itself. Mortensen and Pissarides (1994) and Pissarides (2000), finally, propose a theory of unemployment by giving a clear and definitive description of labor market conditions and how it works, by allowing policy implementation tools to adapt labor market functionality, by providing intuitive comparative statics which are rich in terms of macroeconomic issues.

Meanwhile, the literature in reservation wage approach grows in macroeconomics with the works of Ljungqvist and Sargent (1995) examining the relationship between wage distributions and law of motion of unemployment and employment which are aggregated in the context of dynamic general equilibrium. Certain policies such as threshold in wage, unemployment insurance, progressive payroll tax and the duration effect of job refusals, are examined in order to estimate the optimal reservation-wage which is a function instead of generating a quantitatively determined one. The result is that labor market policies have an effect on reservation wage along with substituting search or work and also keep working in a job or separation from it. A Swedish case is analyzed in Ljungqvist and Sargent (1995) to check whether the model works, in conclusion, the model fully explains the unemployment period starting in the middle 1960s. After that, based on Kydland and Prescott (1982) dynamic stochastic general equilibrium model, Cho (1996) creates a multi-sector model that assumes one unproductive and two productive sectors and labor works in one and has mobility but should spend of their time in the unproductive sector. Cost and time spent looking for a job is needed for analyzing transition from the unproductive sector to one of the productive sector. This picture is drawn from the durable

goods and consumption goods sectors of the US. Economic cycles affect unproductive consumption sectors heavily while it also damages the balance of search behaviour and reemployment allocation. Another search-theoretic model on the basis model of Lucas and Prescott (1974) comes from Gomes et al. (2001) that explain the dynamic relationship of unemployment and its duration from the perspective of the search theory again but under the wing of reservation-wage theorists. Their most significant contribution is that the search behaviour of wealthy households can reach higher-paid jobs rather than lower-paid ones.

As previously stated, the matching theory is discovered in labor economics in the late 1980s, frankly, it dates back to Gale and Shapley (1962) examining the matching of college students after admission and the matching of partners, interestingly, without the game theory. However, the second problem is set on the acceptance quota and is called a classical matching problem while the first one has no restriction, in addition, in the second problem, the agents being matched are dynamic, therefore it is hard to set the problem clearly in order to model it. In a classical matching problem, there are two sets [7] and the goal is a stable matching solution without leaving any single individuals. Hence, there must be some solution which is stable and which successfully pairs them. Gale and Shapley (1962) clearly address a stable matching problem and then offers a revolutionary procedure called deferred acceptance algorithm and named as Gale-Shapley Algorithm given in Figure 3.3 as shown. (Abdulkadiroglu and Sönmez, 2013) state that, even todays, the Gale-Shapley (G-S) algorithm and its derivatives are widely used to match hospitals and students in schools. the G-S algorithm is set up in two different ways including recommending men to women first and vice versa. Women determine their best choice, each man considers the proposals whether they exist or not. Then, each man can retain what he regards as the most attractive proposal by putting it to one side and must refuse the rest of proposals. The women given a rejection at the first round are offered to their second-best man. Each man can accept the best offer again and must refuse the rest of them. This process is repeated several times until there are no women who can make any further offers. Gale and Shapley

[7] A set of women and a set of men.

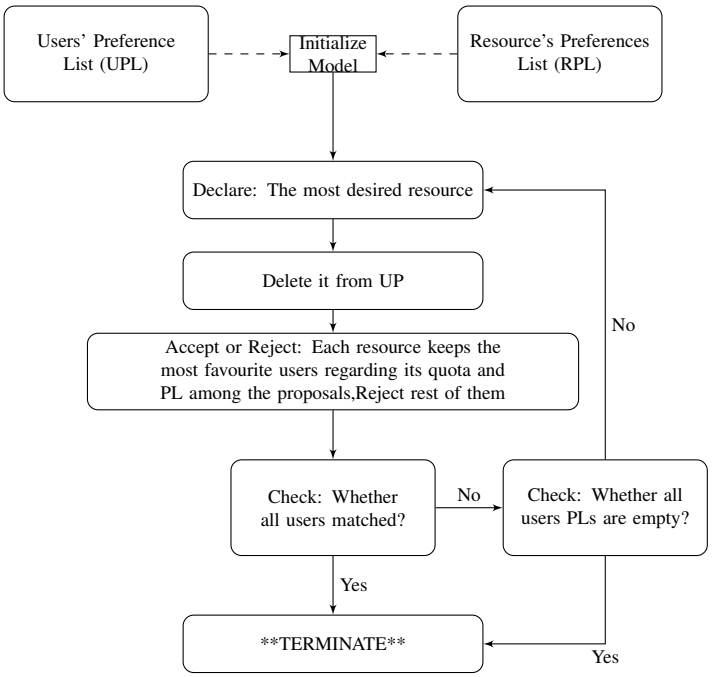


Figure 3.3 : Solution Diagram of Gale-Shapley Algorithm.

(1962) gave mathematical proof that this algorithm can reach a stable solution for every iteration. After a while, Shapley and Scarf (1974) propose a new matching model enhancing it to n-person economic game deriving "coalition optimal" outcomes by using, at that time, a game-theoretic approach to solve the model for market trading. They generated Top Trading Cycles Algorithm that gives an exact solution to find "coalition optimal". Later, Top Trading Cycles Algorithm is used by hospitals around the world to assign kidneys to the most proper candidate to save lives. At the end of the 1970s, another solution for one-sided matching comes from Hylland and Zeckhauser (1979), however, the most crucial point of the solution derived from the model is the questioning of Pareto Optimality of the solution in ex-post and ex-ante situations. They firstly offer the utilities of being matched to a position which satisfies the Pareto Optimal allocation of jobs for individuals. Besides, one-sided matching solutions offered by Hylland and Zeckhauser (1979) are regarded in the matching literature as house-allocation problems.

The first attempt to apply the matching theory in economics is made by Roth (1984). National Intern Matching Program (NIMP) tackles an issue to match residents and hospitals, finally, a stable solution is given by the NIMP Algorithm in the 1960s, which is a resemblance of the G-S algorithm. NIMP

has evolved twice and restructed itself as the National Resident Matching Program (NRMP) which aims to match residents, interns and experts to hospitals. However, Roth (1984) claims the solutions are not stable even if some conditions for propositions are a bit relaxed. Since then, Roth (1984) offered a new algorithm generating a stable solution to sides of agents being matched. The task consists of 20K jobs that will be filled by physicians and interns who experienced at least one interview during the recruitment process. According to the interviews, all applicants propose a Rank Order List (ROLs) which must be submitted to NRMP through one clearinghouse. ROLs are prepared in accordance with the preferences of candidates over available positions and NRMP will run its own algorithm to place all of candidates that have been matched to their preferences in the residency programs. If there exist many differentiated markets that require reconstruction of NRMP algorithm, unstable solutions or market failure can result without the customization of the algorithm used (Roth and Xing, 1994). Algorithmic customization due to the diverse nature of markets actually depends on the proposition for the unstable solutions offered by Roth and Sotomayor (1992) which claims that stated (true) preferences over choices creates dominant strategy for multi-optimal stable matching solution and which cover two-sided matching market problems comprehensively. Abdulkadiroğlu and Sönmez (1998) review the literature of mechanisms where the first one is called a simple serial dictatorship that requires top choices at first declaration of preferences and second is called as random serial dictatorship which allows non-discrimination over preferences ignoring whether it is top or not, and offers an alternative mechanism design for lotteries to solve the house allocation problem by matching the sides efficiently.

Valuable and unique contributions of the creators of the human capital theory (Schultz, Becker), the search theory (Diamond, Mortensen and Pissarides) and the matching theory (Roth) were rewarded by Nobel prizes. The criticisms could not wait any longer and were impelled to the dynamics of the labor market which can fluctuate over time due to microeconomic or macroeconomic reasons. Shimer (2005) demonstrates a new version of the

search-matching model considering observable undulation in unemployment and economic situations against classical assumptions of the matching theory. Shimer (2005) includes two shocks to the classical search-matching model (Mortensen and Pissarides, 1994; Pissarides, 2000), the first one is labor productivity shocks by assuming that the job loss rate is constant, the second is job separation shocks by assuming labor productivity is constant, since volatility of job vacancies to unemployment ratio is significantly higher than volatility in labor productivity. At the same time Hall (2005) changes the view of unemployment by finding empirical proof to look at recessions and proposes a new key variable called job-finding rate which affects fluctuation in unemployment along with separation rate. Hall et al. (1995) previously explained the transition in between employment and unemployment in the light of recession dynamics and concludes that cyclic fluctuation in an economy may lead job destruction which may lead perdurable unemployment in labor markets. So, the probability of finding a job is expected to be low during recession, but the problem might be turned into how it can be measured with. Shimer (2008) reexamines it with a new model on the basis provided before by Kaitz (1970) explaining it with duration of unemployment and with the concept of business cycles he also considers. He proposes Eq.3.3 given below to measure job finding probability depending on unemployment duration:

$$F_{\tau} = \frac{\Sigma_t n^e_{\tau,t}}{\Sigma_t (n^e_{\tau,t} + n^u_{\tau,t} + n^i_{\tau,t})}, \tag{3.3}$$

In the equation above, $n^e_{\tau,t}$ denotes the scaled number of people in duration τ unemployed, in month t who will be in employment at $t+1$, $n^u_{\tau,t}$ denotes total number of unemployed people, $n^i_{\tau,t}$ is used for the inactive population who are not counted in labor force. Later, Shimer (2008) proves his duration dependent job finding probability and its model by giving very basic descriptive statistics from the HLFS microdata covering 30 years. Shimer (2012) also proposes a new methodology for 1948-2010 US data for job finding and losing (or occupation destruction) probabilities by showing the first one is cyclical but the latter does not show cyclical behavior in parallel

with economic fluctuations. In addition to these, Narendranathan and Stewart (1993) question the probability of leaving unemployed while an individual is unemployed likewise whether examining the duration effect is important for an individual by means of the proportional hazard model and conclude that it is so significant if the unemployed individual can extend it for twelve weeks [8]. Ponomareva and Sheen (2013) looked the transition probabilities in four types of states in the labor market as full-time employment, part-time employment, unemployment and inactive, and they found that job finding and losing probabilities behave cyclically and at that time, for the Australian labor market, the latter is so significant that the result is a bit contradictory to Shimer (2012).

Duration dependence is a well-known factor for job finding probability as given by Hall (1979); Pissarides (1979) due to certain factors such as depreciation in skills and self-confidence, losing networks, the cost of searching for a new job and so on. Besides, there are other factors which may be crucial to finding a job. Blau and Robins (1990) offer at first characteristic factors of individuals who search for a job such as marital status, gender, teenager, being black or Hispanic (according to the US, these are significant), receiving unemployment insurance, living in small area (that can affect the number of job vacancies in that area). The model used also covers human capital variables, work experience and education. Since the dependent variable used is different, the number of ways to search for a job, it actually does not estimate the transition from unemployment to employment or reemployment, however, it gives significant clues to the labor market: The job is searched on the job. Per contract offer is greater in employment than unemployment. Holm et al. (1999) discuss job finding probability in the border of factors determined at the household. They not only find characteristics of individuals affect the transition from unemployment to employment, they also give certain evidence that if the disposable income which is higher at the household level and debt that also is higher at the household level may boost this transition positively. Osberg (1993) asserts that the medium of job finding matters, especially business

[8] Van Ours and Vodopivec (2004) find the term as three months (approximately 13 weeks).

cycles change the medium of finding a job, hence, job finding probability changes in accordance with that, as proven in the Canadian labor market. Interestingly, public employment agencies outperform at the time of business cycles and later period, therefore, they play an important role in increasing public social benefits unless there is relatively low level of unemployment. Calvó-Armengol (2004) and Wahba and Zenou (2005) show that, due to the nature of networks that increase public social benefit in terms of decreasing unemployment rates at both country and local levels, social network ties may contribute to increased job finding probability. Social networks have impressively changed after 2005 and have migrated to digital platforms. At that point, Fountain (2005) uses a 1998-2000 data to examine these effects and finds that job finding probability is affected positively only for the short term. In the long term, they have damaging effects on job finding, due to their cost. Later, since the matching and screening costs are lowered during the artificial intelligence age, the effect has been in question and the results are not persuasive.

3.3 Turkish Studies

Before 1966, there are no official records about the structure of the Turkish labor force. In that year, few and limited information about labor was collected while conducting the General Census by only asking demographic information and individuals' labor market statuses. The first serious attempt to collect data about the labor force is around 1988. Bulutay (1992) estimates the labor dynamics in Turkey for the period starting from establishment of the Turkish Republic (1923) to 1989, the ratio of people who searching for a job is relatively high in the period 1988-1992, it is 42% among unemployed people who are approximately between 7.4% and 8.9% of the population. Since most job opportunities were created by the government, Bulutay (1992) states that job finding probability amongst educated people is higher than uneducated or less educated people. Şenses (1996) shows that the labor market in Turkey structural adjustment in the transition from a closed economy is far too protective import-substitutionism to this version; therefore, the transition from

unemployment to employment may be reduced due to the lack of educated individuals in the field of innovation, science and informatics.

Tansel and Tasci (2004) use 2001-2001 HLFS on the Turkey data to estimate unemployment transition probabilities, which may be the first application querying labor market transitions in Turkey. Proportional risk models, log-logistic and log-normal models were used in the study. The results indicate that urban or married individuals are more likely to be unemployed than other groups, but women are less likely to be unemployed than men. At the educational level, the probability of unemployment of college or undergraduate graduates is higher than other education groups, and the probability of becoming unemployed decreases with age. After that, Tasci and Tansel (2005) estimated long-term unemployment durations with two-stage probit models using TURKSTAT's 2000-2001 Household Labor Force Surveys. Living in cities, for both types of unemployment, is reducing unemployment while women are increasing the possibility of long-term unemployment. Geographically, long-term unemployment decreases as we move from west to east. At the educational level, the probability of long-term unemployment decreases as the number of primary school graduates increases, but it still has positive effect, while the effect of long-term unemployment on graduates is negative. The appearance of gender discrimination in the labor market in Turkey, due to undergo more difficult than expected employment of women transition from unemployment is examined and this employment are high influence under finding a job in their field (İlkkaracan, 2012). Tunali and Ercan (2003) also indicate that although vocational school education is expected to help the graduates ease their way into the labor market, statistics reveal that a significant majority continues with higher education. Tasci and Ozdemir (2006) finds that the transition from long-term unemployment to employment may be damaged by living in rural areas, being female, older and have lower education.

3.4 Data

The data are collected from Household Labor Force Surveys (HLFS) conducted in between 2000 and 2016 by the Turkish Statistical Institute

(TURKSTAT). In total, 7,323,449 individuals were surveyed by means of a set of questionnaires consisting of extensive information that were classified under seven headings such as household characteristics, individual's characteristics, employment status, income, unemployment and inactivity, past work experience, labor status one year before the survey. Total number of variables are 110 with three of them derived from other ones. The data as a csv file costs 3.18 GB to an Ext4 [9] file-system, which is an example of a big-data that it is hard to deal with classical computational methods. Rather, the data is compressed under Ext4 system by using "gzip" compression with a level of 6 by means of R that saves the data as an ".rdata' in order to process it with ease. The size of the big data after compression is reduced by up to 15%.

However, since this research is restricted only with both previously unemployed individuals and surveys done in 2003-2000 are not compatible to catch individuals who are previously unemployed and who are employed or unemployed at the survey date, the data merged is filtered against them [10] to focus only on those who are interested in this study. So, given that information, the data are downsized to 203,891 individuals which will be split as train set (188,208 individuals) and test set (15,683).

Due to the inconsistency of the surveys, a map for variables is generated to match them. However, even if the same question is asked to surveyee in the next year, the answers may be changed by TURKSTAT. So, the matching process for questions is not enough, the answers must be compared in terms of both content encodings and means. Variables are categorized into four groups: characteristics, employment, unemployment and past information. Later, some of them are processed to derive certain variables. For instance, it is asked when the individual had a job in his/her past [11]. The answer may not give a clear opinion, so the difference in between survey year and the answer is used to get the period of inactivity[12].

[9] Fourth extended file-system, generally used by Linux OSs.

[10] Individuals who are not in labor force, institutional population and are not actively searching for a job with the definition of ILO are dropped.

[11] ISAYRIL_YIL

[12] INACTIVE=YIL - ISAYRIL_YIL

The tabulation matrix, Table 3.2, is given below by year:

Table 3.2 : Tabulation of HLFS (2004-2016).

Year	$Status_{t0}$	$Unemp_{t1}$	Emp_{t1}
2004	Unemp	13426	7722
2005	Unemp	15136	8139
2006	Unemp	14110	7559
2007	Unemp	13685	7287
2008	Unemp	14024	7404
2009	Unemp	18625	10244
2010	Unemp	20097	9833
2011	Unemp	18223	7977
2012	Unemp	15941	6841
2013	Unemp	14997	6756
2014	Unemp	15064	7067
2015	Unemp	15497	7339
2016	Unemp	15066	7383
		SUM	203,891

The data have so many NAs in both qualitative and quantitative features. To be able to run a ML classifier that is so sensitive to NAs, NAs in this data were replaced as follows:

* Qualitative features/variables are encoded as factors which have levels as provided in Appendix C. In order not to drop NAs, they are regarded as another level which are unknown in each qualitative features.
* In quantitative features, NAs are filled with median's of that feature for the current survey. Results do not show significant changes.

3.5 Research Design

As it was previously stated, since the dependent variable (feature) is a dichotomous one that indicates current employment or unemployment, it should be analyzed with discrete (or qualitative) choice models in econometrics or supervised-classification problem in machine learning literature. Greene (2003) splits discrete choice models as binary, multinomial or ordered [13]. So, in econometric terms, the problem can be regarded as a binary choice problem.

[13] And event counts, but it is not appropriate the case of this study.

In machine learning literature, classification problems can be solved with numerous methodologies. But here, the models which are shallow are preferred. The reason why shallow models are chosen is that since the econometric methodology can be regarded a one-layered neural network with some inputs that are weighted with their coefficients, an activation function that is a sigmoid and one output that is probabilities which are estimated with marginal coefficients is produced consequently. The whole econometric process, therefore, can be called as a shallow neural network which consists of one perceptron. It also allows to compare easily both estimation methodologies. Hence, the algorithms used in this study are given below:

Table 3.3 : Machine Learning Algorithms.

Function $\mathscr{F}$	**Methods**
Naïve Bayesian Classifier	
	∗ Gaussian
	∗ Bernoulli
Support Vector Machines	
	∗ Support Vector Classification (SVC)
	∗ Linear SVC
Decision Trees	
	∗ Extra Trees Classifier
	∗ CART
	∗ Logistic Model Trees
	∗ Gradient Boosted Trees (XGBoost)
Discriminant Analysis	
	∗ Linear Discriminant Analysis
	∗ Quadratic Discriminant Analysis
Stochastic Gradient Descent	
	∗ SGD Classification
Neighbors Classifier	
	∗ KNeighbors

Table 3.3 (Continued) : Machine Learning Algorithms.

Function $\mathscr{F}$	**Methods**
Ensemble Methods	
	∗ Random Forest
	∗ Bootstrapped Aggregation (Bagging)
	∗ Adaboost
Neural Networks	
	∗ Shallow/Wide Artifical Neural Network

All modules except the neural network were run under Python thanks to the Scikit-Learn (Pedregosa et al., 2011), the marvelous package, while shallow neural network was developed under Keras (Chollet, 2015). All were developed with and written in Python.

In these machine learning algorithms, the data must be split into two parts[14]: Train and test data (Breiman and Spector, 1992). An incisive separation of the data makes it vulnerable in prediction. In order to prevent that problem, Stratified K-Fold with randomized but not duplicated is a well suited algorithm to split the data (James et al., 2013). Strata are determined by the years, in the data, it is 13. So at first, Stratified K-Fold algorithm splits the data into 13 strata. Then each stratum is shuffled. Later, shuffled observations are attained into a test set at 8% approximately without duplication. The shape of the train data is [188208,43], while test's is [15683,43].

The final data reached, after being filtered by several conditions stated above, total observations which survived are 203,891 who were unemployed in the previous year of the survey date. To deal with the computationally costly process of ML algorithms, all quantitative variables/features are normalized in 0 and 1 according to the survey year. In addition, the variables that are listed and used in Table B.1 and Table B.2 replaced with NA's. Since any machine learning algorithm is so sensitive to NA's, it should be done. The variables that are used are classified according to their types within a broad classification: quantitative and qualitative variables. NA's in quantitative

[14]Seperation of cross validation/development data is skipped.

variables are accepted as another category (encoded as 999) since all the level of that kind of variables has levels [15]. Hence for the quantitative variables, this process is simplified. All NAs are filled with the median value for the current year. Since median is more robust to outliers and large datasets, it is preferred (Rousseeuw and Bassett Jr, 1990).

Afterwards, a well-known econometric approach, logistic regression, for discrete/binary choice observations is applied to estimate transition probabilities for individuals who were unemployed in the previous term and moves to employment or stayed unemployed again for the current survey period.

3.6 Results

Descriptive statistics for the variables that are used in this study are given in Table B.1 and Table B.2. Quantitative features are summarized in Table B.1 while qualitative features are in Table B.2. Even the variable that was used to estimate population projections, AGIRLIK_ KATSAYISI is also utilized to estimate transition probabilities, as it conveys regional information to estimation. Again, the deflator is also used since it has a small but significant role to convey information about the state of the economic environment for Turkey to the estimation processes. Therefore, no other variables were derived and used in conjunction with the variables specified in the descriptive statistics.

Later, the algorithms stated in Table 3.3 are run to get transition probabilities for the same sample again. For the Support Vector Classifier (SVC), radial basis function is used as kernel with 1.0 penalty parameter. Linear Support Vector uses squared hinge loss function. All tree based classifiers, Extra Tree Classifier and CART, since the data uses two classes which contain impurities, criterion contains only gini for splitting the classes. Learning rate α for the XGBoost classifier is determined as 0.1 while the shallow network has 0.01. Logistic Regression Trees, Stochastic Gradient Descent Classifier and Linear Support Vector have their regularization terms prepared with L^2, the ridge

[15] Or factors, categories etc.

regression is utilized for them not to emerge overfitting. However, Extra Tree Classifier gives overfitted results due to catching information about next term employment for individuals.

Table 3.4 : Best Classifiers Results.

Classifier	Classes	Precision	Recall	F1-score	Support	Confusion Matrix		Accuracy	AUC
Random Forest	0	0.76	0.75	0.75	93740	70549	23191		
Train	1	0.75	0.76	0.76	94468	22852	71616		
	AVG / Total	0.76	0.76	0.76	188208			0.7554	0.8486
Random Forest	0	0.66	0.66	0.66	7811	5146	2665		
Test	1	0.66	0.66	0.66	7872	2653	5219		
	AVG / Total	0.66	0.66	0.66	15683			0.6609	0.7404
XGBoost	0	0.73	0.73	0.73	93740	68501	25239		
Train	1	0.73	0.74	0.73	94468	24955	69513		
	AVG / Total	0.73	0.73	0.73	188208			0.7333	0.8243
XGBoost	0	0.67	0.66	0.67	7811	5176	2635		
Test	1	0.67	0.68	0.67	7872	2534	5338		
	AVG / Total	0.67	0.67	0.67	15683			0.6704	0.7492
Ada Boost	0	0.65	0.6	0.62	93740	56347	37393		
Train	1	0.63	0.67	0.65	94468	30916	63552		
	Avg/Total	0.63	0.63	0.63	188208			0.6371	0.6948
Ada Boost	0	0.64	0.57	0.6	7811	4661	3150		
Test	1	0.62	0.68	0.65	7872	2559	5313		
	Avg/Total	0.63	0.63	0.63	15683			0.6360	0.6958
Extra Tree	0	1	1	1	93740	93740	0		
Train	1	1	1	1	94468	26	94442	OVERFIT	OVERFIT
	Avg/Total	1	1	1	188208			0.9999	1.0000
Extra Tree	0	0.62	0.69	0.65	7811	5409	2402		
Test	1	0.65	0.58	0.61	7872	3324	4548		
	Avg/Total	0.64	0.63	0.63	15683			0.6349	0.7073
						60917	34702		
Logistic Regression		0.6371	0.5999	0.6179	102340	40634	67638	0.6305	0.6300

The results indicate that the ML algorithm is relatively better than an econometric one, the Logistic Regression. In order to keep comparability between econometric and machine learning solutions, the shallow neural network is prepared as one layered since other algorithms including logistic regression has one process. Top two algorithms which get close accuracy rates are XGBoost Gradient Descent Classifier and Random Forest Classifier, their accuracies are 67%, 66%. Both of them use decision trees which is so suitable for the classification problems. As the classes that are employed or unemployed at the survey date but are unemployed in previous term are not imbalanced, namely the ratio between two classes do not exceed 95 % in any year, these classifiers successfully distinguish the classes by using characteristic information about individuals. In addition to these, impurity metrics for all classifiers except SVC, Linear SVC and Shallow NN are calculated and concluded in Table B.4 with Gini impurity, Cohen's κ, Entropy and Class error. Impurities indicate Shallow NN, Extra Tree Classifier and Quadratic Discriminant are not valid for any use.

AdaBoost Classifiers follows these top two, where accuracy rate is around 63.6%. Extra-Tree classifier comes from later than AdaBoost Classifiers and gets 63.5% which is quite similar to AdaBoost's. But these latter classifiers perform so close to econometric approach since logistic regression which has a determined cut-off probability gets nearly 63% accuracy (see Table B.5). In order to predict t1 term's unemployment (0) or employment (1), the cut-off value is taken as 0.5. So if the probability is greater than the cut-off, the label gets a 1 which indicates predicted t1's situation is employment, otherwise, it is 0, that is individual stays unemployed in survey date. While logistic regression is beaten by four machine learning classifiers, it can successfully beat the rest of classifiers that are listed in Table 3.3 and resulted in Table B.3.

When it is elaborated why machine learning estimation methods could beat econometric ones, the following reasons come to mind. As stated in the literature, Mincer (1974)'s equation establishes a non-linear equation where experience is taken as squared as independent variable. Since, especially XGBoost and Random Forests are completely superior tools when there is

a non-linearity between features (explanatory variables) in a classification problem, they can deal with the problem rather than linear solutions even if the non-linear features do not feed the model. Second, linear models are good at monotonic relationships. Non-linearity breaks the assumption about monotonicity, therefore, machine learning estimation methods work splendidly. Theoretically, tree-based models in principle can approximate functions regardless of shape, whereas linear models can only produce functions with a linear shape with respect to a chosen set of features. Third is related to cut-off point. In logistic regression, it is required to determine cut-off as 0.5 where the z-value turns 0 indicating the class is not determined. However, in machine learning algorithms, this cut-off value does not depend on any value. Owing to determine the classes the path of trees that reach 1 or 0, there is no cut-off value which should be optimal.

The last concerns the size of the data, by virtue of processing relatively big data, computational issues take more time than expected. XGBoost and Random Forest classifiers can run in approximately 0.0007% of the time logistic regression and can beat it with ease, machine learning estimation methods can be chosen if the task needs to be solved within a certain time.

3.7 Concluding Remarks

In this study, the main aim is to predict how individuals who are unemployed in previous term pass to employment or stay at current state with his/her characteristics and qualifications. In order to do so, Turkish labor market is handled with care by using relatively bigger data. Side task is to look at Turkish labor market with recent estimation methods which provide more accuracy compared to previous ones.

In order to look at transition from unemployment from a broader angle, the time span was chosen as wide as possible. Household labor force surveys which were decided on to calculate labor related issues were utilized from the beginning 2000. However, since it is impossible to derive the state of previous terms, starting year is accepted as 2004, then the data are expanded to 13 years, to 2016. Most of the variables are matched as much as possible.

Though, some of them can give a clue to the present state of the individual in labor market, these variables are excluded. Survived variables are only related with characteristics, previous term and present qualifications of individuals.

Previous year state of individuals is known, however, present state is taken as unknown, which can be unemployment (0) or employment (1). Since the dependent variable that should be predicted is a dichotomous one, the problem turns out to be a binary choice model. Binary choice models can be solved with logistic regression in econometrics and classification methods in machine learning literature. For machine learning literature, classifiers and a shallow neural network with a sigmoid activation function that can be used to compute transition probabilities of each individual are used to compare them with logistic regression. The family of classifiers are Naïve Bayesian, Support Vector Machines, Decision Trees, Discriminant, Stochastic Gradient Classifier and Ensemble techniques.

By comparing accuracies, specifically the XGBoost classifier which is in Decision Trees, and Random Forest that in Ensemble have nearly the same, 67% accuracy rate to predict present state of individuals in the Turkish labor market. However, logistic regression with the same variables has slightly lower accuracy, 63%, this exceeds half of ML algorithms in general. Since the results obtained belong to the test data, the variables and models can be used to estimate new data produced in the future to predict an individual's transition probability.

It is important to estimate the likelihood of individuals going to employment in the future. Policy-makers, for example, may find individuals who will be employed in the future by using data from the past. They can predict which areas these individuals are educated. Thus, they can develop incentives, subsidiaries for those areas. Moreover, they can invest in education to cover the gap in the human capital that is missing in these areas. Identify common problems in individuals who cannot be employed. For these, they can develop policies aimed at eliminating the problems of re-education, re-training and common problems aimed at increasing their human capital. Based on these estimations, economists can realize their adaptations in the input and output

tables and act to develop social policies. Anyone who wants to use human resources effectively may also benefit from these estimates.

3.8 Limitations & Further Studies

The merging process for the surveys took more time in the study. Sometimes, it is impossible not only to merge the questions, but also to match the answers even if there are no changes in the questions. Due to this, some answers are generalized and are shrunk to cover the situations. In addition, some crucial questions cannot be matched due to significant changes in answers that are totally incompatible to each years answers. Again, since a change in the data collection procedure has occurred after 2014, frankly, the sampling procedure is incompatible with previous years.

Machine learning is actually a short way to deal with larger (big) datasets. However, it needs more powerful computers. Sometimes, it is needed to upgrade the GPU when computationally costly processes are required to run. In this study, decision trees and support vector machines (without a specific kernel) need a high capacity computer. An extreme case is the SVM. Regardless of the minimum requirements of RAM, a standard SVM difficulty is measured with $O(dN^2)/2$, where d is the number of dimensions while N is sample size. In this study, approximately 78.910^{10}, the process can run [16] in 43K-45K seconds. To deal with the complexity of computations, decreasing the normalization of quantitative variables was the first method but this only slightly, about 15%, decreased the time taken for computation.

As can be seen from the results, a good level of accuracy was not attained. The main aim is to compare simple machine learning algorithms and econometric applications (which are considered as a one-layered shallow neural networks), so accuracies are stuck at around 67%-50%. It should not be forgotten that these accuracies are obtained with derived variables which are well-known in labor economics literature. However, a deep neural network algorithm can easily improve these accuracies. This helps policy makers, social security institutions, development agencies, human resources

[16] A GPU (860M) with 1152 cuda cores accompanying quad-core i7 CPU and 16 GB DDR3 RAM in where it is run under Ubuntu and freed RAM ratio is 90%.

departments, headhunters to forecast both individual level and aggregated level of employment and transitions with good accuracy power.

4. PREDICTING SAVING RATES AND PROVIDING PRESCRIPTIVE STATISTICS AT THE LEVEL OF INDIVIDUALS AND HOUSEHOLDS FOR TURKEY: AN IMPLEMENTATION OF DEEP NEURAL NETWORKS

4.1 Introduction

> *Bring together things that have not yet been brought together and did not seem predisposed to be so!*
> (Notes on the Cinematograph)
>
> Bresson (2016)

A considerable part of the economics literature is based on income theories. Regardless of the definition given by these theories, income which is formed after tax liabilites has two parts in general, consumed and saved. Consumed part is used to preserve economic activity in market. Saved part is done to guarantee future consumption or is diverted to investments that may be made to increase future consumption or future investment. This is the basic behavior of households. If they are aggregated into a collective, saving which is a substantial concept in both positive and normative economics is crucial to increase either present or future welfare.

The synonym of saving and investment comes from two main views in economic history, real physical activity of the economic agent in Keynesian view and monetarist view. Keynesian school initially accepts that saving is equivalent to investment. In a Keynesian view, saving is that whatever remains after salary is spent on utilization of merchandise and enterprise, investment is what is spent on services or products which are not consumed, but are durable. In the macroeconomic jargon, saving equals investments, and national accounts use this equation, so, the Keynesian view may also be regarded as the National Account view in terms of saving. As Lerner (1939) gives a clear definition of the equality of saving and investment, the difference between them may only emerge if the definition of either of them

has been changed. Influenced by Robertson (1933a), the monetarist school technically differentiates the definition of saving by changing the equivalency of these two terms and stating that these are not equal but differ rarely. Saving ratio is actually the unit of accounts that exceed the aggregated expenditures proportion to the initial wealth or income. It cannot automatically turn into a value. Besides, investment that can be turned into physical capital, human capital or goods is a ratio of financial intermediaries that directly generate a value.

Apart from historical debates in the history of economics, other disciplines such as pyschology, sociology, neuroscience, advertising etc. also focused on saving behavior. Lunt and Livingstone (1991) and Wärneryd (1999) emphasize, within the view of individuals, people have several reasons to save, such as proving self-control over expenses, risk aversion to present and future threats, increasing mental and social well-being (Lunt and Livingstone, 1992). Argyle and Furnham (2013) examine the sociological factors as well as pyschological ones that affect saving behavior and conclude that age, education and income, migration, personal and community habits learnt may be the potential determinants of savings, besides, these are also influenced by monetary status such as consumption and saving.

In this study, it is focused on to accurately predict the saving ratio at the level of individuals and households in Turkey by providing a prescriptive statistical model. In order to do that, the following section provides a comprehensive literature review on saving in view of both macroeconomics and microeconomics. Since this study uses micro-level data, literature review, only for empirical studies, may mostly cover the microeconomics perspective rather than macroeconomics. Again, Turkish literature review is also appended later to both macroeconomic research and microeconomic ones. Then, methodology which summarizes and compares deep neural networks and econometric models that will be used in the analyses follows. These are given in the results. Finally, the study reaches a conslusion and gives clues for future studies.

4.2 Background

As the father of economics examines, Smith (1776) distinguishes two concepts when he looks at factors causing development in human societies, capital formation and saving. The latter one is added to the saver's capital then that individual may use it for his/her own purposes or lend it to get interest or share the profit made by shared renter. This happens at a micro level, however, if looked into at a macro level, the capital of society also increases with annual gains in the same manner. Therefore it causes economic development for a country.

However, an early elaborative and questioning article from Robertson (1933b) does the first re-conceptualization of saving by giving some definitions, after that, Robertson (1933a) publishes a responsive article discussing with Keynes on the definiton of saving. If looked at closely, Robertson (1933b), at first defines parts of disposable income as saved or spent. He states that an individual makes a saving if he/she spends less than his/her disposable income, but this happens on certain days. He means that the day is an exact one rather than a period which may change the concept of saving. Second, he differentiates a pigeon-hole of the concepts of lacking which may be involved in saving. The part that is not saved may cover the consumption forgone by not to consuming this which can give utility to him/her. So, at that time, he/she is without some utility. Reciprocally, saving may occur without lacking and vice versa. Another concept is hoarding, the unspent part is allocated to security or tools that can be used long term. Hoarding and savings do not neceserrily cover reciprocally. Conversely, Robertson (1933b) states antonyms of these concepts, dissavings, dishoarding and dislacking. Then, the meaning of these concepts may change according to two classes, entrepreneurs and the public. Finally, saving is the excess of total income over income spent on consumption goods, while investment is the excess of total income over income received from the sale of consumption goods, in which Keynes and Kahn assert they are naturally equal to each other. The debate on solving the duality between saving and investment go further with

Lutz (1938) by asserting the factors coming from entrepreneural activites. Notwithstanding that saving and investment are equal in an ordinary economic environment which sustains all expectations in the ex-ante period are met at ex-post period, however, this requires each economic agent to play their roles in accordance with steady development. In fact, it is actually an exceptional scenario that indicates all factors of the system behave in the "normal". Against this argument, Lutz (1938) posits that entrepreneural activity (which will be emphasized later by Schumpeter (1947)) aims to gain abnormal profits. He critizes Keynes (1930)'s equivalency between saving and investments with the assumption that normal profits are present in the market when incomes are counted without abnormalities. Second criticism considering to go out of micro level analyses and to enter macro level one is that it is not possible to predict or assume the tendency of the whole economic system to move up or down even if S [1] $>$ I [2] situtation that gives a clue about the downward trend of the economic system when the factors are assumed to be normal. When putting together the inequality inherited from and abnormal behavior of both entrepereneurs and other economic agents, it is impossible to draw a clear conclusion from the inequality, the only thing that can be known is the uncertainity of economic progess.

Another school where the roots of Swedish economists are coming from Wicksell (1898) responds this discussion after translation of the articles written in Swedish to English and calls themselves the Stockholm Theory. Wicksell (1898) begins the embryo of a theory of output as a whole and defined as a cumulative process that makes a clear distinction between the natural interest rate and market interest rate [3]. In the market, the equilibrium between them maintains monetary equilibrium which stabilizes the price level and real equilibrium that enables the realization of consistency between saving decision of households and investment processes done by firms in accordance with and to the extent of government allowances (Wicksell, 1898). As the thoughts of Wicksell (1898) heavily influence Keynes, Marx and Hayek;

[1] Savings.
[2] Investments.
[3] Actual value of the real interest rate.

Lindahl (1930), Myrdal (1927), Lundberg (1937) and Ohlin (1937) construct their school of thought on the basis of Wicksell. Myrdal (1927) proposes earlier than Robertson (1933b)'s "the day" concept by differentiating ex-ante as expected and ex-post savings where S_e equals to I - C[4] as expected at the beginning of the "day" by income receivers and spenders. Lindahl (1930) reconstructs the ex-ante and ex-post concepts in order to reach a timeless concept, and in that direction, proposes periodicity of savings rather than refraction of a certain time point, it is also used later by Ohlin (1937) later.

After that, Keynes wrote a response to Robertson (1933b) in order to clarify a misunderstanding on the equality of investment, where he also went further to explain the definition of saving (and also investment) later (Robertson, 1933a). At first, he defines the excess of income over a certain periods expenditures, where the part equal to the value of output contributing to accumulated wealth as investment, where their equivalency is inevitable. He underlines period as established by Lindahl (1930) and Keynes (1936). Current excessive part of income over consumption is meant by Keynes as surplus for a while, which may not be equal to investment. Keynes (1936) punctuates the difference between surplus and savings in order to clarify the definitions given by Robertson (1933b) by going deeper and deeper by reemphasizing the specific term, day, which is the same as current.

Later, Keynes (1936) leaves behind the definitions of saving which is clear so far and moves towards the motives of savings which is stable over time and which factors increase the tendency of it among individuals. He gives possible motives triggering saving in individuals, not for firms, as below:

* **Precautionary saving**: This type of saving keeps individuals from unexpected shocks and uncertainty.
* **Foresight**: Economic agents may want to deal with fluctuations in anticipated changes over income and to meet foreseen changes in needs.
* **Calculation**: Individual prefers to get rewards to give borrowings to others from his/her savings, namely, to earn interest.

[4]Consumption.

* **Improvement**: Individuals gradually increase life-quality by increasing future expenditures.
* **Economic Independence**: Here, the individual may want to gain power or economic power to do things that he wants without a clear and specific aim.
* **Enterprise Motive**: As can be understood from the name, an individual may want to be an entrepreneur to establish a firm to carry out business or speculative movements.
* **Bequest**: To leave a fortune to devisees as inheritance.
* **Avarice Motive**: The individual has a tendency to being greedy, the nature of desiring more.

A valuable contribution to these motives is made by Browning and Lusardi (1996) who also implie a chaotic and diversified nature of the human being which has heterogenic motives towards saving behaviour as follows:

* **Accumulation**: Individuals may invest in durable goods such as houses, cars, land and others expecting a future increase in its value or by ensuring his/her heirs future.

The initial and comprehensive examination of motives may yield a valuable understanding of how individuals and households save. Moreover, this may also contribute to shape economic decisions of growth models to apply them in accordance with the country dynamics. In this way, a comprehensive review of how the saving motives were born and how they can affect ethics, society and economics is done by Paschke (1961) by starting from ancient writings to Keynes as follows:

Table 4.1 : Saving Motives Review.

Scholars	Saving motives	Major Motives		
		Ethic	Society	Economy
Xenophon	Capital stock, honor	**		**
Aristotle	household duties, state respect	**	*	
Assisi, F.	discipline, limitation of luxurious life	***		
Calvin	success, predestination	***		
Hobbes Th.	moderation	***		
Mandeville, B.	vanity	**		*
Justi, J. H. G.	prosperity, self-sufficiency	**		*
Quesnay, F.	precautions	**		*
Mercier de la	precautions in income management	**		*
Turgot, A. R. J.	self-denial, carefulness, betterment in status, interest	**	*	*
Hume, D.	usefulness	*		*
Bentham, J.	desires, usefulness	* **	**	* *
Smith, A.	sympathy, blissful happiness, betterment of social standard	**	**	*
Ricardo, D.	profit			***
Malthus, Th. R.	profit	*		**
Rae, J.	intellectual insight, public character	*	**	
Mill, J. St.	desire of capital, development, profit	*		**
Jevons, W. St.	future worries	*		*
Hermann. F. B.	self interest	*	*	
Schmoller, G.	natural condition, social strata		***	
Domela -Nieuwenhuis, A.	mental and social development, profit expectation		**	
Manteuffel, C.	moderation, instinct, future worries, capital increase		***	*
Böhm-Bawerk, V	interest comparison, psyche	*		**
Wicksell, K.	future utility, interest, age, family size, income	*	***	**
Schumpeter, J. A.	interest rate	*		***

Table 4.1 (Continued) : Saving Motives Review.

Scholars	Saving motives	Major Motives		
		Ethic	Society	Economy
Cassel, G.	usefulness, interest, income, future worries, standard of life	*	*	***
Clark, J. B.	interest, income, habits, social standard,			***
Marshall, A.	love for the family		**	**
Stackelberg, H.	limitation of consumption, betterment in the living standard	*		
Reichenau, Ch.	social standard consumption habits, worries of children, cultural development		**	***
Voigt, F.	personality forms, income, standard of life	*	***	***
Keynes, J. M.	real income, social and institutional framework		*	***
Samuelson, P. A.	disposable income			***
Smithies, A.	standard of life, income distribution	**		***
Pigou, A. C.	assets, prosperity future prospects		*	**
Ackley, G.				
Hamburger, W.			*	***
Duesenberry, J. S.	feeling of insecurity, group belonging, high income, standard of life	*	**	***
Katona, G.	a huge variety of the psychological socio-psychological and economic influences	*	***	***
Note:	Signs of Asteriks reflect significance of works.			

As can be seen from the Table 4.1, household saving was discussed almost from the time of the invention of money to the present. While philosophers and theologists emphasized ethical problems of savings while considering both individual and society interests since accumulation, unnecessary hoarding and not sharing the wealth with others are objectionable or

inappropriate behavior in most religions, classical economists, in general, take into account the importance of economical and societal issues by considering making profit at micro level. Later, while marginalist economists focus on the economic interpretation of saving motives, Keynes wants to spread the motives of saving to institutional and social establishments of governments. Lastly, Katona balances between social and economic motives of saving by putting forward the significance of socio-psychological as well as economic influences.

Lerner (1939) summarizes all of the definitions on saving, dicrepancy of saving and investment, the assumptions on them and the objections by criticizing and examining Lutz (1938). In addition, Lerner (1939) also slightly supports Keynes (1930, 1936). He analyzes consumption and saving which can be stable functions and the concept of marginal propensity to consume (MPC), that the concept relies on psychological factors, what refers to long term equilibrium rather than short one. He emphasizes the relationship between multiplier and saving-investment equilibrium as defined by Keynes (1936). MPC and multiplier are, at first, ex-post concepts and are not related with ex-ante concepts which are the expected ones. He gives the cases where MPC=0 and the multiplier is unity and MPC=1 and multiplier is infinity in order to draw a picture of how important the span of the period is. Frankly, the undeniable fact is that saving (or equality to investment) can be analyzed in a certain period of time to reach an equilibrium which can be perfect but actually an imperfect one as proven by Keynes (1936).

On the basis of works done by Fisher (1930) and mostly Ramsey (1928), Keynes (1936) proposes a consumption function that depends on aggregate disposable income which is net income after tax liabilites as follows:

$$C_t = c_0 + \lambda Y_t, \tag{4.1}$$

where $c_0 > 0$.

While C is the definite period of consumption, Y is disposable income and λ is called MPC (marginal propensity to consume) which is in between 0 and 1. Equation 4.1 was extended into an autonomous component of consumption

as above. Partial derivative, $\frac{\delta C}{\delta Y}$, gives MPC while $\frac{C}{Y}$ results in average propensity to consume (APC). The function is also called the Absolute Income Hypothesis (AIH) where Keynes (1936) states that:

"men are disposed, as a rule and on the average, to increase their consumption as their income increases, but not by as much as their income increases."

The term of absolute means that the amount of money that an individual employed is compensated for her/his work. When it is aggregated by going further just one step, it is called household's income. Second step indicates aggregated total income which can be called national income. AIH indicates a current measure meaning that the accounting period of income should reflect the same period of expenditure. Formally, it does not measure less than one week or more than one year in most economic literature.

Against Keynes (1936)'s findings, Kuznets et al. (1946) collect income data (agricultural only) for the US between 1869-1937 to examine the process of APC and MPC. First, he notices that although the relationship between these two concepts seems $MPC < APC$ in the sort term, in the long term this relationship turns into approximately $MPC = APC$ which contradicts the presence of the short run. That paradoxical process is called later as Kuznet's paradox which leads to AIH that being revised as below by Kuznets et al. (1946):

$$C_t = \lambda Y_t. \tag{4.2}$$

Moreover, Kuznets et al. (1946) reveals that APC rises during recessions while it declines during the expansionary period of the economy, which contradicts Keynes (1936)'s approach. For saving, calculations of aggregated saving at the national level and other savings measures derived at first Kuznets et al. (1946) but expanded later by Goldsmith and Salant (1939) show that from the beginning of the study, saving ratios show a nearly constant process over the whole period except for the great depression. Since then, in the later period, economists search for new income theories to understand saving behavior which strongly depends on income.

The criticisms have come from Brady and Friedman (1947) who assert that APC relies on the relative income of recipients, compared to the average income of the population, rather than absolute income. Afterwards, a second income theory which is called the relative income theory is proposed by Bornemann and Duesenberry (1950). This research examines the income, expenditure and saving behavior difference between white and non-white people. If people are divided into income groups at first, and then each income groups averages are compared to those of whites and non-white in the US, Bornemann and Duesenberry (1950) findw out that averages of non-whites are generally lower than whites for each income level. Additionally, the dispersion over income groups at the level of differentiation for whites and non-whites exists, since variation among non-whites are lower than whites, whites relatively have higher incomes than non-whites according to a comparison of the income groups. Bornemann and Duesenberry (1950) generalize these kind of differentiations over a time series data and reaches a conclusion that the consumption function of an individual heavily relies on the current income of others, that is, relative income. In that direction, the relative saving ratio is also bound to others' savings. Hence, aggregated saving cannot be dependent on absolute income. For this purpose, as Bornemann and Duesenberry (1950) proposed and later Ackley (1951) summarized, Equation 4.3 was offered as follows:

$$\frac{C_t}{Y_t} = \alpha + \lambda \frac{Y_t}{Y_0}, \tag{4.3}$$

where C_t = Real per capita consumption of year t

Y_t = Real per capita disposable income of year t

Y_0 = Previous peak of real per capita disposable income and $\alpha, \lambda > 0$.

Tobin (1951) also examines the habitual discrepancy which can be inherited from family, society or ancestors between whites and non-whites and points out that the dependency of income which can be affected by the initial or current wealth is different among non-whites, therefore, possessing lower wealth may lead to shape the income earned or will be earned in the future. This criticisms paves the way for the life-cycle hypothesis.

In order to explain the habitual discrepancies amongst a nation, society, groups, households and of course families, Glick (1947) examines the concept of family life cycle. The changes in life cycle is drawn with a step function, these changes transit to other social units with transaction points then internalized changes are passed on to descendants, as Glick (1947) explains. This process first emerges in a family and then spreads to other social units with an avalanche effect. From that reference point, Modigliani and Brumberg (1954) propose the concept of life cycle income hypothesis (LCH) which covers the process of wealth accumulation over the adult stages of the individuals life cycle. Since each level of life cycle of an individual needs a different kind of desired level of consumption and saving the individual starts to borrow, acquire assets from new consumption habits etc. in order to meet them. The individual actually consumes a certain portion of her/his life-time income. The main aim, as LCH assumes, is that each social unit wants to maximize total utility of their entire life by choosing one of them: to consume or to save. Lydall (1955) and Lansing and Kish (1957) trigger to focus on the accumulation of wealth at the early stages of an individual, in general adult stages. Later stages, since requirements of basic needs stays nearly stable, borrowings are liquidated and then some wealth is accumulated.

Ando and Modigliani (1963) also operationalize the life cycle income of an individual, the equation can be constructed as they offered as follows:

$$A = A_0 + nY_d, \tag{4.4}$$

where A_0 is the initial inherited wealth of the individual for the current period, Y_d is the income of the individual earned in the n working period.

$$s = c(T - n), \tag{4.5}$$

Saving of an individual is as above where the individual's total lifetime is T and retirement period can be donated as $T - n$ what that individual consumes in her/his retirement period as c. While saving also is as Equation 4.6 below:

$$S = n(Y_d - C) + A_0, \tag{4.6}$$

From Equation 4.5 and 4.6;

$$C(T-n) = n(Y_d - C) + A_0. \tag{4.7}$$

Therefore, the substitution in Equation 4.7 takes the form of Equation 4.8:

$$c = \frac{Y_d n}{T} + \frac{A_0}{T}. \tag{4.8}$$

For simplification, $\frac{n}{T} = \alpha$ and $\frac{1}{T} = \lambda$

$$C = \alpha Y_d + \lambda A_0, \tag{4.9}$$

where λ is again MPC of wealth and α is MPC of income while APC now equals $\frac{C}{Y_d} = \frac{A_0}{Y_d} + \alpha$ in Equation 4.9 where also $\alpha, \lambda > 0$ again.

The concept of life cycle income assumes fluctuations over both income and consumption/saving consequently. Moreover, it also requires that the expenditure on durable goods such as automobiles, houses or fridges should be counted as investments in capital which actually yield a return in the form of services since they increase total lifetime utility and the quality of life which also boosts utility again. Ando and Modigliani (1963) assert that the results of life cycle hypothesis would differentiate among income level, age, gender, etnicity, marital status, family size and so on. In addition, the theory requires that information on initial acquired wealth are available, and also requires data on life cycle which impact on retirement plan coverage when the permanent income in a life cycle may fall or fade out.

Owing to the life cycle theory, at the same time, it is pointed out that consumption decisions for a certain time horizon may exceed the peroid of income receipt. Because an individual is going to plan the expenditure he/she makes over a long time, i.e., life time, by assuming a permanent income. At that point, Friedman (1957) splits lifetime income into permanent and transitory, consequently the theory is called as permanent income hypothesis

(PIH). In the light of this, consumption may also be split into two parts as with income. The main criticisms raised by Friedman (1957) is that annual income is not a good predictor for annual consumption, since the permanent income cannot be predicted by using current annual income, it requires a long time series. Friedman (1957) states PIH as follows:

$$c_p = k(i, w, u) y_p, \tag{4.10}$$

Equation 4.10 gives a relation between permanent income and permanent consumption, in which i, w and u denote interest rate or a subset of it at which an individual can lend or borrow, the total property and nonproperty wealth relative to income, taste and preferences of individuals respectively. P subscipts indicate permanent consumption and permanent income. Permanent income actually is a part of income as follows:

$$y = y_p + y_t, \tag{4.11}$$

T subsript denotes the transitory part of income, while this denotation is also used for consumption as below:

$$c = c_p + c_t, \tag{4.12}$$

The first empirical research is carried out Bodkin (1959) on 1414 dividend receipents and their families. It was found that windfall, the transitory part of income, exists. Moreover, since there is no differentiation between Kuznets et al. (1946)'s equation, MPC stays the same as APC similar to Friedman (1957) proposes. After this examination, Kreinin (1961) continues to pursue transitory parts of income by again using windfall. It is found that windfall incomes are generally allocated to consume non-durable goods or saved while permanent incomes are used to acquire durable goods, investments or to increase asset wealth. Gupta (1970) also provides some clues on the parts of permanent and transitory income. Later, Hall (1978) uses RIH to construct another income theory which is called the Random-walk Income Hypothesis (RIH).

The Euler equation of consumption in the presence of future uncertainty is constructed below according to Hall (1978)'s first corollary:

$$u'(c_t) = \beta R \mathbb{E}_t[u'(c_i + 1)], \tag{4.13}$$

where c_t is the consumption in current time which is a discrete time martingale is a stochastic process. Then utility function is assumed as quadratic one as given in 4.14:

$$u(c) = -(1/2)(c_b - c)^2. \tag{4.14}$$

where c_b is the satisfaction or bliss point of the level of consumption where marginal utility give zero utility. MU can be solved as:

$$u'(c) = (c_b - c) \tag{4.15}$$

If there is no effect on future expected utility, $R\beta = 1$ may be assumed, then;

$$(c_b - c_t) = \mathbb{E}_t[(c_b - c_{t+1})]; \tag{4.16}$$

and

$$\mathbb{E}_t[C_{t+1}] = C_t. \tag{4.17}$$

A random walk is a stochastic process ε_{t+1} will present:

$$\varepsilon_{t+1} = c_{t+1} - c_t, \tag{4.18}$$

which subsequently equals,

$$\equiv \quad \Delta c_{t+1}. \tag{4.19}$$

As expected, the random walk process that was proposed by Hall (1978) is a simple representation of expectation of changes in consumption is totally zero as stated in Equation 4.20 that is the final solution of RIH:

$$\mathbb{E}_t[\Delta c_{t+1}] = 0 \tag{4.20}$$

As it can be seen, Hall (1978)'s model is obviously an econometric model where it looks like the sum of errors are zero at final. Error parts of the model show unexpected incomes as following a random walk. Hall (1978) also proves that PIH is valid in using quarterly U.S. data on per-capita consumption of nondurables and services from 1948 to 1977, however, individuals can rationally deal with shifting income. Also, individual is tend to try to smooth their consumption in lifetime but not to comply with a decrease in his/her lifetime or current utility. The saving behavior also fluctates over time in parallel to consumption. Econometrical testing of income and consumption was also done by Davidson et al. (1978) and Engle and Granger (1987) and it was found that they are cointegrated, which these results have proven Hall (1978)'s hypothesis. Flavin (1981) uses an autoregressive model of Friedman (1957)'s permanent income hypothesis, where also extends it in accordance with the rational expectation theory, and raises controversial arguments to the life cycle hypothesis by concluding that the parameters of past coefficients of income are more sensible than others. Since then, Hall and Mishkin (1982) changes the data but used the same model as Flavin (1981), conclude that the results done in micro surveys are concurrent with Flavin (1981)'s.

Since, there are several income (and consumption) theories in economics literature, is not possible to include Fisher (1930) intertemporal consumption models, such as Laibson (2001)'s A Cue-Theory of Consumption, major and influential ones were given in detail.

Life-cycle model of saving behavior is a good contribution to the economic history, however, there are also some criticisms raised about it, at first, by Thurow (1969) who claims that the predictions of the life cycle model are inconsistent if the correlation between consumption and income is examined. King and of Economic Research (1983) also assert the inconsistency while Thaler (1990) binds that inconsistency to sensitivity of consumption to current income. Thaler (1990) claims that income does not follow a path which is linear at the start and end, as middle-aged individual earns more, in that age level, people save more than compared to other periods of their lifetime, specially in their young and old periods. This downward move may not be as

expected as from the economic theory as shown by Avery and Kennickell (1991); Mirer (1979). There may exist a significant fall in consumption (Banks et al., 1998), however, other phenomena may emerge in elder people contradictory to utility maximizers. They may be peace of mind maximizers rather than utility.

The debate of economists on saving will not finish while other social scientists mostly are in favor of the definition indicating the part which is left of the disposable income after expenditures (Lunt and Livingstone, 1991), different from saving motives, for an average individual the first means of saving that come to mind is to devote a portion of the income to deposit money into a bank or purchase other assets to protect the value of money at present as Katona (1975) explains. For an average individual, saving is a rational behaviour to avoid future risks, vulnerabilities and insecurities. Accordingly, Katona (1975) indicates three main categories for saving habits in a society:

* **Contractual saving**: It is done to purchase an asset which is not affordable at present and is done by regular payments to secure the right to use this asset.
* **Discretionary saving**: As in the Keynes (1930), the individual considers himself/herself or descendants to protect future uncertainty.
* **Residual saving**: Default saving, individual who assumes there is no reason to consume more by using other monetary or similar tools, inevitably saves if she/he does not spend all of the income earned.

Katona (1975) shows these types of behavioral saving in the United States for the 1960s since it is wanted to focus on the latter two types. As observed in the average citizens in US, a great proportion of individuals prefer discretionary saving. Later, on the basis of Maslow (1954) theory, Xiao and Anderson (1997); Xiao and Noring (1994) examine eight saving motives proposed by Keynes (1936) against Katona (1975)'s saving motives. They call the structure of saving behavior as reflective of hierarchical financial needs of households and find that diversity in a household's resources increases the use of different saving motives rather than wealth or psychological saving motives (Lindqvist,

1981; Xiao and Olson, 1993). Another different pattern is found by Horioka and Watanabe (1997) in Japan, they assert that retirement, peace of mind and cautionary in case of illness are the main saving motives for Japanese households. The relationship between emergency funds and saving behavior has also been proven by Huston and Chang (1997) and Chen and DeVaney (2001). A comparative study for English, Israeli and Italian individuals by Webley et al. (2000) shows that while the English people tend to save in order to increase consumption in the near future, Italians take saving for their children into consideration more. Another study for Australians reveals that saving motives are concentrated into retirement and holidays in general according to Harris et al. (2002).

Saving motives or habits are not stable concepts since they can spread, pass to descendants or be taught to others. Wärneryd (1999) again classifies saving motives and states the first motive as a continual habit that is not related to either a specific aim or benefits of it. Dynan and Maki (2001), for instance, claim that saving behavior is taught to children to increase income over generations. The motivation for saving in low-income households is encoded by inheriting this motivation to future generations because future generations have a belief that they will generate more income, and that their future consumption with the intergenerational life cycle model can increase over time. Setting aside a constant proportion of income by households is called as “rule of thumb" behaviour which is not compatible with life cycle income hypothesis (Browning and Crossley, 2001; Rodepeter and Winter, 1999). As Carroll and Summers (1991) stress that the evidence of “rule of thumb" can be observed in almost all households. Smith et al. (1997) go further to re-evaluate the importance of bequest apart from the life cycle model, an individual learns to plan to leave a bequest to their descendants for the sake of decrease in their consumption, even use the basic needs to survive and to protect health.

Empirical literature counts several factors affecting saving behaviour at the level of both individuals and households, the most important ones being loss aversion, future income (expectations), number of earners, unemployment,

uncertainty, marital and health status, education, dependency ratio and ethnicity.

Loss aversion literature questioning consumer behavior (savings or consumption) starts with the prospect theory established by Kahneman and Tversky (1979). It mainly argues that the sensitivity of reduction in well-being of individuals is common rather than an increase in well-being. Broadly speaking, in economic terms, individuals are tend to risk averse. Thaler and Benartzi (2004) carry analyze the effect of risk aversion to saving behaviors and find that experiencing a drastic cut in gains directly reduces their disposable income. The optimal compensation against this cut is derived from the household by reducing current consumption and changing the expectations of future income (Browning and Lusardi, 1996). If so, households are likely to allocate an emergency fund (as a precautionary saving) as Chang and Huston (1995) found out. Conversely, Carroll et al. (2000) claim an increase in saving generally follows an upward trend in current income without being temporary or permanent. Likewise, according to Shapiro and Slemrod (1993), expectation of increase in future income may an increase saving behavior.

Expectations may become unclear if there is uncertainty in economic conditions, Sandmo (1970) asserts that uncertainty over a close increase in income may cause a reduction in consumption as well as a great increase in precautionary saving. Carroll and Weil (1994), Muradoglu and Taskin (1996) and Zeldes (1989) also find this behavior in households. Again, it is also found that an increase in the dependency ratio (or family size) or a decrease in the number of earners may significantly affect household saving, especially for investments on durable goods according to Lee et al. (2000) and Chang and Huston (1995), Lusardi (1998) also prove this econometrically. The importance of marital status is questioned by Browning and Lusardi (1996) and concluded that it is in favor of couples against singles, couples save more than singles. However, Bosworth et al. (1991) asserts that in terms of saving behavior, married couples act differently than unmarried couples thanks to income insurance.

Apart from marital status, health status is also so important, especially for elder people. In the latter stages of life cycle, Palumbo (1999), Starr-McCluer (1996) and Hubbard et al. (1994) propose that the health risk becomes the most important factor, since people compensate for health problems by decreasing savings. Kennickell and Lusardi (2003) found that total wealth accumulation on the later stages can be damaged in order to avoid health risks. Due to the emergence of new type of health insurance and diversification in this industry, Rosen and Wu (2004) reveal that the probability of the share of asset allocated for health insurance, superannuation or bequest increases as well, moreover, elder people tend to barter their risky assets to risk-free ones. Race is also another factor to give shape saving behaviour since Avery and Kennickell (1991) bring out white households save more than other ethnic households by bringing a question whether the saving behaviour can be habitual, inherited or learnt from society rather than ethnicity.

By virtue of that sphere of influence of life cycle model is great, Mayer (1972) questions the effect of education on the basis of Modigliani and Brumberg (1954) and discovers that education shaping tastes and preferences affect saving behaviour throughout life. Better educated individual saves more (Avery and Kennickell, 1991; Lee et al., 2000) than others, however, this relationship is not clear since these people earn more, therefore save more. Since then, Solmon (1975) looks at the indirect effect of education to saving behavior and concludes that different types of education have an impact to diversify saving motives and behavior regardless of the level of education. Solmon (1975) proposes different MPCs and APCs of individuals in order to prove this diversification among individuals. Attanasio (1993) uses different cohorts by generating synthetic ones and find hump-shaped saving profiles for each cohort and education group, which means there is a peak in precedence of retirement and a decrease again as before the peak. Chang and Huston (1995) and Huston and Chang (1997) claim education may shape emergency funds, a special kind of precautionary saving, since there is a positive correlation in between them.

4.3 Turkish Studies

Poterba (1994) indicates that Turkey is in the class of having lower saving rates according to the median saving rate. Due to the lower savings, Turkey is one of the countries that are inevitably foreign-dependent to finance its investments. In parallel with this, Uygur (2012) argues that the lack of savings rates among the primary sources of the current account deficit and the increasing dependence of financing of fixed capital investments on savings for sustainable growth away from being sustainable as it is debunked by Aizenman et al. (2004). During the 1978-80 debt crisis, Ozcan and Ozcan (2000) and Metin Özcan and Ziya Özcan (2005) showed that the savings rate decreased less during these periods while the investments decreased rapidly. In 1980s, it was aimed to increase investments in order to move to an export-oriented economic structure by abandoning import substitution policies and attention was given to increasing saving rates. However, as the public savings increased towards 1985, the economy shifted to financial liberalization on the one hand, and the floating exchange rate system increased imports and the private savings were in a period of contraction. Insel et al. (1989) discusses that money mishandling affected private savings in the period 1964-1986, while political inertia and uncertainty are one of the reasons for the low private savings for that period.

Interest rates and high inflation have led to an increase in private savings (Celasun and Tansel, 1993), while financial liberalization in the post-1989 period significantly reduced public savings as asserted by Kepenek and Yentürk (1994). After the 1994 crisis, the private savings, which once again fell back to their previous levels until the 1998-99 Far-East Asian crisis (Van Rijckeghem and Üçer, 2009), irrevocably 25% to 15% (relative to GDP) to a level where there was a continuous decline. In contrast to Turkey after 2000, in particular the developed countries, to increase their savings rate, raised it up to 35% from 25%, while in parallel, the current account deficit in Turkey increased rapidly during this period. Moreover, in the same period, Uygur (2012) finds disposable income groups are taken into consideration and

it is found that the 5th and 4th income groups in the lowest level continuously display negative saving rates. As Matur et al. (2012) reveal the determinants of saving behavior, another important factor which is the interaction of public and private saving is shown. It is also asserted that changes in income does not have any effect on consumption while it hurts governmental savings in parallel with private savings, however, an economy restricted by investment rather than saving may have a small impacts from that as Rodrik (2008) argues. After 2008, a policy, which is generally accepted and implemented by the Turkish Government as Alkin (2016) points out, reliesed on to push only GNP growth that is considered to increase savings. However it cannot be regarded as a fruitful method.

Apart from the macro level studies, Duygan et al. (2005) question the relationship between the consumption of durable goods (as a proxy of saving) and unemployment, it is found that the decision to purchase durable goods is affected by unemployment. Van Rijckeghem and Üçer (2009) show the evolution of savings of households in Turkey starting from 1998 to 2007 and reveals a significant decline in private and domestic savings from approximately 25% to 15%. A comprehensive studies done by Cilasun et al. (2009) and Cilasun (2009) aim to understand the behaviour of income saving and consumption for Turkey in between 2002-2006. In cross sectional study, Cilasun et al. (2009) find that hump-shaped income behavior reaches a peak at ages in 45-49, while better educated households show weak saving profiles compared other education levels. Cohort analyses in Cilasun (2009) show that savings profile also reaches its peak at 50s, but conversely, it moves constantly relative to income. Ceritoglu and Eren (2014) find that increasing the proportion of university graduates relative to increase in labor force participation contributes more to the savings ratio. Ceritoğlu (2013) also finds a similar behavior like the Australians, precautionary saving motives and income risk emerge as main saving motives in Turkey according to budget surveys.

4.4 Data

Household Budget Surveys (HBS) are collected at the level of both households and individuals by TURKSTAT[5] from 2002 to 2016. In addition, a third section about consumption[6] is also asked to individuals. All of the data can be matched according to “BULTEN" [7] variable [8]. If the matching process is intended to be done completely, the variables that are available at household level must be duplicated as the number of individuals surveyed in that household.

Before 2002, there were two surveys done in 1987 and 1994, which were not done regularly in the whole country. However, starting from 2002, HBS is done annually and regularly for Turkey. By means of harmonized indices of consumer prices (HICP), the sample size determined as a good representation of the whole country is increased at 2003. Later, it is decreased again. HBS is required to estimate consumption patterns and private consumption by households to predict their projections in national accounts, to calculate consumer price indices, distribution of income and, each goods consumed daily are recorded to an account. Afterwards, it is given to surveyors.

As mentioned before, HL and IL are the two main components of HBS. HL inquires at first the facilities of a house and durable consumer goods used by the members of that household. Then, it asks about the assets of those households such as car, hotels, summerhouse, land etc. IL asks the characteristics, working information, occupational issues and personal incomes from members of those households.

Total sample at the level of individuals matched with households is 629,372 with 247 variables. Except “BULTEN", 131 variables are merged at household level (HL) while 115 variables are adjusted at individual level (IL).

[5]Turkish Statistical Institute.

[6]Consumption dataset is not used for this study and is not matched.

[7]Since all the variables in the manual are kept in English version of the survey encodings as Turkish, whole structure regardless of translations of them will be used in this study.

[8]Whole tables of the variables are listed in Appendix C - Table C.3 for HL and Appendix C - Table C.4 for IL.

At first, disposable income is calculated from the variable called YILKULGEL indicating personal annual disposable income at the level of household. Total expenditure for that household is given by the variable HARCAMA but monthly, so it is multiplied by 12 to get annual expenditures at the level of households. Therefore, the difference between YILKULGEL and HARCAMA is total savings for that household, likewise, the ratio of saving named TASARRUF can be calculated from the ratio of that difference to annual disposable income called YILKULGEL as visualized Figure 4.1.

Afterwards, some filters are applied onto the merged data, as following:

* Negative savers are excluded, namely their annual expenditure exceeds their disposable income.
* People who may not be candiate saver are excluded, those are individuals who are a member of household but with no chance to earn and bring money to the household, are dropped. They must be 14 years old or younger.
* The households who have equal saving and expenditures are regarded as 0.001.
* Saving ratio is accepted as equal for all members of the household.
* Monetary measured variables before 2005 have 6 zeros since there is a change in TL to YTL (new Turkish Lira). So, their 6 digits are removed by dividing them by 1,000,000.
* After the saving ratio is calculated, all monetary measured variables are deflated according to the deflator of that year. So monetary variables are real ones in economic terms.
* The people or households who have negative monetary measured variables are dropped. It might be a result of from misencoding, incorrect declaration from surveyee or other reasons.

After that process, total number of observations which are at IL are 371,613 with 249 variables again. All NA values for quantitative variables are filled with their medians at survey year while all NAs in qualitative variables are taken as another category (encoded as 999) indicating unknown information.

In order to distinguish them, they are encoded as 99 which can not be exceeded by the values of qualitative variables. Descriptive statistics of quantitative variables are given in Table C.1. Tabulation for qualitative variables is shown in Table C.2.

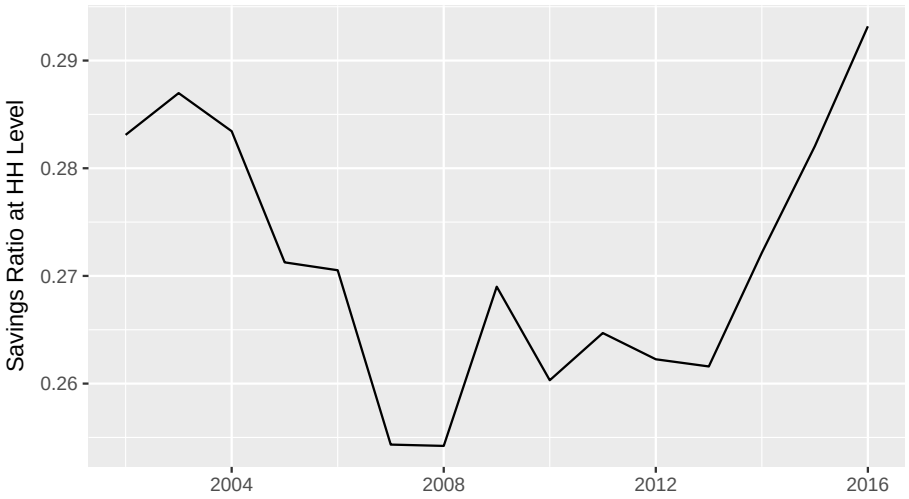


Figure 4.1 : Household Savings Over Years.

4.5 Research Plan and Analysis of Results

As previously stated, econometrically, quantile regression (QREG) will be applied to predict saving rates for both households and individuals. The QREG method is derived to median regression and is more robust to outliers which are abundant in the data. Against it, multilayered perceptrons called deep neural networks (DNN) or deep learning (DL) are constructed to predict saving rates in two ways: standart regression (DRNN) and quantile regression (DQRNN) approach. The main advantage of DL is automatically derived variables which are extracted by each neuron when there are at least one hidden layer in the deep neural networks. It is an alternative version to wide neural networks due not to extract features/variables at the beginning of learning procedure. Second advantage of DL is computational power against econometric methods. Since the data are relatively both large (371,613) and high dimensional (there are 249 variables), it is costly to compute the objective function of QREG. However, in order to minimize the differences between econometric estimation methods and deep learning architecture, DL is constructed similar to QREG. So, it also uses quantile regression methods

while computing gradient descent which is the same as QREG. Errors can be represented as the following:

$$\mathscr{L}(\varepsilon_i|q) = \begin{cases} q\varepsilon_i & if \quad \varepsilon_i \geq 0, \\ (q-1)\varepsilon_i & if \quad \varepsilon_i < 0. \end{cases} \tag{4.21}$$

in where q denotes the quantile and $\varepsilon_i = (y_i - x'\beta)$ are quantile's errors which is not required to be normally distributed but must be iid.

so average loss function is as follows;

$$\begin{aligned} &\mathscr{L}(y, (y_i - x'\beta)|q) \\ &= \frac{1}{N}\sum_{i=1}^{N}(\mathscr{L}(y_i - (y_i - x'\beta|q)) \end{aligned} \tag{4.22}$$

Since there must be correlated variables which can lead to multicollinearity among these 250 features, a distinctive approach is taken for qualitative variables. Apart from categorical dependency and tetrachoric correlation between dichotomous and continuous data, point biserial correlations are searched for to eliminate the variables (Lev et al., 1949; Tate, 1954). Point biserial correlation coefficients (PBSCC) are found by means of SciPy's (Jones et al., 2001) "pointbiserialr" module. The criterion for elimination is determined as if CC is higher than 0.70, the variables are candidates for that. Since CC is a bivariate relationship, the question becomes which variable must be dropped from the dataset. The second criterion is determined by considering the occurence of that variable. If the variable has CC over 0.70 and occured at least twice, it is eliminated, if not, it is manually eliminated according to the Author's intitution. Correlation between dependent variable is not searched for, because, it may lead to a vanishing or exploding gradient descent in DL. Same mechanism is also used for quantitative variables with Pearson's r if it is higher than 0.60 again, then determined variables are dropped.

QREG applications are run under the Python's statmodels environment (Seabold and Perktold, 2010) with the QuantReg package provided by Koenker (2013) which is so competitive to R's quantreg package developed

again by Koenker (2013) that is significantly slower than Python's, approximately 261 times [9] [10]. It allows computing quantile regression for predetermined quantiles, τ's simultaneously, ρ, residuals and fitted values without manual coding. However, at the time of writing this study, it computes R^1 automatically by naming it as Pseduo R-squared [11].

Deep quantile regressions are developed under Tensorflow created by Abadi et al. (2015) with the Python programming language. Tensorflow framework is highly efficient in GPU, deep learning process is estimated with 1152 cuda core graphical processing unit. However, in order to run GPU, accelerator of cuda cores must be run by means of cuDNN [12] developed by Chetlur et al. (2014). These improvements in algebraic computation and parallel programming in DL are maintained with the facilities of CUDA 9.2.

Since saving rates for Turkey at HH level (or individual level) is lower than expected, quantiles for saving rates are determined as .10, .20, .40 and .80 in order to cover low segmented saving rates comprehensively. All quantiles are valid for QREG and Deep Quantile Regression approach.

The dataset is split into two feature sets, quantitative and qualitative ones. The descriptive statistics for quantitative sets are given in Table C.1 while those for qualitative sets are in Table C.2. This separation is so crucial since the normal equation approach needs qualitative variables for categories they have.

The variables which have higher than 0.6 PBSCC in qualitative set as calculated in Table C.13 and which have higher than 0.6 again in Pearson's r are dropped at first automatically according to the first criterion in the quantitative set shown in Table C.13, and then dropped again in the rule of second criterion. Then total features of the dataset are decreased to 203 after the elimination procedure. So, the dataset is suitable for both QuantReg

[9] According to author's experiments.

[10] Another fast implementation of Quantile Regression can be done under Julia provided in here `https://github.com/pkofod/QuantileRegression.jl`https://github.com/pkofod/QuantileRegression.jl maintaining Stata-like results for econometricians, which is easy to read and understand.

[11] The code is here: Statmodel-reference-manual under the source code of quantreg written by Vincent Arel-Bundock for Python community.

[12] CUDA® Deep Neural Network library (cuDNN).

and DL. However, in order to decrease the time of convergence in stochastic gradient descent, all quantitative varaibles are standardized.

In DL, the dataset are split as train and test sets [13]. The shape for the train set is [343028,203] while test's is [28585,203]. The splitting procedure is conducted in the light of Stratified KFold algorithm by choosing years as strata, which are 15 years. Then train and test set are shuffled since there are more repetitions in the household dataset that is merged and matched with the individual dataset. In order to get convergence among gradient descent algorithm with Adam optimizer, deep learning algorithms run with minibatches which has a size of 10,000.

DL architecture is developed and programmed under Python with the Tensorflow framework with the following properties:

* The number of layers is four between 203 inputs and one output consisting of the saving rate prediction, so there are three hidden layers as shown its architecture in Figure 4.5.
* First hidden layer has 250 neurons, second one has 20 and last layer has 4 neurons. Then, one neuron in the fourth layer estimates outputs.
* First activation function is tanh between inputs and first hidden layer. Second activation function is ReLU between first and seccond hidden layer. Third activation function is softmax. Fourth one is tanh that predicts saving rates for each individual.
* Number of epochs is determined as 100 for each quantile.
* A bias term accompanies each layer.
* Learning rate for deep quantile regression, namely α, is determined as 0.003 after numerous repetitions.
* Adam Optimizer is accepted as the best one, so it is programmed with $\beta_1 = 0.9$, $\beta_1 = 0.999$, ε = 1e-08 by not using locking.
* Loss (or cost) function that should be minimized with the Adam Optimizer is encoded from Equation 4.21 for each quantile[14] .

[13] Development or cross validation set is not used.

[14] Combined loss damages total loss, then, it is taken as individually.

4.5.1 Feature selection (FS) process

In this study it is ruled that the more information a study has, the more precise results can be gained. However, recent studies also focus on feature selection procedure to increase their precision. In addition to all steps counted before, a three-step feature selection procedure will also be applied for deep quantile regression and deep regression. These steps are calculated by using Feature Selector (FS) from Jie Yuan [15] as follows:

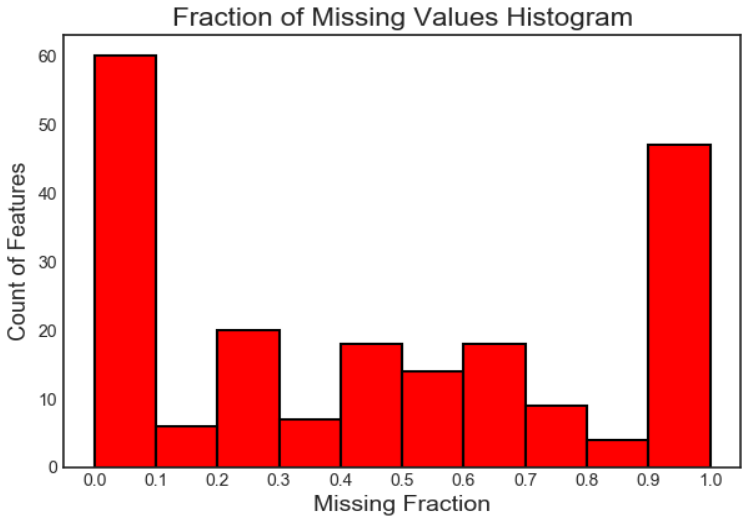


Figure 4.2 : Missing Value Fractions.

[15] https://github.com/Jie-Yuan/FeatureSelector

* Missing value analyses are done through “missing fractions” module of Feature Selector. If a variable has more than 60 % missing values, the variable has been dropped. A main graph and table for missing values are given in Figure 4.2.
* Collinearity analysis is made with “identifty collinear’ module of Feature Selector. Bivariate collinearity may not exceed .80. Graphical abstract and table of the result are given repectively Figure 4.3 and tab:collinall.
* Importance of variables with their factor loadings by using “identify zero importance” module with Gradient Boosting Model are calculated in Feature Selector. First 32 variables in terms of loadings are shown in Figure 4.4 and are listed Table C.17.

The first results in quantile regression are presented in Table C.5 for q=.10, in Table C.7 for q=.20, in Table C.9 for q=.40, in Table C.11 for q=.80. There is no multicollinearity among the explanatory variables/features, so, QREG is healthily finalized. The performance indicator for QREG, R^1 scores for quantiles .10, .20, .40 and .80 are 0.040, 0.056, 0.093 and .090 respectively, the average R^1 is 0.0675 which is relatively low. Second applications of quantile regression with the data applied a feature selection are presented in Table C.6 for q=.10, in Table C.8 for q=.20, in Table C.10 for q=.40, in Table C.12 for q=.80.

In the DL algorithm developed under Python with the Tensorflow framework as represented in Figure 4.6, for most of the quantiles, gradient descent converges after 6 or 8 epochs. However, it was not stopped. Losses for each quantile are 0.024 for .10, 0.044 for .20, 0.069 for .40 and 0.056 for .80th quantile as they can be seen in Table 4.2. The average loss is approximately 0.04825. However, DL is outperformed if R^1 is approximated with the predicted $\hat{y}$, it is 0.9294 which can not be compared with R^1 in the normal equation result. The results do not show significant changes after feature selection procedure in Figure 4.7 and in Table 4.3 .

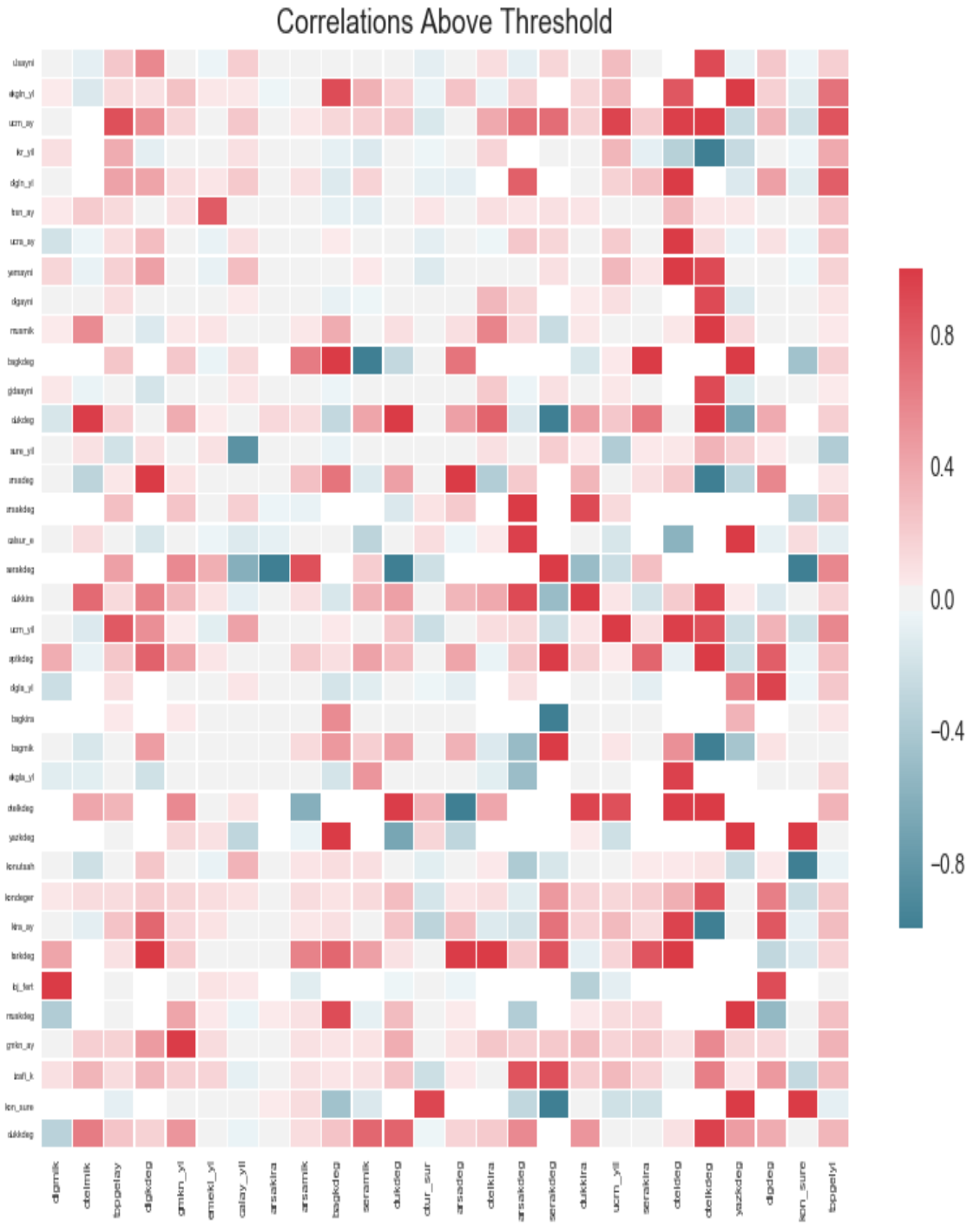


Figure 4.3 : Collinearity of Variables [Top 26].

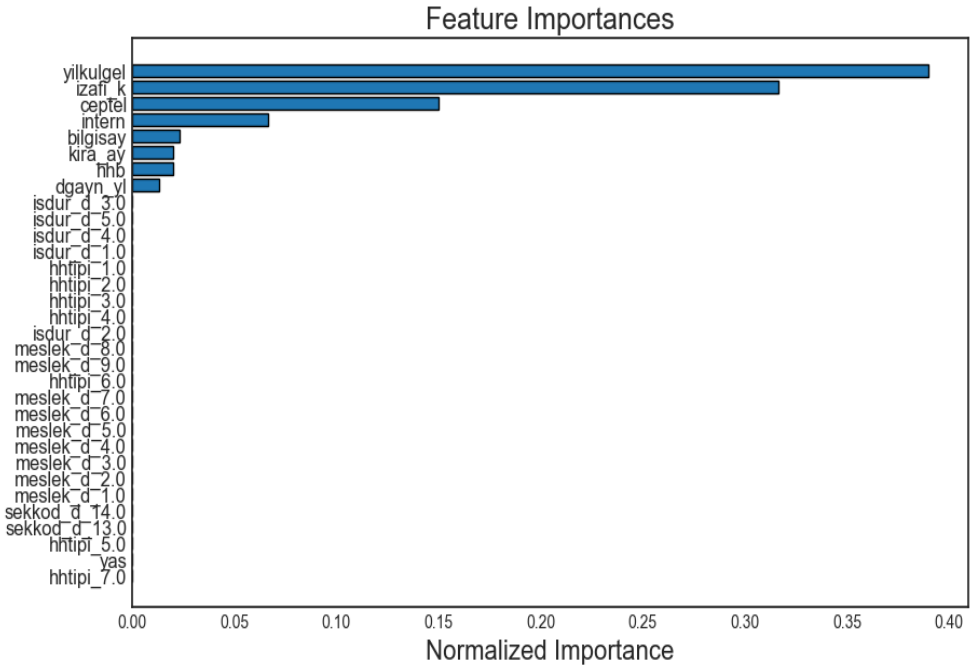


Figure 4.4 : Variable Importance Factor Loadings [Top 32].

Table 4.2 : Performance indicators for QREG, DRNN, DQRNN w/o FS.

Quantile	QREG's R^1	DRNN's R^1 (Total Loss)	DQRNN's R^1 (Total Loss)
.10	0.0400	-	0.9185 (0.0246)
.20	0.0560	-	0.9199 (0.0445)
.40	0.0930	-	0.9404 (0.0693)
.80	0.0900	-	0.9388 (0.0565)
AVG	0.0675	0.7903 (0.3153)	0.9294 (0.0487)

The weights that were learned from the training set comprise a general prescriptive statistics for the saving rate to predict it at household level or individual level. If the parameters used in this study are collected from the members of any household, an agent can predict it with approximately 93% accuracy from the developed deep learning model. As given in Table 4.2 and Table 4.3, feature extraction on variable set do not changes the results significantly. However, standart deep regression model again otperforms well against normal equation quantile regression method with its precision level, 79% on average.

Table 4.3 : Performance indicators for QREG, DRNN, DQRNN w/ FS.

Quantile	QREG's R^1	DRNN's R^1 (Total Loss)	DQRNN's R^1 (Total Loss)
.10	0.0347	-	0.9208 (0.0415)
.20	0.0563	-	0.9081 (0.0617)
.40	0.0884	-	0.9214 (0.0915)
.80	0.1563	-	0.9310 (0.0413)
AVG	0.0839	0.7691 (0.1223)	0.9194 (0.0590)

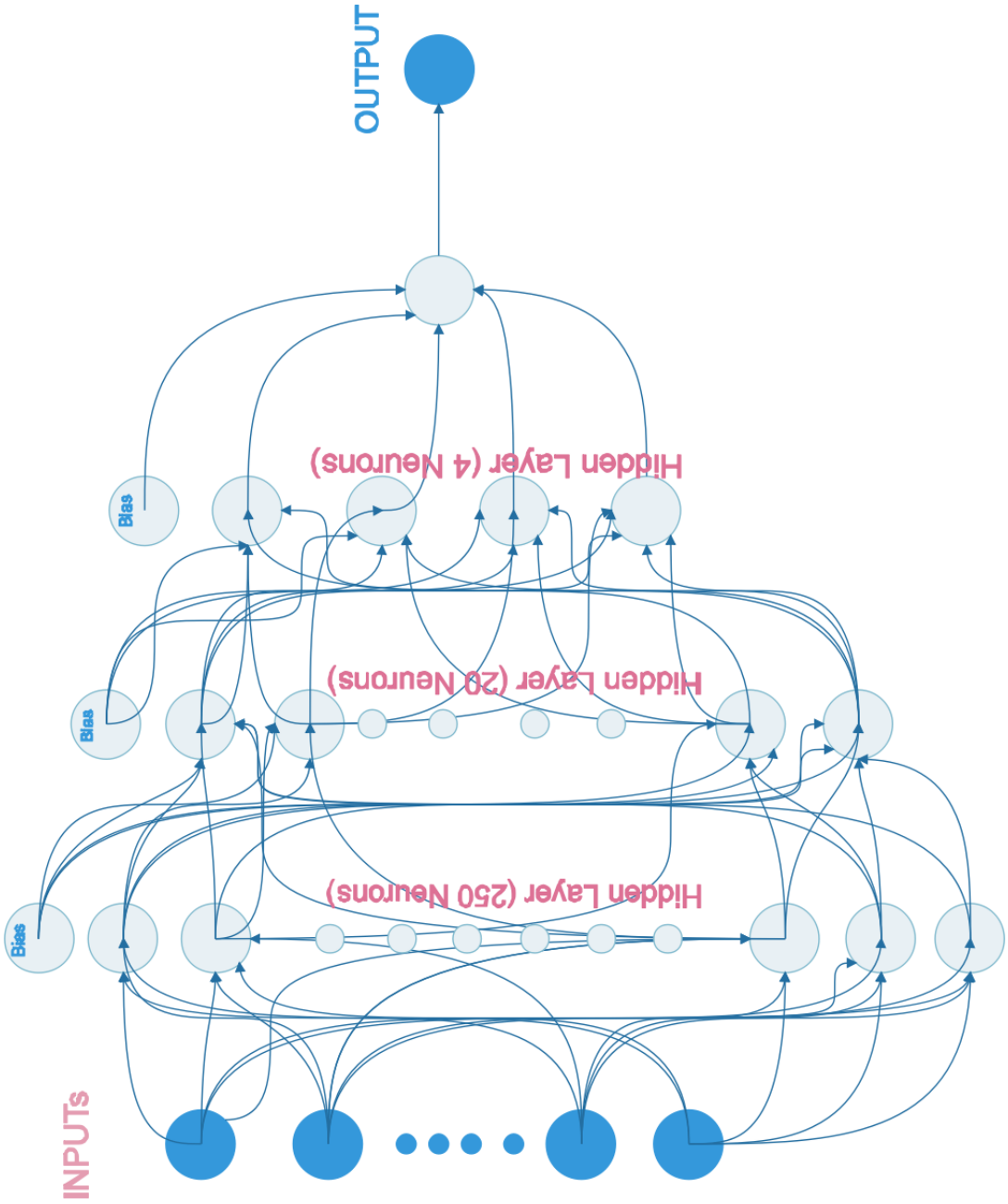


Figure 4.5 : Deep Neural Networks Architecture.

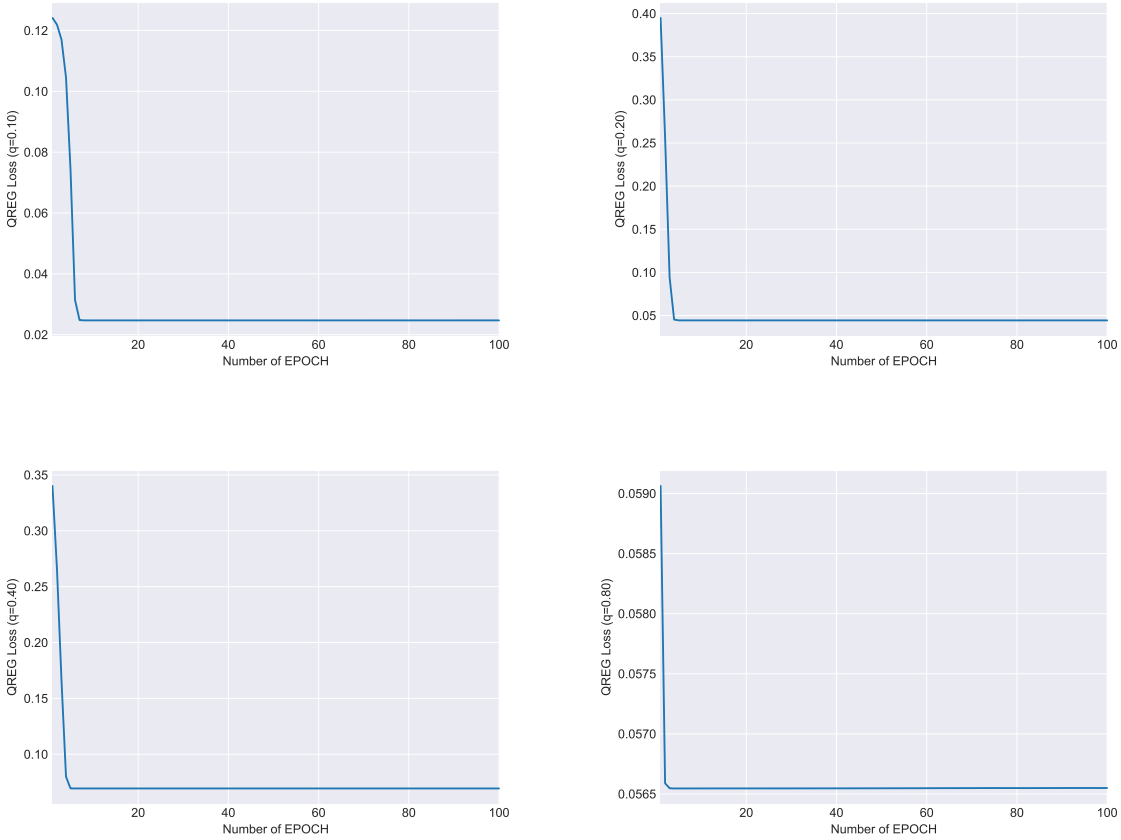


Figure 4.6 : Deep Quantile Regression Results w/o FS.

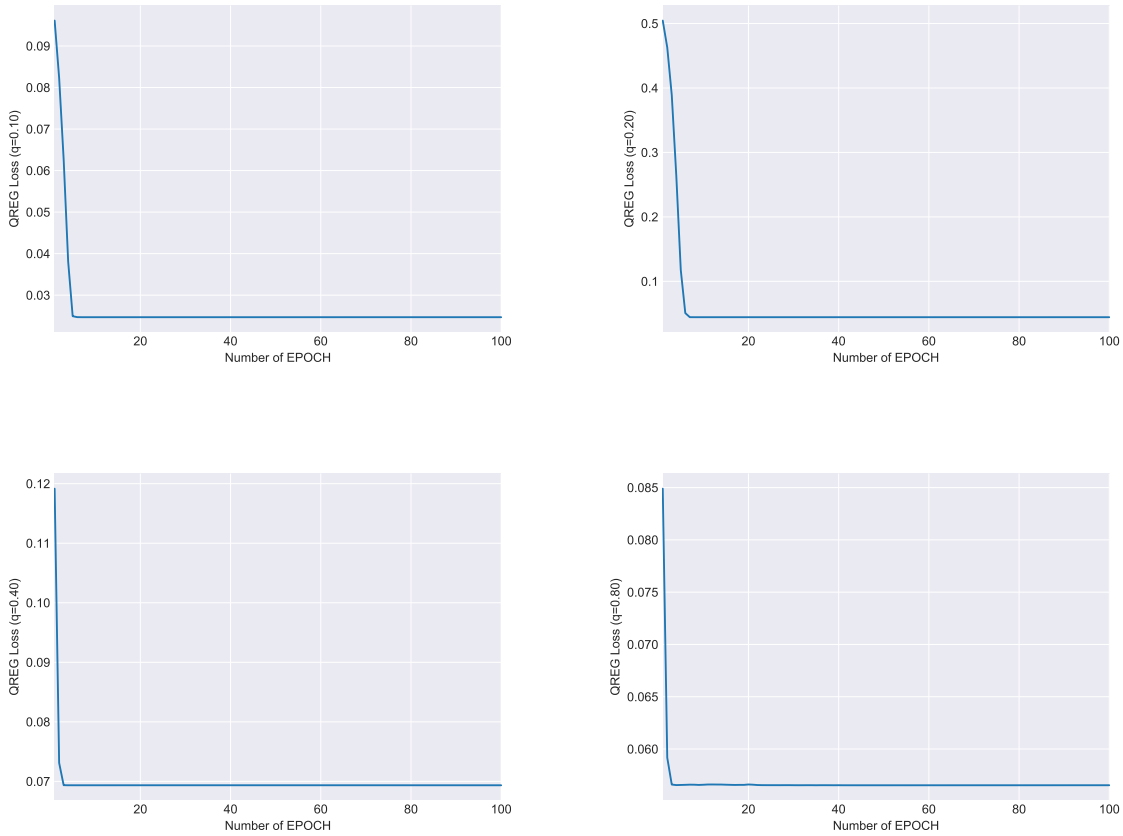


Figure 4.7 : Deep Quantile Regression Results w/ FS.

4.6 Concluding Remarks

This research focuses on predicting saving rates by providing prescriptive statistics for future forecasts by using individual and household consumer behavior, ownership, facilities, characteristics and regional factors. It is done under the age of artifical intelligence that sheds light on the power of data science and computational methods. So, this study provides high accuracy with high dimensional statistics, relatively large data and deep neural networks.

The data of this study comes from Household Budget Surveys at individual and household levels. Individual level of HBS maintains characteristics of individuals, income and consumption behavior, education, experience and occupational information while household levels of HBS provides useful information about private properties, facilities of the house and vehicles used by the members. Since the surveys are abundant in terms of useful information, most of them are adjusted and matched by not losing the meanings of answers, construction of question and contents of information. In the utilization of HBS, the time interval is kept broad as possible, so it is collected from 2002 to 2016. In order to keep most of the variables, without losing the meanings, most of the questions are matched.

At the first stage, the features are analyzed with point biserial cross correlations in order to make them feasible to both quantile regression in normal equation method which may suffer from a multicollinearity problem and deep quantile regression neural network which can easily be damaged by vanishing or exploding gradient descent problems due to weight transfer from one neuron to another. This logic is run again for quantitative variables to save models again those problems by using, however, Pearson's correlation coefficients (r). Then, finally, the dataset has 203 explanatory feature sets. This study go beyond it and decreases feature set by using feature selecting algorithms however results do not show significant changes.

In order to predict saving rates for both individual and household levels, utilized data are processed with quantile regression and deep quantile neural

networks. The quantiles for each process are determined as .10, .20, .40 and .80. The imbalance of quantiles is determined to predict saving rates accurately since it is densely concentrated under 40%. A four-layered deep quantile regression model predicts 93% on average while quantile regression in normal equation method does this at 6.25% on average. In between of them, deep regression model (DRNN) with the same architecture of DQRNN predicts 79% at mean. Since deep quantile neural network surpasses normal equation method with a better prediction power, it can be used by institutions, governments, NGOs, individual researchers who want to develop a policy on saving rates of individuals or households in Turkey or other countries.

4.7 Limitations & Further Studies

As it can be obviously seen, not only the time interval is so long, but the number of variables is so large. To conserve as much data as possible, however, it is required to read the answers carefully, interpret and re-evaluate them to match logically. It took more time than expected if it is taken into account that there are approximately 250 quesions in which 95% of the data needs that process.

Second limitation of this study is again about computational issues. Deep learning algorithms need at least 100,000 data points, but the smallest dimensions which the data could preferably be reduced to were still relatively bigger than expected. Dimensions multiply the process time in both QREG and deep quantile regression. Again, if the number of hidden units and epochs decreases, then all time spent on computation of gradient descent, losses and predictions are again multiplied. The computations in R and Tensorflow framework are run in a GPU (860M) with 1152 cuda cores accompanying a quad-core i7 CPU and 16 GB DDR3 RAM where it is run under Ubuntu. However, in R, estimation of a quantile regression took approximately 11 hours, while deep quantile regression in Python with Tensorflow framework took 7 hours. Due to errors, heating problems in both GPU and CPU, most trials were interrupted. A quantile regression in normal equation method is replicated under Statmodel's quantreg module under Python to decrease computation time. Eventually, Python can run quantile regression model for

four pre-determined quantiles in 2.5 hours which is four times faster than R's quantreg package.

In deep learning, the number of hidden layers may complicate the learning process. However, the main obstacles that increase computation time are the activation functions which placed a zero for each neuron. Initially a deep learning algorithm is run with a small sample, it is seen that most activation functions increase computation time significantly. So, an elimination process is run to get appropriate activation functions which decrease computation time. However, a strong environment, can compute a hard deep learning algorithm and may result in increased accuracy. In this environment as previosuly stated, it is not possible to do that.

The merged dataset provides useful information about individuals and households, so, it can allow deriving new variables to estimate saving rates. It may increase the prediction power of deep learning algorithms. But, frankly, there is no limit, a researcher can go further and further.

5. SEMANTIC ANALYSIS OF THE CENTRAL BANK OF THE REPUBLIC OF TURKEY COMMUNICATION REPORTS AND FORECASTING MODEL WITH LSTM

5.1 Introduction

I jumped in the river, what did I see?
Black-eyed angels swam with me.
A moon full of stars and astral cars
And all the figures I used to see.
(Pyramid Song)

Radiohead (Collins, 2004)

The existence of a gap between what we understand from the words, heaps of words, phrases, sentences as whole or larger units of discourse and what it is indicated in them is an undeniable fact that was debated throughout a significant part of history as summarized in Gordon (1985). Morever, this gap may be narrower in the future but it might be not closed completely as humankind wants. The area that mainly aims to decrease the gap is called semantics, according to Geeraerts (2010), this paper only searches for lexical semantics which is a special area in semantics. In lexical semantics, the meaning of word(s) is evaluated according to its lexical meaning since there may be several meanings that result from the tone of word(s), their order, implicit meaning, irony etc.

Lexical semantic is investigated in natural language processing (NLP) which is a subfield of computer science and of artifical intelligence. In NLP, it is mainly aimed at extracting meaning from unstructured data and to convey it to the computer in order to make the computer understand or to process human languages by using its hidden structures. However, the extracted meaning which is the most valuable part of this process should easily converted into lexical words in order to reverse the process from the starting point, which is called interaction. So, without interaction or the possibility of interaction, extracted meaning may be wasteful.

Breakdown in NLP is achieved by means of a statistical revolution in the beginning of 1990s. Manning et al. (1999) point out that the way of thinking on the rules of linguistics has dramatically changed NLP tasks to Artifical Intelligence (AI) tasks which are syntax, semantics, speech and discourse. Preliminary works in NLP is trying to catch the rules of a language and encode it into codes. However, statistical NLP (SNLP) is attempting to catch the rules and makes a generalization from them in order to start the learning process from text called corpora. Even though, this learning can change from language to language, the main job is so common: identification of semantic relationships. From the beginning to the present, the main SNLP tasks can be categorized as the following:

1. **Categorization**: Lexical linguistic modelling of summaries, indexing and searching, content duplication removal, classification of contents.
2. **Sentiment Analysis**: Indexing the mood of text, extraction of opinion and stances.
3. **Summarization**: Making a short summary from large corpora by putting forwards their significant parts.
4. **Machine Translation**: The hardest task, translation of text to other languages without giving any clue by hand.
5. **Text to Speech/ Speech to Text**: Speech recognition.
6. **Contextual Extraction**: Create structured information from unstructured texts.
7. **Topic Modeling**: Abduction and abstraction of large text to simple meaning.

As the nature of science follows a progressive path, these tasks may be changed in accordance with today's needs, technological improvements and methodological expansions, so these categories should be regarded as a simplification of tasks studied in SNLP in AI.

This research firstly aims to provide an SNLP model to understand central bank reports. In order to do it, the Summaries of Monetary Policy Committee Meetings published by the Central Bank of the Republic of Turkey (CBRT)

are collected and turned into large corpora to extract the meaning of topics listed under three main categories: inflation developments, factors affecting inflation, monetary policy and risks. These summaries are listed in CBRT's oficial website [1] five business days after the meeting. The summaries published between 2006-2018 are collected with a crawler built under Python as PDF files and then turned into large text by using PDFMiner coded under Python again. Then, those corpora are evaluated by using SNLP techniques and are turned into numeric values which measure the meaning of content. In order to prove whether there is an effect on market indicators, an econometric model is constructed to test it. So in the following sections, a literature review on semantic analysis, deep learning techniques and their application on economic contexts are summarized. Then the data and methodology are provided respectively. In the fifth section, the results obtained are shown. In the last two chapters, conclusions and some issues on limitations and expectations on future studies are given.

5.2 Past Researchs

Central Banks, authorized institutions, are generally responsible for monetary policy which mainly includes inflation targeting by making it stable over time (Blinder, 2004). Most of the research on central banks are concentrated on policy, transparency, effectiveness and the role of the CB in a market. However, central banks generally act explicity or implicitly by communicating with the market. Since market participants mostly know how to reach new releases issued by the CB, they directly follow press releases, indirectly they get summarized news from newswire services.

Blinder et al. (2008) concentrate on press releases issued by the CB in the US and examine the effects of it on the financial markets. It is found that the actors in financial markets are directly affected by interest rate policies of the CB. The effectiveness of the interest rate policy is also proven by Guthrie and Wright (2000),Andersson et al. (2002) and Kohn et al. (2003). Ehrmann and Fratzscher (2007) go further in order to discriminate the mood and the style

[1] http://www.tcmb.gov.tr/wps/wcm/connect/EN/TCMB+EN/Main+Menu/Announcements/Press+Releases

of communication strategies followed by the CB of England and the FED, and concludes that the CB of England uses collegial communication strategy but individualistic in decision making, and that the FED is individualistic in press releases but collegial in decision making, these difference do not affect the predictability of these banks policies which are found as equal. These studies confirm that market participants can predict complex signals from the FED minutes as shown in Boukus and Rosenberg (2006). Second impact of the CB may be on exchange rates (Melvin et al., 2009; Sager and Taylor, 2004), a detailed content analysis of the press conference held after the monthly Governing Council meeting shows that this effect is present and obvious on the volatility of exchange rates. This effect was previously proven by Jansen and De Haan (2005) indicating that the European Central Bank and national central bank officials can only intervene for the euro/dollar exchange rate but not for the euro, where the results are so similar in Dewachter et al. (2014). Besides, Gurkaynak et al. (2004) point out actions and words have different effects on asset prices, but statements have greater impact on long term treasury yields rather than actions as confirmed also by Brand et al. (2010) for the ECB again.

Apart from the communication tools stated above, policy-making commitee reports are affecting market actors significantly. Reinhart and Sack (2006) prove that official reports have undoubtly considerable impact on markets rather than individual speeches while testimonies are an accurate form of communication for the FED to shape the market in an efficient way. Another significant finding is that the more detailed communication is done by the FED, the more accurate understanding is appreciated by market reactors. The effects of formality of communication and detailed information are also proven for the Bank of England by Reeves and Sawicki (2007).

5.2.1 Statistical natural language processing (SNLP)

Here, since this literature needs to mention Statistical NLP, a brief literature review of it is given.

In the beginning of the 1950s, natural language processing researchers initially focus on syntax rather than others. A small part of researchers want to change

the focus from syntax to semantics such as Gordon (2003) and Schank (1975). Their aim is to provide a representation strategy on semantic relationships which is knowledge based. First order logic is the first attempt to create such a strategy implemented by Barwise (1982). Another model led by Chomsky (1956) is the production rule. This rule is based on "if" conditions and "then" applications. The procedure of the motivation of production rule starts with recognizing, then solving the conflicts in order to act. Since production rule systems have a simple control mechanism, the rules derived from them can be easily understood by people. Besides, it is really too hard to generate all the rules which is so time-consuming. A network model was first introduced by Pearl (1985) who takes into account only joint probabilities of sequenced words. Immediately after the implementation of networks, semantic networks provided a graphical network connecting nodes that represents knowledge Sowa (1987).

The idea of semantic networks changes the way of thinking about the semantic relationship between words, phrases and even sentences. Minsky (1986)'s groundbreaking work on establishing the inner network of the human mind constituting with smaller parts called agents that can work together simultaneously and perfectly without exogeneous affections profoundly influences works on studying human cognition, later called artifical intelligence (AI). Followers of Minsky (1986) maintains sequential logic based on common sense repository (Lenat and Guha, 1989), thought treasure (Mueller, 1998) and WordNet (Miller, 1998).

After the evolution of classical NLP research, in the late 1990s statistical natural language processing (SNLP) emerges, which changes all NLP works done in history. SNLP may have first been mentioned by Manning et al. (1999) stating that SNLP has a blurred boundary between machine learning, information retrieval and cognitive science. Manning et al. (1999) emphasize the significance of statistics in natural language processing while it also constitutes language/signal processing, probability and linear algebra. Hofmann (2001), afterwards, introduces the unsupervised machine learning algorithm as an SNLP solution while Nigam et al. (2000) make a clear

distinction between supervised and unsupervised machine learning algorithms on text classification in a categorization task.

At the beginning of the 2000s, SNLP wants to enhance the methodologies of those candidate ones that may improve the power of studies. The first one is the maximum likelihood methodology in machine learning proposed by Berger et al. (1996). Later, Nigam et al. (2000) use the expectation maximization algorithm while Lafferty et al. (2001) propose the conditional random fields (CRF). A well-known and old machine learning algorithm called support vector machine (SVM) was first used in categorization tasks by Joachims (2002).

5.2.2 SNLP and central bank communications

Interactions between different disciplines may produce significant results and contribute literature in a distinct angle. As deep learning has extensively influenced many fields in both social and natural sciences, SNLP and economics inevitably come together. Although there is no research paper using neural networks, they use basic tools of SNLP as it can be seen below.

Economic news, which can easily affect economic environment, are important. From a very broad perspective, economic news are examined by Shapiro et al. (2018) who include most of the economic articles published in the newspaper in the period between 1980 and 2015 with SNLP techniques. All semantic inquiries are made to generate an economic semantic index which is prepared with four alternatives. It is observed that current business cycle indicators are highly correlated with economic semantic index scores, furthermore, that index is a good candidate to forecast economic activities that may occur in the near future.

Correa et al. (2017) examine financial stability reports issued by central banks, totalling 982 reports representing 35 countries in the world between 2005 and 2015. The main aim is to provide the FSS index which reflects the change in the financial cyle indicators which are closely bound to asset prices, credits and systemic risk. This research started to create a dictionary on financial stability analysis, however, the words used in the reports were filtered for having a sentiment indicating a positive, negative or neutral connotation.

Final solution of filtering provides 96 positive and 295 negative words that significantly help create the FSS index. After all, in order to examine the sentiment scores in the FSS index, a panel data regression is applied to the frequency of words and topics. In addition to this, a panel VAR approach is also implemented with the FSS index and a financial cycle measure derived from the panel data results. It is found that usability of the FSS index is confirmed while the sentiment scores of central banks predict a deterioration in reports in the beginning of the banking crisis in 2008.

A comprehensive methodological handbook is maintained by Bholat et al. (2015) as statistical text mining by examining several texts collected from Google. The methodologies processed in here are as follows:

* Boolean techniques,
* Dictionary techniques,
* Weighting words,
* Vector space models,
* Latent Semantic Analysis (LSA),
* Latent Dirichlet Allocation,
* Descending Hierarchical Classification,
* Supervised machine learning.

A valuable glossary on text mining and the methodologies covered is also appended to this study while applications of these techniques from several articles are examined for usage of central banks, policy makers and market actors.

Moniz and de Jong (2014) generate an automation system which succesfully forecast effects of central bank communications on interest rate expectations of investors by using the Bank of England's Monetary Policy Committee Minutes, which is so similar to this study. Moniz and de Jong (2014) use aspect detection, polarity detection, naive tone and LDA to catch the semantics of minutes. The results show that investors can use the sentiments of the minutes in order to forecast interest rates.

Since the study directly focuses on the communication issue, Bruno (2017) providing a comrehensive semantic analysis on central banks deserves more attention. It is again analyzed financial stability reports given out by the Bank of Italy, however, the language used is Italian rather than English which has a great advantage on studying SNLP since most of tools are specialized and adapted to the English language. Semantic meanings are established on three twin keywords: stability/instability, expansion/crisis, solidarity/vulnerability. It is found that financial stability reports provide valuables clues from local and global economic environments as in Bruno (2016), which is proven by Latent Dirichlet Analysis, an old school NLP technique different from SNLP.

Another study that uses local language, Kawamura et al. (2016) examine the Bank of Japan's monthly reports within the context of discourse analysis. The main point of the study is to reveal the characteristics of the Bank of Japan and relation to business cycles. However, additional jobs are done such as transparency of the Bank of Japan, sentiment polarity of the words and modality which means auxilary verbs such as should, may, can etc. This research does not stop there, but goes further by examining the relationship of macroeconomic data such as the Tokyo Stock Price Index, consumer price index and providing a forecasting model for future situations. The correlations for them are mostly significant. LDA is also studied in order to look at the relationship between macroeconomic data and expressions, results are so persuasive.

Semantic analyses are also used for predicting market volatility. Zadeh and Zollmann (2009) use the FED minutes to catch semantics in them. In order to do that, Zadeh and Zollmann (2009) use a different approach in text mining such as bag-of-words. They extract n-gram bag-of-words and then generate TFIDF and LOGIP scores derived from frequencies of words. Also, they first use SNLP techniques such as classification for the state of rise or fall of volatility and regression for volatility itself. The results indicate that it is possible to predict in the short-run, in the long-run, SNLP methods fail.

For the period between 2002-2008, Hendry and Madeley (2010) collect communication statements of the Bank of Canada and ask whether there

are affection from these statements to return and its volatility in the short run and to interest rates in the long run. They use latent semantic analysis (LSA) which is a common NLP technique before the statistical ones, and immediately afterwards they apply singular value decomposition. LSA succesfully highlights themes stated in communication statements that also affect market returns and its volatility in the short run when BAX market is also reflected the effects of semantics in statements. In the long run, the statements are able to see future shocks that may hit Canadian economy, risks of balance to economic projections. Hansen and McMahon (2016) examine The Federal Open Market Committee statements in order to test whether CB communications have an effect on economic projections and real sectors for 1994-2012. They found that shocks are the more influencial tool rather than communication tools, in addition to, forward guidance of CB are more effective while communication do not affect any real economic variables.

An interesting article from Ehrmann and Talmi (2019) elaborate whether statements which are issued by the Bank of Canada are similar or not. The similarity of statements actually means changed parts of statements: the more changes exist, the higher volatility is expected. Vice versa, if the market is affected by the statements issued, then less change relatively lead to less volatility in the market. As the results indicate, since changes in the reports are less than expected, namely current press releases are derived from previous one issued, volatiltiy in the market is observed as nearly minimum. An interesting contribution, starting with a blank page in statements is an effective strategy to decrease market volatility.

5.3 Data

In the Central Bank of the Republic of Turkey (CBRT hereafter), since there are no regular minutes or statements issued, Summaries of Monetary Policy Committee Meetings (SMPCM hereafter) are considered as the most frequently used communication tool by the CBRT. SMPCMs[2] are published after the meetings of Monetary Policy Committee which has no regular

[2]These summaries are published in Turkish and English by CBRT, however, this study uses only English versions.

meeting date. However, as it can be understood from the CBRT's official website, they may get together in the fourth Thursday's of the month. After the meeting, SMPCM for this meeting would be published within five business days in accordance with a rule implemented after 2011, so, before this year, there was no regular scheduling to publish SMPCMs.

SMPCMs are issued in CBRT's Official web site in two pages, the first one is under "Main Page - Core Functions - Monetary Policy" [3], the second one is listed under "Main Page - Announcements and Speeches - Press Releases" [4]. Since there is no consistency in terms of sources, that is, some of them are listed as HTML while there is a copy for all reports in PDF file format. Also, in general, there are no HTML reports before 2014. Whence, the most effective solution, downloading all SMPCMs in PDF file format was done with several crawlers written in Python. SMPCMs starting from 2006, when the Monetary Policy Committee convened on 20 July 2006 and the reports released on 31 July 2006 numbered as 2006-39 were collected to the date when the Monetary Policy Committee convened on 7 June 2018 and the reports released on 14 June 2018 numbered as 2018-24. In total, there are 138. All SMPCMs have three chapters:

* Inflation Developments (InD),
* Factors Affecting Inflation (FAI),
* Monetary Policy and Risks (MPR).

These are the first phase.

In the second and hardest phase, downloaded SMPCMs are converted into basic text[5] files by means of a convertor written in Python. This convertor is a parser, analyzer and convertor, called PDFMiner[6]. PDFMiner can succesfully collect corpus with 100% rate, and what's more, if the PDF document was previosuly prepared in a text editor, lossless conversion can be achieved.

[3] http://www.tcmb.gov.tr/wps/wcm/connect/EN/TCMB+EN/Main+Menu/Core+Functions/Monetary+Policy/Monetary+Policy+Committee/

[4] http://www.tcmb.gov.tr/wps/wcm/connect/EN/TCMB+EN/Main+Menu/Announcements/Press+Releases

[5] with txt extension.

[6] https://github.com/pdfminer/pdfminer.six

SMPCMs are those kinds of documents, however, it is a messy one that should be reorganized as a tidy one. Namely, there is no structured corpus in SMPCMs. In order to get clean and tidy corpus with the date of the current SMPCM being processed, unnecessary spaces are removed, “\n” and “\r”[7] by keeping three chapters.

5.4 Research Design and Procedures

The most complex part in this study is methodology. Since it is hard to process text data, the jobs to be done are getting more complicated than a researcher thinks. At first, as peviously stated above, SMPCMs are crawled from CBRT’s official web site as English versions. This is necessary due to the methods explained and used later are compatible mostly English language rather than Turkish.

Descriptive statistics, in this research, are a bit weirder than usual ones. Since this research is closely aimed at running text mining, mined results can be provided for readers in unusual ways. The texts can be represented with words, phrases or sentences. The smallest representations, letters and syllables, are quite pointless, words are accepted as the smallest unit of texts. Then, frequencies are calculated for each chapter [8] and are represented with word clouds as generally use in web sites. The most recurring words are drawn as the largest ones, the least repetitive word is drawn relatively smaller with a different color, and so on.

An enhanced version of descriptive statistics is to calculate n-grams. Cavnar et al. (1994) describe n-gram as ”n-character slice of a longer string”. Here, N stands for number. Although the concept of n-gram is used for slices, in SNLP literature, it is widely used to indicate the number of words sequentially ordered in a sentence. However, it actually relies upon the definition of a token. What token means is the smallest unit of text. It can be regarded from

[7] They are called newline() and carriage-return(̊) escape.

[8]

- Inflation Developments (InD),
- Factors Affecting Inflation (FAI),
- Monetary Policy and Risks (MPR).

letters to passages, it depends on the subject of the research. In this study, since tokens are taken as words, the n-gram definition indicates the number of words that follow each other. So, in general, the maximum number of n-grams can be calculated with the following formula:

$$n-gram_k = X-(n-1) \tag{5.1}$$

where X denotes the number of words in a given sentence K.

In the way of thinking of n-grams, word clouds can easily be regarded as a 1-gram representation of the words in the text. However, in this study, 2-grams (in Chapter D and 3-grams (in Chapter D) are also calculated as well as 1-grams (in Chapter D). However, since, it is impossible to give all n-grams, only top five n-grams are extracted and given in this study as descriptive statistics of SMPCMs. In order to give a picture of the trends in SMPCMs, since they are published from previously published drafts of SMPCMs, the words that are different from previously issued SMPCMs are also extracted.

Later, a semantic tagging procedure is run for the sentences which belong to the three chapters, InD , FAI and MPR respectively. In the tagging procedure, tokens which are words are extracted and categorized according to their role in that sentence such as subjects, verbs etc. In this process, Bird et al. (2009)'s NLTK [9] is used to lemmatize the words. Lemmatization, which is different from stemming which aims to reach the root of the word by eliminaing suffixes and/or prefixes, is the procedure of getting real words that are lexicographically given in dictionaries, namely finding lemmas (headwords, catchwords) as canonical form. Example of lemmatization of a sentence [10] as given below:

* ('A', 'DT'),
* ('revision', 'NN'),
* ('of', 'IN'),

[9] Natural Languate Toolkit for Python.

[10] A revision of the monetary policy stance may be considered, should the fiscal policy deviate significantly from this framework, and consequently have an adverse effect on the medium-term inflation outlook.

* ('the', 'DT'),
* ('monetary', 'JJ'),
* ('policy', 'NN'),
* ('stance', 'NN'),
* ('may', 'MD'),
* ('be', 'VB'),
* ('considered', 'VBN'),
* (',', ','),
* ('should', 'MD'),
* ('the', 'DT'),
* ('fiscal', 'JJ'),
* ('policy', 'NN'),
* ('deviate', 'VBP'),
* ('significantly', 'RB'),
* ('from', 'IN'),
* ('this', 'DT'),
* ('framework', 'NN'),
* (',', ','),
* ('and', 'CC'),
* ('consequently', 'RB'),
* ('have', 'VBP'),
* ('an', 'DT'),
* ('adverse', 'JJ'),
* ('effect', 'NN'),
* ('on', 'IN'),
* ('the', 'DT'),
* ('medium-term', 'JJ'),
* ('inflation', 'NN'),
* ('outlook', 'NN'),
* ('.', '.')

In here, caught or post-tagged lemmas are nouns (NN), adjectives (JJ), verbals (VB* [11]) and adverbs (R*). Rest of them are regarded as unnecessary and/or undeserving points. Then, lemmas are taken as taggers which can be given

[11] Asteriks is for other type of verbal, their initial letter.

points as positive or negative in accordance with the determined sentiment scores by WordNet [12].

Tagging and attaining a sentiment score to each lemma, a sentence is turned into a list of scores. In order to calculate a score for a sentence, the following formula is applied to each of the sentences:

$$Sentiment\ Score_k = \frac{\Sigma_i^n Score_i}{n_k} \tag{5.2}$$

where k is the sentence while n is the number of lemmas post tagged. In the example given below, the sets of n, namely lemmas, are as follows:

Again, the set of n, is as below:

$$\begin{aligned} n = \{&'revision','monetary','policy','stance','be','considered',\\ &'fiscal','policy','deviate','significantly','framework',\\ &'consequently','have','adverse','effect','inflation','outlook'\} \end{aligned}$$

After sentiment evaluation, a vector of score is returned as below:

$$\begin{aligned} Sentiment\ Score\ Vector\ (SSV) = \{&0.0, 0.0, 0.0, -0.0625,\\ &0.009615384615384616, 0.1805,\\ &0.25, 0.0, -0.04164, 0.16, 0.0,\\ &0.0, -0.03289473684210526, -0.1875,\\ &-0.10416, -0.0625, 0.0\} \end{aligned}$$

Since the number of lemmatized words are 17, a division of the sum of *SSV* to 17 gives the sentiment score of the sentence, that is approximately 0.0068006. Afterwards, this process is applied to another level, now, tokens are regarded as passages. After passages, in the third level, it reaches ”Chapter“ level -InD, F.A.I and MPR as tokens- and the process is done for each. Consequently, a general sentiment score for a specific chapter in a specific SMPCM is gathered after three loop processes.

[12] https://wordnet.princeton.edu/

Lastly, in the first step of the loop stated below, a dataset which consists of SMPCM No, date, item number and its chapter stated given under, whole sentence, sentence number and its sentiment score are collected in order to create a forecasting model with recurrent neural networks, which is named as sentence-based monetary policy meeting reports. In this study, the most appropriate network architecture is LSTM [13] created by Gers (1999). Since LSTM is categorized under the Recurrent Neural Network family, it perfectly fits sequential data which the sentences used in SMPCMs are sequentially and semantically ordered to each other. As stated and mentioned in previous chapters which consist of that methodology in a great detail, in general, an LSTM Unit which is part of recurrent neural network family can be represented as follows:

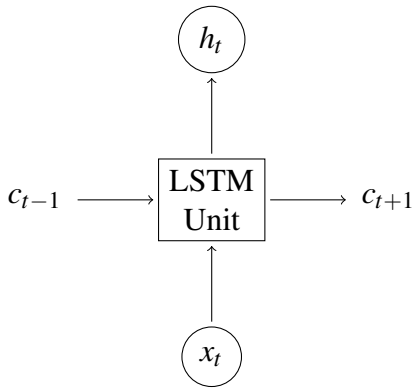


Figure 5.1 : An LSTM Unit.

In Figure 5.1, x_t denotes inputs fed to the neural network, h_t represents outputs obtained after the processes done in LSTM Unit. In the next sequence, x_{t+1} feeds rthe ecurrent neural network and c_{t+1} is injected into the LSTM Unit then it is calculated h_{t+1}. Since this process is recursively conducted, these types of neural networks are classified into the family called recurrent neural networks (RNN hereafter).

Although the RNN family is a newcomer to the machine learning literature (compared to Convolutional NN) and a breakthrough due to its architecture,

[13] Long Short-Term Memory.

LSTM is a special case owing to not only its design but also the processes used. It becomes more understandable and clear if inside of an LSTM Unit is represented as in Figure 5.2. In Equation 5.3, inputs are fed and multiplied with weights, previous hidden units (initially it is zero) and a bias term are added to it, the gate is called a forget gate, the name is given due to its learnable weights may not convey anything to Equation 5.6. This process is done twice to apply activation functions in Equation 5.4 and 5.5 simultaneously. Equation 5.6 calculates the inside of the cell in order to produce the output in Equation 5.7.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{5.3}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{5.4}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{5.5}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{5.6}$$

$$h_t = o_t \circ \sigma_h(c_t) \tag{5.7}$$

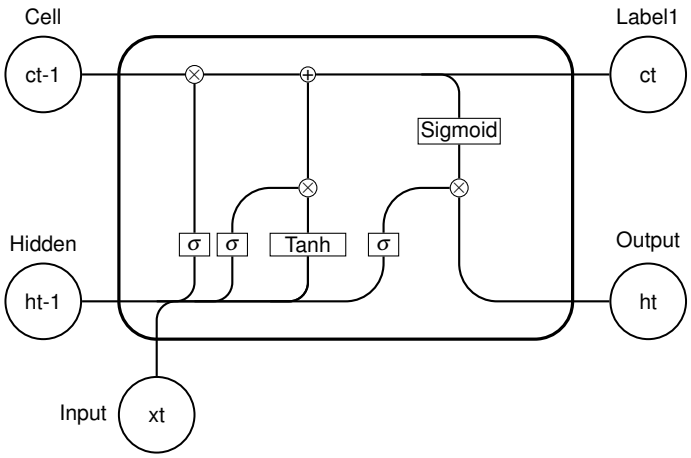


Figure 5.2 : Inside of an LSTM Unit.

After an LSTM is trained with the data from sentence-based monetary policy meeting reports with a deep neural network framework, a forecasting model

is obtained in order to score SMPCM sat the time of issuance. However, instead of taking into account sentiment scores as float, they are classified as positive or negative. It should be negative if sentence sentiment score is less than zero, positive if it is greater than zero. Absolute zeros with regards to 8 decimals are accepted as 'neutral' and these sentences are dropped from data. So, the problem, since there are only labels rather than floats, is becomes a classification problem but a semantic one.

5.5 Results

The maximum number of items, 31, stated in a SMPCM has been observed in Number 2018-12, meeting date 25 April 2018 with the publication date being 30 April 2018. In parallel with economic activity, as it can be seen in Figure 5.5, when economy heats up a bit, the number of items increases as well. When the economy is stable, the number of items is also stabilized between 16-20 items per SMPCM. A significant increase is observed at the beginning of 2018.

The first descriptive results, as mentioned before, are word clouds which are explicitly a representation of 1-grams. However, word clouds are extracted regardless of the year or SMPCM No. bases. As it can obviously be seen in Figure 5.3, 'inflation', 'price' and 'annual' are the most repetitives stated under Inflation Development chapter in the SMPCM. In Figure 5.4, 'Growth', 'recovery', 'demand', 'quarter', 'employment' are the most repetitive words in the Factors Affecting Inflation chapter in SMPCM. For the chapter Monetary Policy and Risks 'inflation', 'policy', 'effect', 'growth', 'fiscal' and 'monetary' are the words which have higher frequencies than other words according to Figure 5.6.

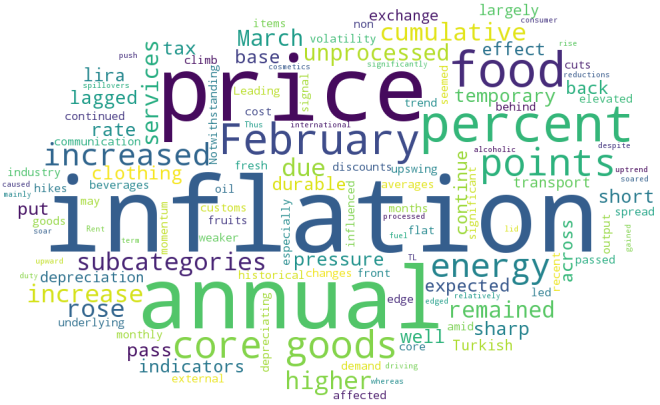


Figure 5.3 : Word Cloud for 'Inflation Developments' Chapter in SMPCMs.

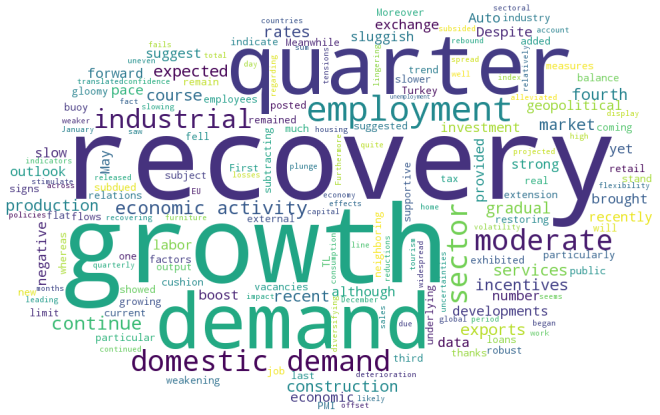


Figure 5.4 : Word Cloud for 'Factors Affecting Inflation' Chapter in SMPCMs.

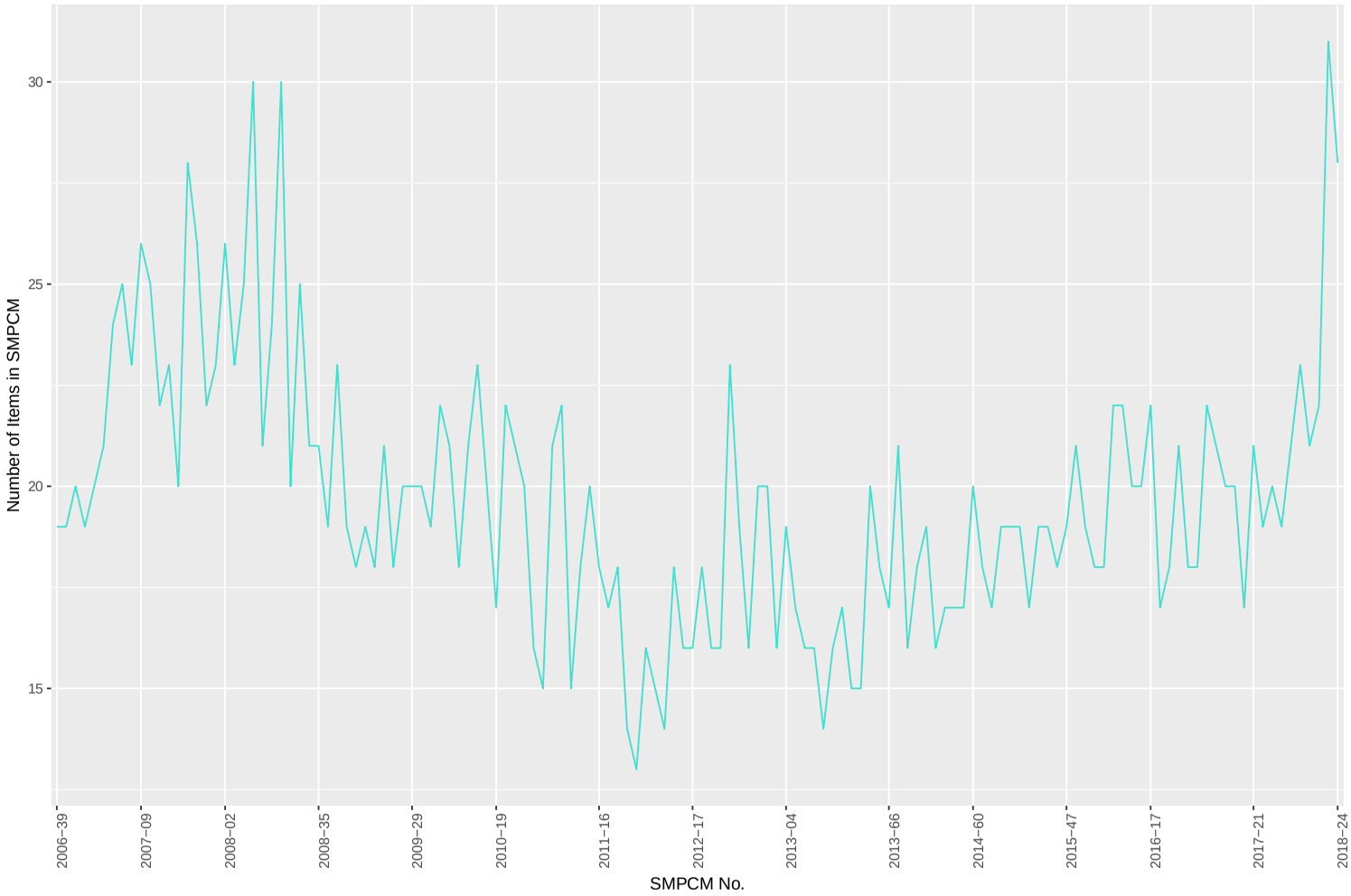


Figure 5.5 : Maximum Number of Items in SMPCMs by No.

Continuing where it is left off, 1-grams can provide a comprehensive descriptive statistics on SMPCMs. In Figure 5.7, 1-grams are extracted according to SMPCM No. Each 1-grams are also classified into their chapters which they belong to. Then, the number of words is calculated for both the chapters and the total. In the crisis period, from the beginning of 2008 [14] to June 2009 [15], a relative increase in the number of words counted in the FAI chapter and in the total can be observed. In the latter periods of the data, this increase is observable again, starting from the late 2015s.

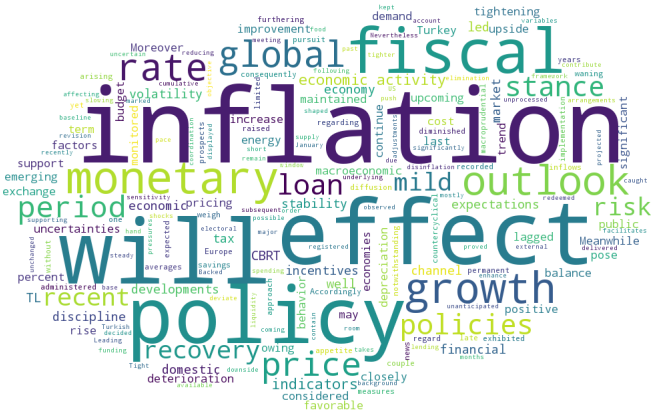


Figure 5.6 : Word Cloud for 'Monetary Policy and Risks'Chapter in SMPCMs.

[14] The last SMPCM in 2007 was issued in December with No.2007-33. The first SMPCM issued in 2008 was numbered 2008-02 in January.

[15] SMPCM issued in June 2009 is numbered as 2009-21.

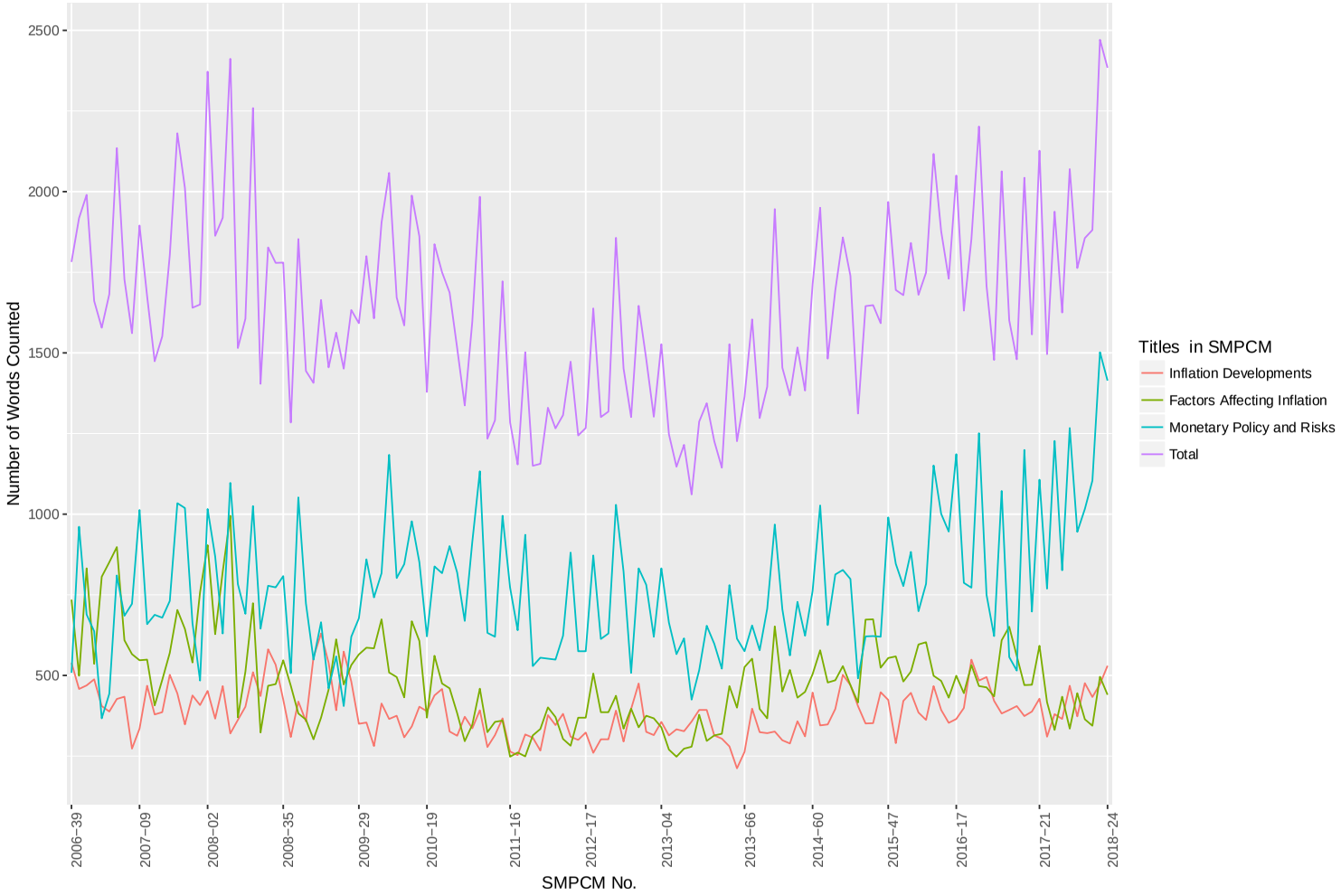


Figure 5.7 : Number of Words counted in SMPCMs by No.

As Ehrmann and Talmi (2019) examined, when the words used in $SMPCM_t$ and $SMPCM_{t-1}$ were compared to each other, it can be observed that there is significant similarity between them. This observation raises a serious suspicion about the preparation of the current issue of the SMPCM. It may be because a previous term's SMPCM was used as a template or emerged from through a high influence previous SMPCM. Since then, the words are compared and unique ones are extracted and drawn against SMPCM No. As drawn in Figure 5.8, the period in between 2006-2011 (SMPCM No.2011-16) and 2015 (SMPCM No.2015-47)-2018 shows a high volatility compared to other periods. Again, these periods have a higher the number of different words used compared to other SMPCMs.

In the second phase of these results, the sentences in items listed under each chapter in the SMPCMS are evaluated semantically according to their sentiment scores. Sentiments are attached to sentences by means of lemmas (namely, lexicographic root of the words) as explained in the methodology. Later, from sentence base, analysis went one step further, to the level of items (like a passage). Each item has its own specific subject that explains an economic issue with or without detail. In the third level, sentiment loop reaches the surface, which the action allows to see the details in an aggregated level, that is, chapters now have their own scores.

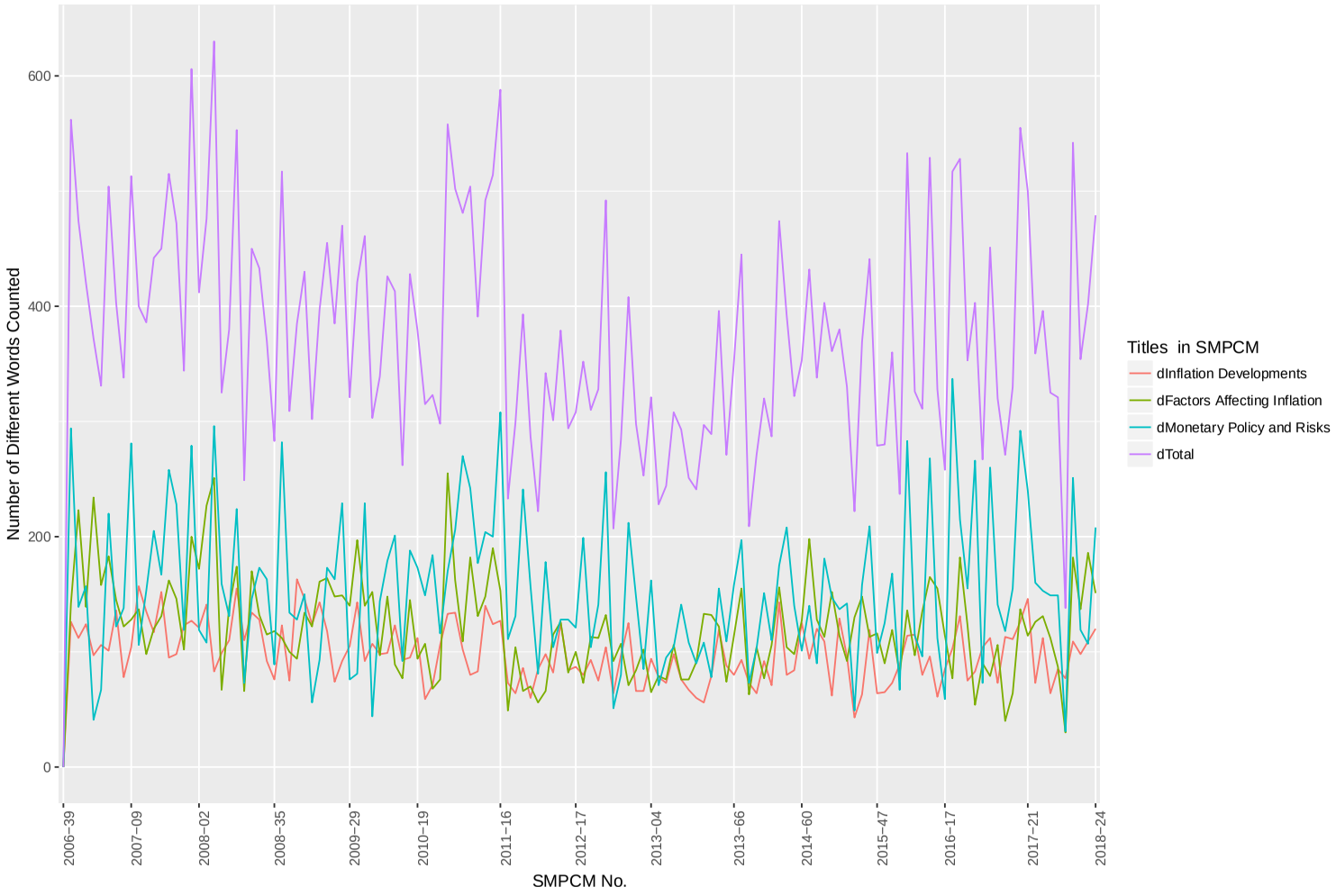


Figure 5.8 : Number of Different Words used in SMPCMs by No.

The scores obtained are represented in four Figures below. Except for the latter one, total scores, each of them represent their own trends along with their SMPCM No. issued. In Figure 5.9, the sentiment score of 'Inflation Developments' chapters in each SMPCM is examined. The band is generally in between -0.1 and +0.1. In SMPCM No. 2007-09 and SMPCM No. 2007-09, it follows a negative course while later than SMPCM No. 2012-17, it generally pursues a positive trends according to its sentiment scores. In Figure 5.10, the chapters of 'Factors Affecting Inflation' are examined and represented with a line graph. It generally provides positive scores except for the period of late 2008 and mid 2009. Since the scores generally follow a negative path rather than other periods, the tone of CBRT there can be described as more pessimistic than other terms. This behaviour is also valid for the chapters of 'Monetary Policy and Risks', again for the same period as it can be seen in Figure 5.11. Except a significant increase which is observed, CBRT is mostly pessimistic during this period. Since the tone of SMPCM is a bit downbeat to the market, it cannot be expected that the market works well.

The picture for the crisis period is a bit clearer in Figure 5.12 that represents the mathematical sum of each of the chapter scores according to their SMPCM No. However, except for that period, after SMPCM No. 2015-47, negative scores are more frequent than positive ones in total. Since the range of scores is volatile between -0.05 and 0.05, the range is than lower than crisis period. Consequently, it cannot be denied that the CBRT has been a bit pessimistic recently.

The picture for the crisis period is a bit clearer in Figure 5.12 that represents the mathematical sum of each of the chapter scores according to their SMPCM No. However, except for that period, after SMPCM No. 2015-47, negative scores are more frequent than positive ones in total. Since the range of scores is volatile between -0.05 and 0.05, the range is than lower than crisis period. Consequently, it cannot be denied that the CBRT has been a bit pessimistic recently.

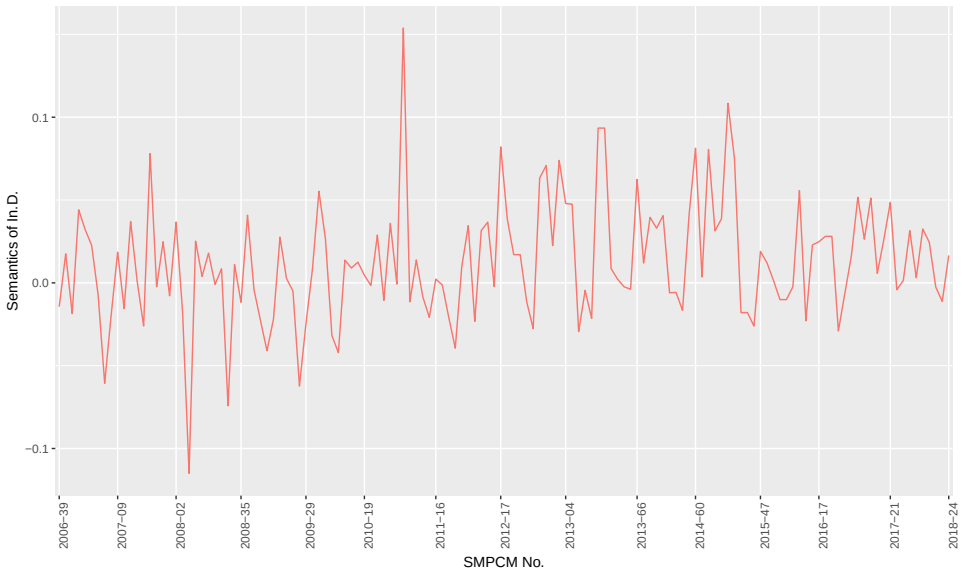


Figure 5.9 : Sentiment Scores of "Inflation Developments" Chapter in SMPCMs.

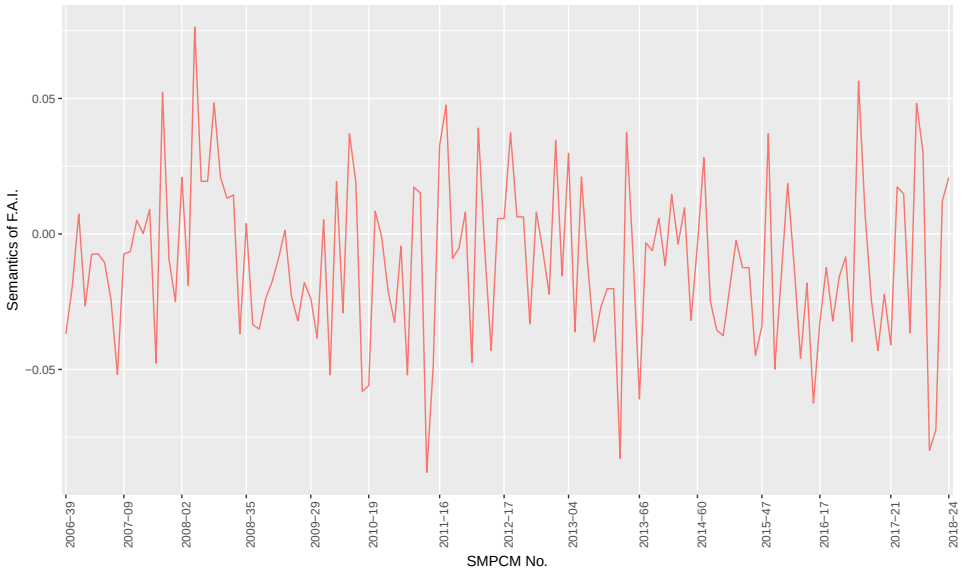


Figure 5.10 : Sentiment Scores of "Factors Affecting Inflation" Chapter in SMPCMs.

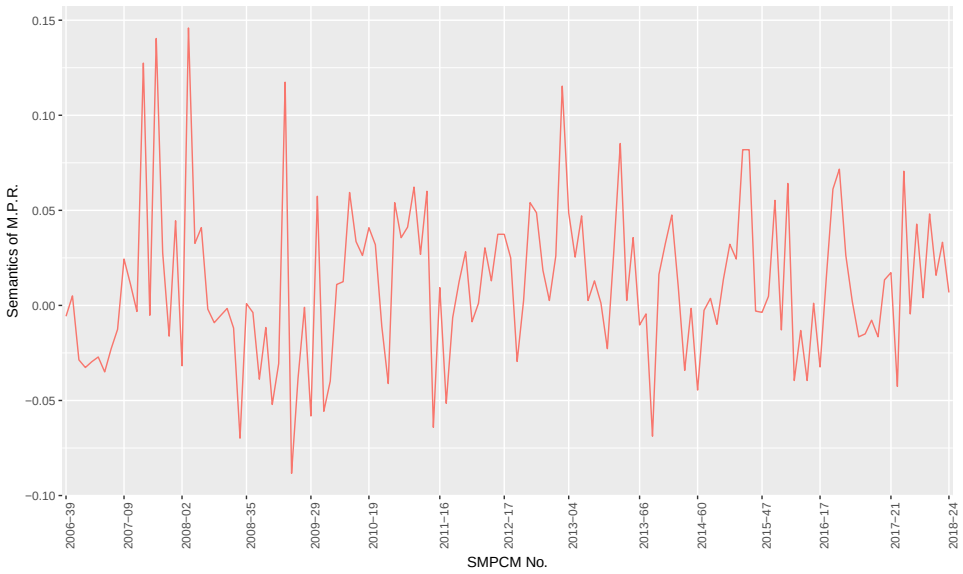


Figure 5.11 : Sentiment Scores of "Monetary Policy and Risks" Chapter in SMPCMs.

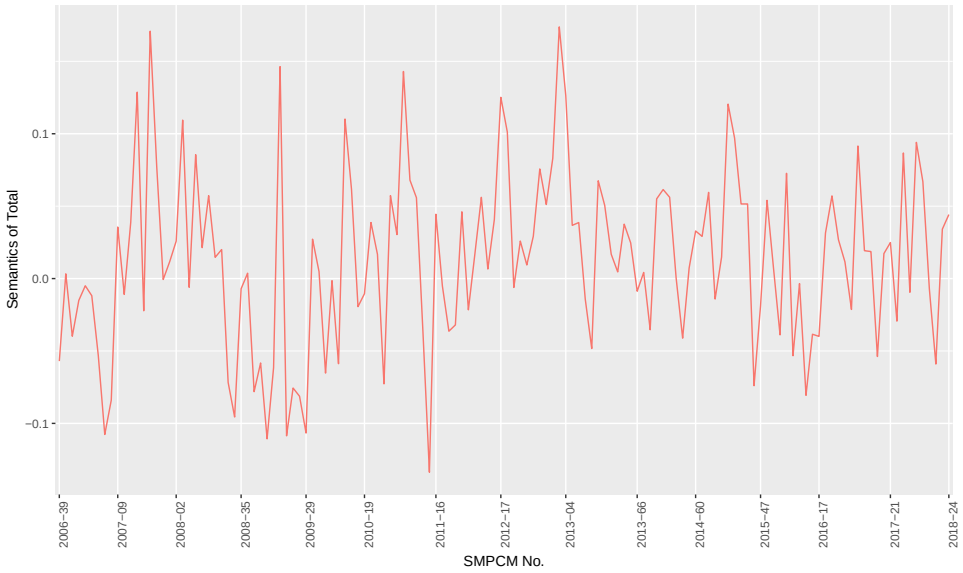


Figure 5.12 : Total Sentiment Scores of SMPCMs.

the LSTM network structure for establishing a forecasting model is done with 10,325 sentences and their sentiment scores as shown in Table 5.1. In general, CBRT's tone is relatively positive as it can be understood from the mean of sentence sentiments. From that sentence-based data, an LSTM model with 254,800 parameters is run under 7 epochs, as structured in Table 5.2.

Table 5.1 : Descriptive Statistics for Sentence-based Sentiments of SMPCMs Data.

Statistics	Value
Count	10361
Mean	0.007635
Std	0.039492
Min	-0.218750
25 %	-0.015086
Median	0.006801
75 %	0.030220
Max	0.250000

In order to converge it faster, minibatches are set to 32, 90% of the data is used as training set, while 10% is used as a test set. The final activation function is determined with softmax in order to get prediction probabilities positive or negative. As mentioned in the methodology section, the dataset consists only of labels, positive / negative labels. Due to this, LSTM is run to solve a classification problem which mostly requires softmax. Instead of using regularization, dropout techniques are used to get rid of gradient problems. With the 'Adam Optimizer', loss function is determined with 'Categorical Cross Entrophy'.

Table 5.2 : LSTM Model Structure.

Layer (type)	Output Shape	Param #
embedding_7 (Embedding)	(None, 102, 128)	256000
spatial_dropout1d_7	(None, 102, 128)	0
lstm_7 (LSTM)	(None, 196)	254800
dense_7 (Dense)	(None, 2)	394
Total params:	511,194	
Trainable params:	511,194	
Nontrainable params:	0	

Training after 7 epochs with a Keras framework under Python, LSTM loss is 0.19 and total accuracy is 93 % which is relatively high for an SNLP model.

When accuracy is split into positive and negative labels, LSTM can accurately predict 51.9% of positive labels and 30.4% negative labels.

5.6 Concluding Remarks

This study attempts to apply statistical natural language processing (SNLP) tools to estimate semantics of summary of monetary policy committee meeting (SMPCM) reports issued by the Central Bank of the Republic of Turkey and proves that information retrieval and machine learning techniques can be used as an estimation technique in economics as well as other fields.

In general, except for the period after 2017, SMPCMs are published by the CBRT monthly in both Turkish and English. Each SMPCM whose English version is used in here has three chapters which are titled 'Inflation Developments' (InD), 'Factors Affecting Inflation' (FAI) and 'Monetary Policy and Risks' (MPR). A crawler was written to get SMPCM which are issued in both HTML and PDF format and then they are converted into corpora according to their publication number and chapter. In order to understand each corpus that turned into a big corpus by appending, item numbers, the number of words and the number of unique words compared to previous SMPCMs are calculated. The results firstly indicate that CBRT uses more words, uses more items during the periods of business cycles.

By using WordNet, the lemmas in each sentence in each corpora are tagged semantically to estimate the tone of CBRT and the meanings of reports. Owing to SNLP tools, sentences are scored between -1 meaning completely negative and +1 meaning completely positive, later, chapter-based generalized sentiment score vectors (SSV) are estimated. As results proved, sentiments in chapters FAI and MPR are more sensible and accurately reflect the economic situation and the message transmitted to the market as well as total sentiment scores according to SMPCM No.

This research goes further by not stopping here. In order to generalize the semantic scores by using the sentence-based SMPCM data, a deep learning algorithm called Long Short-Term Memory (LSTM) model which is a more appropriate tool where the data is sequential or similar to a time series, is

developed to forecast the tone of the SMPCM which may be issues in the near future. After fitting LSTM with 10,361 sentences, the accuracy of it reaches 93% which is quite fascinating in text mining, where the results can be gotten after 7 epochs with 0.19 loss which stays stable and could not be minimized with more epochs.

The model established in this study can be used to semantically understand SMPCM when they are issued by economic agents, policy makers and individuals. Besides, it provides a quantitative measure for a text that can be understood and interpreted in many ways what can change from individual to individual, so, it presents a way to standardize the meaning of it. The techniques used here can be expanded into other areas where the text allows it to be processed.

5.7 Limitations & Further Studies

Frankly, there is a long way to go further in both SNLP and text mining. Semantic scoring, clustering and classification algorithms have been on the path to increase accuracy and the actual meaning in SNLP. Text mining, on the other hand, generates new tools to adapt other languages than English and embraces machine learning techniques as new estimation methods.

Rather than SNLP, this study is restricted with a translated text. This is actually a must, since Turkish, the original language of the SMPCMs, is not well-adapted to SNLP literature at the time of this study. The main complaint about the Turkish language is that it is hard to study with it in SNLP. Since Turkish is a language mostly made up with suffixes; lemmas and roots are not reachable with ease. A second complaint is about technical issues. Since there are no a finalized treebank, parser tool, morphological analyzer etc., the tools provided to use are limited.

Second, the hidden and explicit meanings in sentences can be easily missed if it is translated from its native language to other one. So, any SNLP study faces missing meaning when it studies with a translated corpus. Therefore, in this study, estimation power of the meaning in corpora is a bit limited and not stable, however as the accuracy shows, this limitation can be ignored.

Future studies that may want to focus on semantic analysis of economic text by using information retrieval, machine learning or other SNLP tools may generate their own dictionary at first. Besides, since there is a kind of dictionary [16] prepared by Loughran and McDonald (2011) tailored especially for economic, financial and accounting contexts, it still needs to be more generalized for other words. However, it can be usable for researchers who should mix other dictionaries with Loughran and McDonald (2011)'s dictionary in order to eliminate conflicts.

[16] https://sraf.nd.edu/textual-analysis/resources/

6. CONCLUSION

In this dissertation, three experimental studies are carried out to introduce new estimation methods and their applications to economics. All of the new methods mentioned here are widely used in the field of AI which was advanced at the time this dissertation was written. Of course, this study presents not only methodological methods but also new methods in terms of data acquisition, processing and integration. Due to this, this dissertation contains many innovations in economics in terms of creation and use of big data, high dimensional data, text data and compiling texts to obtain complex but tidy data and make them quantitatively understand.

This dissertation starts with a methodology section that includes all the estimation methods used here and can be perceived as an introduction, followed by three empirical articles, so four parts in total. The methodology section also includes a comprehensive glossary and a detailed mathematical description of the methods used and visualizations at the functional level. In this sense, it is intended to be used as a guide for further studies.

The first chapter, which can be considered as an introduction to new estimation methods, can be separated into two parts: the first section including machine learning, deep learning and a special architecture named "long short term memory" in deep learning literature, followed by the second section explaining econometric methods such as normal equation, logistic and quantile regression. While the econometric section, which contains more references in terms of bibliography, is kept relatively short; the first section, on the contrary, is presented in more detail since this section is considered as less known in economics. In the first section, machine learning algorithms are explained briefly, however, the core elements in machine learning literature such as cost function, slope algorithm, regression and classification problems, neural networks, learning algorithm, forward and

backward methods, backpropagation, recurrent neural networks and activation functions are examined in detail.

In the articles where the abovementioned methods are applied, firstly econometric and basic classification machine learning methods are estimated to compute transition probabilities from unemployment to employment and compared to each other. In the second article, savings rates were then estimated by econometric and deep learning methods similar to the first article. Finally, in the third, the Central Bank of the Republic of Turkey's (CBRT) Monetary Policy Committee Meeting Reports were examined by using recurrent neural networks.

In the first article that examined the transition from employment to unemployment, some basic characteristics of individuals, educational and experiential data, primarily carried out between the years 2004-2016 Turkey Statistical Institute (TURKSTAT) published the Household Labor Force Surveys (HLFS) were compiled. However, as careful review necessitates both semantic and logical matching of questions and answers over the course of years, a manual merge has been performed and the seven-million individual survey data has been coded into one large data frame. Of these, the data that does not contain the slightest hint about the employment information of the individuals in the current period were separated and only the data set of the individuals who were accepted as unemployed in the previous period was limited. So, the data of 203891 individuals were processed. The data is shuffled and layered by Stratified K-Fold algorithm on year basis. Then the data was splitted into to parts for analysis of machine learning classifiers: The training dataset used for ML is limited to 90% and, test data was determined by the remainder. Logistic regression estimations in econometric methodology for estimation of transition probabilities have been realized against the classification algorithms of machine learning. The classification algorithms XGBoost and Random Forest, which produce successful estimates of 67%, have been the best methods to resolve this problem by passing the logistic regression method which provides 63% performance. In this way, any policy maker, economic actor or researcher who has provided information

about the basic characteristics of individuals presented and given in the study can estimate the probability of an unemployed person moving into employment in the next period by 67%.

In recent years, as seen in many of the developing countries, Turkey's saving rate that is one of the most important issues are examined in the second study. The Household Budget Surveys (HBAs) published by TURKSTAT between 2002 and 2016 have been compiled to create a big dataset by taking into account their questions and answers. While twinning was carried out on an individual basis, household data were repeated in the rows of these individuals. Since the questions and answers differed over the years, a manual merge was performed because the data required both semantic and logical matching. By coding this, around seven million individual survey data have been combined under one big data framework. All the data has been tried to be protected as much as possible and a large data size has been obtained with 247 attributes in 629372 lines. Some of these attributes have been finalized with 203 attributes in 371613 lines by subtracting them because of high correlation among variables and derivation of some new variables. Any econometric method used to process data of this magnitude is likely to be interlocked at the stage of inverse of the variance-covariance matrix. However, for comparison, quantile regression method was estimated. Alternatively, a four-layered neural network model has been established by developing a deep learning method based on a quantile regression loss function which run in shorter time and with a very high precise in comparison to quantile regression although the data contains so many attributes of this magnitude. For the deep learning application, the training dataset was determined as 90% by a method similar to the one in the first article, and the rest was used as a test dataset with the Stratified K-Fold randomization algorithm. While the deep learning algorithm based on the quantitative regression loss function, which is run separately for the total number of four (.10, .20, .40 and .80) quantiles, can be trained in small batches of 10,000 in approximately 2 hours and 34 minutes while using the normal equation method for quantile regression was able to produce results in approximately 20 hours. However, when the results are

examined, while the deep learning algorithm can achieve 93% performance, the normal equation quantile regression which does not exceed 15% is well below forecasting performance. All models were also repeated with feature selection, but high performance was not achieved. Deep learning method based on quantile regression can be presented as a good tool to policy makers, researchers and academicians who want to predict future saving rates for individuals and households both in terms of their high performance and in a short time.

Until the period of realization of this dissertation, one of very limited number of research topics in Economics is text-based methods. It remains extremely limited number of articles in the field, moreover, the absence of a known work in Turkey is an important gap for the literature. Therefore, a third article has been realized in order to cover both this gap in the literature and to provide an example that statistical natural language processing methods which may be so effective to analyze text-based data are used. The CBRT's policy-making role, signaling with both written and verbal actions, as well as the country's prestige in terms of its value, has maintained the diligence in these reports and shared with the public for transparency. For that purpose, Summaries of Monetary Policy Committee Meeting (SMPCM) generally published monthly by the Central Bank of the Republic of Turkey were examined in English. These texts are published within five working days of Monetary Policy Committee (PPK) Meetings (starting from 2011), and are held at 14:00 on the 4th Thursday of the current month unless there is an emergent economic event. Instead of the full text of the meetings, these summary reports are presented as a PDF on the CBRT website under three main headings. These three headings are generally listed as "Inflation Developments" (1), "Factors Affecting Inflation" (2) and "Monetary Policy and Risks (3). In the first chapter, the changes in the inflation in the previous period are examined, in the second chapter, the factors affecting the changes in the past period are examined in detail. In the last chapter, the developments in macro conjuncture are summarized.It is seen that these technical reports are generally prepared in a systematic manner and there are no serious changes except for the small changes in the titles. In addition, it

has been found that the paragraphs in each report are given in such a way that they do not change the integrity of the subject, the sentences processed in the articles do not come out of the subject integrity of the article, and the items under the three headings are prepared with the utmost loyalty. Accordingly, a total of 129 SMPCM, between the first summary report published on 20 July 2006 and numbered 2006-39 and the last summary report 2018-24 published on 14 June 2018, were crawled from the CBRT official web page with a Python crawler. Then they has been converted to plain text files to process them at ease. The data were then listed under headings and numbers, and the most commonly used words were subtracted under 1-gram, 2-gram and 3-gram as a time series, and 1-grams were illustrated as word cloud for each of the three titles. Accordingly "prices" under the (1) title, demand under the (2) title, and inflation under the (3) title, are the most commonly used 1-grams. Then, the numbers of items and the number of words used in each SMPCM were drawn in a time-serie graph, and it was shown that the CBRT used more intensive words during the business cycles and written more items in the reports. Afterwards, semantic analysis was performed for each sentence using statistical natural language processing methods byu using WordNet database, and cumulative index values were created on the basis of title or report. In this sense, it was observed that the CBRT spoke in a more pessimistic tone and addressed the market during Turkey's business cycles. The process is totally named as quantification and indexing of SMPCMs. After this indexing, the database was reduced to sentence size, the reports were connected to each other by time series approach and a recurrent neural network model called Long Short-Term Memory was established. With this model, although it is very useful for estimating the tone given to the market, a robust model was developed for predicting the meaning scores of future reports with an accuracy of 93%.

Ultimately, all stages of this thesis, which includes and achieves significant innovations both in terms of implementation and the difficulties of creating its own dataset, have been carried out as open source to be presented to all interested researchers, market and policy makers in the discipline of

economics. Since the datasets in the first two articles can only be provided by TURKSTAT, the dataset of the third study and all the codes used in the abovementioned three articles are presented for future studies.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., ..., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abdulkadiroğlu, A. and Sönmez, T. (1998). Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3):689–701.

Abdulkadiroglu, A. and Sönmez, T. (2013). Matching markets: Theory and practice. *Advances in Economics and Econometrics*, 1:3–47.

Acemoglu, D. (2002). Technical change, inequality, and the labor market. *Journal of economic literature*, 40(1):7–72.

Acemoglu, D. and Autor, D. (2011). Lectures in labor economics. *Manuscript. http://economics. mit. edu/files/4689*, page 22.

Acemoglu, D. and Shimer, R. (1999). Holdups and efficiency with search frictions. *International Economic Review*, 40(4):827–849.

Ackley, G. (1951). Reviewed work: Income, saving, and the theory of consumer behavior by james s. duesenberry. *The Review of Economics and Statistics*, 33(3):255–257.

Aghdam, H. H. and Heravi, E. J. (2017). Guide to convolutional neural networks. *New York, NY: Springer. doi*, 10:978–3.

Agostinelli, F., Hoffman, M., Sadowski, P., and Baldi, P. (2014). Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*.

Agrawal, A., Gans, J., and Goldfarb, A. (2018). *Prediction Machines: The simple economics of artificial intelligence*. Harvard Business Press.

Aizenman, J., Pinto, B., and Radziwill, A. (2004). Sources for financing domestic capital – is foreign saving a viable option for developing

countries? Working Paper 10624, National Bureau of Economic Research.

Alkin, E. (2016). The problematic of savings: The case of turkey as an emerging market economy. *Aurum Sosyal Bilimler Dergisi*, 1(1):55–71.

Alquezar, R. and Sanfeliu, A. (1994). A hybrid connectionist-symbolic approach to regular grammatical inference based on neural learning and hierarchical clustering. In *Grammatical Inference and Applications*, pages 203–211. Springer Berlin Heidelberg.

Andersson, M., Dillén, H., Sellin, P., et al. (2002). Monetary policy signalling and movements in the swedish term structure of interest rates. *Developments driving changes in market functioning*, page 268.

Ando, A. and Modigliani, F. (1963). The “life cycle" hypothesis of saving: Aggregate implications and tests. *The American economic review*, 53(1):55–84.

Argyle, M. and Furnham, A. (2013). *The psychology of money*. Routledge.

Ashenfelter, O., Harmon, C., and Oosterbeek, H. (1999). A review of estimates of the schooling/earnings relationship, with tests for publication bias. *Labour economics*, 6(4):453–470.

Athey, S. (2017). Beyond prediction: Using big data for policy problems. *Science*, 355(6324):483–485.

Athey, S. (2018). The impact of machine learning on economics. In *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press.

Athey, S. and Imbens, G. (2015). A measure of robustness to misspecification. *American Economic Review*, 105(5):476–80.

Athey, S. and Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, 31(2):3–32.

Attanasio, O. P. (1993). An analysis of life-cycle accumulation of financial assets. *Ricerche Economiche*, 47(4):323–354.

Avery, R. B. and Kennickell, A. B. (1991). Household saving in the us. *Review of Income and Wealth*, 37(4):409–432.

Axell, B. (1977). Search market equilibrium. *The Scandinavian Journal of Economics*, pages 20–40.

Banks, J., Blundell, R., and Tanner, S. (1998). Is there a retirement-savings puzzle? *American Economic Review*, pages 769–788.

Barwise, J. (1982). *Handbook of mathematical logic*, volume 90. Elsevier.

Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.

Becker, G. (1964). *Human capital: a theoretical and empirical analysis, with special reference to education*. National bureau of economic research publications: General series. National Bureau of Economic Research; distributed by Columbia University Press.

Begenau, J., Farboodi, M., and Veldkamp, L. (2018). Big data in finance and the growth of large firms. *Journal of Monetary Economics*.

Belloni, A., Chen, D., Chernozhukov, V., and Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6):2369–2429.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Bholat, D. M., Hansen, S., Santos, P. M., and Schonhardt-Bailey, C. (2015). Text mining for central banks. *SSRN Electronic Journal*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Björkegren, D. and Grissen, D. (2017). Behavior revealed in mobile phone usage predicts loan repayment. *arXiv preprint arXiv:1712.05840*.

Blau, D. M. and Robins, P. K. (1990). Job search outcomes for the employed and unemployed. *Journal of Political Economy*, 98(3):637–655.

Blinder, A. S. (2004). The quiet revolution. *Central Banking goes modern*, page 81.

Blinder, A. S., Ehrmann, M., Fratzscher, M., De Haan, J., and Jansen, D.-J. (2008). Central bank communication and monetary policy: A survey of theory and evidence. *Journal of Economic Literature*, 46(4):910–45.

Blundell, R., Dearden, L., Meghir, C., and Sianesi, B. (1999). Human capital investment: the returns from education and training to the individual, the firm and the economy. *Fiscal studies*, 20(1):1–23.

Bodkin, R. (1959). Windfall income and consumption. *The American Economic Review*, 49(4):602–614.

Bornemann, A. and Duesenberry, J. S. (1950). Income, saving, and the theory of consumer behavior. *Southern Economic Journal*, 16(4):485.

Bose, B. K. (2007). Neural network applications in power electronics and motor drives—an introduction and perspective. *IEEE Transactions on Industrial Electronics*, 54(1):14–33.

Bosworth, B., Burtless, G., Sabelhaus, J., Poterba, J. M., and Summers, L. H. (1991). The decline in saving: Evidence from household surveys. *Brookings Papers on Economic Activity*, 1991(1):183–256.

Bougheas, S. and Georgellis, Y. (2004). Early career mobility and earnings profiles of german apprentices: Theory and empirical evidence. *Labour*, 18(2):233–263.

Boukus, E. and Rosenberg, J. V. (2006). The information content of fomc minutes. *SSRN Electronic Journal*.

Bowles, S., Gintis, H., and Osborne, M. (2001). The determinants of earnings: A behavioral approach. *Journal of economic literature*, 39(4):1137–1176.

Bracewell, R. N. and Bracewell, R. N. (1986). *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York.

Brady, D. S. and Friedman, R. D. (1947). Savings and the income distribution. In *Studies in income and wealth*, pages 247–265. NBER.

Brand, C., Buncic, D., and Turunen, J. (2010). The impact of ecb monetary policy decisions and communication on the yield curve. *Journal of the European Economic Association*, 8(6):1266–1298.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L. and Spector, P. (1992). Submodel selection and evaluation in regression. the x-random case. *International statistical review/revue internationale de Statistique*, pages 291–319.

Bresson, R. (2016). *Notes on the Cinematograph*. New York Review of Books.

Browning, M. and Crossley, T. F. (2001). The life-cycle model of consumption and saving. *Journal of Economic Perspectives*, 15(3):3–22.

Browning, M. and Lusardi, A. (1996). Household saving: Micro theories and micro facts. *Journal of Economic literature*, 34(4):1797–1855.

Bruno, G. (2016). Text mining and sentiment extraction in central bank documents. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 1700–1708. IEEE.

Bruno, G. (2017). Central bank communications: Information extraction and semantic analysis. In *The R User Conference, useR! 2017 July 4-7 2017 Brussels, Belgium*, page 253.

Buchanan, J. M. (1987). *Economics: between predictive science and moral philosophy*. Number 7. Texas A&M University Press.

Bulutay, T. (1992). A general framework for employment in turkey. *State Institute of Statistics*, page 223.

Burdett, K. and Judd, K. L. (1983). Equilibrium price dispersion. *Econometrica: Journal of the Econometric Society*, pages 955–969.

Calvó-Armengol, A. (2004). Job contact networks. *Journal of economic Theory*, 115(1):191–206.

Carroll, C. D., Overland, J., and Weil, D. N. (2000). Saving and growth with habit formation. *American Economic Review*, 90(3):341–355.

Carroll, C. D. and Summers, L. H. (1991). Consumption growth parallels income growth: Some new evidence. In *National saving and economic performance*, pages 305–348. University of Chicago Press.

Carroll, C. D. and Weil, D. N. (1994). Saving and growth: a reinterpretation. In *Carnegie-Rochester conference series on public policy*, volume 40, pages 133–192. Elsevier.

Cavnar, W. B., Trenkle, J. M., et al. (1994). N-gram-based text categorization. *Ann arbor mi*, 48113(2):161–175.

Celasun, M. and Tansel, A. (1993). Distributional effects and saving-investment behaviour in a liberalizing economy: the case of turkey. *METU Studies in Development*, 20(3):269–298.

Ceritoğlu, E. (2013). The impact of labour income risk on household saving decisions in turkey. *Review of Economics of the Household*, 11(1):109–129.

Ceritoglu, E. and Eren, O. (2014). The effects of demographic and social changes on household savings in turkey. *Central Bank Review*, 14(3):15.

Çeşmelioğlu, A., Meidl, W., and Pott, A. (2016). There are infinitely many bent functions for which the dual is not bent. *IEEE Transactions on Information Theory*, 62(9):5204–5208.

Chandler, D., Levitt, S. D., and List, J. A. (2011). Predicting and preventing shootings among at-risk youth. *American Economic Review*, 101(3):288–92.

Chang, Y. R. and Huston, S. J. (1995). Patterns of adequate household emergency fund holdings: A comparison of households in 1983 and 1986. *Journal of Financial Counseling and Planning*, 6:119.

Chen, C. and DeVaney, S. A. (2001). The effect of life cycle stages and saving motives on the probability of emergency fund adequacy. In *Proceedings of the Association for Financial Counseling and Planning Education*, pages 176–185.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference*

on knowledge discovery and data mining, pages 785–794. ACM.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., and Newey, W. (2017). Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, 107(5):261–65.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., and Shelhamer, E. (2014). cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*.

Cho, G. L. (1996). Search, unemployment, and the business cycle: A multisectoral dynamic general equilibrium model. *Unpublished manuscript*.

Chollet, F. (2015). Keras. *GitHub repository*.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Cilasun, M. (2009). Income and, consumption and saving behavior of turkish household. *Unpublished doctoral dissertation, Middle East Technical University*.

Cilasun, S. M., Kırdar, M. G., et al. (2009). Türkiye'de hanehalklarının gelir, tüketim ve tasarruf davranışlarının yatay kesitlerle bir analizi. *Iktisat Isletme ve Finans*, 24(280):9–46.

Cingano, F. (2003). Returns to specific skills in industrial districts. *Labour Economics*, 10(2):149–164.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cohn, E., Kiker, B. F., and De Oliveira, M. M. (1987). Further evidence on the screening hypothesis. *Economics Letters*, 25(3):289–294.

Collins, N. (2004). Kid a, and: Amnesiac, and: Hail to the thief. *Computer Music Journal*, 28(1):73–77.

Correa, R., Garud, K., Londono, J. M., and Mislang, N. (2017). Sentiment in central banks' financial stability reports. *Available at SSRN 3091943*.

Cox, D. R. (1992). Regression models and life-tables. In *Breakthroughs in statistics*, pages 527–541. Springer.

Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM.

Davidson, J. E., Hendry, D. F., Srba, F., and Yeo, S. (1978). Econometric modelling of the aggregate time-series relationship between consumers' expenditure and income in the united kingdom. *The Economic Journal*, pages 661–692.

Dearden, L. (1998). Ability, families, education and earnings in britain. Technical report, IFS Working Papers.

Delorme, R. et al. (2010). Deep complexity and the social sciences. *Books*.

Dewachter, H., Erdemlioglu, D., Gnabo, J.-Y., and Lecourt, C. (2014). The intra-day impact of communication on euro-dollar volatility and jumps. *Journal of International Money and Finance*, 43:131–154.

Diamond, P. A. (1971). A model of price adjustment. *Journal of economic theory*, 3(2):156–168.

Diamond, P. A. (1981). Mobility costs, frictional unemployment, and efficiency. *Journal of political Economy*, 89(4):798–812.

Diamond, P. A. (1982a). Aggregate demand management in search equilibrium. *Journal of political Economy*, 90(5):881–894.

Diamond, P. A. (1982b). Wage determination and efficiency in search equilibrium. *The Review of Economic Studies*, 49(2):217–227.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157.

Donaldson, D. and Eaton, B. C. (1976). Firm-specific human capital: a shared investment or optimal entrapment? *Canadian Journal of Economics*, pages 462–472.

Dubé, J.-P. and Misra, S. (2017). Scalable price targeting. Technical report, National Bureau of Economic Research.

Duygan, B. et al. (2005). Aggregate shocks, idiosyncratic risk, and durable goods purchases: Evidence from turkey's 1994 financial crisis. *EUI Finance and Consumption Program Working Paper*.

Dynan, K. E. and Maki, D. M. (2001). Does stock market wealth matter for consumption? *SSRN Electronic Journal*.

Egami, N., Fong, C. J., Grimmer, J., Roberts, M. E., and Stewart, B. M. (2018). How to make causal inferences using texts. *arXiv preprint arXiv:1802.02163*.

Ehrmann, M. and Fratzscher, M. (2007). Communication by central bank committee members: different strategies, same effectiveness? *Journal of Money, Credit and Banking*, 39(2-3):509–541.

Ehrmann, M. and Talmi, J. (2019). Starting from a blank page? semantic similarity in central bank communication and market volatility. *Journal of Monetary Economics*.

Elliott, D. L. (1993). A better activation function for artificial neural networks. *CiteseerX*.

Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276.

Fisher, I. (1930). The theory of interest. *New York*, 43.

Fitzenberger, B., Koenker, R., and Machado, J. A. (2013). *Economic applications of quantile regression*. Springer Science & Business Media.

Flavin, M. A. (1981). The adjustment of consumption to changing expectations about future income. *Journal of political economy*, 89(5):974–1009.

Foresee, F. D. and Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. In *Neural networks, 1997., international conference on*, volume 3, pages 1930–1935. IEEE.

Fountain, C. (2005). Finding a job in the internet age. *Social Forces*, 83(3):1235–1262.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Friedman, M. (1953). *The methodology of positive economics (1953)*. Cambridge University Press.

Friedman, M. (1957). The permanent income hypothesis. In *A theory of the consumption function*, pages 20–37. Princeton University Press.

Gale, D. and Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15.

Galindo, J. and Tamayo, P. (2000). Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. *Computational Economics*, 15(1-2):107–143.

Geeraerts, D. (2010). *Theories of lexical semantics*. Oxford University Press.

Geraci, M. and Bottai, M. (2014). Linear quantile mixed models. *Statistics and computing*, 24(3):461–479.

Gers, F. (1999). Learning to forget: continual prediction with LSTM.

Giannakouros, P. and Chen, L. (2015). A problem-solving approach to data analysis for economics. volume 47, pages 87–114. Informa UK Limited.

Giannone, D., Lenza, M., and Primiceri, G. E. (2018). Economic predictions with big data: The illusion of sparsity. *SSRN Electronic Journal*.

Glaeser, E. L., Kominers, S. D., Luca, M., and Naik, N. (2018). Big data and big cities: The promises and limitations of improved measures of urban life. *Economic Inquiry*, 56(1):114–137.

Glick, P. C. (1947). The family cycle. *American sociological review*, 12(2):164–174.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Godfrey, L. B. and Gashler, M. S. (2015). A continuum among logarithmic, linear, and exponential functions, and its potential to improve

generalization in neural networks. In *Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), 2015 7th International Joint Conference on*, volume 1, pages 481–486. IEEE.

Goldsmith, R. W. and Salant, W. (1939). The volume and components of saving in the united states 1933-1937. In *Studies in Income and Wealth, Volume 3*, pages 215–316. NBER.

Gomes, G. S. d. S. and Ludermir, T. B. (2008). Complementary log-log and probit: activation functions implemented in artificial neural networks. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, pages 939–942. IEEE.

Gomes, J., Greenwood, J., and Rebelo, S. (2001). Equilibrium unemployment. *Journal of Monetary Economics*, 48(1):109–152.

González, W. J. (1998). Prediction and prescription in economics: A philosophical and methodological approach. *Theoria: An International Journal for Theory, History and Foundations of Science*, pages 321–345.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. *arXiv preprint arXiv:1302.4389*.

Gordon, W. T. (1985). A history of semantics.

Gordon, W. T. (2003). *Correlational Analysis and Mechanical Translation*. The MIT Press.

Greene, W. H. (2003). *Econometric analysis*. Pearson Education India.

Groot, W. and Oosterbeek, H. (1994). Earnings effects of different components of schooling; human capital versus screening. *The review of Economics and Statistics*, pages 317–321.

Gujarati, D. N. (2009). *Basic econometrics*. Tata McGraw-Hill Education.

Gupta, K. L. (1970). On some determinants of rural and urban household saving behaviour. *Economic Record*, 46(4):578–583.

Gurkaynak, R. S., Sack, B. P., and Swanson, E. T. (2004). Do actions speak louder than words? the response of asset prices to monetary

policy actions and statements. *SSRN Electronic Journal*.

Guthrie, G. and Wright, J. (2000). Open mouth operations. *Journal of Monetary Economics*, 46(2):489–516.

Haavelmo, T. (1944). The probability approach in econometrics. *Econometrica: Journal of the Econometric Society*, pages iii–115.

Hall, R. E. (1978). Stochastic implications of the life cycle-permanent income hypothesis: theory and evidence. *Journal of political economy*, 86(6):971–987.

Hall, R. E. (1979). A theory of the natural unemployment rate and the duration of employment. *Journal of monetary economics*, 5(2):153–169.

Hall, R. E. (2005). Job loss, job finding, and unemployment in the us economy over the past fifty years. *NBER macroeconomics annual*, 20:101–137.

Hall, R. E., Farber, H., and Haltiwanger, J. (1995). Lost jobs. *Brookings Papers on Economic Activity*, 1995(1):221–273.

Hall, R. E. and Mishkin, F. S. (1982). The sensitivity of consumption to transitory income: Estimates from panel data on households. *Econometrica*, 50(2):461–481.

Hansen, S. and McMahon, M. (2016). Shocking language: Understanding the macroeconomic effects of central bank communication. *Journal of International Economics*, 99:S114–S133.

Harris, M. N., Loundes, J., and Webster, E. (2002). Determinants of household saving in australia. *Economic Record*, 78(241):207–223.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). Overview of supervised learning. In *The elements of statistical learning*, pages 9–41. Springer.

Haykin, S. S. (2001). *Neural networks: a comprehensive foundation*. Tsinghua University Press.

Haykin, S. S. (2009). *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, NJ, USA:.

Hegel, G. W. F. (1861). *Lectures on the Philosophy of History*. Henry G. Bonn.

Hendry, S. and Madeley, A. (2010). Text mining and the information content of bank of canada communications. *SSRN Electronic Journal*.

Hicks, J. (1984). Is economics a science? *Interdisciplinary Science Reviews*, 9(3):213–219.

Higashi, Y. (2002). Firm specific human capital and unemployment in a growing economy. *Japan and the world economy*, 14(1):35–44.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hippenstiel, R. D. (2001). *Detection theory: applications and digital signal processing*. CRC Press.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196.

Holm, P., Kyyrä, T., and Rantala, J. (1999). Household level economic incentives, unemployment trap and job finding probability. *International Tax and Public Finance*, 6(3):361–378.

Horioka, C. Y. and Watanabe, W. (1997). Why do people save? a micro-analysis of motives for household saving in japan. *The Economic Journal*, 107(442):537–552.

Hubbard, R. G., Skinner, J., and Zeldes, S. P. (1994). Expanding the life-cycle model: Precautionary saving and public policy. *The American Economic Review*, 84(2):174–179.

Huston, S. J. and Chang, Y. R. (1997). Adequate emergency fund holdings and household type. *Journal of Financial Counseling and Planning*, 8(1):37.

Hutchison, T. W. (2013). *Positive economics and policy objectives*. Routledge.

Hylland, A. and Zeckhauser, R. (1979). The efficient allocation of individuals to positions. *Journal of Political economy*, 87(2):293–314.

Igami, M. (2017). Artificial intelligence as structural estimation: Economic interpretations of deep blue, bonanza, and alphago. *arXiv preprint arXiv:1710.10967*.

İlkkaracan, İ. (2012). Why so few women in the labor market in turkey? *Feminist Economics*, 18(1):1–37.

Insel, A., McKenzie, G., and of Economics;, S. U. K. D. (1989). *The Turkish Private Saving Behaviour:(1964-1986)*. University of Southampton, Department of Economics.

Jain, A. K., Mao, J., and Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, 29(3):31–44.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Jansen, D.-J. and De Haan, J. (2005). Talking heads: the effects of ecb statements on the euro–dollar exchange rate. *Journal of International Money and Finance*, 24(2):343–361.

Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., and Yan, S. (2016a). Deep learning with s-shaped rectified linear activation units. In *AAAI*, pages 1737–1743.

Jin, X., Yuan, X., Feng, J., and Yan, S. (2016b). Training skinny deep neural networks with iterative hard thresholding methods. *arXiv preprint arXiv:1607.05423*.

Joachims, T. (2002). *Learning to classify text using support vector machines: Methods, theory and algorithms*, volume 186. Kluwer Academic Publishers Norwell.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed <today>].

Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292.

Kaitz, H. B. (1970). Analyzing the length of spells of unemployment. *Monthly Lab. Rev.*, 93:11.

Kalayci, T. and Ozdamar, O. (1995). Wavelet preprocessing for automated neural network detection of eeg spikes. *IEEE engineering in medicine and biology magazine*, 14(2):160–166.

Katona, G. (1975). *Psychological economics.* Elsevier.

Kawamura, K., Kobashi, Y., Shizume, M., and Ueda, K. (2016). Strategic central bank communication: discourse and game-theoretic analyses of the bank of japan's monthly report.

Kennickell, A. and Lusardi, A. (2003). Wealth accumulation and the importance of precautionary savings. *Dartmouth College*.

Kepenek, Y. and Yentürk, N. (1994). *Türkiye ekonomisi*, volume 99. Remzi Kitabevi.

Keynes, J. M. (1930). *A treatise on money: in 2 volumes*. Macmillan & Company.

Keynes, J. M. (1936). *The General Theory of Employment, Interest and Money*. Macmillan.

King, M. and of Economic Research, N. B. (1983). *The Economics of Saving*. ESRC Programme on Taxation, Incentives and the Distribution of Income: Discussion paper. National Bureau of Economic Research.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *arXiv preprint arXiv:1706.02515*.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., and Mullainathan, S. (2017). Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293.

Kleinberg, J., Ludwig, J., Mullainathan, S., and Obermeyer, Z. (2015). Prediction policy problems. *American Economic Review*, 105(5):491–95.

Koenker, R. (2013). quantreg: Quantile regression. r package version 5.05. *Vienna: R Foundation*.

Koenker, R. and Bassett Jr, G. (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50.

Koenker, R. and Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, 15(4):143–156.

Koenker, R. and Machado, J. A. (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the american statistical association*, 94(448):1296–1310.

Kohn, D. L., Sack, B. P., et al. (2003). *Central bank talk: does it matter and why?* Divisions of Research & Statistics and Monetary Affairs, Federal Reserve Board.

Kreinin, M. E. (1961). Windfall income and consumption: Additional evidence. *The American Economic Review*, 51(3):388–390.

Kroch, E. A. and Sjoblom, K. (1994). Schooling as human capital or a signal: some evidence. *Journal of Human Resources*, pages 156–180.

Krueger, A. and Ashenfelter, O. (1992). Estimates of the economic return to schooling from a new sample of twins. Technical report, National Bureau of Economic Research.

Kuznets, S., Epstein, L., and Jenks, E. (1946). *National Product Since 1869*. National Bureau of Economic Research, Inc.

Kydland, F. E. and Prescott, E. C. (1982). Time to build and aggregate fluctuations. *Econometrica: Journal of the Econometric Society*, pages 1345–1370.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Laibson, D. (2001). A cue-theory of consumption. *The Quarterly Journal of Economics*, 116(1):81–119.

Lansing, J. B. and Kish, L. (1957). Family life cycle as an independent variable. *American sociological review*, 22(5):512–519.

Lapedes, A. and Farber, R. (1987). Nonlinear signal processing using neural networks: Prediction and system modelling.

Layard, R. and Schultz, T. W. (1971). Investment in human capital the role of education and of research. *American Journal of Agricultural Economics*, 53(4):692.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K.-R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer.

Lee, S.-L., Park, M.-H., and Montalto, C. P. (2000). The effect of family life cycle and financial management practices on household saving patterns. *International Journal of Human Ecology*, 1(1):79–93.

Lenat, D. B. and Guha, R. V. (1989). Building large knowledge-based systems; representation and inference in the cyc project.

Lerner, A. P. (1939). Saving and investment: Definitions, assumptions, objectives. *The Quarterly Journal of Economics*, 53(4):611–619.

Lev, J. et al. (1949). The point biserial coefficient of correlation. *The Annals of Mathematical Statistics*, 20(1):125–126.

Lin, Y., Brown, L. D., et al. (2004). Statistical properties of the method of regularization with periodic gaussian reproducing kernel. *The Annals of Statistics*, 32(4):1723–1743.

Lindahl, E. (1930). *Penningpolitikens medel (Means of Monetary Policy)*. Malmö: Förlagsaktiebolaget.

Lindqvist, A. (1981). The saving behavior of households. *Subjective measures of households' saving behaviour. Stockholm, Sweden: Unpublished doctoral dissertation, The Stockholm School of Economics*.

Ljungqvist, L. and Sargent, T. J. (1995). Welfare states and unemployment. *Economic Theory*, 6(1):143–160.

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65.

Lucas, R. E. and Prescott, E. C. (1974). Equilibrium search and unemployment. In *Uncertainty in Economics*, pages 515–540. Elsevier.

Lundberg, E. (1937). *Studies in the theory of economic expansion*. PhD thesis, Norstedt.

Lunt, P. K. and Livingstone, S. (1992). *Mass consumption and personal identity: Everyday economic experience*. Open University Press.

Lunt, P. K. and Livingstone, S. M. (1991). Psychological, social and economic determinants of saving: Comparing recurrent and total savings. *Journal of economic Psychology*, 12(4):621–641.

Lusardi, A. (1998). On the importance of the precautionary saving motive. *The American Economic Review*, 88(2):449–453.

Lutz, F. A. (1938). The outcome of the saving-investment discussion. *The Quarterly Journal of Economics*, 52(4):588–614.

Lydall, H. (1955). The life cycle in income, saving, and asset ownership. *Econometrica: Journal of the Econometric Society*, pages 131–150.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30.

Maddala, G. S. (1983). *Limited-dependent and qualitative variables in econometrics*. Number 3. Cambridge university press.

Maddison, A. et al. (1995). Monitoring the world economy, 1820-1992.

Manning, C. D., Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Maslow, A. H. (1954). Personality and motivation. *Harlow, England: Longman*, 1:987.

Matur, E. P., Sabuncu, A., and Bahçeci, S. (2012). Determinants of private saving and interaction between public & private savings in turkey. *Topics in Middle Eastern and North African Economies*, 14.

Mayer, T. (1972). *Permanent income, wealth, and consumption: a critique of the permanent income theory, the life-cycle hypothesis, and related theories*. Univ of California Press.

McCall, J. J. (1970). Economics of information and job search. *The Quarterly Journal of Economics*, pages 113–126.

McCorduck, P. (1979). *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. Wh freeman San Francisco.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999.

Melvin, M., Saborowski, C., Sager, M., and Taylor, M. P. (2009). Bank of england interest rate announcements and the foreign exchange market.

Memon, A. P., Uqaili, M. A., Memon, Z. A., and Tanwani, N. K. (2014). Time-frequency and artificial neural network applications and analysis for electrical system power quality disturbances in matlab. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 3.

Metin Özcan, K. and Ziya Özcan, Y. (2005). Determinants of private savings in the middle east and north africa. In *Money and Finance in the Middle East: Missed Oportunities or Future Prospects?*, pages 95–113. Emerald Group Publishing Limited.

Miller, G. (1998). *WordNet: An electronic lexical database*. MIT press.

Mincer, J. (1958). Investment in human capital and personal income distribution. *Journal of political economy*, 66(4):281–302.

Mincer, J. (1974). Schooling, experience, and earnings. human behavior & social institutions no. 2.

Mincer, J. and Ofek, H. (1982). Interrupted work careers: Depreciation and restoration of human capital. *Journal of human resources*, pages 3–24.

Minsky, M. (1986). Societies of mind.

Mirer, T. W. (1979). The wealth-age relation among the aged. *The American Economic Review*, 69(3):435–443.

Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.

Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning Department Pittsburgh, PA.

Modigliani, F. and Brumberg, R. (1954). Utility analysis and the consumption function: An interpretation of cross-section data. *Franco Modigliani*, 1:388–436.

Moniz, A. and de Jong, F. (2014). Predicting the impact of central bank communications on financial market investors' interest rate expectations. In *European Semantic Web Conference*, pages 144–155. Springer.

Moritz, B. and Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. *SSRN Electronic Journal*.

Mortensen, D. T. (1970). Job search, the duration of unemployment, and the phillips curve. *The American Economic Review*, 60(5):847–862.

Mortensen, D. T. (1982). The matching process as a noncooperative bargaining game. In *The economics of information and uncertainty*, pages 233–258. University of Chicago Press.

Mortensen, D. T. and Pissarides, C. A. (1994). Job creation and job destruction in the theory of unemployment. *The review of economic studies*, 61(3):397–415.

Mueller, E. T. (1998). *Natural language processing with Thought Treasure*. Signiform New York.

Mullainathan, S. and Spiess, J. (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106.

Muradoglu, G. and Taskin, F. (1996). Differences in household savings behavior: evidence from industrial and developing countries. *The Developing Economies*, 34(2):138–153.

Myrdal, G. (1927). The problem of price formation under economic change. *PhD. Dissertation*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international*

conference on machine learning (ICML-10), pages 807–814.

Narendranathan, W. and Stewart, M. B. (1993). How does the benefit effect vary as unemployment spells lengthen? *Journal of Applied Econometrics*, 8(4):361–381.

Ng, A. Y., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E., and Liang, E. (2006). Autonomous inverted helicopter flight via reinforcement learning. In *Experimental Robotics IX*, pages 363–372. Springer.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134.

Ohlin, B. (1937). Some notes on the stockholm theory of savings and investment i. *The Economic Journal*, 47(185):53–69.

Ombao, H., Lindquist, M., Thompson, W., and Aston, J. (2016). *Handbook of Neuroimaging Data Analysis*. CRC Press.

Osberg, L. (1993). Fishing in different pools: job-search strategies and job-finding success in canada in the early 1980s. *Journal of labor economics*, 11(2):348–386.

Osher, S. and Fedkiw, R. (2003). Implicit functions. In *Level Set Methods and Dynamic Implicit Surfaces*, pages 3–16. Springer.

Ozcan, K. and Ozcan, Y. (2000). Determinants of private saving in mena region, iran and turkey. *Research in Middle East Economies*.

Palumbo, M. G. (1999). Uncertain medical expenses and precautionary saving near the end of the life cycle. *The Review of Economic Studies*, 66(2):395–421.

Panicker, M. and Babu, C. (2012). Efficient fpga implementation of sigmoid and bipolar sigmoid activation functions for multilayer perceptrons. *IOSR Journal of Engineering*, pages 1352–1356.

Parascandolo, G., Huttunen, H., and Virtanen, T. (2016). Taming the waves: sine as activation function in deep neural networks.

Parsons, D. O. (1972). Specific human capital: An application to quit rates and layoff rates. *Journal of political economy*, 80(6):1120–1143.

Paschke, W. (1961). *Bestimmungsgründe des persönlichen Sparens: ein Beitrag zur ökonomischen Verhaltensforschung*, volume 17. Duncker & Humblot.

Pati, Y. C. and Krishnaprasad, P. S. (1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. *IEEE Transactions on Neural Networks*, 4(1):73–85.

Pearl, J. (1985). Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, 1985*, pages 329–334.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pissarides, C. (1986). Unemployment and vacancies in britain. *Economic policy*, 1(3):499–541.

Pissarides, C. A. (1979). Job matchings with state employment agencies and random search. *The Economic Journal*, 89(356):818–833.

Pissarides, C. A. (1985). Short-run equilibrium dynamics of unemployment, vacancies, and real wages. *The American Economic Review*, 75(4):676–690.

Pissarides, C. A. (1987). Search, wage bargains and cycles. *The Review of Economic Studies*, 54(3):473–483.

Pissarides, C. A. (1988). The search equilibrium approach to fluctuations in employment. *The American Economic Review*, 78(2):363–368.

Pissarides, C. A. (2000). *Equilibrium unemployment theory*. MIT press.

Ploskas, N. and Samaras, N. (2016). *GPU Programming in Matlab*. Morgan Kaufmann.

Ponomareva, N. and Sheen, J. (2013). Australian labor market dynamics across the ages. *Economic Modelling*, 35:453–463.

Popper, K. (2005). *The logic of scientific discovery*. Routledge.

Poterba, J. M. (1994). State responses to fiscal crises: The effects of budgetary institutions and politics. *Journal of Political Economy*, 102(4):799–821.

Proust, M. (1957). *Remembrance of things past*, volume 1. Chatto & Windus.

Psacharopoulos, G. (1994). Returns to investment in education: A global update. *World development*, 22(9):1325–1343.

Puntanen, S. and Styan, G. P. (1989). The equality of the ordinary least squares estimator and the best linear unbiased estimator. *The American Statistician*, 43(3):153–161.

Ramsey, F. P. (1928). A mathematical theory of saving. *The economic journal*, 38(152):543–559.

Reeves, R. and Sawicki, M. (2007). Do financial markets react to bank of england communication? *European Journal of Political Economy*, 23(1):207–227.

Reinhart, V. and Sack, B. (2006). Grading the federal open market committee's communications. Technical report, mimeo, Federal Reserve Board of Governors.

Reynolds, L. G. (1951). *The structure of labor markets*. Harper.

Rice, H. G. (1953). Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):358–366.

Robertson, D. (1933a). Mr. robertson on" saving and hoarding". *The Economic Journal*, 43(172):699–712.

Robertson, D. H. (1933b). Saving and hoarding. *The Economic Journal*, 43(171):399–413.

Rockoff, J. E., Jacob, B. A., Kane, T. J., and Staiger, D. O. (2011). Can you recognize an effective teacher when you recruit one? *Education finance and Policy*, 6(1):43–74.

Rodepeter, R. and Winter, J. (1999). Rules of thumb in life-cycle savings models. *Discussion Paper No. 99-81, Sonderforschungsbereich 504, University of Mannheim*.

Rodrik, D. (2008). *One economics, many recipes: globalization, institutions, and economic growth*. Princeton University Press.

Rosen, H. S. and Wu, S. (2004). Portfolio choice and health status. *Journal of Financial Economics*, 72(3):457–484.

Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Roth, A. E. (1984). The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of political Economy*, 92(6):991–1016.

Roth, A. E. and Sotomayor, M. (1992). Two-sided matching. *Handbook of game theory with economic applications*, 1:485–541.

Roth, A. E. and Xing, X. (1994). Jumping the gun: Imperfections and institutions related to the timing of market transactions. *The American Economic Review*, pages 992–1044.

Rothaus, O. S. (1976). On "bent" functions. *Journal of Combinatorial Theory, Series A*, 20(3):300–305.

Rothschild, M. (1978). Models of market organization with imperfect information: A survey. In *Uncertainty in Economics*, pages 459–491. Elsevier.

Rousseeuw, P. J. and Bassett Jr, G. W. (1990). The remedian: A robust averaging method for large data sets. *Journal of the American Statistical Association*, 85(409):97–104.

Royden, H. L. and Fitzpatrick, P. (1988). *Real analysis*. Number 8. Macmillan New York.

Russell, S., Norvig, P., and Intelligence, A. (1995). A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27.

Sager, M. J. and Taylor, M. P. (2004). The impact of european central bank governing council announcements on the foreign exchange market: a microstructural analysis. *Journal of International Money and Finance*, 23(7-8):1043–1051.

Sandmo, A. (1970). The effect of uncertainty on saving decisions. *The Review of Economic Studies*, 37(3):353–360.

Sauri, J. (1774). *Cours complet de mathématiques*, volume 1. Ruault.

Savickỳ, P. (1994). On the bent boolean functions that are symmetric. *European Journal of Combinatorics*, 15(4):407–410.

Schank, R. C. (1975). The conceptual approach to language processing. pages 5–21.

Schultz, T. W. (1961). Investment in human capital. *The American economic review*, 51(1):1–17.

Schumpeter, J. A. (1947). The creative response in economic history. *The journal of economic history*, 7(2):149–159.

Seabold, S. and Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Şelariu, M. E., Smarandache, F., and Niţu, M. (2014). Cardinal functions and integral functions. *Collected Papers, V*, page 45.

Sen, A. K. (1986). Prediction and economic theory. *Proc. R. Soc. Lond. A*, 407(1832):3–23.

Şenses, F. (1996). Structural adjustment policies and employment in turkey. *New Perspectives on Turkey*, 15:65–93.

Shapiro, A. H., Sudhof, M., and Wilson, D. (2018). Measuring news sentiment. Federal Reserve Bank of San Francisco.

Shapiro, M. D. and Slemrod, J. (1993). Consumer response to the timing of income: Evidence from a change in tax withholding. Technical report, National Bureau of Economic Research.

Shapley, L. and Scarf, H. (1974). On cores and indivisibility. *Journal of mathematical economics*, 1(1):23–37.

Sheppard, K. (2012). Introduction to python for econometrics, statistics and data analysis. *Self-published, University of Oxford, version*, 2.

Shimer, R. (2005). The cyclical behavior of equilibrium unemployment and vacancies. *American economic review*, 95(1):25–49.

Shimer, R. (2008). The probability of finding a job. *American Economic Review*, 98(2):268–73.

Shimer, R. (2012). Reassessing the ins and outs of unemployment. *Review of Economic Dynamics*, 15(2):127–148.

Simon, H. A. (1989). Herbert a. simon. In *The State of Economic Science: Views of Six Nobel Laureates*, pages 97–110. W.E. Upjohn Institute.

Simon, H. A. (1990). Prediction and prescription in systems modeling. *Operations Research*, 38(1):7–14.

Smith, A. (1776). An inquiry into the nature and causes of the wealth of nations. In Todd, W. B., editor, *The Glasgow Edition of the Works and Correspondence of Adam Smith, Vol. 2: An Inquiry into the Nature and Causes of the Wealth of Nations, Vol. 1*. Oxford University Press.

Smith, J. P. et al. (1997). *Inheritances and bequests*. RAND.

Solmon, L. C. (1975). The relation between schooling and savings behavior: An example of the indirect effects of education. In *Education, income, and human behavior*, pages 253–294. NBER.

Sopena, J. and Alquezar, R. (1994). Improvement of learning in recurrent networks by substituting the sigmoid activation function. In *ICANN*, volume 94, pages 417–420.

Sowa, J. F. (1987). Semantic networks. encyclopedia of artificial intelligence. *Ed. Stuart C Shapiro*.

Spence, M. (1973). Job market signaling. *The Quarterly Journal of Economics*, 87(3):355–374.

Spence, M. (1978). Job market signaling. In *Uncertainty in Economics*, pages 281–306. Elsevier.

Sraffa, P. (1962). Production of commodities by means of commodities. cambridge, 1960. *Google Scholar*.

Stacey, B. (2015). Fukushima: The failure of predictive models. *Munich Personal RePEc Archive*.

Standage, T. (2002). *The Turk: The life and times of the famous eighteenth-century chess-playing machine*. Walker & Company.

Starr-McCluer, M. (1996). Health insurance and precautionary savings. *The American Economic Review*, 86(1):285–295.

Stigler, G. J. (1961). The economics of information. *Journal of political economy*, 69(3):213–225.

Stigler, G. J. (1962). Information in the labor market. *Journal of political economy*, 70(5, Part 2):94–105.

Stigler, S. M. (1986). *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press.

Stokey, N. L. (1989). *Recursive methods in economic dynamics*. Harvard University Press.

Tansel, A. and Tasci, H. M. (2004). Determinants of unemployment duration for men and women in turkey. *IZA Discussion Paper No. 1258*.

Tansel, I., Wagiman, A., and Tziranis, A. (1991). Recognition of chatter with neural networks. *International journal of machine tools and manufacture*, 31(4):539–552.

Tasci, H. M. and Ozdemir, A. R. (2006). Trends in long-term unemployment and determinants of incidence of long-term unemployment in turkey. *Journal of Economics and Social Research*, 7(2):1–33.

Tasci, H. M. and Tansel, A. (2005). Unemployment and transitions in the turkish labor market: evidence from individual level data. *IZA Discussion Paper No. 1663*.

Tate, R. F. (1954). Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, 25(3):603–607.

Thaler, R. H. (1990). Anomalies: Saving, fungibility, and mental accounts. *Journal of economic perspectives*, 4(1):193–205.

Thaler, R. H. and Benartzi, S. (2004). Save more tomorrow™: Using behavioral economics to increase employee saving. *Journal of political Economy*, 112(S1):S164–S187.

Thompson, A. R., Moran, J. M., and Swenson, G. W. (2001). Interferometry and synthesis in radio astronomy.

Thornton, R. J. and Phelps, E. S. (1972). Microeconomic foundations of employment and inflation theory. *Industrial and Labor Relations Review*, 25(3):444.

Thurow, L. C. (1969). The optimum lifetime distribution of consumption expenditures. *The American Economic Review*, 59(3):324–330.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tikhonov, A. N. (1943). On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198.

Tobin, J. (1951). *Relative income, absolute income and saving*. University of Cambridge, Department of applied economics.

Tokareva, N. (2015). *Bent functions: results and applications to cryptography*. Academic Press.

Trottier, L., Giguère, P., and Chaib-draa, B. (2016). Parametric exponential linear unit for deep convolutional neural networks. *arXiv preprint arXiv:1605.09332*.

Tunali, I. and Ercan, H. (2003). Background study on labour market and employment in turkey. *European Training Foundation, Torino (June 2003)*.

Uygur, E. (2006). Ekonometrinin gelişimi: Iktisadın bilim olma çabası. Technical report, Discussion Paper, Turkish Economic Association.

Uygur, E. (2012). Türkiye'de tasarrufların seyri ve etkileyen bazı unsurlar. Technical report, Discussion Paper, Turkish Economic Association.

Van Ours, J. C. and Vodopivec, M. (2004). How changes in benefits entitlement affect job-finding: Lessons from the slovenian experiment. *IZA Discussion Papers*.

Van Rijckeghem, C. and Üçer, M. (2009). The evolution and determinants of the turkish private saving rate: What lessons for policy?

Varian, H. R. (2014). Big data: New tricks for econometrics. *The Journal of Economic Perspectives*, 28(2):3–27.

Verhulst, P.-F. (1838). Notice sur la loi que la population suit dans son accroissement. correspondance mathématique et physique publiée par a. *Quetelet*, 10:113–121.

Verhulst, P.-F. (1977). A note on the law of population growth. In *Mathematical Demography*, pages 333–339. Springer.

Wahba, J. and Zenou, Y. (2005). Density, social networks and job search methods: Theory and application to egypt. *Journal of Development Economics*, 78(2):443–473.

Wang, M., Min, L., Litscher, G., and Li, M. (2017). Dynamic analysis of coupled sawtooth and rectangle cellular neural networks with application in hbv patients' b-scan images. *Integrative Medicine International*, 4(1-2):19–30.

Wärneryd, K.-E. (1999). *The psychology of saving*. Edward Elgar Publishing.

Wasmer, E. (2006). General versus specific skills in labor markets with search frictions and firing costs. *American Economic Review*, 96(3):811–831.

Webley, P., Burlando, R., and Viner, A. (2000). Individual differences, saving motives and saving behaviour: A cross-national study. *Fairness and cooperation*, pages 497–501.

Weisbrod, B. A. (1966). Investing in human capital. *The Journal of Human Resources*, 1(1):5–21.

Weiss, A. (1995). Human capital vs. signalling explanations of wages. *Journal of Economic perspectives*, 9(4):133–154.

Weiss, Y. (1971). Ability and the investment in schooling: A theoretical note on j. mincer's" distribution of labor income".

White, H. (1992). *Artificial neural networks: approximation and learning theory*. Blackwell Publishers, Inc.

Wickerhauser, M. V. (1994). *Adapted wavelet analysis from theory to software*. IEEE press.

Wicksell, K. (1898). *Geldzins und Güterpreise*. Nihon Keizai Hyoron Sha.

Widrow, B. and Lehr, M. A. (1993). Adaptive neural networks and their applications. *International Journal of Intelligent Systems*, 8(4):453–507.

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT press.

Wu, J.-D. and Liu, C.-H. (2008). Investigation of engine fault diagnosis using discrete wavelet transform and neural network. *Expert Systems with Applications*, 35(3):1200–1213.

Xiao, J. J. and Anderson, J. G. (1997). Hierarchical financial needs reflected by household financial asset shares. *Journal of Family and*

Economic Issues, 18(4):333–355.

Xiao, J. J. and Noring, F. E. (1994). Perceived saving motives and hierarchical financial needs. *Financial Counseling and Planning*, 5(1):25–44.

Xiao, J.-J. and Olson, G. I. (1993). Mental accounting and saving behavior. *Family and Consumer Sciences Research Journal*, 22(1):92–109.

Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Zadeh, R. B. and Zollmann, A. (2009). Predicting market-volatility from federal reserve board meeting minutes nlp for finance. *unpublished manuscript*.

Zalta, E. N. et al. (2003). Stanford encyclopedia of philosophy.

Zeldes, S. P. (1989). Optimal consumption with stochastic income: Deviations from certainty equivalence. *The Quarterly Journal of Economics*, 104(2):275–298.

Zeng, Z., Huang, T., and Zheng, W. X. (2010). Multistability of recurrent neural networks with time-varying delays and the piecewise linear activation function. *IEEE Transactions on Neural Networks*, 21(8):1371–1377.

Zheng, H., Yang, Z., Liu, W., Liang, J., and Li, Y. (2015). Improving deep neural networks using softplus units. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–4. IEEE.

Ziliak, S. and McCloskey, D. N. (2008). *The cult of statistical significance: How the standard error costs us jobs, justice, and lives*. University of Michigan Press.

APPENDICES

APPENDIX A : Glossary
APPENDIX B : Chapter 3 Appendices
APPENDIX C : Chapter 4 Appendices
APPENDIX D : Chapter 5 Appendices

Appendix A : Glossary

Activation function: As will be elaborated in the later sections, an activation function is a function that transforms the data from one form to another one.

Accuracy: Like R^2, it is a measure of the performance of a model:

$$ACC = \frac{Correct\ predictions}{Actual\ output}$$

in continuous output,

$$ACC = \frac{True\ Positives + True\ Negatives}{Actual\ output}$$

in discrete output.

AUC (Area under the ROC Curve): As a similar measure used in the economics literature -the area under a reversed lorenz curve and above a 45 degree line drawn in I. quadrant of the cartesian coordinate system (+x,+y) - for a discrete output estimation, it measures the tradeoff between true positive and false positive in order to calculate a measure like R^2.

Backpropagation: The process of optimization of inital weights to adjust them to increase forecasting, it is also called as autograd that defines reverse-mode auto differentiation.

Batch: The sample used as training data for one iteration, epoch is one batch process.

Bias: Measured errors in between actual model and average model.

Bucketing: The way to use a variable that was converted from scale variable to class (as discrete) one.

Confusion matrix: It is the table for see true positive, true negative and false positive and false negative in a chart to see model's performance.

Convergence: It defines the state of equivalency or approximation of the results which may be ignored statistically after each iteration.

Cross-entropy: It compares two probability distributions by generalizing its loss function, it is similar to Kullback–Leibler divergence which is also called as relative entropy.

Decision boundary: In a discrete choice model of classification problem, mostly (but not general) a clear cut seperator between classes.

Deep Model: Multilayered neural networks that contain at least two layers or more.

Dropout regularization: Random or systematic drop in neurons in order to regularize to deal with overfitting.

Ensemble: In order to mix some models and their outputs aggregated to one output. It makes it easier to use methods simultaneously that are irrelevant, but they may fuel some gaps reciprocally.

Epoch: The number of batches that are processed and passed over again and again to train the model.

Estimator: Apart from the meaning defined as thee methods used to estimate a parameter or a density, in statistics, estimator is a collection of functions used to estimate something as read-to-use, especially in Pytorch and Tensorflow.

F1 Score: A measure to estimate the balance of the tradeoff (see Figure A.1) in between precion and recall [1] to determine indirectly the test's accuracy:

$$F1 = 2 * \frac{P * R}{P + R}$$

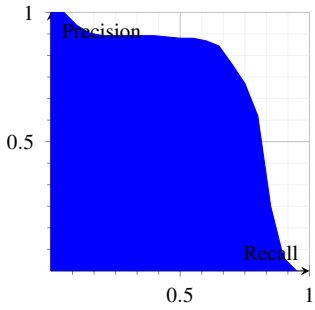


Figure A.1 : Precision vs Recall.

False Positive (FP): Misclassification of negatives as in positives (see. Table A.1).

[1] http://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

False Negative (FN): Misclassification of positives as in negatives (see. Table A.1).

Feature: The variable, attribute, that used to estimate output.

Feature engineering: Choosing the appropriate variable set to increase accuracy of output.

Fitting: Estimate a model.

Gradient: It is a vector containing partial derivatives with respect to all individual independent variables or features.

Gradient descent: It is an algorithm that aims to find the best combination of parameters that are iteratively corrected by means of gradient descent where loss function is minimized.

Hyperparameter: The parameters that should be adjusted or be determined at the beginning of a machine learning model.

Kullback–Leibler divergence: It compares two probability distributions by maintaining expected probability distribution if the first one is feeded. It calculates how the expected and second distributions are divergent to each other.

Label: Ex-ante known output variable.

Layer: Number of layers in between inputs and output(s).

Learning rate: In each iteration, a scalar that determines the weight of gradient descent in order to converge.

Loss Function: A function taking into account absolute value of the difference in between predicted and actual output.

Mean Squared Error (MSE): The mean of squares of the differences between expected and observed values.

Normalization: Standardization between -1 and +1 or 0 and +1.

Outliers: The observations affecting the model more than other observations, which may cause to change the model from actual one.

Overfitting:Overestimate the output variable.

Parameter: Unlike hyperparameter, the paramaters that can be determined by ml in itself.

Perceptron: Firstly invented by Rosenblatt (1957) to define a network model that contains one layer seperating outputs as binary by converting the sum of all inputs with their special weights via an activation function.

Precision: The ratio of true positives to all positives to see the proportion of correctly predicted outcome as following:

$$P = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall: The ratio of true positives to all correct predictions to see relevancy of predictions as below:

$$R = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Regularization: It is the parameter to decrease some effects of certain variable to its affection to outputs, penalizing or punishing parameter, there rate two kinds called as L^1 and L^2.

Regularization rate: In the function of regularization, it is α which determines how much parameters are regularized when it is added to the loss function.

ROC (receiver operating characteristic) Curve: ROC is drawn to compare true positive rate against false positive rate.

Root Mean Squared Error (RMSE): Squared version of MSE (see above).

Scaling: Transform the data in between a predetermined range.

Sparse feature: A variable that takes a value of 1 in certain point of the data, otherwise it is encoded with zeros, one-hot encoding or similar to dummy variable in econometrics

Step: Iterating backward or forward propagation in one batch.

Stochastic gradient descent: Randomly chosen data used to minimize gradient descent in each epoch.

Tensor: Tensorflow's main data type, three dimensional vector representation.

Test set: The dataset to calculate how well your model has been trained, namely, the accuracy reflector.

Training set: The main dataset that is used for learning algorithm in order to fit and estimate the parameters.

True negative (TN) : Correct classification of negatives as in negatives (see. Table A.1).

True positive (TP) : Correct classification of positives as in positives (see. Table A.1).

Underfitting: Underestimate the output variable.

Weight: Similar to coefficient in econometrics, it determines the effect of input to next step.

Wide Model: One or several layered neural network model mostly consisting sparse features.

Validation set: The dataset that may be derived or a continuation of the training set used for determining and fine tuning of hyperparameters (number of variables, layers etc.).

Variance: Quantification of how much predicted output varies from one learning sample/population to another.

Table A.1 : Types of Errors (Confusion Matrix)

		H_0 (Null hypothesis) is	
		True	False
Test outcome	Reject	Type I Error (False Positive)	Correct inference (True Positive)
	Not Rejected	Correct inference (True Negative)	Type II Error (False Negative)

Appendix B : Chapter 3 Appendices

Table B.1 : Descriptive Statistics of Quantitative Features.

Statistic	N	Mean	St. Dev.	Min	Max
yas	203,891	32.150	11.160	15	99
ikamettr	3,569	15.360	11.580	0	71
ikametil	3,569	15.360	11.580	0	71
educ_year	203,891	7.545	3.947	0	15
experience	203,891	14.970	11.650	0	83
deflator	203,891	1.114	0.291	0.681	1.655
agirlik_katsayisi	203,891	160.800	79.950	5.620	754.200
hh_buyukluk	203,891	4.616	2.234	1	31
hh_buyukluk_estimated	203,891	4.485	2.183	1	31
inactive	52,774	7.486	9.646	0	73

Table B.2 : Descriptive Statistics of Qualitative Features

cinsiyet	yas.k	dogum_yer	yurtdisi_durum
Min. :1.00	Min. : 4.00	Min. :1	Min. :1
1st Qu.:1.00	1st Qu.: 5.00	1st Qu.:1	1st Qu.:2
Median :1.00	Median : 7.00	Median :1	Median :2
3rd Qu.:1.00	3rd Qu.: 9.00	3rd Qu.:1	3rd Qu.:2
Max. :2.00	Max. :14.00	Max. :2	Max. :2
		NA's :70381	NA's :100874
yakinlik	okul_biten_k	foet99_k	nuts1
Min. : 1.00	Min. :0.00	Min. : 1	Min. : 1.00
1st Qu.: 1.00	1st Qu.:2.00	1st Qu.: 6	1st Qu.: 3.00
Median : 3.00	Median :3.00	Median : 6	Median : 6.00
3rd Qu.: 3.00	3rd Qu.:4.00	3rd Qu.:12	3rd Qu.: 9.00
Max. :11.00	Max. :6.00	Max. :21	Max. :12.00
		NA's :171336	

Table B.2 (Continued) : Descriptive Statistics of Qualitative Features.

nuts2	onceki_ikamet	yerlesim_tur	fertno_es
Min. : 1.0	Min. :1	Min. :1	Min. : 1
1st Qu.: 5.0	1st Qu.:1	1st Qu.:1	1st Qu.: 2
Median :11.0	Median :1	Median :2	Median :99
3rd Qu.:19.0	3rd Qu.:1	3rd Qu.:3	3rd Qu.:99
Max. :26.0	Max. :2	Max. :3	Max. :99
	NA's :155343	NA's :160119	NA's :13426
fertno_anne	fertno_baba	okur_yazar	egitim_devam_k
Min. : 1	Min. : 1	Min. :1	Min. :1.00
1st Qu.: 2	1st Qu.: 1	1st Qu.:1	1st Qu.:2.00
Median : 3	Median :99	Median :1	Median :2.00
3rd Qu.:99	3rd Qu.:99	3rd Qu.:1	3rd Qu.:2.00
Max. :99	Max. :99	Max. :2	Max. :2.00
NA's :13426	NA's :13426	NA's :122214	
okul_devam_k	sinif_devam	kurs	kurs.amaci
Min. : 1	Min. :1	Min. :1.00	Min. :1
1st Qu.: 4	1st Qu.:1	1st Qu.:2.00	1st Qu.:1
Median : 4	Median :2	Median :2.00	Median :1
3rd Qu.: 5	3rd Qu.:3	3rd Qu.:2.00	3rd Qu.:2
Max. :32	Max. :9	Max. :2.00	Max. :3
NA's :189243	NA's :191939		NA's :202449
medeni_durum	gecmis_is	is_ayril_neden	nace2_gecmis_k
Min. :1	Min. :1	Min. : 1	Min. : 1
1st Qu.:1	1st Qu.:1	1st Qu.: 1	1st Qu.: 2
Median :2	Median :1	Median : 3	Median : 3
3rd Qu.:2	3rd Qu.:2	3rd Qu.: 7	3rd Qu.: 4
Max. :9	Max. :2	Max. :13	Max. :11
NA's :17532	NA's :112294	NA's :164341	NA's :164341
isco08_gecmis_k	isteki_durum_gecmis	nace2_gecenyil_k	isteki_durum_gecenyil
Min. :1	Min. :1	Min. : 1	Min. : 1
1st Qu.:5	1st Qu.:1	1st Qu.: 1	1st Qu.: 1
Median :7	Median :1	Median : 3	Median : 2
3rd Qu.:9	3rd Qu.:2	3rd Qu.: 6	3rd Qu.: 3
Max. :9	Max. :5	Max. :11	Max. :10
NA's :164341	NA's :164341	NA's :151684	NA's :151812

Table B.3 : Classifier Results.

Classifier	Classes	Precision	Recall	F1-score	Support	Confusion Matrix		Accuracy	AUC
Gaussian	0	0.54	0.48	0.51	93740	44687	49053		
Train	1	0.54	0.6	0.57	94468	37512	56956		
	Avg/Total	0.54	0.54	0.54	188208			0.5401	0.5409
Gaussian	0	0.54	0.47	0.5	7811	3668	4143		
Test	1	0.53	0.6	0.57	7872	3139	4733		
	Avg/Total	0.54	0.54	0.53	15683			0.5357	0.5365
Bernoulli	0	0.61	0.15	0.24	93740	14356	79384		
Train	1	0.52	0.9	0.66	94468	9282	85186		
	Avg/Total	0.56	0.53	0.45	188208			0.5289	0.5340
Bernoulli	0	0.6	0.15	0.25	7811	1208	6603		
Test	1	0.52	0.9	0.66	7872	818	7054		
	Avg/Total	0.56	0.53	0.45	15683			0.5268	0.5320
Logistic	0	0	0	0	93740	0	93740		
Train	1	0.5	1	0.67	94468	0	94468		
	Avg/Total	0.25	0.5	0.34	188208			0.5019	0.5395
Logistic	0	0	0	0	7811	0	7811		
Test	1	0.5	1	0.67	7872	0	7872		
	Avg/Total	0.25	0.5	0.34	15683			0.5019	0.5483
KNeighbors	0	0	0.7	1	0.82	93740	0		
Train	1	1	1	0.57	0.73	40605	53863		
	Avg/Total	0.85	0.78	0.77	188208			0.7843	0.9082
KNeighbors	0	0.54	0.78	0.64	7811	6061	1750		
Test	1	0.6	0.34	0.43	7872	5201	2671		
	Avg/Total	0.57	0.56	0.53	15683			0.5568	0.5820
Decision Tree	0	1	1	1	93740	93740	0		
Train	1	1	1	1	94468	26	94442	OVERFIT	OVERFIT

Table B.3 (Continued) : Classifier Results.

Classifier	Classes	Precision	Recall	F1-score	Support	Confusion Matrix		Accuracy	AUC
	Avg/Total	1	1	1	188208			0.9999	1.0000
Decision Tree	0	0.62	0.62	0.62	7811	4830	2981		
Test	1	0.62	0.62	0.62	7872	2968	4904		
	Avg/Total	0.62	0.62	0.62	15683			0.6207	0.6207
Linear Discriminant	0	0.5	1	0.66	93740	93740	0		
Train	1	0	0	0	94468	94468	0		
	Avg/Total	0.25	0.5	0.33	188208			0.4981	0.6429
Linear Discriminant	0	0.5	1	0.66	7811	7811	0		
Test	1	0	0	0	7872	7872	0		
	Avg/Total	0.25	0.5	0.33	15683			0.4981	0.6474
Quadratic Discriminant	0	0.61	0.65	0.63	93740	60771	32969		
Train	1	0.63	0.59	0.61	94468	38795	55673		
	Avg/Total	0.62	0.62	0.62	188208			0.6187	0.6771
Quadratic Discriminant	0	0.6	0.64	0.62	7811	5010	2801		
Test	1	0.62	0.58	0.6	7872	3301	4571		
	Avg/Total	0.61	0.61	0.61	15683			0.6109	0.6665
SGD	0	0	0	0	93740	0	93740		
Train	1	0.5	1	0.67	94468	0	94468		
	Avg/Total	0.25	0.5	0.34	188208			0.4981	0.5000
SGD	0	0	0	0	7811	0	7811		
Test	1	0.5	1	0.67	7872	0	7872		
	Avg/Total	0.25	0.5	0.34	15683			0.4981	0.5000
SVM	0	1	0.75	0.85	93740	69922	23818		
Train	1	0.8	1	0.89	94468	11	94457		
	AVG / Total	0.9	0.87	0.87	188208			0.8734	0.9704
SVM	0	0.73	0	0	7811	11	7800		

Table B.3 (Continued) : Classifier Results.

Classifier	Classes	Precision	Recall	F1-score	Support	Confusion Matrix		Accuracy	AUC
Test	1	0.5	1	0.67	7872	4	7868.0000		
	AVG / Total	0.62	0.5	0.34	15683			0.5024	0.5046
Linear SVC	0	0.54	0.48	0.51	93740	44686	49054		
Train	1	0.54	0.6	0.57	94468	37524	56944		
	AVG / Total	0.54	0.54	0.54	188208			0.539987673	0.5401
Linear SVC	0	0.54	0.47	0.5	7811	3669	4142		
Test	1	0.53	0.6	0.57	7872	3127	4745		
	AVG / Total	0.54	0.54	0.53	15683			0.5365	0.5407
Neural Networks									
Train	Avg/Total							0.5018	0.5000
Neural Networks									
Test	Avg/Total							0.5019	0.5000

Table B.4 : Impurity Measurement Metrics.

ML Agorithm	Set	Gini Impurity	Cohen's (κ)	Entropy	Class Error
Gaussian	Train	0.4962	0.0795	0.9945	0.4564
Gaussian	Test	0.4961	0.0725	0.9945	0.4564
Bernoulli	Train	0.4997	0.0549	0.9996	0.4885
Bernoulli	Test	0.4997	0.0515	0.9996	0.4885
ExtraTrees CL.**	Train	0.0000	0.0000	0.0000	0.9997
ExtraTrees CL.	Test	0.2573	0.0000	0.0000	0.0000
Decision Tree CL.	Train	0.0000	0.0000	0.4999	0.0000
Decision Tree CL.	Test	0.2227	0.0000	0.0000	0.4999
Logistic Regression	Train	0.4999	0.0000	0.9999	0.4979
Logistic Regression	Test	0.4999	0.0000	0.9999	0.4979
XGB CL.	Train	0.0030	0.0002	0.3820	0.1744
XGB CL.	Test	0.3452	0.0019	0.0001	0.4261
Linear Disc.**	Train	0.8224	0.2571	0.5044	0.2374
Linear Disc.	Test	0.1722	0.8906	0.3077	0.5272
Quadratic Disc.	Train	0.2518	0.7541	0.0000	0.9997
Quadratic Disc.	Test	0.2330	0.3883	0.0000	0.0000
SGD CL.	Train	0.0000	0.0000	0.0000	0.5000
SGD CL.	Test	0.0000	0.0000	0.0000	0.0000
KNeighbors CL.	Train	0.5000	0.0000	0.5123	0.0000
KNeighbors Cl.	Test	0.5000	0.0136	0.3316	0.0594
Random Forest CL.	Train	0.0000	0.0000	0.0000	0.9696
Random Forest CL.	Test	0.0594	0.0068	0.0000	0.2651
Bagging CL.	Train	0.0000	0.0000	0.4999	0.2604
Bagging CL.	Test	0.0000	0.0000	0.4999	0.2513
AdaBoost CL.	Train	0.9999	0.4974	0.8808	0.3979
AdaBoost CL.	Test	0.9999	0.4965	0.0042	0.3452

* Note: These statistics are not available for SVC, LinearSVC and Shallow NN.
** CL. and Disc. stand for Classifier and Discriminant respectively.

Table B.5 : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
cinsiyetFemale	-0.1	0.00746	-13.46	<2e-16 ***
yas.k20-24	0.0245	0.00615	3.99	6.5e-05 ***
yas.k25-29	0.0061	0.00905	0.67	0.50052
yas.k30-34	-0.000283	0.013	-0.02	0.98261
yas.k35-39	0.0164	0.017	0.96	0.33541
yas.k40-44	0.0232	0.0211	1.1	0.27166
yas.k45-49	0.0221	0.0253	0.87	0.3818
yas.k50-54	0.0167	0.0296	0.57	0.57132
yas.k55-59	0.0237	0.0339	0.7	0.48396
yas.k60-64	0.0486	0.0389	1.25	0.21152
yas.k65+	0.169	0.0417	4.04	5.3e-05 ***
dogum_yerAbroad	0.0218	0.0145	1.5	0.13249
dogum_yerNA	-0.0482	0.0117	-4.12	3.8e-05 ***
yurtdisi_durumNo	0.0307	0.0118	2.61	0.00900 **
yurtdisi_durumNA	0.0279	0.0125	2.23	0.02549 *
yakinlikSpouse	-0.0301	0.017	-1.77	0.07683 .
yakinlikHis/her children	-0.14	0.00909	-15.38	<2e-16 ***
yakinlikFather/Mother	-0.126	0.0164	-7.69	1.5e-14 ***
yakinlikBrother/Sister	-0.0828	0.0123	-6.75	1.5e-11 ***
yakinlikFather in law/Mother in law	0.0776	0.0349	2.22	0.02620 *
yakinlikSon in law/Daughter in law	-0.0344	0.00976	-3.52	0.00043 ***
yakinlikGrandchild	0.0635	0.019	3.34	0.00085 ***
yakinlikOther relatives	-0.0603	0.0323	-1.87	0.06206 .
yakinlikNon-relatives	0.125	0.0426	2.93	0.00344 **
yakinlikHousekeeper staying at home	0.501	0.00172	290.45	<2e-16 ***
okul_biten_kLiterate but not completed any educational institution	-0.00234	0.00991	-0.24	0.81363
okul_biten_kPrimary school	0.0223	0.0111	2.01	0.04443 *
okul_biten_kSecondary school	0.0187	0.0114	1.64	0.10093
okul_biten_kHigh school	0.0701	0.0232	3.02	0.00253 **
okul_biten_kVocational or technical high school	0.086	0.0232	3.71	0.00021 ***
okul_biten_kHigher education	0.175	0.0444	3.94	8.1e-05 ***
foet99_kArts	-0.00959	0.0178	-0.54	0.59041
foet99_kHumanities	0.0166	0.0163	1.02	0.30856
foet99_kSocial and behavioural science	-0.0259	0.0176	-1.48	0.13994
foet99_kJournalism and information	-0.0859	0.0565	-1.52	0.12836
foet99_kBusiness and administration	-0.0141	0.0122	-1.15	0.24953
foet99_kLaw	0.059	0.0402	1.47	0.14238
foet99_kLife science	-0.0215	0.0342	-0.63	0.52838
foet99_kPhysical science	-0.0397	0.0232	-1.71	0.08735 .

Table B.5 (Continued) : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
foet99_kMathematics and statistics	0.0504	0.0335	1.51	0.13225
foet99_kComputing	0.0119	0.019	0.63	0.53027
foet99_kEngineering and engineering trades	0.057	0.0129	4.41	1.1e-05 ***
foet99_kManufacturing and processing	0.0241	0.016	1.51	0.1307
foet99_krchitecture and building	0.0159	0.0192	0.83	0.40678
foet99_kAgriculture, forestry and fishery	0.00787	0.0234	0.34	0.73697
foet99_kVeterinary	0.076	0.0605	1.26	0.20875
foet99_kHealth	0.119	0.0236	5.02	5.1e-07 ***
foet99_kSocial services	-0.0474	0.0199	-2.38	0.01739 *
foet99_kPersonal services	-0.00652	0.0204	-0.32	0.74945
foet99_kTransport services and environtal protection	0.0599	0.0507	1.18	0.23669
foet99_kSecurity services	-0.00153	0.052	-0.03	0.97652
foet99_kNA	0.0395	0.0119	3.31	0.00092 ***
nuts1TR2 West Marmara	0.0159	0.00881	1.81	0.07037 .
nuts1TR3 Aegean	-0.0215	0.00732	-2.93	0.00339 **
nuts1TR4 East Marmara	-0.0658	0.00646	-10.17	<2e-16 ***
nuts1TR5 West Anatolia	-0.041	0.00802	-5.11	3.2e-07 ***
nuts1TR6 Mediterranean	-0.113	0.00665	-17	<2e-16 ***
nuts1TR7 Central Anatolia	-0.146	0.00798	-18.27	<2e-16 ***
nuts1TR8 West Black Sea	-0.0596	0.00786	-7.58	3.6e-14 ***
nuts1TR9 East Black Sea	-0.0332	0.00723	-4.6	4.3e-06 ***
nuts1TRA Northeast Anatolia	-0.138	0.00919	-15.04	<2e-16 ***
nuts1TRB Centraleast Anatolia	-0.14	0.00718	-19.54	<2e-16 ***
nuts1TRC Southeast Anatolia	-0.21	0.00709	-29.64	<2e-16 ***
nuts2TR21 Edirne, Tekirdağ, Kırklareli	-0.00572	0.0113	-0.5	0.61399
nuts2TR31 İzmir	-0.0641	0.00751	-8.54	<2e-16 ***
nuts2TR32 Denizli, Aydın, Muğla	-0.0126	0.00907	-1.39	0.16518
nuts2TR41 Bursa, Eskişehir, Bilecik	0.122	0.00764	15.99	<2e-16 ***
nuts2TR51 Ankara	-0.0192	0.00844	-2.27	0.02312 *
nuts2TR61 Antalya, Isparta, Burdur	0.0873	0.00932	9.37	<2e-16 ***
nuts2TR62 Adana, Mersin	0.0403	0.0073	5.51	3.5e-08 ***
nuts2TR71 Nevşehir, Aksaray, Niğde, Kırıkkale, Kırşehir	0.0104	0.0107	0.97	0.33033
nuts2TR81 Zonguldak, Karabük, Bartın	-0.00766	0.012	-0.64	0.52427
nuts2TR82 Kastamonu, Çankırı, Sinop	-0.0461	0.0113	-4.08	4.5e-05 ***

Table B.5 (Continued) : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
nuts2TRA1 Erzurum, Erzincan, Bayburt	0.00781	0.012	0.65	0.51413
nuts2TRB1 Malatya, Elazığ, Bingöl, Tunceli	-0.031	0.00953	-3.25	0.00115 **
nuts2TRC1 Gaziantep, Adıyaman, Kilis	0.0757	0.0088	8.59	<2e-16 ***
nuts2TRC2 Diyarbakır, Şanlıurfa	0.1	0.00812	12.35	<2e-16 ***
onceki_ikametIn a foreign country	0.0114	0.0128	0.89	0.37304
onceki_ikametNA	0.0315	0.0054	5.83	5.5e-09 ***
yerlesim_turDistrict center	0.0135	0.00617	2.19	0.02889 *
yerlesim_turSub-district or a village	0.0148	0.00689	2.14	0.03222 *
fertno_esSpouse (or cohabiting partner)	0.105	0.0167	6.29	3.1e-10 ***
fertno_esChild of reference person	0.127	0.0177	7.16	8.2e-13 ***
fertno_esBride or bridegroom	0.127	0.0178	7.15	8.8e-13 ***
fertno_esGrandchild	0.11	0.0207	5.31	1.1e-07 ***
fertno_esMother/father in law	0.1	0.0227	4.42	9.8e-06 ***
fertno_esOther relatives	0.0375	0.0297	1.26	0.20646
fertno_esNon-relatives	0.129	0.0313	4.12	3.8e-05 ***
fertno_esNA	0.0528	0.0179	2.95	0.00321 **
fertno_anne2	-0.0336	0.0111	-3.03	0.00245 **
fertno_anne3	-0.0426	0.0131	-3.24	0.00118 **
fertno_anne4	-0.0331	0.0213	-1.55	0.12088
fertno_anne5	-0.0314	0.0246	-1.28	0.20152
fertno_anne6	-0.0758	0.0311	-2.43	0.01498 *
fertno_anne7	0.0063	0.0403	0.16	0.87564
fertno_anne8	-0.0233	0.0498	-0.47	0.64008
fertno_anne9	-0.0423	0.0735	-0.57	0.5654
fertno_anne10	-0.162	0.0751	-2.15	0.03122 *
fertno_anne11	-0.0945	0.0938	-1.01	0.31363
fertno_anne12	0.133	0.182	0.73	0.46279
fertno_anne13	0.158	0.162	0.97	0.32989
fertno_anne14	-0.499	0.00157	-317.76	<2e-16 ***
fertno_anne15	-0.158	0.21	-0.75	0.4506
fertno_anne17	-0.499	0.00428	-116.54	<2e-16 ***
fertno_anne18	0.279	0.187	1.49	0.13636
fertno_anne19	-0.499	0.00245	-204.1	<2e-16 ***
fertno_anne23	-0.499	0.00324	-154.26	<2e-16 ***
fertno_anne99	-0.00748	0.0105	-0.71	0.47796
fertno_anneNA	-0.0435	0.0147	-2.96	0.00303 **
fertno_baba2	-0.0422	0.0163	-2.59	0.00948 **
fertno_baba3	-0.0341	0.0238	-1.43	0.15266
fertno_baba4	-0.0627	0.0328	-1.91	0.05577 .
fertno_baba5	-0.0645	0.0342	-1.88	0.05953 .

Table B.5 (Continued) : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
fertno_baba6	-0.0176	0.0423	-0.42	0.67655
fertno_baba7	-0.0931	0.05	-1.86	0.06251 .
fertno_baba8	-0.191	0.0656	-2.92	0.00353 **
fertno_baba9	0.0648	0.0872	0.74	0.45741
fertno_baba10	0.153	0.105	1.46	0.14414
fertno_baba11	-0.192	0.156	-1.23	0.2192
fertno_baba12	-0.0254	0.198	-0.13	0.89804
fertno_baba13	0.136	0.201	0.67	0.50114
fertno_baba14	0.292	0.212	1.38	0.16825
fertno_baba15	-0.159	0.32	-0.5	0.61855
fertno_baba16	0.501	0.00579	86.45	<2e-16 ***
fertno_baba99	-0.0481	0.0104	-4.62	3.8e-06 ***
okur_yazarNo	0.00336	0.00945	0.36	0.72179
okur_yazarNA	0.00138	0.00904	0.15	0.87864
egitim_devam_kNo	-0.0452	0.0168	-2.7	0.00702 **
okul_devam_kHigh school	-0.178	0.0671	-2.66	0.00791 **
okul_devam_kVocational or technical high school	-0.0117	0.0906	-0.13	0.89704
okul_devam_kOpen education system	-0.206	0.0621	-3.31	0.00094 ***
okul_devam_kFaculty/high school	-0.193	0.0633	-3.04	0.00234 **
okul_devam_kMaster/doctorate"	-0.188	0.0682	-2.75	0.00598 **
okul_devam_kNA	-0.198	0.0643	-3.07	0.00213 **
sinif_devam2	-0.00447	0.0128	-0.35	0.72708
sinif_devam3	-0.00837	0.0141	-0.6	0.55174
sinif_devam4	-0.0114	0.0149	-0.77	0.44281
sinif_devam5	-0.372	0.0896	-4.15	3.3e-05 ***
sinif_devam6	-0.00323	0.0869	-0.04	0.97038
sinif_devam7	0.0443	0.0685	0.65	0.51774
sinif_devam8	0.00736	0.057	0.13	0.89724
sinif_devam9	-0.152	0.0843	-1.8	0.07111 .
sinif_devamNA	0.0476	0.0138	3.46	0.00055 ***
kursNo	0.16	0.0206	7.78	7.0e-15 ***
kurs.amaciFor Personal or Social	-0.00849	0.0304	-0.28	0.7799
kurs.amaciOther	-0.00335	0.0928	-0.04	0.97121
kurs.amaciNA	-0.000182	0.0293	-0.01	0.99504
medeni_durumMarried	0.0218	0.00564	3.86	0.00011 ***
medeni_durumDivorced	-0.00293	0.0082	-0.36	0.72105
medeni_durumWidowed	0.0136	0.0163	0.83	0.40612
medeni_durumNA	-0.0594	0.00584	-10.17	<2e-16 ***
gecmis_isNo	0.0712	0.00756	9.42	<2e-16 ***
gecmis_isNA	0.164	0.00715	22.93	<2e-16 ***
is_ayril_nedenas working seasonally	0.198	0.0102	19.53	<2e-16 ***

Table B.5 (Continued) : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
is_ayril_neden Dismissed, Liquidated,Bankrupted	0.104	0.0104	10.02	<2e-16 ***
is_ayril_nedenNot satisfied with job	0.0961	0.0108	8.93	<2e-16 ***
is_ayril_nedenOwn illness or disability	0.242	0.00871	27.72	<2e-16 ***
is_ayril_nedenLooking after children or incapacitated adults in the family	0.26	0.0132	19.64	<2e-16 ***
is_ayril_nedenHer spouse's request / Due to marriage	0.249	0.0124	20.02	<2e-16 ***
is_ayril_nedenEducation or training	0.27	0.0103	26.24	<2e-16 ***
is_ayril_nedenRetirement	0.3	0.00857	35.02	<2e-16 ***
is_ayril_nedenMilitary service	0.31	0.00773	40.11	<2e-16 ***
is_ayril_nedenNA	0.307	0.0125	24.48	<2e-16 ***
nace2_gecmis_kManufacturing, mining and quarrying and other industries	0.00324	0.0134	0.24	0.80919
nace2_gecmis_kConstruction	-0.106	0.0137	-7.7	1.3e-14 ***
nace2_gecmis_kWholesale retail trade, transportation and storage, accommodation and service	-0.0619	0.0135	-4.59	4.4e-06 ***
nace2_gecmis_kInformation and communication	-0.0923	0.0381	-2.43	0.01527 *
nace2_gecmis_kFinancial and insurance activities	0.0809	0.023	3.52	0.00043 ***
nace2_gecmis_kReal estate activities	-0.174	0.0499	-3.48	0.00050 ***
nace2_gecmis_kProfessional, scientific and technical activities, administrative and support services	-0.173	0.0185	-9.33	<2e-16 ***
nace2_gecmis_kPublic administration and defense, education, health and social services	0.00522	0.0153	0.34	0.73327
nace2_gecmis_kOther service activities	0.0405	0.0166	2.44	0.01458 *
nace2_gecmis_kUnknown	0.195	0.0281	6.95	3.6e-12 ***
nace2_gecmis_kNA	-0.169	0.0247	-6.83	8.3e-12 ***
isco08_gecmis_kProfessionals	-0.0683	0.0204	-3.35	0.00082 ***
isco08_gecmis_kTechnicians and associate professionals	0.0319	0.0202	1.58	0.11473
isco08_gecmis_kClerical support workers	0.0074	0.0189	0.39	0.69544

Table B.5 (Continued) : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
isco08_gecmis_kService and sales workers	-0.0208	0.0168	-1.23	0.21714
isco08_gecmis_kSkilled agricultural, forestry and fishery workers	-0.0145	0.0201	-0.72	0.47121
isco08_gecmis_kCraft and related trades workers	0.00938	0.0174	0.54	0.58868
isco08_gecmis_kPlant and machine operators, and assemblers	0.00196	0.0182	0.11	0.91425
isco08_gecmis_kElemantary occupations	0.0133	0.0169	0.79	0.43199
isteki_durum_gecmisEmployer	0.239	0.00962	24.78	<2e-16 ***
isteki_durum_gecmisSelf employed	0.0759	0.0125	6.09	1.1e-09 ***
isteki_durum_gecmisUnpaid family worker	0.145	0.0105	13.89	<2e-16 ***
isteki_durum_gecmisNA	0.158	0.0144	11	<2e-16 ***
nace2_gecenyil_kManufacturing, mining and quarrying and other industries	0.00584	0.008	0.73	0.46511
nace2_gecenyil_kConstruction	0.0363	0.0112	3.25	0.00114 **
nace2_gecenyil_kWholesale retail trade, transportation and storage, accommodation and service	0.0011	0.00719	0.15	0.87867
nace2_gecenyil_kInformation and communication	-0.0406	0.0346	-1.18	0.23964
nace2_gecenyil_kFinancial and insurance activities	-0.00498	0.0156	-0.32	0.749
nace2_gecenyil_kReal estate activities	0.0303	0.0431	0.7	0.48169
nace2_gecenyil_kProfessional, scientific and technical activities, administrative and support services	-0.0278	0.0155	-1.8	0.07195 .
nace2_gecenyil_kPublic administration and defense, education, health and social services	-0.00957	0.00872	-1.1	0.27202
nace2_gecenyil_kOther service activities	0.0138	0.0115	1.21	0.22737
nace2_gecenyil_kUnknown	-0.0276	0.0186	-1.49	0.13723
nace2_gecenyil_kNA	0.0879	0.00608	14.47	<2e-16 ***
isteki_durum_gecenyilEmployer	-0.02	0.00967	-2.06	0.03918 *
isteki_durum_gecenyilSelf employed	-0.0254	0.00692	-3.67	0.00024 ***

Table B.5 (Continued) : Logistic Regression.

	dF/dx	Std. Err.	z	P>|z|
isteki_durum_gecenyilUnpaid family worker	-0.0209	0.00795	-2.63	0.00864 **
isteki_durum_gecenyilNA	-0.0406	0.00617	-6.57	5.0e-11 ***
yas	-0.0318	0.00684	-4.65	3.4e-06 ***
ikamettr	-0.00258	0.000809	-3.19	0.00144 **
experience	0.0213	0.0069	3.09	0.00198 **
deflator	0.248	0.0115	21.58	<2e-16 ***
agirlik_katsayisi	-0.0000286	0.0000196	-1.46	0.14494
hh_buyukluk	0.00436	0.00147	2.97	0.00301 **
hh_buyukluk_estimated	-0.00419	0.00157	-2.67	0.00752 **
inactive	0.00238	0.000346	6.87	6.6e-12 ***
Observations	203,891			

Note: *p<0.1; **p<0.05; ***p<0.01

Table B.6 : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
1	CH	YIL	Reference year of survey	NUM, 4 digits	2002-2016
2	CH	BIRIMNO	Number of household	NUM, 7 digits	Last two digits HLFS's year + 5 digits
3	CH	FERT	Queue number of member	NUM, 2 digits	0 - 99
4	CH	CINSIYET	Sex	CAT, 1 digit	1. Male 2. Female
5	CH	YAS	Age completed	NUM, 2 digits	0 - 99
6	CH	YAS.K	Group of completed age	NUM, 1 digit	1. Between 15-19 ages 2. Between 20-24 ages 3. Between 25-29 ages 4. Between 30-34 ages 5. Between 35-39 ages 6. Between 40-44 ages 7. Between 45-49 age s8. Between 50-54 ages 9. Between 55-59 ages 10. Between 60-64 ages 11. 65 ages and over
7	CH	DOGUM_YER	Where were you born?	CAT, 1 digit	1. In Turkey 2. Abroad
8	CH	YURTDISI_DURUM	So far, have you ever been abroad for a period of 12 months or more?	CAT, 1 digit	1. Yes 2. No
9	CH	TR_YIL	Which year did you come to Turkey? (The year will be considered definitive return to the country.)	NUM, 4 digit	1990 - 2016

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
10	CH	BUIL_YASAMA	What kind of place where you resided before?	CAT, 1 digit	1. Province center 2 .District center 3. Subdistrict or village
11	CH	BUIL_YIL	From which year do you live in this province?	NUM, 1 digit	0 - 99
12	CH	YAKINLIK	What is your situation in the family compared to the household head?	CAT, 1 digit	1. Household head 2. Spouse 3. His/her children 4. Father / Mother 5. Brother / Sister 6. Father in law / Mother in law 7. Son in law / Daughter in law 8. Grandchild 9. Other relatives10. Non-relatives11. Housekeeper staying at home
13	CH	OKUL_BITEN_K	What is the latest educational institution/level you graduated from?	CAT, 1 digit	0- Smaller than six years old 1- Literate but not completed any educational institution 2- Primary school (5 year) 3- Secondary school, vocational school at secondary school level or primary education (8 year) 4- High school 5- Vocational or technical high school 6- Higher education (university, faculty or upper)

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
14	CH	FOET99_K	Which section did you graduate from the university last attended?	CAT, 2 digit	1-Teacher training and education science 2-Arts 3-Humanities 4-Social and behavioural science 5-Journalism and information 6-Business and administration 7-Law 8-Life science 9-Physical science 10-Mathematics and statistics11-Computing 12-Engineering and engineering trades 13-Manufacturing and processing 14-Architecture and building 15-Agriculture, forestry and fishery 16-Veterinary 17-Health18-Social services 19-Personal services 20-Transport services and environtal protection 21-Security services
15	EMP	ISCO08_ESAS_K	The appropriate code of his/her work/occupation (ISCO 08)	NUM, 2 digit	ILO ISCO-08 Codes
16	EMP	ISCO_88	The appropriate code of his/her work/occupation (ISCO 08)	NUM, 2 digit	ILO ISCO-88 Codes
17	EMP	NACE2_ESAS_K	The main activity code of establishment your other performing business (NACE Rev 2)	NUM, 2 digit	Nace Revision 2 Codes

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
18	EMP	OZEL_KAMU	What is the status of your workplace?	NUM, 1 digit	1. Private 2. Public 98. Other (Foundations, associations, cooperatives, political parties, non-governmental organizations, international organizations, embassies, etc.)
19	EMP	IS_BASLAMA_YIL	Which year did you start to work in this job/workplace?	NUM, 4 digit	1930 - 2016
20	EMP	ESAS_HAFSAAT_GENEL	Number of hours per week usually worked in the main job	NUM, 1 digit	0-99
21	EMP	ISCO.88.MERGED	The appropriate code of his/her work/occupation (ISCO 88)	NUM, 2 digit	ILO ISCO-88 Codes (Adjusted)
22	EMP	FAAL_DURUM_GECENYIL	What was your labour situation in the same month of theprevious year?	CAT, 2 digit	1- Was working2- Was working in present job3- Retired4- Unemployed (Seeking a job)5- Engaged with household chores6- Was on education or training7- Was ill or disabled8- Was in military service9- Elderly (65 years of age and over)10- Other
23	EMP	ESAS_FIILI	Number of hours actually worked in the main job during thereference week	NUM, 1 digit	0 - 99

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
24		GELIR_GECENAY_K	What is the wage income earned?	NUM, 10 digit	0-9999999999 (Deflated)
25	CH	NUTS1	Statistical Regions of Europe (SE Level 1)	NUM, 1 digit	1.TR1 İstanbul 2.TR2 West Marmara 3.TR3 Aegean 4.TR4 East Marmara 5.TR5 West Anatolia 6.TR6 Mediterranean 7.TR7 Central Anatolia8.TR8 West Black Sea 9.TR9 East Black Sea 10.TRA Northeast Anatolia 11.TRB Centraleast Anatolia 12.TRC Southeast Anatolia

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
26	CH	NUTS2	Statistical Regions of Europe (SE Level 2)	NUM, 1 digit	1.TR10 İstanbul 2.TR21 Edirne, Tekirdağ, Kırklareli 3.TR22 Balıkesir, Çanakkale 4.TR31 İzmir 5.TR32 Denizli, Aydın, Muğla 6.TR33 Manisa, Afyonkarahisar, Kütahya, Uşak 7.TR41 Bursa, Eskişehir, Bilecik 8.TR42 Kocaeli, Sakarya, Düzce, Bolu, Yalova 9.TR51 Ankara 10.TR52 Konya, Karaman11.TR61 Antalya, Isparta, Burdur 12.TR62 Adana, Mersin 13.TR63 Hatay, Kahramanmaraş, Osmaniye 14.TR71 Nevşehir, Aksaray, Niğde, Kırıkkale, Kırşehir 15.TR72 Kayseri, Sivas, Yozgat 16.TR81 Zonguldak, Karabük, Bartın 17.TR82 Kastamonu, Çankırı, Sinop 18.TR83 Samsun, Tokat, Çorum, Amasya 19.TR90 Trabzon, Ordu, Giresun, Rize, Artvin, Gümüşhane 20.TRA1 Erzurum, Erzincan, Bayburt 21.TRA2 Kars, Ağrı, Iğdır, Ardahan 22.TRB1 Malatya, Elazığ, Bingöl, Tunceli 23.TRB2 Van, Muş, Bitlis, Hakkari 24.TRC1 Gaziantep, Adıyaman, Kilis 25.TRC2 Diyarbakır, Şanlıurfa 26.TRC3 Siirt, Mardin, Batman, Şırnak

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
27	CH	T1		NUM, 1 digit	1. Employed 2. Unemployed 3. Not in labour force
28	EMP	ISTEKI_DURUM_K	What is your employment status for the previous year?	NUM, 1 digit	1. Regular employee or casual 2. Employer 3. Self employed4. Unpaid family worker
29	CH,DE	EDUC_YEAR	Sum of years spent on education.	NUM, 2 digit	0-99
30	CH,DE	EXPERIENCE, EXPERIENCE2	Individual experience on working and its square	NUM, 2 digit	0-9999
31	CH	ONCEKI_IKAMET	Please indicate the province where you resided before camethis city?	NUM, 1 digit	1-In Turkey 2-In a foreign country
32	CH	YERLESIM_TUR	The place where you resided before was	NUM, 1 digit	1-Province center 2-District center 3-Subdistrict or a village
33	CH	FERTNO_ES	Relationship to reference person in the household	NUM, 1 digit	1-Reference person 2-Spouse (or cohabiting partner) 3-Child of reference person 4-Bride or bridegroom 5-Grandchild6-Mother/father in law 7-Other relatives8-Non-relatives
34	CH	FERTNO_ANNE	Specify the sequence number of your own mother (if household member)	NUM, 2 digit	0-20

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
35	CH	FERTNO_BABA	Specify the sequence number of your own father (if household member)	NUM, 2 digit	0-20
36	CH	OKUR_YAZAR	Are you literate?	CAT, 1 digit	1. Yes 2. No
37	CH	EGITIM_DEVAM_K	Attendance to regular education (schooling)	CAT, 1 digit	1. Yes 2. No
38	CH	OKUL_DEVAM_K	Level of the education	CAT, 1 digit	1- Primary education (8 year) 2- High school 3- Vocational or technical high school 4- Open education system 5- Faculty/high school 6- Master,doctorate
39	CH	SINIF_DEVAM	Which class are you in?	NUM, 1 digit	0-9
40	CH	KURS	Did you attend any courses, seminars, conferences or receiveprivate lessons or instructions outside the regular educationsystem within the last four weeks?	CAT, 1 digit	1. Yes 2. No
41	CH	KURS.AMACI	What was the purpose of this training you attended?	CAT, 1 digit	1. For Carrier 2. For Personal or Social 3. Other
42	CH	MEDENI_DURUM	Current maritial status of individual sureyed	CAT, 1 digit	1-Single (never married) 2-Married 3-Divorced 4-Widowed

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
43	EMP	CALISMA	Did you work to earn income in cash or in kind in thereference week?	CAT, 1 digit	1. Yes 2. No
44	EMP	CALISMA_BIR_SAAT	Did you work as paid or unpaid worker (even if you were ahousewife, student or retired person) even for one hour inthe reference week?	CAT, 1 digit	1. Yes 2. No
45	EMP	CALISMA_SAHIP_IS	Did you have a business from which you were temporarilyabsent in the reference week or a job that you'll return back?	CAT, 1 digit	1. Yes 2. No
46	EMP	CALISMAMA_NEDEN_REF	Why were you absent from work in the reference week?	CAT, 1 digit	1. Yes 2. No
47	CH	DEFLATOR	Deflator of Survey Year	NUM, 1 digit	0 - 1
48	CH	ID	(YIL*1000000) + (BIRIMNO*1000) + FERT	NUM, 10 digit	0 - 9999999999
49	CH	AGIRLIK_KATSAYISI	Coefficient of weight		

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
50	CH	HH_BUYUKLUK_ESTIMATED	Household size (Total number of members in household, estimated from surveys)	NUM, 2 digit	0-99
51	CH	HH_BUYUKLUK	Household size (Total number of members in household)	NUM, 2 digit	0-99
52	EMP	ISYERI_DURUM	Type of your workplace	CAT, 1 digit	1. Field, garden 2. Regular workplace (Factory, office, shop or etc.) 3. Market place 4. Mobile or irregular workplace 5. At home (His/her or other's home)
53	EMP	CALISAN_SAYI_HH	How many people are working in there?	CAT, 1 digit	1. 10 or less than 10 2. 11-19 3. 20-49 4. 50 or more than 50 5. Don't know, but more than 10
54	EMP	IDARI_SORUMLU	Do you have administrative responsibilities in your job?	CAT, 1 digit	1. Yes 2. No
55	EMP	KAYITLILIK	Are you registered in the Social Security Institution due to your job?	CAT, 1 digit	1. Yes 2. No
56	EMP	ISBUL_YONTEM	How did you find your job?	CAT, 1 digit	1. By myself 2. Through the Turkish Employment Organisation 3. Through the Private Employment Office 4. Through relatives, friends etc. 5. Other (Specify)

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
57	EMP	CALISMA_SEKLI	What is your employment type in this workplace?	CAT, 1 digit	1. Full time 2. Part time
58	EMP	YARIZAMAN_NEDEN	Why are you working in a part time job?	CAT, 1 digit	1. Looking after children in the family 2. Looking after incapacitated adults 3. Looking after children or incapacitated adults in the family 4. Training/education 5. Own illness or disability 6. Other personal or family reasons 7. Could not find a full time job 8. Nature of work 9. Other
59	EMP	IS_SUREKLILIK	What is the permanency of this job?	CAT, 1 digit	1. Permanent 2. Temporary or limited-term employment (Seasonal, daily, under contract, without any contract, occasional job included)
60	EMP	GECICI_NEDEN	Why are you not working in permanent job?	CAT, 1 digit	1. Could not find a permanent job 2. Did not want a permanent job 3. As the contract covers training period (trainee or apprentice etc.) 4. Because of his/her trial period 5. Other
61	EMP	EVDE_CAL_SIKLIK	Are you performing all or a portion of your business in your home?	CAT, 1 digit	1. Usually 2. Sometimes 3. Never

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
62	EMP	EKIS_DURUM	In addition to your main job, during the reference week, did you work any other job or business to earn cash or in kind income or did you have a job that you were temporarily absent?	CAT, 1 digit	1. Yes 2. No
63	EMP	NACE2_EKIS_K	The main activity code of establishment your other performing business (NACE Rev 2)	CAT, 1 digit	1. Agriculture 2. Industry 3. Trade 4. Services
64	EMP	ISTEKI_DURUM_EKIS	What is your employment status in your other workplace (firm/organisation)?	CAT, 1 digit	1. Regular or casual employee 2. Employer 3. Self employed 4. Unpaid family worker
65	EMP	EKIS_FIILI	How many hours did you actually work in your additional job during the reference week?	CAT, 1 digit	
66	EMP	FAZLACAL_NEDEN	Why did you work in your main job more than your usual hours during the reference week?	CAT, 1 digit	1. Variable working hours (flexible working hours) 2. Overtime 3. Other

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
67	EMP	AZCAL_NEDEN	Why did you work in your main job less than your usual hours during the reference week?	CAT, 1 digit	1. Nature of work 2. Annual leave etc. 3. Public holidays 4. There was no work 5. Personal or family reasons 6. Bad weather 7. Own illness, injury or temporary disability 8. Slack work for technical and economic reasons 9. Labour dispute (strike, lock-out, etc.) 10. Education and training11. Variable working hour s12. Maternity leave 13. Start of/change in job during the reference week 14. Work finished in the reference week 15. Other
68	EMP	FAZLACAL_ISTEK	Would you like to work more hours than you actually worked in the reference week to get extra income?	CAT, 1 digit	1. Yes 2. No
69	EMP	FAZLACAL_SEKIL	In which way you would like to work more?	CAT, 1 digit	1. In present job(s) 2. In another job(s) 3. In additional job(s) 4. it does not matter

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
70	EMP	FAZLACAL_BASLAMA	If you had an opportunity to work more hours, could/can you start to work by this new working plan within the reference week or the following 2 weeks?	CAT, 1 digit	1. Yes 2. No
71	EMP	EKIS_ARA_DURUM	Were you looking for another job to replace your present job or as an additional job wıthın last 4 weeks endıng wıth the week reference week?	CAT, 1 digit	1. Yes 2. No

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
72	EMP	EKIS_ARA_NEDEN	Why did/are you seek(ing) another job?	CAT, 1 digit	1. To earn more money without working more 2. Wish to work in another job in addition to his/her current job 3. Wish to work in a different job with more hours than the present job 4. Wish to work in a different job with less hours than the present job 5. Not pleased with working conditions (health and safety conditions of workplace) 6. Not working in his/her usual profession 7. There is a risk of losing his/her job 8. His/her work is temporary or seasonal 98. Other
73	EMP	ARA_EKIS_DURUM	Were you seeking a job as regular employee or do you want to set up your own business?	CAT, 1 digit	1. Want to set up his/her own business2. Seeking a job as regular employee
74	EMP	ARA_EKIS_SEKIL	Were you seeking a full time or part time job?	CAT, 1 digit	1. Full time 2. Full time job, but if he/she cannot find, available part time job 3. Part time 4. Part time, but if he/she cannot find, available full time job 5. Full time or part time job would be accepted

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
75	EMP	EKIS_ARA_SURE	How many months have you been looking for another job to replace your present job or as an additional job? (For less than one month, give code "0")	CAT, 1 digit	0 - 12
76	EMP	EKIS_ARA_KANAL_K	Within last 4 weeks ending with the reference week, did you do anything to seek a job?	CAT, 1 digit	1. Yes 2. No
77	EMP	ISBASI_EKIS	If you had found a job, would you have been able to start working within 2 weeks following the reference week?	CAT, 1 digit	1. Yes2. No
78	UNEMP	ISARA_DURUM	Did you search for a job or have any effort to build your own business ending with the reference week for the last 4 weeks?	CAT, 1 digit	1. Yes 2. No

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
79	UNEMP	ISARA_BASLA_DURUM	What was your situation when you started to look for a job?	CAT, 2 digit	1. Working in a temporary job / work was ended 2. Dismissed 3. Left his/her job by own initiative 4. Closed his/her job or bankrupted 5. Was working as unpaid family worker 6. Retired 7. Was in regular education or just graduated 8. Was in course or any education programme 9. Just completed his military service 10. Busied with household chores98. Other
80	UNEMP	ISARA_KANAL1	Did you apply to employer directly?	CAT, 1 digit	1. Yes 2. No
81	UNEMP	ISARA_KANAL2	Did you ask friends, relatives etc.?	CAT, 1 digit	1. Yes 2. No
82	UNEMP	ISARA_KANAL3	Did you contact with Turkish Employment Office?	CAT, 1 digit	1. Yes 2. No
83	UNEMP	ISARA_KANAL4	Did you contact with private employment agencies?("kariyer.net", "yenibiris.com" etc. in internet)	CAT, 1 digit	1. Yes 2. No
84	UNEMP	ISARA_KANAL5	Did you study advertisements in newspapers or journals?	CAT, 1 digit	1. Yes 2. No

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
85	UNEMP	ISARA_KANAL6	Did you insert or answer advertisements in newspapers?	CAT, 1 digit	1. Yes 2. No
86	UNEMP	ISARA_KANAL7	Did you take a test, interview or examination?	CAT, 1 digit	1. Yes 2. No
87	UNEMP	ISARA_KANAL8	Did you look for land, premises or equipment to establish your own business?	CAT, 1 digit	1. Yes 2. No
88	UNEMP	ISARA_KANAL9	Did you look for permits, licences or financial resources to establish your own business?	CAT, 1 digit	1. Yes 2. No
89	UNEMP	ISARA_KANAL10	Did you wait a call from Turkish Employment Office?	CAT, 1 digit	1. Yes 2. No
90	UNEMP	ISARA_KANAL11	Did you wait the results of an application for a job?	CAT, 1 digit	1. Yes 2. No
91	UNEMP	ISARA_KANAL12	Did you wait the results of a competition for recruiment to the public sector?	CAT, 1 digit	1. Yes 2. No
92	UNEMP	ISARA_KANAL13	Did you use any other methods to find a job?	CAT, 1 digit	1. Yes 2. No

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
93	UNEMP	ISCO08_ARA_K	Code of business or profession sought (ISCO 08)	CAT, 1 digit	1. Managers 2. Professionals 3. Technicians and associate professionals 4. Clerical support workers5. Service and sales workers 6. Skilled agricultural, forestry and fishery workers 7. Craft and related trades workers 8. Plant and machine operators, and assemblers 9. Elemantary occupations
94	UNEMP	ARA_IS_DURUM	Were you seeking a job as regular employee or want to establish your own business?	CAT, 1 digit	1. Want to set up his/her own business 2. Seeking a job as regular employee
95	UNEMP	ARA_IS_SEKIL	Were you seeking a full time or part time job?	CAT, 1 digit	1. Full time 2. Full time, but if not available part time job will be accepted 3. Part time 4. Part time, but if not available full time job will be accepted 5. It dosen't matter. Full time or part time job will be accepted
96	UNEMP	ISARA_SURE	How many months have you been seeking a job?(For less than one month, give code "0")	NUM, 2 digit	0 - 12

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
97	UNEMP	ISARAMAMA_NEDEN	What is your reason that you have not been seeking a job during the last 4 weeks ending with thereference week?	CAT, 2 digit	1. Found-establish a job, but waiting to start 2. Working seosonally / Waiting recall from previous job 3. Made too much effort seeking before but could not found 4. Do not believe in finding a job appropriate for his/her skill/qualification 5. Continuing to his/her education or training6.1. Engaged with household chores (cleaning, washing, cooking and ironing etc.)6.2. Looking after children in the family6.3. Looking after incapacitated adults in the family 6.4. Both Looking after children and incapacitated adults in the family 6.5. Other personal or family reasons 7. Retired (Including early retirement) 8. Disabled or sick 9. Elderly (Not retired, but thinking old for working, 65 years and over) 98. Other
98	UNEMP	BAKIM_NEDEN_ISSIZ	Why are you looking after children or incapacitated adults in the family by yourself?	CAT, 1 digit	1. Because these kind of services are too expensive 2. Because of distrust to the quality of that kind services 3. There is no place which gives that kind of services near home 4. Individual preference98. Other

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
99	UNEMP	ISBASI_ISSIZ	If you had a found a job or established your own business, would you have been able to start working within 2 weeks following the reference week?	CAT, 1 digit	1. Yes 2. No
100	PAST	GECMIS_IS	Have you ever worked before?	CAT, 1 digit	1. Yes 2. No
101	PAST	IS_AYRIL_YIL	Which year did you leave from your latest job?	NUM, 2 Digit	0 - 12
102	PAST	IS_AYRIL_AY	Which month did you leave from your latest job?	NUM, 4 Digit	1930 - 2016
	PAST,DE	INACTIVE	YIL-IS_AYRIL_YIL	NUM, 2 Digit	0 - 99
103	PAST	IS_AYRIL_NEDEN	What was the main reason for leaving from your most recent job?	CAT, 1 digit	1. Job was temporary, came to an end 2. Was working seasonally 3. Dismissed/Liquidated/Bankrupted 4. Not satisfied with job 5. Own illness or disability 6. Looking after children or incapacitated adults in the family 7. Her spouse's request / Due to marriage 8. Education or training 9. Retirement (including early retirement) 10. Military service98. Other

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
104	PAST	NACE2_GECMIS_K	What was the code of main activity of this workplace, firm or organisation? (NACE Rev 2)	CAT, 1 digit	1. Agriculture 2. Manufacturing, mining and quarrying and other industries 3. Construction 4. Wholesale retail trade, transportation and storage, accommodation and service 5. Information and communication 6. Financial and insurance activities 7. Real estate activities 8. Professional, scientific and technical activities, administrative and support services 9. Public administration and defense, education, health and social services 10. Other service activities11. Unknown
105	PAST	ISCO08_GECMIS_K	Code of most recent work (ISCO 08)	CAT, 1 digit	1. Managers 2. Professionals 3. Technicians and associate professionals 4. Clerical support workers 5. Service and sales workers6. Skilled agricultural, forestry and fishery workers 7. Craft and related trades workers 8. Plant and machine operators, and assemblers 9. Elemantary occupations
106	PAST	ISTEKI_DURUM_GECMIS	What was your employment status?	CAT, 1 digit	1. Regular employee or casual 2. Employer 3. Self employed 4. Unpaid family worker

Table B.6 (Continued) : Household Labor Force Survey.

No	CAT	VAR.NAME	EXPLANATION	TYPE*	VALUES
107	PAST	NACE2_GECENYIL_K	Main activity code of your organization or work for the previous year (NACE Rev 2)	CAT, 1 digit	1. Agriculture 2. Manufacturing, mining and quarrying and other industries 3. Construction 4. Wholesale retail trade, transportation and storage, accommodation and service 5. Information and communication 6. Financial and insurance activities 7. Real estate activities 8. Professional, scientific and technical activities, administrative and support services 9. Public administration and defense, education, health and social services 10. Other service activities11. Unknown
108	PAST	ISTEKI_DURUM_GECENYIL	What is your employment status for the previous year?	CAT, 1 digit	1. Regular employee or casual 2. Employer 3. Self employed 4. Unpaid family worker
109	EMP	INCOME.DEF	Deflated Monthly Income	NUM, 2 digit	0-9999999

*NUM: Numerical, CAT: Categorical, ORD: Ordinal.

3.1 R Codes Used to Merge HLFSs

```
## -*- coding: utf-8 -*-
#"""
#Created on Fri Sep 08 18:47:52 2017

#######################################################################
#                          _          _              _            _    #
#     _    _    __ _   ___ (_) _ __   | | __ _    _ | |_  _    _ | | __     #
#    | |  | |  / _  |/ __|| || '_ \  | |/ /| |  | || __|| |  | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    <  | |_| || |_ | |_| ||    <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     _____                                 _  _                      #
#    / ___ \    __ _   _ __ ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_    _ \   / _  || || |    / __|/ _ \ | '_    _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | |  #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \_____/  |___/                                                  #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web    : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################

# Yasin Kütük
# HLFS Soru ve yanıtları uyumlaştırma sistemi
print("HLFS Soru ve yanıtları uyumlaştırma sistemi")
# R Kodları


options("scipen"=10,digits=20)


####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,"systemfit",
              "foreign", "feather", "dplyr",
              "sampleSelection","systemfit",
              "openxlsx") #2000-03 arasındaki STATA dosyalarını okumak için
paketkurulum(paketler)



####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
}
loadlib(x)



####03.Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle


path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/"
prevpath <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/"

veriler <- list.files(path=path)
veriler <- veriler[grep('rdata', veriler)]
```

```
for(i in veriler){
 l <- paste0(path,i)
  load(l)
}



####04.Grand Merger####
env <- ls()
hia <- env[grep('hlfs',env,fixed=T)]



####Değişkenlerin adlarını küçük harf yap####
for(i in hia){
  sil <- get(i)
  names(sil) <- tolower(colnames(sil))
  assign(i,sil)
  rm(sil)
}

####Integer ve Numeric Dönüşümü####
numer <- function(x){as.numeric(as.character(x))}
integ <- function(x){as.integer(as.character(x))}



#Kontrol Parametresi
toplamuzunluk <- sum(length(hlfs2016[,1]),
                     length(hlfs2015[,1]),
                     length(hlfs2014[,1]),
                     length(hlfs2013[,1]),
                     length(hlfs2012[,1]),
                     length(hlfs2011[,1]),
                     length(hlfs2010[,1]),
                     length(hlfs2009[,1]),
                     length(hlfs2008[,1]),
                     length(hlfs2007[,1]),
                     length(hlfs2006[,1]),
                     length(hlfs2005[,1]),
                     length(hlfs2004[,1]),
                     length(hlfs2003[,1]),
                     length(hlfs2002[,1]),
                     length(hlfs2001[,1]),
                     length(hlfs2000[,1]))

#___UZUNLAYICILAR####
h16 <- length(hlfs2016$birimno)
h15 <- length(hlfs2015$birimno)
h14 <- length(hlfs2014$birimno)
h13 <- length(hlfs2013$formno)
h12 <- length(hlfs2012$formno)
h11 <- length(hlfs2011$formno)
h10 <- length(hlfs2010$formno)
h9 <- length(hlfs2009$formno)
h8 <- length(hlfs2008$formno)
h7 <- length(hlfs2007$formno)
h6 <- length(hlfs2006$formno)
h5 <- length(hlfs2005$formno)
h4 <- length(hlfs2004$formno)
h3 <- length(hlfs2003$formno)
h2 <- length(hlfs2002$formno)
h1 <- length(hlfs2001$formno)
h0 <- length(hlfs2000$formno)



#___Yıllar####
full <- data.frame(matrix(NA, nrow = toplamuzunluk))
full[,1] <- data.frame(c(rep(2016,length(hlfs2016$birimno)),
                         rep(2015,length(hlfs2015$birimno)),
                         rep(2014,length(hlfs2014$birimno)),
                         rep(2013,length(hlfs2013$formno)),
                         rep(2012,length(hlfs2012$formno)),
```

```
                          rep(2011,length(hlfs2011$formno)),
                          rep(2010,length(hlfs2010$formno)),
                          rep(2009,length(hlfs2009$formno)),
                          rep(2008,length(hlfs2008$formno)),
                          rep(2007,length(hlfs2007$formno)),
                          rep(2006,length(hlfs2006$formno)),
                          rep(2005,length(hlfs2005$formno)),
                          rep(2004,length(hlfs2004$formno)),
                          rep(2003,length(hlfs2003$formno)),
                          rep(2002,length(hlfs2002$formno)),
                          rep(2001,length(hlfs2001$formno)),
                          rep(2000,length(hlfs2000$formno))))
colnames(full)[1] <- "yil"
full[,1] <- integ(full[,1])

#Kontrol
length(full[,1])==toplamuzunluk


#___Form No####
full[,2] <- data.frame(c(hlfs2016$birimno,
                         hlfs2015$birimno,
                         hlfs2014$birimno,
                         hlfs2013$formno,
                         hlfs2012$formno,
                         hlfs2011$formno,
                         hlfs2010$formno,
                         hlfs2009$formno,
                         hlfs2008$formno,
                         hlfs2007$formno,
                         hlfs2006$formno,
                         hlfs2005$formno,
                         hlfs2004$formno,
                         hlfs2003$formno,
                         hlfs2002$formno,
                         hlfs2001$formno,
                         hlfs2000$formno))
colnames(full)[2] <- "birimno"
full[,2] <- integ(full[,2])
#Kontrol
length(full[,2])==toplamuzunluk


#___Sıra No####
full[,3] <- data.frame(c(hlfs2016$fertno,
                         hlfs2015$fertno,
                         hlfs2014$fertno,
                         hlfs2013$s1,
                         hlfs2012$s1,
                         hlfs2011$s1,
                         hlfs2010$s1,
                         hlfs2009$s1,
                         hlfs2008$s1,
                         hlfs2007$s1,
                         hlfs2006$s1,
                         hlfs2005$s1,
                         hlfs2004$s1,
                         hlfs2003$s1,
                         hlfs2002$s1,
                         hlfs2001$s1,
                         hlfs2000$s1))
colnames(full)[3] <- "fert"
full[,3] <- integ(full[,3])
#Kontrol
length(full[,3])==toplamuzunluk



#___Cinsiyet####
full[,4] <- data.frame(c(hlfs2016$cinsiyet,
                         hlfs2015$cinsiyet,
                         hlfs2014$cinsiyet,
                         hlfs2013$s3,
                         hlfs2012$s3,
```

```
                          hlfs2011$s3,
                          hlfs2010$s3,
                          hlfs2009$s3,
                          hlfs2008$s3,
                          hlfs2007$s3,
                          hlfs2006$s3,
                          hlfs2005$s3,
                          hlfs2004$s3,
                          hlfs2003$s3,
                          hlfs2002$s3,
                          hlfs2001$s3,
                          hlfs2000$s3))
colnames(full)[4] <- "cinsiyet"
full[,4] <- integ(full[,4])
#Kontrol
length(full[,4])==toplamuzunluk



#___Yaş####
full[,5] <- data.frame(c(hlfs2016$yas,
                          hlfs2015$yas,
                          hlfs2014$yas,
                          hlfs2013$s6,
                          hlfs2012$s6,
                          hlfs2011$s6,
                          hlfs2010$s6,
                          hlfs2009$s6,
                          hlfs2008$s6,
                          hlfs2007$s6,
                          hlfs2006$s6,
                          hlfs2005$s6,
                          hlfs2004$s6,
                          rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[5] <- "yas"
full[,5] <- integ(full[,5])


#___Yaş Grubu####
full[,6] <- data.frame(c(hlfs2016$yas_k,
                                    hlfs2015$yas_k,
                                    hlfs2014$yas_k,
                                    hlfs2013$s6_grup,
                                    hlfs2012$s6_grup,
                                    hlfs2011$s6_grup,
                                    hlfs2010$s6_grup,
                                    hlfs2009$s6_grup,
                                    hlfs2008$s6_grup,
                                    hlfs2007$s6_grup,
                                    hlfs2006$s6_grup,
                                    hlfs2005$s6_grup,
                                    hlfs2004$s6_grup,
                                    hlfs2003$s4,
                                    hlfs2002$s4,
                                    hlfs2001$s4,
                                    hlfs2000$s4))
colnames(full)[6] <- "yas.k"

full$yas.k[full$yil>2013 & full$yas.k==8] <- 14
full$yas.k[full$yil>=2014 & full$yas.k==4] <- 5
full$yas.k[full$yil>=2014 & full$yas.k==3] <- 4
full$yas.k[full$yil>=2014 & full$yas.k==51] <- 6
full$yas.k[full$yil>=2014 & full$yas.k==52] <- 7
full$yas.k[full$yil>=2014 & full$yas.k==61] <- 8
full$yas.k[full$yil>=2014 & full$yas.k==62] <- 9
full$yas.k[full$yil>=2014 & full$yas.k==63] <- 10
full$yas.k[full$yil>=2014 & full$yas.k==64] <- 11
full$yas.k[full$yil>=2014 & full$yas.k==71] <- 12
full$yas.k[full$yil>=2014 & full$yas.k==72] <- 13


full[,6] <- integ(full[,6])
```

```
#Kodlama aşağıdaki gibidir
#1- 0 0 -0 4 y a ş a ra s ı
#2- 0 5 -1 1 y a ş a ra s ı
#3- 1 2 -1 4 y a ş a ra s ı
#4- 1 5 -1 9 y a ş a ra s ı
#5- 2 0 -2 4 y a ş a ra s ı
#6- 2 5 -2 9 y a ş a ra s ı
#7- 3 0 -3 4 y a ş a ra s ı
#8- 3 5 -3 9 y a ş a ra s ı
#9- 4 0 -4 4 y a ş a ra s ı
#10 - 4 5 -4 9 y a ş a ra s ı
#11 - 5 0 -5 4 y a ş a ra s ı
#12 - 5 5 -5 9 y a ş a ra s ı
#13 - 6 0 -6 4 y a ş a ra s ı
#14 - 6 5 + y a ş

#___Doğum Yeri####
full[,7] <- data.frame(c(hlfs2016$dogum_yer,
                         hlfs2015$dogum_yer,
                         hlfs2014$dogum_yer,
                         hlfs2013$s7,
                         hlfs2012$s7,
                         hlfs2011$s7,
                         hlfs2010$s7,
                         hlfs2009$s7,
                         rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[7] <- "dogum_yer"
full[,7] <- integ(full[,7])


#___Yurtdışı En az 6 AY####
full[,8] <- data.frame(c(hlfs2016$yurtdisi_durum,
                         hlfs2015$yurtdisi_durum,
                         hlfs2014$yurtdisi_durum,
                         hlfs2013$s8a,
                         hlfs2012$s8a,
                         hlfs2011$s8a,
                         hlfs2010$s8a,
                         hlfs2009$s8a,
                         rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[8] <- "yurtdisi_durum"
#Cevap uyumsuzluğu var aslında ama 2010 sonrasında
#112 ay yaşayanları 6 ay ayaşamız kabul ederek eşleştirdim.

full[,8] <- integ(full[,8])

#___TR Geliş Yılı####
full[,9] <- data.frame(c(hlfs2016$tr_yil,
                         hlfs2015$tr_yil,
                         hlfs2014$tr_yil,
                         hlfs2013$s8b,
                         hlfs2012$s8b,
                         hlfs2011$s8b,
                         hlfs2010$s8b,
                         hlfs2009$s8b,
                         rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[9] <- "tr_yil"
full[,9] <- integ(full[,9])

#___Yaşadığı Yer####
full[,10] <- data.frame(c(hlfs2016$buil_yasama,
                          hlfs2015$buil_yasama,
                          hlfs2014$buil_yasama,
                          hlfs2013$s9a,
                          hlfs2012$s9a,
                          hlfs2011$s9a,
                          hlfs2010$s9a,
                          hlfs2009$s9a,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[10] <- "buil_yasama"
full[,10] <- integ(full[,10])

#___Yaşadığı Yer Yılı####
full[,11] <- data.frame(c(hlfs2016$buil_yil,
```

```
                          hlfs2015$buil_yil,
                          hlfs2014$buil_yil,
                          hlfs2013$s9b,
                          hlfs2012$s9b,
                          hlfs2011$s9b,
                          hlfs2010$s9b,
                          hlfs2009$s9b,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[11] <- "buil_yil"
full[,11] <- integ(full[,11])

#___Referans Kişi Yakınlık####
full[,12] <- data.frame(c(hlfs2016$yakinlik,
                          hlfs2015$yakinlik,
                          hlfs2014$yakinlik,
                          hlfs2013$s11,
                          hlfs2012$s11,
                          hlfs2011$s11,
                          hlfs2010$s11,
                          hlfs2009$s11,
                          hlfs2008$s11,
                          hlfs2007$s11,
                          hlfs2006$s11,
                          hlfs2005$s11,
                          hlfs2004$s7,
                          hlfs2003$s7,
                          hlfs2002$s7,
                          hlfs2001$s7,
                          hlfs2000$s7))
colnames(full)[12] <- "yakinlik"
full[,12] <- integ(full[,12])


#___Okul Durumu####
#DİKKAT-2014 şıkları farklı#, indirgeme yapılacak###
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==5] <- 6
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==4] <- 5
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==31] <- 4
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==32] <- 4
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==3] <- 3
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==2] <- 2
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==1] <- 2
hlfs2014$okul_biten_k_gecici[hlfs2014$okul_biten_k==0] <- 0

hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==5] <- 6
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==4] <- 5
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==31] <- 4
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==32] <- 4
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==3] <- 3
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==2] <- 2
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==1] <- 2
hlfs2015$okul_biten_k_gecici[hlfs2015$okul_biten_k==0] <- 0

hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==5] <- 6
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==4] <- 5
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==31] <- 4
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==32] <- 4
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==3] <- 3
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==2] <- 2
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==1] <- 2
hlfs2016$okul_biten_k_gecici[hlfs2016$okul_biten_k==0] <- 0


full[,13] <- data.frame(c(hlfs2016$okul_biten_k_gecici,
                          hlfs2015$okul_biten_k_gecici,
                          hlfs2014$okul_biten_k_gecici,
                          hlfs2013$s13,hlfs2012$s13,
                          hlfs2011$s13,
                          hlfs2010$s13,
                          hlfs2009$s13,
                          hlfs2008$s14,
                          hlfs2007$s14,
                          hlfs2006$s14,
                          hlfs2005$s14,
```

```
                          hlfs2004$s10,
                          hlfs2003$s9,
                          hlfs2002$s9,
                          hlfs2001$s9,
                          hlfs2000$s9))
colnames(full)[13] <- "okul_biten_k"
full[,13] <- integ(full[,13])

hlfs2014 <- hlfs2014[,-106]
hlfs2015 <- hlfs2015[,-103]
hlfs2016 <- hlfs2016[,-104]



#___Okul Bölümü####
full[,14] <- data.frame(c(hlfs2016$foet99_k,
                          hlfs2015$foet99_k,
                          hlfs2014$foet99_k,
                          hlfs2013$s15kod,
                          hlfs2012$s15kod,
                          hlfs2011$s15kod,
                          hlfs2010$s15kod,
                          hlfs2009$s15kod,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[14] <- "foet99_k"
full[,14] <- integ(full[,14])




#___ISCO-08######
#DİKKAT- 2014 ve 2012'de isco08 var, diğer yıllar isco88,
#2014-isco08'in dönüştürülmesi gerekiyor#
full[,15] <- data.frame(c(hlfs2016$isco08_esas_k,
                          hlfs2015$isco08_esas_k,
                          hlfs2014$isco08_esas_k,
                          hlfs2013$s38kod,
                          hlfs2012$s38kod08,
                          rep(NA,sum(h11,h10,h9,h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[15] <- "isco08_esas_k"
full[,15] <- integ(full[,15])



#___ISCO-88######
#DİKKAT- 2014 ve 2012'de isco08 var, diğer yıllar isco88,
#2014-isco08'in dönüştürülmesi gerekiyor#
full[,16] <- data.frame(c(rep(NA,h14+h15+h16),
                          hlfs2013$s38kod,
                          hlfs2012$s38kod88,
                          hlfs2011$s38kod,
                          hlfs2010$s38kod,
                          hlfs2009$s38kod,
                          hlfs2008$s38kod,
                          hlfs2007$s38kod,
                          hlfs2006$s38kod,
                          hlfs2005$s38kod,
                          hlfs2004$s33kod,
                          hlfs2003$s22kod,
                          hlfs2002$s22kod,
                          hlfs2001$s22kod,
                          rep(NA,h0)))
colnames(full)[16] <- "isco_88"
full[,16] <- integ(full[,16])

#___Nace Rev2####
full[,17] <- data.frame(c(hlfs2016$nace2_esas_k,
                          hlfs2015$nace2_esas_k,
                          hlfs2014$nace2_esas_k,
                          hlfs2013$s33kod,
                          hlfs2012$s33kod,
                          hlfs2011$s33kod,
                          hlfs2010$s33kod,
```

```
                            hlfs2009$s33kodr2,
                            hlfs2008$s33kodr2,
                            hlfs2007$s33kodr2,
                            hlfs2006$s33kodr2,
                            hlfs2005$s33kodr2,
                            hlfs2004$s28kodr2,
                            rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[17] <- "nace2_esas_k"
full[,17] <- integ(full[,17])

#___İşyeri Statü####
full[,18] <- data.frame(c(hlfs2016$ozel_kamu,
                            hlfs2015$ozel_kamu,
                            hlfs2014$ozel_kamu,
                            hlfs2013$s34,
                            hlfs2012$s34,
                            hlfs2011$s34,
                            hlfs2010$s34,
                            hlfs2009$s34,
                            rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[18] <- "ozel_kamu"
full[,18] <- integ(full[,18])


#___İşyerine Başlama Yılı####
full[,19] <- data.frame(c(hlfs2016$is_baslama_yil,
                            hlfs2015$is_baslama_yil,
                            hlfs2014$is_baslama_yil,
                            hlfs2013$s43a,
                            hlfs2012$s43a,
                            hlfs2011$s43a,
                            hlfs2010$s43a,
                            hlfs2009$s43a,
                            hlfs2008$s44a,
                            hlfs2007$s44a,
                            hlfs2006$s44a,
                            hlfs2005$s44a,
                            hlfs2004$s38a,
                            rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[19] <- "is_baslama_yil"
full[,19] <- integ(full[,19])



#___Çalışma_Süresi####
full[,20] <- data.frame(c(hlfs2016$esas_hafsaat_genel,
                            hlfs2015$esas_hafsaat_genel,
                            hlfs2014$esas_hafsaat_genel,
                            hlfs2013$s55a,
                            hlfs2012$s55a,
                            hlfs2011$s55a,
                            hlfs2010$s55a,
                            hlfs2009$s43a,
                            hlfs2008$s61a,
                            hlfs2007$s61a,
                            hlfs2006$s61a,
                            hlfs2005$s61a,
                            hlfs2004$s54a,
                            hlfs2003$s27a,
                            hlfs2002$s27a,
                            hlfs2001$s27a,
                            hlfs2000$s27a))
colnames(full)[20] <- "esas_hafsaat_genel"
full[,20] <- integ(full[,20])


#ISCO_CONVERT_PROCESS####
isco <- read.table(file="/media/DRIVE/Dropbox/_VeriTabanlarim/isco_convert.txt",
                   sep = "\t", header=T)
i08 <- hlfs2012$s38kod08
i88 <- hlfs2012$s38kod88
nace2 <- hlfs2012$s33kod
```

```
#2014 isco08 ile 2012 isco08 aynı mı?#
sort(unique(hlfs2014$isco08_esas_k))==sort(unique(hlfs2012$s38kod08))


iscoset1 <- data.frame(i08,i88,nace2)
iscoset2 <- data.frame(i08,i88)


#View(unique(iscoset[iscoset$i08==11,]))
#View(unique((iscoset1)))
#View(unique((iscoset2)))


sort(unique(hlfs2014$isco08_esas_k))==sort(unique(hlfs2012$s38kod08))


full$isco.88.2 <- ifelse(full$isco08_esas_k==11,11,
                         ifelse(full$isco08_esas_k==12,12,
                         ifelse(full$isco08_esas_k==13,13,
                         ifelse(full$isco08_esas_k==14,13,
                         ifelse(full$isco08_esas_k==21,21,
                         ifelse(full$isco08_esas_k==22,22,
                         ifelse(full$isco08_esas_k==23,23,
                         ifelse(full$isco08_esas_k==24,24,
                         ifelse(full$isco08_esas_k==25,21,
                         ifelse(full$isco08_esas_k==26,24,
                         ifelse(full$isco08_esas_k==31,31,
                         ifelse(full$isco08_esas_k==32,32,
                         ifelse(full$isco08_esas_k==33,34,
                         ifelse(full$isco08_esas_k==34,34,
                         ifelse(full$isco08_esas_k==35,31,
                         ifelse(full$isco08_esas_k==41,41,
                         ifelse(full$isco08_esas_k==42,42,
                         ifelse(full$isco08_esas_k==43,41,
                         ifelse(full$isco08_esas_k==44,41,
                         ifelse(full$isco08_esas_k==51,51,
                         ifelse(full$isco08_esas_k==52,52,
                         ifelse(full$isco08_esas_k==53,51,
                         ifelse(full$isco08_esas_k==54,51,
                         ifelse(full$isco08_esas_k==61,61,
                         ifelse(full$isco08_esas_k==62,61,
                         ifelse(full$isco08_esas_k==63,62,
                         ifelse(full$isco08_esas_k==71,71,
                         ifelse(full$isco08_esas_k==72,72,
                         ifelse(full$isco08_esas_k==73,73,
                         ifelse(full$isco08_esas_k==74,72,
                         ifelse(full$isco08_esas_k==75,74,
                         ifelse(full$isco08_esas_k==81,82,
                         ifelse(full$isco08_esas_k==82,82,
                         ifelse(full$isco08_esas_k==83,83,
                         ifelse(full$isco08_esas_k==91,91,
                         ifelse(full$isco08_esas_k==92,92,
                         ifelse(full$isco08_esas_k==93,93,
                         ifelse(full$isco08_esas_k==94,91,
                         ifelse(full$isco08_esas_k==95,91,
                         ifelse(full$isco08_esas_k==96,91,NA
                         ))))))))))))))))))))))))))))))))))))))))))))))))))))

isco.88.2014 <- full$isco.88.2[full$yil==2014]
isco.88.2015 <- full$isco.88.2[full$yil==2015]
isco.88.2016 <- full$isco.88.2[full$yil==2016]


full[,21] <- data.frame(c(isco.88.2016,
                          isco.88.2015,
                          isco.88.2014,
                          hlfs2013$s38kod,
                          hlfs2012$s38kod88,
                          hlfs2011$s38kod,
                          hlfs2010$s38kod,
                          hlfs2009$s38kod,
                          hlfs2008$s38kod,
                          hlfs2007$s38kod,
                          hlfs2006$s38kod,
```

```
                          hlfs2005$s38kod,
                          hlfs2004$s33kod,
                          hlfs2003$s22kod,
                          hlfs2002$s22kod,
                          hlfs2001$s22kod,
                          rep(NA,h0)))
colnames(full)[21] <- "isco.88.merged"
full[,21] <- integ(full[,21])

#BAĞIMLI DEĞİŞKENİM####
#Bir önceki dönem işsiz olup, bir sonraki dönemde iş bulanlar buradan kodlanacak

##2014 yılını indirgiyorum,2013 e göre
#hlfs2014$sil <- hlfs2014$faal_durum_gecenyil
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==11] <- 2
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==12] <- 1
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==2] <- 4
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==3] <- 6
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==4] <- 3
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==5] <- 7
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==6] <- 9
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==7] <- 5
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==8] <- 8
#hlfs2014$sil[hlfs2014$faal_durum_gecenyil==98] <- 10

##2015 yılını indirgiyorum,2013 e göre
#hlfs2015$sil <- hlfs2015$faal_durum_gecenyil
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==11] <- 2
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==12] <- 1
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==2] <- 4
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==3] <- 6
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==4] <- 3
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==5] <- 7
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==6] <- 9
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==7] <- 5
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==8] <- 8
#hlfs2015$sil[hlfs2015$faal_durum_gecenyil==98] <- 10


##2016 yılını indirgiyorum,2013 e göre
#hlfs2016$sil <- hlfs2016$faal_durum_gecenyil
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==11] <- 2
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==12] <- 1
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==2] <- 4
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==3] <- 6
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==4] <- 3
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==5] <- 7
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==6] <- 9
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==7] <- 5
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==8] <- 8
#hlfs2016$sil[hlfs2016$faal_durum_gecenyil==98] <- 10

full[,22] <- data.frame(c(as.numeric(hlfs2016$faal_durum_gecenyil),
                          as.numeric(hlfs2015$faal_durum_gecenyil),
                          as.numeric(hlfs2014$faal_durum_gecenyil),
                          as.numeric(hlfs2013$s97),
                          as.numeric(hlfs2012$s97),
                          as.numeric(hlfs2011$s97),
                          as.numeric(hlfs2010$s97),
                          as.numeric(hlfs2009$s97),
                          as.numeric(hlfs2008$s108),
                          as.numeric(hlfs2007$s108),
                          as.numeric(hlfs2006$s108),
                          as.numeric(hlfs2005$s108),
                          as.numeric(hlfs2004$s95),
                          as.numeric(hlfs2003$s43),
                          as.numeric(hlfs2002$s43),
                          as.numeric(hlfs2001$s43),
                          as.numeric(hlfs2000$s43)))


colnames(full)[22] <- "faal_durum_gecenyil"
full[,22] <- integ(full[,22])
```

```
#BASE=2013-2009 arası
#2004 yılında işten uzaklaştırıldı diye bir seçenek var, bunu siliyorum
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==2] <- NA

#2014-2016 arasındaki düzeltmeler
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==12] <- 1
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==11] <- 22
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==2] <- 44
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==3] <- 66
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==4] <- 33
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==5] <- 77
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==6] <- 99
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==7] <- 55
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==8] <- 88
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==98] <- 10

full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==22] <- 2
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==33] <- 3
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==44] <- 4
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==55] <- 5
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==66] <- 6
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==77] <- 7
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==88] <- 8
full$faal_durum_gecenyil[full$yil>=2014 & full$faal_durum_gecenyil==99] <- 9

#2008-2005
full$faal_durum_gecenyil[full$yil<=2008 & full$yil>=2005
                         & full$faal_durum_gecenyil==9] <- 10

#2004
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==2] <- NA
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==3] <- 2
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==4] <- 3
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==5] <- 4
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==6] <- 5
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==7] <- 6
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==8] <- 7
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==9] <- 8
full$faal_durum_gecenyil[full$yil==2004 & full$faal_durum_gecenyil==10] <- 10


#2003 ve 2002'yi 2001'e uyarlıyorum#
full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==2] <- 1

full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==3] <- 1

full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==4] <- 2

full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==5] <- 3

full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==6] <- 4

full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==7] <- 5

full$faal_durum_gecenyil[(full$yil==2003 |full$yil==2002)
                         & full$faal_durum_gecenyil==8] <- 6

#2003 ve öncesinde ise tamamen farklı

#Not: 2003 ve öncesi için s43'ü bir önceki dönem 1 yıldır
#işsizdi diye işsizlik <- s43<5 = 0, otherwise=1 diye kodlayabilirim.
#full$faal_durum_gecenyil[(full$yil %in% c(2005,2006,2007,2008))
#& (full$isbulma %in% 9)] <- 10

colnames(full)[22] <- "faal_durum_gecenyil"
full[,22] <- integ(full[,22])
```

```
#___EK Çalışma Saati####
full[,23] <- data.frame(c(hlfs2016$esas_fiili,
                          hlfs2015$esas_fiili,
                          hlfs2014$esas_fiili,
                          hlfs2013$s56a_top,
                          hlfs2012$s56a_top,
                          hlfs2011$s56a_top,
                          hlfs2010$s56a_top,
                          hlfs2009$s56a_top,
                          hlfs2008$s63a_top,
                          hlfs2007$s63a_top,
                          hlfs2006$s63a_top,
                          hlfs2005$s63a_top,
                          hlfs2004$s56a_top,
                          hlfs2003$s28a,
                          hlfs2002$s28a,
                          hlfs2001$s28a,
                          hlfs2000$s28a))
colnames(full)[23] <- "esas_fiili"
full[,23] <- integ(full[,23])



#___Gelir####
full[,24] <- data.frame(c(hlfs2016$gelir_gecenay_k,
                          hlfs2015$gelir_gecenay_k,
                          hlfs2014$gelir_gecenay_k,
                          hlfs2013$s71,
                          hlfs2012$s69,
                          hlfs2011$s69,
                          hlfs2010$s69,
                          hlfs2009$s69,
                          hlfs2008$s78,
                          hlfs2007$s78,
                          hlfs2006$s78,
                          hlfs2005$s78,
                          hlfs2004$s69,
                          hlfs2003$s32c,
                          hlfs2002$s32c,
                          rep(NA,sum(h1,h0))))
colnames(full)[24] <- "gelir_gecenay_k"
full[,24] <- numer(full[,24])


#___Nuts Kodlamaları####
full[,25] <- data.frame(c(hlfs2016$ibbs_1,
                          hlfs2015$ibbs_1,
                          hlfs2014$ibbs_1,
                          hlfs2013$nuts1,
                          hlfs2012$nuts1,
                          hlfs2011$nuts1,
                          hlfs2010$nuts1,
                          hlfs2009$nuts1,
                          hlfs2008$nuts1,
                          hlfs2007$nuts1,
                          hlfs2006$nuts1,
                          hlfs2005$nuts1,
                          hlfs2004$nuts1,
                          rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[25] <- "nuts1"
full[,25] <- integ(full[,25])

full[,26] <- data.frame(c(hlfs2016$ibbs_2,
                          hlfs2015$ibbs_2,
                          hlfs2014$ibbs_2,
                          hlfs2013$nuts2,
                          hlfs2012$nuts2,
                          hlfs2011$nuts2,
                          hlfs2010$nuts2,
                          hlfs2009$nuts2,
                          hlfs2008$nuts2,
                          hlfs2007$nuts2,
                          hlfs2006$nuts2,
                          hlfs2005$nuts2,
```

```
                          hlfs2004$nuts2,
                          rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[26] <- "nuts2"
full[,26] <- integ(full[,26])
#___Durum Kodlaması####

full[,27] <- data.frame(c(hlfs2016$durum,
                          hlfs2015$durum,
                          hlfs2014$durum,
                          hlfs2013$durum,
                          hlfs2012$durum,
                          hlfs2011$durum,
                          hlfs2010$durum,
                          hlfs2009$durum,
                          hlfs2008$durum,
                          hlfs2007$durum,
                          hlfs2006$durum,
                          hlfs2005$durum,
                          hlfs2004$durum,
                          hlfs2003$durum,
                          hlfs2002$durum,
                          hlfs2001$durum,
                          hlfs2000$durum))

colnames(full)[27] <- "t1"
full[,27] <- integ(full[,27])




#___İşteki Durumu####
full[,28] <- data.frame(c(hlfs2016$isteki_durum_k,
                          hlfs2015$isteki_durum_k,
                          hlfs2014$isteki_durum_k,
                          hlfs2013$s39,
                          hlfs2012$s39,
                          hlfs2011$s39,
                          hlfs2010$s39,
                          hlfs2009$s39,
                          hlfs2008$s39,
                          hlfs2007$s39,
                          hlfs2006$s39,
                          hlfs2005$s39,
                          hlfs2004$s34,
                          hlfs2003$s23,
                          hlfs2002$s23,
                          hlfs2001$s23,
                          hlfs2000$s23))
colnames(full)[28] <- "isteki_durum_k"

full$isteki_durum_k[full$yil<2009 & full$isteki_durum_k==1] <- 1
full$isteki_durum_k[full$yil<2009 & full$isteki_durum_k==2] <- 1
full$isteki_durum_k[full$yil<2009 & full$isteki_durum_k==3] <- 2
full$isteki_durum_k[full$yil<2009 & full$isteki_durum_k==4] <- 3
full$isteki_durum_k[full$yil<2009 & full$isteki_durum_k==5] <- 4

colnames(full)[28] <- "isteki_durum_k"
full[,28] <- integ(full[,28])


#____Eğitim Yılı####
full[,29] <- NA
full[,29][full$okul<2] <- 0
full[,29][full$okul==2] <- 5
full[,29][full$okul==3] <- 8
full[,29][full$okul==4] <- 11
full[,29][full$okul==5] <- 11
full[,29][full$okul==6] <- 15
colnames(full)[29] <- "educ_year"
full[,29] <- integ(full[,29])

#____Experience####
full[,30] <- NA
```

```
full[,30] <- ifelse(full$educ_year<9
                    & full$cinsiyet==1,(full$yas-15)-1,NA)
full[,30] <- ifelse(full$educ_year<9
                    & full$cinsiyet==2,(full$yas-15),NA) #Kadınların kodlaması

full[,30] <- ifelse(full$educ_year>8
                    & full$educ_year<12 & full$cinsiyet==1,(full$yas-18)-1,NA)
full[,30] <- ifelse(full$educ_year>8
                    & full$educ_year<12 & full$cinsiyet==2,(full$yas-18),NA)
                    #Kadınların kodlaması

full[,30] <- ifelse(full$educ_year>14
                    & full$cinsiyet==1,(full$yas-22)-1,NA)
full[,30] <- ifelse(full$educ_year>14
                    & full$cinsiyet==2,(full$yas-22),NA)
                    #Kadınların kodlaması



full[,30] <- NA
full[,30] <- ifelse(full$educ_year<9 & full$cinsiyet==1,(full$yas-15)-1,
                ifelse(full$educ_year<9 & full$cinsiyet==2,(full$yas-15),
                ifelse(full$educ_year>8 & full$educ_year<12 & full$cinsiyet==1,(full$yas-18)-1
                ifelse(full$educ_year>8 & full$educ_year<12 & full$cinsiyet==2,(full$yas-18),
                ifelse(full$educ_year>14 & full$cinsiyet==1,(full$yas-22)-1,
                ifelse(full$educ_year>14 & full$cinsiyet==2,(full$yas-22),NA))))))
                #Kadınların kodlaması

colnames(full)[30] <- "experience"
full[,30] <- integ(full[,30])
full$experience[full$experience<0] <- 0


#___Onceki_Ikamet_Bilgileri####
full[,31] <- data.frame(c(hlfs2016$onceki_ikamet,
                          hlfs2015$onceki_ikamet,
                          hlfs2014$onceki_ikamet,
                          hlfs2013$s10a,
                          hlfs2012$s10a,
                          hlfs2011$s10a,
                          hlfs2010$s10a,
                          hlfs2009$s10a,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[31] <- "onceki_ikamet"
full[,31] <- integ(full[,31])


#___Onceki_Yerleşim_Bilgileri####
full[,32] <- data.frame(c(hlfs2016$yerlesim_tur,
                          hlfs2015$yerlesim_tur,
                          hlfs2014$yerlesim_tur,
                          hlfs2013$s10b,
                          hlfs2012$s10b,
                          hlfs2011$s10b,
                          hlfs2010$s10b,
                          hlfs2009$s10b,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[32] <- "yerlesim_tur"
full[,32] <- integ(full[,32])

#___Eş_Bilgileri####
full[,33] <- data.frame(c(hlfs2016$fertno_es,
                          hlfs2015$fertno_es,
                          hlfs2014$fertno_es,
                          hlfs2013$s12a,
                          hlfs2012$s12a,
                          hlfs2011$s12a,
                          hlfs2010$s12a,
                          hlfs2009$s12a,
                          hlfs2008$s12a,
                          hlfs2007$s12a,
```

```
                          hlfs2006$s12a,
                          hlfs2005$s12a,
                          rep(NA,sum(h4,h3,h2,h1,h0))))
colnames(full)[33] <- "fertno_es"
full[,33] <- integ(full[,33])


#___Anne_Bilgileri####
full[,34] <- data.frame(c(hlfs2016$fertno_anne,
                          hlfs2015$fertno_anne,
                          hlfs2014$fertno_anne,
                          hlfs2013$s12b,
                          hlfs2012$s12b,
                          hlfs2011$s12b,
                          hlfs2010$s12b,
                          hlfs2009$s12b,
                          hlfs2008$s12b,
                          hlfs2007$s12b,
                          hlfs2006$s12b,
                          hlfs2005$s12b,
                          rep(NA,sum(h4,h3,h2,h1,h0))))
colnames(full)[34] <- "fertno_anne"
full[,34] <- integ(full[,34])


#___Baba_Bilgileri####
full[,35] <- data.frame(c(hlfs2016$fertno_baba,
                          hlfs2015$fertno_baba,
                          hlfs2014$fertno_baba,
                          hlfs2013$s12c,
                          hlfs2012$s12c,
                          hlfs2011$s12c,
                          hlfs2010$s12c,
                          hlfs2009$s12c,
                          hlfs2008$s12c,
                          hlfs2007$s12c,
                          hlfs2006$s12c,
                          hlfs2005$s12c,
                          rep(NA,sum(h4,h3,h2,h1,h0))))
colnames(full)[35] <- "fertno_baba"
full[,35] <- integ(full[,35])


#___Okuryazarlık_Durumu####
full[,36] <- data.frame(c(hlfs2016$okur_yazar,
                          hlfs2015$okur_yazar,
                          hlfs2014$okur_yazar,
                          hlfs2013$s14,
                          hlfs2012$s14,
                          hlfs2011$s14,
                          hlfs2010$s14,
                          hlfs2009$s14,
                          hlfs2008$s13,
                          hlfs2007$s13,
                          hlfs2006$s13,
                          hlfs2005$s13,
                          hlfs2004$s9,
                          hlfs2003$s8,
                          hlfs2002$s8,
                          hlfs2001$s8,
                          hlfs2000$s8))
colnames(full)[36] <- "okur_yazar"
full[,36] <- integ(full[,36])


#___Okulu_yeni_bitirmiş_olma####
full[,37] <- data.frame(c(hlfs2016$egitim_devam_k,
                          hlfs2015$egitim_devam_k,
                          hlfs2014$egitim_devam_k,
                              hlfs2013$s17,
                              hlfs2012$s17,
                              hlfs2011$s17,
                              hlfs2010$s17,
                              hlfs2009$s17,
```

```
                              hlfs2008$s17,
                              hlfs2007$s17,
                              hlfs2006$s17,
                              hlfs2005$s17,
                              hlfs2004$s13,
                              hlfs2003$s10,
                              hlfs2002$s10,
                              hlfs2001$s10,
                              hlfs2000$s10))
colnames(full)[37] <- "egitim_devam_k"
full[,37] <- integ(full[,37])



#___Okula_devam_etme_durumu####
full[,38] <- data.frame(c(hlfs2016$okul_devam_k,
                          hlfs2015$okul_devam_k,
                          hlfs2014$okul_devam_k,
                          hlfs2013$s18,
                          hlfs2012$s18,
                          hlfs2011$s18,
                          hlfs2010$s18,
                          hlfs2009$s18,
                          hlfs2008$s18a,
                          hlfs2007$s18a,
                          hlfs2006$s18a,
                          hlfs2005$s18a,
                          hlfs2004$s14a,
                          hlfs2003$s9,
                          hlfs2002$s9,
                          hlfs2001$s9,
                          hlfs2000$s9))
colnames(full)[38] <- "okul_devam_k"
full[,38] <- integ(full[,38])


#___Okulda_sınıfa_devam_etme_durumu####
full[,39] <- data.frame(c(hlfs2016$sinif_devam,
                          hlfs2015$sinif_devam,
                          hlfs2014$sinif_devam,
                          hlfs2013$s19,
                          hlfs2012$s19,
                          hlfs2011$s19,
                          hlfs2010$s19,
                          hlfs2009$s19,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
colnames(full)[39] <- "sinif_devam"
full[,39] <- integ(full[,39])




#___Kursa_devam_edip_etmediği####
full[,40] <- data.frame(c(hlfs2016$kurs,
                          hlfs2015$kurs,
                          hlfs2014$kurs,
                          hlfs2013$s21,
                          hlfs2012$s21,
                          hlfs2011$s21,
                          hlfs2010$s21,
                          hlfs2009$s21,
                          hlfs2008$s21,
                          hlfs2007$s21,
                          hlfs2006$s21,
                          hlfs2005$s21,
                          hlfs2004$s16,
                          rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[40] <- "kurs"
full[,40] <- integ(full[,40])


#___Kursun_amaci####
full[,41] <- data.frame(c(rep(NA,sum(h16)),
```

```
                          rep(NA,sum(h15)),
                          rep(NA,sum(h14)),
                          hlfs2013$s22,
                          hlfs2012$s22,
                          hlfs2011$s22,
                          hlfs2010$s22,
                          hlfs2009$s22,
                          hlfs2008$s23,
                          hlfs2007$s23,
                          hlfs2006$s23,
                          hlfs2005$s23,
                          hlfs2004$s18,
                          rep(NA,sum(h3,h2,h1,h0))))
colnames(full)[41] <- "kurs.amaci"
full[,41] <- integ(full[,41])


#___Medeni_durum_bilgisi####
full[,42] <- data.frame(c(hlfs2016$medeni_durum,
                          hlfs2015$medeni_durum,
                          hlfs2014$medeni_durum,
                          hlfs2013$s24,
                          hlfs2012$s24,
                          hlfs2011$s24,
                          hlfs2010$s24,
                          hlfs2009$s19,
                          hlfs2008$s19,
                          hlfs2007$s19,
                          hlfs2006$s19,
                          hlfs2005$s19,
                          hlfs2004$s15,
                          hlfs2003$s11,
                          hlfs2002$s11,
                          hlfs2001$s11,
                          hlfs2000$s11))
colnames(full)[42] <- "medeni_durum"
full[,42] <- integ(full[,42])



#___Çalışma_Durumu####
full[,43] <- data.frame(c(hlfs2016$calisma,
                          hlfs2015$calisma,
                          hlfs2014$calisma,
                          hlfs2013$s26,
                          hlfs2012$s26,
                          hlfs2011$s26,
                          hlfs2010$s26,
                          hlfs2009$s26,
                          hlfs2008$s26,
                          hlfs2007$s26,
                          hlfs2006$s26,
                          hlfs2005$s26,
                          hlfs2004$s21,
                          hlfs2003$s12,
                          hlfs2002$s12,
                          hlfs2001$s12,
                          hlfs2000$s12))
colnames(full)[43] <- "calisma"
full[,43] <- integ(full[,43])



#___Çalışma_Bir_saat####
full[,44] <- data.frame(c(hlfs2016$calisma_bir_saat,
                          hlfs2015$calisma_bir_saat,
                          hlfs2014$calisma_bir_saat,
                          hlfs2013$s27,
                          hlfs2012$s27,
                          hlfs2011$s27,
                          hlfs2010$s27,
                          hlfs2009$s27,
                          hlfs2008$s27,
                          hlfs2007$s27,
```

```
                          hlfs2006$s27,
                          hlfs2005$s27,
                          hlfs2004$s22,
                          hlfs2003$s13,
                          hlfs2002$s13,
                          hlfs2001$s13,
                          hlfs2000$s13))
colnames(full)[44] <- "calisma_bir_saat"
full[,44] <- integ(full[,44])



#___Referans_Geçici_İş_Bir_saat####
full[,45] <- data.frame(c(hlfs2016$calisma_sahip_is,
                          hlfs2015$calisma_sahip_is,
                          hlfs2014$calisma_sahip_is,
                          hlfs2013$s28,
                          hlfs2012$s28,
                          hlfs2011$s28,
                          hlfs2010$s28,
                          hlfs2009$s28,
                          hlfs2008$s28,
                          hlfs2007$s28,
                          hlfs2006$s28,
                          hlfs2005$s28,
                          hlfs2004$s23,
                          hlfs2003$s14,
                          hlfs2002$s14,
                          hlfs2001$s14,
                          hlfs2000$s14))
colnames(full)[45] <- "calisma_sahip_is"
full[,45] <- integ(full[,45])



#___Çalışmama_Nedeni####
full[,46] <- data.frame(c(hlfs2016$calismama_neden_ref,
                          hlfs2015$calismama_neden_ref,
                          hlfs2014$calismama_neden_ref,
                          hlfs2013$s29,
                          hlfs2012$s29,
                          hlfs2011$s29,
                          hlfs2010$s29,
                          hlfs2009$s29,
                          hlfs2008$s29,
                          hlfs2007$s29,
                          hlfs2006$s29,
                          hlfs2005$s29,
                          hlfs2004$s24,
                          hlfs2003$s15,
                          hlfs2002$s15,
                          hlfs2001$s15,
                          hlfs2000$s15))

colnames(full)[46] <- "calismama_neden_ref"
full[,46] <- integ(full[,46])

#2014'ün NA'i 98 işaretlenmiş
full[,46][full$calismama_neden_ref==98 & full$yil==2014 ] <- 10

#Aşağıdaki ara yıllarda şıklar epey farklı
full[,46][full$calismama_neden_ref==9 & (2008<full$yil & full$yil<2014) ] <- 10
full[,46][full$calismama_neden_ref==8 & (2008<full$yil & full$yil<2014) ] <- 9
full[,46][full$calismama_neden_ref==7 & (2008<full$yil & full$yil<2014) ] <- 8
full[,46][full$calismama_neden_ref==6 & (2008<full$yil & full$yil<2014) ] <- 7
full[,46][full$calismama_neden_ref==5 & (2008<full$yil & full$yil<2014) ] <- 6

#2003 ve öncesi ise epey epey farklı
full[,46][full$calismama_neden_ref==2 & (full$yil<2004) ] <- 33
full[,46][full$calismama_neden_ref==3 & (full$yil<2004) ] <- 22
full[,46][full$calismama_neden_ref==5 & (full$yil<2004) ] <- 88
full[,46][full$calismama_neden_ref==6 & (full$yil<2004) ] <- 55
full[,46][full$calismama_neden_ref==7 & (full$yil<2004) ] <- 99
```

```
full[,46][full$calismama_neden_ref==8 & (full$yil<2004) ] <- 66

full[,46][full$calismama_neden_ref==9 & (full$yil<2004) ] <- 10

full[,46][full$calismama_neden_ref==33 & (full$yil<2004) ] <- 3
full[,46][full$calismama_neden_ref==22 & (full$yil<2004) ] <- 2
full[,46][full$calismama_neden_ref==88 & (full$yil<2004) ] <- 8
full[,46][full$calismama_neden_ref==55 & (full$yil<2004) ] <- 5
full[,46][full$calismama_neden_ref==99 & (full$yil<2004) ] <- 9
full[,46][full$calismama_neden_ref==66 & (full$yil<2004) ] <- 6


colnames(full)[46] <- "calismama_neden_ref"
full[,46] <- integ(full[,46])


#BAĞIMLI_DEĞİŞKENİM_2_Recoding####
# t0:İşsiz (2) olmalı,
# t1:İşi var(1), işsiz(2) durum değişkenindeki kodlama şeklinde)
#Not: 2003 ve öncesi için s43'ü bir önceki dönem 1 yıldır işsizdi diye
#işsizlik <- s43<5 = 0, otherwise=1 diye kodlayabilirim.

full[,47] <- rep(NA,dim(full)[1])
full[,47] <- ifelse(full$yil<=2003 &  full$faal_durum_gecenyil==6,0,1)

full[full$yil<=2003 & is.na(full$faal_durum_gecenyil)==T, 47] <-1
full[,47] <- ifelse(full$yil>=2004 & full$faal_durum_gecenyil==4,0,full[,47]  )



#2003 ve öncesi yıllıklarda s43, 5.şık ve sonrası 1 yıl önce işsizdi demek.
colnames(full)[47]<- "t0"
full[,47] <- integ(full[,47])

# istihdam  işsiz işgücünde değil 4
#t1 1370814   75084  103385       0
#t0  302566  150206 2540998       4
#302566 kişi t0'da işsizken, t1'de işgücüne geçmiş demekki



#04.Sample Oluşturma####

#Deflator,base(2009)####
deflator <- openxlsx::read.xlsx(xlsxFile=
      "/media/DRIVE/Dropbox/_VeriTabanlarim/Deflator_WDI.xlsx",
        sheet = 2, colNames=T)
full$deflator[full$yil==2000] <- deflator$deflator[1]
full$deflator[full$yil==2001] <- deflator$deflator[2]
full$deflator[full$yil==2002] <- deflator$deflator[3]
full$deflator[full$yil==2003] <- deflator$deflator[4]
full$deflator[full$yil==2004] <- deflator$deflator[5]
full$deflator[full$yil==2005] <- deflator$deflator[6]
full$deflator[full$yil==2006] <- deflator$deflator[7]
full$deflator[full$yil==2007] <- deflator$deflator[8]
full$deflator[full$yil==2008] <- deflator$deflator[9]
full$deflator[full$yil==2009] <- deflator$deflator[10]
full$deflator[full$yil==2010] <- deflator$deflator[11]
full$deflator[full$yil==2011] <- deflator$deflator[12]
full$deflator[full$yil==2012] <- deflator$deflator[13]
full$deflator[full$yil==2013] <- deflator$deflator[14]
full$deflator[full$yil==2014] <- deflator$deflator[15]
full$deflator[full$yil==2015] <- deflator$deflator[16]
full$deflator[full$yil==2016] <- deflator$deflator[17]



#___Dosyanın içeriğini görüntüleme#####
#View(head(full))


#___Unique identifier oluşturma####
class(full$t0)
full$id <- ((full$yil)*1000000000)+(full$birimno)*1000+
```

```
  as.numeric(as.character(full$fert))



#___Faktor_veya_ağırlık_katsayıları####
full[,50] <- data.frame(c(hlfs2016$agirlik_katsayisi*1000,
                          hlfs2015$agirlik_katsayisi*1000,
                          hlfs2014$agirlik_katsayisi*1000,
                          hlfs2013$faktor,
                          hlfs2012$faktor,
                          hlfs2011$faktor,
                          hlfs2010$faktor,
                          hlfs2009$faktor,
                          hlfs2008$faktor,
                          hlfs2007$faktor,
                          hlfs2006$faktor,
                          hlfs2005$faktor,
                          hlfs2004$faktor,
                          hlfs2003$fakyil,
                          hlfs2002$fakyil,
                          hlfs2001$fakyil,
                          hlfs2000$fakyil))
colnames(full)[50] <- "agirlik_katsayisi"


#___Hanehalkı_büyüklüğü####
#Bu veri sadece 2014 ve sonrasında var, sadece [count(fertno), by(formno)]
#ile hesaplanmamış, 15 yaşının altındakiler sayılmamış,
#2013 ve öncesi için [count(fertno), by(formno)] ile eklenecek
full2 <- data.frame()
for(j in 2000:2016){
  k <- data.frame(table(full$yil[full$yil==j],full$birimno[full$yil==j]))
  names(k) <- c('yil','birimno','hh_buyukluk_estimated')
  z <- merge(full[full$yil==j,],k,by = c('yil','birimno'))
  full2 <- rbind(z,full2)
}
full <- full2
rm(k,z,full2)



full[,52] <- data.frame(c(hlfs2016$hh_buyukluk,
                          hlfs2015$hh_buyukluk,
                          hlfs2014$hh_buyukluk,
                          full$hh_buyukluk_estimated[full$yil==2013],
                          full$hh_buyukluk_estimated[full$yil==2012],
                          full$hh_buyukluk_estimated[full$yil==2011],
                          full$hh_buyukluk_estimated[full$yil==2010],
                          full$hh_buyukluk_estimated[full$yil==2009],
                          full$hh_buyukluk_estimated[full$yil==2008],
                          full$hh_buyukluk_estimated[full$yil==2007],
                          full$hh_buyukluk_estimated[full$yil==2006],
                          full$hh_buyukluk_estimated[full$yil==2005],
                          full$hh_buyukluk_estimated[full$yil==2004],
                          full$hh_buyukluk_estimated[full$yil==2003],
                          full$hh_buyukluk_estimated[full$yil==2002],
                          full$hh_buyukluk_estimated[full$yil==2001],
                          full$hh_buyukluk_estimated[full$yil==2000]))


colnames(full)[52] <- "hh_buyukluk"
full[,52] <- integ(full[,52])



#___İş yeri durumu####
full[,53] <- data.frame(c(hlfs2016$isyeri_durum,
                          hlfs2015$isyeri_durum,
                          hlfs2014$isyeri_durum,
                          hlfs2013$s35,
                          hlfs2012$s35,
                          hlfs2011$s35,
                          hlfs2010$s35,
```

```
                          hlfs2009$s35,
                          hlfs2008$s35,
                          hlfs2007$s35,
                          hlfs2006$s35,
                          hlfs2005$s35,
                          hlfs2004$s30,
                          hlfs2003$s18,
                          hlfs2002$s18,
                          hlfs2001$s18,
                          hlfs2000$s18))


colnames(full)[53] <- "isyeri_durum"
#şıkları farklı olanlar değişecek
full[,53][full$yil<=2004 & full$isyeri_durum==5] <- 4
full[,53][full$yil<=2004 & full$isyeri_durum==6] <- 5

full[,53] <- integ(full[,53])



#___İş yeri kişi sayıları####
full[,54] <- data.frame(c(hlfs2016$calisan_sayi_hh,
                          hlfs2015$calisan_sayi_hh,
                          hlfs2014$calisan_sayi_hh,
                          hlfs2013$s37a,
                          hlfs2012$s37a,
                          hlfs2011$s37a,
                          hlfs2010$s37a,
                          hlfs2009$s37a,
                          hlfs2008$s37a,
                          hlfs2007$s37a,
                          hlfs2006$s37a,
                          hlfs2005$s37a,
                          hlfs2004$s32a,
                          hlfs2003$s21,
                          hlfs2002$s21,
                          hlfs2001$s21,
                          hlfs2000$s21))


colnames(full)[54] <- "calisan_sayi_hh"
#şıkları farklı olanlar değişecek
full[,54][full$yil>=2004 & full$calisan_sayi_hh==5] <- NA
full[,54][full$yil>=2004 & full$calisan_sayi_hh==6] <- NA

full[,54] <- integ(full[,54])



#___İdari sorumluluk####
full[,55] <- data.frame(c(hlfs2016$idari_sorumlu,
                          hlfs2015$idari_sorumlu,
                          hlfs2014$idari_sorumlu,
                          hlfs2013$s41,
                          hlfs2012$s41,
                          hlfs2011$s41,
                          hlfs2010$s41,
                          hlfs2009$s41,
                          rep(NA, sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

full[,55] <- integ(full[,55])
colnames(full)[55] <- "idari_sorumlu"


#___SGK Kaydınız var mı?####
full[,56] <- data.frame(c(hlfs2016$kayitlilik,
                          hlfs2015$kayitlilik,
                          hlfs2014$kayitlilik,
                          hlfs2013$s42,
                          hlfs2012$s42,
                          hlfs2011$s42,
                          hlfs2010$s42,
                          hlfs2009$s42,
```

```
                          hlfs2008$s43,
                          hlfs2007$s43,
                          hlfs2006$s43,
                          hlfs2005$s43,
                          hlfs2004$s37,
                          hlfs2003$s25,
                          hlfs2002$s25,
                          hlfs2001$s25,
                          hlfs2000$s25))

full[,56] <- integ(full[,56])
colnames(full)[56] <- "kayitlilik"



#___İş bulma yöntemi####
#Diğer şıkları farklı
full[,57] <- data.frame(c(hlfs2016$isbul_yontem,
                          hlfs2015$isbul_yontem,
                          hlfs2014$isbul_yontem,
                          hlfs2013$s44,
                          hlfs2012$s44,
                          hlfs2011$s44,
                          hlfs2010$s44,
                          hlfs2009$s44,
                          hlfs2008$s45,
                          hlfs2007$s45,
                          hlfs2006$s45,
                          hlfs2005$s45,
                          hlfs2004$s39,
                          rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[57] <- "isbul_yontem"
full[,57][full$yil>=2014 & full$isbulma.yontem==98] <- 5
full[,57] <- integ(full[,57])




#___Çalışma sekli####
full[,58] <- data.frame(c(hlfs2016$calisma_sekli,
                          hlfs2015$calisma_sekli,
                          hlfs2014$calisma_sekli,
                          hlfs2013$s45,
                          hlfs2012$s45,
                          hlfs2011$s45,
                          hlfs2010$s45,
                          hlfs2009$s45,
                          hlfs2008$s46,
                          hlfs2007$s46,
                          hlfs2006$s46,
                          hlfs2005$s46,
                          hlfs2004$s40,
                          rep(NA, sum(h3,h2,h1,h0))))


full[,58] <- integ(full[,58])
colnames(full)[58] <- "calisma_sekli"



#___Yarrızamanlı çalışma nedeni####
full[,59] <- data.frame(c(hlfs2016$yarizaman_neden,
                          hlfs2015$yarizaman_neden,
                          hlfs2014$yarizaman_neden,
                          hlfs2013$s46,
                          hlfs2012$s46,
                          hlfs2011$s46,
                          hlfs2010$s46,
                          hlfs2009$s46,
                          rep(NA, sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))
```

```
full[,59] <- integ(full[,59])
colnames(full)[59] <- "yarizaman_neden"



#___İş süreklilik durumu####
#2004 öncesindeki 3. şık ile 2. şıkkı birleştiriyorum
full[,60] <- data.frame(c(hlfs2016$is_sureklilik,
                          hlfs2015$is_sureklilik,
                          hlfs2014$is_sureklilik,
                          hlfs2013$s48,
                          hlfs2012$s48,
                          hlfs2011$s48,
                          hlfs2010$s48,
                          hlfs2009$s48,
                          hlfs2008$s49,
                          hlfs2007$s49,
                          hlfs2006$s49,
                          hlfs2005$s49,
                          hlfs2004$s42,
                          rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[60] <- "is_sureklilik"
full[,60][full$yil<=2003 & full$is_sureklilik==3] <- 2

full[,60] <- integ(full[,60])




#___Geçici iş nedeni####
#şık farklılığı var, düzenle
full[,61] <- data.frame(c(hlfs2016$gecici_neden,
                          hlfs2015$gecici_neden,
                          hlfs2014$gecici_neden,
                          hlfs2013$s49,
                          hlfs2012$s49,
                          hlfs2011$s49,
                          hlfs2010$s49,
                          hlfs2009$s49,
                          rep(NA, sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[61] <- "gecici_neden"
full[,61][full$yil>=2014 & full$gecici_neden==98] <- 5

full[,61] <- integ(full[,61])




#___Evden çalışma sıklığı####
full[,62] <- data.frame(c(hlfs2016$evde_cal_siklik,
                          hlfs2015$evde_cal_siklik,
                          hlfs2014$evde_cal_siklik,
                          hlfs2013$s51,
                          hlfs2012$s51,
                          hlfs2011$s51,
                          hlfs2010$s51,
                          hlfs2009$s51,
                          rep(NA, sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[62] <- "evde_cal_siklik"
full[,62] <- integ(full[,62])



#___Ek iş durumu####
full[,63] <- data.frame(c(hlfs2016$ekis_durum,
                          hlfs2015$ekis_durum,
```

```
                            hlfs2014$ekis_durum,
                            hlfs2013$s52,
                            hlfs2012$s52,
                            hlfs2011$s52,
                            hlfs2010$s52,
                            hlfs2009$s52,
                            hlfs2008$s58,
                            hlfs2007$s58,
                            hlfs2006$s58,
                            hlfs2005$s58,
                            hlfs2004$s51,
                            rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[63] <- "ekis_durum"
full[,63] <- integ(full[,63])



#___Ek iş NACE Rev.2 kodu - 4 tür var sadece####
full[,64] <- data.frame(c(hlfs2016$nace2_ekis_k,
                            hlfs2015$nace2_ekis_k,
                            hlfs2014$nace2_ekis_k,
                            hlfs2013$s53kod,
                            hlfs2012$s53kod,
                            hlfs2011$s53kod,
                            hlfs2010$s53kod,
                            hlfs2009$s53kod,
                            hlfs2008$s59kod,
                            hlfs2007$s59kod,
                            hlfs2006$s59kod,
                            hlfs2005$s59kod,
                            hlfs2004$s52kod,
                            rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[64] <- "nace2_ekis_k"
full[,64] <- integ(full[,64])



#___Ek iş içindeki iş durumu###
full[,65] <- data.frame(c(hlfs2016$isteki_durum_ekis,
                            hlfs2015$isteki_durum_ekis,
                            hlfs2014$isteki_durum_ekis,
                            hlfs2013$s54,
                            hlfs2012$s54,
                            hlfs2011$s54,
                            hlfs2010$s54,
                            hlfs2009$s54,
                            rep(NA, sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[65] <- "isteki_durum_ekis"
full[,65] <- integ(full[,65])



#___Ek işte toplam çalışılan saat###
full[,66] <- data.frame(c(hlfs2016$ekis_fiili,
                            hlfs2015$ekis_fiili,
                            hlfs2014$ekis_fiili,
                            hlfs2013$s56b_top,
                            hlfs2012$s56b_top,
                            hlfs2011$s56b_top,
                            hlfs2010$s56b_top,
                            hlfs2009$s56b_top,
                            hlfs2008$s63b_top,
                            hlfs2007$s63b_top,
                            hlfs2006$s63b_top,
                            hlfs2005$s63b_top,
                            hlfs2004$s56b_top,
                            hlfs2003$s28b,
                            hlfs2002$s28b,
                            hlfs2001$s28b,
                            hlfs2000$s28b))
```

```
colnames(full)[66] <- "ekis_fiili"
full[,66] <- integ(full[,66])



#___Fazla çalışma nedeni####
#2014 öncesindeki 98.diğer seçeneği 3 olarak değiştirilecek
full[,67] <- data.frame(c(hlfs2016$fazlacal_neden,
                          hlfs2015$fazlacal_neden,
                          hlfs2014$fazlacal_neden,
                          hlfs2013$s57,
                          hlfs2012$s57,
                          hlfs2011$s57,
                          hlfs2010$s57,
                          hlfs2009$s57,
                          hlfs2008$s65,
                          hlfs2007$s65,
                          hlfs2006$s65,
                          hlfs2005$s65,
                          hlfs2004$s57,
                          rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[67] <- "fazlacal_neden"
full[,67][full$yil>=2014 & full$fazlacal_neden==98] <- 3

full[,67] <- integ(full[,67])



#___Az çalışma nedeni####
#2004, 2005-2008 ve 2009-2016 arasında basit uyumsuzluklar var,
#kaydırmalar yapılacak.
full[,68] <- data.frame(c(hlfs2016$azcal_neden,
                          hlfs2015$azcal_neden,
                          hlfs2014$azcal_neden,
                          hlfs2013$s58,
                          hlfs2012$s58,
                          hlfs2011$s58,
                          hlfs2010$s58,
                          hlfs2009$s58,
                          hlfs2008$s66,
                          hlfs2007$s66,
                          hlfs2006$s66,
                          hlfs2005$s66,
                          hlfs2004$s58,
                          rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[68] <- "azcal_neden"

full[,68][full$yil>=2014 & full$azcal_neden==98] <- 16
full[,68][full$yil>=2014 & full$azcal_neden==14] <- 15
full[,68][full$yil>=2014 & full$azcal_neden==13] <- 14

full[,68][full$yil==2004 & full$azcal_neden==15] <- 16
full[,68][full$yil==2004 & full$azcal_neden==14] <- 15
full[,68][full$yil==2004 & full$azcal_neden==13] <- 14
full[,68][full$yil==2004 & full$azcal_neden==12] <- 13
full[,68][full$yil==2004 & full$azcal_neden==11] <- 12
full[,68][full$yil==2004 & full$azcal_neden==10] <- 11
full[,68][full$yil==2004 & full$azcal_neden==9] <- 10
full[,68][full$yil==2004 & full$azcal_neden==8] <- 9
full[,68][full$yil==2004 & full$azcal_neden==7] <- 8
full[,68][full$yil==2004 & full$azcal_neden==6] <- 7
full[,68][full$yil==2004 & full$azcal_neden==5] <- 6
full[,68][full$yil==2004 & full$azcal_neden==4] <- 5
full[,68][full$yil==2004 & full$azcal_neden==3] <- 4


full[,68] <- integ(full[,68])
```

```
#___Fazla çalışma isteği####
#2008 öncesinde s67'den indirgeme yapılacak,  .
full[,69] <- data.frame(c(hlfs2016$fazlacal_istek,
                          hlfs2015$fazlacal_istek,
                          hlfs2014$fazlacal_istek,
                          hlfs2013$s60a,
                          hlfs2012$s60a,
                          hlfs2011$s60a,
                          hlfs2010$s60a,
                          hlfs2009$s60a,
                          hlfs2008$s67,
                          hlfs2007$s67,
                          hlfs2006$s67,
                          hlfs2005$s67,
                          hlfs2004$s59,
                          hlfs2003$s30,
                          hlfs2002$s30,
                          hlfs2001$s30,
                          hlfs2000$s30))

colnames(full)[69] <- "fazlacal_istek"
full[,69] <- integ(full[,69])
full[,69][full$yil==2004 & full$fazlacal_istek<5] <- 1
full[,69][full$yil==2004 & full$fazlacal_istek==5] <- 2
full[,69][full$yil==2005 & full$fazlacal_istek<5] <- 1
full[,69][full$yil==2005 & full$fazlacal_istek==5] <- 2
full[,69][full$yil==2006 & full$fazlacal_istek<5] <- 1
full[,69][full$yil==2006 & full$fazlacal_istek==5] <- 2
full[,69][full$yil==2007 & full$fazlacal_istek<5] <- 1
full[,69][full$yil==2007 & full$fazlacal_istek==5] <- 2
full[,69][full$yil==2008 & full$fazlacal_istek<5] <- 1
full[,69][full$yil==2008 & full$fazlacal_istek==5] <- 2

full[,69] <- integ(full[,69])




#___Fazla çalışma şekli####
#2008 öncesinde s67'den değiştirilerek derlenecek  .
full[,70] <- data.frame(c(hlfs2016$fazlacal_sekil,
                          hlfs2015$fazlacal_sekil,
                          hlfs2014$fazlacal_sekil,
                          hlfs2013$s60b,
                          hlfs2012$s60b,
                          hlfs2011$s60b,
                          hlfs2010$s60b,
                          hlfs2009$s60b,
                          hlfs2008$s67,
                          hlfs2007$s67,
                          hlfs2006$s67,
                          hlfs2005$s67,
                          hlfs2004$s59,
                          rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[70] <- "fazlacal_sekil"
full[,70] <- integ(full[,70])
full[,70][full$yil<=2008 & full$fazlacal_istek==5] <- 4
full[,70][full$yil<=2008 & full$fazlacal_istek==4] <- 3
full[,70][full$yil<=2008 & full$fazlacal_istek==3] <- 2
full[,70][full$yil<=2008 & full$fazlacal_istek==2] <- 1

full[,70] <- integ(full[,70])




#___Fazla çalışmaya başlar mısınız?####
full[,71] <- data.frame(c(hlfs2016$fazlacal_baslama,
```

```
                          hlfs2015$fazlacal_baslama,
                          hlfs2014$fazlacal_baslama,
                          hlfs2013$s61,
                          hlfs2012$s61,
                          hlfs2011$s61,
                          hlfs2010$s61,
                          hlfs2009$s61,
                          rep(NA, sum(h8,h7,h6,h5)),
                          hlfs2004$s61,
                          rep(NA, sum(h3,h2,h1,h0))))

colnames(full)[71] <- "fazlacal_baslama"
full[,71] <- integ(full[,71])



#___Ek iş arama isteği####
full[,72] <- data.frame(c(hlfs2016$ekis_ara_durum,
                          hlfs2015$ekis_ara_durum,
                          hlfs2014$ekis_ara_durum,
                          hlfs2013$s62,
                          hlfs2012$s62,
                          hlfs2011$s62,
                          hlfs2010$s62,
                          hlfs2009$s62,
                          hlfs2008$s70,
                          hlfs2007$s70,
                          hlfs2006$s70,
                          hlfs2005$s70,
                          hlfs2004$s62,
                          hlfs2003$s33,
                          hlfs2002$s33,
                          hlfs2001$s33,
                          hlfs2000$s33))

colnames(full)[72] <- "ekis_ara_durum"
full[,72] <- integ(full[,72])



#___Ek iş aramama nedeni####
#şıklara dair düzeltmeler defterde, kodlanacak.
full[,73] <- data.frame(c(hlfs2016$ekis_ara_neden,
                          hlfs2015$ekis_ara_neden,
                          hlfs2014$ekis_ara_neden,
                          hlfs2013$s63,
                          hlfs2012$s63,
                          hlfs2011$s63,
                          hlfs2010$s63,
                          hlfs2009$s63,
                          hlfs2008$s71,
                          hlfs2007$s71,
                          hlfs2006$s71,
                          hlfs2005$s71,
                          hlfs2004$s63,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[73] <- "ekis_ara_neden"
full[,73] <- integ(full[,73])



full[,73][full$yil<=2007 & full$ekis_ara_neden==7] <- NA
full[,73][full$yil==2007 & full$ekis_ara_neden==8] <- 7
full[,73][full$yil==2007 & full$ekis_ara_neden==9] <- 8
full[,73][full$yil==2007 & full$ekis_ara_neden==10] <- 9

full[,73][full$yil==2008 & full$ekis_ara_neden==7] <- NA
full[,73][full$yil==2008 & full$ekis_ara_neden==8] <- 7
full[,73][full$yil==2008 & full$ekis_ara_neden==9] <- 8
full[,73][full$yil==2008 & full$ekis_ara_neden==10] <- 9

full[,73][full$yil>2008 & full$ekis_ara_neden==98] <- 9
full[,73][full$yil>2008 & full$ekis_ara_neden==2] <- NA
```

```
full[,73][full$yil>2008 & full$ekis_ara_neden==5] <- 2

full[,73][full$yil>2008 & full$ekis_ara_neden==4] <- 88
full[,73][full$yil>2008 & full$ekis_ara_neden==3] <- 77
full[,73][full$yil>2008 & full$ekis_ara_neden==6] <- 33
full[,73][full$yil>2008 & full$ekis_ara_neden==7] <- 55
full[,73][full$yil>2008 & full$ekis_ara_neden==8] <- 66

full[,73][full$yil>2008 & full$ekis_ara_neden==88] <- 8
full[,73][full$yil>2008 & full$ekis_ara_neden==77] <- 7
full[,73][full$yil>2008 & full$ekis_ara_neden==33] <- 3
full[,73][full$yil>2008 & full$ekis_ara_neden==66] <- 6
full[,73][full$yil>2008 & full$ekis_ara_neden==55] <- 5

full[,73] <- integ(full[,73])

#___Ek iş arama durumu####
#Bir sonraki soru ile birleştirilerek şıklar derlenecek
full[,74] <- data.frame(c(hlfs2016$ara_ekis_durum,
                          hlfs2015$ara_ekis_durum,
                          hlfs2014$ara_ekis_durum,
                          hlfs2013$s64a,
                          hlfs2012$s64a,
                          hlfs2011$s64a,
                          hlfs2010$s64a,
                          hlfs2009$s64a,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[74] <- "ara_ekis_durum"




#___Ek iş arama şekli####
#Bir önceki soru ile birleştirilerek şıklar derlenecek
full[,75] <- data.frame(c(hlfs2016$ara_ekis_sekil,
                          hlfs2015$ara_ekis_sekil,
                          hlfs2014$ara_ekis_sekil,
                          hlfs2013$s64b,
                          hlfs2012$s64b,
                          hlfs2011$s64b,
                          hlfs2010$s64b,
                          hlfs2009$s64b,
                          hlfs2008$s72,
                          hlfs2007$s72,
                          hlfs2006$s72,
                          hlfs2005$s72,
                          hlfs2004$s64,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[75] <- "ara_ekis_sekil"
full[,75] <- integ(full[,75])




full[,75][full$yil>2008  & full$ara_ekis_sekil==5] <- 6
full[,75][full$yil>2008  & full$ara_ekis_sekil==4] <- 5
full[,75][full$yil>2008  & full$ara_ekis_sekil==3] <- 4
full[,75][full$yil>2008  & full$ara_ekis_sekil==2] <- 3
full[,75][full$yil>2008  & full$ara_ekis_sekil==1] <- 2

full[,75][full$yil>2008  & full$ara_ekis_sekil==1] <- 1


#74. sorunun eksik kısımlarını da buradan derleyebilirim
full[,74][full$yil<2009  & full$ara_ekis_durum==1] <- 1
full[,74][full$yil<2009  & full$ara_ekis_durum>1] <- 2

full[,74] <- integ(full[,74])
full[,75] <- integ(full[,75])
```

```
#___Ek iş arama süresi####
full[,76] <- data.frame(c(hlfs2016$ekis_ara_sure,
                          hlfs2015$ekis_ara_sure,
                          hlfs2014$ekis_ara_sure,
                          hlfs2013$s65,
                          hlfs2012$s65,
                          hlfs2011$s65,
                          hlfs2010$s65,
                          hlfs2009$s65,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[76] <- "ekis_ara_sure"
full[,76] <- integ(full[,76])



#___Ek iş arama kanalları####
full[,77] <- data.frame(c(hlfs2016$ekis_ara_kanal_k,
                          hlfs2015$ekis_ara_kanal_k,
                          hlfs2014$ekis_ara_kanal_k,
                          hlfs2013$s66,
                          hlfs2012$s66,
                          hlfs2011$s66,
                          hlfs2010$s66,
                          hlfs2009$s66,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[77] <- "ekis_ara_kanal_k"
full[,77] <- integ(full[,77])




#___İşe başlama, ek iş####
full[,78] <- data.frame(c(hlfs2016$isbasi_ekis,
                          hlfs2015$isbasi_ekis,
                          hlfs2014$isbasi_ekis,
                          hlfs2013$s67,
                          hlfs2012$s67,
                          hlfs2011$s67,
                          hlfs2010$s67,
                          hlfs2009$s67,
                          rep(NA,sum(h8,h7,h6,h5,h4)),
                          hlfs2003$s41,
                          hlfs2002$s41,
                          hlfs2001$s41,
                          hlfs2000$s41))


colnames(full)[78] <- "isbasi_ekis"
full[,78] <- integ(full[,78])



#___İş arama durumu, referans haftası içinde iş aradı mı?###
full[,79] <- data.frame(c(hlfs2016$isara_durum,
                          hlfs2015$isara_durum,
                          hlfs2014$isara_durum,
                          hlfs2013$s75,
                          hlfs2012$s75,
                          hlfs2011$s75,
                          hlfs2010$s75,
                          hlfs2009$s75,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))


colnames(full)[79] <- "isara_durum"
full[,79] <- integ(full[,79])
```

```
#___İş arama durumu, referans haftası içinde iş aradı mı?###
full[,80] <- data.frame(c(hlfs2016$isara_basla_durum,
                          hlfs2015$isara_basla_durum,
                          hlfs2014$isara_basla_durum,
                          hlfs2013$s77,
                          hlfs2012$s77,
                          hlfs2011$s77,
                          hlfs2010$s77,
                          hlfs2009$s77,
                          hlfs2008$s89,
                          hlfs2007$s89,
                          hlfs2006$s89,
                          hlfs2005$s89,
                          hlfs2004$s80,
                          hlfs2003$s34,
                          hlfs2002$s34,
                          hlfs2001$s34,
                          hlfs2000$s34))

colnames(full)[80] <- "isara_basla_durum"
full[,80] <- integ(full[,80])

full[,80][full$yil>=2009 & full$isara_basla_durum==98] <- 11

full[,80][full$yil>=2006 & full$yil<=2008 & full$isara_basla_durum==9] <- 7
full[,80][full$yil>=2006 & full$yil<=2008 & full$isara_basla_durum==10] <- 9
full[,80][full$yil>=2006 & full$yil<=2008 & full$isara_basla_durum==11] <- 10
full[,80][full$yil>=2006 & full$yil<=2008 & full$isara_basla_durum==12] <- 11


full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==3] <- 44
full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==4] <- 33
full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==5] <- 33

full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==33] <- 3
full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==44] <- 4

full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==10] <- 11
full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==9] <- 10
full[,80][full$yil>=2002 & full$yil<=2003 & full$isara_basla_durum==8] <- 9

full[,80][full$yil<=2001 & full$isara_basla_durum==4] <- 33
full[,80][full$yil<=2001 & full$isara_basla_durum==5] <- 33
full[,80][full$yil<=2001 & full$isara_basla_durum==3] <- 44

full[,80][full$yil<=2001 & full$isara_basla_durum==33] <- 3
full[,80][full$yil<=2001 & full$isara_basla_durum==44] <- 4

full[,80][full$yil<=2001 & full$isara_basla_durum==9] <- 11
full[,80][full$yil<=2001 & full$isara_basla_durum==8] <- 9

full[,80] <- integ(full[,80])



#___İş arama kanal1?###
full[,81] <- data.frame(c(hlfs2016$isara_kanal1,
                          hlfs2015$isara_kanal1,
                          hlfs2014$isara_kanal1,
                          hlfs2013$s78a,
                          hlfs2012$s78a,
                          hlfs2011$s78a,
                          hlfs2010$s78a,
                          hlfs2009$s78a,
                          hlfs2008$s90a,
                          hlfs2007$s90a,
                          hlfs2006$s90a,
                          hlfs2005$s90a,
                          hlfs2004$s81a,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[81] <- "isara_kanal1"
full[,81] <- integ(full[,81])
```

```
#___İş arama kanal2?###
full[,82] <- data.frame(c(hlfs2016$isara_kanal2,
                          hlfs2015$isara_kanal2,
                          hlfs2014$isara_kanal2,
                          hlfs2013$s78b,
                          hlfs2012$s78b,
                          hlfs2011$s78b,
                          hlfs2010$s78b,
                          hlfs2009$s78b,
                          hlfs2008$s90b,
                          hlfs2007$s90b,
                          hlfs2006$s90b,
                          hlfs2005$s90b,
                          hlfs2004$s81b,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[82] <- "isara_kanal1"
full[,82] <- integ(full[,82])




#___İş arama kanal3?###
full[,83] <- data.frame(c(hlfs2016$isara_kanal3,
                          hlfs2015$isara_kanal3,
                          hlfs2014$isara_kanal3,
                          hlfs2013$s78c,
                          hlfs2012$s78c,
                          hlfs2011$s78c,
                          hlfs2010$s78c,
                          hlfs2009$s78c,
                          hlfs2008$s90c,
                          hlfs2007$s90c,
                          hlfs2006$s90c,
                          hlfs2005$s90c,
                          hlfs2004$s81c,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[83] <- "isara_kanal3"
full[,83] <- integ(full[,83])


#___İş arama kanal4?###
full[,84] <- data.frame(c(hlfs2016$isara_kanal4,
                          hlfs2015$isara_kanal4,
                          hlfs2014$isara_kanal4,
                          hlfs2013$s78d,
                          hlfs2012$s78d,
                          hlfs2011$s78d,
                          hlfs2010$s78d,
                          hlfs2009$s78d,
                          hlfs2008$s90d,
                          hlfs2007$s90d,
                          hlfs2006$s90d,
                          hlfs2005$s90d,
                          hlfs2004$s81d,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[84] <- "isara_kanal4"
full[,84] <- integ(full[,84])


#___İş arama kanal5?###
full[,85] <- data.frame(c(hlfs2016$isara_kanal5,
                          hlfs2015$isara_kanal5,
                          hlfs2014$isara_kanal5,
                          hlfs2013$s78e,
                          hlfs2012$s78e,
                          hlfs2011$s78e,
                          hlfs2010$s78e,
```

```
                          hlfs2009$s78e,
                          hlfs2008$s90e,
                          hlfs2007$s90e,
                          hlfs2006$s90e,
                          hlfs2005$s90e,
                          hlfs2004$s81e,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[85] <- "isara_kanal5"
full[,85] <- integ(full[,85])



#___İş arama kanal6?###
full[,86] <- data.frame(c(hlfs2016$isara_kanal6,
                          hlfs2015$isara_kanal6,
                          hlfs2014$isara_kanal6,
                          hlfs2013$s78f,
                          hlfs2012$s78f,
                          hlfs2011$s78f,
                          hlfs2010$s78f,
                          hlfs2009$s78f,
                          hlfs2008$s90f,
                          hlfs2007$s90f,
                          hlfs2006$s90f,
                          hlfs2005$s90f,
                          hlfs2004$s81f,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[86] <- "isara_kanal6"
full[,86] <- integ(full[,86])



#___İş arama kanal7?###
full[,87] <- data.frame(c(hlfs2016$isara_kanal7,
                          hlfs2015$isara_kanal7,
                          hlfs2014$isara_kanal7,
                          hlfs2013$s78g,
                          hlfs2012$s78g,
                          hlfs2011$s78g,
                          hlfs2010$s78g,
                          hlfs2009$s78g,
                          hlfs2008$s90h,
                          hlfs2007$s90h,
                          hlfs2006$s90h,
                          hlfs2005$s90h,
                          hlfs2004$s81h,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[87] <- "isara_kanal7"
full[,87] <- integ(full[,87])


#___İş arama kanal8?###
full[,88] <- data.frame(c(hlfs2016$isara_kanal8,
                          hlfs2015$isara_kanal8,
                          hlfs2014$isara_kanal8,
                          hlfs2013$s78h,
                          hlfs2012$s78h,
                          hlfs2011$s78h,
                          hlfs2010$s78h,
                          hlfs2009$s78h,
                          hlfs2008$s90i,
                          hlfs2007$s90i,
                          hlfs2006$s90i,
                          hlfs2005$s90i,
                          hlfs2004$s81i,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[88] <- "isara_kanal8"
full[,88] <- integ(full[,88])
```

```
#___İş arama kanal9?###
full[,89] <- data.frame(c(hlfs2016$isara_kanal9,
                          hlfs2015$isara_kanal9,
                          hlfs2014$isara_kanal9,
                          hlfs2013$s78i,
                          hlfs2012$s78i,
                          hlfs2011$s78i,
                          hlfs2010$s78i,
                          hlfs2009$s78i,
                          hlfs2008$s90j,
                          hlfs2007$s90j,
                          hlfs2006$s90j,
                          hlfs2005$s90j,
                          hlfs2004$s81j,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[89] <- "isara_kanal9"
full[,89] <- integ(full[,89])


#___İş arama kanal10?###
full[,90] <- data.frame(c(hlfs2016$isara_kanal10,
                          hlfs2015$isara_kanal10,
                          hlfs2014$isara_kanal10,
                          hlfs2013$s78j,
                          hlfs2012$s78j,
                          hlfs2011$s78j,
                          hlfs2010$s78j,
                          hlfs2009$s78j,
                          hlfs2008$s90k,
                          hlfs2007$s90k,
                          hlfs2006$s90k,
                          hlfs2005$s90k,
                          hlfs2004$s81k,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[90] <- "isara_kanal10"
full[,90] <- integ(full[,90])



#___İş arama kanal11?###
full[,91] <- data.frame(c(hlfs2016$isara_kanal11,
                          hlfs2015$isara_kanal11,
                          hlfs2014$isara_kanal11,
                          hlfs2013$s78k,
                          hlfs2012$s78k,
                          hlfs2011$s78k,
                          hlfs2010$s78k,
                          hlfs2009$s78k,
                          hlfs2008$s90l,
                          hlfs2007$s90l,
                          hlfs2006$s90l,
                          hlfs2005$s90l,
                          hlfs2004$s81l,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[91] <- "isara_kanal11"
full[,91] <- integ(full[,91])



#___İş arama kanal12?###
full[,92] <- data.frame(c(hlfs2016$isara_kanal12,
                          hlfs2015$isara_kanal12,
                          hlfs2014$isara_kanal12,
                          hlfs2013$s78l,
                          hlfs2012$s78l,
                          hlfs2011$s78l,
                          hlfs2010$s78l,
                          hlfs2009$s78l,
                          hlfs2008$s90m,
                          hlfs2007$s90m,
```

```
                          hlfs2006$s90m,
                          hlfs2005$s90m,
                          hlfs2004$s81m,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[92] <- "isara_kanal12"
full[,92] <- integ(full[,92])



#___İş arama kanal13?###
full[,93] <- data.frame(c(hlfs2016$isara_kanal13,
                          hlfs2015$isara_kanal13,
                          hlfs2014$isara_kanal13,
                          hlfs2013$s78m,
                          hlfs2012$s78m,
                          hlfs2011$s78m,
                          hlfs2010$s78m,
                          hlfs2009$s78m,
                          hlfs2008$s90n,
                          hlfs2007$s90n,
                          hlfs2006$s90n,
                          hlfs2005$s90n,
                          hlfs2004$s81n,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[93] <- "isara_kanal13"
full[,93] <- integ(full[,93])



#___İş arama isco?###
full[,94] <- data.frame(c(hlfs2016$isco08_ara_k,
                          hlfs2015$isco08_ara_k,
                          hlfs2014$isco08_ara_k,
                          hlfs2013$s79kod,
                          hlfs2012$s79kod,
                          hlfs2011$s79kod,
                          hlfs2010$s79kod,
                          hlfs2009$s79kod,
                          hlfs2008$s91kod,
                          hlfs2007$s91kod,
                          hlfs2006$s91kod,
                          hlfs2005$s91kod,
                          hlfs2004$s82kod,
                          rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[94] <- "isco08_ara_k"
full[,94] <- integ(full[,94])







#___İş arama durum###
full[,95] <- data.frame(c(hlfs2016$ara_is_durum,
                          hlfs2015$ara_is_durum,
                          hlfs2014$ara_is_durum,
                          hlfs2013$s80a,
                          hlfs2012$s80a,
                          hlfs2011$s80a,
                          hlfs2010$s80a,
                          hlfs2009$s80a,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[95] <- "ara_is_durum"
full[,95] <- integ(full[,95])




#___İş arama şekli####
```

```
#Bir önceki soru ile birleştirilerek şıklar derlenecek
full[,96] <- data.frame(c(hlfs2016$ara_is_sekil,
                          hlfs2015$ara_is_sekil,
                          hlfs2014$ara_is_sekil,
                          hlfs2013$s80b,
                          hlfs2012$s80b,
                          hlfs2011$s80b,
                          hlfs2010$s80b,
                          hlfs2009$s80b,
                          hlfs2008$s92,
                          hlfs2007$s92,
                          hlfs2006$s92,
                          hlfs2005$s92,
                          hlfs2004$s92,
                          hlfs2003$s39,
                          hlfs2002$s39,
                          hlfs2001$s39,
                          hlfs2000$s39))

colnames(full)[96] <- "ara_is_sekil"
full[,96] <- integ(full[,96])


full[,96][full$yil<=2003 & full$ara_is_sekil==3] <- 6
full[,96][full$yil<=2003 & full$ara_is_sekil==2] <- 4
full[,96][full$yil<=2003 & full$ara_is_sekil==1] <- 2


full[,96][full$yil>2008  & full$ara_is_sekil==5] <- 6
full[,96][full$yil>2008  & full$ara_is_sekil==4] <- 5
full[,96][full$yil>2008  & full$ara_is_sekil==3] <- 4
full[,96][full$yil>2008  & full$ara_is_sekil==2] <- 3
full[,96][full$yil>2008  & full$ara_is_sekil==1] <- 2

full[,96][full$yil>2008  & full$ara_is_sekil==1] <- 1


#74. sorunun eksik kısımlarını da buradan derleyebilirim
full[,96][full$yil<2009  & full$ara_is_durum==1] <- 1
full[,96][full$yil<2009  & full$ara_is_durum>1] <- 2


full[,96] <- integ(full[,96])


#___İş arama Süresi####
#Oldukça önemli bir soru, şıklarda yıl bazında birr yığılma var,
#kategorilere dönüştürülebilir! 47ay 8, 48ay 2884, 49ay 6 kişi, uygun değil.
full[,97] <- data.frame(c(hlfs2016$isara_sure,
                          hlfs2015$isara_sure,
                          hlfs2014$isara_sure,
                          hlfs2013$s81,
                          hlfs2012$s81,
                          hlfs2011$s81,
                          hlfs2010$s81,
                          hlfs2009$s81,
                          hlfs2008$s93,
                          hlfs2007$s93,
                          hlfs2006$s93,
                          hlfs2005$s93,
                          hlfs2004$s84,
                          hlfs2003$s40,
                          hlfs2002$s40,
                          hlfs2001$s40,
                          hlfs2000$s40))

colnames(full)[97] <- "isara_sure"
full[,97] <- integ(full[,97])




#___İş aramama nedenleri####
```

```
#Şıklarda ciddi uyumsuzluklar ver, defterde kodladım
full[,98] <- data.frame(c(hlfs2016$isaramama_neden,
                          hlfs2015$isaramama_neden,
                          hlfs2014$isaramama_neden,
                          hlfs2013$s83,
                          hlfs2012$s83,
                          hlfs2011$s83,
                          hlfs2010$s83,
                          hlfs2009$s83,
                          hlfs2008$s94,
                          hlfs2007$s94,
                          hlfs2006$s94,
                          hlfs2005$s94,
                          hlfs2004$s85,
                          hlfs2003$s37,
                          hlfs2002$s37,
                          hlfs2001$s37,
                          hlfs2000$s37))

colnames(full)[98] <- "isaramama_neden"
full[,98] <- integ(full[,98])

#2014-2016 arasındaki uyumluluklar
full[,98][full$yil>=2014  & full$isaramama_neden==61] <- 6
full[,98][full$yil>=2014  & full$isaramama_neden==62] <- 8
full[,98][full$yil>=2014  & full$isaramama_neden==63] <- 99
full[,98][full$yil>=2014  & full$isaramama_neden==64] <- 10
full[,98][full$yil>=2014  & full$isaramama_neden==65] <- 11
full[,98][full$yil>=2014  & full$isaramama_neden==13] <- 12
full[,98][full$yil>=2014  & full$isaramama_neden==9] <- 13
full[,98][full$yil>=2014  & full$isaramama_neden==99] <- 9
full[,98][full$yil>=2014  & full$isaramama_neden==98] <- 14

#2008-2005 arasındaki uyumsuzluklar
full[,98][full$yil>=2005 & full$yil<=2008  & full$isaramama_neden==3] <- 2
full[,98][full$yil>=2005 & full$yil<=2008  & full$isaramama_neden==14] <- NA
full[,98][full$yil>=2005 & full$yil<=2008  & full$isaramama_neden==15] <- 14

#2004 dönemindeki uyumsuzlukar, önceki 8,9,10 için 8'i bir kaç defa saydırdım.
full[,98][full$yil==2004 & full$isaramama_neden==3] <- 2
full[,98][full$yil==2004 & full$isaramama_neden==13] <- 14
full[,98][full$yil==2004 & full$isaramama_neden==11] <- 13
full[,98][full$yil==2004 & full$isaramama_neden==10] <- 12
full[,98][full$yil==2004 & full$isaramama_neden==9] <- 11


#2003 öncesindeki dönemindeki uyumsuzlukar
full[,98][full$yil<=2003 & full$isaramama_neden==3] <- 2
full[,98][full$yil<=2003 & full$isaramama_neden==12] <- 14
full[,98][full$yil<=2003 & full$isaramama_neden==11] <- 13
full[,98][full$yil<=2003 & full$isaramama_neden==10] <- 11
full[,98][full$yil<=2003 & full$isaramama_neden==6] <- 12
full[,98][full$yil<=2003 & full$isaramama_neden==7] <- 12
full[,98][full$yil<=2003 & full$isaramama_neden==8] <- 7

full[,98] <- integ(full[,98])

#___Bakım nedenleri####
full[,99] <- data.frame(c(hlfs2016$bakim_neden_issiz,
                          hlfs2015$bakim_neden_issiz,
                          hlfs2014$bakim_neden_issiz,
                          hlfs2013$s84,
                          hlfs2012$s84,
                          hlfs2011$s84,
                          hlfs2010$s84,
                          hlfs2009$s84,
                          rep(NA,sum(h8,h7,h6,h5,h4,h3,h2,h1,h0))))

colnames(full)[99] <- "bakim_neden_issiz"
full[,99] <- integ(full[,99])



#___İş bulsa hemen işe başlar mıydı?####
```

```
full[,100] <- data.frame(c(hlfs2016$isbasi_issiz,
                           hlfs2015$isbasi_issiz,
                           hlfs2014$isbasi_issiz,
                           hlfs2013$s85,
                           hlfs2012$s85,
                           hlfs2011$s85,
                           hlfs2010$s85,
                           hlfs2009$s85,
                           hlfs2008$s86,
                           hlfs2007$s86,
                           hlfs2006$s86,
                           hlfs2005$s86,
                           hlfs2004$s86,
                           hlfs2003$s41,
                           hlfs2002$s41,
                           hlfs2001$s41,
                           hlfs2000$s41))

colnames(full)[100] <- "isbasi_issiz"
full[,100] <- integ(full[,100])



#___İş bulsa referans haftasında işe başlar mıydı?####
full[,101] <- data.frame(c(hlfs2016$gecmis_is,
                           hlfs2015$gecmis_is,
                           hlfs2014$gecmis_is,
                           hlfs2013$s91,
                           hlfs2012$s91,
                           hlfs2011$s91,
                           hlfs2010$s91,
                           hlfs2009$s91,
                           hlfs2008$s102,
                           hlfs2007$s102,
                           hlfs2006$s102,
                           hlfs2005$s102,
                           hlfs2004$s89,
                           hlfs2003$s43,
                           hlfs2002$s43,
                           hlfs2001$s43,
                           hlfs2000$s43))

colnames(full)[101] <- "gecmis_is"
full[,101] <- integ(full[,101])

#2003 öncesindeki uyumsuzluklar
full[,101][full$yil<=2003 & full$gecmis_is!=8] <- 1
full[,101][full$yil<=2003 & full$gecmis_is==8] <- 2

full[,101] <- integ(full[,101])




#___İşten ayrılma yılı####
full[,102] <- data.frame(c(hlfs2016$is_ayril_yil,
                           hlfs2015$is_ayril_yil,
                           hlfs2014$is_ayril_yil,
                           hlfs2013$s92a,
                           hlfs2012$s92a,
                           hlfs2011$s92a,
                           hlfs2010$s92a,
                           hlfs2009$s92a,
                           hlfs2008$s103a,
                           hlfs2007$s103a,
                           hlfs2006$s103a,
                           hlfs2005$s103a,
                           hlfs2004$s90a,
                           rep(NA,sum(h3,h2,h1,h0))))


colnames(full)[102] <- "is_ayril_yil"
full[,102] <- integ(full[,102])
```

```
#___İşten ayrılma ayı####
full[,103] <- data.frame(c(hlfs2016$is_ayril_ay,
                           hlfs2015$is_ayril_ay,
                           hlfs2014$is_ayril_ay,
                           hlfs2013$s92b,
                           hlfs2012$s92b,
                           hlfs2011$s92b,
                           hlfs2010$s92b,
                           hlfs2009$s92b,
                           hlfs2008$s103b,
                           hlfs2007$s103b,
                           hlfs2006$s103b,
                           hlfs2005$s103b,
                           hlfs2004$s90b,
                           rep(NA,sum(h3,h2,h1,h0))))


colnames(full)[103] <- "is_ayril_ay"
full[,103] <- integ(full[,103])



#___İşten ayrılma neden####
full[,104] <- data.frame(c(hlfs2016$is_ayril_neden,
                           hlfs2015$is_ayril_neden,
                           hlfs2014$is_ayril_neden,
                           hlfs2013$s93,
                           hlfs2012$s93,
                           hlfs2011$s93,
                           hlfs2010$s93,
                           hlfs2009$s93,
                           hlfs2008$s104,
                           hlfs2007$s104,
                           hlfs2006$s104,
                           hlfs2005$s104,
                           hlfs2004$s91,
                           hlfs2003$s47,
                           hlfs2002$s47,
                           hlfs2001$s47,
                           hlfs2000$s47))


colnames(full)[104] <- "is_ayril_neden"
full[,104] <- integ(full[,104])

full[,104][full$yil>=2014 & full$is_ayril_neden==98] <- 11

full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==1] <- 111
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==2] <- 1
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==3] <- 2
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==4] <- 3
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==5] <- 3
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==6] <- 4
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==7] <- 4
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==8] <- 4
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==9] <- 5
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==10] <- 6
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==111] <- 9
full[,104][full$yil<=2003 & full$yil>=2002 & full$is_ayril_neden==12] <- 11


full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==1] <- 111
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==2] <- 1
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==3] <- 2
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==4] <- 3
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==5] <- 4
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==6] <- 4
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==7] <- 4
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==8] <- 5
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==9] <- 6
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==10] <- 7
full[,104][full$yil<=2001 & full$yil>=2000 & full$is_ayril_neden==111] <- 9
```

```
full[,104] <- integ(full[,104])



#___Eski işin NACE kodu####
full[,105] <- data.frame(c(hlfs2016$nace2_gecmis_k,
                           hlfs2015$nace2_gecmis_k,
                           hlfs2014$nace2_gecmis_k,
                           hlfs2013$s94kod,
                           hlfs2012$s94kod,
                           hlfs2011$s94kod,
                           hlfs2010$s94kod,
                           hlfs2009$s94kod,
                           hlfs2008$s105kod,
                           hlfs2007$s105kod,
                           hlfs2006$s105kod,
                           hlfs2005$s105kod,
                           hlfs2004$s92kod,
                           rep(NA,sum(h3,h2,h1,h0))))



colnames(full)[105] <- "nace2_gecmis_k"
full[,105] <- integ(full[,105])


full[,105][full$yil==2009 & full$nace2_gecmis_k==3] <- 2
full[,105][full$yil==2009 & full$nace2_gecmis_k==5] <- 3
full[,105][full$yil==2009 & full$nace2_gecmis_k==6] <- 44
full[,105][full$yil==2009 & full$nace2_gecmis_k==9] <- 5
full[,105][full$yil==2009 & full$nace2_gecmis_k==10] <- 66
full[,105][full$yil==2009 & full$nace2_gecmis_k==7] <- 10
full[,105][full$yil==2009 & full$nace2_gecmis_k==4] <- 10
full[,105][full$yil==2009 & full$nace2_gecmis_k==8] <- 10
full[,105][full$yil==2009 & full$nace2_gecmis_k==44] <- 4
full[,105][full$yil==2009 & full$nace2_gecmis_k==66] <- 6



full[,105][full$yil<=2008 & full$nace2_gecmis_k==3] <- 2
full[,105][full$yil<=2008 & full$nace2_gecmis_k==5] <- 3
full[,105][full$yil<=2008 & full$nace2_gecmis_k==6] <- 44
full[,105][full$yil<=2008 & full$nace2_gecmis_k==8] <- 66
full[,105][full$yil<=2008 & full$nace2_gecmis_k==7] <- 10
full[,105][full$yil<=2008 & full$nace2_gecmis_k==4] <- 10
full[,105][full$yil<=2008 & full$nace2_gecmis_k==8] <- 10
full[,105][full$yil<=2008 & full$nace2_gecmis_k==44] <- 4
full[,105][full$yil<=2008 & full$nace2_gecmis_k==66] <- 6

full[,105] <- integ(full[,105])


#___İşten ayrılma neden####
full[,106] <- data.frame(c(hlfs2016$isco08_gecmis_k,
                           hlfs2015$isco08_gecmis_k,
                           hlfs2014$isco08_gecmis_k,
                           hlfs2013$s95kod,
                           hlfs2012$s95kod,
                           hlfs2011$s95kod,
                           hlfs2010$s95kod,
                           hlfs2009$s95kod,
                           hlfs2008$s106kod,
                           hlfs2007$s106kod,
                           hlfs2006$s106kod,
                           hlfs2005$s106kod,
                           hlfs2004$s93kod,
                           rep(NA,sum(h3,h2,h1,h0))))

colnames(full)[106] <- "isco08_gecmis_k"
full[,106] <- integ(full[,106])
```

```
#___Geçmiş işteki durum####
full[,107] <- data.frame(c(hlfs2016$isteki_durum_gecmis,
                           hlfs2015$isteki_durum_gecmis,
                           hlfs2014$isteki_durum_gecmis,
                           hlfs2013$s96,
                           hlfs2012$s96,
                           hlfs2011$s96,
                           hlfs2010$s96,
                           hlfs2009$s96,
                           hlfs2008$s107,
                           hlfs2007$s107,
                           hlfs2006$s107,
                           hlfs2005$s107,
                           hlfs2004$s94,
                           hlfs2003$s46,
                           hlfs2002$s46,
                           hlfs2001$s46,
                           hlfs2000$s46))

colnames(full)[107] <- "isteki_durum_gecmis"
full[,107] <- integ(full[,107])



#___Geçen yılki işin NACE kodu####
full[,108] <- data.frame(c(hlfs2016$nace2_gecenyil_k,
                           hlfs2015$nace2_gecenyil_k,
                           hlfs2014$nace2_gecenyil_k,
                           hlfs2013$s98kod,
                           hlfs2012$s98kod,
                           hlfs2011$s98kod,
                           hlfs2010$s98kod,
                           hlfs2009$s98kod,
                           hlfs2008$s109kod,
                           hlfs2007$s109kod,
                           hlfs2006$s109kod,
                           hlfs2005$s109kod,
                           hlfs2004$s96kod,
                           rep(NA,sum(h3,h2,h1,h0))))



colnames(full)[108] <- "nace2_gecenyil_k"
full[,108] <- integ(full[,108])


full[,108][full$yil==2009 & full$nace2_gecenyil_k==3] <- 2
full[,108][full$yil==2009 & full$nace2_gecenyil_k==5] <- 3
full[,108][full$yil==2009 & full$nace2_gecenyil_k==6] <- 44
full[,108][full$yil==2009 & full$nace2_gecenyil_k==9] <- 5
full[,108][full$yil==2009 & full$nace2_gecenyil_k==10] <- 66
full[,108][full$yil==2009 & full$nace2_gecenyil_k==7] <- 10
full[,108][full$yil==2009 & full$nace2_gecenyil_k==4] <- 10
full[,108][full$yil==2009 & full$nace2_gecenyil_k==8] <- 10
full[,108][full$yil==2009 & full$nace2_gecenyil_k==44] <- 4
full[,108][full$yil==2009 & full$nace2_gecenyil_k==66] <- 6



full[,108][full$yil<=2008 & full$nace2_gecenyil_k==3] <- 2
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==5] <- 3
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==6] <- 44
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==8] <- 66
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==7] <- 10
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==4] <- 10
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==8] <- 10
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==44] <- 4
full[,108][full$yil<=2008 & full$nace2_gecenyil_k==66] <- 6

full[,108] <- integ(full[,108])
```

```
#___Geçmiş yıldaki durum####
full[,109] <- data.frame(c(hlfs2016$isteki_durum_gecenyil,
                           hlfs2015$isteki_durum_gecenyil,
                           hlfs2014$isteki_durum_gecenyil,
                           hlfs2013$s99,
                           hlfs2012$s99,
                           hlfs2011$s99,
                           hlfs2010$s99,
                           hlfs2009$s99,
                           hlfs2008$s110,
                           hlfs2007$s110,
                           hlfs2006$s110,
                           hlfs2005$s97,
                           hlfs2004$s94,
                           rep(NA,sum(h3,h2,h1,h0))))


colnames(full)[109] <- "isteki_durum_gecenyil"
full[,109] <- integ(full[,109])


#####################################################################
#*******************************************************************#
##**********************BURAYA KADAR RUN*************************##
#*******************************************************************#
#####################################################################



#Veri şeması####
sink(file=paste0(path,'VeriSemasi.txt'))
str(full)
sink()

#İngilizce değişken adları belirleme####
#write.csv(colnames(full),file=paste0(prevpath,'HLFS-Merge-Vars.csv'), sep=';')

#names(full) <- data.frame(read.csv(file=paste0(
#prevpath,'HLFS-Merge-Vars-Eng.csv'), sep=','))[,2]

#Deflated Income####
full$income.def <- full$gelir_gecenay_k/full$deflator


save(full,
file='/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2016_full_base.rdata')

load(
file='/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2016_full_base.rdata')



#Düzeltmeler####


#İşgücüne dahil olmayan nüfusu çıkar####
full <- full[full$t1<3,]

#NA olanları çıkar####
full <- full[!is.na(full$t1)==T,]
full <- full[!is.na(full$t0)==T,]

#t0 ile uyumlaştır, 2 unemp'i 0 yap
full$t1[full$t1==2] <- 0



#DERIVED Variables####
full$ikamettr <- full$yil-full$tr_yil
full$ikamettr <- integ(full$ikamettr)
```

```
full$ikametil <- full$yil-full$buil_yil
full$ikametil <- integ(full$ikamettr)

full$experience2 <- full$experience^2
full$experience2 <- integ(full$experience2)

full$inactive <- full$yil-full$is_ayril_yil
full$inactive[full$inactive<0] <- NA
full$inactive  <- integ(full$inactive)


rownames(full) <- NULL

save(full,
file='/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2016_full.rdata')

load(
file='/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2016_full.rdata')

#write.csv(full, file = "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2016_full.
```

3.2 R Codes for Variable Generation

```
## -*- coding: utf-8 -*-
#"""
#Created on Fri Sep 08 18:47:52 2017

######################################################################
#                                                                    #
#     _    _    __ _   ___  (_)  _ __    | | __ _    _ | |_  _    _ | | __     #
#    | | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    < | |_| || |_ | |_| ||   <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                          #
#     ____                                 _  _                      #
#    / __ \    __ _   _ __  ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_    _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                   #
######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web    : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
######################################################################

# Yasin Kütük
# HLFS ek değişkenler ve sadece ekonomik olarak mantıklı değişkenler,
#feature engineering yapmıyorum
print("HLFS ek değişkenler ve sadece ekonomik olarak mantıklı
      değişkenler, feature engineering yapmıyorum")
# R Kodları

#Options####
digits=4
options("scipen"=10,digits=digits)


#Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,"systemfit",
              "foreign", "feather", "dplyr",
              "sampleSelection","systemfit",
              "openxlsx")
paketkurulum(paketler)

#.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
}
loadlib(x)



#Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle
path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/"
veriler <- list.files(path=path)
pos = grep('full.rdata', veriler)
load(paste0(path,veriler[pos]))
attach(full)

#Hangi veriler var, kontrol et?####
colnames(full)
```

```
#NA summary####
nasummary <- function(v){
  if(!any(is.na(v))){
    res <- c(summary(v),"NA's"=0)
  } else{
    res <- summary(v)
  }
  return(res)
}


#[1] "year"                "formid"              "individual"
#"gender"              "age"                 "age.group"
#"birth.loc"           "abroad.exp6"
#[9] "tr.coming.year"      "living.place"        "living.year"
#"reference"           "school"              "school.field"
#"isco.08"             "isco.88"
#[17] "nace.rev2"          "workplace.status"    "workplace.start.year"
#"working.start.year"  "isco.88.merged"      "job.find"
#"weekly.working.hours" "income"
#[25] "nuts1"              "nuts2"               "t1"
#"working.status"      "education.year"      "experience"
#"prev.residence"      "prev.living"
#[33] "spouse"             "mother"              "father"
#"literacy"            "recent.school.grad"  "contd.school"
#"contd.school.year"   "course.attandence"
#[41] "course.aim"         "maritial.status"     "working"
#"working.one.hour"    "temp.job"            "unemp.reason"
#"t0"                  "deflator"
#[49] "id"                 "income.def"



#Mincer denklemindeki gibi exp^2 ekle####
full$experience2 <- (experience)^2

#Türkiye'ye gelinen yıl ile kaç yıldır burada ikamet ediyor veya yurtdışı deneyimi hesapla
full$tr.coming.exp <- (year-tr.coming.year)


#Hanehalkındaki çocuk sayısı

full$hh.children <-



#Modified dosyayı farklı isimle kaydetme####
save(full,
     file = "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2014_full_mod.rdata")
```

3.3 R Codes for Analysis

```
## -*- coding: utf-8 -*-
#"""
#Created on Wed Feb 28 18:47:52 2018

#######################################################################
#                          _          _              _          _      #
#     _   _   __ _   ___  (_)  _ __   | | __ _    _  | |_  _   _ | | __  #
#    | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                                 _  _                       #
#    / __ \    __ _  _ __ ___     __ _ (_)| |     ___  ___    _ __ ___   #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_   _ \  #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                    #
#######################################################################
#@author: Yasin KÜTÜK          ######################################
#@web   : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################

# Yasin Kütük
# HLFS Soru ve yanıtları uyumlaştırma sistemi
print("HLFS Statistical Computations")
# R Kodları


#Initials#####
options(digits = 4)
if(.Platform$OS.type=="windows"){
  path="d://Dropbox//_Dissertation//Yazim//Paper1_ML//Analyses//"
  respath="d://Dropbox//_Dissertation//Yazim//Paper1_ML//Analyses//Results//"
  print("Hocam Windows'dasın!")
} else {
  path="/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper1_ML/Analyses/"
  respath="/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper1_ML/Analyses/Results/"
  print("Abi Linux bu!")
}



####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,"systemfit",  "foreign",
              "Hmisc", "dplyr", "sampleSelection","systemfit",
              "openxlsx", "stargazer", "reporttools", "tab" ,
              "xtable", "margins", "pROC", "InformationValue", "mfx")
paketkurulum(paketler)

####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
}
loadlib(x)


#Load the Data####
load(
file='/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/hlfs2000-2016_full.rdata')
```

```
# Data Restriction, sadece geçmiş dönemde işsiz olanlar dikkate alınacak
full <- full[full$t0==0 & full$yil>2003,]


#Labeling####
full$yil <- factor(full$yil,
                        levels = seq(2004,2016),
                        labels = c('2004', '2005', '2006', '2007', '2008', '2009', '2010',
                                   '2011', '2012', '2013', '2014', '2015', '2016'))


full$cinsiyet <- factor(full$cinsiyet,
                        levels = seq(1,2),
                        labels = c('Male', 'Female'))


full$yas.k <- factor(full$yas.k,
                        levels = seq(4,14),
                        labels = c('15 -19', '20-24', '25-29', '30-34', '35-39',
                                   '40-44', '45-49', '50-54', '55-59', '60-64', '65+'))



full$dogum_yer <- factor(full$dogum_yer,
                        levels = seq(1,2),
                        labels = c('InTurkey', 'Abroad'))



full$yurtdisi_durum <- factor(full$yurtdisi_durum,
                         levels = seq(1,2),
                         labels = c('Yes', 'No'))

full$yakinlik <- factor(full$yakinlik,
                            levels = seq(1,11),
                            labels = c('Household head', 'Spouse','His/her children',
                                       'Father/Mother', 'Brother/Sister',
                                       'Father in law/Mother in law',
                                       'Son in law/Daughter in law',
                                       'Grandchild', 'Other relatives',
                                       'Non-relatives',
                                       'Housekeeper staying at home'))



full$okul_biten_k <- factor(full$okul_biten_k,
                           levels = seq(0,6),
                           labels = c('Smaller than six years old',
                                      'Literate but not completed any educational institutio
                                      'Primary school',
                                      'Secondary school',
                                      'High school',
                                      'Vocational or technical high school',
                                      'Higher education'))


full$foet99_k <- factor(full$foet99_k,
                         levels = seq(1,21),
                         labels = c('Teacher training and education science',
                                    'Arts', 'Humanities',
                                    'Social and behavioural science',
                                    'Journalism and information',
                                    'Business and administration',
                                    'Law', 'Life science',
                                    'Physical science', 'Mathematics and statistics',
                                    'Computing',
                                    'Engineering and engineering trades',
                                    'Manufacturing and processing',
                                    'rchitecture and building', 'Agriculture,
                                    forestry and fishery',
```

```
                                    'Veterinary','Health',
                                    'Social services', 'Personal services',
                                    'Transport services and environtal protection',
                                    'Security services'))

full$nuts1 <- factor(full$nuts1,
                     levels = seq(1,12),
                     labels = c('TR1 İstanbul', 'TR2 West Marmara', 'TR3 Aegean',
                                'TR4 East Marmara', 'TR5 West Anatolia',
                                'TR6 Mediterranean',
                                'TR7 Central Anatolia', 'TR8 West Black Sea',
                                'TR9 East Black Sea', 'TRA Northeast Anatolia',
                                'TRB Centraleast Anatolia', 'TRC Southeast Anatolia'))


full$nuts2 <- factor(full$nuts2,
                   levels = seq(1,26),
                   labels = c('TR10 İstanbul',
                              'TR21 Edirne, Tekirdağ, Kırklareli',
                              'TR22 Balıkesir Çanakkale' , 'TR31 İzmir',
                              'TR32 Denizli, Aydın, Muğla',
                              'TR33 Manisa, Afyonkarahisar, Kütahya, Uşak',
                              'TR41 Bursa, Eskişehir, Bilecik',
                              'TR42 Kocaeli, Sakarya, Düzce, Bolu, Yalova',
                              'TR51 Ankara','TR52 Konya, Karaman',
                              'TR61 Antalya, Isparta, Burdur',
                              'TR62 Adana, Mersin','TR63 Hatay,
                            Kahramanmaraş, Osmaniye',
                              'TR71 Nevşehir, Aksaray, Niğde, Kırıkkale, Kırşehir',
                              'TR72 Kayseri, Sivas, Yozgat',
                              'TR81 Zonguldak, Karabük, Bartın',
                              'TR82 Kastamonu, Çankırı, Sinop',
                              'TR83 Samsun, Tokat, Çorum, Amasya',
                              'TR90 Trabzon, Ordu, Giresun, Rize, Artvin, Gümüşhane',
                              'TRA1 Erzurum, Erzincan, Bayburt',
                              'TRA2 Kars, Ağrı, Iğdır, Ardahan',
                              'TRB1 Malatya, Elazığ, Bingöl, Tunceli',
                              'TRB2 Van, Muş, Bitlis, Hakkari',
                              'TRC1 Gaziantep, Adıyaman, Kilis',
                              'TRC2 Diyarbakır, Şanlıurfa',
                              'TRC3 Siirt, Mardin, Batman, Şırnak'))




full$onceki_ikamet <- factor(full$onceki_ikamet,
                     levels = seq(1,2),
                     labels = c('In Turkey','In a foreign country'))


full$yerlesim_tur <- factor(full$yerlesim_tur,
                            levels = seq(1,3),
                            labels = c('Province center',
                                       'District center',
                                       'Sub-district or a village'))



full$fertno_es <- factor(full$fertno_es,
                           levels = seq(1,8),
                           labels = c('Reference person',
                                      'Spouse (or cohabiting partner)',
                                      'Child of reference person',
                                      'Bride or bridegroom',
                                      'Grandchild', 'Mother/father in law',
                                      'Other relatives', 'Non-relatives'))


full$okur_yazar <- factor(full$okur_yazar,
                        levels = seq(1,2),
                        labels = c('Yes','No'))
```

```
full$egitim_devam_k <- factor(full$egitim_devam_k,
                          levels = seq(1,2),
                          labels = c('Yes','No'))


full$okul_devam_k <- factor(full$okul_devam_k,
                            levels = seq(1,6),
                            labels = c('Primary education','High school',
                                       'Vocational or technical high school',
                                       'Open education system',
                                       'Faculty/high school',
                                       'Master/doctorate"'))

full$sinif_devam <- factor(full$sinif_devam,
                           levels = seq(1,9),
                           labels = c('1' ,'2','3','4','5','6','7','8','9'))

full$kurs <- factor(full$kurs,
                            levels = seq(1,2),
                            labels = c('Yes','No'))


full$kurs.amaci <- factor(full$kurs.amaci,
                    levels = seq(1,3),
                    labels = c('For Carrier','For Personal or Social','Other'))


full$medeni_durum <- factor(full$medeni_durum,
                    levels = seq(1,4),
                    labels = c('Single','Married','Divorced', 'Widowed'))

full$gecmis_is <- factor(full$gecmis_is,
                    levels = seq(1,2),
                    labels = c('Yes','No'))

full$is_ayril_neden <- factor(full$is_ayril_neden,
                     levels = c(seq(1,10),98),
                     labels = c('Job was temporary, came to an end',
                          'as working seasonally',
                          'Dismissed,Liquidated,Bankrupted',
                          'Not satisfied with job',
                          'Own illness or disability',
                          'Looking after children or incapacitated adults in the family',
                          "Her spouse's request / Due to marriage",
                          'Education or training',
                          'Retirement', 'Military service',
                          'Other'))

full$nace2_gecmis_k <- factor(full$nace2_gecmis_k,
                      levels = seq(1,11),
                      labels = c('Agriculture',
'Manufacturing, mining and quarrying and other industries',
'Construction',
'Wholesale retail trade, transportation and storage, accommodation and service',
'Information and communication', 'Financial and insurance activities',
'Real estate activities',
'Professional, scientific and technical activities, administrative and support services',
'Public administration and defense, education, health and social services',
'Other service activities', 'Unknown'))


full$nace2_gecenyil_k <- factor(full$nace2_gecenyil_k,
                        levels = seq(1,11),
                        labels = c('Agriculture',
'Manufacturing, mining and quarrying and other industries',
'Construction',
'Wholesale retail trade, transportation and storage, accommodation and   service',
'Information and communication',
'Financial and insurance activities',
'Real estate activities',
'Professional, scientific and technical activities, administrative and support services',
'Public administration and defense, education, health and social services',
'Other service activities', 'Unknown'))
```

```
full$isco08_gecmis_k <- factor(full$isco08_gecmis_k,
                      levels = seq(1,9),
                         labels = c('Managers','Professionals',
'Technicians and associate professionals',
'Clerical support workers',
'Service and sales workers',
'Skilled agricultural, forestry and fishery workers',
'Craft and related trades workers ',
'Plant and machine operators, and assemblers',
'Elemantary occupations'))



full$isteki_durum_gecmis <- factor(full$isteki_durum_gecmis,
                                levels = seq(1,4),
                                labels = c('Regular employee or casual','Employer',
                                           'Self employed','Unpaid family worker'))


full$isteki_durum_gecenyil <- factor(full$isteki_durum_gecenyil,
                                levels = seq(1,4),
                                labels = c('Regular employee or casual',
                                           'Employer','Self employed',
                                           'Unpaid family worker'))



full$t1 <- factor(full$t1, levels = seq(0,1),  labels = c('Unemp','Emp'))



#Tabulative Statistics####
sink(file=paste0(respath,"01.Tabulation.txt"))
k1 <- data.frame(table(full$t0, full$yil))
k2 <- data.frame(table(full$t1, full$yil))

k <- data.frame(merge(k1,k2,by=c('Var1','Var2')))

k <- k[order(k[,2]),]
k <- k[,c(2,1,3,4)]

colnames(k) <- c('Year','t0', 't1_Unemp', 't1_Emp')
k$t0 <- factor(k$t0 , levels = c(0,1), labels = c("Unemp", "Emp"))

k[dim(k)[1] +1,] <- c('', '', 'SUM', sum(as.numeric(k$t1_Emp)))

row.names(k) <- NULL

print(xtable(k), include.rownames=FALSE)
sink()


#Data frame properties####
sink(file=paste0(respath,"02.DataFrameStr.txt"))

str(full)
sink()




summaryfunction= function (x){
  if( is.numeric(x)!=TRUE) {stop("Supplied X is not numeric")}
  mysummary = data.frame(
    "Min." =as.numeric( min(x)),
    "1st Qu." = quantile(x,na.rm = T)[2],
    "Median" = median(x),
    "Mean" = mean(x),
    "3rd Qu." = quantile(x,na.rm = T)[4],
    "Max." = max(x),
    "St.Dev." = sd(x),
    row.names=""
```

```
  )
  names(mysummary) = c("Min.","1st␣Qu.","Median","Mean","3rd␣Qu.","Max.", "St.Dev.")
  return( mysummary )
}


#Quantitative Vars Selection####
quan <- c("yas","ikamettr","ikametil", "educ_year",
          "experience", "deflator","agirlik_katsayisi",
          "hh_buyukluk", "hh_buyukluk_estimated", "inactive")
#Qualitative Vars
qual <- c('t1','cinsiyet', 'yas.k', 'dogum_yer',
          'yurtdisi_durum','yakinlik', 'okul_biten_k',
          'foet99_k', 'nuts1', 'nuts2', 'onceki_ikamet',
          'yerlesim_tur', 'fertno_es', 'fertno_anne',
          'fertno_baba', 'okur_yazar', 'egitim_devam_k', 'okul_devam_k',
          'sinif_devam', 'kurs', 'kurs.amaci', 'medeni_durum',
          'gecmis_is', 'is_ayril_neden', 'nace2_gecmis_k', 'isco08_gecmis_k',
          'isteki_durum_gecmis', 'nace2_gecenyil_k', 'isteki_durum_gecenyil')



desc <- full[,quan]

tabu <- full[,qual]


#Tabulation Descriptives####
sink(file=paste0(respath,"03.Descriptives.txt"))
stargazer(desc)
sink()




#Descriptive Statistics
sink(file=paste0(respath,"04.Tabulates.txt"))

xtable(summary(tabu[,1:4]))
xtable(summary(tabu[,5:8]))
xtable(summary(tabu[,9:12]))
xtable(summary(tabu[,13:16]))
xtable(summary(tabu[,17:20]))
xtable(summary(tabu[,21:24]))
xtable(summary(tabu[,25:28]))
sink()



#Logit Application####
#Median-NA Filler####


#Qualitative Data, NA is a factor
for(i in 1:ncol(tabu)){
  tabu[, i] <- addNA(tabu[, i])
}

#Quantitative Data NA <- Median Replacer
for(i in 1:ncol(desc)){
  desc[is.na(desc[,i]), i] <- median(desc[,i], na.rm = TRUE)
}

logitdata <- data.frame(tabu,desc)
logitdata$t1 <- full$t1



sink(file=paste0(respath,"05.Logit.txt"))
logit <- glm(t1~.,data=logitdata, binomial(link = "logit"))
```

```
#Estimation for Marginal Effects
logitm <- logitmfx(t1~.,data=logitdata)
stargazer(logitm$mfxest,
          apply.coef = exp,
          apply.se = exp,
          title            =  "Logistic␣Regression",
          dep.var.caption  = "Employment␣Transition␣Probability",
          dep.var.labels   = 'P(Emp=1)',
          nobs = T,keep.stat = c("n", "adj.rsq","rsq","f","ser"))


sink()


logitdata$probs=predict(logit,type=c("response"))


logitdata$expectedt1 <- ifelse(logitdata$probs>=0.5,1,0)



sink(file=paste0(respath,"05.Logit_Predictions.txt"))
library(pROC)
g <- roc(t1 ~ expectedt1, data = logitdata)


wi <-8
he <- 8
pdf(paste0(respath,'LogitROC.pdf'), width = wi, height = he)
plot(g)
dev.off()



#Metrics about Logit Regression
conf <- data.frame(confusionMatrix(logitdata$t1,
                                   logitdata$expectedt1, threshold = 0.5))
conf <- conf[,1:2]
conf

precision <- conf[1,1]/ sum(conf[1,])
recall <- conf[1,1]/ sum(conf[,1])
f1 <-2/ (precision^-1 + recall^-1)
accuracy <- (conf[1,1]+conf[2,2])/( sum(conf[,1])+sum(conf[,2]))

print('Precision␣is')
precision

print('Recall␣is')
recall

print('F1␣is')
f1

print('Accuracy␣is')
accuracy
sink()

#Stata Ported Margin Package
if (!require("remotes")) {
  install.packages("remotes")
  library("remotes")
}
install_github("leeper/prediction")
install_github("leeper/margins")

# building vignettes takes a moment, so for a quicker install set:
install_github("leeper/margins", build_vignettes = FALSE)

(marg1 <- margins(logit))
```

3.4 Python Codes for Feature (Variable) Engineering

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 08 18:47:52 2017

#######################################################################
#                      _                 _                 _        #
#     _    _    __ _   ___  (_)  _ __    | |  __ _    _ |_  _    _ |  | __     #
#    | |  | |  / _  |/ __|| || '_ \  | |/ /| |  | || __|| |  | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||   <  | |_| || |_ | |_| ||   <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                              _  _                          #
#    / __ \    __ _   _ __ ___     __ _ (_)| |    ___   ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |   / __|/ _ \ | '_   _ \   #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
"""

# Kütüphaneleri yükle
from __future__ import print_function

from sklearn.feature_extraction import DictVectorizer
from tabulate import tabulate
import numpy as np
import pandas as pd
import os
import gc
import matplotlib.pyplot as plt
import seaborn as sns
import re
import sys
from rpy2.robjects import r
from rpy2.robjects import pandas2ri

pandas2ri.activate()
import matplotlib.pyplot as plt
from scipy import stats

# use seaborn plotting defaults
import seaborn as sns; sns.set()

from sklearn.pipeline import make_pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.cross_validation import StratifiedKFold, cross_val_score, KFold
from sklearn.grid_search import GridSearchCV
from sklearn.ensemble.gradient_boosting import GradientBoostingClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold
from sklearn.metrics import accuracy_score, log_loss

from keras.models import Sequential
from keras.layers import Dense
from keras import optimizers
from keras.utils import plot_model
# import pydot
# import graphviz

import xgboost as xgb  # Bilgisayar kitleniyor, xgboost'u ayarla
```

```
import plotly.offline as py

py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls

# Çok ciddi uyarılar yoksa, uyarı gelmesin!
import warnings

warnings.filterwarnings('ignore')

# HLFS Datalarını R'den alıyorum
fileloc = "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/"

# Klasördeki dosyalar
p = sns.color_palette()
print('# File sizes')
for f in os.listdir(fileloc):
    if 'zip' not in f:
        print(f.ljust(30) +
              str(round(os.path.getsize(fileloc + f) / 1000000, 2)) + 'MB')

# HLFS'deki Rdata yüklenecek
rfile = r['load'](fileloc + os.listdir(fileloc)[
    os.listdir(fileloc).index('hlfs2000-2016_full.rdata')])
# dataframe'in adı full idi

hlfs = pd.DataFrame(r['full'])

# Intro
hlfs.shape
hlfs.describe()
hlfs.info()
len(hlfs.index) - hlfs.count()



# Imputer or Median filler uygulama
feats = list(hlfs.columns.values)
for col in hlfs:
    hlfs[col].fillna(hlfs[col].median(), inplace=True)

len(hlfs.index) - hlfs.count()

# Some plots
# By Transition
t0_unemp_count = hlfs[hlfs['t0'] == 2]['t1'].value_counts()
t0_emp_count = hlfs[hlfs['t0'] == 1]['t1'].value_counts()
df = pd.DataFrame([t0_unemp_count, t0_emp_count])
df.index = ['Prev.Term.Unemp-Emp', 'Prev.Term.Unemp-Unemp']
plt.interactive(False)
df.plot(kind='bar', stacked=False, figsize=(15, 8))
plt.show()



figure = plt.figure(figsize=(15,8))
plt.hist(hlfs[hlfs['t0'] == 2]['t1'], stacked=True, color = ['g'],
         bins = 30,label = ['Survived','Dead'])
plt.xlabel('Fare')
plt.ylabel('Number of passengers')
plt.legend()
plt.show()


hlfs.t0.describe()
hlfs.t1.describe()  # Kontrol t0=2 (işsiz) ve t1=1,2 olmalı

# Data Restriction, sadece geçmiş dönemde işsiz olanlar dikkate alınacak
hlfs = hlfs[(hlfs['t1'] == 0) | (hlfs['t1'] == 1)]
hlfs = hlfs[hlfs['t0'] == 0.0]

# NA replacement for RData
hlfs = hlfs.replace('-2147483648', np.NaN)
```

```
####Initial runlar buraya kadar Hocam!######
#############################################################

characteristics=['year', 'formid', 'individual',
                 'gender', 'age', 'age.group', 'birth.loc',
                 'abroad.exp6', 'tr.coming.year', 'living.place',
 'living.year', 'reference', 'school', 'school.field',
 'jobprevyear',  'nuts1', 'nuts2',
 'education.year', 'experience', 'prev.residence',
 'prev.living', 'spouse', 'mother', 'father',
 'literacy', 'recent.school.grad', 'contd.school',
 'contd.school.year', 'course.attandence', 'course.aim', 'maritial.status',
  't0', 'deflator', 'id', 'factor', 'hh_population_estimated',
 'hh_population', 'jobseek.dummy', 'jobseek.prevstatus',
 'jobseek.trough1', 'jobseek.trough2',
 'jobseek.trough3', 'jobseek.trough4', 'jobseek.trough5',
 'jobseek.trough6', 'jobseek.trough7', 'jobseek.trough8',
 'jobseek.trough9', 'jobseek.trough10', 'jobseek.trough11',
 'jobseek.trough12', 'jobseek.trough13', 'jobseek.isco',
 'jobseek.status', 'jobseek.type', 'jobseek.duration',
 'jobseek.no.reason', 'care.reason', 'job.unemp', 'job.prev',
 'jobleft.year', 'jobleft.month', 'jobleft.reason',
 'jobprev.nace', 'jobprev.isco', 'jobprev.status', 'jobprevyear.nace',
 'jobprevyear.status']

hlfscols = list(hlfs.columns)
hlfsnotcols = list(set(hlfscols) - set(characteristics))



# Feature yeri ve processingler
def status(feature):
    print('Processing', feature, ':␣ok')


# Train and Test Sampling, General
X_train, X_test, y_train, y_test = \
train_test_split(hlfs.loc[:, hlfs.columns.difference(hlfsnotcols)],\
hlfs.t1, test_size=0.2  )







# training=train.drop('t1', 1, inplace=True)

#RandomForest Classifier Selection

# Feature Selection
clf = RandomForestClassifier(bootstrap=False, class_weight=None, criterion='gini',
            max_depth=20, max_features='auto', max_leaf_nodes=None,
             min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
            oob_score=False, random_state=0, verbose=0, warm_start=False)
clf = clf.fit(X_train, y_train)


predicted = clf.predict(X_test)
accuracy = accuracy_score(y_test, predicted)
accuracy




# Use a grid over parameters of interest
clf = RandomForestClassifier()
param_grid = {'max_features': ['sqrt', 'auto', 'log2'],
           "n_estimators" : [19,38],
           "max_depth" : [1, 5, 10, 15, 20, 25, 30],
```

```
            "min_samples_leaf" : [1, 2, 4, 6, 8, 10],
            'bootstrap': [True, False]}

CV_rfc = GridSearchCV(estimator=clf, param_grid=param_grid, cv= 10)
CV_rfc.fit(X_train, y_train)
print CV_rfc.best_params_


#https://ahmedbesbes.com/how-to-score-08134-in-titanic-kaggle-challenge.html
# Açılacak ve run_gs eğer GS iyi çalışmıyorsa.
run_gridsearch = True

if run_gridsearch:
    parameter_grid = {
        'max_depth': [4, 6, 8],
        'n_estimators': [140, 20],
        'max_features': ['sqrt', 'auto', 'log2']
        'min_samples_split': [1, 3, 10],
        'min_samples_leaf': [1, 3, 10],
        'bootstrap': [True, False],
    }
    forest = RandomForestClassifier()
    cross_validation = StratifiedKFold(n_splits=10)

    grid_search = GridSearchCV(forest,
                               scoring='accuracy',
                               param_grid=parameter_grid,
                               cv=cross_validation)
    grid_search.fit(X_train,y_train)
    model = grid_search
    parameters = grid_search.best_params_
    print('Best␣score:␣{}'.format(grid_search.best_score_))
    print('Best␣parameters:␣{}'.format(grid_search.best_params_))
else:
    parameters = {'bootstrap': False, 'min_samples_leaf': 3, 'n_estimators': 50,
                  'min_samples_split': 10, 'max_features': 'sqrt', 'max_depth': 6}

    model = RandomForestClassifier(**parameters)
    model.fit(X_train,y_train)

# Accuracy tahmini için
def compute_score(clf, X_test, y_test, scoring='accuracy'):
    xval = cross_val_score(clf, X_test, y_test, cv=5, scoring=scoring)
    return np.mean(xval)


acc = compute_score(clf, y_train, y_test, scoring='accuracy') * 100
# Percentile olarak göstermek için 100 ile çarp.

print "Yasin,␣Random␣Forest␣Classifier'ın␣\
tutarlılığı␣tam␣olarak␣%%%s)␣kadar!" % acc

'''
Bir önceki run sonucum:
Yasin, Random Forest Classifier'ın tutarlılığı \
tam olarak %99.9982591669) kadar!
'''




# Train and Test Sampling, Stratified Yearly
kfold_splits = 10  # i determined geneal k-fold
skf = StratifiedKFold(n_splits=kfold_splits, shuffle=False)
# Shuffle false since i do not want to use them again.

# Loop through the indices the split() method returns
for index, (train_indices, val_indices) in enumerate(skf.split(X, y)):
    print "Training␣on␣fold␣" + str(index + 1) + "/10..."
    # Generate batches from indices
    xtrain, xval = X[train_indices], X[val_indices]
    ytrain, yval = y[train_indices], y[val_indices]
```

```
    # Clear model, and create it
    model = None
    model = create_model()
    # Debug message I guess
    # print "Training new iteration on " + str(xtrain.shape[0]) +
    #" training samples, " + str(xval.shape[0]) +
    #" validation samples, this may be a while..."
    history = train_model(model, xtrain, ytrain, xval, yval)
    accuracy_history = history.history['acc']
    val_accuracy_history = history.history['val_acc']
    print "Last␣training␣accuracy:␣" + str(accuracy_history[-1])
    + ",␣last␣validation␣accuracy:␣" + str(val_accuracy_history[-1])


def get_combined_data():
    # target'lar sonrasında silinecek
    targets = train.t1
    train.drop('t1', 1, inplace=True)
    combined = train.append(test)
    combined.reset_index(inplace=True)
    combined.drop('index', inplace=True, axis=1)
    return combined


combined = get_combined_data()
combined.shape
combined.info()




features = pd.DataFrame()
features['feature'] = X_train.columns
features['importance'] = clf.feature_importances_
features.sort_values(by=['importance'], ascending=True, inplace=True)
features.set_index('feature', inplace=True)
features.plot(kind='barh', figsize=(20, 20))
plt.show()

# RandomForestClassifier'a göre model hyperparametering
# Train Partitioning
model = SelectFromModel(clf, prefit=True)
train_reduced = model.transform(X_train)
train_reduced.shape

# Test Partitioning
test_reduced = model.transform(X_test.loc[:, X_test.columns != 't1'])
test_reduced.shape



# SVM ve Logistic Regs Tahminlerim
# Parameters başlangıçları

model = SVC(kernel='linear')
model.fit(X_train, y_train)

plt.scatter(hlfs.nuts1, hlfs.experience, c=hlfs.t1,  cmap='autumn')
plt.show()


classifiers = [
    SVC(probability=True),
    LogisticRegression()]

log_cols = ["Classifier", "Accuracy"]
log = pd.DataFrame(columns=log_cols)

sss = StratifiedShuffleSplit(n_splits=10, test_size=0.1, random_state=0)

X = hlfs.loc[:, hlfs.columns != 't1']
X = X.values
y = hlfs['t1']
```

```
y = y.values

acc_dict = {}

for train_index, test_index in sss.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    for clf in classifiers:
        name = clf.__class__.__name__
        clf.fit(X_train, y_train)
        train_predictions = clf.predict(X_test)
        acc = accuracy_score(y_test, train_predictions)
        if name in acc_dict:
            acc_dict[name] += acc
        else:
            acc_dict[name] = acc
        print("Hocam " + str(clf)[:3] + " Tamamlandı")

for clf in acc_dict:
    acc_dict[clf] = acc_dict[clf] / 10.0
    log_entry = pd.DataFrame([[clf, acc_dict[clf]]], columns=log_cols)
    log = log.append(log_entry)

scorelogit = acc_dict.values()[0] * 100
scoresvm = acc_dict.values()[1] * 100
print "Hocam, Logistic Reg'in başarısı: %%%s" % scorelogit
print "Hocam, Support Vector Machine'in başarısı: %%%s" % scoresvm

'''
Bir önceki run sonucum:
Yasin,  Logistic Reg'in başarısı: %86.0
Support Vector Machine'in başarısı: %91.32 !
'''

plt.xlabel('Accuracy')
plt.title('Classifier Accuracy')

sns.set_color_codes("muted")
sns.barplot(x='Accuracy', y='Classifier', data=log, color="b")
plt.show()

# ANN ile ML tahminim ama arayüz Keras ve tek layer, geriplanda Tensorflow
#ile GPU run yapılacak
# Parametrelerim
epochs = 10
batch_size = 10
valid_split = 0.2
learning_rate = 1

# Sevgili inputlar
# X=train
# y=targets
# X_test=test
# y_test=

# create model
model = Sequential()
model.add(Dense(2, input_dim=49, activation='tanh'))
model.add(Dense(1, activation='sigmoid'))
# Compile model
sgd = optimizers.SGD(lr=learning_rate, clipnorm=1.)
adam = optimizers.Adam(lr=learning_rate, beta_1=0.9, beta_2=0.999,
                       epsilon=1e-08, decay=0.0)
model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
# Fit the model
model.fit(X, y, epochs=epochs, batch_size=batch_size,
          validation_split=valid_split)
# evaluate the model
scores = model.evaluate(X, y)
sonuc = str(model.metrics_names[1]) + str(scores[1] * 100)
print "Hocam Artifial neural networks başarısı: %%%s" % sonuc

# Tam olarak modelim şöyle bir şey:
```

```
# plot_model(model)

# from IPython.display import SVG
# from keras.utils.vis_utils import model_to_dot

SVG(model_to_dot(model).create(prog='dot', format='svg'))
```

3.5 Python Codes for Classifiers in Machine Learning

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 08 18:47:52 2017

######################################################################
#                          _          _              _          _     #
#     _   _    __ _   ___ (_) _ __   | | __ _   _  | |_  _   _ | | __ #
#    | | | |  / _  |/ __|| || '_ \  | |/ /| | | || __|| | | || |/ /    #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <     #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\    #
#     |___/                                                          #
#     ____                                 _  _                      #
#    / __ \    __ _   _ __ ___     __ _ (_)| |     ___   ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_   _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                     #
######################################################################
#@author: Yasin KÜTÜK          ######################################
#@web   : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
######################################################################
"""

# Kütüphaneleri yükle
from __future__ import print_function

from sklearn.feature_extraction import DictVectorizer
from tabulate import tabulate
import numpy as np
import pandas as pd
import os
import gc
import matplotlib.pyplot as plt
import seaborn as sns
import re
import sys

from rpy2.robjects import r
from rpy2.robjects import pandas2ri

pandas2ri.activate()
import matplotlib.pyplot as plt

from sklearn.pipeline import make_pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectKBest
from sklearn.ensemble.gradient_boosting import GradientBoostingClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, VotingClassifier,\
BaggingClassifier, AdaBoostClassifier, ExtraTreesClassifier, \
GradientBoostingClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.svm import SVC, NuSVC, SVR, LinearSVC
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.model_selection import StratifiedShuffleSplit, StratifiedKFold, \
GroupKFold, ShuffleSplit, GridSearchCV
from sklearn.metrics import accuracy_score, auc, \
roc_auc_score,cohen_kappa_score
from sklearn import metrics
from sklearn.naive_bayes import GaussianNB, BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis, \
QuadraticDiscriminantAnalysis
from sklearn import preprocessing
from sklearn.calibration import CalibratedClassifierCV
from mlxtend.classifier import EnsembleVoteClassifier
from sklearn.multiclass import OneVsRestClassifier
import itertools
```

```
from keras import backend as K
import os

def set_keras_backend(backend):

    if K.backend() != backend:
        os.environ['KERAS_BACKEND'] = backend
        reload(K)
        assert K.backend() == backend

set_keras_backend("tensorflow")



from keras.models import Sequential
from keras.layers import Dense
from keras import optimizers, initializers
from keras.utils import plot_model
# import pydot
# import graphviz
from keras import backend as K
import os
import time


from xgboost import XGBClassifier # Bilgisayar kitleniyor, xgboost'u ayarla



import plotly.offline as py


py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls

# Çok ciddi uyarılar yoksa, uyarı gelmesin!
import warnings
warnings.filterwarnings('ignore')

# HLFS Datalarını R'den alıyorum
fileloc = "/media/DRIVE/Dropbox/_VeriTabanlarim/HLFS/RData/"




# Klasördeki dosyalar
p = sns.color_palette()
print('#_File_sizes')
for f in os.listdir(fileloc):
    if 'zip' not in f:
        print(f.ljust(30) +
              str(round(os.path.getsize(fileloc + f) / 1000000, 2)) + 'MB')



# HLFS'deki Rdata yüklenecek
rfile = r['load'](fileloc + os.listdir(fileloc)[
    os.listdir(fileloc).index('hlfs2000-2016_full.rdata')])

hlfs = pd.DataFrame(r['full'])



# Intro
hlfs.shape
hlfs.describe()
hlfs.info()
len(hlfs.index) - hlfs.count()


# NA replacement for RData
```

```
tic = time.clock()
hlfs = hlfs.replace(-2147483648, 999)
toc = time.clock()
toc - tic



# Imputer or Median filler uygulama
feats = list(hlfs.columns.values)
for col in hlfs:
    hlfs[col].fillna(hlfs[col].median(), inplace=True)

len(hlfs.index) - hlfs.count()







hlfs.t0.describe()
hlfs.t1.describe()  # Kontrol t0=2 (işsiz) ve t1=1,2 olmalı


# Data Restriction, sadece geçmiş dönemde işsiz olanlar dikkate alınacak,
#2003 öncesi drop edilecek uyumsuzluktan dolayı
hlfs = hlfs[hlfs['t0'] == 0]
hlfs = hlfs[hlfs['yil']>2003]


#Index Resetting
hlfs = hlfs.reset_index()



####Initial runlar buraya kadar Hocam!######
################################################################################

characteristics=['yil',
                 'id',
                 'cinsiyet',
                 'yas',
                 'yas.k',
                 'dogum_yer',
                 'yurtdisi_durum',
                 'ikamettr',
                 'buil_yasama',
                 'ikametil',
                 'yakinlik',
                 'okul_biten_k',
                 'foet99_k',
                 'nuts1',
                 'nuts2',
                 'educ_year',
                 'experience',
                 'experience2',
                 'onceki_ikamet',
                 'yerlesim_tur',
                 'fertno_es',
                 'fertno_anne',
                 'fertno_baba',
                 'okur_yazar',
                 'egitim_devam_k',
                 'okul_devam_k',
                 'sinif_devam',
                 'kurs',
                 'kurs.amaci',
                 'medeni_durum',
                 'deflator',
                 'agirlik_katsayisi',
                 'hh_buyukluk',
                 'hh_buyukluk_estimated',
                 'gecmis_is',
```

```
                'inactive',
                'is_ayril_neden',
                'nace2_gecmis_k',
                'isco08_gecmis_k',
                'isteki_durum_gecmis',
                'nace2_gecenyil_k',
                'isteki_durum_gecenyil',
                't0']
```

```
#from sklearn.svm import SVC
#clf=SVC(probability=True)
#clf.fit(X,Y)
#print clf.predict_proba(W) #No error
```

```
hlfscols = list(hlfs.columns)
hlfsnotcols = list(set(hlfscols) - set(characteristics))
```

```
# Feature yeri ve processingler
def status(feature):
    print('Processing', feature, ':␣ok')
```

```
# Train and Test Sampling, General
#X_train, X_test, y_train, y_test = train_test_split(hlfs.loc[:,
# hlfs.columns.difference(hlfsnotcols)],hlfs.t1, test_size=0.2)
```

```
#Data and labels
X = hlfs.loc[:, hlfs.columns.difference(hlfsnotcols)]
y = hlfs.t1
```

```
##X ve Label ayrımı, grup stratifiedkfold için
#X = X.iloc[0:100000,]
#y = y.iloc[0:100000,]
#groups = hlfs.yil
```

```
##Normalization
#yil=X.yil
#X = X.drop('yil',1)
#min_max_scaler = preprocessing.MinMaxScaler()
#X = pd.DataFrame(min_max_scaler.fit_transform(X))
#X['yil'] = yil
```

```
groups = X.yil

#Her yıl'ı fold alıp, fold'lardan rassal veri seçtiriyorum
# Train and Test Sampling, Stratified Yearly
kfold_splits = 13  # i determined geneal k-fold
skf = StratifiedKFold(n_splits=kfold_splits, shuffle=True)
# Shuffle false since i do not want to use them again.
skf.get_n_splits(X, y, groups=X.yil)

print(skf)
```

```
for train_index, test_index in skf.split(X, y, groups=groups):
   print("TRAIN:", train_index, "TEST:", test_index)
   X_train, X_test = X.iloc[train_index], X.iloc[test_index]
   y_train, y_test = y.iloc[train_index], y.iloc[test_index]



#New metrics definition (Bülent Hoca's Feedbacks - October24/2018)
def gini_impurity(p):
   return (p)*(1 - (p)) + (1 - p)*(1 - (1-p))

def entropy(p):
   return - p*np.log2(p) - (1 - p)*np.log2((1 - p))

def classification_error(p):
   return 1 - np.max([p, 1 - p])



#https://github.com/wreardan/cart/blob/master/python/cart.py
#For whole reports
def reporter():
    print(model)
    expected = y_train
    predicted = model.predict(X_train)
    accuracy = accuracy_score(expected, predicted)
    print('Train accuracy is {}.'.format(accuracy))
    print(metrics.classification_report(expected, predicted))
    print(metrics.confusion_matrix(expected, predicted))
    expectedt = y_test
    predictedt = model.predict(X_test)
    accuracyt = accuracy_score(expectedt, predictedt)
    print('Test accuracy is {}.'.format(accuracyt))
    print(metrics.classification_report(expectedt, predictedt))
    print(metrics.confusion_matrix(expectedt, predictedt))
    probs = model.predict_proba(X_train)[:, 1]
    auc = roc_auc_score(y_train, probs)
    print('Train AUC is {}.'.format(auc))
    probst = model.predict_proba(X_test)[:, 1]
    auct = roc_auc_score(y_test, probst)
    print('Test AUC is {}.'.format(auct))
    gini_train = gini_impurity(model.predict_proba(X_train)[:,1][0])
    gini_test = gini_impurity(model.predict_proba(X_test)[:,1][0])
    print('Train Gini Impurity is {}.'.format(gini_train))
    print('Test Gini Impurity is {}.'.format(gini_test))
    cohen_train = cohen_kappa_score(expected, predicted, labels=None,\
                                    weights=None )
    cohen_test = cohen_kappa_score(expectedt, predictedt, labels=None,\
                                   weights=None)
    print("Train Cohen's Kappa is {}.".format(cohen_train))
    print("Test Cohen's Kappa is {}.".format(cohen_test))
    entropy_train = entropy(model.predict_proba(X_train)[:,1][0])
    entropy_test = entropy(model.predict_proba(X_test)[:,1][0])
    print('Train Entropy is {}.'.format(entropy_train))
    print('Test Entropy is {}.'.format(entropy_test))
    ci_train = classification_error(model.predict_proba(X_train)[:,1][0])
    ci_test = classification_error(model.predict_proba(X_test)[:,1][0])
    print('Train Classification Error is {}.'.format(ci_train))
    print('Test Classification Error is {}.'.format(ci_test))
    ent = [entropy(p) if p != 0 else None for p in probs]
    scaled_ent = [e*0.5 if e else None for e in ent]
    c_err = [classification_error(i) for i in probs]
    fig = plt.figure()
    ax = plt.subplot(111)
    for j, lab, ls, c, in zip(
          [ent, scaled_ent, gini_impurity(probs), c_err],
          ['Entropy', 'Entropy (scaled)', 'Gini Impurity', \
           'Misclassification Error'],
          ['-', '-', '--', '-.'],
          ['lightgray', 'red', 'green', 'blue']):
       line = ax.plot(probs, j, label=lab, linestyle=ls, lw=1, color=c)

    ax.legend(loc='upper left', bbox_to_anchor=(0.01, 0.85), ncol=1, \
              fancybox=True, shadow=False)
```

```
    ax.axhline(y=0.5, linewidth=1, color='k', linestyle='--')
    ax.axhline(y=1.0, linewidth=1, color='k', linestyle='--')

    plt.ylim([0, 1.1])
    plt.xlabel('p(j=1)')
    plt.ylabel('Impurity␣Index')
    plt.show()

    ent = [entropy(p) if p != 0 else None for p in probst]
    scaled_ent = [e*0.5 if e else None for e in ent]
    c_err = [classification_error(i) for i in probst]
    fig = plt.figure()
    ax = plt.subplot(111)
    for j, lab, ls, c, in zip(
          [ent, scaled_ent, gini_impurity(probst), c_err],
          ['Entropy', 'Entropy␣(scaled)', 'Gini␣Impurity',\
           'Misclassification␣Error'],
          ['-', '-', '--', '-.'],
          ['lightgray', 'red', 'green', 'blue']):
       line = ax.plot(probst, j, label=lab, linestyle=ls, lw=1, color=c)

    ax.legend(loc='upper␣left', bbox_to_anchor=(0.01, 0.85), ncol=1,\
              fancybox=True, shadow=False)

    ax.axhline(y=0.5, linewidth=1, color='k', linestyle='--')
    ax.axhline(y=1.0, linewidth=1, color='k', linestyle='--')

    plt.ylim([0, 1.1])
    plt.xlabel('p(j=1)')
    plt.ylabel('Impurity␣Index')
    plt.show()


#Naive Bayes Classifier

#Gaussian NB
tic = time.clock()
model = GaussianNB()
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic







#Bernoulli NB
tic = time.clock()
model = BernoulliNB()
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic



#SVM

#X ve Label ayrımı, grup stratifiedkfold için
#X_train = X_train.iloc[0:1000,]
#y_train = y_train.iloc[0:1000,]
#groups = hlfs.yil


#Support Vector Machines : approx 13h
tic = time.clock()
model = SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=True, random_state=True, shrinking=True,
```

```
  tol=0.001, verbose=False)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic



#LinearSVC
tic = time.clock()
model = LinearSVC(penalty='l2', loss='squared_hinge', dual=True, tol=0.0001,
                  C=1.0, multi_class='ovr',
fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0,
 random_state=None, max_iter=1000)
model = CalibratedClassifierCV(model)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic




#Decision TREES
# Extra Tree Classifier
tic = time.clock()
model = ExtraTreesClassifier(n_estimators=10, criterion='gini',
max_depth=None, min_samples_split=2, min_samples_leaf=1,
 min_weight_fraction_leaf=0.0,
max_features='auto', max_leaf_nodes=None,
bootstrap=False, oob_score=False, n_jobs=1,
random_state=None, verbose=0, warm_start=False, class_weight=None)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic


#CART
tic = time.clock()
model = DecisionTreeClassifier(criterion='gini',
splitter='best', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_features=None, random_state=None,
max_leaf_nodes=None,
class_weight=None, presort=False)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic




#Logistic Regression Tree
tic = time.clock()
model = LogisticRegression(penalty='l2', dual=False,
tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1,
class_weight=None, random_state=None, solver='liblinear',
max_iter=100, multi_class='ovr', verbose=0, warm_start=False, n_jobs=1)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic


#XGBoost Classifier fit model on training data
tic = time.clock()
model = XGBClassifier(learning_rate = 0.1, n_estimators = 1000, max_depth = 5,
min_child_weight = 1, gamma = 0, subsample = 0.8, colsample_bytree = 0.8,
objective = 'binary:logistic', nthread = 4, scale_pos_weight = 1,seed = 27)
```

```
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
#Discriminant Analysis
#Linear Discriminant
tic = time.clock()
model = LinearDiscriminantAnalysis(solver='lsqr', shrinkage='auto',
priors=None,n_components=None, store_covariance=False, tol=0.0001)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
#Quadratic Discriminant
tic = time.clock()
model = QuadraticDiscriminantAnalysis(priors=None, reg_param=0.78,
                store_covariances=None, tol=0.001)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
#Stochastic Gradient Descent
#Stochastic Gradient Descent Classifier
tic = time.clock()
model = SGDClassifier(loss='modified_huber', penalty='l2',
alpha=0.0001, l1_ratio=0.15, fit_intercept=True,
 shuffle=True, verbose=0, epsilon=0.1, n_jobs=1,
random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5,
class_weight=None, warm_start=False, average=False)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
#Neighbors Classifier
#KNeighbors Classifiers
tic = time.clock()
model = KNeighborsClassifier(n_neighbors=2, weights='uniform',
algorithm='auto', leaf_size=30, p=2, metric='minkowski',
metric_params=None, n_jobs=1)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
#Ensemble
#RandomForest Classifier
tic = time.clock()
model = RandomForestClassifier(bootstrap=False, class_weight=None,
                               criterion='gini',
```

```
            max_depth=20, max_features='auto', max_leaf_nodes=None,
            min_samples_leaf=20, min_samples_split=10,
            min_weight_fraction_leaf=0.0, n_estimators=20, n_jobs=1,
            oob_score=False, random_state=0, verbose=0, warm_start=False)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
# Bagging
tic = time.clock()
model = BaggingClassifier(base_estimator=None, n_estimators=10,
max_samples=1.0, max_features=1.0, bootstrap=True,
bootstrap_features=False, oob_score=False, warm_start=False,
n_jobs=1, random_state=None, verbose=0)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
# AdaBoost Classifier
tic = time.clock()
model = AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=1, n_estimators=100, random_state=None)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
# Gradient Boosting Classifier
tic = time.clock()
model = GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.45, loss='deviance', max_depth=3,
              max_features='sqrt', max_leaf_nodes=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=1000,
              presort='auto', random_state=True, subsample=0.9, verbose=0,
              warm_start=False)
model = model.fit(X_train, y_train)
reporter()
toc = time.clock()
toc - tic
```

```
#Shallow Artifical Neural Networks
tic = time.clock()
# ANN ile ML tahminim ama arayüz Keras ve tek layer, geriplanda Tensorflow ile
# GPU run yapılacak
# Parametrelerim
epochs = 10
batch_size = 100
valid_split = 0.1
learning_rate = 0.001

# Create model
ann = Sequential()
ann.add(Dense(128, input_dim=43, activation='tanh',
              kernel_initializer=initializers.random_normal(stddev=0.01) ))
```

```
ann.add(Dense(1, activation='softmax'))

# Compile model
sgd = optimizers.SGD(lr=learning_rate, clipnorm=.5)
adam = optimizers.Adam(lr=learning_rate, beta_1=0.9,
                       beta_2=0.999, epsilon=1e-08, decay=0.1)
ann.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
# Fit the model
ann.fit(X_train, y_train, epochs=epochs, batch_size=batch_size,
        validation_split=valid_split, shuffle=True)
# evaluate the model
scoretrain = ann.evaluate(X_train, y_train)[1]
scoretest = ann.evaluate(X_test, y_test)[1]

print('Train accuracy is {}.'.format(scoretrain))
print('Test accuracy is {}.'.format(scoretest))

predx = ann.predict_classes(X_train, verbose=0)
predxt = ann.predict_classes(X_test, verbose=0)

pred = ann.predict_proba(X_train, verbose=0)
roc = roc_auc_score(y_train, pred)
roc
print('Train ROC is {}.'.format(roc))
predt = ann.predict_proba(X_test, verbose=0)
roct = roc_auc_score(y_test, predt)
print('Test ROC is {}.'.format(roct))

toc = time.clock()
toc - tic



gini_train = gini_impurity(pred)
gini_test = gini_impurity(predt)
print('Train Gini Impurity is {}.'.format(gini_train))
print('Test Gini Impurity is {}.'.format(gini_test))
cohen_train = cohen_kappa_score(y_train, predx, labels=None, weights=None )
cohen_test = cohen_kappa_score(y_test, predxt, labels=None, weights=None)
print("Train Cohen's Kappa is {}.".format(cohen_train))
print("Test Cohen's Kappa is {}.".format(cohen_test))
entropy_train = entropy(pred)
entropy_test = entropy(predt)
print('Train Entropy is {}.'.format(entropy_train))
print('Test Entropy is {}.'.format(entropy_test))
ci_train = classification_error(pred)
ci_test = classification_error(predt)
print('Train Classification Error is {}.'.format(ci_train))
print('Test Classification Error is {}.'.format(ci_test))

ent = [entropy(p) if p != 0 else None for p in pred]
scaled_ent = [e*0.5 if e else None for e in ent]
c_err = [classification_error(i) for i in pred]
fig = plt.figure()
ax = plt.subplot(111)
for j, lab, ls, c, in zip(
      [ent, scaled_ent, gini(pred), c_err],
      ['Entropy', 'Entropy (scaled)', 'Gini Impurity',\
       'Misclassification Error'],
      ['-', '-', '--', '-.'],
      ['lightgray', 'red', 'green', 'blue']):
   line = ax.plot(pred, j, label=lab, linestyle=ls, lw=1, color=c)

ax.legend(loc='upper left', bbox_to_anchor=(0.01, 0.85), ncol=1,
          fancybox=True, shadow=False)

ax.axhline(y=0.5, linewidth=1, color='k', linestyle='--')
ax.axhline(y=1.0, linewidth=1, color='k', linestyle='--')

plt.ylim([0, 1.1])
plt.xlabel('p(j=1)')
plt.ylabel('Impurity Index')
plt.show()
```

```
ent = [entropy(p) if p != 0 else None for p in predt]
scaled_ent = [e*0.5 if e else None for e in ent]
c_err = [classification_error(i) for i in predt]
fig = plt.figure()
ax = plt.subplot(111)
for j, lab, ls, c, in zip(
      [ent, scaled_ent, gini(predt), c_err],
      ['Entropy', 'Entropy (scaled)', 'Gini Impurity',\
       'Misclassification Error'],
      ['-', '-', '--', '-.'],
      ['lightgray', 'red', 'green', 'blue']):
   line = ax.plot(predt, j, label=lab, linestyle=ls, lw=1, color=c)

ax.legend(loc='upper left', bbox_to_anchor=(0.01, 0.85), ncol=1,
          fancybox=True, shadow=False)

ax.axhline(y=0.5, linewidth=1, color='k', linestyle='--')
ax.axhline(y=1.0, linewidth=1, color='k', linestyle='--')

plt.ylim([0, 1.1])
plt.xlabel('p(j=1)')
plt.ylabel('Impurity Index')
plt.show()
```

```
#Ensemble Voting classifier
tic = time.clock()
#XGBoost Classifier fit model on training data
tic = time.clock()
m1 = XGBClassifier(learning_rate = 0.1, n_estimators = 1000, max_depth = 5,
min_child_weight = 1, gamma = 0, subsample = 0.8, colsample_bytree = 0.8,
objective = 'binary:logistic', nthread = 4, scale_pos_weight = 1,seed = 27)


#RandomForest Classifier
m2 = RandomForestClassifier(bootstrap=False, class_weight=None,
                            criterion='gini',
            max_depth=20, max_features='auto', max_leaf_nodes=None,
            min_samples_leaf=20, min_samples_split=10,
            min_weight_fraction_leaf=0.0, n_estimators=20, n_jobs=1,
            oob_score=False, random_state=0, verbose=0, warm_start=False)


# Extra Tree Classifier
m3 = ExtraTreesClassifier(n_estimators=10, criterion='gini',
max_depth=None, min_samples_split=2, min_samples_leaf=1,
 min_weight_fraction_leaf=0.0,
max_features='auto', max_leaf_nodes=None, bootstrap=False,
 oob_score=False, n_jobs=1,
random_state=None, verbose=0, warm_start=False, class_weight=None)


#Quadratic Discriminant
m4 = QuadraticDiscriminantAnalysis(priors=None, reg_param=0.78,
                                   store_covariances=None, tol=0.001)
```

```
#Model Ensemble
tic = time.clock()
classifiers=[m1, m2, m3, m4]
```

```
ensemble = EnsembleVoteClassifier(clfs=classifiers , voting='soft')

wei = [1,2,3,4]
weilist = list(itertools.permutations(wei))
weilist =[list(p) for p in weilist]

params = {'weights':weilist}
grid_Search = GridSearchCV(param_grid = params, estimator=ensemble)
grid_Search.fit(X_train, y_train)
print(grid_Search.best_params_)

#{'weights': [1, 3, 2, 4]}
reporter()
toc = time.clock()
toc - tic
```

Appendix C : Chapter 4 Appendices

Table C.1 : Descriptive Statistics for Quantitative Variables.

Statistic*	N	Mean	St. Dev.	Min	Max
yas	371,613	6.009	3.778	1	13
ailegrup	179,380	1.152	0.437	1	9
sagliks2	25	1.920	1.115	1	5
egitim	338,831	3.982	2.565	1	11
sure_yil	130,726	13.970	15.050	0	99
calsur	130,726	45.680	19.670	0	99
kac_kisi	37,937	3.035	1.847	1	9
ucrn_ay	47,656	699.700	959.300	0	26,641
ucra_ay	152,311	18.010	76.260	0	2,957
ucrn_yil	115,909	4,464	8,907	0	400,985
ikr_yil	47,656	346.400	2,110	0	131,049
prim_yil	63,211	185.100	1,632	0	166,618
ucra_yil	115,908	364.500	1,037	0	35,611
konayni	113,022	29.680	389.600	0	23,762
ulsayni	115,908	104.100	425.100	0	21,039
egsayni	115,908	5.964	139	0	26,564
yemayni	115,908	199.500	557.900	0	15,938
kresayni	115,908	0.637	45.020	0	7,320
giyayni	115,908	2.800	64.700	0	12,397
gidaayni	115,908	12.220	134.600	0	7,426
digayni	115,756	10.340	172.300	0	21,707
mutsn_ay	13,400	1,297	2,906	−5,566	74,257
mutsa_ay	138,262	2.931	41.890	0	3,703
mutsn_yl	20,607	12,119	30,650	−2,700	1,168,819
mutsa_yl	20,607	303.300	1,399	0	53,128
targ_ay	10,625	585.400	1,207	−1,020	21,651
targ_yl	16,894	5,689	12,090	−12,240	259,814
calsur_e	7,012	27.670	18.570	1	99
ekgln_ay	6,219	527	11,563	−1,113	906,427
ekgla_ay	7,943	4.721	64.250	0	4,458
ekgln_yl	6,219	4,405	15,364	−13,000	531,282
ekgla_yl	6,219	85.030	589.200	0	19,447
calay_yil	164,554	8.511	5.031	0	12
calsur_d	9,167	50.430	19.280	1	99
digln_yl	8,118	4,003	7,086	−7,426	177,094
digla_yl	8,118	215.400	775.500	0	21,251
gmkn_ay	200,408	17.430	206.900	0	30,445
gmka_ay	229,992	0.002	0.637	0	277.700
gmkn_yl	286,344	185.200	2,242	0	365,342

Table C.1 (Continued) : Descriptive Statistics for Quantitative Variables

Statistic*	N	Mean	St. Dev.	Min	Max
gmka_yl	286,344	12.860	419.300	0	87,410
mulkn_ay	200,408	8.358	202	−2,000	37,500
mulka_ay	217,614	0.005	1.065	0	410.900
banka_yl	286,344	35.510	1,449	0	450,000
doviz_yl	200,408	2.011	207.700	0	72,000
nema_yl	200,408	39.050	968.300	−7,000	285,699
temet_yl	200,408	9.549	947.500	−30,501	247,931
kar_yl	200,408	0.781	145	0	57,190
toprn_yl	205,848	21.680	805.100	0	138,133
topra_yl	205,848	0.905	116	0	46,672
trsn_ay	200,408	95.400	356.400	0	14,168
trsa_ay	230,038	3.191	57.450	0	11,931
emekl_yl	286,344	692.100	2,944	0	150,441
viade_yl	154,156	42.180	209.500	0	8,075
yasli_yl	286,344	8.031	119.600	0	20,769
sosy_yl	232,396	5.694	137.600	0	30,883
hasta_yl	85,936	7.216	117.200	0	5,232
dul_yl	286,344	121.500	1,030	0	74,379
gazi_yl	286,344	13.600	310.600	0	39,207
burs_yl	286,344	7.408	124.600	0	10,413
issiz_yl	286,344	2.074	139.600	0	58,340
ddestek_yl	196,456	45.260	658.900	0	95,692
dayni_yl	286,344	7.662	119.100	0	15,262
ydemk_yl	286,344	17.650	771.100	0	170,010
yddav_yl	286,344	10.010	417.400	0	64,356
ydayn_yl	286,344	0.429	52.220	0	17,558
dgnfk_yl	286,344	54.950	728.800	0	137,256
dgbrs_yl	123,087	51.480	1,051	0	123,966
dsemk_yl	143,459	15.220	622.800	0	77,008
dgayn_yl	286,344	24.080	476.100	0	198,017
topgn_yl	225,862	395.400	2,265	−5,566	907,480
topga_ay	205,540	19.080	97.700	0	11,931
topgelay	230,991	403.600	2,255	−5,499	907,480
topgna_yl	306,940	4,305	11,727	−13,000	851,999
topga_yl	286,345	211.900	957	0	198,636
topgelyl	306,940	4,536	13,490	−13,000	1,169,351
indeks_arl	333,029	1.051	0.042	0.991	1.241
kira_ay	104,514	128.700	219.100	0	4,427
loj_fert	7,004	1.069	0.276	1	3
izafi_k	301,213	170.600	268.600	0	7,525
konutsah	260,833	1,738	661.800	1	2,014
kon_sure	14,038	16.400	12.970	0	90
kon_borc	243,939	16,243	71,096	1	6,000,000
kondeger	309,677	35,502	63,934	0	1,672,177
otur_sur	323,897	13.100	12.400	0	99

Table C.1 (Continued) : Descriptive Statistics for Quantitative Variables

Statistic*	N	Mean	St. Dev.	Min	Max
bina_tar	371,613	3.823	0.934	1	5
oda_say	173,714	3.595	0.895	1	16
kon_alan	173,714	108	36.270	10	870
oda_say.1	197,899	3.520	0.819	1	10
kon_alan.1	197,899	102.700	28.320	25	700
konutiki	371,613	1.930	0.254	1	2
konut2_n	21,719	2.674	0.484	1	3
sudepo	80,220	0.126	0.331	0	1
telefon	371,613	0.659	0.495	0	5
ceptel	371,613	1.806	1.347	0	13
bilgisay	371,613	0.321	0.531	0	9
intern	371,613	0.221	0.459	0	9
televiz	259,600	1.231	0.560	0	5
video	179,380	0.054	0.231	0	3
lcdtele	192,233	0.368	0.571	0	8
dvd_vcd	259,600	0.314	0.472	0	3
kamera	371,613	0.041	0.209	0	4
uyduant	216,229	0.747	0.465	0	5
buzdolab	371,613	0.998	0.176	0	5
derindon	371,613	0.109	0.316	0	4
bulasik	371,613	0.372	0.486	0	3
m_firin	371,613	0.101	0.303	0	2
camasir	371,613	0.895	0.321	0	3
kurutma	371,613	0.010	0.106	0	2
haliyika	371,613	0.120	0.328	0	3
klima	371,613	0.128	0.428	0	9
otomobil	371,613	0.207	0.407	0	3
otoadet	185,940	0.647	0.509	0	5
isveroto	259,600	0.012	0.109	0	1
isotoadt	115,126	0.053	0.231	0	4
motosik	259,600	0.041	0.199	0	1
motoadet	122,716	0.161	0.381	0	3
deniztas	217,361	0.001	0.026	0	1
dentadet	112,155	0.002	0.052	0	3
muskonut	371,613	0.580	0.629	0	2
musmik	158,926	1.145	0.436	1	12
muskira	158,926	0.066	0.293	0	7
muskdeg	9,115	159.900	243.300	0	3,553
aptmik	131,267	1.378	1.014	1	51
aptkira	131,267	0.308	0.849	0	50
aptkdeg	27,777	318.900	549.400	0	11,438
yazmik	8,333	1.039	0.263	1	9
yazkira	8,333	0.033	0.229	0	8
yazkdeg	241	175.400	217.700	0	928.200
tarmik	82,294	47.210	91.230	1	2,200

Table C.1 (Continued) : Descriptive Statistics for Quantitative Variables

Statistic*	N	Mean	St. Dev.	Min	Max
tarkira	82,294	2.472	21.810	0	2,200
tarkdeg	6,355	202.200	938.100	0	13,613
bagmik	35,116	13.600	129	1	6,740
bagkira	35,116	0.195	2.600	0	130
bagkdeg	675	245.100	749.900	0	7,969
seramik	2,727	1,157	1,749	1	9,999
serakira	2,727	89.410	682.600	0	9,999
serakdeg	81	997.100	1,537	0	7,084
arsamik	15,996	716.300	1,075	1	9,999
arsakira	15,996	5.268	166.100	0	8,000
arsakdeg	122	411	596.800	0	2,500
arsadeg	5,552	34,844	291,137	0	8,233,841
dukmik	20,428	1.406	1.458	1	52
dukkira	20,428	0.663	1.428	0	52
dukkdeg	8,859	535.500	1,254	0	29,703
dukdeg	9,656	70,851	300,413	0	9,723,291
otelmik	241	1.145	0.491	1	4
otelkira	241	0.328	0.674	0	4
otelkdeg	60	609.500	1,037	0	3,900
oteldeg	116	77,183	204,028	0.086	1,045,883
digmik	1,185	0.838	1.405	0	7.084
digkira	1,185	0.203	0.718	0	3.542
digkdeg	231	678.400	1,329	0	7,426
digdeg	1,185	37,884	218,145	0	4,950,437
yilkulgel	371,613	16,896	28,041	0	1,372,821
efb_oecd	337,798	2.619	0.987	1	13.300
harcama	371,613	956	1,365	0	43,855
hhb	371,613	4.768	2.365	1	30
faktor	329,718	1,670	1,245	96.340	15,421
tasarruf	371,613	8,455	17,148	0.153	754,061
tasarruf_orani	371,613	0.274	0.183	001	0.966

*All nominal variables are deflated.

Table C.2 : Descriptive Statistics of Qualitative Features.

cinsiyet	yakinlik	sagliks	gunengel	calengel
Min. :1.00	Min. : 0.00	Min. :1.00	Min. :1	Min. :1
1st Qu.:1.00	1st Qu.: 0.00	1st Qu.:1.00	1st Qu.:2	1st Qu.:2
Median :2.00	Median : 2.00	Median :1.00	Median :2	Median :2
3rd Qu.:2.00	3rd Qu.: 2.00	3rd Qu.:3.00	3rd Qu.:2	3rd Qu.:2
Max. :2.00	Max. :10.00	Max. :4.00	Max. :2	Max. :2
			NA's :106356	NA's :106356
calengel	ogrenci	egitim_devam	medenidr	calis_ay
Min. :1	Min. :1	Min. :1	Min. :1	Min. :1
1st Qu.:2	1st Qu.:1	1st Qu.:1	1st Qu.:1	1st Qu.:1
Median :2	Median :2	Median :1	Median :2	Median :2
3rd Qu.:2	3rd Qu.:2	3rd Qu.:2	3rd Qu.:2	3rd Qu.:2
Max. :2	Max. :2	Max. :9	Max. :4	Max. :2
NA's :106356	NA's :89071	NA's :339608	NA's :85271	NA's :85271
calis_ay	is_arama	isneden	isbasi	calis_yl
Min. :1	Min. :1	Min. : 1	Min. :1	Min. :1
1st Qu.:1	1st Qu.:2	1st Qu.: 3	1st Qu.:2	1st Qu.:2
Median :2	Median :2	Median : 3	Median :2	Median :2
3rd Qu.:2	3rd Qu.:2	3rd Qu.: 4	3rd Qu.:2	3rd Qu.:2
Max. :2	Max. :2	Max. :11	Max. :2	Max. :2
NA's :85271	NA's :215997	NA's :231057	NA's :215997	NA's :259139
calis_yl	sekkod	meslek	cal_kisi	sgk
Min. :1	Min. : 1	Min. :0	Min. :1	Min. :1
1st Qu.:2	1st Qu.: 1	1st Qu.:4	1st Qu.:1	1st Qu.:1
Median :2	Median : 5	Median :6	Median :1	Median :3
3rd Qu.:2	3rd Qu.: 8	3rd Qu.:7	3rd Qu.:3	3rd Qu.:3
Max. :2	Max. :14	Max. :9	Max. :4	Max. :3
NA's :259139	NA's :241764	NA's :240887	NA's :283680	NA's :283680
sgk	is_statu	huk_dur	koopera	sir_kon
Min. :1	Min. :1	Min. :1	Min. :1	Min. :1
1st Qu.:1	1st Qu.:1	1st Qu.:1	1st Qu.:2	1st Qu.:2
Median :3	Median :1	Median :1	Median :2	Median :2
3rd Qu.:3	3rd Qu.:1	3rd Qu.:2	3rd Qu.:2	3rd Qu.:2
Max. :3	Max. :3	Max. :8	Max. :2	Max. :2
NA's :283680	NA's :283680	NA's :316554	NA's :352136	NA's :358724

Table C.2 (Continued) : Descriptive Statistics of Qualitative Features.

sir_kon	isdur	ist_tesk	guc_kul	sendika
Min. :1	Min. :1	Min. :1	Min. :1	Min. :1
1st Qu.:2	1st Qu.:1	1st Qu.:1	1st Qu.:1	1st Qu.:2
Median :2	Median :2	Median :1	Median :1	Median :2
3rd Qu.:2	3rd Qu.:4	3rd Qu.:1	3rd Qu.:3	3rd Qu.:2
Max. :2	Max. :5	Max. :3	Max. :3	Max. :2
NA's :358724	NA's :240887	NA's :297003	NA's :328340	NA's :323957
sendika	isfaal	mulk_dur	ekis_yl	ekis_ay
Min. :1	Min. :1	Min. :1	Min. :1	Min. :1
1st Qu.:2	1st Qu.:2	1st Qu.:1	1st Qu.:2	1st Qu.:1
Median :2	Median :2	Median :2	Median :2	Median :1
3rd Qu.:2	3rd Qu.:2	3rd Qu.:2	3rd Qu.:2	3rd Qu.:2
Max. :2	Max. :2	Max. :3	Max. :2	Max. :2
NA's :323957	NA's :278481	NA's :358213	NA's :283680	NA's :364601
ekis_ay	sekkod_e	meslek_e	isdur_e	cl_neden
Min. :1	Min. : 1	Min. :0	Min. :1	Min. :1
1st Qu.:1	1st Qu.: 1	1st Qu.:6	1st Qu.:2	1st Qu.:2
Median :1	Median : 1	Median :6	Median :4	Median :3
3rd Qu.:2	3rd Qu.: 6	3rd Qu.:8	3rd Qu.:4	3rd Qu.:6
Max. :2	Max. :14	Max. :9	Max. :5	Max. :7
NA's :364601	NA's :364630	NA's :364601	NA's :364601	NA's :353432
cl_neden	ayni_is	sekkod_d	meslek_d	isdur_d
Min. :1	Min. :1	Min. : 1	Min. :1	Min. :1
1st Qu.:2	1st Qu.:1	1st Qu.: 3	1st Qu.:5	1st Qu.:1
Median :3	Median :1	Median : 5	Median :7	Median :1
3rd Qu.:6	3rd Qu.:1	3rd Qu.: 7	3rd Qu.:9	3rd Qu.:2
Max. :7	Max. :2	Max. :14	Max. :9	Max. :5
NA's :353432	NA's :364830	NA's :362540	NA's :362446	NA's :362446
isdur_d	hhtipi	konuttip	mulkiyet	kira_sek
Min. :1	Min. :1.00	Min. :1.00	Min. :1.0	Min. :1
1st Qu.:1	1st Qu.:2.00	1st Qu.:1.00	1st Qu.:1.0	1st Qu.:2
Median :1	Median :3.00	Median :2.00	Median :1.0	Median :2
3rd Qu.:2	3rd Qu.:5.00	3rd Qu.:3.00	3rd Qu.:2.0	3rd Qu.:2
Max. :5	Max. :7.00	Max. :4.00	Max. :4.0	Max. :2
NA's :362446				NA's :301213

Table C.2 (Continued) : Descriptive Statistics of Qualitative Features.

kira_sek	yapituru	zemin_sal	zemin_oda	zemin_mut
Min. :1	Min. :1	Min. : 1	Min. : 1	Min. : 1
1st Qu.:2	1st Qu.:1	1st Qu.: 1	1st Qu.: 1	1st Qu.: 3
Median :2	Median :1	Median : 2	Median : 2	Median : 3
3rd Qu.:2	3rd Qu.:5	3rd Qu.: 6	3rd Qu.: 6	3rd Qu.: 6
Max. :2	Max. :8	Max. :98	Max. :98	Max. :98
NA's :301213	NA's :216229	NA's :181492	NA's :175659	NA's :179555
zemin_mut	zemin_ban	isit_sis	yakit_ yakit_2	
Min. : 1	Min. : 1	Min. :1	Min. :1.00	Min. :1
1st Qu.: 3	1st Qu.: 3	1st Qu.:1	1st Qu.:1.00	1st Qu.:2
Median : 3	Median : 3	Median :1	Median :1.00	Median :2
3rd Qu.: 6	3rd Qu.: 6	3rd Qu.:2	3rd Qu.:3.00	3rd Qu.:2
Max. :98	Max. :98	Max. :4	Max. :6.00	Max. :6
NA's :179555	NA's :183052	NA's :3645	NA's :31	NA's :192420
yakit_mut	yakit_su	zor_alis		
Min. :1	Min. :1	Min. :1	Min. :1	Min. :1
1st Qu.:2	1st Qu.:4	1st Qu.:3	1st Qu.:3	1st Qu.:2
Median :2	Median :5	Median :4	Median :5	Median :3
3rd Qu.:2	3rd Qu.:5	3rd Qu.:4	3rd Qu.:6	3rd Qu.:3
Max. :6	Max. :6	Max. :8	Max. :8	Max. :4
NA's :192420	NA's :359562	NA's :197970	NA's :198370	NA's :203052
zor_alis	zor_banka	zor_posta	zor_ulas	zor_saglik
Min. :1	Min. :1	Min. :1	Min. :1	Min. :1
1st Qu.:2	1st Qu.:2	1st Qu.:2	1st Qu.:2	1st Qu.:2
Median :3	Median :3	Median :3	Median :3	Median :3
3rd Qu.:3	3rd Qu.:3	3rd Qu.:3	3rd Qu.:3	3rd Qu.:3
Max. :4	Max. :4	Max. :4	Max. :4	Max. :4
NA's :203052	NA's :204211	NA's :204120	NA's :203162	NA's :203327
zor_saglik	zor_egiti	banyo	sauna	jakuzi
Min. :1	Min. :1	Min. :0.000	Min. :0	Min. :0
1st Qu.:2	1st Qu.:3	1st Qu.:1.000	1st Qu.:0	1st Qu.:0
Median :3	Median :3	Median :1.000	Median :0	Median :0
3rd Qu.:3	3rd Qu.:3	3rd Qu.:1.000	3rd Qu.:0	3rd Qu.:0
Max. :4	Max. :4	Max. :1.000	Max. :1	Max. :1
NA's :203327	NA's :202924		NA's :23996	NA's :23996

Table C.2 (Continued) : Descriptive Statistics of Qualitative Features.

jakuzi	tuvalet	mutfak	copogut	kalorif
Min. :0	Min. :0.000	Min. :0.000	Min. :0.0000	Min. :0.000
1st Qu.:0	1st Qu.:1.000	1st Qu.:1.000	1st Qu.:0.0000	1st Qu.:0.000
Median :0	Median :1.000	Median :1.000	Median :0.0000	Median :0.000
3rd Qu.:0	3rd Qu.:1.000	3rd Qu.:1.000	3rd Qu.:0.0000	3rd Qu.:1.000
Max. :1	Max. :1.000	Max. :1.000	Max. :1.0000	Max. :1.000
NA's :23996				
kalorif	tabanisi	su_sis	elektri	dog_gaz
Min. :0.000	Min. :0.0000	Min. :0.00	Min. :0	Min. :0.000
1st Qu.:0.000	1st Qu.:0.0000	1st Qu.:1.00	1st Qu.:1	1st Qu.:0.000
Median :0.000	Median :0.0000	Median :1.00	Median :1	Median :0.000
3rd Qu.:1.000	3rd Qu.:0.0000	3rd Qu.:1.00	3rd Qu.:1	3rd Qu.:0.000
Max. :1.000	Max. :1.0000	Max. :1.00	Max. :1	Max. :1.000
			NA's :112013	
dog_gaz	sicaksu	kabloyay	asansor	garaj
Min. :0.000	Min. :0.000	Min. :0.0000	Min. :0	Min. :0.0000
1st Qu.:0.000	1st Qu.:0.000	1st Qu.:0.0000	1st Qu.:0	1st Qu.:0.0000
Median :0.000	Median :1.000	Median :0.0000	Median :0	Median :0.0000
3rd Qu.:0.000	3rd Qu.:1.000	3rd Qu.:0.0000	3rd Qu.:0	3rd Qu.:0.0000
Max. :1.000	Max. :1.000	Max. :1.0000	Max. :1	Max. :1.0000
			NA's :23996	
garaj	havuz	jenrator	apartman	yazlik
Min. :0.0000	Min. :0	Min. :0	Min. :0.000	Min. :0.0000
1st Qu.:0.0000	1st Qu.:0	1st Qu.:0	1st Qu.:0.000	1st Qu.:0.0000
Median :0.0000	Median :0	Median :0	Median :0.000	Median :0.0000
3rd Qu.:0.0000	3rd Qu.:0	3rd Qu.:0	3rd Qu.:1.000	3rd Qu.:0.0000
Max. :1.0000	Max. :1	Max. :1	Max. :1.000	Max. :1.0000
	NA's :23996	NA's :179380		
yazlik	tarla	bag	sera	arsa
Min. :0.0000	Min. :0.000	Min. :0.0000	Min. :0.0000	Min. :0.000
1st Qu.:0.0000	1st Qu.:0.000	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.:0.000
Median :0.0000	Median :0.000	Median :0.0000	Median :0.0000	Median :0.000
3rd Qu.:0.0000	3rd Qu.:0.000	3rd Qu.:0.0000	3rd Qu.:0.0000	3rd Qu.:0.000
Max. :1.0000	Max. :1.000	Max. :1.0000	Max. :1.0000	Max. :1.000

Table C.2 (Continued) : Descriptive Statistics of Qualitative Features.

arsa	dukkan	otel	diger	kirkntkd
Min. :0.000	Min. :0.000	Min. :0.0000	Min. :0	Min. :1
1st Qu.:0.000	1st Qu.:0.000	1st Qu.:0.0000	1st Qu.:0	1st Qu.:1
Median :0.000	Median :0.000	Median :0.0000	Median :0	Median :2
3rd Qu.:0.000	3rd Qu.:0.000	3rd Qu.:0.0000	3rd Qu.:0	3rd Qu.:2
Max. :1.000	Max. :1.000	Max. :1.0000	Max. :1	Max. :2
	NA's :18		NA's :192233	NA's :91585

Table C.3 : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
year	Survey Year	NUM, 4 digits	2002-2016	1
bulten	Household ID number	NUM, 7 digits	Last two digits HBS's year + 5 digits	2
hhtipi	Household type	CAT, 1 digit	1 Nuclear household with a child2 Nuclear household with two children3 Nuclear household with three or more children4 Couple without children5 Patriarchal or extensive household6 One adult household7 Persons who live together	3
konuttip	Dwelling type	CAT, 1 digit	1. Detached house – Independent2. Detached house – Attached to a single wall3. Apartment house4. Other	4
mulkiyet	Ownership status of the dwelling	CAT, 1 digit	1 Owner2 Tenant3 Lodging4 Other	5
kira_ay	Monthly rent (TL)	NUM,	0 - 999999999999	6
kira_sek	Type of renting of the dwelling	CAT, 1 digit	1 With furnishings2 Without furnishings	7
loj_fert	Member order number of the individual benefiting from the lodging	Numeric, 2 digits	0 - 99	8
izafi_k	Imputed rent	NUM,	0 - 999999999999	9
konutsah	Year when ownership of the dwelling was acquired	NUM, 4 digits	1900-2016	10
kon_sure	Duration of residence in dwelling living	NUM, 2 digit	0-99	11
kon_borc	Whether any payment for the dwelling was made or not	NUM, 1 digit	1 Yes2 No	12
kondeger	Market value of the dwelling	NUM,	0 - 999999999999	13
otur_sur	Duration of residence	NUM, 2 digits	0-99	14
bina_tar	Construction date of the building	ORD, 1 digit	1 1945 -2 1946 - 19603 1961 - 19804 1981 - 19955 1996 +	15
yapituru	Building raw material	CAT, 1 digit	1 Reinforced concrete2 Steel3 Wood4 Briquette5 Stone6 Bricks7 Adobe8 Other	16

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
oda_say	Number of rooms	NUM, 2 digits	1 – 99	17
kon_alan	Size of the dwelling	NUM, 3 digits	25 – 999	18
zemin_sal	Construction type of the ground of lounge in dwelling	CAT, 1 digit	1 Parquet2 Wood3 Floor tile (ceramic)4 Vinyl floor covering (pvc)5 Carpet floor covering6 Coarse concrete7 Mosaic8 Marble9 Other	19
zemin_oda	Construction type of the ground of rooms in dwelling	CAT, 1 digit	1 Parquet2 Wood3 Floor tile (ceramic)4 Vinyl floor covering (pvc)5 Carpet floor covering6 Coarse concrete7 Mosaic8 Marble9 Other	20
zemin_mut	Construction type of the ground of kitchen in dwelling	CAT, 1 digit	1 Parquet2 Wood3 Floor tile (ceramic)4 Vinyl floor covering (pvc)5 Carpet floor covering6 Coarse concrete7 Mosaic8 Marble9 Other	21
zemin_ban	Construction type of the ground of bath in dwelling	CAT, 1 digit	1 Parquet2 Wood3 Floor tile (ceramic)4 Vinyl floor covering (pvc)5 Carpet floor covering6 Coarse concrete7 Mosaic8 Marble9 Other	22
isit_sis	Heating system of the dwelling	CAT, 1 digit	1 Stove2 Common or central heating3 Central heating boiler/Wall hung gas boiler4 Other	23
yakit_1	Fuel type	CAT, 1 digit	1 Wood2 Coal3 Natural gas4 Electric5 Dried cow dung6 Other	24
yakit_2	Fuel type	CAT, 1 digit	1 Wood2 Coal3 Natural gas4 Electric5 Dried cow dung6 Other	25
yakit_3	Fuel type	CAT, 1 digit	1 Wood2 Coal3 Natural gas4 Electric5 Dried cow dung6 Other	26
yakit_mut	Type of fuel used mostly in your dwelling for cooking	CAT, 1 digit	1 Wood2 Coal3 Natural Gas4 LPG5 Electric6 Dried cow dung7 Solar energy system8 Other	27
yakit_su	Type of fuel used mostly in your dwelling for hot water	CAT, 1 digit	1 Wood2 Coal3 Natural Gas4 LPG5 Electric6 Dried cow dung7 Solar energy system8 Other	28

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
zor_alis	Accessing to the services related to "daily shopping" considering the locationof dwelling	CAT, 1 digit	1 With a great difficulty2 With difficulty3 Easily4 Very easily	29
zor_banka	Accessing to the services related to "banking" considering the location of	CAT, 1 digit	1 With a great difficulty2 With difficulty3 Easily4 Very easily	30
zor_posta	Accessing to the services related to "post" considering the location ofdwelling	CAT, 1 digit	1 With a great difficulty2 With difficulty3 Easily4 Very easily	31
zor_ulas	Accessing to the services related to "public transportation" considering thelocation of dwelling	CAT, 1 digit	1 With a great difficulty2 With difficulty3 Easily4 Very easily	32
zor_saglik	Accessing to the services related to "health center" considering thelocation of dwelling	CAT, 1 digit	1 With a great difficulty2 With difficulty3 Easily4 Very easily	33
zor_egiti	Accessing to the services related to "compulsory education" considering thelocation of dwelling	CAT, 1 digit	1 With a great difficulty2 With difficulty3 Easily4 Very easily	34
oda_say	The number of rooms in house	NUM, 2 digit	0-99	35
kon_alan	The number of rooms in house	NUM, 2 digit	0-99	36
konutiki	Whether household has secondary residence	NUM, 1 digit	1 Yes2 No	37
konut2_n	Location of second residence	NUM, 1 digit	1 Abroad2 Another province in Turkey3 The same province	38
banyo	Bathroom availability	CAT, 1 digit	0 Not available1 Available	39
sauna	Sauna availability	CAT, 1 digit	0 Not available1 Available	40
jakuzi	Jacuzzi availability	CAT, 1 digit	0 Not available1 Available	41
tuvalet	Toilet availability	CAT, 1 digit	0 Not available1 Available	42

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
mutfak	Kitchen availability	CAT, 1 digit	0 Not available1 Available	43
copogut	Waste disposer availability	CAT, 1 digit	0 Not available1 Available	44
kalorif	Central heating availability	CAT, 1 digit	0 Not available1 Available	45
tabanisi	Floor heating availability	CAT, 1 digit	0 Not available1 Available	46
sudepo	Do you have water tank?	CAT, 1 digit	0 Not available1 Available	47
su_sis	Piped water system availability	CAT, 1 digit	0 Not available1 Available	48
elektri	Electricity availability	CAT, 1 digit	0 Not available1 Available	49
dog_gaz	Natural gas availability	CAT, 1 digit	0 Not available1 Available	50
sicaksu	Hot water availability	CAT, 1 digit	0 Not available1 Available	51
kabloyay	Cable broadcasting (TV) availability	CAT, 1 digit	0 Not available1 Available	52
asansor	Elevator availability	CAT, 1 digit	0 Not available1 Available	53
garaj	Garage availability	CAT, 1 digit	0 Not available1 Available	54
havuz	Swimming pool availability	CAT, 1 digit	0 Not available1 Available	55
jenrator	Generator (independent or collective) availability	CAT, 1 digit	0 Not available1 Available	56
telefon	Number of telephone subscription	CAT, 1 digit	0 Not available1 Available	57
ceptel	Number of mobile phones	NUM, 1 digit	0 Not availableIf any, quantity 1-9	58
bilgisay	Number of computers	NUM, 1 digit	0 Not availableIf any, quantity 1-9	59
intern	Number of internet service subscription	NUM, 1 digit	0 Not availableIf any, quantity 1-9	60
televiz	Number of LCD or plasma televisions	NUM, 1 digit	0 Not availableIf any, quantity 1-9	61
video		NUM, 1 digit	0 Not availableIf any, quantity 1-9	62
lcdtele	Number of LCD or plasma televisions	NUM, 1 digit	0 Not availableIf any, quantity 1-9	63
dvd_vcd		NUM, 1 digit	0 Not availableIf any, quantity 1-9	64
kamera	Number of camera	NUM, 1 digit	0 Not availableIf any, quantity 1-9	65
uyduant	Number of satellite antenna	NUM, 1 digit	0 Not availableIf any, quantity 1-9	66
buzdolab	Number of refrigerators	NUM, 1 digit	0 Not availableIf any, quantity 1-9	67

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
derindon	Number of freezers	NUM, 1 digit	0 Not availableIf any, quantity 1-9	68
bulasik	Number of diswaher	NUM, 1 digit	0 Not availableIf any, quantity 1-9	69
m_firin	Number of microwave stove	NUM, 1 digit	0 Not availableIf any, quantity 1-9	70
camasir	Number of washing machine	NUM, 1 digit	0 Not availableIf any, quantity 1-9	71
kurutma	Number of dryers	NUM, 1 digit	0 Not availableIf any, quantity 1-9	72
haliyika	Number of carpet cleaner	NUM, 1 digit	0 Not availableIf any, quantity 1-9	73
klima	Number of air-conditioner	NUM, 1 digit	0 Not availableIf any, quantity 1-9	74
otomobil	Number of game console	NUM, 1 digit	0 Not availableIf any, quantity 1-9	75
otoadet	Number of automobiles	NUM, 1 digit	0-9	76
isveroto	Do you have free automobiles provided by the employer?	CAT, 1 digit	0. No1. Yes	77
isotoadt	Number of the free automobiles provided by the employer	NUM, 1 digit	0-9	78
motosik	Do you have motorcylcle?	CAT, 1 digit	0. No1. Yes	79
motoadet	Number of motorcycle	NUM, 1 digit	0-9	80
deniztas	Do you have seecraft?	CAT, 1 digit	0. No1. Yes	81
dentadet	Number of seacraft	NUM, 1 digit	0-9	82
muskonut	Detached dwelling ownership	NUM, 1 digit	0 Not available1 Available	83
musmik	Number of detached dwelling	NUM, 2 digits	0-99	84
muskira	Number of the detached dwelling rented out	NUM, 2 digits	0-99	85
muskdeg	Total monthly rental income of the detached dwelling	NUM,	0 - 999999999999	86
apartman	Whether or not the household has a flat	CAT, 1 digit	0 Not available1 Available	87
aptmik	Number of flat	NUM, 2 digits	0-99	88
aptkira	Number of flats rented out	NUM, 2 digits	0-99	89

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
aptkdeg	Total monthly rental income of the flat rented out	NUM,	0 - 999999999999	90
yazlik	Whether or not the household has a summer house	CAT, 1 digit	0 Not available1 Available	91
yazmik	Number of summer house	NUM, 2 digits	0-99	92
yazkira	Number of summer house rented out	NUM, 2 digits	0-99	93
yazkdeg	Total monthly rental income of the summer house rented out	NUM,	0 - 999999999999	94
tarla	Whether or not the household has a field	CAT, 1 digit	0 Not available1 Available	95
tarmik	Area of field	NUM, 4 digits	0-9999	96
tarkira	Area of field rented out	NUM, 4 digits	0-9999	97
tarkdeg	Total monthly rental income of field rented out	NUM,	0 - 999999999999	98
bag	Whether or not the household has a plantation	CAT, 1 digit	0 Not available1 Available	99
bagmik	Area of plantation	NUM, 4 digits	0-9999	100
bagkira	Area of plantation rented out	NUM, 4 digits	0-9999	101
bagkdeg	Total monthly rental income of plantation rented out	NUM,	0 - 999999999999	102
sera	Whether or not the household has a greenhouse	CAT, 1 digit	0 Not available1 Available	103
seramik	Area of greenhouse	NUM, 4 digits	0-9999	104
serakira	Area of greenhouse rented out	NUM, 4 digits	0-9999	105
serakdeg	Total monthly rental income of greenhouse rented out	NUM,	0 - 999999999999	106
arsa	Whether or not the household has land	CAT, 1 digit	0 Not available1 Available	107

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
arsamik	Area of land	NUM, 4 digits	0-9999	108
arsakira	Area of land rented out	NUM, 4 digits	0-9999	109
arsakdeg	Total monthly rental income of land rented out	NUM,	0 - 999999999999	110
arsadeg	Total value of lands owned	NUM,	1 - 999999999999	111
dukkan	Whether or not the household has a shop	CAT, 1 digit	0 Not available1 Available	112
dukmik	Number of shop	NUM, 2 digits	0-99	113
dukkira	Number of shop rented out	NUM, 2 digits	0-99	114
dukkdeg	Total monthly rental income of the shop rented out	NUM,	0 - 999999999999	115
dukdeg	Total value of shop	NUM,	0 - 999999999999	116
otel	Whether or not the household has a hotel	CAT, 1 digit	0 Not available1 Available	117
otelmik	Number of hotel	NUM, 2 digits	0-99	118
otelkira	Number of hotel rented out	NUM, 2 digits	0-99	119
otelkdeg	Total monthly rental income of the shop rented out	NUM,	0 - 999999999999	120
oteldeg	Monetary value of Hotel	NUM,	1 - 999999999999	121
diger	Do you have any other real estate?	CAT, 1 digit	0 Not available1 Available	122
digmik	Number of other real estates	NUM,	0 - 999999999999	123
digkira	Number of real estates rented out	NUM,	0 - 999999999999	124
digkdeg	Total monthly rental income of real estates rented out	NUM,	0 - 999999999999	125
digdeg	Total value of Real Estates	NUM,	0 - 999999999999	126
yilkulgel	Annual disposable income of household	NUM,	0 - 999999999999	127
efb_oecd	E quivalent size of household by modified OECD scale	NUM, 4 digits	0 - 9999	128

Table C.3 (Continued) : Budget Survey - HH Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
harcama	Household monthly consumption expenditure (TL)	NUM,	0 - 999999999999	129
hhb	Household size	NUM, 2 digits	0-20	130
kirkntkd	Settlement type code: Rural or Urban	NUM, 1 digit	1 Rural2 Urban	131
faktor	Population weight	NUM, 10 digits	0 - 9999999999	132

Table C.4 : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
year	Survey Year	NUM, 4 digits	2002-2016	1
bulten	Household ID number	NUM, 7 digits	Last two digits HBS's year + 5 digits	2
fertno	Individual order no	NUM, 2 digits	0 - 99	3
cinsiyet	Gender	CAT, 1 digit	1. Male2. Female	4
yas	Age groups	ORD, 2 digits	1. 0-5 years2. 6-14 years3. 15-19 years4. 20-24 years5. 25-29 years6. 30-34 years7. 35-39 years8. 40-44 years9. 45-49 years10. 50-54 years11. 55-59 years12. 60-64 years13. 65+years	5
yakinlik	Relationship with the reference person of the household	CAT, 2 digits	1. Reference person2. Spouse3. Son/Doughter4. Father/Mother5. Father-in-law/Mother-in-law6. Son-in-law/Doughter-in-law7. Brother/Sister8. Grandchild9. Other relative10. Housemaid lives together with household11. Non-relative	6
ailegrup	The group of families in dwelling	NUM, 1 digit	0 - 9	7
sagliks	Whether health insurance	CAT, 1 digit	1. Yes - SGK All2. Yes - Other3. Yes - GSS (General Health Insurance, Green Health Card)4. No	8
sagliks2	DROP	DROP	DROP	9
gunengel	Whether the member has been limited in activities usually do because of a health or mental problem	CAT, 1 digit	1. Yes2. No	10

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
calengel	Whether the member has been limited in activities related to work because of a health or mental problem	CAT, 1 digit	1. Yes2. No	11
egitim	Education status	ORD, 2 digits	1. Illiterate2. Literate – not completed a school3. Primary school4. Primary education5. Secondary school6. Junior Vocational High School7. High School8. Senior Vocational High School9. 2-3 year-College10. 4-year-College or University11. Post Graduate/PhD.	12
ogrenci	Status of being taught in an education institution	CAT, 1 digit	1. Yes2. No	13
egitim_devam	Type of educational institution attended	CAT, 2 digits	0 Not completed any school1 Primary school21 Secondary school22 Vocational school at secondary school level23 Primary education31 High school32 Vocational school at high school level4 Higher educational institutions for 2-3 years5 Higher educational ins. and faculties for 4 years61 Faculties for 5 or 6 years62 Masters7 Doctorate	14
medenidr	Marital status	CAT, 1 digit	1. Never married2. Married3. Widowed4. Divorced	15
calis_ay	Employment status as of survey month	CAT, 1 digit	1. Worked2. Not worked but work connection is going on3. Not worked	16

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
is_arama	Status of looking for employment during last 4 weeks	CAT, 1 digit	1. Yes2. No	17
isneden	Reason for not seeking a job	CAT, 2 digits	1. Found a job, established his/her own business and waiting to get started2. Continuing education and training3. Busy with housework4. Retired5. Property income earners6. Elderly (not retired, however, thinking him/herself too old to work, aged 65+)7. Disabled or ill10. Seasonal worker11. Other(PS:There is no jump in order, it is encoded as above inevitably)	18
isbasi	Status of getting started to work in 15-day time	CAT, 1 digit	1. Yes2. No	19
calis_yl	Is individual worked in current year?	CAT, 1 digit	1. Yes2. No	20

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
sekkod	Main industry code in the main job	CAT, 2 digits	1. Agriculture, forestry, hunting and fishing2. Mining and quarrying3. Manufacturing4. Electricity, gas and water supply5. Construction6. Wholesale and retail trade; repair of motor vehicles, motorcycles and personel andhousehold goods7. Hotels and restaurants8. Transport, communication and storage9. Financial intermediation10. Real estate, renting and business activities11. Public administration and defence; compulsory social security, excluding armed forces12. Education13. Health and social work14. Other social, community and personal service activities	21
meslek	Code for the occupation in the main job	CAT, 1 digit	1. Managers2. Professionals3. Technicians and associate professionals4. Clerical support workers5. Service and sales workers6. Skilled agricultural, forestry and fishery workers7. Craft and related trades workers8. Plant and machine operators and assemblers9. Elementary occupations	22
sure_yil	Duration of employment in the main job (years)	NUM, 2 digits	0 - 99	23

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
calsur	Weekly time of employment in the main job (hour	NUM, 2 digits	0 - 99	24
cal_kisi	Number of employers in firm where individuals works	CAT, 1 digit	1. 1-9 kişi2. 10-24 kişi3. 25-49 kişi4. 50 +	25
kac_kisi	Number of individuals who work in the same firm	NUM, 1 digit	0 - 9	26
sgk	Whether individual has social insurance	CAT, 1 digit	1. SGK2. Other3. None	27
is_statu	Type of firm's sector	CAT, 1 digit	1. Private2. Public3. State-owned enterprise	28
huk_dur	Type of firm status	CAT, 1 digit	1. Individual ownership2. Ordinary partnership3. Unlimited partnership4. Limited partnership5. Limited company6. Incorporated company7. Cooperative8. Other	29
koopera	Is the firm worked a cooperative or Incorporated company	CAT, 1 digit	1. Yes2. No	30
sir_kon	The role of individual in the firm	CAT, 1 digit	1. Shareholder, partner2. Wage worker	31
isdur	Employment status in the main job	CAT, 2 digits	1. Regular employee (Wage earner)2. Casual employee (seasonal workers or daily work)3. Employer4. Self employed5. Unpaid family worker	32
ist_tesk	Type of employment	CAT, 1 digit	1. Permanent job2. Fixed-term employment contract (temporary)3. Casual type of work (Working without contract)	33

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
guc_kul	Body power usage in working	CAT, 1 digit	1. Physical2. Mental3. Both of them	34
sendika	Union Membership	CAT, 1 digit	1. Yes2. No	35
ucrn_ay	Monthly wage income (In cash)	NUM,	0 - 999999999999	36
ucra_ay	Monthly wage income (In-kind)	NUM,	0 - 999999999999	37
ucrn_yil	Annual wage income (In cash)	NUM,	0 - 999999999999	38
ikr_yil	Annual bounty	NUM,	0 - 999999999999	39
prim_yil	Annual bonus income, premium, tips etc.	NUM,	0 - 999999999999	40
ucra_yil	Annual total wage income (In-kind)	NUM,	0 - 999999999999	41
konayni	Annual income in-kind earned in the form of house/lodging	NUM,	0 - 999999999999	42
ulsayni	Annual income in-kind earned in the form of transportation	NUM,	0 - 999999999999	43
egsayni	Annual income in-kind earned in the form of discounts for electricity, gas,water, telephone etc.	NUM,	0 - 999999999999	44
yemayni	Annual income in-kind earned in the form of meals	NUM,	0 - 999999999999	45
kresayni	Annual income in-kind earned in the form of day nursary payment	NUM,	0 - 999999999999	46
giyayni	Annual income in-kind earned in the form of clothing	NUM,	0 - 999999999999	47

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
gidaayni	Annual income in-kind earned in the form of food	NUM,	0 - 999999999999	48
digayni	Other annual incomes in-kind as part of wage and salary	NUM,	0 - 999999999999	49
isfaal	Main activity of the establishment for entrepreneurs	CAT, 1 digit	1. Agriculture2. Non-agriculture	50
mulk_dur	Ownership status of entrepreneur	CAT, 1 digit	1. Rented2. Owner3. Other	51
mutsn_ay	Annual entrepreneurial income in cash	NUM,	0 - 999999999999	52
mutsa_ay	Monthly entrepreneurial income in cash	NUM,	0 - 999999999999	53
mutsn_yl	Annual entrepreneurial income in kind	NUM,	0 - 999999999999	54
mutsa_yl	Monthly entrepreneurial income in kind	NUM,	0 - 999999999999	55
targ_ay	Monthly net agricultural income	NUM,	0 - 999999999999	56
targ_yl	Annual net agricultural income	NUM,	0 - 999999999999	57
ekis_yl	Additional job in current year	CAT, 1 digit	1. Yes2. No	58
ekis_ay	Additional job in current year	CAT, 1 digit	1. Yes2. No	59
calsur_e	Additional working hours	NUM, 2 digits	0 - 99	60

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
sekkod_e	Main industry code in the additional job	CAT, 2 digits	1. Agriculture, forestry, hunting and fishing2. Mining and quarrying3. Manufacturing4. Electricity, gas and water supply5. Construction6. Wholesale and retail trade; repair of motor vehicles, motorcycles and personel andhousehold goods7. Hotels and restaurants8. Transport, communication and storage9. Financial intermediation10. Real estate, renting and business activities11. Public administration and defence; compulsory social security, excluding armed forces12. Education13. Health and social work14. Other social, community and personal service activities	61
meslek_e	Code for the occupation in the additional job	CAT, 1 digit	1. Managers2. Professionals3. Technicians and associate professionals4. Clerical support workers5. Service and sales workers6. Skilled agricultural, forestry and fishery workers7. Craft and related trades workers8. Plant and machine operators and assemblers9. Elementary occupations	62
isdur_e	Employment status in the additional job	CAT, 1 digit	1. Regular employee (Wage earner)2. Casual employee (seasonal workers or daily work)3. Employer4. Self employed5. Unpaid family worker	63

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
ekgln_ay	Monthly in cash income from additional job	NUM,	0 - 999999999999	64
ekgla_ay	monthly in kind income from additional job	NUM,	0 - 999999999999	65
ekgln_yl	Annual in cash income from additional job	NUM,	0 - 999999999999	66
ekgla_yl	Annual in kind income from additional job	NUM,	0 - 999999999999	67
calay_yil	Duration of employment in the additional job in current year (months)	NUM, 2 digits	0 - 12	68
cl_neden	The reason for not working	CAT, 1 digit	1. Seasonal job2. Change the job3. Quit the job4. Fired from the job5. I work if I find a job6. I just started to work7. Other	69
ayni_is	In-kind job	CAT, 1 digit	1. Yes2. No	70
calsur_d	Work hours in previous job	NUM, 2 digits	Jan-99	71

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
sekkod_d	Main industry code in previous job	CAT, 2 digits	1. Agriculture, forestry, hunting and fishing2. Mining and quarrying3. Manufacturing4. Electricity, gas and water supply5. Construction6. Wholesale and retail trade; repair of motor vehicles, motorcycles and personel andhousehold goods7. Hotels and restaurants8. Transport, communication and storage9. Financial intermediation10. Real estate, renting and business activities11. Public administration and defence; compulsory social security, excluding armed forces12. Education13. Health and social work14. Other social, community and personal service activities	72
meslek_d	Code for the occupation in previous job	CAT, 1 digit	1. Managers2. Professionals3. Technicians and associate professionals4. Clerical support workers5. Service and sales workers6. Skilled agricultural, forestry and fishery workers7. Craft and related trades workers8. Plant and machine operators and assemblers9. Elementary occupations	73
isdur_d	Employment status in previous job	CAT, 1 digit	1. Regular employee (Wage earner)2. Casual employee (seasonal workers or daily work)3. Employer4. Self employed5. Unpaid family worker	74

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
digln_yl	Annual in cash income from previous jobs	NUM,	0 - 999999999999	75
digla_yl	Annual in-kind income from previous jobs	NUM,	0 - 999999999999	76
gmkn_ay	Monthly real estate (rental) income in cash	NUM,	0 - 999999999999	77
gmka_ay	Monthly real estate (rental) income in kind	NUM,	0 - 999999999999	78
gmkn_yl	Annual real estate (rental) income in cash	NUM,	1 - 999999999999	79
gmka_yl	Annual real estate (rental) income in kind	NUM,	0 - 999999999999	80
mulkn_ay	Monthly in cash income from property	NUM,	0 - 999999999999	81
mulka_ay	Monthly in-kind income from property	NUM,	0 - 999999999999	82
banka_yl	Annual interest on bank deposits	NUM,	0 - 999999999999	83
doviz_yl	Annual income from exchange	NUM,	0 - 999999999999	84
nema_yl	Annual income from interest	NUM,	0 - 999999999999	85
temet_yl	Annual income from dividend	NUM,	0 - 999999999999	86
kar_yl	Annual income from share	NUM,	0 - 999999999999	87
toprn_yl	Annual in cash income from land	NUM,	0 - 999999999999	88
topra_yl	Annual in-kind income from land	NUM,	0 - 999999999999	89
trsn_ay	Annual in cash income from transfers	NUM,	0 - 999999999999	90
trsa_ay	Annual in-kind income from transfers	NUM,	0 - 999999999999	91

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
emekl_yl	Annual income from pension	NUM,	0 - 999999999999	92
viade_yl	Annual income from tax repayment	NUM,	0 - 999999999999	93
yasli_yl	Annual elderly pay (in cash)	NUM,	0 - 999999999999	94
sosy_yl	Benefits from social assistance fund and other family allowances (in cash)	NUM,	0 - 999999999999	95
hasta_yl	Annual sickness benefit (cash)	NUM,	0 - 999999999999	96
dul_yl	Annual widow, orphan pension (cash)	NUM,	0 - 999999999999	97
gazi_yl	Annual veteran pension and disability pay and sickness benefits (cash)	NUM,	0 - 999999999999	98
burs_yl	Annual scholarship income (in cash)	NUM,	0 - 999999999999	99
issiz_yl	Annual unemployment pay (in cash)	NUM,	0 - 999999999999	100
ddestek_yl	Annual direct income support and fuel and milk support paid to farmers in cash	NUM,	0 - 999999999999	101
dayni_yl	Annual in-kind income received from State	NUM,	0 - 999999999999	102
ydemk_yl	Annual pension from abroad (cash)	NUM,	0 - 999999999999	103
yddav_yl	Annual income as foreign currency, benefit or scholarship etc. from abroad(in cash)	NUM,	0 - 999999999999	104
ydayn_yl	Annual in-kind income from abroad	NUM,	0 - 999999999999	105

Table C.4 (Continued) : Budget Survey - Individual Questionnaire.

VARIABLE	EXPLANATION	TYPE	VALUES	No
dgnfk_yl	Annual other transfer income like alimony, allowance, scholarship, almsetc. from persons or other private institutions (in cash)	NUM,	0 - 999999999999	106
dgbrs_yl	Annual in cash income from scholarships	NUM,	0 - 999999999999	107
dsemk_yl	Annual other transfers in cash from private pension system (bank,foundations etc.)	NUM,	0 - 999999999999	108
dgayn_yl	Annual other transfers in-kind from persons or other private institutions	NUM,	0 - 999999999999	109
topgn_yl	Total annual individual income in cash	NUM,	0 - 999999999999	110
topga_ay	Total monthly in cash income	NUM,	0 - 999999999999	111
topgelay	Total monthly income	NUM,	0 - 999999999999	112
topgna_yl	Total annual in-kind income	NUM,	0 - 999999999999	113
topga_yl	Total annual individual income in kind	NUM,	0 - 999999999999	114
topgelyl	Total annual individual income	NUM,	0 - 999999999999	115
indeks_arl	Coefficient for inflating the annual income in respect with year 2012	NUM, 5 digits	0..99,999	116

*NUM: Numerical, CAT: Categorical, ORD: Ordinal.

Table C.5 : Quantile Regression Results - 10th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.04005
Model:	QuantReg	**Bandwidth:**	0.01004
Method:	Least Squares	**Sparsity:**	0.4567
Date:	Cts, 23 Haz 2018	**No. Observations:**	371613
Time:	19:01:46	**Df Residuals:**	371226

Quantile Regression Coefficients for 10th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	-0.0453	0.021	-2.128	0.033	-0.087	-0.004
year[T.2003]	0.0113	0.001	7.920	0.000	0.008	0.014
year[T.2004]	0.0077	0.002	4.101	0.000	0.004	0.011
year[T.2005]	-0.0009	0.002	-0.507	0.612	-0.004	0.002
year[T.2006]	0.0022	0.002	1.330	0.184	-0.001	0.005
year[T.2007]	-0.0006	0.002	-0.357	0.721	-0.004	0.003
year[T.2008]	0.0049	0.002	3.028	0.002	0.002	0.008
year[T.2009]	0.0123	0.001	10.051	0.000	0.010	0.015
year[T.2010]	0.0101	0.001	8.390	0.000	0.008	0.012
year[T.2011]	0.0081	0.001	6.864	0.000	0.006	0.010
year[T.2012]	-0.0141	0.001	-12.857	0.000	-0.016	-0.012
year[T.2013]	-0.0046	0.001	-3.931	0.000	-0.007	-0.002
year[T.2014]	-0.0069	0.001	-5.888	0.000	-0.009	-0.005
year[T.2015]	-0.0118	0.002	-6.706	0.000	-0.015	-0.008
year[T.2016]	0.0016	0.002	0.894	0.371	-0.002	0.005
cinsiyet[T.2]	0.0009	0.001	1.345	0.178	-0.000	0.002
yakinlik[T.1.0]	0.0059	0.001	5.405	0.000	0.004	0.008
yakinlik[T.2.0]	0.0041	0.001	2.982	0.003	0.001	0.007
yakinlik[T.3.0]	0.0092	0.002	4.843	0.000	0.005	0.013
yakinlik[T.4.0]	0.0012	0.003	0.410	0.682	-0.004	0.007
yakinlik[T.5.0]	-0.0029	0.004	-0.724	0.469	-0.011	0.005
yakinlik[T.6.0]	0.0078	0.002	3.794	0.000	0.004	0.012
yakinlik[T.7.0]	0.0057	0.002	2.910	0.004	0.002	0.010
yakinlik[T.8.0]	0.0082	0.003	3.278	0.001	0.003	0.013
yakinlik[T.9.0]	-0.0004	0.004	-0.089	0.929	-0.009	0.008
yakinlik[T.10.0]	-0.0070	0.008	-0.845	0.398	-0.023	0.009
sagliks[T.2.0]	0.0102	0.002	5.475	0.000	0.007	0.014
sagliks[T.3.0]	-0.0072	0.001	-7.703	0.000	-0.009	-0.005
sagliks[T.4.0]	-0.0071	0.001	-9.704	0.000	-0.008	-0.006
ogrenci[T.2.0]	0.0032	0.001	3.208	0.001	0.001	0.005
egitim_devam[T.2]	-0.0010	0.002	-0.473	0.636	-0.005	0.003
egitim_devam[T.3]	0.0011	0.003	0.379	0.705	-0.005	0.007
egitim_devam[T.4]	-0.0014	0.004	-0.312	0.755	-0.010	0.007
egitim_devam[T.5]	-0.0087	0.003	-2.533	0.011	-0.015	-0.002
egitim_devam[T.6]	-0.0021	0.006	-0.354	0.724	-0.014	0.010
egitim_devam[T.7]	-1.143e-05	0.020	-0.001	1.000	-0.038	0.038
egitim_devam[T.8]	2.798e-05	0.030	0.001	0.999	-0.058	0.058

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-4.938e-05	0.000	-0.256	0.798	-0.000	0.000
egitim_devam[T.999]	-0.0034	0.001	-2.335	0.020	-0.006	-0.001
medenidr[T.2.0]	-0.0043	0.001	-3.215	0.001	-0.007	-0.002
medenidr[T.3.0]	-0.0028	0.002	-1.343	0.179	-0.007	0.001
medenidr[T.4.0]	-0.0056	0.002	-2.463	0.014	-0.010	-0.001
is_arama[T.2]	1.265e-05	0.002	0.008	0.994	-0.003	0.003
is_arama[T.999]	0.0070	0.002	3.338	0.001	0.003	0.011
isneden[T.2.0]	0.0007	0.010	0.071	0.944	-0.019	0.020
isneden[T.3.0]	0.0003	0.010	0.034	0.973	-0.019	0.020
isneden[T.4.0]	0.0021	0.010	0.210	0.833	-0.017	0.021
isneden[T.5.0]	0.0018	0.015	0.119	0.906	-0.028	0.031
isneden[T.6.0]	-0.0012	0.010	-0.125	0.901	-0.021	0.018
isneden[T.7.0]	0.0027	0.010	0.272	0.786	-0.017	0.022
isneden[T.10.0]	-0.0047	0.011	-0.447	0.655	-0.025	0.016
isneden[T.11.0]	-0.0006	0.010	-0.061	0.952	-0.020	0.019
sekkod[T.2.0]	-0.0072	0.005	-1.479	0.139	-0.017	0.002
sekkod[T.3.0]	-0.0031	0.002	-1.492	0.136	-0.007	0.001
sekkod[T.4.0]	0.0041	0.004	1.124	0.261	-0.003	0.011
sekkod[T.5.0]	-0.0040	0.002	-1.668	0.095	-0.009	0.001
sekkod[T.6.0]	0.0037	0.002	1.809	0.070	-0.000	0.008
sekkod[T.7.0]	-0.0021	0.003	-0.811	0.417	-0.007	0.003
sekkod[T.8.0]	0.0030	0.002	1.197	0.231	-0.002	0.008
sekkod[T.9.0]	0.0005	0.004	0.121	0.904	-0.008	0.009
sekkod[T.10.0]	0.0007	0.004	0.173	0.862	-0.007	0.008
sekkod[T.11.0]	0.0069	0.003	2.700	0.007	0.002	0.012
sekkod[T.12.0]	0.0034	0.003	1.213	0.225	-0.002	0.009
sekkod[T.13.0]	0.0012	0.003	0.409	0.682	-0.004	0.007
sekkod[T.14.0]	-0.0027	0.003	-1.071	0.284	-0.008	0.002
sgk[T.2.0]	0.0004	0.008	0.048	0.961	-0.015	0.015
sgk[T.3.0]	-0.0055	0.001	-4.068	0.000	-0.008	-0.003
is_statu[T.2]	-0.0056	0.002	-2.306	0.021	-0.010	-0.001
is_statu[T.3]	0.0133	0.005	2.445	0.014	0.003	0.024
is_statu[T.999]	-0.0017	0.002	-0.815	0.415	-0.006	0.002
koopera[T.2]	0.0048	0.003	1.445	0.149	-0.002	0.011
koopera[T.999]	0.0026	0.003	0.862	0.389	-0.003	0.009
sir_kon[T.2]	-0.0006	0.005	-0.115	0.908	-0.011	0.010
sir_kon[T.999]	0.0006	0.005	0.119	0.905	-0.009	0.010
isdur[T.2.0]	0.0068	0.005	1.247	0.212	-0.004	0.017
isdur[T.3.0]	0.0135	0.003	4.587	0.000	0.008	0.019
isdur[T.4.0]	0.0147	0.002	6.639	0.000	0.010	0.019
isdur[T.5.0]	-0.0034	0.002	-1.412	0.158	-0.008	0.001
ist_tesk[T.2.0]	-0.0058	0.005	-1.108	0.268	-0.016	0.004
ist_tesk[T.3.0]	0.0028	0.006	0.487	0.626	-0.008	0.014
guc_kul[T.2]	-0.0002	0.002	-0.083	0.934	-0.004	0.004
guc_kul[T.3]	0.0054	0.002	3.412	0.001	0.002	0.009

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.999]	0.0120	0.002	6.451	0.000	0.008	0.016
isfaal[T.2.0]	0.0136	0.002	5.811	0.000	0.009	0.018
mulk_dur[T.2.0]	-0.0061	0.003	-2.432	0.015	-0.011	-0.001
mulk_dur[T.3.0]	-0.0174	0.003	-5.219	0.000	-0.024	-0.011
ekis_yl[T.2.0]	-0.0039	0.003	-1.123	0.262	-0.011	0.003
ekis_ay[T.2.0]	0.0038	0.004	0.935	0.350	-0.004	0.012
sekkod_e[T.2.0]	-2.715e-06	0.013	-0.000	1.000	-0.026	0.026
sekkod_e[T.3.0]	-0.0027	0.010	-0.264	0.792	-0.023	0.018
sekkod_e[T.4.0]	-0.0002	0.031	-0.008	0.994	-0.060	0.060
sekkod_e[T.5.0]	0.0044	0.008	0.524	0.600	-0.012	0.021
sekkod_e[T.6.0]	0.0040	0.008	0.477	0.633	-0.012	0.021
sekkod_e[T.7.0]	-0.0049	0.011	-0.432	0.666	-0.027	0.018
sekkod_e[T.8.0]	-0.0023	0.011	-0.203	0.839	-0.024	0.020
sekkod_e[T.9.0]	0.0003	0.026	0.010	0.992	-0.050	0.051
sekkod_e[T.10.0]	3.783e-05	0.019	0.002	0.998	-0.037	0.037
sekkod_e[T.11.0]	0.0004	0.012	0.032	0.975	-0.023	0.024
sekkod_e[T.12.0]	-0.0014	0.015	-0.097	0.923	-0.030	0.027
sekkod_e[T.13.0]	-0.0001	0.018	-0.006	0.995	-0.036	0.036
sekkod_e[T.14.0]	-0.0006	0.010	-0.062	0.950	-0.019	0.018
meslek_e[T.1.0]	0.0096	0.007	1.432	0.152	-0.004	0.023
meslek_e[T.2.0]	-0.0019	0.009	-0.202	0.840	-0.021	0.017
meslek_e[T.3.0]	0.0009	0.010	0.087	0.931	-0.019	0.021
meslek_e[T.4.0]	-0.0002	0.023	-0.007	0.994	-0.045	0.045
meslek_e[T.5.0]	-0.0005	0.009	-0.056	0.955	-0.017	0.016
meslek_e[T.6.0]	0.0056	0.008	0.699	0.484	-0.010	0.021
meslek_e[T.7.0]	-0.0011	0.007	-0.155	0.877	-0.015	0.013
meslek_e[T.8.0]	-0.0114	0.008	-1.419	0.156	-0.027	0.004
meslek_e[T.9.0]	0.0009	0.006	0.142	0.887	-0.011	0.013
isdur_e[T.2.0]	-0.0051	0.007	-0.755	0.451	-0.018	0.008
isdur_e[T.3.0]	0.0061	0.010	0.605	0.545	-0.014	0.026
isdur_e[T.4.0]	-0.0009	0.007	-0.141	0.888	-0.014	0.012
isdur_e[T.5.0]	0.0081	0.009	0.951	0.342	-0.009	0.025
cl_neden[T.2.0]	0.0047	0.005	0.896	0.370	-0.006	0.015
cl_neden[T.3.0]	-0.0028	0.003	-0.959	0.338	-0.008	0.003
cl_neden[T.4.0]	0.0046	0.005	0.862	0.389	-0.006	0.015
cl_neden[T.5.0]	0.0042	0.006	0.735	0.462	-0.007	0.015
cl_neden[T.6.0]	0.0057	0.004	1.403	0.161	-0.002	0.014
cl_neden[T.7.0]	-0.0067	0.005	-1.486	0.137	-0.016	0.002
ayni_is[T.2.0]	-0.0111	0.005	-2.152	0.031	-0.021	-0.001
sekkod_d[T.2.0]	0.0001	0.022	0.005	0.996	-0.043	0.043
sekkod_d[T.3.0]	0.0096	0.007	1.319	0.187	-0.005	0.024
sekkod_d[T.4.0]	-0.0008	0.012	-0.065	0.948	-0.024	0.023
sekkod_d[T.5.0]	-0.0048	0.007	-0.721	0.471	-0.018	0.008
sekkod_d[T.6.0]	0.0040	0.008	0.521	0.603	-0.011	0.019
sekkod_d[T.7.0]	-0.0033	0.009	-0.366	0.714	-0.021	0.014

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.8.0]	0.0031	0.008	0.373	0.709	-0.013	0.019
sekkod_d[T.9.0]	0.0003	0.018	0.016	0.987	-0.034	0.035
sekkod_d[T.10.0]	-0.0003	0.013	-0.023	0.981	-0.025	0.025
sekkod_d[T.11.0]	0.0002	0.009	0.021	0.984	-0.018	0.018
sekkod_d[T.12.0]	-0.0006	0.013	-0.042	0.966	-0.027	0.026
sekkod_d[T.13.0]	-0.0002	0.015	-0.012	0.990	-0.029	0.029
sekkod_d[T.14.0]	0.0028	0.009	0.325	0.745	-0.014	0.020
meslek_d[T.2.0]	0.0019	0.013	0.150	0.881	-0.023	0.027
meslek_d[T.3.0]	0.0031	0.010	0.308	0.758	-0.016	0.023
meslek_d[T.4.0]	0.0018	0.011	0.175	0.861	-0.019	0.022
meslek_d[T.5.0]	0.0005	0.009	0.059	0.953	-0.017	0.018
meslek_d[T.6.0]	-0.0053	0.011	-0.483	0.629	-0.027	0.016
meslek_d[T.7.0]	-0.0021	0.009	-0.238	0.812	-0.020	0.015
meslek_d[T.8.0]	-0.0015	0.010	-0.155	0.877	-0.020	0.017
meslek_d[T.9.0]	0.0017	0.009	0.184	0.854	-0.016	0.020
isdur_d[T.2.0]	-0.0043	0.004	-1.055	0.292	-0.012	0.004
isdur_d[T.3.0]	0.0006	0.013	0.047	0.963	-0.025	0.026
isdur_d[T.4.0]	0.0014	0.007	0.202	0.840	-0.012	0.015
isdur_d[T.5.0]	-0.0051	0.008	-0.672	0.501	-0.020	0.010
hhtipi[T.2.0]	-0.0016	0.001	-2.053	0.040	-0.003	-7.23e-05
hhtipi[T.3.0]	-0.0005	0.001	-0.526	0.599	-0.002	0.001
hhtipi[T.4.0]	0.0045	0.001	4.144	0.000	0.002	0.007
hhtipi[T.5.0]	0.0117	0.001	12.189	0.000	0.010	0.014
hhtipi[T.6.0]	0.0040	0.001	3.008	0.003	0.001	0.007
hhtipi[T.7.0]	-9.079e-05	0.003	-0.032	0.974	-0.006	0.005
konuttip[T.2.0]	0.0007	0.001	0.641	0.521	-0.001	0.003
konuttip[T.3.0]	0.0037	0.001	4.168	0.000	0.002	0.005
konuttip[T.4.0]	-5.939e-05	0.007	-0.008	0.993	-0.014	0.014
mulkiyet[T.2.0]	-0.0024	0.001	-2.404	0.016	-0.004	-0.000
mulkiyet[T.3.0]	0.0057	0.002	2.660	0.008	0.002	0.010
mulkiyet[T.4.0]	-0.0171	0.001	-14.873	0.000	-0.019	-0.015
kira_sek[T.2.0]	-0.0126	0.008	-1.556	0.120	-0.028	0.003
yapituru[T.2.0]	-1.808e-05	0.019	-0.001	0.999	-0.037	0.037
yapituru[T.3.0]	0.0035	0.002	1.475	0.140	-0.001	0.008
yapituru[T.4.0]	0.0006	0.001	0.449	0.654	-0.002	0.003
yapituru[T.5.0]	-0.0084	0.002	-5.363	0.000	-0.011	-0.005
yapituru[T.6.0]	-0.0028	0.001	-2.445	0.014	-0.005	-0.001
yapituru[T.7.0]	0.0017	0.002	1.039	0.299	-0.002	0.005
yapituru[T.8.0]	-0.0080	0.007	-1.233	0.217	-0.021	0.005
zemin_sal[T.2.0]	0.0006	0.001	0.500	0.617	-0.002	0.003
zemin_sal[T.3.0]	-0.0014	0.001	-1.139	0.255	-0.004	0.001
zemin_sal[T.4.0]	-0.0037	0.001	-2.665	0.008	-0.006	-0.001
zemin_sal[T.5.0]	0.0026	0.003	0.926	0.354	-0.003	0.008
zemin_sal[T.6.0]	0.0033	0.001	2.268	0.023	0.000	0.006
zemin_sal[T.7.0]	0.0117	0.003	4.334	0.000	0.006	0.017

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.8.0]	-0.0013	0.008	-0.155	0.877	-0.017	0.015
zemin_sal[T.9.0]	0.0003	0.003	0.090	0.928	-0.005	0.006
zemin_sal[T.98.0]	0.0015	0.006	0.260	0.795	-0.010	0.013
zemin_mut[T.2.0]	0.0027	0.003	1.079	0.281	-0.002	0.008
zemin_mut[T.3.0]	0.0060	0.002	3.473	0.001	0.003	0.009
zemin_mut[T.4.0]	0.0099	0.003	3.131	0.002	0.004	0.016
zemin_mut[T.5.0]	-0.0068	0.006	-1.080	0.280	-0.019	0.006
zemin_mut[T.6.0]	0.0057	0.002	2.489	0.013	0.001	0.010
zemin_mut[T.7.0]	-0.0025	0.003	-0.740	0.459	-0.009	0.004
zemin_mut[T.8.0]	-0.0052	0.007	-0.751	0.452	-0.019	0.008
zemin_mut[T.9.0]	0.0046	0.004	1.169	0.242	-0.003	0.012
zemin_mut[T.98.0]	0.0007	0.006	0.120	0.905	-0.011	0.013
zemin_ban[T.2.0]	-0.0001	0.015	-0.010	0.992	-0.029	0.029
zemin_ban[T.3.0]	-0.0018	0.010	-0.172	0.863	-0.022	0.019
zemin_ban[T.4.0]	-0.0011	0.013	-0.090	0.928	-0.026	0.024
zemin_ban[T.5.0]	-9.517e-06	0.017	-0.001	1.000	-0.033	0.033
zemin_ban[T.6.0]	0.0011	0.010	0.110	0.913	-0.019	0.022
zemin_ban[T.7.0]	0.0021	0.011	0.199	0.843	-0.019	0.023
zemin_ban[T.8.0]	-0.0025	0.012	-0.204	0.839	-0.026	0.021
zemin_ban[T.9.0]	0.0042	0.011	0.371	0.710	-0.018	0.026
zemin_ban[T.98.0]	-1.438e-05	0.016	-0.001	0.999	-0.031	0.031
isit_sis[T.2.0]	-0.0052	0.001	-4.607	0.000	-0.007	-0.003
isit_sis[T.3.0]	-0.0046	0.001	-3.743	0.000	-0.007	-0.002
isit_sis[T.4.0]	0.0040	0.002	1.790	0.073	-0.000	0.008
yakit_1[T.2.0]	0.0038	0.001	4.128	0.000	0.002	0.006
yakit_1[T.3.0]	0.0017	0.001	1.309	0.191	-0.001	0.004
yakit_1[T.4.0]	-0.0021	0.001	-1.469	0.142	-0.005	0.001
yakit_1[T.5.0]	0.0023	0.002	1.363	0.173	-0.001	0.006
yakit_1[T.6.0]	-0.0013	0.002	-0.856	0.392	-0.004	0.002
yakit_2[T.2.0]	0.0049	0.002	3.133	0.002	0.002	0.008
yakit_2[T.3.0]	-0.0006	0.009	-0.067	0.946	-0.019	0.017
yakit_2[T.4.0]	-0.0009	0.003	-0.286	0.775	-0.007	0.005
yakit_2[T.5.0]	0.0057	0.002	2.749	0.006	0.002	0.010
yakit_2[T.6.0]	-2.098e-06	0.003	-0.001	0.999	-0.005	0.005
yakit_3[T.2.0]	0.0088	0.013	0.680	0.497	-0.017	0.034
yakit_3[T.3.0]	-0.0005	0.021	-0.025	0.980	-0.042	0.041
yakit_3[T.4.0]	0.0007	0.011	0.065	0.948	-0.021	0.022
yakit_3[T.5.0]	0.0024	0.010	0.235	0.814	-0.017	0.022
yakit_3[T.6.0]	-0.0083	0.011	-0.756	0.450	-0.030	0.013
yakit_mut[T.2.0]	0.0038	0.005	0.755	0.450	-0.006	0.014
yakit_mut[T.3.0]	-0.0036	0.003	-1.162	0.245	-0.010	0.002
yakit_mut[T.4.0]	-0.0038	0.002	-1.945	0.052	-0.008	3e-05
yakit_mut[T.5.0]	0.0046	0.003	1.435	0.151	-0.002	0.011
yakit_mut[T.6.0]	0.0070	0.004	1.902	0.057	-0.000	0.014
yakit_mut[T.7.0]	4.716e-05	0.015	0.003	0.997	-0.029	0.029

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.8.0]	-0.0007	0.009	-0.079	0.937	-0.018	0.017
yakit_su[T.2.0]	-0.0003	0.003	-0.083	0.934	-0.007	0.006
yakit_su[T.3.0]	-0.0070	0.003	-2.524	0.012	-0.012	-0.002
yakit_su[T.4.0]	-0.0056	0.002	-3.004	0.003	-0.009	-0.002
yakit_su[T.5.0]	-0.0037	0.002	-2.419	0.016	-0.007	-0.001
yakit_su[T.6.0]	0.0067	0.002	3.477	0.001	0.003	0.010
yakit_su[T.7.0]	-0.0075	0.002	-4.483	0.000	-0.011	-0.004
yakit_su[T.8.0]	0.0021	0.005	0.406	0.685	-0.008	0.012
zor_egiti[T.2.0]	-0.0062	0.002	-3.483	0.000	-0.010	-0.003
zor_egiti[T.3.0]	-0.0028	0.002	-1.669	0.095	-0.006	0.000
zor_egiti[T.4.0]	-0.0005	0.002	-0.274	0.784	-0.004	0.003
banyo[T.1.0]	0.0028	0.001	1.932	0.053	-3.99e-05	0.006
sauna[T.1.0]	-0.0004	0.010	-0.040	0.968	-0.020	0.019
jakuzi[T.1.0]	-0.0087	0.004	-2.241	0.025	-0.016	-0.001
tuvalet[T.1.0]	-0.0045	0.001	-4.931	0.000	-0.006	-0.003
mutfak[T.1.0]	0.0013	0.002	0.770	0.442	-0.002	0.005
copogut[T.1.0]	-0.0018	0.005	-0.389	0.697	-0.011	0.007
tabanisi[T.1.0]	0.0062	0.004	1.615	0.106	-0.001	0.014
su_sis[T.1.0]	0.0152	0.001	10.786	0.000	0.012	0.018
elektri[T.1.0]	0.0019	0.016	0.116	0.908	-0.030	0.034
sicaksu[T.1.0]	-0.0009	0.001	-1.271	0.204	-0.002	0.000
kabloyay[T.1.0]	0.0004	0.001	0.347	0.728	-0.002	0.003
asansor[T.1.0]	-0.0004	0.001	-0.484	0.628	-0.002	0.001
garaj[T.1.0]	0.0031	0.001	2.674	0.007	0.001	0.005
havuz[T.1.0]	-0.0019	0.003	-0.566	0.571	-0.009	0.005
jenrator[T.1.0]	-0.0059	0.003	-2.225	0.026	-0.011	-0.001
apartman[T.1.0]	0.0071	0.001	7.645	0.000	0.005	0.009
yazlik[T.1.0]	-0.0105	0.002	-5.541	0.000	-0.014	-0.007
tarla[T.1.0]	0.0008	0.001	1.093	0.274	-0.001	0.002
bag[T.1.0]	0.0026	0.001	2.960	0.003	0.001	0.004
sera[T.1.0]	-0.0208	0.003	-7.425	0.000	-0.026	-0.015
arsa[T.1.0]	0.0003	0.001	0.230	0.818	-0.002	0.003
dukkan[T.1.0]	0.0158	0.001	13.437	0.000	0.013	0.018
otel[T.1.0]	-0.0002	0.013	-0.013	0.990	-0.026	0.025
diger[T.1.0]	0.0145	0.005	2.669	0.008	0.004	0.025
kirkntkd[T.2.0]	-0.0055	0.001	-7.805	0.000	-0.007	-0.004
yas	0.0002	0.000	0.996	0.319	-0.000	0.000
ailegrup	-7.54e-06	1.91e-06	-3.956	0.000	-1.13e-05	-3.8e-06
egitim	0.0005	0.000	3.430	0.001	0.000	0.001
sure_yil	-2.585e-06	6e-06	-0.431	0.666	-1.43e-05	9.17e-06
kac_kisi	-1.108e-06	1.63e-06	-0.682	0.495	-4.3e-06	2.08e-06
ucrn_ay	-6.16e-07	1.63e-06	-0.378	0.705	-3.81e-06	2.58e-06
ucra_ay	-5.068e-05	1.2e-05	-4.210	0.000	-7.43e-05	-2.71e-05
ucrn_yil	4.469e-07	1.06e-07	4.204	0.000	2.39e-07	6.55e-07
ikr_yil	2.964e-06	4.76e-07	6.227	0.000	2.03e-06	3.9e-06

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
prim_yil	2.671e-08	4.21e-07	0.063	0.949	-7.98e-07	8.52e-07
konayni	8.882e-07	1.66e-06	0.535	0.593	-2.37e-06	4.14e-06
ulsayni	-1.995e-06	1.43e-06	-1.397	0.163	-4.8e-06	8.05e-07
egsayni	-4.484e-06	3.45e-06	-1.300	0.194	-1.12e-05	2.28e-06
yemayni	1.347e-06	1.25e-06	1.079	0.281	-1.1e-06	3.79e-06
kresayni	-2.241e-05	4.81e-06	-4.660	0.000	-3.18e-05	-1.3e-05
giyayni	1.908e-05	5.88e-06	3.243	0.001	7.55e-06	3.06e-05
gidaayni	-2.723e-06	3.06e-06	-0.889	0.374	-8.73e-06	3.28e-06
digayni	-2.813e-06	3.27e-06	-0.859	0.390	-9.23e-06	3.61e-06
calsur_e	0.0002	0.000	2.034	0.042	7.7e-06	0.000
ekgla_ay	1.687e-05	6.16e-05	0.274	0.784	-0.000	0.000
ekgln_yl	-5.131e-07	3.05e-07	-1.681	0.093	-1.11e-06	8.52e-08
ekgla_yl	1.879e-06	5.54e-06	0.339	0.734	-8.98e-06	1.27e-05
calay_yil	0.0001	0.000	1.223	0.221	-7.48e-05	0.000
digln_yl	2.051e-06	4.35e-07	4.711	0.000	1.2e-06	2.9e-06
digla_yl	2.029e-06	2.62e-06	0.775	0.438	-3.1e-06	7.16e-06
gmkn_ay	-7.375e-07	2.7e-06	-0.273	0.785	-6.03e-06	4.55e-06
gmka_ay	0.0004	0.000	3.217	0.001	0.000	0.001
gmkn_yl	4.725e-07	1.99e-07	2.377	0.017	8.29e-08	8.62e-07
gmka_yl	6.082e-07	3.67e-07	1.658	0.097	-1.11e-07	1.33e-06
mulka_ay	-0.0002	0.000	-1.146	0.252	-0.000	0.000
banka_yl	-1.888e-07	2.44e-07	-0.774	0.439	-6.67e-07	2.89e-07
doviz_yl	1.751e-06	1.25e-06	1.402	0.161	-6.98e-07	4.2e-06
nema_yl	-6.271e-07	4.37e-07	-1.436	0.151	-1.48e-06	2.29e-07
temet_yl	-2.05e-06	6.72e-07	-3.049	0.002	-3.37e-06	-7.32e-07
kar_yl	-3.96e-07	8.47e-07	-0.467	0.640	-2.06e-06	1.26e-06
toprn_yl	1.058e-06	3.87e-07	2.732	0.006	2.99e-07	1.82e-06
topra_yl	1.643e-06	6.09e-06	0.270	0.787	-1.03e-05	1.36e-05
trsn_ay	-1.642e-05	1.66e-06	-9.907	0.000	-1.97e-05	-1.32e-05
emekl_yl	1.601e-06	1.56e-07	10.287	0.000	1.3e-06	1.91e-06
viade_yl	8.623e-06	2.9e-06	2.970	0.003	2.93e-06	1.43e-05
yasli_yl	6.273e-06	1.46e-06	4.308	0.000	3.42e-06	9.13e-06
sosy_yl	4.194e-06	1.37e-06	3.071	0.002	1.52e-06	6.87e-06
hasta_yl	7.074e-07	1.62e-06	0.435	0.663	-2.48e-06	3.89e-06
dul_yl	2.304e-06	3.17e-07	7.275	0.000	1.68e-06	2.92e-06
gazi_yl	2.713e-06	6.88e-07	3.946	0.000	1.37e-06	4.06e-06
burs_yl	1.558e-06	7.9e-07	1.973	0.049	1.01e-08	3.11e-06
issiz_yl	4.223e-06	2.9e-06	1.458	0.145	-1.45e-06	9.9e-06
ddestek_yl	3.245e-06	7.33e-07	4.427	0.000	1.81e-06	4.68e-06
dayni_yl	3.439e-06	2.14e-06	1.609	0.108	-7.51e-07	7.63e-06
ydemk_yl	1.716e-06	3.49e-07	4.914	0.000	1.03e-06	2.4e-06
yddav_yl	-6.028e-07	5.56e-07	-1.085	0.278	-1.69e-06	4.87e-07
ydayn_yl	1.036e-05	9.44e-06	1.097	0.273	-8.15e-06	2.89e-05
dgnfk_yl	1.804e-08	2.49e-07	0.072	0.942	-4.7e-07	5.06e-07
dgbrs_yl	1.063e-06	5.81e-07	1.830	0.067	-7.55e-08	2.2e-06

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
dsemk_yl	-5.737e-07	8.52e-07	-0.673	0.501	-2.24e-06	1.1e-06
dgayn_yl	6.804e-06	1.25e-06	5.460	0.000	4.36e-06	9.25e-06
topga_ay	6.162e-06	7.64e-06	0.807	0.420	-8.81e-06	2.11e-05
topgelay	2.569e-07	4.25e-07	0.604	0.546	-5.76e-07	1.09e-06
topga_yl	1.768e-06	8.18e-07	2.160	0.031	1.64e-07	3.37e-06
topgelyl	-3.309e-07	5.91e-08	-5.599	0.000	-4.47e-07	-2.15e-07
indeks_arl	0.0468	0.007	6.511	0.000	0.033	0.061
kira_ay	-8.342e-05	3.13e-06	-26.679	0.000	-8.95e-05	-7.73e-05
loj_fert	0.0029	0.005	0.587	0.557	-0.007	0.013
izafi_k	-5.897e-05	2.52e-06	-23.426	0.000	-6.39e-05	-5.4e-05
konutsah	3.901e-06	9.12e-07	4.277	0.000	2.11e-06	5.69e-06
kon_sure	0.0003	9.41e-05	2.689	0.007	6.86e-05	0.000
kon_borc	-3.446e-08	5.82e-09	-5.918	0.000	-4.59e-08	-2.3e-08
kondeger	-1.018e-07	9.34e-09	-10.902	0.000	-1.2e-07	-8.35e-08
otur_sur	4.761e-05	2.87e-05	1.657	0.097	-8.7e-06	0.000
bina_tar	0.0017	0.000	5.285	0.000	0.001	0.002
oda_say	0.0027	0.001	4.808	0.000	0.002	0.004
kon_alan	-2.943e-05	1.48e-05	-1.989	0.047	-5.84e-05	-4.26e-07
konutiki	0.0054	0.001	4.483	0.000	0.003	0.008
konut2_n	-0.0122	0.002	-6.361	0.000	-0.016	-0.008
sudepo	0.0041	0.002	2.625	0.009	0.001	0.007
telefon	-0.0022	0.001	-3.746	0.000	-0.003	-0.001
ceptel	-0.0029	0.000	-11.049	0.000	-0.003	-0.002
bilgisay	0.0001	0.001	0.224	0.823	-0.001	0.001
intern	-0.0018	0.001	-2.574	0.010	-0.003	-0.000
televiz	0.0004	0.001	0.622	0.534	-0.001	0.001
video	0.0050	0.002	3.330	0.001	0.002	0.008
lcdtele	-0.0128	0.001	-18.767	0.000	-0.014	-0.011
dvd_vcd	0.0009	0.001	1.343	0.179	-0.000	0.002
kamera	-0.0058	0.001	-5.200	0.000	-0.008	-0.004
uyduant	0.0006	0.001	0.839	0.401	-0.001	0.002
buzdolab	-0.0058	0.001	-4.173	0.000	-0.008	-0.003
derindon	-0.0014	0.001	-1.821	0.069	-0.003	0.000
bulasik	0.0041	0.001	6.343	0.000	0.003	0.005
m_firin	-0.0047	0.001	-5.897	0.000	-0.006	-0.003
camasir	-0.0061	0.001	-6.306	0.000	-0.008	-0.004
kurutma	-0.0104	0.002	-4.875	0.000	-0.015	-0.006
haliyika	-0.0022	0.001	-2.891	0.004	-0.004	-0.001
klima	-0.0005	0.001	-0.722	0.470	-0.002	0.001
otomobil	-0.0129	0.001	-20.065	0.000	-0.014	-0.012
otoadet	-0.0239	0.001	-28.609	0.000	-0.026	-0.022
isotoadt	-0.0112	0.002	-6.020	0.000	-0.015	-0.008
motoadet	-0.0056	0.001	-5.601	0.000	-0.008	-0.004
dentadet	0.0034	0.007	0.478	0.633	-0.011	0.017
muskonut	0.0057	0.001	7.404	0.000	0.004	0.007

Table C.5 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
musmik	-0.0004	0.001	-0.384	0.701	-0.002	0.002
muskira	0.0049	0.002	3.089	0.002	0.002	0.008
muskdeg	-5.934e-05	7.57e-06	-7.838	0.000	-7.42e-05	-4.45e-05
aptkira	0.0064	0.001	10.532	0.000	0.005	0.008
aptkdeg	-2.95e-05	1.75e-06	-16.893	0.000	-3.29e-05	-2.61e-05
yazmik	-0.0018	0.006	-0.290	0.772	-0.014	0.011
yazkira	-0.0016	0.008	-0.185	0.854	-0.018	0.015
yazkdeg	0.0003	5.59e-05	4.738	0.000	0.000	0.000
tarmik	0.0001	6.67e-06	15.662	0.000	9.14e-05	0.000
tarkira	-0.0001	1.97e-05	-6.289	0.000	-0.000	-8.53e-05
tarkdeg	1.014e-05	2.14e-06	4.734	0.000	5.94e-06	1.43e-05
bagmik	5.668e-06	4.74e-06	1.197	0.231	-3.62e-06	1.5e-05
bagkira	0.0018	0.000	3.881	0.000	0.001	0.003
bagkdeg	-7.612e-06	1.26e-05	-0.603	0.547	-3.24e-05	1.71e-05
seramik	1.283e-05	1.78e-06	7.213	0.000	9.34e-06	1.63e-05
serakira	5.663e-05	7.29e-06	7.765	0.000	4.23e-05	7.09e-05
serakdeg	-0.0002	1.37e-05	-15.489	0.000	-0.000	-0.000
arsamik	1.616e-06	1.05e-06	1.537	0.124	-4.45e-07	3.68e-06
arsakira	-2.792e-06	7.05e-06	-0.396	0.692	-1.66e-05	1.1e-05
arsakdeg	1.474e-05	1.51e-05	0.979	0.328	-1.48e-05	4.42e-05
arsadeg	2.337e-08	6.72e-09	3.480	0.001	1.02e-08	3.65e-08
dukkira	-0.0133	0.001	-17.555	0.000	-0.015	-0.012
dukkdeg	-6.037e-08	1.29e-06	-0.047	0.963	-2.58e-06	2.46e-06
dukdeg	1.59e-09	1.01e-08	0.158	0.874	-1.81e-08	2.13e-08
otelmik	0.0019	0.029	0.065	0.948	-0.056	0.059
otelkira	0.0002	0.029	0.007	0.995	-0.057	0.058
otelkdeg	2.6e-05	1.39e-05	1.870	0.062	-1.26e-06	5.32e-05
oteldeg	1.491e-09	9.57e-09	0.156	0.876	-1.73e-08	2.03e-08
digmik	0.0055	0.008	0.668	0.504	-0.011	0.021
digkira	0.0030	0.013	0.230	0.818	-0.023	0.029
digkdeg	4.344e-05	4.53e-06	9.594	0.000	3.46e-05	5.23e-05
digdeg	-2.373e-08	1.15e-08	-2.069	0.039	-4.62e-08	-1.25e-09
yilkulgel	2.244e-06	2.41e-08	93.260	0.000	2.2e-06	2.29e-06
hhb	-0.0030	0.000	-19.326	0.000	-0.003	-0.003
faktor	1.634e-06	2.26e-07	7.218	0.000	1.19e-06	2.08e-06

Table C.6 : Quantile Regression Results w/ FS - 10th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.03469
Model:	QuantReg	**Bandwidth:**	0.01004
Method:	Least Squares	**Sparsity:**	0.4604
Date:	Paz, 21 Tem 2019	**No. Observations:**	371613
Time:	17:44:16	**Df Residuals:**	371357

Quantile Regression Coefficients w/ FS for 10th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	0.1251	0.029	4.355	0.000	0.069	0.181
year[T.2003]	0.0032	0.001	2.842	0.004	0.001	0.005
year[T.2004]	-0.0016	0.002	-0.970	0.332	-0.005	0.002
year[T.2005]	-0.0023	0.001	-1.621	0.105	-0.005	0.000
year[T.2006]	-0.0006	0.001	-0.445	0.657	-0.003	0.002
year[T.2007]	0.0002	0.001	0.110	0.912	-0.003	0.003
year[T.2008]	-0.0050	0.001	-3.324	0.001	-0.008	-0.002
year[T.2009]	-0.0042	0.002	-2.254	0.024	-0.008	-0.001
year[T.2010]	-0.0047	0.002	-2.557	0.011	-0.008	-0.001
year[T.2011]	-0.0064	0.002	-3.442	0.001	-0.010	-0.003
year[T.2012]	-0.0105	0.002	-5.738	0.000	-0.014	-0.007
year[T.2013]	-0.0060	0.002	-3.260	0.001	-0.010	-0.002
year[T.2014]	-0.0048	0.002	-2.588	0.010	-0.008	-0.001
year[T.2015]	-0.0045	0.002	-2.450	0.014	-0.008	-0.001
year[T.2016]	0.0068	0.002	3.693	0.000	0.003	0.010
cinsiyet[T.2]	-0.0010	0.001	-1.537	0.124	-0.002	0.000
yakinlik[T.1.0]	0.0059	0.001	5.768	0.000	0.004	0.008
yakinlik[T.2.0]	0.0034	0.001	2.705	0.007	0.001	0.006
yakinlik[T.3.0]	0.0061	0.002	3.331	0.001	0.003	0.010
yakinlik[T.4.0]	-0.0003	0.003	-0.107	0.915	-0.006	0.005
yakinlik[T.5.0]	-0.0009	0.004	-0.227	0.820	-0.009	0.007
yakinlik[T.6.0]	0.0075	0.002	3.834	0.000	0.004	0.011
yakinlik[T.7.0]	0.0072	0.002	3.907	0.000	0.004	0.011
yakinlik[T.8.0]	0.0051	0.002	2.130	0.033	0.000	0.010
yakinlik[T.9.0]	-0.0012	0.004	-0.301	0.763	-0.009	0.007
yakinlik[T.10.0]	-0.0206	0.007	-3.113	0.002	-0.034	-0.008
sagliks[T.2.0]	0.0053	0.002	3.003	0.003	0.002	0.009
sagliks[T.3.0]	-0.0070	0.001	-7.754	0.000	-0.009	-0.005
sagliks[T.4.0]	-0.0042	0.001	-5.875	0.000	-0.006	-0.003
ogrenci[T.2.0]	0.0014	0.001	1.447	0.148	-0.001	0.003
egitim_devam[T.2]	0.0012	0.002	0.571	0.568	-0.003	0.005
egitim_devam[T.3]	0.0003	0.003	0.118	0.906	-0.005	0.006
egitim_devam[T.4]	-0.0030	0.004	-0.793	0.428	-0.011	0.004
egitim_devam[T.5]	-0.0086	0.003	-2.626	0.009	-0.015	-0.002
egitim_devam[T.6]	-0.0128	0.005	-2.445	0.014	-0.023	-0.003
egitim_devam[T.7]	-0.0059	0.019	-0.320	0.749	-0.042	0.030
egitim_devam[T.8]	0.0173	0.041	0.426	0.670	-0.062	0.097

Table C.6 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-0.0182	0.106	-0.173	0.863	-0.225	0.189
egitim_devam[T.999]	-8.347e-05	0.001	-0.059	0.953	-0.003	0.003
medenidr[T.2.0]	-0.0048	0.001	-3.744	0.000	-0.007	-0.002
medenidr[T.3.0]	0.0045	0.002	2.431	0.015	0.001	0.008
medenidr[T.4.0]	0.0019	0.002	0.834	0.404	-0.003	0.006
is_arama[T.2]	0.0016	0.002	0.990	0.322	-0.002	0.005
is_arama[T.999]	0.0045	0.002	2.383	0.017	0.001	0.008
isneden[T.2.0]	0.0055	0.009	0.618	0.537	-0.012	0.023
isneden[T.3.0]	0.0091	0.009	1.013	0.311	-0.008	0.027
isneden[T.4.0]	0.0131	0.009	1.465	0.143	-0.004	0.031
isneden[T.5.0]	0.0188	0.015	1.260	0.208	-0.010	0.048
isneden[T.6.0]	0.0102	0.009	1.128	0.259	-0.008	0.028
isneden[T.7.0]	0.0090	0.009	0.990	0.322	-0.009	0.027
isneden[T.10.0]	0.0172	0.010	1.733	0.083	-0.002	0.037
isneden[T.11.0]	0.0031	0.009	0.339	0.734	-0.015	0.021
sekkod[T.2.0]	0.0020	0.005	0.392	0.695	-0.008	0.012
sekkod[T.3.0]	-0.0018	0.002	-0.859	0.391	-0.006	0.002
sekkod[T.4.0]	-0.0057	0.004	-1.585	0.113	-0.013	0.001
sekkod[T.5.0]	-0.0061	0.002	-2.602	0.009	-0.011	-0.001
sekkod[T.6.0]	0.0014	0.002	0.725	0.468	-0.002	0.005
sekkod[T.7.0]	-0.0056	0.003	-2.254	0.024	-0.011	-0.001
sekkod[T.8.0]	-0.0017	0.002	-0.694	0.488	-0.007	0.003
sekkod[T.9.0]	-0.0060	0.004	-1.541	0.123	-0.014	0.002
sekkod[T.10.0]	-0.0028	0.004	-0.768	0.442	-0.010	0.004
sekkod[T.11.0]	-0.0019	0.002	-0.784	0.433	-0.007	0.003
sekkod[T.12.0]	0.0046	0.003	1.700	0.089	-0.001	0.010
sekkod[T.13.0]	-0.0030	0.003	-1.065	0.287	-0.009	0.003
sekkod[T.14.0]	-0.0031	0.003	-1.226	0.220	-0.008	0.002
sgk[T.2.0]	0.0077	0.008	0.911	0.362	-0.009	0.024
sgk[T.3.0]	-0.0041	0.001	-3.182	0.001	-0.007	-0.002
is_statu[T.2]	0.0022	0.002	1.072	0.284	-0.002	0.006
is_statu[T.3]	0.0276	0.005	5.428	0.000	0.018	0.038
is_statu[T.999]	-0.0035	0.002	-2.319	0.020	-0.007	-0.001
koopera[T.2]	-0.0007	0.003	-0.193	0.847	-0.007	0.006
koopera[T.999]	-0.0011	0.003	-0.342	0.733	-0.007	0.005
sir_kon[T.2]	-0.0164	0.005	-3.300	0.001	-0.026	-0.007
sir_kon[T.999]	-0.0123	0.005	-2.606	0.009	-0.022	-0.003
isdur[T.2.0]	0.0041	0.002	1.905	0.057	-0.000	0.008
isdur[T.3.0]	0.0048	0.002	1.996	0.046	8.64e-05	0.010
isdur[T.4.0]	0.0032	0.002	1.752	0.080	-0.000	0.007
isdur[T.5.0]	0.0009	0.002	0.416	0.677	-0.003	0.005
ist_tesk[T.2.0]	-0.0039	0.003	-1.549	0.121	-0.009	0.001
ist_tesk[T.3.0]	-0.0028	0.003	-1.054	0.292	-0.008	0.002
guc_kul[T.2]	-0.0015	0.002	-0.759	0.448	-0.005	0.002
guc_kul[T.3]	-0.0007	0.002	-0.443	0.658	-0.004	0.002

Table C.6 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.999]	0.0042	0.002	2.364	0.018	0.001	0.008
isfaal[T.2.0]	-0.0018	0.002	-0.787	0.431	-0.006	0.003
mulk_dur[T.2.0]	-0.0103	0.002	-4.323	0.000	-0.015	-0.006
mulk_dur[T.3.0]	-0.0122	0.003	-3.781	0.000	-0.018	-0.006
ekis_yl[T.2.0]	0.0002	0.003	0.055	0.956	-0.006	0.007
ekis_ay[T.2.0]	0.0120	0.004	2.877	0.004	0.004	0.020
sekkod_e[T.2.0]	0.1252	0.063	1.982	0.047	0.001	0.249
sekkod_e[T.3.0]	0.0551	0.011	4.896	0.000	0.033	0.077
sekkod_e[T.4.0]	0.2467	0.045	5.429	0.000	0.158	0.336
sekkod_e[T.5.0]	0.0355	0.010	3.487	0.000	0.016	0.055
sekkod_e[T.6.0]	0.1010	0.010	10.007	0.000	0.081	0.121
sekkod_e[T.7.0]	0.0837	0.012	6.898	0.000	0.060	0.108
sekkod_e[T.8.0]	0.0722	0.013	5.772	0.000	0.048	0.097
sekkod_e[T.9.0]	0.4108	0.050	8.200	0.000	0.313	0.509
sekkod_e[T.10.0]	0.1587	0.025	6.291	0.000	0.109	0.208
sekkod_e[T.11.0]	0.0998	0.014	7.319	0.000	0.073	0.127
sekkod_e[T.12.0]	0.3177	0.023	13.991	0.000	0.273	0.362
sekkod_e[T.13.0]	0.3555	0.030	11.764	0.000	0.296	0.415
sekkod_e[T.14.0]	0.0498	0.010	5.004	0.000	0.030	0.069
meslek_e[T.1.0]	-0.0315	0.008	-3.786	0.000	-0.048	-0.015
meslek_e[T.2.0]	-0.2608	0.012	-22.461	0.000	-0.284	-0.238
meslek_e[T.3.0]	-0.1093	0.011	-10.336	0.000	-0.130	-0.089
meslek_e[T.4.0]	0.0059	0.037	0.159	0.874	-0.067	0.079
meslek_e[T.5.0]	-0.0481	0.009	-5.089	0.000	-0.067	-0.030
meslek_e[T.6.0]	0.0399	0.009	4.396	0.000	0.022	0.058
meslek_e[T.7.0]	-0.0288	0.008	-3.388	0.001	-0.045	-0.012
meslek_e[T.8.0]	-0.0394	0.009	-4.255	0.000	-0.058	-0.021
meslek_e[T.9.0]	0.0169	0.008	2.153	0.031	0.002	0.032
isdur_e[T.2.0]	0.0531	0.008	6.839	0.000	0.038	0.068
isdur_e[T.3.0]	0.0318	0.010	3.082	0.002	0.012	0.052
isdur_e[T.4.0]	0.0194	0.007	2.763	0.006	0.006	0.033
isdur_e[T.5.0]	0.0174	0.008	2.048	0.041	0.001	0.034
cl_neden[T.2.0]	-0.0042	0.005	-0.869	0.385	-0.014	0.005
cl_neden[T.3.0]	0.0071	0.003	2.788	0.005	0.002	0.012
cl_neden[T.4.0]	0.0092	0.005	1.839	0.066	-0.001	0.019
cl_neden[T.5.0]	0.0163	0.005	3.005	0.003	0.006	0.027
cl_neden[T.6.0]	0.0092	0.004	2.435	0.015	0.002	0.017
cl_neden[T.7.0]	-0.0019	0.004	-0.439	0.661	-0.010	0.006
ayni_is[T.2.0]	-0.0096	0.005	-1.888	0.059	-0.020	0.000
sekkod_d[T.2.0]	0.0156	0.016	0.989	0.323	-0.015	0.046
sekkod_d[T.3.0]	0.0276	0.007	3.938	0.000	0.014	0.041
sekkod_d[T.4.0]	0.0349	0.012	2.888	0.004	0.011	0.059
sekkod_d[T.5.0]	0.0153	0.006	2.396	0.017	0.003	0.028
sekkod_d[T.6.0]	0.0212	0.007	2.893	0.004	0.007	0.036
sekkod_d[T.7.0]	0.0129	0.009	1.503	0.133	-0.004	0.030

Table C.6 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.8.0]	0.0323	0.008	3.926	0.000	0.016	0.048
sekkod_d[T.9.0]	0.0471	0.019	2.509	0.012	0.010	0.084
sekkod_d[T.10.0]	0.0296	0.013	2.212	0.027	0.003	0.056
sekkod_d[T.11.0]	0.0347	0.010	3.445	0.001	0.015	0.054
sekkod_d[T.12.0]	0.0232	0.013	1.726	0.084	-0.003	0.049
sekkod_d[T.13.0]	0.0136	0.014	0.944	0.345	-0.015	0.042
sekkod_d[T.14.0]	0.0145	0.008	1.742	0.081	-0.002	0.031
meslek_d[T.2.0]	-0.0326	0.012	-2.715	0.007	-0.056	-0.009
meslek_d[T.3.0]	-0.0165	0.010	-1.624	0.104	-0.037	0.003
meslek_d[T.4.0]	-0.0287	0.010	-2.799	0.005	-0.049	-0.009
meslek_d[T.5.0]	-0.0242	0.009	-2.717	0.007	-0.042	-0.007
meslek_d[T.6.0]	-0.0105	0.011	-0.925	0.355	-0.033	0.012
meslek_d[T.7.0]	-0.0321	0.009	-3.663	0.000	-0.049	-0.015
meslek_d[T.8.0]	-0.0353	0.009	-3.756	0.000	-0.054	-0.017
meslek_d[T.9.0]	-0.0277	0.009	-3.061	0.002	-0.045	-0.010
isdur_d[T.2.0]	0.0040	0.004	0.945	0.345	-0.004	0.012
isdur_d[T.3.0]	-0.0162	0.014	-1.184	0.237	-0.043	0.011
isdur_d[T.4.0]	-0.0067	0.006	-1.070	0.284	-0.019	0.006
isdur_d[T.5.0]	-0.0030	0.008	-0.402	0.688	-0.018	0.012
hhtipi[T.2.0]	-0.0039	0.001	-5.000	0.000	-0.005	-0.002
hhtipi[T.3.0]	-0.0067	0.001	-8.182	0.000	-0.008	-0.005
hhtipi[T.4.0]	0.0040	0.001	3.802	0.000	0.002	0.006
hhtipi[T.5.0]	-0.0011	0.001	-1.350	0.177	-0.003	0.001
hhtipi[T.6.0]	0.0028	0.001	2.097	0.036	0.000	0.005
hhtipi[T.7.0]	-0.0024	0.003	-0.899	0.369	-0.008	0.003
konuttip[T.2.0]	-0.0023	0.001	-2.172	0.030	-0.004	-0.000
konuttip[T.3.0]	-0.0013	0.001	-1.444	0.149	-0.003	0.000
konuttip[T.4.0]	0.0107	0.008	1.365	0.172	-0.005	0.026
mulkiyet[T.2.0]	-0.0052	0.001	-6.528	0.000	-0.007	-0.004
mulkiyet[T.3.0]	0.0107	0.002	5.999	0.000	0.007	0.014
mulkiyet[T.4.0]	-0.0016	0.001	-1.780	0.075	-0.003	0.000
kira_sek[T.2.0]	0.0170	0.006	2.858	0.004	0.005	0.029
yapituru[T.2.0]	-0.0270	0.018	-1.512	0.131	-0.062	0.008
yapituru[T.3.0]	0.0002	0.002	0.084	0.933	-0.004	0.005
yapituru[T.4.0]	-0.0056	0.001	-4.098	0.000	-0.008	-0.003
yapituru[T.5.0]	-0.0032	0.002	-1.983	0.047	-0.006	-3.64e-05
yapituru[T.6.0]	-0.0039	0.001	-3.453	0.001	-0.006	-0.002
yapituru[T.7.0]	0.0039	0.002	2.334	0.020	0.001	0.007
yapituru[T.8.0]	-0.0089	0.007	-1.348	0.177	-0.022	0.004
zemin_sal[T.2.0]	0.0017	0.001	1.549	0.121	-0.000	0.004
zemin_sal[T.3.0]	0.0016	0.001	1.289	0.197	-0.001	0.004
zemin_sal[T.4.0]	-0.0015	0.001	-1.081	0.280	-0.004	0.001
zemin_sal[T.5.0]	-0.0038	0.003	-1.480	0.139	-0.009	0.001
zemin_sal[T.6.0]	0.0011	0.001	0.786	0.432	-0.002	0.004
zemin_sal[T.7.0]	0.0115	0.003	4.248	0.000	0.006	0.017

Table C.6 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.8.0]	0.0035	0.008	0.453	0.651	-0.012	0.019
zemin_sal[T.9.0]	-0.0013	0.003	-0.494	0.621	-0.007	0.004
zemin_sal[T.98.0]	-0.0081	0.006	-1.286	0.199	-0.021	0.004
zemin_mut[T.2.0]	0.0103	0.002	4.131	0.000	0.005	0.015
zemin_mut[T.3.0]	0.0045	0.002	2.647	0.008	0.001	0.008
zemin_mut[T.4.0]	0.0115	0.003	3.765	0.000	0.006	0.018
zemin_mut[T.5.0]	0.0029	0.006	0.476	0.634	-0.009	0.015
zemin_mut[T.6.0]	0.0104	0.002	4.649	0.000	0.006	0.015
zemin_mut[T.7.0]	-0.0019	0.003	-0.577	0.564	-0.008	0.005
zemin_mut[T.8.0]	-0.0107	0.006	-1.768	0.077	-0.022	0.001
zemin_mut[T.9.0]	0.0127	0.004	3.407	0.001	0.005	0.020
zemin_mut[T.98.0]	0.0261	0.007	3.526	0.000	0.012	0.041
zemin_ban[T.2.0]	-0.1534	0.019	-8.001	0.000	-0.191	-0.116
zemin_ban[T.3.0]	-0.1209	0.017	-6.998	0.000	-0.155	-0.087
zemin_ban[T.4.0]	-0.1352	0.018	-7.351	0.000	-0.171	-0.099
zemin_ban[T.5.0]	0.0257	0.044	0.579	0.563	-0.061	0.112
zemin_ban[T.6.0]	-0.1228	0.017	-7.099	0.000	-0.157	-0.089
zemin_ban[T.7.0]	-0.1139	0.017	-6.529	0.000	-0.148	-0.080
zemin_ban[T.8.0]	-0.1198	0.018	-6.591	0.000	-0.155	-0.084
zemin_ban[T.9.0]	-0.1127	0.018	-6.326	0.000	-0.148	-0.078
zemin_ban[T.98.0]	-0.0437	0.022	-2.008	0.045	-0.086	-0.001
isit_sis[T.2.0]	-0.0049	0.001	-4.384	0.000	-0.007	-0.003
isit_sis[T.3.0]	-0.0057	0.001	-4.666	0.000	-0.008	-0.003
isit_sis[T.4.0]	-0.0008	0.002	-0.365	0.715	-0.005	0.004
yakit_1[T.2.0]	0.0036	0.001	3.924	0.000	0.002	0.005
yakit_1[T.3.0]	0.0012	0.001	0.962	0.336	-0.001	0.004
yakit_1[T.4.0]	0.0016	0.001	1.165	0.244	-0.001	0.004
yakit_1[T.5.0]	0.0039	0.002	2.322	0.020	0.001	0.007
yakit_1[T.6.0]	-0.0009	0.002	-0.623	0.533	-0.004	0.002
yakit_2[T.2.0]	0.0005	0.002	0.302	0.762	-0.003	0.004
yakit_2[T.3.0]	-0.0083	0.009	-0.905	0.365	-0.026	0.010
yakit_2[T.4.0]	-0.0106	0.003	-3.683	0.000	-0.016	-0.005
yakit_2[T.5.0]	0.0019	0.002	0.919	0.358	-0.002	0.006
yakit_2[T.6.0]	0.0014	0.003	0.503	0.615	-0.004	0.007
yakit_3[T.2.0]	0.0330	0.008	4.255	0.000	0.018	0.048
yakit_3[T.3.0]	0.0270	0.015	1.810	0.070	-0.002	0.056
yakit_3[T.4.0]	0.0236	0.006	3.640	0.000	0.011	0.036
yakit_3[T.5.0]	0.0211	0.006	3.723	0.000	0.010	0.032
yakit_3[T.6.0]	0.0145	0.006	2.246	0.025	0.002	0.027
yakit_mut[T.2.0]	0.0180	0.005	3.630	0.000	0.008	0.028
yakit_mut[T.3.0]	0.0011	0.003	0.362	0.718	-0.005	0.007
yakit_mut[T.4.0]	-0.0021	0.002	-1.050	0.294	-0.006	0.002
yakit_mut[T.5.0]	0.0042	0.003	1.319	0.187	-0.002	0.010
yakit_mut[T.6.0]	0.0108	0.004	3.038	0.002	0.004	0.018
yakit_mut[T.7.0]	0.0452	0.018	2.560	0.010	0.011	0.080

Table C.6 (Continued) : Quantile Regression Coefficients for 10th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.8.0]	0.0153	0.010	1.522	0.128	-0.004	0.035
yakit_su[T.2.0]	-0.0138	0.003	-4.670	0.000	-0.020	-0.008
yakit_su[T.3.0]	-0.0090	0.003	-3.193	0.001	-0.015	-0.003
yakit_su[T.4.0]	-0.0063	0.002	-3.312	0.001	-0.010	-0.003
yakit_su[T.5.0]	-0.0057	0.002	-3.676	0.000	-0.009	-0.003
yakit_su[T.6.0]	-0.0043	0.002	-2.273	0.023	-0.008	-0.001
yakit_su[T.7.0]	-0.0051	0.002	-3.044	0.002	-0.008	-0.002
yakit_su[T.8.0]	0.0113	0.005	2.090	0.037	0.001	0.022
zor_egiti[T.2.0]	-0.0050	0.002	-2.889	0.004	-0.008	-0.002
zor_egiti[T.3.0]	-0.0012	0.002	-0.749	0.454	-0.004	0.002
zor_egiti[T.4.0]	-0.0017	0.002	-0.968	0.333	-0.005	0.002
banyo[T.1.0]	0.0026	0.001	1.811	0.070	-0.000	0.005
sauna[T.1.0]	-0.0207	0.008	-2.632	0.008	-0.036	-0.005
jakuzi[T.1.0]	-0.0264	0.004	-7.304	0.000	-0.034	-0.019
tuvalet[T.1.0]	-0.0048	0.001	-5.366	0.000	-0.007	-0.003
mutfak[T.1.0]	-0.0009	0.002	-0.499	0.618	-0.004	0.003
copogut[T.1.0]	-0.0232	0.004	-5.299	0.000	-0.032	-0.015
tabanisi[T.1.0]	-0.0090	0.004	-2.561	0.010	-0.016	-0.002
su_sis[T.1.0]	0.0114	0.001	8.243	0.000	0.009	0.014
elektri[T.1.0]	-0.0315	0.013	-2.390	0.017	-0.057	-0.006
sicaksu[T.1.0]	-0.0016	0.001	-2.338	0.019	-0.003	-0.000
kabloyay[T.1.0]	-0.0017	0.001	-1.400	0.162	-0.004	0.001
asansor[T.1.0]	-0.0003	0.001	-0.376	0.707	-0.002	0.001
garaj[T.1.0]	-0.0069	0.001	-6.069	0.000	-0.009	-0.005
havuz[T.1.0]	-0.0224	0.003	-6.540	0.000	-0.029	-0.016
jenrator[T.1.0]	-0.0173	0.003	-6.671	0.000	-0.022	-0.012
apartman[T.1.0]	0.0087	0.001	10.585	0.000	0.007	0.010
yazlik[T.1.0]	-0.0052	0.002	-3.265	0.001	-0.008	-0.002
tarla[T.1.0]	0.0013	0.001	1.957	0.050	-1.81e-06	0.003
bag[T.1.0]	-0.0004	0.001	-0.519	0.604	-0.002	0.001
sera[T.1.0]	0.0039	0.003	1.448	0.148	-0.001	0.009
arsa[T.1.0]	0.0028	0.001	2.434	0.015	0.001	0.005
dukkan[T.1.0]	0.0080	0.001	7.629	0.000	0.006	0.010
otel[T.1.0]	-0.0191	0.009	-2.158	0.031	-0.036	-0.002
diger[T.1.0]	0.0343	0.004	8.397	0.000	0.026	0.042
kirkntkd[T.2.0]	-0.0061	0.001	-8.936	0.000	-0.007	-0.005
izafi_k	-6.668e-05	1.77e-06	-37.766	0.000	-7.01e-05	-6.32e-05
ceptel	-0.0035	0.000	-14.210	0.000	-0.004	-0.003
yilkulgel	1.9e-06	1.92e-08	98.911	0.000	1.86e-06	1.94e-06

Table C.7 : Quantile Regression Results - 20th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.05560
Model:	QuantReg	**Bandwidth:**	0.01035
Method:	Least Squares	**Sparsity:**	0.4239
Date:	Cts, 23 Haz 2018	**No. Observations:**	371613
Time:	19:43:43	**Df Residuals:**	371226

Quantile Regression Coefficients for 20th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	-0.1047	0.029	-3.603	0.000	-0.162	-0.048
year[T.2003]	0.0138	0.002	8.046	0.000	0.010	0.017
year[T.2004]	0.0105	0.002	4.380	0.000	0.006	0.015
year[T.2005]	0.0143	0.002	6.739	0.000	0.010	0.018
year[T.2006]	0.0171	0.002	8.180	0.000	0.013	0.021
year[T.2007]	0.0082	0.002	3.847	0.000	0.004	0.012
year[T.2008]	-0.0139	0.002	-6.861	0.000	-0.018	-0.010
year[T.2009]	-0.0030	0.001	-2.025	0.043	-0.006	-9.72e-05
year[T.2010]	-0.0088	0.001	-6.008	0.000	-0.012	-0.006
year[T.2011]	-0.0022	0.001	-1.490	0.136	-0.005	0.001
year[T.2012]	-0.0119	0.001	-8.554	0.000	-0.015	-0.009
year[T.2013]	-0.0033	0.001	-2.284	0.022	-0.006	-0.000
year[T.2014]	0.0002	0.001	0.161	0.872	-0.003	0.003
year[T.2015]	0.0139	0.002	7.764	0.000	0.010	0.017
year[T.2016]	0.0258	0.002	14.492	0.000	0.022	0.029
cinsiyet[T.2]	-0.0006	0.001	-0.679	0.497	-0.002	0.001
yakinlik[T.1.0]	0.0375	0.001	25.655	0.000	0.035	0.040
yakinlik[T.2.0]	0.0365	0.002	19.359	0.000	0.033	0.040
yakinlik[T.3.0]	0.0257	0.003	10.098	0.000	0.021	0.031
yakinlik[T.4.0]	0.0396	0.004	10.133	0.000	0.032	0.047
yakinlik[T.5.0]	0.0444	0.006	7.794	0.000	0.033	0.056
yakinlik[T.6.0]	0.0500	0.003	17.739	0.000	0.044	0.056
yakinlik[T.7.0]	0.0441	0.003	16.554	0.000	0.039	0.049
yakinlik[T.8.0]	0.0412	0.003	12.175	0.000	0.035	0.048
yakinlik[T.9.0]	0.0308	0.006	5.569	0.000	0.020	0.042
yakinlik[T.10.0]	-0.0046	0.009	-0.529	0.597	-0.022	0.012
sagliks[T.2.0]	0.0163	0.002	6.693	0.000	0.012	0.021
sagliks[T.3.0]	-0.0270	0.001	-21.508	0.000	-0.030	-0.025
sagliks[T.4.0]	-0.0119	0.001	-11.920	0.000	-0.014	-0.010
ogrenci[T.2.0]	0.0034	0.001	2.533	0.011	0.001	0.006
egitim_devam[T.2]	-0.0006	0.003	-0.206	0.837	-0.006	0.005
egitim_devam[T.3]	-0.0040	0.004	-1.018	0.309	-0.012	0.004
egitim_devam[T.4]	-0.0058	0.005	-1.094	0.274	-0.016	0.005
egitim_devam[T.5]	-0.0151	0.004	-3.347	0.001	-0.024	-0.006
egitim_devam[T.6]	0.0084	0.008	1.112	0.266	-0.006	0.023
egitim_devam[T.7]	0.0410	0.024	1.677	0.094	-0.007	0.089
egitim_devam[T.8]	0.0283	0.050	0.566	0.571	-0.070	0.126

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	0.0090	0.095	0.095	0.924	-0.177	0.195
egitim_devam[T.999]	-0.0033	0.002	-1.689	0.091	-0.007	0.001
medenidr[T.2.0]	-0.0127	0.002	-6.938	0.000	-0.016	-0.009
medenidr[T.3.0]	-0.0010	0.003	-0.361	0.718	-0.007	0.005
medenidr[T.4.0]	-0.0018	0.003	-0.571	0.568	-0.008	0.004
is_arama[T.2]	0.0077	0.002	3.436	0.001	0.003	0.012
is_arama[T.999]	0.0200	0.003	6.991	0.000	0.014	0.026
isneden[T.2.0]	0.1459	0.007	21.095	0.000	0.132	0.159
isneden[T.3.0]	0.1475	0.007	21.654	0.000	0.134	0.161
isneden[T.4.0]	0.1466	0.007	20.800	0.000	0.133	0.160
isneden[T.5.0]	0.0119	0.013	0.934	0.350	-0.013	0.037
isneden[T.6.0]	0.1453	0.007	20.338	0.000	0.131	0.159
isneden[T.7.0]	0.1507	0.007	20.766	0.000	0.136	0.165
isneden[T.10.0]	0.1552	0.009	17.256	0.000	0.138	0.173
isneden[T.11.0]	0.1377	0.007	18.716	0.000	0.123	0.152
sekkod[T.2.0]	0.0443	0.007	6.077	0.000	0.030	0.059
sekkod[T.3.0]	0.0172	0.003	5.956	0.000	0.012	0.023
sekkod[T.4.0]	0.0177	0.005	3.579	0.000	0.008	0.027
sekkod[T.5.0]	0.0122	0.003	3.773	0.000	0.006	0.018
sekkod[T.6.0]	0.0153	0.003	5.558	0.000	0.010	0.021
sekkod[T.7.0]	0.0093	0.003	2.715	0.007	0.003	0.016
sekkod[T.8.0]	0.0215	0.003	6.391	0.000	0.015	0.028
sekkod[T.9.0]	0.0223	0.005	4.118	0.000	0.012	0.033
sekkod[T.10.0]	0.0067	0.005	1.335	0.182	-0.003	0.017
sekkod[T.11.0]	0.0246	0.003	7.125	0.000	0.018	0.031
sekkod[T.12.0]	0.0339	0.004	8.796	0.000	0.026	0.042
sekkod[T.13.0]	0.0255	0.004	6.434	0.000	0.018	0.033
sekkod[T.14.0]	0.0144	0.003	4.144	0.000	0.008	0.021
sgk[T.2.0]	-0.0118	0.011	-1.047	0.295	-0.034	0.010
sgk[T.3.0]	-0.0057	0.002	-3.105	0.002	-0.009	-0.002
is_statu[T.2]	0.0015	0.003	0.463	0.644	-0.005	0.008
is_statu[T.3]	0.0252	0.007	3.509	0.000	0.011	0.039
is_statu[T.999]	-0.0025	0.003	-0.850	0.395	-0.008	0.003
koopera[T.2]	0.0328	0.005	7.037	0.000	0.024	0.042
koopera[T.999]	0.0333	0.004	7.844	0.000	0.025	0.042
sir_kon[T.2]	-0.0621	0.008	-8.167	0.000	-0.077	-0.047
sir_kon[T.999]	-0.0637	0.007	-8.749	0.000	-0.078	-0.049
isdur[T.2.0]	0.0172	0.007	2.487	0.013	0.004	0.031
isdur[T.3.0]	0.0136	0.004	3.554	0.000	0.006	0.021
isdur[T.4.0]	0.0213	0.003	7.287	0.000	0.016	0.027
isdur[T.5.0]	0.0430	0.003	13.173	0.000	0.037	0.049
ist_tesk[T.2.0]	-0.0114	0.007	-1.755	0.079	-0.024	0.001
ist_tesk[T.3.0]	5.408e-05	0.007	0.007	0.994	-0.014	0.014
guc_kul[T.2]	0.0029	0.003	1.067	0.286	-0.002	0.008
guc_kul[T.3]	0.0019	0.002	0.839	0.401	-0.002	0.006

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.999]	-0.0032	0.003	-1.284	0.199	-0.008	0.002
isfaal[T.2.0]	-0.0221	0.003	-6.722	0.000	-0.029	-0.016
mulk_dur[T.2.0]	-0.0117	0.003	-3.517	0.000	-0.018	-0.005
mulk_dur[T.3.0]	-0.0115	0.004	-2.575	0.010	-0.020	-0.003
ekis_yl[T.2.0]	-0.0219	0.005	-4.693	0.000	-0.031	-0.013
ekis_ay[T.2.0]	0.0075	0.005	1.363	0.173	-0.003	0.018
sekkod_e[T.2.0]	0.0537	0.053	1.015	0.310	-0.050	0.157
sekkod_e[T.3.0]	0.1563	0.018	8.919	0.000	0.122	0.191
sekkod_e[T.4.0]	0.0285	0.042	0.674	0.500	-0.054	0.111
sekkod_e[T.5.0]	0.0772	0.014	5.639	0.000	0.050	0.104
sekkod_e[T.6.0]	0.0790	0.014	5.716	0.000	0.052	0.106
sekkod_e[T.7.0]	0.0523	0.017	3.051	0.002	0.019	0.086
sekkod_e[T.8.0]	-0.1554	0.015	-10.607	0.000	-0.184	-0.127
sekkod_e[T.9.0]	0.0333	0.042	0.784	0.433	-0.050	0.117
sekkod_e[T.10.0]	-0.0140	0.028	-0.493	0.622	-0.070	0.042
sekkod_e[T.11.0]	0.0589	0.018	3.213	0.001	0.023	0.095
sekkod_e[T.12.0]	-0.0134	0.024	-0.571	0.568	-0.060	0.033
sekkod_e[T.13.0]	0.0403	0.032	1.255	0.210	-0.023	0.103
sekkod_e[T.14.0]	0.0163	0.014	1.202	0.230	-0.010	0.043
meslek_e[T.1.0]	0.0183	0.011	1.600	0.110	-0.004	0.041
meslek_e[T.2.0]	0.0347	0.017	1.990	0.047	0.001	0.069
meslek_e[T.3.0]	0.0898	0.017	5.165	0.000	0.056	0.124
meslek_e[T.4.0]	-0.0855	0.028	-3.036	0.002	-0.141	-0.030
meslek_e[T.5.0]	-0.0218	0.012	-1.753	0.080	-0.046	0.003
meslek_e[T.6.0]	0.0423	0.010	4.104	0.000	0.022	0.062
meslek_e[T.7.0]	-0.0732	0.012	-5.947	0.000	-0.097	-0.049
meslek_e[T.8.0]	-0.0273	0.013	-2.180	0.029	-0.052	-0.003
meslek_e[T.9.0]	0.0175	0.009	2.008	0.045	0.000	0.035
isdur_e[T.2.0]	-0.0074	0.010	-0.768	0.443	-0.026	0.012
isdur_e[T.3.0]	0.1655	0.017	9.954	0.000	0.133	0.198
isdur_e[T.4.0]	0.0306	0.010	3.213	0.001	0.012	0.049
isdur_e[T.5.0]	0.0382	0.012	3.313	0.001	0.016	0.061
cl_neden[T.2.0]	-0.0398	0.007	-6.076	0.000	-0.053	-0.027
cl_neden[T.3.0]	-0.0079	0.004	-2.098	0.036	-0.015	-0.001
cl_neden[T.4.0]	-0.0232	0.007	-3.433	0.001	-0.036	-0.010
cl_neden[T.5.0]	-0.0092	0.007	-1.262	0.207	-0.024	0.005
cl_neden[T.6.0]	-0.0111	0.005	-2.118	0.034	-0.021	-0.001
cl_neden[T.7.0]	-0.0205	0.006	-3.553	0.000	-0.032	-0.009
ayni_is[T.2.0]	-0.0164	0.007	-2.378	0.017	-0.030	-0.003
sekkod_d[T.2.0]	0.0382	0.034	1.113	0.266	-0.029	0.106
sekkod_d[T.3.0]	0.0496	0.010	5.030	0.000	0.030	0.069
sekkod_d[T.4.0]	0.0657	0.017	3.887	0.000	0.033	0.099
sekkod_d[T.5.0]	0.0112	0.009	1.247	0.213	-0.006	0.029
sekkod_d[T.6.0]	0.0345	0.010	3.341	0.001	0.014	0.055
sekkod_d[T.7.0]	0.0169	0.012	1.436	0.151	-0.006	0.040

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.8.0]	0.0577	0.012	4.873	0.000	0.035	0.081
sekkod_d[T.9.0]	0.0829	0.026	3.198	0.001	0.032	0.134
sekkod_d[T.10.0]	0.0206	0.018	1.161	0.246	-0.014	0.055
sekkod_d[T.11.0]	0.0704	0.015	4.858	0.000	0.042	0.099
sekkod_d[T.12.0]	0.0840	0.019	4.366	0.000	0.046	0.122
sekkod_d[T.13.0]	0.0655	0.021	3.103	0.002	0.024	0.107
sekkod_d[T.14.0]	0.0503	0.012	4.230	0.000	0.027	0.074
meslek_d[T.2.0]	-0.0639	0.016	-4.038	0.000	-0.095	-0.033
meslek_d[T.3.0]	0.0167	0.014	1.221	0.222	-0.010	0.043
meslek_d[T.4.0]	-0.0096	0.014	-0.709	0.479	-0.036	0.017
meslek_d[T.5.0]	0.0002	0.012	0.017	0.987	-0.023	0.023
meslek_d[T.6.0]	0.0070	0.015	0.467	0.641	-0.022	0.036
meslek_d[T.7.0]	-0.0018	0.012	-0.151	0.880	-0.025	0.021
meslek_d[T.8.0]	0.0012	0.013	0.099	0.921	-0.023	0.026
meslek_d[T.9.0]	0.0037	0.012	0.321	0.748	-0.019	0.026
isdur_d[T.2.0]	0.0054	0.006	0.927	0.354	-0.006	0.017
isdur_d[T.3.0]	-0.0146	0.018	-0.811	0.417	-0.050	0.021
isdur_d[T.4.0]	-0.0008	0.009	-0.098	0.922	-0.018	0.016
isdur_d[T.5.0]	0.0148	0.011	1.293	0.196	-0.008	0.037
hhtipi[T.2.0]	-0.0063	0.001	-5.977	0.000	-0.008	-0.004
hhtipi[T.3.0]	-0.0094	0.001	-8.440	0.000	-0.012	-0.007
hhtipi[T.4.0]	-0.0024	0.001	-1.651	0.099	-0.005	0.000
hhtipi[T.5.0]	0.0082	0.001	7.099	0.000	0.006	0.010
hhtipi[T.6.0]	-0.0051	0.002	-2.832	0.005	-0.009	-0.002
hhtipi[T.7.0]	0.0101	0.004	2.710	0.007	0.003	0.017
konuttip[T.2.0]	-0.0082	0.001	-5.777	0.000	-0.011	-0.005
konuttip[T.3.0]	-0.0052	0.001	-4.388	0.000	-0.008	-0.003
konuttip[T.4.0]	0.0114	0.011	1.062	0.288	-0.010	0.032
mulkiyet[T.2.0]	-0.0140	0.001	-11.753	0.000	-0.016	-0.012
mulkiyet[T.3.0]	0.0173	0.003	6.296	0.000	0.012	0.023
mulkiyet[T.4.0]	-0.0127	0.001	-8.851	0.000	-0.015	-0.010
kira_sek[T.2.0]	0.0175	0.008	2.183	0.029	0.002	0.033
yapituru[T.2.0]	0.0610	0.024	2.493	0.013	0.013	0.109
yapituru[T.3.0]	-0.0057	0.003	-1.747	0.081	-0.012	0.001
yapituru[T.4.0]	0.0002	0.002	0.120	0.905	-0.003	0.004
yapituru[T.5.0]	-0.0065	0.002	-2.948	0.003	-0.011	-0.002
yapituru[T.6.0]	-0.0064	0.002	-4.071	0.000	-0.009	-0.003
yapituru[T.7.0]	0.0010	0.002	0.438	0.661	-0.003	0.006
yapituru[T.8.0]	-0.0201	0.009	-2.267	0.023	-0.037	-0.003
zemin_sal[T.2.0]	-0.0011	0.002	-0.707	0.480	-0.004	0.002
zemin_sal[T.3.0]	-0.0023	0.002	-1.342	0.180	-0.006	0.001
zemin_sal[T.4.0]	-0.0031	0.002	-1.659	0.097	-0.007	0.001
zemin_sal[T.5.0]	-0.0050	0.004	-1.423	0.155	-0.012	0.002
zemin_sal[T.6.0]	-0.0050	0.002	-2.615	0.009	-0.009	-0.001
zemin_sal[T.7.0]	0.0009	0.004	0.248	0.804	-0.006	0.008

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.8.0]	0.0343	0.011	2.986	0.003	0.012	0.057
zemin_sal[T.9.0]	-0.0052	0.004	-1.311	0.190	-0.013	0.003
zemin_sal[T.98.0]	0.0333	0.009	3.890	0.000	0.017	0.050
zemin_mut[T.2.0]	-0.0013	0.003	-0.395	0.693	-0.008	0.005
zemin_mut[T.3.0]	-0.0020	0.002	-0.846	0.397	-0.007	0.003
zemin_mut[T.4.0]	0.0081	0.004	1.946	0.052	-5.8e-05	0.016
zemin_mut[T.5.0]	-0.0010	0.009	-0.115	0.909	-0.018	0.016
zemin_mut[T.6.0]	-0.0006	0.003	-0.204	0.838	-0.007	0.005
zemin_mut[T.7.0]	-0.0075	0.004	-1.672	0.095	-0.016	0.001
zemin_mut[T.8.0]	-0.0275	0.008	-3.397	0.001	-0.043	-0.012
zemin_mut[T.9.0]	0.0247	0.005	4.557	0.000	0.014	0.035
zemin_mut[T.98.0]	-0.0469	0.010	-4.750	0.000	-0.066	-0.028
zemin_ban[T.2.0]	0.0167	0.016	1.049	0.294	-0.015	0.048
zemin_ban[T.3.0]	-0.0377	0.010	-3.724	0.000	-0.058	-0.018
zemin_ban[T.4.0]	-0.0610	0.013	-4.649	0.000	-0.087	-0.035
zemin_ban[T.5.0]	0.0552	0.037	1.506	0.132	-0.017	0.127
zemin_ban[T.6.0]	-0.0331	0.010	-3.241	0.001	-0.053	-0.013
zemin_ban[T.7.0]	-0.0380	0.011	-3.574	0.000	-0.059	-0.017
zemin_ban[T.8.0]	-0.0242	0.013	-1.898	0.058	-0.049	0.001
zemin_ban[T.9.0]	-0.0507	0.012	-4.347	0.000	-0.074	-0.028
zemin_ban[T.98.0]	0.1204	0.031	3.878	0.000	0.060	0.181
isit_sis[T.2.0]	0.0049	0.002	3.228	0.001	0.002	0.008
isit_sis[T.3.0]	0.0007	0.002	0.442	0.658	-0.003	0.004
isit_sis[T.4.0]	0.0192	0.003	6.182	0.000	0.013	0.025
yakit_1[T.2.0]	-0.0055	0.001	-4.460	0.000	-0.008	-0.003
yakit_1[T.3.0]	-0.0013	0.002	-0.748	0.455	-0.005	0.002
yakit_1[T.4.0]	0.0023	0.002	1.182	0.237	-0.002	0.006
yakit_1[T.5.0]	-0.0082	0.002	-3.637	0.000	-0.013	-0.004
yakit_1[T.6.0]	-0.0086	0.002	-4.145	0.000	-0.013	-0.005
yakit_2[T.2.0]	-0.0058	0.002	-2.669	0.008	-0.010	-0.002
yakit_2[T.3.0]	-0.0511	0.011	-4.599	0.000	-0.073	-0.029
yakit_2[T.4.0]	-0.0272	0.004	-6.954	0.000	-0.035	-0.020
yakit_2[T.5.0]	-0.0072	0.003	-2.503	0.012	-0.013	-0.002
yakit_2[T.6.0]	-0.0035	0.004	-0.920	0.358	-0.011	0.004
yakit_3[T.2.0]	-0.4098	0.014	-29.484	0.000	-0.437	-0.383
yakit_3[T.3.0]	-0.0588	0.029	-2.028	0.043	-0.116	-0.002
yakit_3[T.4.0]	-0.4412	0.013	-34.267	0.000	-0.466	-0.416
yakit_3[T.5.0]	-0.3957	0.012	-32.031	0.000	-0.420	-0.372
yakit_3[T.6.0]	-0.4464	0.013	-34.623	0.000	-0.472	-0.421
yakit_mut[T.2.0]	0.0220	0.007	3.188	0.001	0.008	0.036
yakit_mut[T.3.0]	-0.0048	0.004	-1.157	0.247	-0.013	0.003
yakit_mut[T.4.0]	-0.0016	0.003	-0.605	0.545	-0.007	0.004
yakit_mut[T.5.0]	0.0040	0.004	0.934	0.350	-0.004	0.012
yakit_mut[T.6.0]	0.0451	0.005	9.315	0.000	0.036	0.055
yakit_mut[T.7.0]	0.0622	0.023	2.733	0.006	0.018	0.107

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.8.0]	0.0571	0.013	4.287	0.000	0.031	0.083
yakit_su[T.2.0]	-0.0053	0.004	-1.305	0.192	-0.013	0.003
yakit_su[T.3.0]	-0.0007	0.004	-0.194	0.846	-0.008	0.007
yakit_su[T.4.0]	-0.0062	0.003	-2.438	0.015	-0.011	-0.001
yakit_su[T.5.0]	-0.0014	0.002	-0.664	0.507	-0.006	0.003
yakit_su[T.6.0]	-0.0078	0.003	-3.022	0.003	-0.013	-0.003
yakit_su[T.7.0]	-0.0031	0.002	-1.377	0.169	-0.008	0.001
yakit_su[T.8.0]	0.0375	0.008	4.931	0.000	0.023	0.052
zor_egiti[T.2.0]	-0.0084	0.002	-3.539	0.000	-0.013	-0.004
zor_egiti[T.3.0]	-0.0016	0.002	-0.714	0.475	-0.006	0.003
zor_egiti[T.4.0]	0.0010	0.002	0.417	0.677	-0.004	0.006
banyo[T.1.0]	0.0029	0.002	1.427	0.154	-0.001	0.007
sauna[T.1.0]	0.0144	0.011	1.313	0.189	-0.007	0.036
jakuzi[T.1.0]	-0.0162	0.005	-3.334	0.001	-0.026	-0.007
tuvalet[T.1.0]	-0.0093	0.001	-7.488	0.000	-0.012	-0.007
mutfak[T.1.0]	0.0081	0.002	3.355	0.001	0.003	0.013
copogut[T.1.0]	-0.0075	0.006	-1.236	0.216	-0.019	0.004
tabanisi[T.1.0]	0.0229	0.005	4.672	0.000	0.013	0.033
su_sis[T.1.0]	0.0168	0.002	8.871	0.000	0.013	0.020
elektri[T.1.0]	-0.0143	0.020	-0.727	0.467	-0.053	0.024
sicaksu[T.1.0]	0.0045	0.001	4.855	0.000	0.003	0.006
kabloyay[T.1.0]	0.0082	0.002	4.890	0.000	0.005	0.011
asansor[T.1.0]	0.0036	0.001	3.022	0.003	0.001	0.006
garaj[T.1.0]	-0.0013	0.002	-0.858	0.391	-0.004	0.002
havuz[T.1.0]	0.0301	0.005	6.231	0.000	0.021	0.040
jenrator[T.1.0]	-0.0021	0.004	-0.571	0.568	-0.009	0.005
apartman[T.1.0]	0.0126	0.001	11.204	0.000	0.010	0.015
yazlik[T.1.0]	0.0110	0.002	5.044	0.000	0.007	0.015
tarla[T.1.0]	0.0152	0.001	16.246	0.000	0.013	0.017
bag[T.1.0]	0.0073	0.001	6.385	0.000	0.005	0.010
sera[T.1.0]	0.0160	0.004	4.358	0.000	0.009	0.023
arsa[T.1.0]	0.0069	0.002	4.478	0.000	0.004	0.010
dukkan[T.1.0]	0.0316	0.001	22.050	0.000	0.029	0.034
otel[T.1.0]	0.0454	0.013	3.593	0.000	0.021	0.070
diger[T.1.0]	0.0531	0.006	9.503	0.000	0.042	0.064
kirkntkd[T.2.0]	-0.0137	0.001	-14.735	0.000	-0.016	-0.012
yas	0.0015	0.000	6.882	0.000	0.001	0.002
ailegrup	1.593e-05	2.25e-06	7.095	0.000	1.15e-05	2.03e-05
egitim	0.0009	0.000	4.767	0.000	0.001	0.001
sure_yil	1.083e-05	7.7e-06	1.407	0.159	-4.26e-06	2.59e-05
kac_kisi	-7.468e-06	2.23e-06	-3.350	0.001	-1.18e-05	-3.1e-06
ucrn_ay	-4.551e-06	1.97e-06	-2.307	0.021	-8.42e-06	-6.85e-07
ucra_ay	-9.398e-05	1.51e-05	-6.231	0.000	-0.000	-6.44e-05
ucrn_yil	4.402e-07	1.28e-07	3.439	0.001	1.89e-07	6.91e-07
ikr_yil	3.965e-06	6.59e-07	6.012	0.000	2.67e-06	5.26e-06

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
prim_yil	3.722e-07	5.11e-07	0.728	0.466	-6.29e-07	1.37e-06
konayni	-2.38e-06	2e-06	-1.188	0.235	-6.31e-06	1.55e-06
ulsayni	5.526e-07	1.85e-06	0.299	0.765	-3.07e-06	4.17e-06
egsayni	1.506e-06	4.93e-06	0.305	0.760	-8.16e-06	1.12e-05
yemayni	2.122e-06	1.65e-06	1.288	0.198	-1.11e-06	5.35e-06
kresayni	-7.206e-07	9.34e-06	-0.077	0.938	-1.9e-05	1.76e-05
giyayni	2.466e-05	9.28e-06	2.657	0.008	6.47e-06	4.28e-05
gidaayni	4.084e-06	3.7e-06	1.103	0.270	-3.17e-06	1.13e-05
digayni	-3.926e-06	3.26e-06	-1.202	0.229	-1.03e-05	2.47e-06
calsur_e	0.0007	0.000	4.941	0.000	0.000	0.001
ekgla_ay	6.875e-06	2.09e-05	0.329	0.742	-3.41e-05	4.78e-05
ekgln_yl	-2.741e-06	2.95e-07	-9.302	0.000	-3.32e-06	-2.16e-06
ekgla_yl	-3.922e-07	6.06e-06	-0.065	0.948	-1.23e-05	1.15e-05
calay_yil	0.0005	0.000	3.617	0.000	0.000	0.001
digln_yl	3.184e-06	5.11e-07	6.235	0.000	2.18e-06	4.18e-06
digla_yl	1.371e-05	3.96e-06	3.458	0.001	5.94e-06	2.15e-05
gmkn_ay	7.478e-06	3.18e-06	2.350	0.019	1.24e-06	1.37e-05
gmka_ay	0.0004	0.000	1.465	0.143	-0.000	0.001
gmkn_yl	-1.399e-06	2.11e-07	-6.629	0.000	-1.81e-06	-9.86e-07
gmka_yl	6.653e-07	3.39e-07	1.961	0.050	4.56e-10	1.33e-06
mulka_ay	-0.0007	0.000	-4.794	0.000	-0.001	-0.000
banka_yl	-1.015e-06	2.39e-07	-4.240	0.000	-1.48e-06	-5.46e-07
doviz_yl	-1.242e-06	1.24e-06	-1.000	0.317	-3.67e-06	1.19e-06
nema_yl	-1.083e-06	3.85e-07	-2.810	0.005	-1.84e-06	-3.28e-07
temet_yl	-2.157e-06	6.03e-07	-3.575	0.000	-3.34e-06	-9.74e-07
kar_yl	-2.786e-07	8.57e-07	-0.325	0.745	-1.96e-06	1.4e-06
toprn_yl	8.068e-07	7.09e-07	1.138	0.255	-5.82e-07	2.2e-06
topra_yl	-1.269e-05	5.48e-06	-2.317	0.020	-2.34e-05	-1.96e-06
trsn_ay	-1.156e-05	2.3e-06	-5.018	0.000	-1.61e-05	-7.04e-06
emekl_yl	8.959e-07	2.11e-07	4.238	0.000	4.82e-07	1.31e-06
viade_yl	4.334e-06	3.89e-06	1.115	0.265	-3.29e-06	1.2e-05
yasli_yl	9.95e-06	1.91e-06	5.223	0.000	6.22e-06	1.37e-05
sosy_yl	1.753e-06	1.86e-06	0.941	0.347	-1.9e-06	5.4e-06
hasta_yl	-7.271e-07	2.09e-06	-0.349	0.727	-4.82e-06	3.36e-06
dul_yl	2.065e-06	4.39e-07	4.706	0.000	1.21e-06	2.93e-06
gazi_yl	4.083e-06	1.06e-06	3.836	0.000	2e-06	6.17e-06
burs_yl	1.002e-06	1.02e-06	0.979	0.328	-1e-06	3.01e-06
issiz_yl	8.552e-06	2.9e-06	2.952	0.003	2.87e-06	1.42e-05
ddestek_yl	1.185e-06	8.47e-07	1.398	0.162	-4.76e-07	2.85e-06
dayni_yl	2.809e-05	3.22e-06	8.720	0.000	2.18e-05	3.44e-05
ydemk_yl	1.555e-06	4.86e-07	3.199	0.001	6.02e-07	2.51e-06
yddav_yl	3.595e-07	6.21e-07	0.579	0.563	-8.58e-07	1.58e-06
ydayn_yl	7.814e-06	7.72e-06	1.012	0.312	-7.33e-06	2.3e-05
dgnfk_yl	-2.092e-06	3.62e-07	-5.785	0.000	-2.8e-06	-1.38e-06
dgbrs_yl	-1.973e-06	7.19e-07	-2.745	0.006	-3.38e-06	-5.64e-07

Table C.7 (Continued) : Quantile Regression Coefficients for 20th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
dsemk_yl	1.302e-09	1.07e-06	0.001	0.999	-2.09e-06	2.09e-06
dgayn_yl	1.18e-05	1.44e-06	8.173	0.000	8.97e-06	1.46e-05
topga_ay	1.055e-05	8.57e-06	1.231	0.218	-6.25e-06	2.73e-05
topgelay	1.723e-07	3.45e-07	0.499	0.618	-5.04e-07	8.49e-07
topga_yl	2.901e-07	1.02e-06	0.284	0.776	-1.71e-06	2.29e-06
topgelyl	2.539e-06	5.9e-08	43.054	0.000	2.42e-06	2.65e-06
indeks_arl	0.3117	0.010	31.769	0.000	0.292	0.331
kira_ay	-3.064e-05	4.07e-06	-7.521	0.000	-3.86e-05	-2.27e-05
loj_fert	0.0944	0.009	10.516	0.000	0.077	0.112
izafi_k	-5.386e-05	3.35e-06	-16.098	0.000	-6.04e-05	-4.73e-05
konutsah	5.149e-06	1.14e-06	4.523	0.000	2.92e-06	7.38e-06
kon_sure	0.0002	0.000	1.449	0.147	-6.46e-05	0.000
kon_borc	3.569e-08	8.66e-09	4.121	0.000	1.87e-08	5.27e-08
kondeger	7.494e-08	1.19e-08	6.314	0.000	5.17e-08	9.82e-08
otur_sur	-0.0002	3.43e-05	-5.274	0.000	-0.000	-0.000

Table C.8 : Quantile Regression Results w/ FS - 20th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.05634
Model:	QuantReg	**Bandwidth:**	0.01040
Method:	Least Squares	**Sparsity:**	0.4266
Date:	Paz, 21 Tem 2019	**No. Observations:**	371613
Time:	18:10:31	**Df Residuals:**	371357

Quantile Regression Coefficients w/ FS for 20th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	0.1326	0.034	3.891	0.000	0.066	0.199
year[T.2003]	0.0031	0.001	2.178	0.029	0.000	0.006
year[T.2004]	-4.35e-05	0.002	-0.021	0.983	-0.004	0.004
year[T.2005]	-0.0014	0.002	-0.797	0.425	-0.005	0.002
year[T.2006]	0.0023	0.002	1.304	0.192	-0.001	0.006
year[T.2007]	-0.0004	0.002	-0.240	0.810	-0.004	0.003
year[T.2008]	-0.0083	0.002	-4.493	0.000	-0.012	-0.005
year[T.2009]	-0.0007	0.002	-0.293	0.770	-0.005	0.004
year[T.2010]	-0.0036	0.002	-1.572	0.116	-0.008	0.001
year[T.2011]	-0.0002	0.002	-0.079	0.937	-0.005	0.004
year[T.2012]	-0.0088	0.002	-3.852	0.000	-0.013	-0.004
year[T.2013]	-0.0047	0.002	-2.062	0.039	-0.009	-0.000
year[T.2014]	-0.0003	0.002	-0.126	0.899	-0.005	0.004
year[T.2015]	0.0059	0.002	2.586	0.010	0.001	0.010
year[T.2016]	0.0172	0.002	7.571	0.000	0.013	0.022
cinsiyet[T.2]	-0.0016	0.001	-2.059	0.039	-0.003	-7.83e-05
yakinlik[T.1.0]	0.0103	0.001	8.157	0.000	0.008	0.013
yakinlik[T.2.0]	0.0051	0.002	3.244	0.001	0.002	0.008
yakinlik[T.3.0]	0.0066	0.002	2.895	0.004	0.002	0.011
yakinlik[T.4.0]	0.0019	0.003	0.541	0.588	-0.005	0.009
yakinlik[T.5.0]	0.0089	0.005	1.743	0.081	-0.001	0.019
yakinlik[T.6.0]	0.0126	0.002	5.168	0.000	0.008	0.017
yakinlik[T.7.0]	0.0074	0.002	3.269	0.001	0.003	0.012
yakinlik[T.8.0]	0.0018	0.003	0.621	0.535	-0.004	0.008
yakinlik[T.9.0]	-0.0021	0.005	-0.419	0.675	-0.012	0.008
yakinlik[T.10.0]	-0.0368	0.008	-4.512	0.000	-0.053	-0.021
sagliks[T.2.0]	0.0105	0.002	4.831	0.000	0.006	0.015
sagliks[T.3.0]	-0.0141	0.001	-12.607	0.000	-0.016	-0.012
sagliks[T.4.0]	-0.0055	0.001	-6.221	0.000	-0.007	-0.004
ogrenci[T.2.0]	0.0025	0.001	2.050	0.040	0.000	0.005
egitim_devam[T.2]	-0.0003	0.003	-0.106	0.915	-0.005	0.005
egitim_devam[T.3]	-0.0040	0.004	-1.144	0.253	-0.011	0.003
egitim_devam[T.4]	-0.0058	0.005	-1.224	0.221	-0.015	0.003
egitim_devam[T.5]	-0.0120	0.004	-2.979	0.003	-0.020	-0.004
egitim_devam[T.6]	-0.0074	0.007	-1.136	0.256	-0.020	0.005
egitim_devam[T.7]	0.0312	0.022	1.409	0.159	-0.012	0.075
egitim_devam[T.8]	0.0864	0.049	1.751	0.080	-0.010	0.183

Table C.8 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-0.0351	0.093	-0.377	0.706	-0.218	0.147
egitim_devam[T.999]	-0.0008	0.002	-0.485	0.627	-0.004	0.003
medenidr[T.2.0]	-0.0092	0.002	-5.777	0.000	-0.012	-0.006
medenidr[T.3.0]	0.0074	0.002	3.233	0.001	0.003	0.012
medenidr[T.4.0]	0.0017	0.003	0.614	0.539	-0.004	0.007
is_arama[T.2]	0.0012	0.002	0.623	0.533	-0.003	0.005
is_arama[T.999]	0.0074	0.002	3.132	0.002	0.003	0.012
isneden[T.2.0]	0.0102	0.011	0.955	0.339	-0.011	0.031
isneden[T.3.0]	0.0144	0.011	1.352	0.176	-0.006	0.035
isneden[T.4.0]	0.0225	0.011	2.113	0.035	0.002	0.043
isneden[T.5.0]	0.0449	0.019	2.366	0.018	0.008	0.082
isneden[T.6.0]	0.0148	0.011	1.378	0.168	-0.006	0.036
isneden[T.7.0]	0.0190	0.011	1.754	0.079	-0.002	0.040
isneden[T.10.0]	0.0207	0.012	1.746	0.081	-0.003	0.044
isneden[T.11.0]	0.0054	0.011	0.498	0.618	-0.016	0.027
sekkod[T.2.0]	0.0202	0.006	3.142	0.002	0.008	0.033
sekkod[T.3.0]	-0.0016	0.003	-0.604	0.546	-0.007	0.004
sekkod[T.4.0]	-0.0015	0.004	-0.344	0.731	-0.010	0.007
sekkod[T.5.0]	-0.0058	0.003	-2.017	0.044	-0.011	-0.000
sekkod[T.6.0]	0.0029	0.002	1.181	0.237	-0.002	0.008
sekkod[T.7.0]	-0.0054	0.003	-1.746	0.081	-0.011	0.001
sekkod[T.8.0]	0.0020	0.003	0.668	0.504	-0.004	0.008
sekkod[T.9.0]	0.0019	0.005	0.396	0.692	-0.008	0.011
sekkod[T.10.0]	-0.0023	0.005	-0.513	0.608	-0.011	0.007
sekkod[T.11.0]	0.0016	0.003	0.537	0.591	-0.004	0.008
sekkod[T.12.0]	0.0054	0.003	1.609	0.108	-0.001	0.012
sekkod[T.13.0]	-0.0005	0.004	-0.149	0.882	-0.007	0.006
sekkod[T.14.0]	-0.0008	0.003	-0.260	0.795	-0.007	0.005
sgk[T.2.0]	0.0256	0.010	2.460	0.014	0.005	0.046
sgk[T.3.0]	-0.0065	0.002	-4.115	0.000	-0.010	-0.003
is_statu[T.2]	0.0040	0.003	1.572	0.116	-0.001	0.009
is_statu[T.3]	0.0202	0.006	3.308	0.001	0.008	0.032
is_statu[T.999]	-0.0022	0.002	-1.200	0.230	-0.006	0.001
koopera[T.2]	-0.0075	0.004	-1.797	0.072	-0.016	0.001
koopera[T.999]	-0.0058	0.004	-1.451	0.147	-0.014	0.002
sir_kon[T.2]	-0.0091	0.006	-1.468	0.142	-0.021	0.003
sir_kon[T.999]	-0.0003	0.006	-0.049	0.961	-0.012	0.011
isdur[T.2.0]	0.0069	0.003	2.611	0.009	0.002	0.012
isdur[T.3.0]	0.0108	0.003	3.682	0.000	0.005	0.017
isdur[T.4.0]	0.0113	0.002	5.046	0.000	0.007	0.016
isdur[T.5.0]	0.0065	0.003	2.342	0.019	0.001	0.012
ist_tesk[T.2.0]	-0.0069	0.003	-2.267	0.023	-0.013	-0.001
ist_tesk[T.3.0]	0.0011	0.003	0.332	0.740	-0.005	0.007
guc_kul[T.2]	-0.0015	0.002	-0.633	0.526	-0.006	0.003
guc_kul[T.3]	-0.0006	0.002	-0.309	0.757	-0.004	0.003

Table C.8 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.999]	0.0027	0.002	1.255	0.210	-0.002	0.007
isfaal[T.2.0]	-0.0060	0.003	-2.064	0.039	-0.012	-0.000
mulk_dur[T.2.0]	-0.0115	0.003	-3.892	0.000	-0.017	-0.006
mulk_dur[T.3.0]	-0.0097	0.004	-2.422	0.015	-0.017	-0.002
ekis_yl[T.2.0]	-0.0064	0.004	-1.599	0.110	-0.014	0.001
ekis_ay[T.2.0]	0.0135	0.005	2.898	0.004	0.004	0.023
sekkod_e[T.2.0]	0.0672	0.049	1.375	0.169	-0.029	0.163
sekkod_e[T.3.0]	0.0702	0.013	5.577	0.000	0.046	0.095
sekkod_e[T.4.0]	0.2052	0.040	5.086	0.000	0.126	0.284
sekkod_e[T.5.0]	0.0299	0.011	2.802	0.005	0.009	0.051
sekkod_e[T.6.0]	0.1013	0.011	9.573	0.000	0.081	0.122
sekkod_e[T.7.0]	0.0723	0.014	5.255	0.000	0.045	0.099
sekkod_e[T.8.0]	0.0734	0.014	5.371	0.000	0.047	0.100
sekkod_e[T.9.0]	0.1802	0.038	4.696	0.000	0.105	0.255
sekkod_e[T.10.0]	0.1378	0.027	5.032	0.000	0.084	0.191
sekkod_e[T.11.0]	0.0950	0.015	6.376	0.000	0.066	0.124
sekkod_e[T.12.0]	0.1149	0.018	6.222	0.000	0.079	0.151
sekkod_e[T.13.0]	0.1235	0.024	5.152	0.000	0.077	0.171
sekkod_e[T.14.0]	0.0505	0.011	4.666	0.000	0.029	0.072
meslek_e[T.1.0]	0.0326	0.009	3.565	0.000	0.015	0.051
meslek_e[T.2.0]	-0.0264	0.012	-2.175	0.030	-0.050	-0.003
meslek_e[T.3.0]	0.0144	0.013	1.133	0.257	-0.011	0.039
meslek_e[T.4.0]	-0.2119	0.014	-14.897	0.000	-0.240	-0.184
meslek_e[T.5.0]	0.0020	0.011	0.190	0.849	-0.019	0.023
meslek_e[T.6.0]	0.1003	0.010	10.154	0.000	0.081	0.120
meslek_e[T.7.0]	0.0255	0.010	2.614	0.009	0.006	0.045
meslek_e[T.8.0]	0.0175	0.011	1.635	0.102	-0.003	0.039
meslek_e[T.9.0]	0.0454	0.008	5.479	0.000	0.029	0.062
isdur_e[T.2.0]	0.0301	0.008	3.557	0.000	0.014	0.047
isdur_e[T.3.0]	0.0043	0.012	0.356	0.721	-0.020	0.028
isdur_e[T.4.0]	0.0040	0.008	0.488	0.625	-0.012	0.020
isdur_e[T.5.0]	-0.0052	0.010	-0.521	0.602	-0.025	0.014
cl_neden[T.2.0]	-0.0045	0.006	-0.772	0.440	-0.016	0.007
cl_neden[T.3.0]	0.0139	0.003	4.405	0.000	0.008	0.020
cl_neden[T.4.0]	0.0129	0.006	2.100	0.036	0.001	0.025
cl_neden[T.5.0]	0.0289	0.007	4.372	0.000	0.016	0.042
cl_neden[T.6.0]	0.0165	0.005	3.559	0.000	0.007	0.026
cl_neden[T.7.0]	-0.0001	0.005	-0.027	0.978	-0.010	0.010
ayni_is[T.2.0]	-0.0117	0.006	-1.838	0.066	-0.024	0.001
sekkod_d[T.2.0]	0.0203	0.025	0.823	0.410	-0.028	0.069
sekkod_d[T.3.0]	0.0288	0.009	3.380	0.001	0.012	0.045
sekkod_d[T.4.0]	0.0376	0.014	2.630	0.009	0.010	0.066
sekkod_d[T.5.0]	0.0106	0.008	1.361	0.173	-0.005	0.026
sekkod_d[T.6.0]	0.0179	0.009	1.977	0.048	0.000	0.036
sekkod_d[T.7.0]	0.0104	0.010	0.997	0.319	-0.010	0.031

Table C.8 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.8.0]	0.0227	0.010	2.267	0.023	0.003	0.042
sekkod_d[T.9.0]	0.0201	0.021	0.969	0.333	-0.021	0.061
sekkod_d[T.10.0]	0.0081	0.016	0.503	0.615	-0.024	0.040
sekkod_d[T.11.0]	0.0255	0.012	2.092	0.036	0.002	0.049
sekkod_d[T.12.0]	0.0377	0.016	2.335	0.020	0.006	0.069
sekkod_d[T.13.0]	0.0213	0.017	1.220	0.223	-0.013	0.056
sekkod_d[T.14.0]	0.0248	0.010	2.398	0.016	0.005	0.045
meslek_d[T.2.0]	-0.0536	0.014	-3.728	0.000	-0.082	-0.025
meslek_d[T.3.0]	-0.0201	0.012	-1.629	0.103	-0.044	0.004
meslek_d[T.4.0]	-0.0332	0.012	-2.694	0.007	-0.057	-0.009
meslek_d[T.5.0]	-0.0306	0.011	-2.905	0.004	-0.051	-0.010
meslek_d[T.6.0]	-0.0153	0.014	-1.098	0.272	-0.043	0.012
meslek_d[T.7.0]	-0.0464	0.010	-4.436	0.000	-0.067	-0.026
meslek_d[T.8.0]	-0.0405	0.011	-3.584	0.000	-0.063	-0.018
meslek_d[T.9.0]	-0.0332	0.011	-3.102	0.002	-0.054	-0.012
isdur_d[T.2.0]	-0.0036	0.005	-0.706	0.480	-0.013	0.006
isdur_d[T.3.0]	0.0154	0.017	0.906	0.365	-0.018	0.049
isdur_d[T.4.0]	-0.0118	0.008	-1.525	0.127	-0.027	0.003
isdur_d[T.5.0]	-0.0073	0.010	-0.753	0.452	-0.026	0.012
hhtipi[T.2.0]	-0.0067	0.001	-7.049	0.000	-0.009	-0.005
hhtipi[T.3.0]	-0.0099	0.001	-9.844	0.000	-0.012	-0.008
hhtipi[T.4.0]	0.0057	0.001	4.408	0.000	0.003	0.008
hhtipi[T.5.0]	-0.0015	0.001	-1.476	0.140	-0.004	0.001
hhtipi[T.6.0]	0.0021	0.002	1.288	0.198	-0.001	0.005
hhtipi[T.7.0]	0.0020	0.003	0.590	0.555	-0.005	0.008
konuttip[T.2.0]	-0.0028	0.001	-2.172	0.030	-0.005	-0.000
konuttip[T.3.0]	-0.0023	0.001	-2.177	0.029	-0.004	-0.000
konuttip[T.4.0]	0.0289	0.010	2.994	0.003	0.010	0.048
mulkiyet[T.2.0]	-0.0130	0.001	-13.248	0.000	-0.015	-0.011
mulkiyet[T.3.0]	0.0206	0.002	9.301	0.000	0.016	0.025
mulkiyet[T.4.0]	-0.0035	0.001	-3.104	0.002	-0.006	-0.001
kira_sek[T.2.0]	0.0088	0.007	1.169	0.242	-0.006	0.023
yapituru[T.2.0]	0.0297	0.023	1.271	0.204	-0.016	0.075
yapituru[T.3.0]	-0.0020	0.003	-0.693	0.488	-0.008	0.004
yapituru[T.4.0]	-0.0035	0.002	-2.097	0.036	-0.007	-0.000
yapituru[T.5.0]	-0.0069	0.002	-3.502	0.000	-0.011	-0.003
yapituru[T.6.0]	-0.0056	0.001	-4.020	0.000	-0.008	-0.003
yapituru[T.7.0]	-0.0045	0.002	-2.217	0.027	-0.009	-0.001
yapituru[T.8.0]	-0.0224	0.008	-2.762	0.006	-0.038	-0.007
zemin_sal[T.2.0]	0.0051	0.001	3.653	0.000	0.002	0.008
zemin_sal[T.3.0]	0.0030	0.002	1.947	0.052	-2e-05	0.006
zemin_sal[T.4.0]	0.0012	0.002	0.687	0.492	-0.002	0.004
zemin_sal[T.5.0]	-0.0014	0.003	-0.443	0.658	-0.008	0.005
zemin_sal[T.6.0]	0.0021	0.002	1.210	0.226	-0.001	0.005
zemin_sal[T.7.0]	0.0103	0.003	3.141	0.002	0.004	0.017

Table C.8 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.8.0]	0.0070	0.010	0.716	0.474	-0.012	0.026
zemin_sal[T.9.0]	0.0032	0.004	0.885	0.376	-0.004	0.010
zemin_sal[T.98.0]	-0.0021	0.007	-0.285	0.776	-0.016	0.012
zemin_mut[T.2.0]	0.0047	0.003	1.514	0.130	-0.001	0.011
zemin_mut[T.3.0]	0.0020	0.002	0.960	0.337	-0.002	0.006
zemin_mut[T.4.0]	0.0047	0.004	1.248	0.212	-0.003	0.012
zemin_mut[T.5.0]	-0.0087	0.008	-1.141	0.254	-0.024	0.006
zemin_mut[T.6.0]	0.0035	0.003	1.292	0.196	-0.002	0.009
zemin_mut[T.7.0]	-0.0010	0.004	-0.238	0.812	-0.009	0.007
zemin_mut[T.8.0]	-0.0085	0.008	-1.099	0.272	-0.024	0.007
zemin_mut[T.9.0]	0.0099	0.005	2.051	0.040	0.000	0.019
zemin_mut[T.98.0]	-0.0025	0.009	-0.288	0.773	-0.019	0.014
zemin_ban[T.2.0]	-0.0934	0.020	-4.628	0.000	-0.133	-0.054
zemin_ban[T.3.0]	-0.1007	0.017	-5.840	0.000	-0.134	-0.067
zemin_ban[T.4.0]	-0.1037	0.019	-5.474	0.000	-0.141	-0.067
zemin_ban[T.5.0]	0.1560	0.065	2.406	0.016	0.029	0.283
zemin_ban[T.6.0]	-0.0962	0.017	-5.566	0.000	-0.130	-0.062
zemin_ban[T.7.0]	-0.1012	0.017	-5.783	0.000	-0.135	-0.067
zemin_ban[T.8.0]	-0.1113	0.019	-5.934	0.000	-0.148	-0.075
zemin_ban[T.9.0]	-0.0916	0.018	-5.063	0.000	-0.127	-0.056
zemin_ban[T.98.0]	-0.0101	0.026	-0.384	0.701	-0.062	0.042
isit_sis[T.2.0]	-0.0051	0.001	-3.742	0.000	-0.008	-0.002
isit_sis[T.3.0]	-0.0049	0.001	-3.297	0.001	-0.008	-0.002
isit_sis[T.4.0]	0.0019	0.003	0.686	0.493	-0.004	0.007
yakit_1[T.2.0]	-0.0003	0.001	-0.257	0.797	-0.002	0.002
yakit_1[T.3.0]	0.0014	0.002	0.903	0.367	-0.002	0.005
yakit_1[T.4.0]	-0.0003	0.002	-0.156	0.876	-0.004	0.003
yakit_1[T.5.0]	0.0013	0.002	0.624	0.533	-0.003	0.005
yakit_1[T.6.0]	-0.0106	0.002	-5.698	0.000	-0.014	-0.007
yakit_2[T.2.0]	-0.0020	0.002	-1.015	0.310	-0.006	0.002
yakit_2[T.3.0]	-0.0174	0.012	-1.511	0.131	-0.040	0.005
yakit_2[T.4.0]	-0.0148	0.004	-4.168	0.000	-0.022	-0.008
yakit_2[T.5.0]	0.0024	0.003	0.908	0.364	-0.003	0.007
yakit_2[T.6.0]	0.0025	0.003	0.750	0.453	-0.004	0.009
yakit_3[T.2.0]	0.0095	0.010	0.968	0.333	-0.010	0.029
yakit_3[T.3.0]	-0.0139	0.017	-0.830	0.407	-0.047	0.019
yakit_3[T.4.0]	-0.0115	0.008	-1.392	0.164	-0.028	0.005
yakit_3[T.5.0]	-0.0109	0.007	-1.475	0.140	-0.025	0.004
yakit_3[T.6.0]	-0.0098	0.008	-1.185	0.236	-0.026	0.006
yakit_mut[T.2.0]	0.0018	0.006	0.284	0.777	-0.010	0.014
yakit_mut[T.3.0]	3.444e-05	0.004	0.009	0.993	-0.007	0.008
yakit_mut[T.4.0]	-0.0084	0.002	-3.462	0.001	-0.013	-0.004
yakit_mut[T.5.0]	-0.0063	0.004	-1.621	0.105	-0.014	0.001
yakit_mut[T.6.0]	0.0314	0.004	7.137	0.000	0.023	0.040
yakit_mut[T.7.0]	0.0617	0.022	2.867	0.004	0.020	0.104

Table C.8 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.8.0]	0.0622	0.012	4.980	0.000	0.038	0.087
yakit_su[T.2.0]	-0.0126	0.004	-3.427	0.001	-0.020	-0.005
yakit_su[T.3.0]	-0.0169	0.003	-4.888	0.000	-0.024	-0.010
yakit_su[T.4.0]	-0.0093	0.002	-4.021	0.000	-0.014	-0.005
yakit_su[T.5.0]	-0.0053	0.002	-2.787	0.005	-0.009	-0.002
yakit_su[T.6.0]	-0.0092	0.002	-4.016	0.000	-0.014	-0.005
yakit_su[T.7.0]	-0.0078	0.002	-3.763	0.000	-0.012	-0.004
yakit_su[T.8.0]	0.0111	0.007	1.703	0.088	-0.002	0.024
zor_egiti[T.2.0]	-0.0100	0.002	-4.701	0.000	-0.014	-0.006
zor_egiti[T.3.0]	-0.0017	0.002	-0.835	0.404	-0.006	0.002
zor_egiti[T.4.0]	-0.0023	0.002	-1.057	0.291	-0.007	0.002
banyo[T.1.0]	0.0019	0.002	1.067	0.286	-0.002	0.005
sauna[T.1.0]	-0.0517	0.010	-5.404	0.000	-0.070	-0.033
jakuzi[T.1.0]	-0.0488	0.004	-11.053	0.000	-0.057	-0.040
tuvalet[T.1.0]	-0.0081	0.001	-7.221	0.000	-0.010	-0.006
mutfak[T.1.0]	0.0052	0.002	2.386	0.017	0.001	0.010
copogut[T.1.0]	-0.0371	0.005	-6.835	0.000	-0.048	-0.026
tabanisi[T.1.0]	-0.0043	0.004	-0.984	0.325	-0.013	0.004
su_sis[T.1.0]	0.0181	0.002	10.618	0.000	0.015	0.021
elektri[T.1.0]	-0.0061	0.019	-0.323	0.747	-0.043	0.031
sicaksu[T.1.0]	0.0001	0.001	0.139	0.890	-0.002	0.002
kabloyay[T.1.0]	-0.0024	0.001	-1.613	0.107	-0.005	0.001
asansor[T.1.0]	0.0010	0.001	0.919	0.358	-0.001	0.003
garaj[T.1.0]	-0.0082	0.001	-5.838	0.000	-0.011	-0.005
havuz[T.1.0]	-0.0457	0.004	-10.898	0.000	-0.054	-0.037
jenrator[T.1.0]	-0.0200	0.003	-6.159	0.000	-0.026	-0.014
apartman[T.1.0]	0.0108	0.001	10.609	0.000	0.009	0.013
yazlik[T.1.0]	-0.0063	0.002	-3.223	0.001	-0.010	-0.002
tarla[T.1.0]	0.0052	0.001	6.099	0.000	0.003	0.007
bag[T.1.0]	5.25e-05	0.001	0.051	0.960	-0.002	0.002
sera[T.1.0]	0.0121	0.003	3.632	0.000	0.006	0.019
arsa[T.1.0]	0.0019	0.001	1.326	0.185	-0.001	0.005
dukkan[T.1.0]	0.0095	0.001	7.376	0.000	0.007	0.012
otel[T.1.0]	-0.0478	0.011	-4.407	0.000	-0.069	-0.027
diger[T.1.0]	0.0317	0.005	6.332	0.000	0.022	0.042
kirkntkd[T.2.0]	-0.0085	0.001	-10.123	0.000	-0.010	-0.007
izafi_k	-0.0001	2.18e-06	-52.817	0.000	-0.000	-0.000
ceptel	-0.0066	0.000	-22.063	0.000	-0.007	-0.006
yilkulgel	2.87e-06	2.07e-08	138.885	0.000	2.83e-06	2.91e-06

Table C.9 : Quantile Regression Results - 40th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.09311
Model:	QuantReg	**Bandwidth:**	0.01145
Method:	Least Squares	**Sparsity:**	0.4157
Date:	Cts, 23 Haz 2018	**No. Observations:**	371613
Time:	20:27:30	**Df Residuals:**	371226

Quantile Regression Coefficients for 40th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	0.2010	0.033	6.084	0.000	0.136	0.266
year[T.2003]	0.0113	0.002	5.293	0.000	0.007	0.016
year[T.2004]	-0.0045	0.003	-1.502	0.133	-0.010	0.001
year[T.2005]	0.0012	0.003	0.467	0.641	-0.004	0.006
year[T.2006]	0.0041	0.003	1.579	0.114	-0.001	0.009
year[T.2007]	-0.0152	0.003	-5.682	0.000	-0.020	-0.010
year[T.2008]	-0.0154	0.003	-6.117	0.000	-0.020	-0.010
year[T.2009]	-0.0059	0.002	-3.189	0.001	-0.010	-0.002
year[T.2010]	-0.0124	0.002	-6.796	0.000	-0.016	-0.009
year[T.2011]	-0.0085	0.002	-4.664	0.000	-0.012	-0.005
year[T.2012]	-0.0124	0.002	-7.155	0.000	-0.016	-0.009
year[T.2013]	-0.0119	0.002	-6.570	0.000	-0.015	-0.008
year[T.2014]	0.0059	0.002	3.216	0.001	0.002	0.010
year[T.2015]	0.0250	0.002	11.208	0.000	0.021	0.029
year[T.2016]	0.0310	0.002	14.038	0.000	0.027	0.035
cinsiyet[T.2]	-0.0013	0.001	-1.227	0.220	-0.003	0.001
yakinlik[T.1.0]	0.0555	0.002	30.430	0.000	0.052	0.059
yakinlik[T.2.0]	0.0601	0.002	25.686	0.000	0.056	0.065
yakinlik[T.3.0]	0.0383	0.003	11.890	0.000	0.032	0.045
yakinlik[T.4.0]	0.0608	0.005	12.754	0.000	0.051	0.070
yakinlik[T.5.0]	0.0576	0.007	8.171	0.000	0.044	0.071
yakinlik[T.6.0]	0.0740	0.004	21.106	0.000	0.067	0.081
yakinlik[T.7.0]	0.0749	0.003	22.554	0.000	0.068	0.081
yakinlik[T.8.0]	0.0697	0.004	16.631	0.000	0.062	0.078
yakinlik[T.9.0]	0.0628	0.007	9.094	0.000	0.049	0.076
yakinlik[T.10.0]	0.0438	0.011	3.880	0.000	0.022	0.066
sagliks[T.2.0]	0.0414	0.003	13.721	0.000	0.035	0.047
sagliks[T.3.0]	-0.0410	0.002	-26.181	0.000	-0.044	-0.038
sagliks[T.4.0]	-0.0129	0.001	-10.348	0.000	-0.015	-0.010
ogrenci[T.2.0]	0.0036	0.002	2.127	0.033	0.000	0.007
egitim_devam[T.2]	-0.0020	0.004	-0.531	0.595	-0.009	0.005
egitim_devam[T.3]	-0.0102	0.005	-2.092	0.036	-0.020	-0.001
egitim_devam[T.4]	-0.0118	0.007	-1.794	0.073	-0.025	0.001
egitim_devam[T.5]	-0.0223	0.006	-3.986	0.000	-0.033	-0.011
egitim_devam[T.6]	0.0200	0.009	2.201	0.028	0.002	0.038
egitim_devam[T.7]	-0.0469	0.030	-1.558	0.119	-0.106	0.012
egitim_devam[T.8]	0.0170	0.061	0.279	0.780	-0.102	0.137

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-0.0200	0.118	-0.170	0.865	-0.251	0.211
egitim_devam[T.999]	-0.0029	0.002	-1.215	0.224	-0.008	0.002
medenidr[T.2.0]	-0.0223	0.002	-9.739	0.000	-0.027	-0.018
medenidr[T.3.0]	-0.0017	0.004	-0.462	0.644	-0.009	0.005
medenidr[T.4.0]	-0.0062	0.004	-1.573	0.116	-0.014	0.002
is_arama[T.2]	0.0122	0.003	4.364	0.000	0.007	0.018
is_arama[T.999]	0.0314	0.004	8.806	0.000	0.024	0.038
isneden[T.2.0]	0.1317	0.014	9.589	0.000	0.105	0.159
isneden[T.3.0]	0.1373	0.014	10.056	0.000	0.111	0.164
isneden[T.4.0]	0.1306	0.014	9.432	0.000	0.103	0.158
isneden[T.5.0]	0.2824	0.026	11.032	0.000	0.232	0.333
isneden[T.6.0]	0.1332	0.014	9.573	0.000	0.106	0.160
isneden[T.7.0]	0.1376	0.014	9.827	0.000	0.110	0.165
isneden[T.10.0]	0.1334	0.016	8.593	0.000	0.103	0.164
isneden[T.11.0]	0.1151	0.014	8.164	0.000	0.087	0.143
sekkod[T.2.0]	0.0340	0.009	3.815	0.000	0.017	0.051
sekkod[T.3.0]	0.0102	0.004	2.812	0.005	0.003	0.017
sekkod[T.4.0]	0.0076	0.006	1.235	0.217	-0.004	0.020
sekkod[T.5.0]	-0.0016	0.004	-0.394	0.693	-0.010	0.006
sekkod[T.6.0]	0.0121	0.003	3.480	0.001	0.005	0.019
sekkod[T.7.0]	0.0022	0.004	0.507	0.612	-0.006	0.011
sekkod[T.8.0]	0.0204	0.004	4.835	0.000	0.012	0.029
sekkod[T.9.0]	0.0165	0.007	2.456	0.014	0.003	0.030
sekkod[T.10.0]	-0.0054	0.006	-0.861	0.389	-0.018	0.007
sekkod[T.11.0]	0.0166	0.004	3.834	0.000	0.008	0.025
sekkod[T.12.0]	0.0280	0.005	5.801	0.000	0.019	0.037
sekkod[T.13.0]	0.0197	0.005	3.965	0.000	0.010	0.029
sekkod[T.14.0]	0.0129	0.004	2.938	0.003	0.004	0.021
sgk[T.2.0]	-0.0027	0.014	-0.192	0.848	-0.031	0.025
sgk[T.3.0]	-0.0048	0.002	-2.106	0.035	-0.009	-0.000
is_statu[T.2]	0.0096	0.004	2.356	0.018	0.002	0.018
is_statu[T.3]	0.0420	0.009	4.759	0.000	0.025	0.059
is_statu[T.999]	-0.0038	0.004	-1.052	0.293	-0.011	0.003
koopera[T.2]	0.0112	0.006	1.840	0.066	-0.001	0.023
koopera[T.999]	0.0064	0.006	1.147	0.252	-0.005	0.017
sir_kon[T.2]	-0.1465	0.009	-16.794	0.000	-0.164	-0.129
sir_kon[T.999]	-0.1473	0.008	-17.760	0.000	-0.164	-0.131
isdur[T.2.0]	0.0113	0.009	1.317	0.188	-0.006	0.028
isdur[T.3.0]	0.0111	0.005	2.407	0.016	0.002	0.020
isdur[T.4.0]	0.0287	0.004	8.089	0.000	0.022	0.036
isdur[T.5.0]	0.0602	0.004	14.740	0.000	0.052	0.068
ist_tesk[T.2.0]	0.0008	0.008	0.104	0.917	-0.015	0.017
ist_tesk[T.3.0]	0.0093	0.009	1.033	0.302	-0.008	0.027
guc_kul[T.2]	0.0052	0.003	1.513	0.130	-0.002	0.012
guc_kul[T.3]	0.0036	0.003	1.295	0.195	-0.002	0.009

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.999]	-0.0015	0.003	-0.491	0.624	-0.008	0.005
isfaal[T.2.0]	-0.0201	0.004	-4.919	0.000	-0.028	-0.012
mulk_dur[T.2.0]	-0.0177	0.004	-4.319	0.000	-0.026	-0.010
mulk_dur[T.3.0]	-0.0206	0.006	-3.727	0.000	-0.032	-0.010
ekis_yl[T.2.0]	-0.0174	0.006	-3.085	0.002	-0.028	-0.006
ekis_ay[T.2.0]	-0.0033	0.007	-0.496	0.620	-0.016	0.010
sekkod_e[T.2.0]	0.0590	0.078	0.760	0.447	-0.093	0.211
sekkod_e[T.3.0]	-0.0151	0.020	-0.769	0.442	-0.053	0.023
sekkod_e[T.4.0]	-0.0217	0.051	-0.422	0.673	-0.122	0.079
sekkod_e[T.5.0]	0.0022	0.016	0.137	0.891	-0.030	0.034
sekkod_e[T.6.0]	0.0201	0.017	1.184	0.236	-0.013	0.053
sekkod_e[T.7.0]	0.0218	0.021	1.035	0.301	-0.020	0.063
sekkod_e[T.8.0]	0.0525	0.022	2.440	0.015	0.010	0.095
sekkod_e[T.9.0]	-0.0051	0.048	-0.104	0.917	-0.100	0.090
sekkod_e[T.10.0]	-0.0479	0.036	-1.325	0.185	-0.119	0.023
sekkod_e[T.11.0]	0.0703	0.023	3.120	0.002	0.026	0.114
sekkod_e[T.12.0]	0.0717	0.031	2.328	0.020	0.011	0.132
sekkod_e[T.13.0]	-0.0135	0.038	-0.353	0.724	-0.089	0.062
sekkod_e[T.14.0]	-0.0036	0.017	-0.205	0.838	-0.038	0.030
meslek_e[T.1.0]	-0.0354	0.014	-2.534	0.011	-0.063	-0.008
meslek_e[T.2.0]	-0.0178	0.021	-0.844	0.399	-0.059	0.024
meslek_e[T.3.0]	0.0016	0.019	0.082	0.935	-0.036	0.039
meslek_e[T.4.0]	-0.0512	0.037	-1.391	0.164	-0.123	0.021
meslek_e[T.5.0]	0.0009	0.016	0.059	0.953	-0.030	0.032
meslek_e[T.6.0]	-0.0024	0.013	-0.189	0.850	-0.028	0.023
meslek_e[T.7.0]	0.0469	0.015	3.145	0.002	0.018	0.076
meslek_e[T.8.0]	-0.0158	0.016	-0.977	0.329	-0.048	0.016
meslek_e[T.9.0]	0.0319	0.011	2.925	0.003	0.011	0.053
isdur_e[T.2.0]	0.0023	0.012	0.186	0.853	-0.022	0.026
isdur_e[T.3.0]	0.1012	0.018	5.712	0.000	0.066	0.136
isdur_e[T.4.0]	0.0669	0.012	5.719	0.000	0.044	0.090
isdur_e[T.5.0]	0.0597	0.014	4.213	0.000	0.032	0.088
cl_neden[T.2.0]	-0.0340	0.008	-4.186	0.000	-0.050	-0.018
cl_neden[T.3.0]	0.0028	0.004	0.629	0.529	-0.006	0.012
cl_neden[T.4.0]	-0.0163	0.008	-1.930	0.054	-0.033	0.000
cl_neden[T.5.0]	0.0029	0.009	0.328	0.743	-0.015	0.020
cl_neden[T.6.0]	-0.0042	0.006	-0.656	0.512	-0.017	0.008
cl_neden[T.7.0]	-0.0133	0.007	-1.882	0.060	-0.027	0.001
ayni_is[T.2.0]	-0.0294	0.009	-3.352	0.001	-0.047	-0.012
sekkod_d[T.2.0]	-0.1115	0.039	-2.824	0.005	-0.189	-0.034
sekkod_d[T.3.0]	0.0308	0.012	2.506	0.012	0.007	0.055
sekkod_d[T.4.0]	0.0460	0.021	2.235	0.025	0.006	0.086
sekkod_d[T.5.0]	-0.0195	0.012	-1.679	0.093	-0.042	0.003
sekkod_d[T.6.0]	0.0008	0.013	0.065	0.948	-0.024	0.026
sekkod_d[T.7.0]	-0.0013	0.015	-0.086	0.931	-0.031	0.028

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.8.0]	0.0422	0.014	2.928	0.003	0.014	0.070
sekkod_d[T.9.0]	-0.0120	0.030	-0.405	0.685	-0.070	0.046
sekkod_d[T.10.0]	0.0181	0.023	0.781	0.435	-0.027	0.063
sekkod_d[T.11.0]	0.0385	0.017	2.249	0.024	0.005	0.072
sekkod_d[T.12.0]	0.0300	0.024	1.240	0.215	-0.017	0.077
sekkod_d[T.13.0]	0.0214	0.025	0.844	0.399	-0.028	0.071
sekkod_d[T.14.0]	-0.0077	0.015	-0.522	0.602	-0.037	0.021
meslek_d[T.2.0]	-0.0177	0.021	-0.850	0.395	-0.059	0.023
meslek_d[T.3.0]	-0.0051	0.017	-0.299	0.765	-0.039	0.028
meslek_d[T.4.0]	-0.0002	0.017	-0.009	0.993	-0.034	0.034
meslek_d[T.5.0]	-0.0111	0.015	-0.743	0.457	-0.040	0.018
meslek_d[T.6.0]	-0.0314	0.019	-1.643	0.100	-0.069	0.006
meslek_d[T.7.0]	-0.0149	0.015	-1.004	0.316	-0.044	0.014
meslek_d[T.8.0]	-0.0239	0.016	-1.516	0.130	-0.055	0.007
meslek_d[T.9.0]	-0.0306	0.015	-2.075	0.038	-0.059	-0.002
isdur_d[T.2.0]	0.0279	0.007	3.922	0.000	0.014	0.042
isdur_d[T.3.0]	-0.0579	0.025	-2.349	0.019	-0.106	-0.010
isdur_d[T.4.0]	-0.0103	0.011	-0.975	0.330	-0.031	0.010
isdur_d[T.5.0]	0.0203	0.014	1.408	0.159	-0.008	0.049
hhtipi[T.2.0]	-0.0091	0.001	-6.920	0.000	-0.012	-0.007
hhtipi[T.3.0]	-0.0170	0.001	-12.229	0.000	-0.020	-0.014
hhtipi[T.4.0]	-0.0057	0.002	-3.182	0.001	-0.009	-0.002
hhtipi[T.5.0]	0.0109	0.001	7.584	0.000	0.008	0.014
hhtipi[T.6.0]	-0.0116	0.002	-5.160	0.000	-0.016	-0.007
hhtipi[T.7.0]	0.0019	0.005	0.414	0.679	-0.007	0.011
konuttip[T.2.0]	-0.0069	0.002	-3.875	0.000	-0.010	-0.003
konuttip[T.3.0]	-0.0131	0.001	-8.819	0.000	-0.016	-0.010
konuttip[T.4.0]	0.0037	0.013	0.282	0.778	-0.022	0.029
mulkiyet[T.2.0]	-0.0135	0.001	-9.028	0.000	-0.016	-0.011
mulkiyet[T.3.0]	0.0324	0.003	9.463	0.000	0.026	0.039
mulkiyet[T.4.0]	-0.0133	0.002	-7.466	0.000	-0.017	-0.010
kira_sek[T.2.0]	-0.0876	0.011	-8.330	0.000	-0.108	-0.067
yapituru[T.2.0]	0.0190	0.031	0.622	0.534	-0.041	0.079
yapituru[T.3.0]	-0.0104	0.004	-2.560	0.010	-0.018	-0.002
yapituru[T.4.0]	0.0060	0.002	2.579	0.010	0.001	0.011
yapituru[T.5.0]	-0.0036	0.003	-1.291	0.197	-0.009	0.002
yapituru[T.6.0]	-0.0104	0.002	-5.334	0.000	-0.014	-0.007
yapituru[T.7.0]	0.0041	0.003	1.459	0.145	-0.001	0.010
yapituru[T.8.0]	-0.0309	0.011	-2.825	0.005	-0.052	-0.009
zemin_sal[T.2.0]	-0.0051	0.002	-2.679	0.007	-0.009	-0.001
zemin_sal[T.3.0]	-0.0100	0.002	-4.779	0.000	-0.014	-0.006
zemin_sal[T.4.0]	-0.0052	0.002	-2.268	0.023	-0.010	-0.001
zemin_sal[T.5.0]	-0.0150	0.004	-3.402	0.001	-0.024	-0.006
zemin_sal[T.6.0]	-0.0037	0.002	-1.590	0.112	-0.008	0.001
zemin_sal[T.7.0]	-0.0045	0.004	-0.996	0.319	-0.013	0.004

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.8.0]	-0.0320	0.013	-2.430	0.015	-0.058	-0.006
zemin_sal[T.9.0]	-0.0168	0.005	-3.440	0.001	-0.026	-0.007
zemin_sal[T.98.0]	0.0375	0.010	3.682	0.000	0.018	0.057
zemin_mut[T.2.0]	0.0136	0.004	3.188	0.001	0.005	0.022
zemin_mut[T.3.0]	0.0089	0.003	3.054	0.002	0.003	0.015
zemin_mut[T.4.0]	0.0194	0.005	3.747	0.000	0.009	0.030
zemin_mut[T.5.0]	0.0107	0.011	0.984	0.325	-0.011	0.032
zemin_mut[T.6.0]	0.0084	0.004	2.231	0.026	0.001	0.016
zemin_mut[T.7.0]	-0.0037	0.006	-0.661	0.509	-0.014	0.007
zemin_mut[T.8.0]	0.0219	0.010	2.105	0.035	0.002	0.042
zemin_mut[T.9.0]	0.0366	0.007	5.443	0.000	0.023	0.050
zemin_mut[T.98.0]	-0.0253	0.012	-2.100	0.036	-0.049	-0.002
zemin_ban[T.2.0]	0.0796	0.019	4.187	0.000	0.042	0.117
zemin_ban[T.3.0]	-0.0296	0.012	-2.422	0.015	-0.054	-0.006
zemin_ban[T.4.0]	-0.0731	0.016	-4.511	0.000	-0.105	-0.041
zemin_ban[T.5.0]	0.0570	0.044	1.281	0.200	-0.030	0.144
zemin_ban[T.6.0]	-0.0184	0.012	-1.494	0.135	-0.043	0.006
zemin_ban[T.7.0]	-0.0242	0.013	-1.881	0.060	-0.049	0.001
zemin_ban[T.8.0]	-0.0562	0.016	-3.582	0.000	-0.087	-0.025
zemin_ban[T.9.0]	-0.0366	0.014	-2.559	0.011	-0.065	-0.009
zemin_ban[T.98.0]	0.0180	0.034	0.523	0.601	-0.049	0.085
isit_sis[T.2.0]	0.0150	0.002	7.962	0.000	0.011	0.019
isit_sis[T.3.0]	0.0095	0.002	4.648	0.000	0.006	0.014
isit_sis[T.4.0]	0.0093	0.004	2.444	0.015	0.002	0.017
yakit_1[T.2.0]	-0.0079	0.002	-5.157	0.000	-0.011	-0.005
yakit_1[T.3.0]	-0.0143	0.002	-6.415	0.000	-0.019	-0.010
yakit_1[T.4.0]	0.0132	0.002	5.532	0.000	0.009	0.018
yakit_1[T.5.0]	0.0115	0.003	4.081	0.000	0.006	0.017
yakit_1[T.6.0]	-0.0106	0.003	-4.071	0.000	-0.016	-0.005
yakit_2[T.2.0]	-0.0092	0.003	-3.417	0.001	-0.015	-0.004
yakit_2[T.3.0]	-0.0197	0.016	-1.198	0.231	-0.052	0.013
yakit_2[T.4.0]	-0.0285	0.005	-5.834	0.000	-0.038	-0.019
yakit_2[T.5.0]	-0.0085	0.004	-2.357	0.018	-0.016	-0.001
yakit_2[T.6.0]	0.0058	0.005	1.235	0.217	-0.003	0.015
yakit_3[T.2.0]	-0.3575	0.013	-27.056	0.000	-0.383	-0.332
yakit_3[T.3.0]	-0.4352	0.022	-20.170	0.000	-0.477	-0.393
yakit_3[T.4.0]	-0.3139	0.011	-28.544	0.000	-0.335	-0.292
yakit_3[T.5.0]	-0.2810	0.010	-29.054	0.000	-0.300	-0.262
yakit_3[T.6.0]	-0.2792	0.011	-25.350	0.000	-0.301	-0.258
yakit_mut[T.2.0]	-0.0026	0.008	-0.311	0.756	-0.019	0.014
yakit_mut[T.3.0]	0.0006	0.005	0.118	0.906	-0.010	0.011
yakit_mut[T.4.0]	0.0130	0.003	3.870	0.000	0.006	0.020
yakit_mut[T.5.0]	0.0353	0.005	6.561	0.000	0.025	0.046
yakit_mut[T.6.0]	0.0826	0.006	13.579	0.000	0.071	0.095
yakit_mut[T.7.0]	0.0701	0.028	2.535	0.011	0.016	0.124

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.8.0]	0.0374	0.017	2.263	0.024	0.005	0.070
yakit_su[T.2.0]	0.0099	0.005	1.942	0.052	-9.07e-05	0.020
yakit_su[T.3.0]	0.0095	0.005	2.014	0.044	0.000	0.019
yakit_su[T.4.0]	-0.0124	0.003	-3.865	0.000	-0.019	-0.006
yakit_su[T.5.0]	-0.0055	0.003	-2.086	0.037	-0.011	-0.000
yakit_su[T.6.0]	-0.0048	0.003	-1.478	0.139	-0.011	0.002
yakit_su[T.7.0]	-0.0037	0.003	-1.295	0.195	-0.009	0.002
yakit_su[T.8.0]	0.0173	0.009	1.922	0.055	-0.000	0.035
zor_egiti[T.2.0]	-0.0144	0.003	-4.868	0.000	-0.020	-0.009
zor_egiti[T.3.0]	-0.0090	0.003	-3.246	0.001	-0.014	-0.004
zor_egiti[T.4.0]	-0.0084	0.003	-2.749	0.006	-0.014	-0.002
banyo[T.1.0]	0.0010	0.003	0.415	0.679	-0.004	0.006
sauna[T.1.0]	-0.0063	0.014	-0.468	0.640	-0.033	0.020
jakuzi[T.1.0]	0.0117	0.006	1.884	0.060	-0.000	0.024
tuvalet[T.1.0]	-0.0123	0.002	-7.902	0.000	-0.015	-0.009
mutfak[T.1.0]	0.0178	0.003	5.880	0.000	0.012	0.024
copogut[T.1.0]	0.0213	0.008	2.789	0.005	0.006	0.036
tabanisi[T.1.0]	0.0307	0.006	5.056	0.000	0.019	0.043
su_sis[T.1.0]	0.0296	0.002	12.469	0.000	0.025	0.034
elektri[T.1.0]	0.0226	0.021	1.066	0.286	-0.019	0.064
sicaksu[T.1.0]	0.0041	0.001	3.477	0.001	0.002	0.006
kabloyay[T.1.0]	0.0087	0.002	4.160	0.000	0.005	0.013
asansor[T.1.0]	0.0056	0.001	3.807	0.000	0.003	0.009
garaj[T.1.0]	0.0081	0.002	4.151	0.000	0.004	0.012
havuz[T.1.0]	0.0187	0.006	3.119	0.002	0.007	0.030
jenrator[T.1.0]	-0.0139	0.005	-3.054	0.002	-0.023	-0.005
apartman[T.1.0]	0.0227	0.001	16.033	0.000	0.020	0.025
yazlik[T.1.0]	0.0190	0.003	6.943	0.000	0.014	0.024
tarla[T.1.0]	0.0242	0.001	20.767	0.000	0.022	0.026
bag[T.1.0]	0.0095	0.001	6.647	0.000	0.007	0.012
sera[T.1.0]	0.0264	0.005	5.726	0.000	0.017	0.035
arsa[T.1.0]	0.0085	0.002	4.390	0.000	0.005	0.012
dukkan[T.1.0]	0.0546	0.002	30.557	0.000	0.051	0.058
otel[T.1.0]	0.0562	0.016	3.562	0.000	0.025	0.087
diger[T.1.0]	0.0918	0.007	13.357	0.000	0.078	0.105
kirkntkd[T.2.0]	-0.0177	0.001	-15.201	0.000	-0.020	-0.015
yas	0.0034	0.000	12.466	0.000	0.003	0.004
ailegrup	4.617e-07	2.8e-06	0.165	0.869	-5.03e-06	5.95e-06
egitim	0.0016	0.000	6.856	0.000	0.001	0.002
sure_yil	3.695e-05	9.53e-06	3.876	0.000	1.83e-05	5.56e-05
kac_kisi	-1.041e-05	2.76e-06	-3.776	0.000	-1.58e-05	-5.01e-06
ucrn_ay	-5.51e-06	2.27e-06	-2.428	0.015	-9.96e-06	-1.06e-06
ucra_ay	-0.0001	1.71e-05	-8.677	0.000	-0.000	-0.000
ucrn_yil	1.957e-07	1.4e-07	1.401	0.161	-7.8e-08	4.69e-07
ikr_yil	3.846e-06	7.31e-07	5.259	0.000	2.41e-06	5.28e-06

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
prim_yil	4.796e-07	6.1e-07	0.787	0.432	-7.15e-07	1.67e-06
konayni	3.614e-06	2.34e-06	1.544	0.123	-9.75e-07	8.2e-06
ulsayni	-1.244e-06	2.22e-06	-0.559	0.576	-5.6e-06	3.12e-06
egsayni	9.148e-07	5.5e-06	0.166	0.868	-9.86e-06	1.17e-05
yemayni	5.598e-07	2.02e-06	0.278	0.781	-3.39e-06	4.51e-06
kresayni	4.308e-06	1.12e-05	0.383	0.702	-1.77e-05	2.64e-05
giyayni	2.019e-05	1.12e-05	1.797	0.072	-1.83e-06	4.22e-05
gidaayni	3.348e-06	4.37e-06	0.766	0.444	-5.22e-06	1.19e-05
digayni	-6.017e-06	3.62e-06	-1.662	0.096	-1.31e-05	1.08e-06
calsur_e	0.0013	0.000	7.760	0.000	0.001	0.002
ekgla_ay	-7.164e-05	3.78e-05	-1.896	0.058	-0.000	2.43e-06
ekgln_yl	-1.147e-06	3.47e-07	-3.310	0.001	-1.83e-06	-4.68e-07
ekgla_yl	1.362e-05	7.39e-06	1.842	0.065	-8.73e-07	2.81e-05
calay_yil	0.0008	0.000	4.640	0.000	0.000	0.001
digln_yl	6.24e-06	5.65e-07	11.040	0.000	5.13e-06	7.35e-06
digla_yl	1.176e-05	4.48e-06	2.626	0.009	2.98e-06	2.05e-05
gmkn_ay	5.995e-06	3.7e-06	1.619	0.105	-1.26e-06	1.33e-05
gmka_ay	0.0005	0.000	1.481	0.139	-0.000	0.001
gmkn_yl	-2.569e-06	2.39e-07	-10.767	0.000	-3.04e-06	-2.1e-06
gmka_yl	4.301e-07	3.63e-07	1.186	0.236	-2.81e-07	1.14e-06
mulka_ay	0.0002	0.000	0.840	0.401	-0.000	0.001
banka_yl	-4.741e-07	2.58e-07	-1.841	0.066	-9.79e-07	3.08e-08
doviz_yl	-2.99e-06	1.47e-06	-2.038	0.042	-5.87e-06	-1.14e-07
nema_yl	-1.758e-06	3.88e-07	-4.530	0.000	-2.52e-06	-9.97e-07
temet_yl	-1.955e-06	5.75e-07	-3.398	0.001	-3.08e-06	-8.28e-07
kar_yl	-2.535e-06	1.27e-06	-1.998	0.046	-5.02e-06	-4.85e-08
toprn_yl	3.171e-07	6.37e-07	0.498	0.619	-9.31e-07	1.57e-06
topra_yl	6.22e-06	6.15e-06	1.011	0.312	-5.83e-06	1.83e-05
trsn_ay	-1.731e-05	2.81e-06	-6.167	0.000	-2.28e-05	-1.18e-05
emekl_yl	1.348e-06	2.59e-07	5.208	0.000	8.41e-07	1.86e-06
viade_yl	7.489e-06	4.9e-06	1.527	0.127	-2.12e-06	1.71e-05
yasli_yl	9.65e-06	2.32e-06	4.162	0.000	5.11e-06	1.42e-05
sosy_yl	5.076e-06	2.52e-06	2.018	0.044	1.47e-07	1e-05
hasta_yl	2.986e-07	2.49e-06	0.120	0.904	-4.58e-06	5.18e-06
dul_yl	2.14e-06	5.49e-07	3.898	0.000	1.06e-06	3.22e-06
gazi_yl	5.474e-06	1.33e-06	4.122	0.000	2.87e-06	8.08e-06
burs_yl	1.735e-06	1.27e-06	1.366	0.172	-7.54e-07	4.22e-06
issiz_yl	6.144e-06	3.2e-06	1.919	0.055	-1.32e-07	1.24e-05
ddestek_yl	2.12e-06	8.53e-07	2.486	0.013	4.49e-07	3.79e-06
dayni_yl	2.918e-05	3.9e-06	7.486	0.000	2.15e-05	3.68e-05
ydemk_yl	2.902e-06	5.91e-07	4.907	0.000	1.74e-06	4.06e-06
yddav_yl	7.096e-07	7.45e-07	0.952	0.341	-7.51e-07	2.17e-06
ydayn_yl	2.678e-06	6.29e-06	0.426	0.670	-9.65e-06	1.5e-05
dgnfk_yl	-2.991e-06	4.47e-07	-6.698	0.000	-3.87e-06	-2.12e-06
dgbrs_yl	-2.626e-06	9.28e-07	-2.831	0.005	-4.44e-06	-8.08e-07

Table C.9 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
dsemk_yl	2.075e-06	1.28e-06	1.619	0.105	-4.36e-07	4.59e-06
dgayn_yl	1.977e-05	1.53e-06	12.923	0.000	1.68e-05	2.28e-05
topga_ay	2.414e-05	7.26e-06	3.326	0.001	9.91e-06	3.84e-05
topgelay	4.671e-08	2.54e-07	0.184	0.854	-4.51e-07	5.44e-07
topga_yl	5.706e-07	1.19e-06	0.479	0.632	-1.76e-06	2.91e-06
topgelyl	3.536e-06	5.52e-08	64.088	0.000	3.43e-06	3.64e-06
indeks_arl	0.2598	0.012	21.406	0.000	0.236	0.284
kira_ay	-4.644e-05	5.11e-06	-9.092	0.000	-5.65e-05	-3.64e-05
loj_fert	-0.0063	0.010	-0.637	0.524	-0.026	0.013
izafi_k	-9.745e-05	4.19e-06	-23.276	0.000	-0.000	-8.92e-05
konutsah	5.89e-06	1.41e-06	4.177	0.000	3.13e-06	8.65e-06
kon_sure	0.0004	0.000	2.631	0.009	0.000	0.001
kon_borc	9.128e-08	1.03e-08	8.877	0.000	7.11e-08	1.11e-07
kondeger	1.381e-07	1.43e-08	9.667	0.000	1.1e-07	1.66e-07
otur_sur	-0.0002	4.32e-05	-5.708	0.000	-0.000	-0.000

Table C.10 : Quantile Regression Results w/ FS - 40th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.08841
Model:	QuantReg	**Bandwidth:**	0.01171
Method:	Least Squares	**Sparsity:**	0.4289
Date:	Paz, 21 Tem 2019	**No. Observations:**	371613
Time:	18:36:30	**Df Residuals:**	371357

Quantile Regression Coefficients w/ FS for 40th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	0.2160	0.042	5.199	0.000	0.135	0.297
year[T.2003]	0.0021	0.002	1.190	0.234	-0.001	0.005
year[T.2004]	-0.0099	0.003	-3.927	0.000	-0.015	-0.005
year[T.2005]	-0.0080	0.002	-3.688	0.000	-0.012	-0.004
year[T.2006]	-0.0029	0.002	-1.348	0.178	-0.007	0.001
year[T.2007]	-0.0145	0.002	-6.624	0.000	-0.019	-0.010
year[T.2008]	-0.0161	0.002	-7.056	0.000	-0.021	-0.012
year[T.2009]	-0.0116	0.003	-4.073	0.000	-0.017	-0.006
year[T.2010]	-0.0143	0.003	-5.085	0.000	-0.020	-0.009
year[T.2011]	-0.0124	0.003	-4.403	0.000	-0.018	-0.007
year[T.2012]	-0.0136	0.003	-4.825	0.000	-0.019	-0.008
year[T.2013]	-0.0157	0.003	-5.569	0.000	-0.021	-0.010
year[T.2014]	-0.0016	0.003	-0.589	0.556	-0.007	0.004
year[T.2015]	0.0080	0.003	2.830	0.005	0.002	0.013
year[T.2016]	0.0158	0.003	5.637	0.000	0.010	0.021
cinsiyet[T.2]	-0.0009	0.001	-0.913	0.361	-0.003	0.001
yakinlik[T.1.0]	0.0112	0.002	7.179	0.000	0.008	0.014
yakinlik[T.2.0]	0.0074	0.002	3.841	0.000	0.004	0.011
yakinlik[T.3.0]	0.0074	0.003	2.625	0.009	0.002	0.013
yakinlik[T.4.0]	0.0021	0.004	0.504	0.614	-0.006	0.010
yakinlik[T.5.0]	0.0054	0.006	0.855	0.392	-0.007	0.018
yakinlik[T.6.0]	0.0113	0.003	3.767	0.000	0.005	0.017
yakinlik[T.7.0]	0.0113	0.003	4.027	0.000	0.006	0.017
yakinlik[T.8.0]	0.0006	0.004	0.163	0.870	-0.007	0.008
yakinlik[T.9.0]	-0.0003	0.006	-0.057	0.955	-0.012	0.012
yakinlik[T.10.0]	-0.0123	0.010	-1.221	0.222	-0.032	0.007
sagliks[T.2.0]	0.0190	0.003	7.090	0.000	0.014	0.024
sagliks[T.3.0]	-0.0195	0.001	-14.149	0.000	-0.022	-0.017
sagliks[T.4.0]	-0.0038	0.001	-3.444	0.001	-0.006	-0.002
ogrenci[T.2.0]	0.0027	0.001	1.839	0.066	-0.000	0.006
egitim_devam[T.2]	-0.0007	0.003	-0.214	0.830	-0.007	0.006
egitim_devam[T.3]	-0.0054	0.004	-1.247	0.212	-0.014	0.003
egitim_devam[T.4]	-0.0091	0.006	-1.554	0.120	-0.021	0.002
egitim_devam[T.5]	-0.0186	0.005	-3.758	0.000	-0.028	-0.009
egitim_devam[T.6]	-0.0117	0.008	-1.453	0.146	-0.027	0.004
egitim_devam[T.7]	0.0304	0.027	1.132	0.257	-0.022	0.083
egitim_devam[T.8]	0.0264	0.054	0.485	0.628	-0.080	0.133

Table C.10 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-0.1345	0.098	-1.367	0.172	-0.327	0.058
egitim_devam[T.999]	-0.0031	0.002	-1.448	0.148	-0.007	0.001
medenidr[T.2.0]	-0.0143	0.002	-7.252	0.000	-0.018	-0.010
medenidr[T.3.0]	0.0081	0.003	2.866	0.004	0.003	0.014
medenidr[T.4.0]	0.0024	0.003	0.693	0.488	-0.004	0.009
is_arama[T.2]	0.0006	0.002	0.248	0.804	-0.004	0.005
is_arama[T.999]	0.0047	0.003	1.600	0.110	-0.001	0.010
isneden[T.2.0]	0.0123	0.013	0.930	0.352	-0.014	0.038
isneden[T.3.0]	0.0238	0.013	1.812	0.070	-0.002	0.050
isneden[T.4.0]	0.0343	0.013	2.595	0.009	0.008	0.060
isneden[T.5.0]	0.0846	0.024	3.593	0.000	0.038	0.131
isneden[T.6.0]	0.0276	0.013	2.069	0.039	0.001	0.054
isneden[T.7.0]	0.0291	0.013	2.164	0.030	0.003	0.055
isneden[T.10.0]	0.0320	0.015	2.171	0.030	0.003	0.061
isneden[T.11.0]	0.0079	0.014	0.582	0.560	-0.019	0.034
sekkod[T.2.0]	0.0156	0.008	1.972	0.049	9.79e-05	0.031
sekkod[T.3.0]	-0.0006	0.003	-0.178	0.859	-0.007	0.006
sekkod[T.4.0]	0.0007	0.005	0.121	0.904	-0.010	0.011
sekkod[T.5.0]	-0.0056	0.004	-1.579	0.114	-0.013	0.001
sekkod[T.6.0]	0.0037	0.003	1.191	0.234	-0.002	0.010
sekkod[T.7.0]	-0.0081	0.004	-2.117	0.034	-0.016	-0.001
sekkod[T.8.0]	0.0065	0.004	1.740	0.082	-0.001	0.014
sekkod[T.9.0]	0.0010	0.006	0.172	0.864	-0.011	0.013
sekkod[T.10.0]	-0.0083	0.006	-1.488	0.137	-0.019	0.003
sekkod[T.11.0]	0.0061	0.004	1.609	0.108	-0.001	0.013
sekkod[T.12.0]	0.0040	0.004	0.966	0.334	-0.004	0.012
sekkod[T.13.0]	4.425e-05	0.004	0.010	0.992	-0.008	0.009
sekkod[T.14.0]	0.0069	0.004	1.783	0.075	-0.001	0.014
sgk[T.2.0]	0.0094	0.013	0.749	0.454	-0.015	0.034
sgk[T.3.0]	-0.0082	0.002	-4.180	0.000	-0.012	-0.004
is_statu[T.2]	0.0108	0.003	3.433	0.001	0.005	0.017
is_statu[T.3]	0.0379	0.007	5.063	0.000	0.023	0.053
is_statu[T.999]	-0.0012	0.002	-0.540	0.589	-0.006	0.003
koopera[T.2]	0.0093	0.005	1.807	0.071	-0.001	0.019
koopera[T.999]	0.0073	0.005	1.483	0.138	-0.002	0.017
sir_kon[T.2]	-0.0095	0.008	-1.245	0.213	-0.025	0.005
sir_kon[T.999]	-0.0026	0.007	-0.355	0.722	-0.017	0.012
isdur[T.2.0]	0.0077	0.003	2.368	0.018	0.001	0.014
isdur[T.3.0]	0.0240	0.004	6.692	0.000	0.017	0.031
isdur[T.4.0]	0.0231	0.003	8.376	0.000	0.018	0.029
isdur[T.5.0]	0.0220	0.003	6.485	0.000	0.015	0.029
ist_tesk[T.2.0]	-0.0056	0.004	-1.505	0.132	-0.013	0.002
ist_tesk[T.3.0]	0.0021	0.004	0.540	0.589	-0.006	0.010
guc_kul[T.2]	0.0027	0.003	0.926	0.354	-0.003	0.009

Table C.10 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.3]	-6.789e-05	0.002	-0.028	0.978	-0.005	0.005
guc_kul[T.999]	0.0044	0.003	1.682	0.093	-0.001	0.010
isfaal[T.2.0]	-0.0086	0.004	-2.418	0.016	-0.016	-0.002
mulk_dur[T.2.0]	-0.0185	0.004	-5.069	0.000	-0.026	-0.011
mulk_dur[T.3.0]	-0.0177	0.005	-3.594	0.000	-0.027	-0.008
ekis_yl[T.2.0]	-0.0081	0.005	-1.623	0.105	-0.018	0.002
ekis_ay[T.2.0]	0.0145	0.006	2.527	0.011	0.003	0.026
sekkod_e[T.2.0]	0.1923	0.069	2.807	0.005	0.058	0.327
sekkod_e[T.3.0]	0.1058	0.015	6.906	0.000	0.076	0.136
sekkod_e[T.4.0]	0.1471	0.040	3.695	0.000	0.069	0.225
sekkod_e[T.5.0]	0.0531	0.013	4.036	0.000	0.027	0.079
sekkod_e[T.6.0]	0.1049	0.013	8.091	0.000	0.080	0.130
sekkod_e[T.7.0]	0.0917	0.017	5.470	0.000	0.059	0.125
sekkod_e[T.8.0]	0.1142	0.017	6.792	0.000	0.081	0.147
sekkod_e[T.9.0]	0.2464	0.038	6.451	0.000	0.172	0.321
sekkod_e[T.10.0]	0.1748	0.030	5.913	0.000	0.117	0.233
sekkod_e[T.11.0]	0.1233	0.018	6.919	0.000	0.088	0.158
sekkod_e[T.12.0]	0.1597	0.024	6.791	0.000	0.114	0.206
sekkod_e[T.13.0]	0.1478	0.029	5.073	0.000	0.091	0.205
sekkod_e[T.14.0]	0.0860	0.013	6.390	0.000	0.060	0.112
meslek_e[T.1.0]	0.0794	0.014	5.487	0.000	0.051	0.108
meslek_e[T.2.0]	-0.0013	0.018	-0.071	0.943	-0.037	0.035
meslek_e[T.3.0]	0.0530	0.018	2.937	0.003	0.018	0.088
meslek_e[T.4.0]	-0.0545	0.031	-1.764	0.078	-0.115	0.006
meslek_e[T.5.0]	0.0388	0.016	2.437	0.015	0.008	0.070
meslek_e[T.6.0]	0.1637	0.015	10.813	0.000	0.134	0.193
meslek_e[T.7.0]	0.0745	0.015	4.953	0.000	0.045	0.104
meslek_e[T.8.0]	0.0562	0.016	3.507	0.000	0.025	0.088
meslek_e[T.9.0]	0.1121	0.014	8.247	0.000	0.085	0.139
isdur_e[T.2.0]	0.0214	0.010	2.055	0.040	0.001	0.042
isdur_e[T.3.0]	0.0049	0.015	0.321	0.748	-0.025	0.035
isdur_e[T.4.0]	0.0128	0.010	1.256	0.209	-0.007	0.033
isdur_e[T.5.0]	0.0169	0.012	1.370	0.171	-0.007	0.041
cl_neden[T.2.0]	0.0031	0.007	0.432	0.666	-0.011	0.017
cl_neden[T.3.0]	0.0178	0.004	4.584	0.000	0.010	0.025
cl_neden[T.4.0]	0.0189	0.008	2.502	0.012	0.004	0.034
cl_neden[T.5.0]	0.0353	0.008	4.396	0.000	0.020	0.051
cl_neden[T.6.0]	0.0175	0.006	3.073	0.002	0.006	0.029
cl_neden[T.7.0]	0.0126	0.006	1.993	0.046	0.000	0.025
ayni_is[T.2.0]	-0.0178	0.008	-2.281	0.023	-0.033	-0.003
sekkod_d[T.2.0]	0.0202	0.030	0.670	0.503	-0.039	0.080
sekkod_d[T.3.0]	0.0278	0.011	2.641	0.008	0.007	0.048
sekkod_d[T.4.0]	0.0348	0.018	1.968	0.049	0.000	0.069
sekkod_d[T.5.0]	0.0008	0.010	0.083	0.934	-0.019	0.020

Table C.10 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.6.0]	0.0054	0.011	0.489	0.625	-0.016	0.027
sekkod_d[T.7.0]	0.0047	0.013	0.365	0.715	-0.020	0.030
sekkod_d[T.8.0]	0.0176	0.012	1.420	0.156	-0.007	0.042
sekkod_d[T.9.0]	-0.0030	0.025	-0.120	0.904	-0.052	0.046
sekkod_d[T.10.0]	-0.0011	0.020	-0.058	0.954	-0.040	0.037
sekkod_d[T.11.0]	0.0227	0.015	1.531	0.126	-0.006	0.052
sekkod_d[T.12.0]	0.0413	0.020	2.049	0.040	0.002	0.081
sekkod_d[T.13.0]	0.0253	0.021	1.199	0.230	-0.016	0.067
sekkod_d[T.14.0]	0.0145	0.013	1.137	0.256	-0.011	0.040
meslek_d[T.2.0]	-0.0768	0.018	-4.279	0.000	-0.112	-0.042
meslek_d[T.3.0]	-0.0369	0.015	-2.431	0.015	-0.067	-0.007
meslek_d[T.4.0]	-0.0529	0.015	-3.466	0.001	-0.083	-0.023
meslek_d[T.5.0]	-0.0476	0.013	-3.650	0.000	-0.073	-0.022
meslek_d[T.6.0]	-0.0283	0.017	-1.626	0.104	-0.062	0.006
meslek_d[T.7.0]	-0.0626	0.013	-4.832	0.000	-0.088	-0.037
meslek_d[T.8.0]	-0.0660	0.014	-4.719	0.000	-0.093	-0.039
meslek_d[T.9.0]	-0.0579	0.013	-4.384	0.000	-0.084	-0.032
isdur_d[T.2.0]	0.0076	0.006	1.207	0.227	-0.005	0.020
isdur_d[T.3.0]	-0.0241	0.021	-1.145	0.252	-0.065	0.017
isdur_d[T.4.0]	-0.0142	0.009	-1.518	0.129	-0.033	0.004
isdur_d[T.5.0]	-0.0122	0.013	-0.970	0.332	-0.037	0.012
hhtipi[T.2.0]	-0.0113	0.001	-9.620	0.000	-0.014	-0.009
hhtipi[T.3.0]	-0.0184	0.001	-14.791	0.000	-0.021	-0.016
hhtipi[T.4.0]	0.0083	0.002	5.199	0.000	0.005	0.011
hhtipi[T.5.0]	-0.0042	0.001	-3.283	0.001	-0.007	-0.002
hhtipi[T.6.0]	0.0023	0.002	1.165	0.244	-0.002	0.006
hhtipi[T.7.0]	0.0057	0.004	1.406	0.160	-0.002	0.014
konuttip[T.2.0]	-0.0057	0.002	-3.573	0.000	-0.009	-0.003
konuttip[T.3.0]	-0.0053	0.001	-4.015	0.000	-0.008	-0.003
konuttip[T.4.0]	0.0091	0.012	0.778	0.436	-0.014	0.032
mulkiyet[T.2.0]	-0.0220	0.001	-18.104	0.000	-0.024	-0.020
mulkiyet[T.3.0]	0.0254	0.003	9.326	0.000	0.020	0.031
mulkiyet[T.4.0]	-0.0065	0.001	-4.767	0.000	-0.009	-0.004
kira_sek[T.2.0]	0.0018	0.010	0.185	0.854	-0.017	0.020
yapituru[T.2.0]	0.0751	0.028	2.708	0.007	0.021	0.129
yapituru[T.3.0]	-0.0034	0.004	-0.946	0.344	-0.010	0.004
yapituru[T.4.0]	0.0016	0.002	0.800	0.424	-0.002	0.006
yapituru[T.5.0]	-0.0078	0.002	-3.188	0.001	-0.013	-0.003
yapituru[T.6.0]	-0.0081	0.002	-4.717	0.000	-0.012	-0.005
yapituru[T.7.0]	0.0048	0.003	1.913	0.056	-0.000	0.010
yapituru[T.8.0]	-0.0128	0.010	-1.314	0.189	-0.032	0.006
zemin_sal[T.2.0]	0.0055	0.002	3.243	0.001	0.002	0.009
zemin_sal[T.3.0]	0.0037	0.002	2.011	0.044	9.46e-05	0.007
zemin_sal[T.4.0]	0.0032	0.002	1.554	0.120	-0.001	0.007

Table C.10 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.5.0]	0.0003	0.004	0.069	0.945	-0.007	0.008
zemin_sal[T.6.0]	0.0025	0.002	1.205	0.228	-0.002	0.007
zemin_sal[T.7.0]	0.0039	0.004	0.974	0.330	-0.004	0.012
zemin_sal[T.8.0]	-0.0061	0.012	-0.521	0.603	-0.029	0.017
zemin_sal[T.9.0]	-0.0023	0.004	-0.520	0.603	-0.011	0.006
zemin_sal[T.98.0]	0.0203	0.009	2.279	0.023	0.003	0.038
zemin_mut[T.2.0]	0.0050	0.004	1.305	0.192	-0.002	0.012
zemin_mut[T.3.0]	0.0020	0.003	0.773	0.439	-0.003	0.007
zemin_mut[T.4.0]	0.0162	0.005	3.501	0.000	0.007	0.025
zemin_mut[T.5.0]	0.0088	0.010	0.921	0.357	-0.010	0.028
zemin_mut[T.6.0]	0.0076	0.003	2.261	0.024	0.001	0.014
zemin_mut[T.7.0]	-0.0028	0.005	-0.559	0.576	-0.012	0.007
zemin_mut[T.8.0]	-0.0083	0.009	-0.900	0.368	-0.026	0.010
zemin_mut[T.9.0]	0.0202	0.006	3.359	0.001	0.008	0.032
zemin_mut[T.98.0]	0.0072	0.010	0.691	0.490	-0.013	0.028
zemin_ban[T.2.0]	-0.0966	0.022	-4.314	0.000	-0.141	-0.053
zemin_ban[T.3.0]	-0.1306	0.018	-7.308	0.000	-0.166	-0.096
zemin_ban[T.4.0]	-0.1542	0.020	-7.575	0.000	-0.194	-0.114
zemin_ban[T.5.0]	0.1790	0.055	3.271	0.001	0.072	0.286
zemin_ban[T.6.0]	-0.1214	0.018	-6.760	0.000	-0.157	-0.086
zemin_ban[T.7.0]	-0.1230	0.018	-6.737	0.000	-0.159	-0.087
zemin_ban[T.8.0]	-0.1444	0.020	-7.226	0.000	-0.184	-0.105
zemin_ban[T.9.0]	-0.1354	0.019	-7.069	0.000	-0.173	-0.098
zemin_ban[T.98.0]	-0.0601	0.033	-1.837	0.066	-0.124	0.004
isit_sis[T.2.0]	-0.0038	0.002	-2.253	0.024	-0.007	-0.000
isit_sis[T.3.0]	-0.0002	0.002	-0.134	0.893	-0.004	0.003
isit_sis[T.4.0]	0.0069	0.003	2.030	0.042	0.000	0.014
yakit_1[T.2.0]	-0.0027	0.001	-1.964	0.049	-0.005	-6.22e-06
yakit_1[T.3.0]	-0.0138	0.002	-7.044	0.000	-0.018	-0.010
yakit_1[T.4.0]	0.0026	0.002	1.199	0.230	-0.002	0.007
yakit_1[T.5.0]	0.0103	0.003	4.092	0.000	0.005	0.015
yakit_1[T.6.0]	-0.0151	0.002	-6.551	0.000	-0.020	-0.011
yakit_2[T.2.0]	0.0003	0.002	0.130	0.897	-0.004	0.005
yakit_2[T.3.0]	-0.0160	0.014	-1.132	0.258	-0.044	0.012
yakit_2[T.4.0]	-0.0156	0.004	-3.568	0.000	-0.024	-0.007
yakit_2[T.5.0]	0.0030	0.003	0.927	0.354	-0.003	0.009
yakit_2[T.6.0]	0.0100	0.004	2.400	0.016	0.002	0.018
yakit_3[T.2.0]	-0.0141	0.012	-1.168	0.243	-0.038	0.010
yakit_3[T.3.0]	-0.0378	0.021	-1.840	0.066	-0.078	0.002
yakit_3[T.4.0]	-0.0322	0.010	-3.194	0.001	-0.052	-0.012
yakit_3[T.5.0]	-0.0283	0.009	-3.168	0.002	-0.046	-0.011
yakit_3[T.6.0]	-0.0095	0.010	-0.935	0.350	-0.029	0.010
yakit_mut[T.2.0]	0.0112	0.008	1.490	0.136	-0.004	0.026
yakit_mut[T.3.0]	0.0065	0.005	1.399	0.162	-0.003	0.016

Table C.10 (Continued) : Quantile Regression Coefficients for 40th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.4.0]	-0.0047	0.003	-1.569	0.117	-0.011	0.001
yakit_mut[T.5.0]	-0.0122	0.005	-2.541	0.011	-0.022	-0.003
yakit_mut[T.6.0]	0.0365	0.005	6.762	0.000	0.026	0.047
yakit_mut[T.7.0]	0.0992	0.025	3.933	0.000	0.050	0.149
yakit_mut[T.8.0]	0.0570	0.015	3.740	0.000	0.027	0.087
yakit_su[T.2.0]	0.0037	0.005	0.815	0.415	-0.005	0.013
yakit_su[T.3.0]	-0.0114	0.004	-2.696	0.007	-0.020	-0.003
yakit_su[T.4.0]	-0.0134	0.003	-4.697	0.000	-0.019	-0.008
yakit_su[T.5.0]	-0.0061	0.002	-2.608	0.009	-0.011	-0.002
yakit_su[T.6.0]	-0.0058	0.003	-2.033	0.042	-0.011	-0.000
yakit_su[T.7.0]	-0.0076	0.003	-3.009	0.003	-0.013	-0.003
yakit_su[T.8.0]	0.0194	0.008	2.419	0.016	0.004	0.035
zor_egiti[T.2.0]	-0.0154	0.003	-5.828	0.000	-0.021	-0.010
zor_egiti[T.3.0]	-0.0057	0.002	-2.290	0.022	-0.010	-0.001
zor_egiti[T.4.0]	-0.0067	0.003	-2.468	0.014	-0.012	-0.001
banyo[T.1.0]	-0.0006	0.002	-0.280	0.779	-0.005	0.004
sauna[T.1.0]	-0.0384	0.012	-3.220	0.001	-0.062	-0.015
jakuzi[T.1.0]	-0.0570	0.005	-10.399	0.000	-0.068	-0.046
tuvalet[T.1.0]	-0.0101	0.001	-7.289	0.000	-0.013	-0.007
mutfak[T.1.0]	0.0027	0.003	1.002	0.316	-0.003	0.008
copogut[T.1.0]	-0.0532	0.007	-7.862	0.000	-0.066	-0.040
tabanisi[T.1.0]	0.0137	0.005	2.547	0.011	0.003	0.024
su_sis[T.1.0]	0.0299	0.002	14.236	0.000	0.026	0.034
elektri[T.1.0]	-0.0052	0.023	-0.227	0.821	-0.051	0.040
sicaksu[T.1.0]	6.497e-05	0.001	0.062	0.950	-0.002	0.002
kabloyay[T.1.0]	-0.0093	0.002	-5.032	0.000	-0.013	-0.006
asansor[T.1.0]	-0.0010	0.001	-0.761	0.447	-0.004	0.002
garaj[T.1.0]	-0.0127	0.002	-7.346	0.000	-0.016	-0.009
havuz[T.1.0]	-0.0430	0.005	-8.197	0.000	-0.053	-0.033
jenrator[T.1.0]	-0.0390	0.004	-9.660	0.000	-0.047	-0.031
apartman[T.1.0]	0.0130	0.001	10.372	0.000	0.011	0.015
yazlik[T.1.0]	-0.0063	0.002	-2.613	0.009	-0.011	-0.002
tarla[T.1.0]	0.0113	0.001	10.889	0.000	0.009	0.013
bag[T.1.0]	0.0044	0.001	3.438	0.001	0.002	0.007
sera[T.1.0]	0.0182	0.004	4.435	0.000	0.010	0.026
arsa[T.1.0]	-0.0042	0.002	-2.422	0.015	-0.008	-0.001
dukkan[T.1.0]	0.0166	0.002	10.449	0.000	0.013	0.020
otel[T.1.0]	-0.0587	0.013	-4.356	0.000	-0.085	-0.032
diger[T.1.0]	0.0116	0.006	1.887	0.059	-0.000	0.024
kirkntkd[T.2.0]	-0.0124	0.001	-12.035	0.000	-0.014	-0.010
izafi_k	-0.0002	2.56e-06	-66.249	0.000	-0.000	-0.000
ceptel	-0.0111	0.000	-30.199	0.000	-0.012	-0.010
yilkulgel	4.26e-06	1.97e-08	215.705	0.000	4.22e-06	4.3e-06

Table C.11 : Quantile Regression Results - 80th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.09017
Model:	QuantReg	**Bandwidth:**	0.01040
Method:	Least Squares	**Sparsity:**	0.7362
Date:	Cts, 23 Haz 2018	**No. Observations:**	371613
Time:	21:05:37	**Df Residuals:**	371226

Quantile Regression Coefficients for 80th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	-1.0234	0.048	-21.174	0.000	-1.118	-0.929
year[T.2003]	0.0378	0.002	15.823	0.000	0.033	0.042
year[T.2004]	0.0178	0.003	5.170	0.000	0.011	0.025
year[T.2005]	0.0251	0.003	8.417	0.000	0.019	0.031
year[T.2006]	0.0224	0.003	7.615	0.000	0.017	0.028
year[T.2007]	0.0015	0.003	0.511	0.609	-0.004	0.007
year[T.2008]	-0.0172	0.003	-6.119	0.000	-0.023	-0.012
year[T.2009]	-0.0086	0.002	-4.141	0.000	-0.013	-0.005
year[T.2010]	-0.0170	0.002	-8.293	0.000	-0.021	-0.013
year[T.2011]	-0.0100	0.002	-4.878	0.000	-0.014	-0.006
year[T.2012]	-0.0096	0.002	-4.928	0.000	-0.013	-0.006
year[T.2013]	-0.0078	0.002	-3.855	0.000	-0.012	-0.004
year[T.2014]	0.0113	0.002	5.508	0.000	0.007	0.015
year[T.2015]	0.0259	0.003	10.228	0.000	0.021	0.031
year[T.2016]	0.0363	0.002	14.600	0.000	0.031	0.041
cinsiyet[T.2]	0.0016	0.001	1.273	0.203	-0.001	0.004
yakinlik[T.1.0]	0.0549	0.002	26.770	0.000	0.051	0.059
yakinlik[T.2.0]	0.0571	0.003	21.748	0.000	0.052	0.062
yakinlik[T.3.0]	0.0406	0.004	11.064	0.000	0.033	0.048
yakinlik[T.4.0]	0.0556	0.005	10.341	0.000	0.045	0.066
yakinlik[T.5.0]	0.0618	0.008	8.022	0.000	0.047	0.077
yakinlik[T.6.0]	0.0728	0.004	18.522	0.000	0.065	0.080
yakinlik[T.7.0]	0.0706	0.004	18.947	0.000	0.063	0.078
yakinlik[T.8.0]	0.0695	0.005	14.779	0.000	0.060	0.079
yakinlik[T.9.0]	0.0755	0.008	9.767	0.000	0.060	0.091
yakinlik[T.10.0]	0.0840	0.013	6.490	0.000	0.059	0.109
sagliks[T.2.0]	0.0500	0.003	14.979	0.000	0.043	0.057
sagliks[T.3.0]	-0.0433	0.002	-24.618	0.000	-0.047	-0.040
sagliks[T.4.0]	-0.0058	0.001	-4.105	0.000	-0.009	-0.003
ogrenci[T.2.0]	0.0041	0.002	2.158	0.031	0.000	0.008
egitim_devam[T.2]	0.0120	0.004	2.958	0.003	0.004	0.020
egitim_devam[T.3]	-0.0262	0.006	-4.653	0.000	-0.037	-0.015
egitim_devam[T.4]	0.0185	0.007	2.616	0.009	0.005	0.032
egitim_devam[T.5]	-0.0165	0.006	-2.605	0.009	-0.029	-0.004
egitim_devam[T.6]	0.0179	0.010	1.825	0.068	-0.001	0.037
egitim_devam[T.7]	-0.0244	0.036	-0.677	0.498	-0.095	0.046
egitim_devam[T.8]	-0.0101	0.070	-0.144	0.885	-0.147	0.126

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-7.696e-05	0.132	-0.001	1.000	-0.259	0.258
egitim_devam[T.999]	-0.0006	0.003	-0.204	0.839	-0.006	0.005
medenidr[T.2.0]	-0.0331	0.003	-12.839	0.000	-0.038	-0.028
medenidr[T.3.0]	-0.0177	0.004	-4.349	0.000	-0.026	-0.010
medenidr[T.4.0]	-0.0306	0.004	-6.857	0.000	-0.039	-0.022
is_arama[T.2]	0.0203	0.003	6.503	0.000	0.014	0.026
is_arama[T.999]	0.0399	0.004	9.946	0.000	0.032	0.048
isneden[T.2.0]	0.1134	0.019	6.063	0.000	0.077	0.150
isneden[T.3.0]	0.1330	0.019	7.142	0.000	0.096	0.169
isneden[T.4.0]	0.1230	0.019	6.546	0.000	0.086	0.160
isneden[T.5.0]	0.2256	0.030	7.526	0.000	0.167	0.284
isneden[T.6.0]	0.1341	0.019	7.109	0.000	0.097	0.171
isneden[T.7.0]	0.1183	0.019	6.241	0.000	0.081	0.155
isneden[T.10.0]	0.1511	0.020	7.424	0.000	0.111	0.191
isneden[T.11.0]	0.1084	0.019	5.696	0.000	0.071	0.146
sekkod[T.2.0]	0.0233	0.010	2.275	0.023	0.003	0.043
sekkod[T.3.0]	-0.0030	0.004	-0.716	0.474	-0.011	0.005
sekkod[T.4.0]	0.0011	0.007	0.155	0.877	-0.013	0.015
sekkod[T.5.0]	-0.0081	0.005	-1.726	0.084	-0.017	0.001
sekkod[T.6.0]	0.0005	0.004	0.115	0.908	-0.007	0.008
sekkod[T.7.0]	0.0103	0.005	2.105	0.035	0.001	0.020
sekkod[T.8.0]	0.0120	0.005	2.488	0.013	0.003	0.021
sekkod[T.9.0]	-0.0017	0.008	-0.224	0.822	-0.017	0.013
sekkod[T.10.0]	-0.0192	0.007	-2.623	0.009	-0.034	-0.005
sekkod[T.11.0]	0.0030	0.005	0.609	0.543	-0.007	0.013
sekkod[T.12.0]	0.0158	0.005	2.871	0.004	0.005	0.027
sekkod[T.13.0]	0.0100	0.006	1.762	0.078	-0.001	0.021
sekkod[T.14.0]	0.0063	0.005	1.271	0.204	-0.003	0.016
sgk[T.2.0]	-0.0389	0.016	-2.392	0.017	-0.071	-0.007
sgk[T.3.0]	-0.0122	0.003	-4.723	0.000	-0.017	-0.007
is_statu[T.2]	0.0120	0.005	2.634	0.008	0.003	0.021
is_statu[T.3]	0.0200	0.010	1.981	0.048	0.000	0.040
is_statu[T.999]	0.0094	0.004	2.345	0.019	0.002	0.017
koopera[T.2]	0.0464	0.007	6.551	0.000	0.033	0.060
koopera[T.999]	0.0221	0.007	3.357	0.001	0.009	0.035
sir_kon[T.2]	-0.0057	0.010	-0.570	0.569	-0.025	0.014
sir_kon[T.999]	-0.0148	0.009	-1.566	0.117	-0.033	0.004
isdur[T.2.0]	-0.0217	0.009	-2.301	0.021	-0.040	-0.003
isdur[T.3.0]	0.0276	0.005	5.439	0.000	0.018	0.037
isdur[T.4.0]	0.0362	0.004	9.243	0.000	0.029	0.044
isdur[T.5.0]	0.0731	0.005	15.594	0.000	0.064	0.082
ist_tesk[T.2.0]	0.0238	0.009	2.706	0.007	0.007	0.041
ist_tesk[T.3.0]	0.0360	0.010	3.624	0.000	0.017	0.055
guc_kul[T.2]	0.0100	0.004	2.619	0.009	0.003	0.018

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.3]	-0.0011	0.003	-0.353	0.724	-0.007	0.005
guc_kul[T.999]	-0.0061	0.004	-1.739	0.082	-0.013	0.001
isfaal[T.2.0]	-0.0072	0.005	-1.523	0.128	-0.016	0.002
mulk_dur[T.2.0]	0.0097	0.005	2.046	0.041	0.000	0.019
mulk_dur[T.3.0]	0.0242	0.006	3.933	0.000	0.012	0.036
ekis_yl[T.2.0]	0.0147	0.007	2.265	0.024	0.002	0.028
ekis_ay[T.2.0]	0.0266	0.007	3.587	0.000	0.012	0.041
sekkod_e[T.2.0]	0.0422	0.091	0.462	0.644	-0.137	0.221
sekkod_e[T.3.0]	-0.0363	0.022	-1.637	0.102	-0.080	0.007
sekkod_e[T.4.0]	-0.1008	0.067	-1.501	0.133	-0.232	0.031
sekkod_e[T.5.0]	-0.0008	0.018	-0.045	0.964	-0.036	0.034
sekkod_e[T.6.0]	0.0342	0.019	1.799	0.072	-0.003	0.071
sekkod_e[T.7.0]	0.0250	0.024	1.021	0.307	-0.023	0.073
sekkod_e[T.8.0]	0.0415	0.025	1.656	0.098	-0.008	0.091
sekkod_e[T.9.0]	0.0516	0.057	0.907	0.364	-0.060	0.163
sekkod_e[T.10.0]	0.0189	0.043	0.438	0.662	-0.066	0.104
sekkod_e[T.11.0]	0.0253	0.027	0.945	0.344	-0.027	0.078
sekkod_e[T.12.0]	-0.0068	0.038	-0.179	0.858	-0.082	0.068
sekkod_e[T.13.0]	-0.0159	0.049	-0.327	0.744	-0.111	0.079
sekkod_e[T.14.0]	0.1121	0.018	6.177	0.000	0.077	0.148
meslek_e[T.1.0]	-0.0180	0.017	-1.067	0.286	-0.051	0.015
meslek_e[T.2.0]	0.0106	0.024	0.438	0.662	-0.037	0.058
meslek_e[T.3.0]	0.0155	0.023	0.682	0.495	-0.029	0.060
meslek_e[T.4.0]	0.0431	0.043	1.003	0.316	-0.041	0.127
meslek_e[T.5.0]	-0.0270	0.019	-1.416	0.157	-0.064	0.010
meslek_e[T.6.0]	0.0351	0.015	2.354	0.019	0.006	0.064
meslek_e[T.7.0]	0.0706	0.017	4.156	0.000	0.037	0.104
meslek_e[T.8.0]	0.0100	0.019	0.521	0.602	-0.028	0.048
meslek_e[T.9.0]	0.0233	0.013	1.836	0.066	-0.002	0.048
isdur_e[T.2.0]	0.0844	0.014	5.986	0.000	0.057	0.112
isdur_e[T.3.0]	7.83e-05	0.022	0.003	0.997	-0.044	0.044
isdur_e[T.4.0]	0.0599	0.014	4.280	0.000	0.032	0.087
isdur_e[T.5.0]	0.0759	0.017	4.427	0.000	0.042	0.109
cl_neden[T.2.0]	0.0189	0.009	2.106	0.035	0.001	0.036
cl_neden[T.3.0]	0.0070	0.005	1.401	0.161	-0.003	0.017
cl_neden[T.4.0]	0.0518	0.009	5.656	0.000	0.034	0.070
cl_neden[T.5.0]	0.0545	0.010	5.623	0.000	0.036	0.073
cl_neden[T.6.0]	0.0139	0.007	1.940	0.052	-0.000	0.028
cl_neden[T.7.0]	0.0341	0.008	4.438	0.000	0.019	0.049
ayni_is[T.2.0]	-0.0157	0.010	-1.597	0.110	-0.035	0.004
sekkod_d[T.2.0]	-0.0251	0.045	-0.557	0.578	-0.114	0.063
sekkod_d[T.3.0]	0.0266	0.013	1.990	0.047	0.000	0.053
sekkod_d[T.4.0]	0.0143	0.023	0.633	0.527	-0.030	0.059
sekkod_d[T.5.0]	0.0024	0.013	0.191	0.848	-0.023	0.027

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.6.0]	0.0124	0.014	0.884	0.377	-0.015	0.040
sekkod_d[T.7.0]	-0.0017	0.017	-0.100	0.920	-0.034	0.031
sekkod_d[T.8.0]	0.0015	0.016	0.094	0.925	-0.031	0.034
sekkod_d[T.9.0]	-0.0184	0.036	-0.516	0.606	-0.088	0.052
sekkod_d[T.10.0]	-0.0158	0.026	-0.619	0.536	-0.066	0.034
sekkod_d[T.11.0]	0.0135	0.019	0.725	0.469	-0.023	0.050
sekkod_d[T.12.0]	0.0086	0.027	0.322	0.748	-0.044	0.061
sekkod_d[T.13.0]	-0.0151	0.030	-0.505	0.614	-0.074	0.044
sekkod_d[T.14.0]	0.0152	0.016	0.936	0.349	-0.017	0.047
meslek_d[T.2.0]	-0.0498	0.023	-2.174	0.030	-0.095	-0.005
meslek_d[T.3.0]	-0.0461	0.020	-2.347	0.019	-0.085	-0.008
meslek_d[T.4.0]	-0.0425	0.020	-2.120	0.034	-0.082	-0.003
meslek_d[T.5.0]	-0.0162	0.017	-0.959	0.338	-0.049	0.017
meslek_d[T.6.0]	0.0135	0.021	0.632	0.528	-0.028	0.056
meslek_d[T.7.0]	-0.0091	0.017	-0.539	0.590	-0.042	0.024
meslek_d[T.8.0]	-0.0306	0.018	-1.700	0.089	-0.066	0.005
meslek_d[T.9.0]	-0.0011	0.017	-0.065	0.948	-0.034	0.032
isdur_d[T.2.0]	0.0065	0.008	0.820	0.412	-0.009	0.022
isdur_d[T.3.0]	-0.0165	0.028	-0.598	0.550	-0.071	0.038
isdur_d[T.4.0]	-0.0358	0.012	-2.962	0.003	-0.059	-0.012
isdur_d[T.5.0]	-0.0197	0.017	-1.180	0.238	-0.052	0.013
hhtipi[T.2.0]	-0.0137	0.001	-9.230	0.000	-0.017	-0.011
hhtipi[T.3.0]	-0.0223	0.002	-14.323	0.000	-0.025	-0.019
hhtipi[T.4.0]	-0.0075	0.002	-3.657	0.000	-0.011	-0.003
hhtipi[T.5.0]	0.0105	0.002	6.478	0.000	0.007	0.014
hhtipi[T.6.0]	-0.0098	0.003	-3.889	0.000	-0.015	-0.005
hhtipi[T.7.0]	0.0107	0.005	2.081	0.037	0.001	0.021
konuttip[T.2.0]	-0.0083	0.002	-4.138	0.000	-0.012	-0.004
konuttip[T.3.0]	-0.0161	0.002	-9.593	0.000	-0.019	-0.013
konuttip[T.4.0]	-0.0253	0.014	-1.769	0.077	-0.053	0.003
mulkiyet[T.2.0]	-0.0115	0.002	-6.868	0.000	-0.015	-0.008
mulkiyet[T.3.0]	0.0413	0.004	10.674	0.000	0.034	0.049
mulkiyet[T.4.0]	-0.0122	0.002	-6.007	0.000	-0.016	-0.008
kira_sek[T.2.0]	0.1237	0.015	8.186	0.000	0.094	0.153
yapituru[T.2.0]	0.0035	0.035	0.101	0.920	-0.065	0.072
yapituru[T.3.0]	-0.0265	0.005	-5.824	0.000	-0.035	-0.018
yapituru[T.4.0]	-0.0039	0.003	-1.525	0.127	-0.009	0.001
yapituru[T.5.0]	-0.0260	0.003	-8.205	0.000	-0.032	-0.020
yapituru[T.6.0]	-0.0125	0.002	-5.705	0.000	-0.017	-0.008
yapituru[T.7.0]	0.0071	0.003	2.241	0.025	0.001	0.013
yapituru[T.8.0]	0.0450	0.012	3.849	0.000	0.022	0.068
zemin_sal[T.2.0]	-8.099e-05	0.002	-0.038	0.970	-0.004	0.004
zemin_sal[T.3.0]	-0.0022	0.002	-0.964	0.335	-0.007	0.002
zemin_sal[T.4.0]	-0.0093	0.003	-3.608	0.000	-0.014	-0.004

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.5.0]	-0.0296	0.005	-5.908	0.000	-0.039	-0.020
zemin_sal[T.6.0]	-2.772e-05	0.003	-0.011	0.992	-0.005	0.005
zemin_sal[T.7.0]	-0.0004	0.005	-0.080	0.936	-0.010	0.010
zemin_sal[T.8.0]	-0.0278	0.014	-2.002	0.045	-0.055	-0.001
zemin_sal[T.9.0]	-0.0011	0.005	-0.202	0.840	-0.012	0.010
zemin_sal[T.98.0]	0.0305	0.011	2.772	0.006	0.009	0.052
zemin_mut[T.2.0]	-0.0085	0.005	-1.781	0.075	-0.018	0.001
zemin_mut[T.3.0]	-0.0050	0.003	-1.525	0.127	-0.011	0.001
zemin_mut[T.4.0]	0.0076	0.006	1.291	0.197	-0.004	0.019
zemin_mut[T.5.0]	0.0165	0.012	1.352	0.176	-0.007	0.040
zemin_mut[T.6.0]	-0.0025	0.004	-0.589	0.556	-0.011	0.006
zemin_mut[T.7.0]	-0.0114	0.006	-1.826	0.068	-0.024	0.001
zemin_mut[T.8.0]	0.0174	0.011	1.518	0.129	-0.005	0.040
zemin_mut[T.9.0]	-0.0178	0.008	-2.333	0.020	-0.033	-0.003
zemin_mut[T.98.0]	0.0445	0.013	3.479	0.001	0.019	0.070
zemin_ban[T.2.0]	0.0523	0.023	2.294	0.022	0.008	0.097
zemin_ban[T.3.0]	0.0121	0.016	0.774	0.439	-0.019	0.043
zemin_ban[T.4.0]	-0.0295	0.020	-1.493	0.135	-0.068	0.009
zemin_ban[T.5.0]	-0.0319	0.063	-0.503	0.615	-0.156	0.092
zemin_ban[T.6.0]	0.0210	0.016	1.336	0.181	-0.010	0.052
zemin_ban[T.7.0]	0.0044	0.016	0.272	0.785	-0.027	0.036
zemin_ban[T.8.0]	0.0259	0.019	1.350	0.177	-0.012	0.063
zemin_ban[T.9.0]	0.0460	0.018	2.597	0.009	0.011	0.081
zemin_ban[T.98.0]	-0.0100	0.040	-0.249	0.804	-0.089	0.069
isit_sis[T.2.0]	0.0172	0.002	8.059	0.000	0.013	0.021
isit_sis[T.3.0]	0.0148	0.002	6.342	0.000	0.010	0.019
isit_sis[T.4.0]	0.0083	0.004	1.921	0.055	-0.000	0.017
yakit_1[T.2.0]	-0.0094	0.002	-5.463	0.000	-0.013	-0.006
yakit_1[T.3.0]	-0.0219	0.003	-8.693	0.000	-0.027	-0.017
yakit_1[T.4.0]	0.0146	0.003	5.375	0.000	0.009	0.020
yakit_1[T.5.0]	0.0191	0.003	5.954	0.000	0.013	0.025
yakit_1[T.6.0]	-0.0098	0.003	-3.359	0.001	-0.016	-0.004
yakit_2[T.2.0]	-0.0183	0.003	-6.125	0.000	-0.024	-0.012
yakit_2[T.3.0]	-0.0178	0.018	-1.010	0.312	-0.052	0.017
yakit_2[T.4.0]	-0.0033	0.005	-0.624	0.533	-0.014	0.007
yakit_2[T.5.0]	-0.0023	0.004	-0.566	0.572	-0.010	0.006
yakit_2[T.6.0]	-0.0010	0.005	-0.183	0.855	-0.011	0.009
yakit_3[T.2.0]	-0.0658	0.017	-3.978	0.000	-0.098	-0.033
yakit_3[T.3.0]	-0.0652	0.029	-2.272	0.023	-0.121	-0.009
yakit_3[T.4.0]	-0.0146	0.013	-1.122	0.262	-0.040	0.011
yakit_3[T.5.0]	0.0065	0.012	0.566	0.571	-0.016	0.029
yakit_3[T.6.0]	0.0173	0.013	1.329	0.184	-0.008	0.043
yakit_mut[T.2.0]	0.0179	0.009	1.917	0.055	-0.000	0.036
yakit_mut[T.3.0]	-0.0282	0.006	-4.723	0.000	-0.040	-0.017

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.4.0]	-0.0054	0.004	-1.398	0.162	-0.013	0.002
yakit_mut[T.5.0]	0.0055	0.006	0.885	0.376	-0.007	0.018
yakit_mut[T.6.0]	0.0322	0.007	4.718	0.000	0.019	0.046
yakit_mut[T.7.0]	-0.0120	0.033	-0.360	0.719	-0.077	0.053
yakit_mut[T.8.0]	0.0351	0.019	1.875	0.061	-0.002	0.072
yakit_su[T.2.0]	-0.0023	0.006	-0.409	0.683	-0.014	0.009
yakit_su[T.3.0]	0.0253	0.005	4.690	0.000	0.015	0.036
yakit_su[T.4.0]	-0.0080	0.004	-2.234	0.025	-0.015	-0.001
yakit_su[T.5.0]	-0.0026	0.003	-0.858	0.391	-0.008	0.003
yakit_su[T.6.0]	0.0112	0.004	3.064	0.002	0.004	0.018
yakit_su[T.7.0]	0.0073	0.003	2.254	0.024	0.001	0.014
yakit_su[T.8.0]	0.0232	0.010	2.323	0.020	0.004	0.043
zor_egiti[T.2.0]	-0.0185	0.003	-5.603	0.000	-0.025	-0.012
zor_egiti[T.3.0]	-0.0131	0.003	-4.248	0.000	-0.019	-0.007
zor_egiti[T.4.0]	-0.0136	0.003	-4.010	0.000	-0.020	-0.007
banyo[T.1.0]	-0.0054	0.003	-1.922	0.055	-0.011	0.000
sauna[T.1.0]	0.0792	0.014	5.483	0.000	0.051	0.107
jakuzi[T.1.0]	0.0175	0.007	2.541	0.011	0.004	0.031
tuvalet[T.1.0]	-0.0161	0.002	-9.141	0.000	-0.020	-0.013
mutfak[T.1.0]	0.0218	0.003	6.354	0.000	0.015	0.028
copogut[T.1.0]	-0.0107	0.009	-1.207	0.227	-0.028	0.007
tabanisi[T.1.0]	0.0199	0.007	2.872	0.004	0.006	0.034
su_sis[T.1.0]	0.0277	0.003	10.297	0.000	0.022	0.033
elektri[T.1.0]	0.3541	0.039	9.014	0.000	0.277	0.431
sicaksu[T.1.0]	0.0022	0.001	1.690	0.091	-0.000	0.005
kabloyay[T.1.0]	0.0072	0.002	3.028	0.002	0.003	0.012
asansor[T.1.0]	0.0078	0.002	4.722	0.000	0.005	0.011
garaj[T.1.0]	0.0128	0.002	5.879	0.000	0.009	0.017
havuz[T.1.0]	0.0316	0.007	4.816	0.000	0.019	0.044
jenrator[T.1.0]	0.0046	0.005	0.926	0.355	-0.005	0.014
apartman[T.1.0]	0.0204	0.002	12.565	0.000	0.017	0.024
yazlik[T.1.0]	0.0373	0.003	12.232	0.000	0.031	0.043
tarla[T.1.0]	0.0314	0.001	24.071	0.000	0.029	0.034
bag[T.1.0]	0.0195	0.002	12.126	0.000	0.016	0.023
sera[T.1.0]	0.0464	0.005	9.139	0.000	0.036	0.056
arsa[T.1.0]	0.0106	0.002	4.906	0.000	0.006	0.015
dukkan[T.1.0]	0.0725	0.002	36.395	0.000	0.069	0.076
otel[T.1.0]	-0.0105	0.020	-0.537	0.591	-0.049	0.028
diger[T.1.0]	0.0900	0.008	11.463	0.000	0.075	0.105
kirkntkd[T.2.0]	-0.0247	0.001	-18.872	0.000	-0.027	-0.022
yas	0.0027	0.000	8.874	0.000	0.002	0.003
ailegrup	1.325e-05	3.14e-06	4.224	0.000	7.1e-06	1.94e-05
egitim	0.0003	0.000	1.041	0.298	-0.000	0.001
sure_yil	8.568e-05	1.05e-05	8.155	0.000	6.51e-05	0.000

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
kac_kisi	-2.124e-05	3.08e-06	-6.890	0.000	-2.73e-05	-1.52e-05
ucrn_ay	-2.615e-06	2.31e-06	-1.130	0.258	-7.15e-06	1.92e-06
ucra_ay	-0.0002	1.87e-05	-13.016	0.000	-0.000	-0.000
ucrn_yil	3.12e-07	1.38e-07	2.262	0.024	4.16e-08	5.82e-07
ikr_yil	4.838e-06	7.62e-07	6.351	0.000	3.34e-06	6.33e-06
prim_yil	2.022e-06	6.27e-07	3.225	0.001	7.93e-07	3.25e-06
konayni	2.873e-06	2.65e-06	1.083	0.279	-2.33e-06	8.07e-06
ulsayni	-4.348e-06	2.47e-06	-1.764	0.078	-9.18e-06	4.84e-07
egsayni	4.514e-07	6.13e-06	0.074	0.941	-1.16e-05	1.25e-05
yemayni	-6.321e-06	2.25e-06	-2.814	0.005	-1.07e-05	-1.92e-06
kresayni	1.703e-06	1.44e-05	0.118	0.906	-2.65e-05	2.99e-05
giyayni	6.626e-06	1.22e-05	0.541	0.588	-1.74e-05	3.06e-05
gidaayni	-4.732e-06	4.79e-06	-0.988	0.323	-1.41e-05	4.65e-06
digayni	-3.215e-06	3.92e-06	-0.820	0.412	-1.09e-05	4.46e-06
calsur_e	0.0008	0.000	4.120	0.000	0.000	0.001
ekgla_ay	-0.0001	2.72e-05	-4.024	0.000	-0.000	-5.62e-05
ekgln_yl	1.18e-06	3.05e-07	3.868	0.000	5.82e-07	1.78e-06
ekgla_yl	1.605e-05	7.94e-06	2.021	0.043	4.84e-07	3.16e-05
calay_yil	0.0008	0.000	3.951	0.000	0.000	0.001
digln_yl	5.605e-06	5.88e-07	9.539	0.000	4.45e-06	6.76e-06
digla_yl	5.09e-06	4.83e-06	1.054	0.292	-4.38e-06	1.46e-05
gmkn_ay	3.533e-06	4.21e-06	0.839	0.401	-4.72e-06	1.18e-05
gmka_ay	0.0005	0.000	1.153	0.249	-0.000	0.001
gmkn_yl	-3.083e-06	2.8e-07	-11.024	0.000	-3.63e-06	-2.53e-06
gmka_yl	9.616e-08	4.09e-07	0.235	0.814	-7.05e-07	8.98e-07
mulka_ay	0.0003	0.000	0.950	0.342	-0.000	0.001
banka_yl	-1.696e-06	2.36e-07	-7.178	0.000	-2.16e-06	-1.23e-06
doviz_yl	-2.58e-06	1.98e-06	-1.305	0.192	-6.45e-06	1.29e-06
nema_yl	-2.797e-06	3.9e-07	-7.171	0.000	-3.56e-06	-2.03e-06
temet_yl	-2.739e-06	5.37e-07	-5.097	0.000	-3.79e-06	-1.69e-06
kar_yl	-5.121e-06	2.18e-06	-2.344	0.019	-9.4e-06	-8.39e-07
toprn_yl	1.171e-06	5.68e-07	2.060	0.039	5.71e-08	2.28e-06
topra_yl	-2.694e-07	5.08e-06	-0.053	0.958	-1.02e-05	9.69e-06
trsn_ay	-1.389e-05	3.14e-06	-4.419	0.000	-2.01e-05	-7.73e-06
emekl_yl	8.796e-07	2.8e-07	3.140	0.002	3.31e-07	1.43e-06
viade_yl	-8.435e-06	5.69e-06	-1.484	0.138	-1.96e-05	2.71e-06
yasli_yl	1.001e-05	2.57e-06	3.903	0.000	4.98e-06	1.5e-05
sosy_yl	6.884e-06	2.68e-06	2.571	0.010	1.64e-06	1.21e-05
hasta_yl	5.61e-06	2.71e-06	2.073	0.038	3.06e-07	1.09e-05
dul_yl	3.359e-07	6.24e-07	0.538	0.591	-8.88e-07	1.56e-06
gazi_yl	5.748e-06	1.44e-06	3.989	0.000	2.92e-06	8.57e-06
burs_yl	1.33e-06	1.42e-06	0.935	0.350	-1.46e-06	4.12e-06
issiz_yl	8.288e-06	2.94e-06	2.816	0.005	2.52e-06	1.41e-05
ddestek_yl	4.483e-07	8.26e-07	0.543	0.587	-1.17e-06	2.07e-06

Table C.11 (Continued) : Quantile Regression Coefficients for 80th Quantile.

	Coef.	Std Err.	t	Prob.	[.025	.975]
dayni_yl	3.431e-05	4.06e-06	8.456	0.000	2.64e-05	4.23e-05
ydemk_yl	2.513e-06	6.46e-07	3.893	0.000	1.25e-06	3.78e-06
yddav_yl	1.907e-06	7.52e-07	2.537	0.011	4.34e-07	3.38e-06
ydayn_yl	-6.213e-06	5.13e-06	-1.211	0.226	-1.63e-05	3.84e-06
dgnfk_yl	-2.174e-06	5.34e-07	-4.074	0.000	-3.22e-06	-1.13e-06
dgbrs_yl	-4.964e-06	1.12e-06	-4.446	0.000	-7.15e-06	-2.78e-06
dsemk_yl	5e-07	1.5e-06	0.333	0.739	-2.44e-06	3.44e-06
dgayn_yl	2.644e-05	1.53e-06	17.229	0.000	2.34e-05	2.94e-05
topga_ay	4.643e-05	5.78e-06	8.031	0.000	3.51e-05	5.78e-05
topgelay	1.693e-06	1.74e-07	9.713	0.000	1.35e-06	2.03e-06
topga_yl	2.704e-06	1.26e-06	2.145	0.032	2.33e-07	5.18e-06
topgelyl	4.241e-06	4.84e-08	87.708	0.000	4.15e-06	4.34e-06
indeks_arl	0.4964	0.013	36.971	0.000	0.470	0.523
kira_ay	-3.613e-05	6.12e-06	-5.900	0.000	-4.81e-05	-2.41e-05
loj_fert	0.0147	0.013	1.158	0.247	-0.010	0.040
izafi_k	-0.0001	5.04e-06	-22.511	0.000	-0.000	-0.000
konutsah	2.771e-06	1.58e-06	1.756	0.079	-3.21e-07	5.86e-06
kon_sure	8.914e-05	0.000	0.497	0.619	-0.000	0.000
kon_borc	1.139e-07	1.07e-08	10.599	0.000	9.28e-08	1.35e-07
kondeger	1.967e-07	1.57e-08	12.532	0.000	1.66e-07	2.27e-07
otur_sur	-0.0001	4.91e-05	-2.209	0.027	-0.000	-1.22e-05

Table C.12 : Quantile Regression Results w/ FS - 80th Quantile.

Dep. Variable:	tasarruf_orani	**Pseudo R-squared:**	0.1563
Model:	QuantReg	**Bandwidth:**	0.009921
Method:	Least Squares	**Sparsity:**	0.6074
Date:	Paz, 21 Tem 2019	**No. Observations:**	371613
Time:	19:04:21	**Df Residuals:**	371357

Quantile Regression Coefficients w/ FS for 80th Quantile.

	Coef.	**Std Err.**	**t**	**Prob.**	**[.025**	**.975]**
Intercept	0.3763	0.050	7.593	0.000	0.279	0.473
year[T.2003]	0.0068	0.002	3.383	0.001	0.003	0.011
year[T.2004]	-0.0137	0.003	-4.580	0.000	-0.020	-0.008
year[T.2005]	-0.0095	0.003	-3.773	0.000	-0.014	-0.005
year[T.2006]	-0.0041	0.002	-1.628	0.104	-0.009	0.001
year[T.2007]	-0.0183	0.003	-7.220	0.000	-0.023	-0.013
year[T.2008]	-0.0239	0.003	-9.103	0.000	-0.029	-0.019
year[T.2009]	-0.0210	0.003	-6.413	0.000	-0.027	-0.015
year[T.2010]	-0.0254	0.003	-7.835	0.000	-0.032	-0.019
year[T.2011]	-0.0215	0.003	-6.626	0.000	-0.028	-0.015
year[T.2012]	-0.0239	0.003	-7.403	0.000	-0.030	-0.018
year[T.2013]	-0.0261	0.003	-8.049	0.000	-0.032	-0.020
year[T.2014]	-0.0095	0.003	-2.964	0.003	-0.016	-0.003
year[T.2015]	0.0071	0.003	2.180	0.029	0.001	0.013
year[T.2016]	0.0107	0.003	3.320	0.001	0.004	0.017
cinsiyet[T.2]	-0.0006	0.001	-0.542	0.588	-0.003	0.002
yakinlik[T.1.0]	0.0072	0.002	3.980	0.000	0.004	0.011
yakinlik[T.2.0]	0.0012	0.002	0.514	0.607	-0.003	0.006
yakinlik[T.3.0]	-0.0052	0.003	-1.571	0.116	-0.012	0.001
yakinlik[T.4.0]	-0.0077	0.005	-1.591	0.112	-0.017	0.002
yakinlik[T.5.0]	-0.0117	0.007	-1.618	0.106	-0.026	0.002
yakinlik[T.6.0]	0.0064	0.003	1.830	0.067	-0.000	0.013
yakinlik[T.7.0]	0.0069	0.003	2.105	0.035	0.000	0.013
yakinlik[T.8.0]	-0.0044	0.004	-1.061	0.289	-0.013	0.004
yakinlik[T.9.0]	0.0042	0.007	0.586	0.558	-0.010	0.018
yakinlik[T.10.0]	0.0126	0.012	1.085	0.278	-0.010	0.035
sagliks[T.2.0]	0.0352	0.003	11.370	0.000	0.029	0.041
sagliks[T.3.0]	-0.0106	0.002	-6.614	0.000	-0.014	-0.007
sagliks[T.4.0]	0.0125	0.001	9.746	0.000	0.010	0.015
ogrenci[T.2.0]	0.0020	0.002	1.153	0.249	-0.001	0.005
egitim_devam[T.2]	-0.0010	0.004	-0.273	0.785	-0.008	0.006
egitim_devam[T.3]	-0.0083	0.005	-1.664	0.096	-0.018	0.001
egitim_devam[T.4]	0.0033	0.007	0.492	0.623	-0.010	0.017
egitim_devam[T.5]	-0.0084	0.006	-1.462	0.144	-0.020	0.003
egitim_devam[T.6]	-0.0253	0.009	-2.711	0.007	-0.044	-0.007
egitim_devam[T.7]	-0.0404	0.032	-1.282	0.200	-0.102	0.021
egitim_devam[T.8]	0.0131	0.063	0.209	0.835	-0.110	0.136

Table C.12 (Continued) : Quantile Regression Coefficients w/ FS for .80

	Coef.	Std Err.	t	Prob.	[.025	.975]
egitim_devam[T.9]	-0.0998	0.132	-0.754	0.451	-0.359	0.160
egitim_devam[T.999]	0.0025	0.002	1.006	0.314	-0.002	0.007
medenidr[T.2.0]	-0.0192	0.002	-8.378	0.000	-0.024	-0.015
medenidr[T.3.0]	0.0037	0.003	1.127	0.260	-0.003	0.010
medenidr[T.4.0]	0.0033	0.004	0.832	0.405	-0.005	0.011
is_arama[T.2]	0.0003	0.003	0.104	0.917	-0.005	0.006
is_arama[T.999]	0.0048	0.003	1.401	0.161	-0.002	0.011
isneden[T.2.0]	0.0189	0.016	1.166	0.244	-0.013	0.051
isneden[T.3.0]	0.0335	0.016	2.074	0.038	0.002	0.065
isneden[T.4.0]	0.0421	0.016	2.598	0.009	0.010	0.074
isneden[T.5.0]	0.1136	0.028	4.069	0.000	0.059	0.168
isneden[T.6.0]	0.0472	0.016	2.884	0.004	0.015	0.079
isneden[T.7.0]	0.0470	0.016	2.862	0.004	0.015	0.079
isneden[T.10.0]	0.0595	0.018	3.316	0.001	0.024	0.095
isneden[T.11.0]	0.0146	0.017	0.880	0.379	-0.018	0.047
sekkod[T.2.0]	0.0259	0.009	2.853	0.004	0.008	0.044
sekkod[T.3.0]	-0.0016	0.004	-0.421	0.673	-0.009	0.006
sekkod[T.4.0]	-0.0054	0.006	-0.840	0.401	-0.018	0.007
sekkod[T.5.0]	0.0024	0.004	0.581	0.561	-0.006	0.011
sekkod[T.6.0]	0.0091	0.004	2.524	0.012	0.002	0.016
sekkod[T.7.0]	0.0058	0.004	1.301	0.193	-0.003	0.015
sekkod[T.8.0]	0.0124	0.004	2.825	0.005	0.004	0.021
sekkod[T.9.0]	-0.0083	0.007	-1.201	0.230	-0.022	0.005
sekkod[T.10.0]	0.0004	0.007	0.059	0.953	-0.012	0.013
sekkod[T.11.0]	0.0059	0.004	1.332	0.183	-0.003	0.015
sekkod[T.12.0]	0.0022	0.005	0.454	0.650	-0.007	0.012
sekkod[T.13.0]	-0.0009	0.005	-0.185	0.853	-0.011	0.009
sekkod[T.14.0]	0.0110	0.005	2.437	0.015	0.002	0.020
sgk[T.2.0]	-0.0221	0.015	-1.512	0.130	-0.051	0.007
sgk[T.3.0]	-0.0086	0.002	-3.757	0.000	-0.013	-0.004
is_statu[T.2]	0.0111	0.004	3.040	0.002	0.004	0.018
is_statu[T.3]	0.0253	0.009	2.942	0.003	0.008	0.042
is_statu[T.999]	-0.0093	0.003	-3.524	0.000	-0.014	-0.004
koopera[T.2]	-0.0072	0.006	-1.196	0.232	-0.019	0.005
koopera[T.999]	-0.0009	0.006	-0.148	0.882	-0.012	0.010
sir_kon[T.2]	0.0089	0.009	0.992	0.321	-0.009	0.026
sir_kon[T.999]	0.0235	0.009	2.728	0.006	0.007	0.040
isdur[T.2.0]	0.0152	0.004	3.966	0.000	0.008	0.023
isdur[T.3.0]	0.0216	0.004	5.300	0.000	0.014	0.030
isdur[T.4.0]	0.0370	0.003	11.611	0.000	0.031	0.043
isdur[T.5.0]	0.0337	0.004	8.494	0.000	0.026	0.041
ist_tesk[T.2.0]	-0.0022	0.004	-0.496	0.620	-0.011	0.006
ist_tesk[T.3.0]	0.0023	0.005	0.497	0.619	-0.007	0.011
guc_kul[T.2]	0.0004	0.003	0.125	0.901	-0.006	0.007
guc_kul[T.3]	-0.0021	0.003	-0.741	0.459	-0.008	0.003

Table C.12 (Continued) : Quantile Regression Coefficients w/ FS for .80

	Coef.	Std Err.	t	Prob.	[.025	.975]
guc_kul[T.999]	0.0120	0.003	3.953	0.000	0.006	0.018
isfaal[T.2.0]	-0.0092	0.004	-2.167	0.030	-0.018	-0.001
mulk_dur[T.2.0]	-0.0146	0.004	-3.429	0.001	-0.023	-0.006
mulk_dur[T.3.0]	-0.0246	0.006	-4.290	0.000	-0.036	-0.013
ekis_yl[T.2.0]	-0.0097	0.006	-1.667	0.096	-0.021	0.002
ekis_ay[T.2.0]	0.0233	0.007	3.476	0.001	0.010	0.036
sekkod_e[T.2.0]	0.0793	0.098	0.806	0.421	-0.114	0.272
sekkod_e[T.3.0]	0.0705	0.018	3.899	0.000	0.035	0.106
sekkod_e[T.4.0]	-0.0673	0.061	-1.109	0.267	-0.186	0.052
sekkod_e[T.5.0]	0.0614	0.016	3.917	0.000	0.031	0.092
sekkod_e[T.6.0]	0.0865	0.015	5.744	0.000	0.057	0.116
sekkod_e[T.7.0]	0.1331	0.019	6.949	0.000	0.096	0.171
sekkod_e[T.8.0]	0.1556	0.019	8.097	0.000	0.118	0.193
sekkod_e[T.9.0]	0.1193	0.038	3.137	0.002	0.045	0.194
sekkod_e[T.10.0]	0.0115	0.039	0.291	0.771	-0.066	0.089
sekkod_e[T.11.0]	0.1289	0.021	6.197	0.000	0.088	0.170
sekkod_e[T.12.0]	0.2196	0.026	8.468	0.000	0.169	0.270
sekkod_e[T.13.0]	0.0111	0.044	0.254	0.800	-0.075	0.097
sekkod_e[T.14.0]	0.1089	0.015	7.089	0.000	0.079	0.139
meslek_e[T.1.0]	0.0171	0.014	1.234	0.217	-0.010	0.044
meslek_e[T.2.0]	-0.1059	0.022	-4.900	0.000	-0.148	-0.064
meslek_e[T.3.0]	-0.0106	0.020	-0.533	0.594	-0.049	0.028
meslek_e[T.4.0]	0.0122	0.044	0.280	0.780	-0.073	0.098
meslek_e[T.5.0]	0.0123	0.016	0.768	0.443	-0.019	0.044
meslek_e[T.6.0]	0.1053	0.015	6.902	0.000	0.075	0.135
meslek_e[T.7.0]	0.0266	0.015	1.815	0.070	-0.002	0.055
meslek_e[T.8.0]	-0.0021	0.016	-0.131	0.896	-0.034	0.029
meslek_e[T.9.0]	0.0678	0.012	5.529	0.000	0.044	0.092
isdur_e[T.2.0]	0.0673	0.013	5.283	0.000	0.042	0.092
isdur_e[T.3.0]	0.0326	0.019	1.760	0.078	-0.004	0.069
isdur_e[T.4.0]	0.0646	0.013	5.132	0.000	0.040	0.089
isdur_e[T.5.0]	0.0602	0.015	4.014	0.000	0.031	0.090
cl_neden[T.2.0]	0.0061	0.008	0.732	0.464	-0.010	0.023
cl_neden[T.3.0]	0.0168	0.005	3.730	0.000	0.008	0.026
cl_neden[T.4.0]	0.0265	0.009	3.029	0.002	0.009	0.044
cl_neden[T.5.0]	0.0420	0.009	4.577	0.000	0.024	0.060
cl_neden[T.6.0]	0.0132	0.007	1.981	0.048	0.000	0.026
cl_neden[T.7.0]	0.0113	0.007	1.549	0.121	-0.003	0.026
ayni_is[T.2.0]	-0.0094	0.009	-1.043	0.297	-0.027	0.008
sekkod_d[T.2.0]	-0.0007	0.034	-0.021	0.983	-0.067	0.066
sekkod_d[T.3.0]	0.0220	0.012	1.823	0.068	-0.002	0.046
sekkod_d[T.4.0]	0.0276	0.021	1.330	0.184	-0.013	0.068
sekkod_d[T.5.0]	0.0096	0.012	0.829	0.407	-0.013	0.032
sekkod_d[T.6.0]	0.0097	0.013	0.774	0.439	-0.015	0.034
sekkod_d[T.7.0]	0.0116	0.015	0.782	0.434	-0.017	0.041

Table C.12 (Continued) : Quantile Regression Coefficients w/ FS for .80

	Coef.	Std Err.	t	Prob.	[.025	.975]
sekkod_d[T.8.0]	0.0258	0.014	1.817	0.069	-0.002	0.054
sekkod_d[T.9.0]	0.0020	0.029	0.068	0.946	-0.056	0.060
sekkod_d[T.10.0]	0.0312	0.023	1.385	0.166	-0.013	0.075
sekkod_d[T.11.0]	0.0169	0.017	0.992	0.321	-0.016	0.050
sekkod_d[T.12.0]	0.0110	0.022	0.509	0.611	-0.031	0.053
sekkod_d[T.13.0]	0.0125	0.024	0.516	0.606	-0.035	0.060
sekkod_d[T.14.0]	0.0102	0.015	0.693	0.488	-0.019	0.039
meslek_d[T.2.0]	-0.0589	0.019	-3.082	0.002	-0.096	-0.021
meslek_d[T.3.0]	-0.0393	0.017	-2.315	0.021	-0.073	-0.006
meslek_d[T.4.0]	-0.0495	0.017	-2.907	0.004	-0.083	-0.016
meslek_d[T.5.0]	-0.0305	0.014	-2.133	0.033	-0.059	-0.002
meslek_d[T.6.0]	-0.0394	0.020	-1.992	0.046	-0.078	-0.001
meslek_d[T.7.0]	-0.0483	0.014	-3.376	0.001	-0.076	-0.020
meslek_d[T.8.0]	-0.0398	0.016	-2.567	0.010	-0.070	-0.009
meslek_d[T.9.0]	-0.0507	0.014	-3.500	0.000	-0.079	-0.022
isdur_d[T.2.0]	0.0028	0.007	0.378	0.705	-0.012	0.017
isdur_d[T.3.0]	-0.0019	0.024	-0.077	0.938	-0.049	0.045
isdur_d[T.4.0]	-0.0002	0.011	-0.018	0.985	-0.021	0.021
isdur_d[T.5.0]	0.0123	0.015	0.824	0.410	-0.017	0.042
hhtipi[T.2.0]	-0.0136	0.001	-10.003	0.000	-0.016	-0.011
hhtipi[T.3.0]	-0.0259	0.001	-18.027	0.000	-0.029	-0.023
hhtipi[T.4.0]	0.0170	0.002	9.219	0.000	0.013	0.021
hhtipi[T.5.0]	-0.0088	0.002	-5.822	0.000	-0.012	-0.006
hhtipi[T.6.0]	0.0119	0.002	5.160	0.000	0.007	0.016
hhtipi[T.7.0]	0.0213	0.005	4.512	0.000	0.012	0.031
konuttip[T.2.0]	-0.0001	0.002	-0.062	0.950	-0.004	0.003
konuttip[T.3.0]	-0.0103	0.002	-6.723	0.000	-0.013	-0.007
konuttip[T.4.0]	-0.0473	0.013	-3.564	0.000	-0.073	-0.021
mulkiyet[T.2.0]	-0.0213	0.001	-15.023	0.000	-0.024	-0.019
mulkiyet[T.3.0]	0.0359	0.003	11.369	0.000	0.030	0.042
mulkiyet[T.4.0]	-0.0035	0.002	-2.197	0.028	-0.007	-0.000
kira_sek[T.2.0]	0.0120	0.011	1.047	0.295	-0.010	0.034
yapituru[T.2.0]	0.0955	0.030	3.138	0.002	0.036	0.155
yapituru[T.3.0]	-0.0168	0.004	-4.031	0.000	-0.025	-0.009
yapituru[T.4.0]	-0.0009	0.002	-0.391	0.696	-0.006	0.004
yapituru[T.5.0]	-0.0110	0.003	-3.837	0.000	-0.017	-0.005
yapituru[T.6.0]	-0.0002	0.002	-0.106	0.916	-0.004	0.004
yapituru[T.7.0]	0.0096	0.003	3.269	0.001	0.004	0.015
yapituru[T.8.0]	0.0005	0.011	0.047	0.963	-0.021	0.022
zemin_sal[T.2.0]	0.0066	0.002	3.387	0.001	0.003	0.010
zemin_sal[T.3.0]	0.0119	0.002	5.584	0.000	0.008	0.016
zemin_sal[T.4.0]	-0.0009	0.002	-0.395	0.693	-0.006	0.004
zemin_sal[T.5.0]	-0.0104	0.005	-2.294	0.022	-0.019	-0.002
zemin_sal[T.6.0]	0.0104	0.002	4.339	0.000	0.006	0.015
zemin_sal[T.7.0]	0.0187	0.005	4.034	0.000	0.010	0.028

Table C.12 (Continued) : Quantile Regression Coefficients w/ FS for .80

	Coef.	Std Err.	t	Prob.	[.025	.975]
zemin_sal[T.8.0]	0.0132	0.013	0.983	0.326	-0.013	0.040
zemin_sal[T.9.0]	0.0129	0.005	2.615	0.009	0.003	0.023
zemin_sal[T.98.0]	0.0097	0.010	1.003	0.316	-0.009	0.029
zemin_mut[T.2.0]	0.0156	0.004	3.535	0.000	0.007	0.024
zemin_mut[T.3.0]	0.0032	0.003	1.072	0.284	-0.003	0.009
zemin_mut[T.4.0]	0.0191	0.005	3.551	0.000	0.009	0.030
zemin_mut[T.5.0]	0.0424	0.011	3.844	0.000	0.021	0.064
zemin_mut[T.6.0]	0.0108	0.004	2.770	0.006	0.003	0.018
zemin_mut[T.7.0]	0.0076	0.006	1.335	0.182	-0.004	0.019
zemin_mut[T.8.0]	0.0128	0.011	1.166	0.243	-0.009	0.034
zemin_mut[T.9.0]	0.0119	0.007	1.712	0.087	-0.002	0.026
zemin_mut[T.98.0]	0.0502	0.011	4.437	0.000	0.028	0.072
zemin_ban[T.2.0]	-0.0457	0.026	-1.728	0.084	-0.098	0.006
zemin_ban[T.3.0]	-0.0667	0.021	-3.146	0.002	-0.108	-0.025
zemin_ban[T.4.0]	-0.1152	0.024	-4.772	0.000	-0.162	-0.068
zemin_ban[T.5.0]	-0.0374	0.056	-0.662	0.508	-0.148	0.073
zemin_ban[T.6.0]	-0.0575	0.021	-2.702	0.007	-0.099	-0.016
zemin_ban[T.7.0]	-0.0712	0.022	-3.294	0.001	-0.114	-0.029
zemin_ban[T.8.0]	-0.0811	0.024	-3.402	0.001	-0.128	-0.034
zemin_ban[T.9.0]	-0.0707	0.023	-3.115	0.002	-0.115	-0.026
zemin_ban[T.98.0]	-0.0724	0.039	-1.866	0.062	-0.148	0.004
isit_sis[T.2.0]	-0.0063	0.002	-3.248	0.001	-0.010	-0.002
isit_sis[T.3.0]	-0.0023	0.002	-1.087	0.277	-0.006	0.002
isit_sis[T.4.0]	0.0174	0.004	4.414	0.000	0.010	0.025
yakit_1[T.2.0]	-0.0077	0.002	-4.930	0.000	-0.011	-0.005
yakit_1[T.3.0]	-0.0248	0.002	-11.017	0.000	-0.029	-0.020
yakit_1[T.4.0]	-0.0018	0.002	-0.742	0.458	-0.007	0.003
yakit_1[T.5.0]	0.0080	0.003	2.728	0.006	0.002	0.014
yakit_1[T.6.0]	-0.0268	0.003	-9.933	0.000	-0.032	-0.022
yakit_2[T.2.0]	-0.0088	0.003	-3.191	0.001	-0.014	-0.003
yakit_2[T.3.0]	-0.0027	0.016	-0.163	0.870	-0.035	0.030
yakit_2[T.4.0]	-0.0134	0.005	-2.656	0.008	-0.023	-0.004
yakit_2[T.5.0]	0.0078	0.004	2.065	0.039	0.000	0.015
yakit_2[T.6.0]	0.0113	0.005	2.335	0.020	0.002	0.021
yakit_3[T.2.0]	0.0202	0.014	1.432	0.152	-0.007	0.048
yakit_3[T.3.0]	-0.0142	0.024	-0.578	0.563	-0.062	0.034
yakit_3[T.4.0]	-0.0376	0.012	-3.193	0.001	-0.061	-0.015
yakit_3[T.5.0]	-0.0030	0.010	-0.289	0.772	-0.023	0.017
yakit_3[T.6.0]	0.0082	0.012	0.695	0.487	-0.015	0.031
yakit_mut[T.2.0]	-0.0038	0.009	-0.437	0.662	-0.021	0.013
yakit_mut[T.3.0]	-0.0224	0.005	-4.228	0.000	-0.033	-0.012
yakit_mut[T.4.0]	-0.0270	0.003	-7.724	0.000	-0.034	-0.020
yakit_mut[T.5.0]	-0.0070	0.006	-1.262	0.207	-0.018	0.004
yakit_mut[T.6.0]	0.0064	0.006	1.018	0.309	-0.006	0.019
yakit_mut[T.7.0]	0.0249	0.029	0.870	0.384	-0.031	0.081

Table C.12 (Continued) : Quantile Regression Coefficients w/ FS for .80

	Coef.	Std Err.	t	Prob.	[.025	.975]
yakit_mut[T.8.0]	0.0201	0.018	1.141	0.254	-0.014	0.055
yakit_su[T.2.0]	-0.0024	0.005	-0.459	0.646	-0.013	0.008
yakit_su[T.3.0]	0.0007	0.005	0.139	0.889	-0.009	0.010
yakit_su[T.4.0]	-0.0205	0.003	-6.220	0.000	-0.027	-0.014
yakit_su[T.5.0]	-0.0127	0.003	-4.637	0.000	-0.018	-0.007
yakit_su[T.6.0]	-0.0087	0.003	-2.633	0.008	-0.015	-0.002
yakit_su[T.7.0]	-0.0087	0.003	-2.949	0.003	-0.014	-0.003
yakit_su[T.8.0]	0.0005	0.009	0.052	0.958	-0.018	0.019
zor_egiti[T.2.0]	-0.0101	0.003	-3.313	0.001	-0.016	-0.004
zor_egiti[T.3.0]	-0.0062	0.003	-2.156	0.031	-0.012	-0.001
zor_egiti[T.4.0]	-0.0063	0.003	-2.023	0.043	-0.012	-0.000
banyo[T.1.0]	-0.0099	0.003	-3.815	0.000	-0.015	-0.005
sauna[T.1.0]	-0.0242	0.014	-1.743	0.081	-0.051	0.003
jakuzi[T.1.0]	-0.0469	0.006	-7.312	0.000	-0.059	-0.034
tuvalet[T.1.0]	-0.0165	0.002	-10.245	0.000	-0.020	-0.013
mutfak[T.1.0]	0.0098	0.003	3.122	0.002	0.004	0.016
copogut[T.1.0]	-0.0371	0.008	-4.706	0.000	-0.053	-0.022
tabanisi[T.1.0]	0.0303	0.006	4.818	0.000	0.018	0.043
su_sis[T.1.0]	0.0249	0.002	10.080	0.000	0.020	0.030
elektri[T.1.0]	-0.0663	0.030	-2.227	0.026	-0.125	-0.008
sicaksu[T.1.0]	-0.0108	0.001	-8.936	0.000	-0.013	-0.008
kabloyay[T.1.0]	-0.0122	0.002	-5.677	0.000	-0.016	-0.008
asansor[T.1.0]	-0.0054	0.002	-3.547	0.000	-0.008	-0.002
garaj[T.1.0]	-0.0080	0.002	-3.971	0.000	-0.012	-0.004
havuz[T.1.0]	-0.0290	0.006	-4.763	0.000	-0.041	-0.017
jenrator[T.1.0]	-0.0385	0.005	-8.191	0.000	-0.048	-0.029
apartman[T.1.0]	0.0116	0.001	7.982	0.000	0.009	0.014
yazlik[T.1.0]	-0.0086	0.003	-3.018	0.003	-0.014	-0.003
tarla[T.1.0]	0.0095	0.001	7.887	0.000	0.007	0.012
bag[T.1.0]	0.0071	0.001	4.791	0.000	0.004	0.010
sera[T.1.0]	0.0251	0.005	5.267	0.000	0.016	0.034
arsa[T.1.0]	-0.0096	0.002	-4.819	0.000	-0.014	-0.006
dukkan[T.1.0]	0.0162	0.002	8.706	0.000	0.013	0.020
otel[T.1.0]	-0.0382	0.016	-2.434	0.015	-0.069	-0.007
diger[T.1.0]	0.0089	0.007	1.246	0.213	-0.005	0.023
kirkntkd[T.2.0]	-0.0142	0.001	-11.896	0.000	-0.017	-0.012
izafi_k	-0.0003	3.36e-06	-78.175	0.000	-0.000	-0.000
ceptel	-0.0202	0.000	-47.730	0.000	-0.021	-0.019
yilkulgel	6.344e-06	1.52e-08	416.479	0.000	6.31e-06	6.37e-06

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
year	cinsiyet	-0.0033	0.0439	
year	yakinlik	-0.0291	0.0000	
year	sagliks	-0.1989	0.0000	
year	gunengel	-0.6704	0.0000	
year	calengel	-0.6704	0.0000	
year	ogrenci	0.0712	0.0000	
year	egitim_devam	-0.1206	0.0000	
year	medenidr	0.0757	0.0000	
year	calis_ay	-0.0513	0.0000	
year	is_arama	0.0625	0.0000	
year	isneden	0.0136	0.0000	
year	isbasi	0.0625	0.0000	
year	calis_yl	0.3883	0.0000	
year	sekkod	0.0461	0.0000	
year	meslek	-0.0512	0.0000	
year	cal_kisi	0.2849	0.0000	
year	sgk	0.1745	0.0000	
year	is_statu	0.2849	0.0000	
year	huk_dur	0.3427	0.0000	
year	koopera	-0.0879	0.0000	
year	sir_kon	0.1106	0.0000	
year	isdur	-0.047	0.0000	
year	ist_tesk	0.0206	0.0000	
year	guc_kul	0.1378	0.0000	
year	sendika	0.1801	0.0000	
year	isfaal	-0.0232	0.0000	
year	mulk_dur	0.027	0.0000	
year	ekis_yl	0.0434	0.0000	
year	ekis_ay	-0.0173	0.0000	
year	sekkod_e	-0.0257	0.0000	
year	meslek_e	0.0035	0.0337	
year	isdur_e	0.0202	0.0000	
year	cl_neden	-0.0224	0.0000	
year	ayni_is	-0.0155	0.0000	
year	sekkod_d	0.0093	0.0000	
year	meslek_d	0.0061	0.0002	
year	isdur_d	-0.0291	0.0000	
year	hhtipi	0.001	0.5591	
year	konuttip	0.0587	0.0000	
year	mulkiyet	0.1192	0.0000	
year	kira_sek	-0.0204	0.0000	
year	yapituru	-0.3497	0.0000	
year	zemin_sal	0.1176	0.0000	
year	zemin_oda	0.1343	0.0000	
year	zemin_mut	0.1136	0.0000	
year	zemin_ban	0.1382	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
year	isit_sis	0.2549	0.0000	
year	yakit_1	0.16	0.0000	
year	yakit_2	-0.0693	0.0000	
year	yakit_3	-0.0627	0.0000	
year	yakit_mut	-0.3126	0.0000	
year	yakit_su	-0.2238	0.0000	
year	zor_alis	-0.1255	0.0000	
year	zor_banka	-0.2825	0.0000	
year	zor_posta	-0.2514	0.0000	
year	zor_ulas	-0.1429	0.0000	
year	zor_saglik	-0.176	0.0000	
year	zor_egiti	-0.079	0.0000	
year	banyo	0.0542	0.0000	
year	sauna	0.0135	0.0000	
year	jakuzi	0.0262	0.0000	
year	tuvalet	0.03	0.0000	
year	mutfak	0.0475	0.0000	
year	copogut	-0.009	0.0000	
year	kalorif	0.207	0.0000	
year	tabanisi	0.0229	0.0000	
year	su_sis	0.1354	0.0000	
year	elektri	0.0111	0.0000	
year	dog_gaz	0.2731	0.0000	
year	sicaksu	0.2554	0.0000	
year	kabloyay	0.0104	0.0000	
year	asansor	0.1401	0.0000	
year	garaj	0.0963	0.0000	
year	havuz	0.0588	0.0000	
year	jenrator	0.0941	0.0000	
year	apartman	0.0261	0.0000	
year	yazlik	0.018	0.0000	
year	tarla	0.0469	0.0000	
year	bag	0.0375	0.0000	
year	sera	0.0152	0.0000	
year	arsa	0.0475	0.0000	
year	dukkan	-0.0055	0.0008	
year	otel	-0.0013	0.4303	
year	diger	-0.0568	0.0000	
year	kirkntkd	0.2651	0.0000	
cinsiyet	yakinlik	0.1635	0.0000	
cinsiyet	sagliks	-0.0316	0.0000	
cinsiyet	gunengel	-0.004	0.0146	
cinsiyet	calengel	-0.004	0.0145	
cinsiyet	ogrenci	0.0327	0.0000	
cinsiyet	egitim_devam	0.0162	0.0000	
cinsiyet	medenidr	0.1017	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
cinsiyet	calis_ay	0.299	0.0000	
cinsiyet	is_arama	-0.308	0.0000	
cinsiyet	isneden	0.015	0.0000	
cinsiyet	isbasi	-0.308	0.0000	
cinsiyet	calis_yl	-0.2392	0.0000	
cinsiyet	sekkod	-0.0712	0.0000	
cinsiyet	meslek	0.299	0.0000	
cinsiyet	cal_kisi	0.2394	0.0000	
cinsiyet	sgk	0.2287	0.0000	
cinsiyet	is_statu	0.2395	0.0000	
cinsiyet	huk_dur	0.1804	0.0000	
cinsiyet	koopera	0.082	0.0000	
cinsiyet	sir_kon	0.1126	0.0000	
cinsiyet	isdur	0.0422	0.0000	
cinsiyet	ist_tesk	-0.0578	0.0000	
cinsiyet	guc_kul	0.1927	0.0000	
cinsiyet	sendika	0.2065	0.0000	
cinsiyet	isfaal	0.1418	0.0000	
cinsiyet	mulk_dur	0.0565	0.0000	
cinsiyet	ekis_yl	0.0982	0.0000	
cinsiyet	ekis_ay	-0.0483	0.0000	
cinsiyet	sekkod_e	-0.0588	0.0000	
cinsiyet	meslek_e	0.015	0.0000	
cinsiyet	isdur_e	0.0512	0.0000	
cinsiyet	cl_neden	-0.0291	0.0000	
cinsiyet	ayni_is	-0.0305	0.0000	
cinsiyet	sekkod_d	-0.0156	0.0000	
cinsiyet	meslek_d	0.012	0.0000	
cinsiyet	isdur_d	0	0.9806	
cinsiyet	hhtipi	0.0589	0.0000	
cinsiyet	konuttip	0	0.9903	
cinsiyet	mulkiyet	-0.0042	0.0106	
cinsiyet	kira_sek	0.0018	0.2673	
cinsiyet	yapituru	0.0026	0.1150	
cinsiyet	zemin_sal	-0.0026	0.1181	
cinsiyet	zemin_oda	-0.0028	0.0897	
cinsiyet	zemin_mut	-0.0012	0.4488	
cinsiyet	zemin_ban	-0.0019	0.2438	
cinsiyet	isit_sis	0.0009	0.5656	
cinsiyet	yakit_1	-0.0012	0.4467	
cinsiyet	yakit_2	0.0034	0.0377	
cinsiyet	yakit_3	-0.0007	0.6705	
cinsiyet	yakit_mut	0.0008	0.6107	
cinsiyet	yakit_su	-0.0022	0.1709	
cinsiyet	zor_alis	-0.0017	0.3128	
cinsiyet	zor_banka	0.0006	0.7258	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
cinsiyet	zor_posta	-0.0005	0.7771	
cinsiyet	zor_ulas	-0.0009	0.5971	
cinsiyet	zor_saglik	-0.0012	0.4611	
cinsiyet	zor_egiti	-0.0016	0.3240	
cinsiyet	banyo	-0.0022	0.1728	
cinsiyet	sauna	0.0001	0.9648	
cinsiyet	jakuzi	0.0012	0.4515	
cinsiyet	tuvalet	-0.0006	0.7331	
cinsiyet	mutfak	-0.0007	0.6551	
cinsiyet	copogut	0.0014	0.3980	
cinsiyet	kalorif	0.0021	0.1985	
cinsiyet	tabanisi	0.0007	0.6562	
cinsiyet	su_sis	-0.0027	0.1000	
cinsiyet	elektri	0.0009	0.5737	
cinsiyet	dog_gaz	0.0013	0.4239	
cinsiyet	sicaksu	-0.0021	0.2079	
cinsiyet	kabloyay	0.0036	0.0276	
cinsiyet	asansor	0.0018	0.2632	
cinsiyet	garaj	-0.0011	0.5151	
cinsiyet	havuz	0.0006	0.7276	
cinsiyet	jenrator	0.0001	0.9427	
cinsiyet	apartman	0.0043	0.0095	
cinsiyet	yazlik	0.0021	0.2070	
cinsiyet	tarla	0.0001	0.9398	
cinsiyet	bag	0.0003	0.8760	
cinsiyet	sera	-0.0012	0.4563	
cinsiyet	arsa	-0.0024	0.1430	
cinsiyet	dukkan	-0.0022	0.1702	
cinsiyet	otel	0.0001	0.9570	
cinsiyet	diger	-0.0013	0.4267	
cinsiyet	kirkntkd	-0.0054	0.0011	
yakinlik	sagliks	0.1679	0.0000	
yakinlik	gunengel	0.0826	0.0000	
yakinlik	calengel	0.0826	0.0000	
yakinlik	ogrenci	-0.2196	0.0000	
yakinlik	egitim_devam	-0.1379	0.0000	
yakinlik	medenidr	-0.1675	0.0000	
yakinlik	calis_ay	0.2761	0.0000	
yakinlik	is_arama	0.0265	0.0000	
yakinlik	isneden	-0.0369	0.0000	
yakinlik	isbasi	0.0265	0.0000	
yakinlik	calis_yl	0.0024	0.1460	
yakinlik	sekkod	-0.0831	0.0000	
yakinlik	meslek	0.2762	0.0000	
yakinlik	cal_kisi	0.2012	0.0000	
yakinlik	sgk	0.1844	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
yakinlik	is_statu	0.2012	0.0000	
yakinlik	huk_dur	0.1387	0.0000	
yakinlik	koopera	0.0575	0.0000	
yakinlik	sir_kon	0.0736	0.0000	
yakinlik	isdur	-0.0017	0.3064	
yakinlik	ist_tesk	-0.0557	0.0000	
yakinlik	guc_kul	0.1459	0.0000	
yakinlik	sendika	0.1564	0.0000	
yakinlik	isfaal	0.1623	0.0000	
yakinlik	mulk_dur	0.039	0.0000	
yakinlik	ekis_yl	0.0836	0.0000	
yakinlik	ekis_ay	-0.0375	0.0000	
yakinlik	sekkod_e	-0.0549	0.0000	
yakinlik	meslek_e	0.01	0.0000	
yakinlik	isdur_e	0.0514	0.0000	
yakinlik	cl_neden	-0.0307	0.0000	
yakinlik	ayni_is	-0.0162	0.0000	
yakinlik	sekkod_d	-0.0111	0.0000	
yakinlik	meslek_d	0.0051	0.0020	
yakinlik	isdur_d	-0.0066	0.0001	
yakinlik	hhtipi	0.2984	0.0000	
yakinlik	konuttip	-0.1119	0.0000	
yakinlik	mulkiyet	-0.0735	0.0000	
yakinlik	kira_sek	-0.0117	0.0000	
yakinlik	yapituru	0.0539	0.0000	
yakinlik	zemin_sal	0.0237	0.0000	
yakinlik	zemin_oda	0.0219	0.0000	
yakinlik	zemin_mut	0.0125	0.0000	
yakinlik	zemin_ban	0.0271	0.0000	
yakinlik	isit_sis	-0.0759	0.0000	
yakinlik	yakit_1	-0.0468	0.0000	
yakinlik	yakit_2	0.0405	0.0000	
yakinlik	yakit_3	-0.009	0.0000	
yakinlik	yakit_mut	0.0218	0.0000	
yakinlik	yakit_su	0.0218	0.0000	
yakinlik	zor_alis	-0.0566	0.0000	
yakinlik	zor_banka	-0.048	0.0000	
yakinlik	zor_posta	-0.0475	0.0000	
yakinlik	zor_ulas	-0.0483	0.0000	
yakinlik	zor_saglik	-0.0545	0.0000	
yakinlik	zor_egiti	-0.032	0.0000	
yakinlik	banyo	-0.0405	0.0000	
yakinlik	sauna	-0.0033	0.0450	
yakinlik	jakuzi	-0.0128	0.0000	
yakinlik	tuvalet	-0.0627	0.0000	
yakinlik	mutfak	-0.0325	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
yakinlik	copogut	-0.009	0.0000	
yakinlik	kalorif	-0.0783	0.0000	
yakinlik	tabanisi	-0.013	0.0000	
yakinlik	su_sis	-0.0596	0.0000	
yakinlik	elektri	0.0029	0.0800	
yakinlik	dog_gaz	-0.0654	0.0000	
yakinlik	sicaksu	-0.0824	0.0000	
yakinlik	kabloyay	-0.0377	0.0000	
yakinlik	asansor	-0.0612	0.0000	
yakinlik	garaj	-0.0064	0.0001	
yakinlik	havuz	-0.0135	0.0000	
yakinlik	jenrator	-0.0134	0.0000	
yakinlik	apartman	-0.0694	0.0000	
yakinlik	yazlik	-0.028	0.0000	
yakinlik	tarla	0.1193	0.0000	
yakinlik	bag	0.049	0.0000	
yakinlik	sera	0.0205	0.0000	
yakinlik	arsa	-0.0085	0.0000	
yakinlik	dukkan	0.007	0.0000	
yakinlik	otel	-0.0032	0.0480	
yakinlik	diger	0.0046	0.0047	
yakinlik	kirkntkd	-0.0826	0.0000	
sagliks	gunengel	0.1651	0.0000	
sagliks	calengel	0.1651	0.0000	
sagliks	ogrenci	-0.0172	0.0000	
sagliks	egitim_devam	0.0324	0.0000	
sagliks	medenidr	-0.0738	0.0000	
sagliks	calis_ay	0.0144	0.0000	
sagliks	is_arama	0.0464	0.0000	
sagliks	isneden	0.0238	0.0000	
sagliks	isbasi	0.0464	0.0000	
sagliks	calis_yl	-0.0109	0.0000	
sagliks	sekkod	-0.1511	0.0000	
sagliks	meslek	0.0147	0.0000	
sagliks	cal_kisi	-0.0605	0.0000	
sagliks	sgk	0.2086	0.0000	
sagliks	is_statu	-0.0603	0.0000	
sagliks	huk_dur	-0.1372	0.0000	
sagliks	koopera	0.0092	0.0000	
sagliks	sir_kon	0.0825	0.0000	
sagliks	isdur	0.1795	0.0000	
sagliks	ist_tesk	0.0888	0.0000	
sagliks	guc_kul	0.0575	0.0000	
sagliks	sendika	0.0559	0.0000	
sagliks	isfaal	-0.0477	0.0000	
sagliks	mulk_dur	0.0293	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
sagliks	ekis_yl	-0.0266	0.0000	
sagliks	ekis_ay	0.0324	0.0000	
sagliks	sekkod_e	0.0002	0.9213	
sagliks	meslek_e	0.0386	0.0000	
sagliks	isdur_e	-0.0225	0.0000	
sagliks	cl_neden	0.0474	0.0000	
sagliks	ayni_is	0.0311	0.0000	
sagliks	sekkod_d	-0.0128	0.0000	
sagliks	meslek_d	0.0085	0.0000	
sagliks	isdur_d	0.0399	0.0000	
sagliks	hhtipi	0.1362	0.0000	
sagliks	konuttip	-0.2861	0.0000	
sagliks	mulkiyet	-0.0357	0.0000	
sagliks	kira_sek	0.0051	0.0020	
sagliks	yapituru	0.2357	0.0000	
sagliks	zemin_sal	0.0548	0.0000	
sagliks	zemin_oda	0.0583	0.0000	
sagliks	zemin_mut	0.0465	0.0000	
sagliks	zemin_ban	0.0596	0.0000	
sagliks	isit_sis	-0.247	0.0000	
sagliks	yakit_1	-0.1004	0.0000	
sagliks	yakit_2	0.1254	0.0000	
sagliks	yakit_3	-0.009	0.0000	
sagliks	yakit_mut	0.1072	0.0000	
sagliks	yakit_su	0.0742	0.0000	
sagliks	zor_alis	-0.0833	0.0000	
sagliks	zor_banka	-0.0494	0.0000	
sagliks	zor_posta	-0.0557	0.0000	
sagliks	zor_ulas	-0.0774	0.0000	
sagliks	zor_saglik	-0.0765	0.0000	
sagliks	zor_egiti	-0.0581	0.0000	
sagliks	banyo	-0.2077	0.0000	
sagliks	sauna	-0.0104	0.0000	
sagliks	jakuzi	-0.0314	0.0000	
sagliks	tuvalet	-0.272	0.0000	
sagliks	mutfak	-0.1643	0.0000	
sagliks	copogut	-0.0221	0.0000	
sagliks	kalorif	-0.2669	0.0000	
sagliks	tabanisi	-0.0329	0.0000	
sagliks	su_sis	-0.2391	0.0000	
sagliks	elektri	-0.0142	0.0000	
sagliks	dog_gaz	-0.2113	0.0000	
sagliks	sicaksu	-0.3415	0.0000	
sagliks	kabloyay	-0.0899	0.0000	
sagliks	asansor	-0.1597	0.0000	
sagliks	garaj	-0.0843	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
sagliks	havuz	-0.0309	0.0000	
sagliks	jenrator	-0.0471	0.0000	
sagliks	apartman	-0.2405	0.0000	
sagliks	yazlik	-0.0663	0.0000	
sagliks	tarla	0.15	0.0000	
sagliks	bag	0.0243	0.0000	
sagliks	sera	-0.0007	0.6571	
sagliks	arsa	-0.06	0.0000	
sagliks	dukkan	-0.0576	0.0000	
sagliks	otel	-0.0089	0.0000	
sagliks	diger	-0.0005	0.7668	
sagliks	kirkntkd	-0.2039	0.0000	
gunengel	calengel	1	0.0000	YES
gunengel	ogrenci	0.0151	0.0000	
gunengel	egitim_devam	0.1944	0.0000	
gunengel	medenidr	-0.0301	0.0000	
gunengel	calis_ay	0.091	0.0000	
gunengel	is_arama	0.026	0.0000	
gunengel	isneden	-0.0304	0.0000	
gunengel	isbasi	0.026	0.0000	
gunengel	calis_yl	-0.1321	0.0000	
gunengel	sekkod	-0.0433	0.0000	
gunengel	meslek	0.091	0.0000	
gunengel	cal_kisi	-0.0694	0.0000	
gunengel	sgk	-0.0296	0.0000	
gunengel	is_statu	-0.0693	0.0000	
gunengel	huk_dur	-0.1607	0.0000	
gunengel	koopera	0.1489	0.0000	
gunengel	sir_kon	-0.0217	0.0000	
gunengel	isdur	0.0234	0.0000	
gunengel	ist_tesk	-0.0259	0.0000	
gunengel	guc_kul	0.0337	0.0000	
gunengel	sendika	-0.0233	0.0000	
gunengel	isfaal	0.0266	0.0000	
gunengel	mulk_dur	-0.0034	0.0386	
gunengel	ekis_yl	0.017	0.0000	
gunengel	ekis_ay	-0.0144	0.0000	
gunengel	sekkod_e	-0.0112	0.0000	
gunengel	meslek_e	-0.0034	0.0400	
gunengel	isdur_e	0.0128	0.0000	
gunengel	cl_neden	-0.0055	0.0009	
gunengel	ayni_is	-0.0034	0.0407	
gunengel	sekkod_d	-0.0154	0.0000	
gunengel	meslek_d	0.012	0.0000	
gunengel	isdur_d	-0.0064	0.0001	
gunengel	hhtipi	-0.0039	0.0172	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
gunengel	konuttip	-0.0528	0.0000	
gunengel	mulkiyet	-0.0679	0.0000	
gunengel	kira_sek	0.0136	0.0000	
gunengel	yapituru	0.199	0.0000	
gunengel	zemin_sal	-0.0891	0.0000	
gunengel	zemin_oda	-0.0945	0.0000	
gunengel	zemin_mut	-0.0829	0.0000	
gunengel	zemin_ban	-0.1493	0.0000	
gunengel	isit_sis	-0.1783	0.0000	
gunengel	yakit_1	-0.117	0.0000	
gunengel	yakit_2	0.0573	0.0000	
gunengel	yakit_3	0.0343	0.0000	
gunengel	yakit_mut	0.1752	0.0000	
gunengel	yakit_su	0.129	0.0000	
gunengel	zor_alis	0.0693	0.0000	
gunengel	zor_banka	0.1633	0.0000	
gunengel	zor_posta	0.1434	0.0000	
gunengel	zor_ulas	0.0772	0.0000	
gunengel	zor_saglik	0.099	0.0000	
gunengel	zor_egiti	0.0399	0.0000	
gunengel	banyo	-0.0557	0.0000	
gunengel	sauna	-0.0136	0.0000	
gunengel	jakuzi	-0.0183	0.0000	
gunengel	tuvalet	-0.0168	0.0000	
gunengel	mutfak	-0.0665	0.0000	
gunengel	copogut	0.0172	0.0000	
gunengel	kalorif	-0.1452	0.0000	
gunengel	tabanisi	-0.0259	0.0000	
gunengel	su_sis	-0.1157	0.0000	
gunengel	elektri	-0.0093	0.0000	
gunengel	dog_gaz	-0.1759	0.0000	
gunengel	sicaksu	-0.2097	0.0000	
gunengel	kabloyay	-0.0091	0.0000	
gunengel	asansor	-0.0986	0.0000	
gunengel	garaj	-0.0522	0.0000	
gunengel	havuz	-0.034	0.0000	
gunengel	jenrator	-0.0543	0.0000	
gunengel	apartman	-0.0437	0.0000	
gunengel	yazlik	-0.0114	0.0000	
gunengel	tarla	-0.0287	0.0000	
gunengel	bag	-0.0189	0.0000	
gunengel	sera	-0.0076	0.0000	
gunengel	arsa	-0.0375	0.0000	
gunengel	dukkan	-0.0051	0.0021	
gunengel	otel	-0.0002	0.8892	
gunengel	diger	0.0445	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
gunengel	kirkntkd	-0.0559	0.0000	
calengel	ogrenci	0.0151	0.0000	
calengel	egitim_devam	0.1944	0.0000	
calengel	medenidr	-0.0302	0.0000	
calengel	calis_ay	0.0909	0.0000	
calengel	is_arama	0.026	0.0000	
calengel	isneden	-0.0304	0.0000	
calengel	isbasi	0.026	0.0000	
calengel	calis_yl	-0.1321	0.0000	
calengel	sekkod	-0.0433	0.0000	
calengel	meslek	0.0909	0.0000	
calengel	cal_kisi	-0.0694	0.0000	
calengel	sgk	-0.0296	0.0000	
calengel	is_statu	-0.0694	0.0000	
calengel	huk_dur	-0.1608	0.0000	
calengel	koopera	0.1489	0.0000	
calengel	sir_kon	-0.0217	0.0000	
calengel	isdur	0.0234	0.0000	
calengel	ist_tesk	-0.0259	0.0000	
calengel	guc_kul	0.0337	0.0000	
calengel	sendika	-0.0233	0.0000	
calengel	isfaal	0.0266	0.0000	
calengel	mulk_dur	-0.0034	0.0385	
calengel	ekis_yl	0.017	0.0000	
calengel	ekis_ay	-0.0144	0.0000	
calengel	sekkod_e	-0.0112	0.0000	
calengel	meslek_e	-0.0034	0.0400	
calengel	isdur_e	0.0128	0.0000	
calengel	cl_neden	-0.0055	0.0009	
calengel	ayni_is	-0.0034	0.0407	
calengel	sekkod_d	-0.0154	0.0000	
calengel	meslek_d	0.012	0.0000	
calengel	isdur_d	-0.0064	0.0001	
calengel	hhtipi	-0.0039	0.0169	
calengel	konuttip	-0.0528	0.0000	
calengel	mulkiyet	-0.0679	0.0000	
calengel	kira_sek	0.0136	0.0000	
calengel	yapituru	0.199	0.0000	
calengel	zemin_sal	-0.0891	0.0000	
calengel	zemin_oda	-0.0945	0.0000	
calengel	zemin_mut	-0.0829	0.0000	
calengel	zemin_ban	-0.1493	0.0000	
calengel	isit_sis	-0.1783	0.0000	
calengel	yakit_1	-0.117	0.0000	
calengel	yakit_2	0.0573	0.0000	
calengel	yakit_3	0.0343	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
calengel	yakit_mut	0.1752	0.0000	
calengel	yakit_su	0.129	0.0000	
calengel	zor_alis	0.0693	0.0000	
calengel	zor_banka	0.1633	0.0000	
calengel	zor_posta	0.1434	0.0000	
calengel	zor_ulas	0.0772	0.0000	
calengel	zor_saglik	0.099	0.0000	
calengel	zor_egiti	0.0399	0.0000	
calengel	banyo	-0.0557	0.0000	
calengel	sauna	-0.0136	0.0000	
calengel	jakuzi	-0.0183	0.0000	
calengel	tuvalet	-0.0167	0.0000	
calengel	mutfak	-0.0665	0.0000	
calengel	copogut	0.0172	0.0000	
calengel	kalorif	-0.1452	0.0000	
calengel	tabanisi	-0.0259	0.0000	
calengel	su_sis	-0.1157	0.0000	
calengel	elektri	-0.0093	0.0000	
calengel	dog_gaz	-0.1759	0.0000	
calengel	sicaksu	-0.2097	0.0000	
calengel	kabloyay	-0.0091	0.0000	
calengel	asansor	-0.0986	0.0000	
calengel	garaj	-0.0522	0.0000	
calengel	havuz	-0.034	0.0000	
calengel	jenrator	-0.0543	0.0000	
calengel	apartman	-0.0437	0.0000	
calengel	yazlik	-0.0114	0.0000	
calengel	tarla	-0.0287	0.0000	
calengel	bag	-0.0189	0.0000	
calengel	sera	-0.0076	0.0000	
calengel	arsa	-0.0375	0.0000	
calengel	dukkan	-0.0051	0.0021	
calengel	otel	-0.0002	0.8890	
calengel	diger	0.0445	0.0000	
calengel	kirkntkd	-0.0559	0.0000	
ogrenci	egitim_devam	0.6121	0.0000	
ogrenci	medenidr	0.2499	0.0000	
ogrenci	calis_ay	-0.2979	0.0000	
ogrenci	is_arama	-0.0423	0.0000	
ogrenci	isneden	0.2412	0.0000	
ogrenci	isbasi	-0.0423	0.0000	
ogrenci	calis_yl	0.0216	0.0000	
ogrenci	sekkod	0.0097	0.0000	
ogrenci	meslek	-0.2979	0.0000	
ogrenci	cal_kisi	-0.2272	0.0000	
ogrenci	sgk	-0.1418	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
ogrenci	is_statu	-0.2272	0.0000	
ogrenci	huk_dur	-0.1806	0.0000	
ogrenci	koopera	-0.089	0.0000	
ogrenci	sir_kon	-0.0732	0.0000	
ogrenci	isdur	0.0896	0.0000	
ogrenci	ist_tesk	0.0656	0.0000	
ogrenci	guc_kul	-0.1351	0.0000	
ogrenci	sendika	-0.1431	0.0000	
ogrenci	isfaal	-0.106	0.0000	
ogrenci	mulk_dur	-0.0235	0.0000	
ogrenci	ekis_yl	-0.0627	0.0000	
ogrenci	ekis_ay	0.0344	0.0000	
ogrenci	sekkod_e	0.0349	0.0000	
ogrenci	meslek_e	0.0047	0.0042	
ogrenci	isdur_e	-0.0353	0.0000	
ogrenci	cl_neden	0.0187	0.0000	
ogrenci	ayni_is	0.0179	0.0000	
ogrenci	sekkod_d	0.0015	0.3711	
ogrenci	meslek_d	0.0009	0.5865	
ogrenci	isdur_d	0.0141	0.0000	
ogrenci	hhtipi	0.0411	0.0000	
ogrenci	konuttip	-0.0224	0.0000	
ogrenci	mulkiyet	-0.0285	0.0000	
ogrenci	kira_sek	0.001	0.5601	
ogrenci	yapituru	-0.0067	0.0000	
ogrenci	zemin_sal	0.0149	0.0000	
ogrenci	zemin_oda	0.017	0.0000	
ogrenci	zemin_mut	0.0143	0.0000	
ogrenci	zemin_ban	0.0067	0.0000	
ogrenci	isit_sis	0.0132	0.0000	
ogrenci	yakit_1	0.0007	0.6499	
ogrenci	yakit_2	-0.0116	0.0000	
ogrenci	yakit_3	-0.0066	0.0001	
ogrenci	yakit_mut	-0.022	0.0000	
ogrenci	yakit_su	-0.019	0.0000	
ogrenci	zor_alis	-0.0106	0.0000	
ogrenci	zor_banka	-0.0193	0.0000	
ogrenci	zor_posta	-0.0185	0.0000	
ogrenci	zor_ulas	-0.0157	0.0000	
ogrenci	zor_saglik	-0.0159	0.0000	
ogrenci	zor_egiti	-0.0173	0.0000	
ogrenci	banyo	0.0014	0.3924	
ogrenci	sauna	0.0022	0.1848	
ogrenci	jakuzi	-0.0022	0.1848	
ogrenci	tuvalet	-0.0083	0.0000	
ogrenci	mutfak	-0.0012	0.4684	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
ogrenci	copogut	0.0009	0.5941	
ogrenci	kalorif	-0.0015	0.3472	
ogrenci	tabanisi	-0.0032	0.0522	
ogrenci	su_sis	0.0073	0.0000	
ogrenci	elektri	-0.0026	0.1196	
ogrenci	dog_gaz	0.016	0.0000	
ogrenci	sicaksu	0.0134	0.0000	
ogrenci	kabloyay	0.0068	0.0000	
ogrenci	asansor	-0.002	0.2211	
ogrenci	garaj	0.0046	0.0052	
ogrenci	havuz	0.003	0.0657	
ogrenci	jenrator	0.0038	0.0190	
ogrenci	apartman	-0.0053	0.0011	
ogrenci	yazlik	0.0093	0.0000	
ogrenci	tarla	0.0396	0.0000	
ogrenci	bag	0.0255	0.0000	
ogrenci	sera	0.0069	0.0000	
ogrenci	arsa	0.0024	0.1408	
ogrenci	dukkan	0.0008	0.6050	
ogrenci	otel	-0.0009	0.5668	
ogrenci	diger	-0.0052	0.0015	
ogrenci	kirkntkd	0.0055	0.0008	
egitim_devam	medenidr	0.1124	0.0000	
egitim_devam	calis_ay	-0.169	0.0000	
egitim_devam	is_arama	-0.0765	0.0000	
egitim_devam	isneden	0.1237	0.0000	
egitim_devam	isbasi	-0.0765	0.0000	
egitim_devam	calis_yl	-0.0602	0.0000	
egitim_devam	sekkod	-0.0023	0.1578	
egitim_devam	meslek	-0.169	0.0000	
egitim_devam	cal_kisi	-0.1319	0.0000	
egitim_devam	sgk	-0.0776	0.0000	
egitim_devam	is_statu	-0.1319	0.0000	
egitim_devam	huk_dur	-0.1196	0.0000	
egitim_devam	koopera	-0.0309	0.0000	
egitim_devam	sir_kon	-0.0429	0.0000	
egitim_devam	isdur	0.0614	0.0000	
egitim_devam	ist_tesk	0.0368	0.0000	
egitim_devam	guc_kul	-0.0687	0.0000	
egitim_devam	sendika	-0.0768	0.0000	
egitim_devam	isfaal	-0.0648	0.0000	
egitim_devam	mulk_dur	-0.0142	0.0000	
egitim_devam	ekis_yl	-0.0367	0.0000	
egitim_devam	ekis_ay	0.0198	0.0000	
egitim_devam	sekkod_e	0.0196	0.0000	
egitim_devam	meslek_e	0.0036	0.0297	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
egitim_devam	isdur_e	-0.0196	0.0000	
egitim_devam	cl_neden	0.0084	0.0000	
egitim_devam	ayni_is	0.0085	0.0000	
egitim_devam	sekkod_d	-0.008	0.0000	
egitim_devam	meslek_d	0.0094	0.0000	
egitim_devam	isdur_d	0.0082	0.0000	
egitim_devam	hhtipi	0.019	0.0000	
egitim_devam	konuttip	-0.0374	0.0000	
egitim_devam	mulkiyet	-0.046	0.0000	
egitim_devam	kira_sek	0.0054	0.0009	
egitim_devam	yapituru	0.0874	0.0000	
egitim_devam	zemin_sal	-0.0034	0.0393	
egitim_devam	zemin_oda	-0.0023	0.1684	
egitim_devam	zemin_mut	-0.0014	0.3977	
egitim_devam	zemin_ban	-0.0248	0.0000	
egitim_devam	isit_sis	-0.033	0.0000	
egitim_devam	yakit_1	-0.0288	0.0000	
egitim_devam	yakit_2	0.006	0.0003	
egitim_devam	yakit_3	0.0064	0.0001	
egitim_devam	yakit_mut	0.0326	0.0000	
egitim_devam	yakit_su	0.0281	0.0000	
egitim_devam	zor_alis	0.0143	0.0000	
egitim_devam	zor_banka	0.0421	0.0000	
egitim_devam	zor_posta	0.0337	0.0000	
egitim_devam	zor_ulas	0.0067	0.0000	
egitim_devam	zor_saglik	0.0187	0.0000	
egitim_devam	zor_egiti	-0.008	0.0000	
egitim_devam	banyo	-0.0088	0.0000	
egitim_devam	sauna	0.0025	0.1272	
egitim_devam	jakuzi	-0.0038	0.0200	
egitim_devam	tuvalet	-0.0066	0.0001	
egitim_devam	mutfak	-0.0091	0.0000	
egitim_devam	copogut	0.0066	0.0001	
egitim_devam	kalorif	-0.0383	0.0000	
egitim_devam	tabanisi	-0.0075	0.0000	
egitim_devam	su_sis	-0.0223	0.0000	
egitim_devam	elektri	-0.0027	0.1048	
egitim_devam	dog_gaz	-0.0369	0.0000	
egitim_devam	sicaksu	-0.0456	0.0000	
egitim_devam	kabloyay	0.0039	0.0185	
egitim_devam	asansor	-0.0272	0.0000	
egitim_devam	garaj	-0.0083	0.0000	
egitim_devam	havuz	-0.0037	0.0235	
egitim_devam	jenrator	-0.0136	0.0000	
egitim_devam	apartman	-0.0104	0.0000	
egitim_devam	yazlik	0.0023	0.1652	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
egitim_devam	tarla	0.0139	0.0000	
egitim_devam	bag	0.0133	0.0000	
egitim_devam	sera	0.0028	0.0889	
egitim_devam	arsa	-0.0104	0.0000	
egitim_devam	dukkan	0.0001	0.9647	
egitim_devam	otel	-0.002	0.2293	
egitim_devam	diger	0.0143	0.0000	
egitim_devam	kirkntkd	-0.0008	0.6085	
medenidr	calis_ay	0.0254	0.0000	
medenidr	is_arama	0.1036	0.0000	
medenidr	isneden	0.1509	0.0000	
medenidr	isbasi	0.1036	0.0000	
medenidr	calis_yl	0.1225	0.0000	
medenidr	sekkod	0.0016	0.3335	
medenidr	meslek	0.0254	0.0000	
medenidr	cal_kisi	0.0298	0.0000	
medenidr	sgk	0.0079	0.0000	
medenidr	is_statu	0.0298	0.0000	
medenidr	huk_dur	0.0427	0.0000	
medenidr	koopera	0.0094	0.0000	
medenidr	sir_kon	0.0396	0.0000	
medenidr	isdur	0.0257	0.0000	
medenidr	ist_tesk	-0.0071	0.0000	
medenidr	guc_kul	0.0447	0.0000	
medenidr	sendika	0.0483	0.0000	
medenidr	isfaal	-0.0758	0.0000	
medenidr	mulk_dur	-0.0019	0.2469	
medenidr	ekis_yl	-0.0215	0.0000	
medenidr	ekis_ay	0.0052	0.0015	
medenidr	sekkod_e	0.018	0.0000	
medenidr	meslek_e	-0.0038	0.0211	
medenidr	isdur_e	-0.0141	0.0000	
medenidr	cl_neden	0.0062	0.0002	
medenidr	ayni_is	-0.0082	0.0000	
medenidr	sekkod_d	0.0024	0.1489	
medenidr	meslek_d	0.0226	0.0000	
medenidr	isdur_d	-0.0247	0.0000	
medenidr	hhtipi	0.1411	0.0000	
medenidr	konuttip	-0.0003	0.8494	
medenidr	mulkiyet	0.0341	0.0000	
medenidr	kira_sek	0.006	0.0002	
medenidr	yapituru	-0.0298	0.0000	
medenidr	zemin_sal	0.0095	0.0000	
medenidr	zemin_oda	0.0107	0.0000	
medenidr	zemin_mut	0.0098	0.0000	
medenidr	zemin_ban	0.0173	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
medenidr	isit_sis	0.0139	0.0000	
medenidr	yakit_1	0.0044	0.0071	
medenidr	yakit_2	-0.0154	0.0000	
medenidr	yakit_3	-0.0025	0.1281	
medenidr	yakit_mut	-0.0204	0.0000	
medenidr	yakit_su	-0.025	0.0000	
medenidr	zor_alis	-0.0137	0.0000	
medenidr	zor_banka	-0.031	0.0000	
medenidr	zor_posta	-0.0276	0.0000	
medenidr	zor_ulas	-0.0152	0.0000	
medenidr	zor_saglik	-0.0185	0.0000	
medenidr	zor_egiti	-0.0131	0.0000	
medenidr	banyo	-0.0011	0.5056	
medenidr	sauna	-0.0043	0.0095	
medenidr	jakuzi	0.0043	0.0093	
medenidr	tuvalet	-0.003	0.0655	
medenidr	mutfak	0.0011	0.5042	
medenidr	copogut	0.0009	0.5628	
medenidr	kalorif	0.0074	0.0000	
medenidr	tabanisi	0.0001	0.9606	
medenidr	su_sis	0.0189	0.0000	
medenidr	elektri	-0.0051	0.0018	
medenidr	dog_gaz	0.0138	0.0000	
medenidr	sicaksu	0.011	0.0000	
medenidr	kabloyay	0.0029	0.0758	
medenidr	asansor	0.011	0.0000	
medenidr	garaj	0.0084	0.0000	
medenidr	havuz	0.0123	0.0000	
medenidr	jenrator	0.0075	0.0000	
medenidr	apartman	-0.0124	0.0000	
medenidr	yazlik	-0.0032	0.0495	
medenidr	tarla	0.0062	0.0002	
medenidr	bag	0.0017	0.3089	
medenidr	sera	0.0036	0.0280	
medenidr	arsa	-0.001	0.5369	
medenidr	dukkan	-0.0082	0.0000	
medenidr	otel	-0.0039	0.0182	
medenidr	diger	-0.006	0.0003	
medenidr	kirkntkd	0.0037	0.0259	
calis_ay	is_arama	-0.6253	0.0000	
calis_ay	isneden	0.1435	0.0000	
calis_ay	isbasi	-0.6253	0.0000	
calis_ay	calis_yl	-0.4853	0.0000	
calis_ay	sekkod	-0.1057	0.0000	
calis_ay	meslek	1	0.0000	YES
calis_ay	cal_kisi	0.7558	0.0000	YES

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
calis_ay	sgk	0.4911	0.0000	
calis_ay	is_statu	0.7558	0.0000	YES
calis_ay	huk_dur	0.5661	0.0000	
calis_ay	koopera	0.3193	0.0000	
calis_ay	sir_kon	0.2573	0.0000	
calis_ay	isdur	-0.2322	0.0000	
calis_ay	ist_tesk	-0.224	0.0000	
calis_ay	guc_kul	0.4928	0.0000	
calis_ay	sendika	0.5206	0.0000	
calis_ay	isfaal	0.287	0.0000	
calis_ay	mulk_dur	0.0677	0.0000	
calis_ay	ekis_yl	0.1883	0.0000	
calis_ay	ekis_ay	-0.1034	0.0000	
calis_ay	sekkod_e	-0.1112	0.0000	
calis_ay	meslek_e	-0.0077	0.0000	
calis_ay	isdur_e	0.1096	0.0000	
calis_ay	cl_neden	0.0113	0.0000	
calis_ay	ayni_is	-0.0658	0.0000	
calis_ay	sekkod_d	-0.0193	0.0000	
calis_ay	meslek_d	0.0099	0.0000	
calis_ay	isdur_d	0.0361	0.0000	
calis_ay	hhtipi	0.0386	0.0000	
calis_ay	konuttip	0.0335	0.0000	
calis_ay	mulkiyet	0.002	0.2294	
calis_ay	kira_sek	0.0071	0.0000	
calis_ay	yapituru	-0.0043	0.0094	
calis_ay	zemin_sal	0.0033	0.0424	
calis_ay	zemin_oda	0.0016	0.3348	
calis_ay	zemin_mut	0.0001	0.9631	
calis_ay	zemin_ban	-0.0024	0.1498	
calis_ay	isit_sis	-0.0016	0.3151	
calis_ay	yakit_1	0.0046	0.0053	
calis_ay	yakit_2	0.0012	0.4571	
calis_ay	yakit_3	0.0011	0.5140	
calis_ay	yakit_mut	0.0273	0.0000	
calis_ay	yakit_su	0.0082	0.0000	
calis_ay	zor_alis	0.0209	0.0000	
calis_ay	zor_banka	0.0343	0.0000	
calis_ay	zor_posta	0.0336	0.0000	
calis_ay	zor_ulas	0.0241	0.0000	
calis_ay	zor_saglik	0.0266	0.0000	
calis_ay	zor_egiti	0.0243	0.0000	
calis_ay	banyo	0.0003	0.8347	
calis_ay	sauna	-0.0017	0.2884	
calis_ay	jakuzi	-0.0066	0.0001	
calis_ay	tuvalet	0.0297	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
calis_ay	mutfak	-0.0001	0.9651	
calis_ay	copogut	-0.0075	0.0000	
calis_ay	kalorif	0.0036	0.0286	
calis_ay	tabanisi	-0.002	0.2265	
calis_ay	su_sis	0.0201	0.0000	
calis_ay	elektri	0.0018	0.2696	
calis_ay	dog_gaz	-0.0053	0.0013	
calis_ay	sicaksu	-0.0171	0.0000	
calis_ay	kabloyay	-0.0051	0.0017	
calis_ay	asansor	-0.01	0.0000	
calis_ay	garaj	-0.0202	0.0000	
calis_ay	havuz	-0.008	0.0000	
calis_ay	jenrator	-0.0161	0.0000	
calis_ay	apartman	0.0375	0.0000	
calis_ay	yazlik	0.0007	0.6529	
calis_ay	tarla	-0.1197	0.0000	
calis_ay	bag	-0.0869	0.0000	
calis_ay	sera	-0.0377	0.0000	
calis_ay	arsa	-0.0154	0.0000	
calis_ay	dukkan	0.0008	0.6279	
calis_ay	otel	-0.0043	0.0095	
calis_ay	diger	-0.0073	0.0000	
calis_ay	kirkntkd	0.0657	0.0000	
is_arama	isneden	-0.2295	0.0000	
is_arama	isbasi	1	0.0000	YES
is_arama	calis_yl	0.7762	0.0000	YES
is_arama	sekkod	0.0661	0.0000	
is_arama	meslek	-0.6253	0.0000	
is_arama	cal_kisi	-0.4726	0.0000	
is_arama	sgk	-0.3071	0.0000	
is_arama	is_statu	-0.4726	0.0000	
is_arama	huk_dur	-0.354	0.0000	
is_arama	koopera	-0.1996	0.0000	
is_arama	sir_kon	-0.1609	0.0000	
is_arama	isdur	0.1452	0.0000	
is_arama	ist_tesk	0.1401	0.0000	
is_arama	guc_kul	-0.3081	0.0000	
is_arama	sendika	-0.3256	0.0000	
is_arama	isfaal	-0.1782	0.0000	
is_arama	mulk_dur	-0.0423	0.0000	
is_arama	ekis_yl	-0.1177	0.0000	
is_arama	ekis_ay	0.0647	0.0000	
is_arama	sekkod_e	0.0695	0.0000	
is_arama	meslek_e	0.0048	0.0034	
is_arama	isdur_e	-0.0685	0.0000	
is_arama	cl_neden	-0.0427	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
is_arama	ayni_is	0.0411	0.0000	
is_arama	sekkod_d	0.0162	0.0000	
is_arama	meslek_d	0.0093	0.0000	
is_arama	isdur_d	-0.0778	0.0000	
is_arama	hhtipi	-0.072	0.0000	
is_arama	konuttip	-0.0539	0.0000	
is_arama	mulkiyet	0.0615	0.0000	
is_arama	kira_sek	-0.0045	0.0058	
is_arama	yapituru	0.0072	0.0000	
is_arama	zemin_sal	0.0096	0.0000	
is_arama	zemin_oda	0.0112	0.0000	
is_arama	zemin_mut	0.0088	0.0000	
is_arama	zemin_ban	0.0189	0.0000	
is_arama	isit_sis	-0.0273	0.0000	
is_arama	yakit_1	-0.0077	0.0000	
is_arama	yakit_2	0.0258	0.0000	
is_arama	yakit_3	-0.0069	0.0000	
is_arama	yakit_mut	-0.0157	0.0000	
is_arama	yakit_su	0.0011	0.5024	
is_arama	zor_alis	-0.0469	0.0000	
is_arama	zor_banka	-0.0613	0.0000	
is_arama	zor_posta	-0.06	0.0000	
is_arama	zor_ulas	-0.0443	0.0000	
is_arama	zor_saglik	-0.0513	0.0000	
is_arama	zor_egiti	-0.0329	0.0000	
is_arama	banyo	-0.0364	0.0000	
is_arama	sauna	-0.0022	0.1899	
is_arama	jakuzi	0.0061	0.0002	
is_arama	tuvalet	-0.0654	0.0000	
is_arama	mutfak	-0.0296	0.0000	
is_arama	copogut	-0.0008	0.6437	
is_arama	kalorif	-0.0318	0.0000	
is_arama	tabanisi	-0.0004	0.7908	
is_arama	su_sis	-0.0569	0.0000	
is_arama	elektri	-0.0043	0.0089	
is_arama	dog_gaz	-0.0219	0.0000	
is_arama	sicaksu	-0.0318	0.0000	
is_arama	kabloyay	-0.0268	0.0000	
is_arama	asansor	-0.0022	0.1774	
is_arama	garaj	0.0108	0.0000	
is_arama	havuz	0.007	0.0000	
is_arama	jenrator	0.0159	0.0000	
is_arama	apartman	-0.0951	0.0000	
is_arama	yazlik	-0.035	0.0000	
is_arama	tarla	0.0946	0.0000	
is_arama	bag	0.0569	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
is_arama	sera	0.0337	0.0000	
is_arama	arsa	-0.0017	0.3063	
is_arama	dukkan	-0.0231	0.0000	
is_arama	otel	-0.0007	0.6882	
is_arama	diger	0.0015	0.3688	
is_arama	kirkntkd	-0.0767	0.0000	
isneden	isbasi	-0.2295	0.0000	
isneden	calis_yl	-0.1645	0.0000	
isneden	sekkod	-0.0152	0.0000	
isneden	meslek	0.1435	0.0000	
isneden	cal_kisi	0.1085	0.0000	
isneden	sgk	0.0705	0.0000	
isneden	is_statu	0.1085	0.0000	
isneden	huk_dur	0.0813	0.0000	
isneden	koopera	0.0458	0.0000	
isneden	sir_kon	0.0369	0.0000	
isneden	isdur	-0.0333	0.0000	
isneden	ist_tesk	-0.0322	0.0000	
isneden	guc_kul	0.0707	0.0000	
isneden	sendika	0.0747	0.0000	
isneden	isfaal	0.0389	0.0000	
isneden	mulk_dur	0.0097	0.0000	
isneden	ekis_yl	0.027	0.0000	
isneden	ekis_ay	-0.0148	0.0000	
isneden	sekkod_e	-0.016	0.0000	
isneden	meslek_e	-0.0011	0.5012	
isneden	isdur_e	0.0157	0.0000	
isneden	cl_neden	0.0325	0.0000	
isneden	ayni_is	-0.0094	0.0000	
isneden	sekkod_d	-0.0309	0.0000	
isneden	meslek_d	-0.0079	0.0000	
isneden	isdur_d	0.0923	0.0000	
isneden	hhtipi	0.1091	0.0000	
isneden	konuttip	-0.0316	0.0000	
isneden	mulkiyet	-0.0386	0.0000	
isneden	kira_sek	0.001	0.5545	
isneden	yapituru	0.0004	0.8150	
isneden	zemin_sal	0.0127	0.0000	
isneden	zemin_oda	0.0154	0.0000	
isneden	zemin_mut	0.0135	0.0000	
isneden	zemin_ban	0.0197	0.0000	
isneden	isit_sis	-0.0159	0.0000	
isneden	yakit_1	-0.0099	0.0000	
isneden	yakit_2	-0.0015	0.3744	
isneden	yakit_3	-0.0022	0.1806	
isneden	yakit_mut	0.0032	0.0543	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
isneden	yakit_su	-0.0049	0.0026	
isneden	zor_alis	-0.0179	0.0000	
isneden	zor_banka	-0.0209	0.0000	
isneden	zor_posta	-0.0206	0.0000	
isneden	zor_ulas	-0.0203	0.0000	
isneden	zor_saglik	-0.0194	0.0000	
isneden	zor_egiti	-0.0169	0.0000	
isneden	banyo	-0.0152	0.0000	
isneden	sauna	-0.003	0.0651	
isneden	jakuzi	-0.0039	0.0171	
isneden	tuvalet	-0.0226	0.0000	
isneden	mutfak	-0.0082	0.0000	
isneden	copogut	-0.0024	0.1423	
isneden	kalorif	-0.0211	0.0000	
isneden	tabanisi	-0.0027	0.1031	
isneden	su_sis	0.0036	0.0294	
isneden	elektri	-0.0019	0.2353	
isneden	dog_gaz	-0.016	0.0000	
isneden	sicaksu	-0.0158	0.0000	
isneden	kabloyay	0.0041	0.0123	
isneden	asansor	-0.0146	0.0000	
isneden	garaj	-0.007	0.0000	
isneden	havuz	-0.0051	0.0019	
isneden	jenrator	-0.0067	0.0000	
isneden	apartman	0.0024	0.1437	
isneden	yazlik	0.0119	0.0000	
isneden	tarla	0.0256	0.0000	
isneden	bag	0.0153	0.0000	
isneden	sera	-0.0037	0.0241	
isneden	arsa	0.0011	0.5143	
isneden	dukkan	0.0042	0.0100	
isneden	otel	0.0011	0.5036	
isneden	diger	-0.0034	0.0397	
isneden	kirkntkd	-0.0107	0.0000	
isbasi	calis_yl	0.7762	0.0000	YES
isbasi	sekkod	0.0661	0.0000	
isbasi	meslek	-0.6253	0.0000	
isbasi	cal_kisi	-0.4726	0.0000	
isbasi	sgk	-0.3071	0.0000	
isbasi	is_statu	-0.4726	0.0000	
isbasi	huk_dur	-0.354	0.0000	
isbasi	koopera	-0.1996	0.0000	
isbasi	sir_kon	-0.1609	0.0000	
isbasi	isdur	0.1452	0.0000	
isbasi	ist_tesk	0.1401	0.0000	
isbasi	guc_kul	-0.3081	0.0000	

Table C.13 : Point BiSerial Correlation for Qualitative Variables

Variable 1	Variable 2	BiSerial Corr.Coeff.	Significance	Drop
isbasi	sendika	-0.3256	0.0000	
isbasi	isfaal	-0.1782	0.0000	
isbasi	mulk_dur	-0.0423	0.0000	
isbasi	ekis_yl	-0.1177	0.0000	
isbasi	ekis_ay	0.0647	0.0000	
isbasi	sekkod_e	0.0695	0.0000	
isbasi	meslek_e	0.0048	0.0034	
isbasi	isdur_e	-0.0685	0.0000	
isbasi	cl_neden	-0.0427	0.0000	
isbasi	ayni_is	0.0411	0.0000	
isbasi	sekkod_d	0.0162	0.0000	
isbasi	meslek_d	0.0093	0.0000	
isbasi	isdur_d	-0.0778	0.0000	
isbasi	hhtipi	-0.072	0.0000	
isbasi	konuttip	-0.0539	0.0000	
isbasi	mulkiyet	0.0615	0.0000	
isbasi	kira_sek	-0.0045	0.0059	
isbasi	yapituru	0.0073	0.0000	
isbasi	zemin_sal	0.0096	0.0000	
isbasi	zemin_oda	0.0112	0.0000	
isbasi	zemin_mut	0.0088	0.0000	
isbasi	zemin_ban	0.0189	0.0000	
isbasi	isit_sis	-0.0273	0.0000	
isbasi	yakit_1	-0.0077	0.0000	
isbasi	yakit_2	0.0258	0.0000	
isbasi	yakit_3	-0.0069	0.0000	
isbasi	yakit_mut	-0.0157	0.0000	
isbasi	yakit_su	0.0011	0.5016	
isbasi	zor_alis	-0.0469	0.0000	
isbasi	zor_banka	-0.0613	0.0000	
isbasi	zor_posta	-0.06	0.0000	
isbasi	zor_ulas	-0.0443	0.0000	
isbasi	zor_saglik	-0.0513	0.0000	
isbasi	zor_egiti	-0.0329	0.0000	

Note: year is accepted as categorical.

Table C.14 : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
yas	ailegrup	0.0816	0.0000	
yas	sagliks2	-0.0043	0.0084	
yas	egitim	0.0472	0.0000	
yas	sure_yil	-0.2449	0.0000	
yas	calsur	-0.2496	0.0000	
yas	kac_kisi	-0.1125	0.0000	
yas	ucrn_ay	0.0595	0.0000	
yas	ucra_ay	0.0251	0.0000	
yas	ucrn_yil	0.0287	0.0000	
yas	ikr_yil	0.0341	0.0000	
yas	prim_yil	0.0173	0.0000	
yas	ucra_yil	0.0473	0.0000	
yas	konayni	0.0287	0.0000	
yas	ulsayni	0.0313	0.0000	
yas	egsayni	0.0148	0.0000	
yas	yemayni	0.0328	0.0000	
yas	kresayni	0.0013	0.4419	
yas	giyayni	0.0096	0.0000	
yas	gidaayni	0.0151	0.0000	
yas	digayni	0.0122	0.0000	
yas	mutsn_ay	0.0251	0.0000	
yas	mutsa_ay	0.0314	0.0000	
yas	mutsn_yl	0.0356	0.0000	
yas	mutsa_yl	0.0415	0.0000	
yas	targ_ay	0.0344	0.0000	
yas	targ_yl	0.0449	0.0000	
yas	calsur_e	0.004	0.0148	
yas	ekgln_ay	0.0027	0.0948	
yas	ekgla_ay	0.0033	0.0447	
yas	ekgln_yl	0.0173	0.0000	
yas	ekgla_yl	0.0114	0.0000	
yas	calay_yil	-0.1834	0.0000	
yas	calsur_d	-0.0083	0.0000	
yas	digln_yl	0.0147	0.0000	
yas	digla_yl	0.002	0.2151	
yas	gmkn_ay	0.0796	0.0000	
yas	gmka_ay	0.0061	0.0002	
yas	gmkn_yl	0.0881	0.0000	
yas	gmka_yl	0.0318	0.0000	
yas	mulkn_ay	0.0368	0.0000	
yas	mulka_ay	0.0088	0.0000	
yas	banka_yl	0.034	0.0000	
yas	doviz_yl	0.0112	0.0000	
yas	nema_yl	0.0148	0.0000	
yas	temet_yl	0.0057	0.0005	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
yas	kar_yl	0.0019	0.2429	
yas	toprn_yl	0.0247	0.0000	
yas	topra_yl	0.011	0.0000	
yas	trsn_ay	0.3421	0.0000	
yas	trsa_ay	0.0262	0.0000	
yas	emekl_yl	0.4153	0.0000	
yas	viade_yl	0.1544	0.0000	
yas	yasli_yl	0.1227	0.0000	
yas	sosy_yl	0.027	0.0000	
yas	hasta_yl	0.0237	0.0000	
yas	dul_yl	0.202	0.0000	
yas	gazi_yl	0.0394	0.0000	
yas	burs_yl	0.0619	0.0000	
yas	issiz_yl	0.0051	0.0018	
yas	ddestek_yl	0.0549	0.0000	
yas	dayni_yl	0.0511	0.0000	
yas	ydemk_yl	0.0565	0.0000	
yas	yddav_yl	0.0146	0.0000	
yas	ydayn_yl	0.0038	0.0203	
yas	dgnfk_yl	0.0491	0.0000	
yas	dgbrs_yl	0.0322	0.0000	
yas	dsemk_yl	0.0204	0.0000	
yas	dgayn_yl	0.0489	0.0000	
yas	topgn_yl	0.1047	0.0000	
yas	topga_ay	0.0444	0.0000	
yas	topgelay	0.1061	0.0000	
yas	topgna_yl	0.2663	0.0000	
yas	topga_yl	0.0863	0.0000	
yas	topgelyl	0.2653	0.0000	
yas	indeks_arl	-0.0405	0.0000	
yas	kira_ay	0.0148	0.0000	
yas	loj_fert	-0.005	0.0021	
yas	izafi_k	0.0666	0.0000	
yas	konutsah	-0.0899	0.0000	
yas	kon_sure	0.0457	0.0000	
yas	kon_borc	0.0536	0.0000	
yas	kondeger	0.0526	0.0000	
yas	otur_sur	0.2179	0.0000	
yas	bina_tar	-0.1239	0.0000	
yas	oda_say	-0.0282	0.0000	
yas	kon_alan	-0.0067	0.0000	
yas	konutiki	-0.0881	0.0000	
yas	konut2_n	-0.0481	0.0000	
yas	sudepo	0.0117	0.0000	
yas	telefon	0.0984	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
yas	ceptel	-0.0291	0.0000	
yas	bilgisay	-0.0083	0.0000	
yas	intern	0.0098	0.0000	
yas	televiz	0.0183	0.0000	
yas	video	-0.0022	0.1735	
yas	lcdtele	0.0223	0.0000	
yas	dvd_vcd	-0.0615	0.0000	
yas	kamera	-0.0106	0.0000	
yas	uyduant	-0.0188	0.0000	
yas	buzdolab	0.0275	0.0000	
yas	derindon	0.0554	0.0000	
yas	bulasik	0.0514	0.0000	
yas	m_firin	0.0119	0.0000	
yas	camasir	0.039	0.0000	
yas	kurutma	0.0029	0.0728	
yas	haliyika	-0.0223	0.0000	
yas	klima	0.0287	0.0000	
yas	otomobil	-0.0171	0.0000	
yas	otoadet	-0.0572	0.0000	
yas	isveroto	-0.0152	0.0000	
yas	isotoadt	-0.015	0.0000	
yas	motosik	-0.0111	0.0000	
yas	motoadet	-0.0025	0.1302	
yas	deniztas	0.0042	0.0107	
yas	dentadet	0.009	0.0000	
yas	muskonut	0.0742	0.0000	
yas	musmik	0.0685	0.0000	
yas	muskira	0.0367	0.0000	
yas	muskdeg	0.0152	0.0000	
yas	aptmik	0.0833	0.0000	
yas	aptkira	0.0614	0.0000	
yas	aptkdeg	0.0285	0.0000	
yas	yazmik	0.0131	0.0000	
yas	yazkira	0.0067	0.0000	
yas	yazkdeg	0.0031	0.0574	
yas	tarmik	0.0081	0.0000	
yas	tarkira	0.0263	0.0000	
yas	tarkdeg	0.0055	0.0007	
yas	bagmik	0.0043	0.0092	
yas	bagkira	0.0119	0.0000	
yas	bagkdeg	0.0037	0.0230	
yas	seramik	0.0007	0.6845	
yas	serakira	0.0028	0.0869	
yas	serakdeg	0.0019	0.2362	
yas	arsamik	0.0138	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
yas	arsakira	0.0009	0.5813	
yas	arsakdeg	0.0013	0.4167	
yas	arsadeg	-0.0015	0.3607	
yas	dukmik	0.0246	0.0000	
yas	dukkira	0.0379	0.0000	
yas	dukkdeg	0.0165	0.0000	
yas	dukdeg	0.007	0.0000	
yas	otelmik	0.0037	0.0240	
yas	otelkira	0.0045	0.0057	
yas	otelkdeg	0.0011	0.5008	
yas	oteldeg	0.0014	0.4056	
yas	digmik	0.0017	0.2982	
yas	digkira	-0.0001	0.9368	
yas	digkdeg	-0.0019	0.2586	
yas	digdeg	0.002	0.2267	
yas	yilkulgel	0.0164	0.0000	
yas	efb_oecd	-0.2252	0.0000	
yas	hhb	-0.2954	0.0000	
yas	faktor	0.0071	0.0000	
ailegrup	sagliks2	0.0046	0.0055	
ailegrup	egitim	0.0743	0.0000	
ailegrup	sure_yil	-0.0486	0.0000	
ailegrup	calsur	-0.0519	0.0000	
ailegrup	kac_kisi	0.3491	0.0000	
ailegrup	ucrn_ay	0.0048	0.0031	
ailegrup	ucra_ay	-0.0424	0.0000	
ailegrup	ucrn_yil	-0.0066	0.0001	
ailegrup	ikr_yil	-0.0284	0.0000	
ailegrup	prim_yil	0.0247	0.0000	
ailegrup	ucra_yil	0.0784	0.0000	
ailegrup	konayni	0.0223	0.0000	
ailegrup	ulsayni	0.0465	0.0000	
ailegrup	egsayni	0.0091	0.0000	
ailegrup	yemayni	0.0827	0.0000	
ailegrup	kresayni	0.0051	0.0019	
ailegrup	giyayni	0.0057	0.0005	
ailegrup	gidaayni	0.0273	0.0000	
ailegrup	digayni	0.0076	0.0000	
ailegrup	mutsn_ay	-0.0138	0.0000	
ailegrup	mutsa_ay	-0.022	0.0000	
ailegrup	mutsn_yl	-0.0014	0.3906	
ailegrup	mutsa_yl	0.0001	0.9615	
ailegrup	targ_ay	-0.016	0.0000	
ailegrup	targ_yl	0.0115	0.0000	
ailegrup	calsur_e	-0.0014	0.3824	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
ailegrup	ekgln_ay	0.0014	0.3809	
ailegrup	ekgla_ay	0.0019	0.2584	
ailegrup	ekgln_yl	-0.0026	0.1062	
ailegrup	ekgla_yl	0.001	0.5617	
ailegrup	calay_yil	-0.3033	0.0000	
ailegrup	calsur_d	0.0138	0.0000	
ailegrup	digln_yl	0.0045	0.0065	
ailegrup	digla_yl	0.0109	0.0000	
ailegrup	gmkn_ay	-0.0083	0.0000	
ailegrup	gmka_ay	-0.003	0.0720	
ailegrup	gmkn_yl	0.0261	0.0000	
ailegrup	gmka_yl	0.0261	0.0000	
ailegrup	mulkn_ay	-0.015	0.0000	
ailegrup	mulka_ay	-0.0046	0.0050	
ailegrup	banka_yl	-0.0003	0.8592	
ailegrup	doviz_yl	-0.0044	0.0072	
ailegrup	nema_yl	-0.013	0.0000	
ailegrup	temet_yl	-0.0044	0.0076	
ailegrup	kar_yl	-0.0018	0.2647	
ailegrup	toprn_yl	-0.0086	0.0000	
ailegrup	topra_yl	-0.0022	0.1819	
ailegrup	trsn_ay	-0.0616	0.0000	
ailegrup	trsa_ay	-0.0015	0.3688	
ailegrup	emekl_yl	0.0757	0.0000	
ailegrup	viade_yl	-0.2008	0.0000	
ailegrup	yasli_yl	-0.0019	0.2355	
ailegrup	sosy_yl	0.0214	0.0000	
ailegrup	hasta_yl	0.0295	0.0000	
ailegrup	dul_yl	0.0393	0.0000	
ailegrup	gazi_yl	0.0275	0.0000	
ailegrup	burs_yl	0.0454	0.0000	
ailegrup	issiz_yl	0.0124	0.0000	
ailegrup	ddestek_yl	0.0281	0.0000	
ailegrup	dayni_yl	0.0399	0.0000	
ailegrup	ydemk_yl	0.0096	0.0000	
ailegrup	yddav_yl	0.0116	0.0000	
ailegrup	ydayn_yl	0.0043	0.0087	
ailegrup	dgnfk_yl	0.0711	0.0000	
ailegrup	dgbrs_yl	-0.0561	0.0000	
ailegrup	dsemk_yl	0.0052	0.0016	
ailegrup	dgayn_yl	0.03	0.0000	
ailegrup	topgn_yl	-0.0426	0.0000	
ailegrup	topga_ay	-0.0326	0.0000	
ailegrup	topgelay	-0.0438	0.0000	
ailegrup	topgna_yl	0.0934	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
ailegrup	topga_yl	0.0826	0.0000	
ailegrup	topgelyl	0.0977	0.0000	
ailegrup	indeks_arl	-0.3242	0.0000	
ailegrup	kira_ay	0.0574	0.0000	
ailegrup	loj_fert	0.0158	0.0000	
ailegrup	izafi_k	0.1655	0.0000	
ailegrup	konutsah	-0.2984	0.0000	
ailegrup	kon_sure	0.0477	0.0000	
ailegrup	kon_borc	0.1772	0.0000	
ailegrup	kondeger	0.0228	0.0000	
ailegrup	otur_sur	-0.017	0.0000	
ailegrup	bina_tar	0.1237	0.0000	
ailegrup	oda_say	-0.2839	0.0000	
ailegrup	kon_alan	0.1443	0.0000	
ailegrup	konutiki	-0.0454	0.0000	
ailegrup	konut2_n	-0.0122	0.0000	
ailegrup	sudepo	0.1612	0.0000	
ailegrup	telefon	-0.4039	0.0000	
ailegrup	ceptel	0.5094	0.0000	
ailegrup	bilgisay	0.3471	0.0000	
ailegrup	intern	0.3552	0.0000	
ailegrup	televiz	-0.1232	0.0000	
ailegrup	video	-0.1671	0.0000	
ailegrup	lcdtele	0.4082	0.0000	
ailegrup	dvd_vcd	-0.1513	0.0000	
ailegrup	kamera	0.0818	0.0000	
ailegrup	uyduant	-0.0926	0.0000	
ailegrup	buzdolab	0.0257	0.0000	
ailegrup	derindon	0.1816	0.0000	
ailegrup	bulasik	0.2796	0.0000	
ailegrup	m_firin	0.1538	0.0000	
ailegrup	camasir	0.2018	0.0000	
ailegrup	kurutma	0.0439	0.0000	
ailegrup	haliyika	-0.0648	0.0000	
ailegrup	klima	0.1772	0.0000	
ailegrup	otomobil	-0.143	0.0000	
ailegrup	otoadet	-0.4431	0.0000	
ailegrup	isveroto	-0.0032	0.0522	
ailegrup	isotoadt	0.0567	0.0000	
ailegrup	motosik	-0.01	0.0000	
ailegrup	motoadet	0.0924	0.0000	
ailegrup	deniztas	0.004	0.0144	
ailegrup	dentadet	0.0118	0.0000	
ailegrup	muskonut	0.2093	0.0000	
ailegrup	musmik	0.0266	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
ailegrup	muskira	-0.0079	0.0000	
ailegrup	muskdeg	0.0177	0.0000	
ailegrup	aptmik	0.0307	0.0000	
ailegrup	aptkira	0.0311	0.0000	
ailegrup	aptkdeg	0.0313	0.0000	
ailegrup	yazmik	0.0009	0.5741	
ailegrup	yazkira	-0.0024	0.1457	
ailegrup	yazkdeg	-0.0018	0.2765	
ailegrup	tarmik	-0.0195	0.0000	
ailegrup	tarkira	0.0000	0.9834	
ailegrup	tarkdeg	-0.0063	0.0001	
ailegrup	bagmik	0.0051	0.0018	
ailegrup	bagkira	-0.0046	0.0052	
ailegrup	bagkdeg	-0.0064	0.0001	
ailegrup	seramik	-0.006	0.0003	
ailegrup	serakira	0.0023	0.1627	
ailegrup	serakdeg	0.0021	0.1996	
ailegrup	arsamik	0.0096	0.0000	
ailegrup	arsakira	-0.0006	0.7141	
ailegrup	arsakdeg	0.0034	0.0407	
ailegrup	arsadeg	-0.0127	0.0000	
ailegrup	dukmik	0.016	0.0000	
ailegrup	dukkira	0.0158	0.0000	
ailegrup	dukkdeg	0.0211	0.0000	
ailegrup	dukdeg	-0.0331	0.0000	
ailegrup	otelmik	0.0008	0.6325	
ailegrup	otelkira	-0.005	0.0021	
ailegrup	otelkdeg	-0.0032	0.0483	
ailegrup	oteldeg	-0.0052	0.0017	
ailegrup	digmik	-0.0308	0.0000	
ailegrup	digkira	-0.0163	0.0000	
ailegrup	digkdeg	-0.0082	0.0000	
ailegrup	digdeg	-0.013	0.0000	
ailegrup	yilkulgel	0.1399	0.0000	
ailegrup	efb_oecd	-0.1161	0.0000	
ailegrup	hhb	-0.102	0.0000	
ailegrup	faktor	0.1521	0.0000	
sagliks2	egitim	-0.0061	0.0002	
sagliks2	sure_yil	0.003	0.0716	
sagliks2	calsur	0.0028	0.0869	
sagliks2	kac_kisi	0.0005	0.7669	
sagliks2	ucrn_ay	-0.012	0.0000	
sagliks2	ucra_ay	-0.0018	0.2846	
sagliks2	ucrn_yil	-0.0085	0.0000	
sagliks2	ikr_yil	0.0007	0.6715	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sagliks2	prim_yil	0.0005	0.7627	
sagliks2	ucra_yil	-0.0002	0.9034	
sagliks2	konayni	0.0005	0.7717	
sagliks2	ulsayni	0.0015	0.3515	
sagliks2	egsayni	0.0003	0.8362	
sagliks2	yemayni	-0.0009	0.5659	
sagliks2	kresayni	0.0001	0.9637	
sagliks2	giyayni	0.0003	0.8458	
sagliks2	gidaayni	0.0005	0.7485	
sagliks2	digayni	-0.0034	0.0357	
sagliks2	mutsn_ay	-0.0036	0.0296	
sagliks2	mutsa_ay	-0.0078	0.0000	
sagliks2	mutsn_yl	-0.003	0.0698	
sagliks2	mutsa_yl	-0.0019	0.2556	
sagliks2	targ_ay	0.0004	0.8274	
sagliks2	targ_yl	0.0005	0.7769	
sagliks2	calsur_e	0.0002	0.8940	
sagliks2	ekgln_ay	0.0000	0.9841	
sagliks2	ekgla_ay	0.0001	0.9634	
sagliks2	ekgln_yl	0.0002	0.8867	
sagliks2	ekgla_yl	0.0002	0.9042	
sagliks2	calay_yil	-0.0013	0.4379	
sagliks2	calsur_d	-0.0016	0.3302	
sagliks2	digln_yl	-0.0007	0.6478	
sagliks2	digla_yl	-0.0017	0.3082	
sagliks2	gmkn_ay	-0.0008	0.6237	
sagliks2	gmka_ay	0.0000	0.9813	
sagliks2	gmkn_yl	-0.0003	0.8671	
sagliks2	gmka_yl	0.0002	0.8917	
sagliks2	mulkn_ay	0.0001	0.9532	
sagliks2	mulka_ay	0.0001	0.9756	
sagliks2	banka_yl	-0.0007	0.6618	
sagliks2	doviz_yl	0.0001	0.9586	
sagliks2	nema_yl	-0.0006	0.7360	
sagliks2	temet_yl	0.0001	0.9682	
sagliks2	kar_yl	0	0.9854	
sagliks2	toprn_yl	0.0002	0.9095	
sagliks2	topra_yl	0.0001	0.9655	
sagliks2	trsn_ay	-0.0097	0.0000	
sagliks2	trsa_ay	0.0004	0.8309	
sagliks2	emekl_yl	-0.0033	0.0448	
sagliks2	viade_yl	-0.0099	0.0000	
sagliks2	yasli_yl	0.0006	0.6938	
sagliks2	sosy_yl	0.0003	0.8373	
sagliks2	hasta_yl	0.0003	0.8788	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sagliks2	dul_yl	-0.0124	0.0000	
sagliks2	gazi_yl	0.0004	0.7973	
sagliks2	burs_yl	0.0004	0.7960	
sagliks2	issiz_yl	0.0002	0.9207	
sagliks2	ddestek_yl	0.0005	0.7790	
sagliks2	dayni_yl	0.0006	0.7033	
sagliks2	ydemk_yl	0.0003	0.8645	
sagliks2	yddav_yl	0.0002	0.9073	
sagliks2	ydayn_yl	0.0001	0.9689	
sagliks2	dgnfk_yl	0.0006	0.7003	
sagliks2	dgbrs_yl	-0.0001	0.9667	
sagliks2	dsemk_yl	0.0001	0.9295	
sagliks2	dgayn_yl	0.0006	0.7179	
sagliks2	topgn_yl	-0.0055	0.0008	
sagliks2	topga_ay	-0.0032	0.0528	
sagliks2	topgelay	-0.0056	0.0006	
sagliks2	topgna_yl	-0.0073	0.0000	
sagliks2	topga_yl	-0.0006	0.6948	
sagliks2	topgelyl	-0.0072	0.0000	
sagliks2	indeks_arl	-0.0002	0.9009	
sagliks2	kira_ay	-0.0001	0.9471	
sagliks2	loj_fert	0.0003	0.8676	
sagliks2	izafi_k	-0.0065	0.0001	
sagliks2	konutsah	-0.0025	0.1209	
sagliks2	kon_sure	-0.0005	0.7721	
sagliks2	kon_borc	0.0015	0.3590	
sagliks2	kondeger	-0.0059	0.0003	
sagliks2	otur_sur	-0.001	0.5307	
sagliks2	bina_tar	-0.0005	0.7616	
sagliks2	oda_say	-0.0024	0.1417	
sagliks2	kon_alan	0.0012	0.4551	
sagliks2	konutiki	0.0016	0.3214	
sagliks2	konut2_n	0.0036	0.0294	
sagliks2	sudepo	-0.0027	0.1044	
sagliks2	telefon	-0.005	0.0023	
sagliks2	ceptel	-0.0014	0.3862	
sagliks2	bilgisay	-0.0006	0.7149	
sagliks2	intern	-0.0003	0.8392	
sagliks2	televiz	-0.0089	0.0000	
sagliks2	video	0.0013	0.4197	
sagliks2	lcdtele	0.002	0.2205	
sagliks2	dvd_vcd	-0.0074	0.0000	
sagliks2	kamera	-0.0015	0.3576	
sagliks2	uyduant	-0.0023	0.1535	
sagliks2	buzdolab	-0.0001	0.9568	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sagliks2	derindon	0.0008	0.6426	
sagliks2	bulasik	-0.0025	0.1272	
sagliks2	m_firin	-0.0016	0.3272	
sagliks2	camasir	-0.0027	0.1000	
sagliks2	kurutma	0.0008	0.6456	
sagliks2	haliyika	-0.001	0.5469	
sagliks2	klima	-0.0021	0.1917	
sagliks2	otomobil	-0.0023	0.1647	
sagliks2	otoadet	-0.0036	0.0276	
sagliks2	isveroto	-0.0028	0.0826	
sagliks2	isotoadt	-0.0015	0.3674	
sagliks2	motosik	0.0014	0.3891	
sagliks2	motoadet	0.0019	0.2504	
sagliks2	deniztas	0.0002	0.9202	
sagliks2	dentadet	0.0002	0.9014	
sagliks2	muskonut	0.0055	0.0008	
sagliks2	musmik	-0.0005	0.7611	
sagliks2	muskira	0.0012	0.4670	
sagliks2	muskdeg	0.0002	0.8918	
sagliks2	aptmik	-0.0056	0.0007	
sagliks2	aptkira	-0.0064	0.0001	
sagliks2	aptkdeg	-0.0003	0.8677	
sagliks2	yazmik	0.0002	0.9121	
sagliks2	yazkira	0.0002	0.9136	
sagliks2	yazkdeg	0.0000	0.9808	
sagliks2	tarmik	0.0012	0.4555	
sagliks2	tarkira	0.0004	0.7907	
sagliks2	tarkdeg	0.0002	0.9016	
sagliks2	bagmik	0.0002	0.9092	
sagliks2	bagkira	0.0002	0.9087	
sagliks2	bagkdeg	0.0001	0.9522	
sagliks2	seramik	0.0003	0.8400	
sagliks2	serakira	0.0001	0.9556	
sagliks2	serakdeg	0.0000	0.9823	
sagliks2	arsamik	-0.0067	0.0000	
sagliks2	arsakira	0.0001	0.9738	
sagliks2	arsakdeg	0.0000	0.9842	
sagliks2	arsadeg	-0.0003	0.8655	
sagliks2	dukmik	-0.0023	0.1682	
sagliks2	dukkira	-0.0045	0.0056	
sagliks2	dukkdeg	-0.002	0.2337	
sagliks2	dukdeg	-0.0001	0.9644	
sagliks2	otelmik	0.0001	0.9711	
sagliks2	otelkira	0.0001	0.9555	
sagliks2	otelkdeg	0.0000	0.9812	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sagliks2	oteldeg	0.0000	0.9802	
sagliks2	digmik	0.0002	0.8817	
sagliks2	digkira	0.0001	0.9374	
sagliks2	digkdeg	0.0001	0.9685	
sagliks2	digdeg	0.0001	0.9499	
sagliks2	yilkulgel	-0.0033	0.0434	
sagliks2	efb_oecd	0.0036	0.0278	
sagliks2	hhb	0.0042	0.0107	
sagliks2	faktor	-0.0034	0.0379	
egitim	sure_yil	-0.3306	0.0000	
egitim	calsur	-0.3288	0.0000	
egitim	kac_kisi	-0.0014	0.4075	
egitim	ucrn_ay	0.2427	0.0000	
egitim	ucra_ay	0.1766	0.0000	
egitim	ucrn_yil	0.3767	0.0000	
egitim	ikr_yil	0.1085	0.0000	
egitim	prim_yil	0.1051	0.0000	
egitim	ucra_yil	0.2537	0.0000	
egitim	konayni	0.0751	0.0000	
egitim	ulsayni	0.1871	0.0000	
egitim	egsayni	0.0368	0.0000	
egitim	yemayni	0.2382	0.0000	
egitim	kresayni	0.0112	0.0000	
egitim	giyayni	0.0469	0.0000	
egitim	gidaayni	0.0476	0.0000	
egitim	digayni	0.0432	0.0000	
egitim	mutsn_ay	0.058	0.0000	
egitim	mutsa_ay	0.0171	0.0000	
egitim	mutsn_yl	0.0745	0.0000	
egitim	mutsa_yl	0.024	0.0000	
egitim	targ_ay	0.0003	0.8657	
egitim	targ_yl	0.001	0.5474	
egitim	calsur_e	-0.0198	0.0000	
egitim	ekgln_ay	0.0054	0.0011	
egitim	ekgla_ay	0.0068	0.0000	
egitim	ekgln_yl	0.0337	0.0000	
egitim	ekgla_yl	0.0094	0.0000	
egitim	calay_yil	-0.0403	0.0000	
egitim	calsur_d	0.0024	0.1355	
egitim	digln_yl	0.0459	0.0000	
egitim	digla_yl	0.0317	0.0000	
egitim	gmkn_ay	0.0546	0.0000	
egitim	gmka_ay	-0.0015	0.3643	
egitim	gmkn_yl	0.0671	0.0000	
egitim	gmka_yl	0.0246	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
egitim	mulkn_ay	0.0446	0.0000	
egitim	mulka_ay	-0.0023	0.1566	
egitim	banka_yl	0.0448	0.0000	
egitim	doviz_yl	0.0157	0.0000	
egitim	nema_yl	0.0327	0.0000	
egitim	temet_yl	0.0127	0.0000	
egitim	kar_yl	0.004	0.0150	
egitim	toprn_yl	0.0038	0.0197	
egitim	topra_yl	0.0011	0.5209	
egitim	trsn_ay	0.081	0.0000	
egitim	trsa_ay	-0.0013	0.4132	
egitim	emekl_yl	0.132	0.0000	
egitim	viade_yl	0.183	0.0000	
egitim	yasli_yl	-0.058	0.0000	
egitim	sosy_yl	-0.0169	0.0000	
egitim	hasta_yl	-0.017	0.0000	
egitim	dul_yl	-0.0647	0.0000	
egitim	gazi_yl	-0.0178	0.0000	
egitim	burs_yl	-0.0135	0.0000	
egitim	issiz_yl	0.0176	0.0000	
egitim	ddestek_yl	-0.0028	0.0862	
egitim	dayni_yl	-0.0209	0.0000	
egitim	ydemk_yl	-0.0007	0.6522	
egitim	yddav_yl	0.0034	0.0401	
egitim	ydayn_yl	0.0032	0.0547	
egitim	dgnfk_yl	0.0425	0.0000	
egitim	dgbrs_yl	0.0141	0.0000	
egitim	dsemk_yl	0.0226	0.0000	
egitim	dgayn_yl	0.0166	0.0000	
egitim	topgn_yl	0.1274	0.0000	
egitim	topga_ay	0.1092	0.0000	
egitim	topgelay	0.1314	0.0000	
egitim	topgna_yl	0.3447	0.0000	
egitim	topga_yl	0.2128	0.0000	
egitim	topgelyl	0.3515	0.0000	
egitim	indeks_arl	-0.0331	0.0000	
egitim	kira_ay	0.1343	0.0000	
egitim	loj_fert	0.0094	0.0000	
egitim	izafi_k	0.2853	0.0000	
egitim	konutsah	-0.0126	0.0000	
egitim	kon_sure	-0.0267	0.0000	
egitim	kon_borc	0.0768	0.0000	
egitim	kondeger	0.2039	0.0000	
egitim	otur_sur	-0.136	0.0000	
egitim	bina_tar	0.1191	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
egitim	oda_say	0.0715	0.0000	
egitim	kon_alan	0.1211	0.0000	
egitim	konutiki	-0.0888	0.0000	
egitim	konut2_n	-0.0811	0.0000	
egitim	sudepo	0.0644	0.0000	
egitim	telefon	0.0986	0.0000	
egitim	ceptel	0.2117	0.0000	
egitim	bilgisay	0.3143	0.0000	
egitim	intern	0.2655	0.0000	
egitim	televiz	0.14	0.0000	
egitim	video	0.0657	0.0000	
egitim	lcdtele	0.163	0.0000	
egitim	dvd_vcd	0.1148	0.0000	
egitim	kamera	0.1436	0.0000	
egitim	uyduant	-0.0109	0.0000	
egitim	buzdolab	0.0602	0.0000	
egitim	derindon	0.0468	0.0000	
egitim	bulasik	0.3184	0.0000	
egitim	m_firin	0.1618	0.0000	
egitim	camasir	0.174	0.0000	
egitim	kurutma	0.0691	0.0000	
egitim	haliyika	0.1386	0.0000	
egitim	klima	0.1439	0.0000	
egitim	otomobil	0.1421	0.0000	
egitim	otoadet	0.0442	0.0000	
egitim	isveroto	0.0399	0.0000	
egitim	isotoadt	0.073	0.0000	
egitim	motosik	-0.0268	0.0000	
egitim	motoadet	-0.0265	0.0000	
egitim	deniztas	0.0082	0.0000	
egitim	dentadet	0.0133	0.0000	
egitim	muskonut	-0.1019	0.0000	
egitim	musmik	-0.0294	0.0000	
egitim	muskira	0.0014	0.3868	
egitim	muskdeg	0.0151	0.0000	
egitim	aptmik	0.0695	0.0000	
egitim	aptkira	0.0765	0.0000	
egitim	aptkdeg	0.0528	0.0000	
egitim	yazmik	0.0215	0.0000	
egitim	yazkira	0.0172	0.0000	
egitim	yazkdeg	0.0024	0.1507	
egitim	tarmik	-0.0353	0.0000	
egitim	tarkira	-0.0024	0.1397	
egitim	tarkdeg	0.002	0.2194	
egitim	bagmik	0.003	0.0703	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
egitim	bagkira	0.0015	0.3661	
egitim	bagkdeg	0.002	0.2157	
egitim	seramik	-0.0041	0.0129	
egitim	serakira	0.0036	0.0269	
egitim	serakdeg	0.0053	0.0013	
egitim	arsamik	0.0173	0.0000	
egitim	arsakira	0.0018	0.2635	
egitim	arsakdeg	0.0038	0.0208	
egitim	arsadeg	0.0043	0.0083	
egitim	dukmik	0.029	0.0000	
egitim	dukkira	0.0375	0.0000	
egitim	dukkdeg	0.0307	0.0000	
egitim	dukdeg	0.0173	0.0000	
egitim	otelmik	0.0019	0.2467	
egitim	otelkira	0.0034	0.0385	
egitim	otelkdeg	0.0000	0.9952	
egitim	oteldeg	0.0054	0.0010	
egitim	digmik	0.0021	0.1991	
egitim	digkira	-0.0021	0.2048	
egitim	digkdeg	0.0037	0.0228	
egitim	digdeg	0.0025	0.1277	
egitim	yilkulgel	0.2569	0.0000	
egitim	efb_oecd	-0.1612	0.0000	
egitim	hhb	-0.206	0.0000	
egitim	faktor	0.0402	0.0000	
sure_yil	calsur	0.9993	0.0000	YES
sure_yil	kac_kisi	0.4582	0.0000	
sure_yil	ucrn_ay	-0.1269	0.0000	
sure_yil	ucra_ay	-0.2795	0.0000	
sure_yil	ucrn_yil	-0.2895	0.0000	
sure_yil	ikr_yil	-0.1152	0.0000	
sure_yil	prim_yil	-0.0815	0.0000	
sure_yil	ucra_yil	-0.3547	0.0000	
sure_yil	konayni	-0.0759	0.0000	
sure_yil	ulsayni	-0.2475	0.0000	
sure_yil	egsayni	-0.0558	0.0000	
sure_yil	yemayni	-0.3603	0.0000	
sure_yil	kresayni	-0.0123	0.0000	
sure_yil	giyayni	-0.0518	0.0000	
sure_yil	gidaayni	-0.0848	0.0000	
sure_yil	digayni	-0.0557	0.0000	
sure_yil	mutsn_ay	-0.0618	0.0000	
sure_yil	mutsa_ay	-0.0799	0.0000	
sure_yil	mutsn_yl	-0.0763	0.0000	
sure_yil	mutsa_yl	-0.0978	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sure_yil	targ_ay	-0.058	0.0000	
sure_yil	targ_yl	-0.0755	0.0000	
sure_yil	calsur_e	-0.0361	0.0000	
sure_yil	ekgln_ay	-0.0054	0.0010	
sure_yil	ekgla_ay	-0.0125	0.0000	
sure_yil	ekgln_yl	-0.0387	0.0000	
sure_yil	ekgla_yl	-0.0327	0.0000	
sure_yil	calay_yil	-0.115	0.0000	
sure_yil	calsur_d	-0.0242	0.0000	
sure_yil	digln_yl	-0.0137	0.0000	
sure_yil	digla_yl	-0.0151	0.0000	
sure_yil	gmkn_ay	-0.023	0.0000	
sure_yil	gmka_ay	-0.0006	0.7094	
sure_yil	gmkn_yl	-0.0279	0.0000	
sure_yil	gmka_yl	-0.0138	0.0000	
sure_yil	mulkn_ay	-0.0149	0.0000	
sure_yil	mulka_ay	0.0002	0.8910	
sure_yil	banka_yl	-0.0102	0.0000	
sure_yil	doviz_yl	-0.0031	0.0582	
sure_yil	nema_yl	-0.0176	0.0000	
sure_yil	temet_yl	-0.0073	0.0000	
sure_yil	kar_yl	-0.0042	0.0114	
sure_yil	toprn_yl	-0.0043	0.0088	
sure_yil	topra_yl	-0.0017	0.2963	
sure_yil	trsn_ay	0.0406	0.0000	
sure_yil	trsa_ay	-0.011	0.0000	
sure_yil	emekl_yl	0.0167	0.0000	
sure_yil	viade_yl	-0.1376	0.0000	
sure_yil	yasli_yl	-0.0085	0.0000	
sure_yil	sosy_yl	-0.0076	0.0000	
sure_yil	hasta_yl	0.0076	0.0000	
sure_yil	dul_yl	0.0749	0.0000	
sure_yil	gazi_yl	0.0125	0.0000	
sure_yil	burs_yl	0.0253	0.0000	
sure_yil	issiz_yl	-0.0026	0.1070	
sure_yil	ddestek_yl	-0.0691	0.0000	
sure_yil	dayni_yl	-0.0383	0.0000	
sure_yil	ydemk_yl	0.0139	0.0000	
sure_yil	yddav_yl	0.0027	0.1046	
sure_yil	ydayn_yl	-0.0003	0.8740	
sure_yil	dgnfk_yl	-0.0055	0.0008	
sure_yil	dgbrs_yl	0.0091	0.0000	
sure_yil	dsemk_yl	0.0064	0.0001	
sure_yil	dgayn_yl	-0.0139	0.0000	
sure_yil	topgn_yl	-0.1621	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sure_yil	topga_ay	-0.1985	0.0000	
sure_yil	topgelay	-0.1697	0.0000	
sure_yil	topgna_yl	-0.3898	0.0000	
sure_yil	topga_yl	-0.3222	0.0000	
sure_yil	topgelyl	-0.4027	0.0000	
sure_yil	indeks_arl	0.0279	0.0000	
sure_yil	kira_ay	-0.0345	0.0000	
sure_yil	loj_fert	-0.0122	0.0000	
sure_yil	izafi_k	0.0168	0.0000	
sure_yil	konutsah	0.0122	0.0000	
sure_yil	kon_sure	-0.0054	0.0010	
sure_yil	kon_borc	-0.0104	0.0000	
sure_yil	kondeger	0.0137	0.0000	
sure_yil	otur_sur	-0.0304	0.0000	
sure_yil	bina_tar	0.0082	0.0000	
sure_yil	oda_say	0.0054	0.0011	
sure_yil	kon_alan	-0.0138	0.0000	
sure_yil	konutiki	0.0142	0.0000	
sure_yil	konut2_n	-0.0021	0.1919	
sure_yil	sudepo	-0.0043	0.0084	
sure_yil	telefon	0.0066	0.0001	
sure_yil	ceptel	-0.0686	0.0000	
sure_yil	bilgisay	-0.0478	0.0000	
sure_yil	intern	-0.0497	0.0000	
sure_yil	televiz	0.0068	0.0000	
sure_yil	video	0.0075	0.0000	
sure_yil	lcdtele	-0.0334	0.0000	
sure_yil	dvd_vcd	-0.0002	0.8958	
sure_yil	kamera	-0.0173	0.0000	
sure_yil	uyduant	-0.0114	0.0000	
sure_yil	buzdolab	-0.0122	0.0000	
sure_yil	derindon	-0.0197	0.0000	
sure_yil	bulasik	-0.0181	0.0000	
sure_yil	m_firin	-0.0143	0.0000	
sure_yil	camasir	0.0018	0.2656	
sure_yil	kurutma	-0.0061	0.0002	
sure_yil	haliyika	0.0128	0.0000	
sure_yil	klima	-0.0036	0.0272	
sure_yil	otomobil	-0.0154	0.0000	
sure_yil	otoadet	0.0008	0.6091	
sure_yil	isveroto	-0.005	0.0023	
sure_yil	isotoadt	-0.0172	0.0000	
sure_yil	motosik	-0.0304	0.0000	
sure_yil	motoadet	-0.0483	0.0000	
sure_yil	deniztas	-0.0029	0.0739	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
sure_yil	dentadet	-0.0031	0.0553	
sure_yil	muskonut	-0.0471	0.0000	
sure_yil	musmik	-0.011	0.0000	
sure_yil	muskira	0.0159	0.0000	
sure_yil	muskdeg	0.0004	0.8136	
sure_yil	aptmik	0.0162	0.0000	
sure_yil	aptkira	0.0141	0.0000	
sure_yil	aptkdeg	0.0051	0.0019	
sure_yil	yazmik	0.0017	0.2974	
sure_yil	yazkira	0.0011	0.4865	
sure_yil	yazkdeg	0.0045	0.0061	
sure_yil	tarmik	-0.0257	0.0000	
sure_yil	tarkira	0.0106	0.0000	
sure_yil	tarkdeg	0.0044	0.0070	
sure_yil	bagmik	-0.0053	0.0011	
sure_yil	bagkira	0	0.9887	
sure_yil	bagkdeg	-0.0014	0.3990	
sure_yil	seramik	-0.0204	0.0000	
sure_yil	serakira	-0.0002	0.9216	
sure_yil	serakdeg	0.0026	0.1116	
sure_yil	arsamik	-0.0043	0.0089	
sure_yil	arsakira	-0.0029	0.0771	
sure_yil	arsakdeg	-0.0004	0.8236	
sure_yil	arsadeg	0.0016	0.3314	
sure_yil	dukmik	0.0028	0.0856	
sure_yil	dukkira	0.011	0.0000	
sure_yil	dukkdeg	0.0027	0.1044	
sure_yil	dukdeg	0.003	0.0658	
sure_yil	otelmik	0.001	0.5393	
sure_yil	otelkira	0.0005	0.7436	
sure_yil	otelkdeg	0.0035	0.0336	
sure_yil	oteldeg	0.0002	0.8989	
sure_yil	digmik	-0.0038	0.0192	
sure_yil	digkira	-0.0011	0.4998	
sure_yil	digkdeg	0.0024	0.1423	
sure_yil	digdeg	0.0015	0.3718	
sure_yil	yilkulgel	-0.0796	0.0000	
sure_yil	efb_oecd	0.0389	0.0000	
sure_yil	hhb	0.0675	0.0000	
sure_yil	faktor	-0.0407	0.0000	
calsur	kac_kisi	0.4559	0.0000	
calsur	ucrn_ay	-0.1273	0.0000	
calsur	ucra_ay	-0.2732	0.0000	
calsur	ucrn_yil	-0.2898	0.0000	
calsur	ikr_yil	-0.1148	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
calsur	prim_yil	-0.0808	0.0000	
calsur	ucra_yil	-0.3514	0.0000	
calsur	konayni	-0.0765	0.0000	
calsur	ulsayni	-0.2454	0.0000	
calsur	egsayni	-0.0552	0.0000	
calsur	yemayni	-0.356	0.0000	
calsur	kresayni	-0.0123	0.0000	
calsur	giyayni	-0.0515	0.0000	
calsur	gidaayni	-0.0838	0.0000	
calsur	digayni	-0.055	0.0000	
calsur	mutsn_ay	-0.0599	0.0000	
calsur	mutsa_ay	-0.0767	0.0000	
calsur	mutsn_yl	-0.0749	0.0000	
calsur	mutsa_yl	-0.0949	0.0000	
calsur	targ_ay	-0.0584	0.0000	
calsur	targ_yl	-0.0766	0.0000	
calsur	calsur_e	-0.0356	0.0000	
calsur	ekgln_ay	-0.0054	0.0009	
calsur	ekgla_ay	-0.0122	0.0000	
calsur	ekgln_yl	-0.0382	0.0000	
calsur	ekgla_yl	-0.0323	0.0000	
calsur	calay_yil	-0.1158	0.0000	
calsur	calsur_d	-0.0231	0.0000	
calsur	digln_yl	-0.0128	0.0000	
calsur	digla_yl	-0.0139	0.0000	
calsur	gmkn_ay	-0.0227	0.0000	
calsur	gmka_ay	-0.0006	0.7354	
calsur	gmkn_yl	-0.0284	0.0000	
calsur	gmka_yl	-0.0151	0.0000	
calsur	mulkn_ay	-0.0148	0.0000	
calsur	mulka_ay	0.0002	0.8936	
calsur	banka_yl	-0.0103	0.0000	
calsur	doviz_yl	-0.0031	0.0587	
calsur	nema_yl	-0.0176	0.0000	
calsur	temet_yl	-0.0071	0.0000	
calsur	kar_yl	-0.004	0.0138	
calsur	toprn_yl	-0.0043	0.0091	
calsur	topra_yl	-0.0018	0.2691	
calsur	trsn_ay	0.0402	0.0000	
calsur	trsa_ay	-0.0107	0.0000	
calsur	emekl_yl	0.0149	0.0000	
calsur	viade_yl	-0.137	0.0000	
calsur	yasli_yl	-0.01	0.0000	
calsur	sosy_yl	-0.008	0.0000	
calsur	hasta_yl	0.0071	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
calsur	dul_yl	0.0746	0.0000	
calsur	gazi_yl	0.0124	0.0000	
calsur	burs_yl	0.0248	0.0000	
calsur	issiz_yl	-0.0026	0.1099	
calsur	ddestek_yl	-0.0704	0.0000	
calsur	dayni_yl	-0.0385	0.0000	
calsur	ydemk_yl	0.0136	0.0000	
calsur	yddav_yl	0.0025	0.1221	
calsur	ydayn_yl	-0.0004	0.8217	
calsur	dgnfk_yl	-0.0058	0.0004	
calsur	dgbrs_yl	0.0092	0.0000	
calsur	dsemk_yl	0.0064	0.0001	
calsur	dgayn_yl	-0.0137	0.0000	
calsur	topgn_yl	-0.1601	0.0000	
calsur	topga_ay	-0.1935	0.0000	
calsur	topgelay	-0.1675	0.0000	
calsur	topgna_yl	-0.3897	0.0000	
calsur	topga_yl	-0.3184	0.0000	
calsur	topgelyl	-0.4024	0.0000	
calsur	indeks_arl	0.0291	0.0000	
calsur	kira_ay	-0.0348	0.0000	
calsur	loj_fert	-0.0122	0.0000	
calsur	izafi_k	0.0177	0.0000	
calsur	konutsah	0.0198	0.0000	
calsur	kon_sure	-0.0059	0.0003	
calsur	kon_borc	-0.0147	0.0000	
calsur	kondeger	0.0176	0.0000	
calsur	otur_sur	-0.0331	0.0000	
calsur	bina_tar	0.0091	0.0000	
calsur	oda_say	0.006	0.0002	
calsur	kon_alan	-0.015	0.0000	
calsur	konutiki	0.0151	0.0000	
calsur	konut2_n	-0.0026	0.1174	
calsur	sudepo	-0.0035	0.0311	
calsur	telefon	0.008	0.0000	
calsur	ceptel	-0.069	0.0000	
calsur	bilgisay	-0.0488	0.0000	
calsur	intern	-0.0505	0.0000	
calsur	televiz	0.0086	0.0000	
calsur	video	0.0083	0.0000	
calsur	lcdtele	-0.0371	0.0000	
calsur	dvd_vcd	0.0033	0.0413	
calsur	kamera	-0.0175	0.0000	
calsur	uyduant	-0.0124	0.0000	
calsur	buzdolab	-0.0123	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
calsur	derindon	-0.022	0.0000	
calsur	bulasik	-0.0189	0.0000	
calsur	m_firin	-0.0152	0.0000	
calsur	camasir	0.0026	0.1116	
calsur	kurutma	-0.0068	0.0000	
calsur	haliyika	0.0136	0.0000	
calsur	klima	-0.0041	0.0120	
calsur	otomobil	-0.0139	0.0000	
calsur	otoadet	0.0039	0.0173	
calsur	isveroto	-0.004	0.0156	
calsur	isotoadt	-0.0168	0.0000	
calsur	motosik	-0.0295	0.0000	
calsur	motoadet	-0.0488	0.0000	
calsur	deniztas	-0.0029	0.0748	
calsur	dentadet	-0.0032	0.0525	
calsur	muskonut	-0.0556	0.0000	
calsur	musmik	-0.0116	0.0000	
calsur	muskira	0.0159	0.0000	
calsur	muskdeg	0.0004	0.7993	
calsur	aptmik	0.0161	0.0000	
calsur	aptkira	0.0137	0.0000	
calsur	aptkdeg	0.0049	0.0031	
calsur	yazmik	0.0017	0.3056	
calsur	yazkira	0.0012	0.4801	
calsur	yazkdeg	0.0045	0.0057	
calsur	tarmik	-0.0269	0.0000	
calsur	tarkira	0.0104	0.0000	
calsur	tarkdeg	0.0044	0.0075	
calsur	bagmik	-0.0056	0.0007	
calsur	bagkira	-0.0002	0.9048	
calsur	bagkdeg	-0.0014	0.3849	
calsur	seramik	-0.0205	0.0000	
calsur	serakira	-0.0002	0.9219	
calsur	serakdeg	0.0027	0.1014	
calsur	arsamik	-0.0044	0.0074	
calsur	arsakira	-0.0028	0.0826	
calsur	arsakdeg	-0.0004	0.8229	
calsur	arsadeg	0.0017	0.3128	
calsur	dukmik	0.0029	0.0777	
calsur	dukkira	0.011	0.0000	
calsur	dukkdeg	0.0026	0.1101	
calsur	dukdeg	0.0032	0.0488	
calsur	otelmik	0.0011	0.5121	
calsur	otelkira	0.0005	0.7724	
calsur	otelkdeg	0.0035	0.0334	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
calsur	oteldeg	0.0003	0.8639	
calsur	digmik	-0.0037	0.0250	
calsur	digkira	-0.0011	0.5197	
calsur	digkdeg	0.0024	0.1476	
calsur	digdeg	0.0015	0.3497	
calsur	yilkulgel	-0.0797	0.0000	
calsur	efb_oecd	0.0398	0.0000	
calsur	hhb	0.0687	0.0000	
calsur	faktor	-0.0409	0.0000	
kac_kisi	ucrn_ay	0.0787	0.0000	
kac_kisi	ucra_ay	-0.0162	0.0000	
kac_kisi	ucrn_yil	0.0248	0.0000	
kac_kisi	ikr_yil	0.0194	0.0000	
kac_kisi	prim_yil	0.0114	0.0000	
kac_kisi	ucra_yil	0.0275	0.0000	
kac_kisi	konayni	-0.0025	0.1220	
kac_kisi	ulsayni	0.0393	0.0000	
kac_kisi	egsayni	-0.0101	0.0000	
kac_kisi	yemayni	0.0255	0.0000	
kac_kisi	kresayni	0.0026	0.1180	
kac_kisi	giyayni	0.0036	0.0265	
kac_kisi	gidaayni	-0.0026	0.1102	
kac_kisi	digayni	-0.0018	0.2709	
kac_kisi	mutsn_ay	-0.0516	0.0000	
kac_kisi	mutsa_ay	-0.1047	0.0000	
kac_kisi	mutsn_yl	-0.0475	0.0000	
kac_kisi	mutsa_yl	-0.086	0.0000	
kac_kisi	targ_ay	-0.0772	0.0000	
kac_kisi	targ_yl	-0.0493	0.0000	
kac_kisi	calsur_e	-0.0446	0.0000	
kac_kisi	ekgln_ay	-0.0008	0.6104	
kac_kisi	ekgla_ay	-0.0044	0.0072	
kac_kisi	ekgln_yl	-0.0173	0.0000	
kac_kisi	ekgla_yl	-0.0167	0.0000	
kac_kisi	calay_yil	-0.0721	0.0000	
kac_kisi	calsur_d	-0.0059	0.0003	
kac_kisi	digln_yl	0.0029	0.0731	
kac_kisi	digla_yl	0.0027	0.0947	
kac_kisi	gmkn_ay	-0.013	0.0000	
kac_kisi	gmka_ay	0.0015	0.3735	
kac_kisi	gmkn_yl	0.0027	0.0974	
kac_kisi	gmka_yl	0.009	0.0000	
kac_kisi	mulkn_ay	-0.0032	0.0523	
kac_kisi	mulka_ay	-0.0013	0.4374	
kac_kisi	banka_yl	0.0006	0.7364	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
kac_kisi	doviz_yl	-0.0014	0.4037	
kac_kisi	nema_yl	0.0019	0.2417	
kac_kisi	temet_yl	-0.0026	0.1109	
kac_kisi	kar_yl	0.0004	0.8275	
kac_kisi	toprn_yl	-0.0069	0.0000	
kac_kisi	topra_yl	-0.0018	0.2832	
kac_kisi	trsn_ay	-0.0146	0.0000	
kac_kisi	trsa_ay	-0.0102	0.0000	
kac_kisi	emekl_yl	0.0234	0.0000	
kac_kisi	viade_yl	-0.0241	0.0000	
kac_kisi	yasli_yl	-0.0563	0.0000	
kac_kisi	sosy_yl	-0.0054	0.0009	
kac_kisi	hasta_yl	0.0103	0.0000	
kac_kisi	dul_yl	0.0383	0.0000	
kac_kisi	gazi_yl	0.0095	0.0000	
kac_kisi	burs_yl	0.017	0.0000	
kac_kisi	issiz_yl	0.0055	0.0008	
kac_kisi	ddestek_yl	-0.0185	0.0000	
kac_kisi	dayni_yl	0.0036	0.0270	
kac_kisi	ydemk_yl	0.0086	0.0000	
kac_kisi	yddav_yl	0.0048	0.0032	
kac_kisi	ydayn_yl	0.0018	0.2728	
kac_kisi	dgnfk_yl	0.0254	0.0000	
kac_kisi	dgbrs_yl	-0.0137	0.0000	
kac_kisi	dsemk_yl	0.003	0.0695	
kac_kisi	dgayn_yl	0.0002	0.8853	
kac_kisi	topgn_yl	-0.0901	0.0000	
kac_kisi	topga_ay	-0.05	0.0000	
kac_kisi	topgelay	-0.0918	0.0000	
kac_kisi	topgna_yl	-0.0844	0.0000	
kac_kisi	topga_yl	-0.011	0.0000	
kac_kisi	topgelyl	-0.0831	0.0000	
kac_kisi	indeks_arl	-0.1016	0.0000	
kac_kisi	kira_ay	0.0275	0.0000	
kac_kisi	loj_fert	0.0044	0.0072	
kac_kisi	izafi_k	0.1319	0.0000	
kac_kisi	konutsah	-0.1026	0.0000	
kac_kisi	kon_sure	0.0167	0.0000	
kac_kisi	kon_borc	0.0619	0.0000	
kac_kisi	kondeger	0.0539	0.0000	
kac_kisi	otur_sur	-0.1012	0.0000	
kac_kisi	bina_tar	0.0924	0.0000	
kac_kisi	oda_say	-0.0991	0.0000	
kac_kisi	kon_alan	0.0504	0.0000	
kac_kisi	konutiki	-0.0073	0.0000	

Table C.14 (Continued) : Pearson Correlation for Quantitative Variables.

Variable 1	Variable 2	Pearson Corr.Coeff.	Significance	Drop
kac_kisi	konut2_n	-0.015	0.0000	
kac_kisi	sudepo	0.0563	0.0000	
kac_kisi	telefon	-0.1347	0.0000	
kac_kisi	ceptel	0.2031	0.0000	
kac_kisi	bilgisay	0.1522	0.0000	
kac_kisi	intern	0.1398	0.0000	
kac_kisi	televiz	0.0026	0.1064	
kac_kisi	video	-0.0344	0.0000	
kac_kisi	lcdtele	0.1425	0.0000	
kac_kisi	dvd_vcd	-0.0098	0.0000	

Table C.15 : Percentage of Missing Values.

Variable	Missing Fraction
ayni_is	0.981747149
sekkod_e	0.981208946
ekis_ay	0.981130908
isdur_e	0.981130908
meslek_e	0.981130908
sekkod_d	0.975584815
isdur_d	0.975331864
meslek_d	0.975331864
yakit_3	0.967571102
mulk_dur	0.963940982
cl_neden	0.951075447
kira_sek	0.810555605
ist_tesk	0.799226615
ekis_yl	0.763374801
sgk	0.763374801
isfaal	0.74938444
sekkod	0.650580039
isdur	0.648220057
isneden	0.621767807
yapituru	0.581866081
zor_egiti	0.546062705
yakit_su	0.533808021
yakit_mut	0.532731632
yakit_2	0.51779674
diger	0.517293528
zemin_ban	0.492587719
zemin_sal	0.488389803
zemin_mut	0.483177392
jenrator	0.482706472
elektri	0.301423793
kirkntkd	0.246452627
ogrenci	0.239687524
medenidr	0.229461833
havuz	0.064572553
sauna	0.064572553
jakuzi	0.064572553
asansor	0.064572553
isit_sis	0.009808591
Rest of Them	0
.	.
.	.
.	.

Figure A.1 : Collinearity of Variables [Full Heatmap]

Table C.16 : Highly Correlated Variables Above rho>.80.

Correlated Feature	Pearson's Correlation	Dropped Feature
ucrn_ay	0.9423	ucrn_yil
sure_yil	-0.8436	calay_yil
gmkn_ay	0.9895	gmkn_yl
trsn_ay	0.8145	emekl_yl
ucrn_ay	0.8785	topgelay
ucrn_yil	0.8309	topgelay
ucrn_ay	0.8563	topgelyl
konutsah	-1.0000	kon_sure
kon_sure	0.9336	otur_sur
calsur_e	1.0000	yazkdeg
ekgln_yl	1.0000	yazkdeg
kon_sure	1.0000	yazkdeg
muskdeg	0.9931	yazkdeg
ekgln_yl	0.9021	bagkdeg
muskdeg	0.9036	bagkdeg
yazkdeg	1.0000	bagkdeg
bagkdeg	-1.0000	seramik
tarkdeg	0.8451	serakira
bagkdeg	1.0000	serakira
izafi_k	0.8739	serakdeg
kon_sure	-0.9940	serakdeg
aptkdeg	0.9875	serakdeg
tarkdeg	0.8483	serakdeg
bagmik	1.0000	serakdeg
bagkira	-1.0000	serakdeg
serakdeg	0.8751	arsamik
serakdeg	-0.9998	arsakira
calsur_e	0.9631	arsakdeg
izafi_k	0.8520	arsakdeg
tarkdeg	0.9872	arsadeg
arsakdeg	0.9115	dukkira
serakdeg	-1.0000	dukdeg
dukdeg	0.9811	otelmik
tarkdeg	1.0000	otelkira
ucrn_ay	1.0000	otelkdeg
ucrn_yil	0.8783	otelkdeg
ikr_yil	-1.0000	otelkdeg
ulsayni	0.9103	otelkdeg
yemayni	0.9103	otelkdeg
gidaayni	0.9103	otelkdeg
digayni	0.9103	otelkdeg
kira_ay	-1.0000	otelkdeg
kondeger	0.8540	otelkdeg
musmik	0.9958	otelkdeg
aptkdeg	1.0000	otelkdeg

Table C.16 (Continued) : Highly Correlated Variables Above rho>.80.

Correlated Feature	Pearson's Correlation	Dropped Feature
bagmik	-1.0000	otelkdeg
arsadeg	-1.0000	otelkdeg
dukkira	0.9450	otelkdeg
dukkdeg	0.9563	otelkdeg
dukdeg	0.9792	otelkdeg
ucrn_ay	0.9750	oteldeg
ucra_ay	0.9932	oteldeg
ucrn_yil	0.9690	oteldeg
yemayni	0.9960	oteldeg
ekgln_yl	0.8414	oteldeg
ekgla_yl	0.9542	oteldeg
digln_yl	1.0000	oteldeg
kira_ay	0.9490	oteldeg
tarkdeg	1.0000	oteldeg
otelkdeg	0.9836	oteldeg
loj_fert	1.0000	digmik
tarkdeg	1.0000	digkdeg
arsadeg	0.9978	digkdeg
digla_yl	0.9505	digdeg
kira_ay	0.8393	digdeg
loj_fert	0.9056	digdeg
aptkdeg	0.8004	digdeg

Table C.17 : Feature Importance based on Training Gradient Boosting Models.

Feature	Importance Factor	Normalized Importance Factor	Cumulative Importance
yilkulgel	11.7000	0.3900	0.3900
izafi_k	9.5000	0.3167	0.7067
ceptel	4.5000	0.1500	0.8567
intern	2.0000	0.0667	0.9233
bilgisay	0.7000	0.0233	0.9467
kira_ay	0.6000	0.0200	0.9667
hhb	0.6000	0.0200	0.9867
Rest of Them	0.0000	0.0000	1.0000
.	.	.	.
.	.	.	.
.	.	.	.

3.1 R Codes for Converter of HBSs to CSV

```
## -*- coding: utf-8 -*-
#"""
#Created on Wed Dec 20 23:11:04 2017

#######################################################################
#                          _          _              _            _    #
#    _   _   __ _  ____ (_) _ __   | | __ _   _ | |_  _   _ | | __     #
#   | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#   | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#    \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#    |___/                                                            #
#    ____                                 _  _                        #
#   / __ \    __ _  _ __ ____     __ _ (_)| |    ___  ____   _ __ ___    #
#  / / _  |  / _  || '_    _ \   / _  || || |   / __|/ _ \ | '_    _ \   #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
#######################################################################
#"""
#

# Yasin Kütük
# HLFS Soru ve yanıtları uyumlaştırma sistemi
print("HBS Soru ve yanıtları uyumlaştırma sistemi")
# R Kodları




####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,"systemfit",  "foreign",
              "dplyr", "sampleSelection","systemfit",
              "openxlsx")
paketkurulum(paketler)

####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
}
loadlib(x)



####03.Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle


path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"
prevpath <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"


#2002
hbs2002fert <- read.spss(paste0(path,"2002/Data/SPSS/fert_cd_02.sav"),
                         to.data.frame = T)
hbs2002hane <- read.spss(paste0(path,"2002/Data/SPSS/hane_cd_02.sav"),
```

```
                          to.data.frame = T)
#2003
hbs2003fert <- read.spss(paste0(path,"2003/Data/Spss/fert_cd.sav"),
                          to.data.frame = T)
hbs2003hane <- read.spss(paste0(path,"2003/Data/Spss/hane_cd.sav"),
                          to.data.frame = T)
#2004
hbs2004fert <- read.spss(paste0(path,"2004/Data/SPSS/fert_cd_04.sav"),
                          to.data.frame = T)
hbs2004hane <- read.spss(paste0(path,"2004/Data/SPSS/hane_cd_04.sav"),
                          to.data.frame = T)
#2005
hbs2005fert <- read.spss(paste0(path,"2005/Data/SPSS/fert_cd_05.sav"),
                          to.data.frame = T)
hbs2005hane <- read.spss(paste0(path,"2005/Data/SPSS/hane_cd_05.sav"),
                          to.data.frame = T)
#2006
hbs2006fert <- read.spss(paste0(path,"2006/Data/SPSS/fert_cd_06.sav"),
                          to.data.frame = T)
hbs2006hane <- read.spss(paste0(path,"2006/Data/SPSS/hane_cd_06.sav"),
                          to.data.frame = T)
#2007
hbs2007fert <- read.spss(paste0(path,"2007/Data/SPSS/fert_cd_07.sav"),
                          to.data.frame = T)
hbs2007hane <- read.spss(paste0(path,"2007/Data/SPSS/hane_cd_07.sav"),
                          to.data.frame = T)
#2008
hbs2008fert <- read.spss(paste0(path,"2008/Data/SPSS/fert_cd_08.sav"),
                          to.data.frame = T)
hbs2008hane <- read.spss(paste0(path,"2008/Data/SPSS/hane_cd_08.sav"),
                          to.data.frame = T)
#2009
hbs2009fert <- read.spss(paste0(path,"2009/Data/SPSS/fert_cd_09.sav"),
                          to.data.frame = T)
hbs2009hane <- read.spss(paste0(path,"2009/Data/SPSS/hane_cd_09.sav"),
                          to.data.frame = T)
#2010
hbs2010fert <- read.spss(paste0(path,"2010/Data/SPSS/fert_cd_2010.sav"),
                          to.data.frame = T)
hbs2010hane <- read.spss(paste0(path,"2010/Data/SPSS/hane_cd_2010.sav"),
                          to.data.frame = T)
#2011
hbs2011fert <- read.spss(paste0(path,"2011/Data/SPSS/fert_cd_2011.sav"),
                          to.data.frame = T)
hbs2011hane <- read.spss(paste0(path,"2011/Data/SPSS/hane_cd_2011.sav"),
                          to.data.frame = T)
#2012
hbs2012fert <- read.csv(paste0(path,"2012/downloads/fert_cd_2012.csv"),
                         sep=";", header = T)
hbs2012hane <- read.table(paste0(path,"2012/downloads/hane_cd_2012.csv"),
                          sep=";", header = T,fileEncoding = 'latin1')
#2013
hbs2013fert <- read.csv(paste0(path,"2013/turkce/downloads/fert_cd_2013.csv"),
                         sep=";", header = T)
hbs2013hane <- read.table(paste0(path,"2013/turkce/downloads/hane_cd_2013.csv"),
                          sep=";", header = T,fileEncoding = 'latin1')
#2014
hbs2014fert <- read.csv(paste0(path,"2014/downloads/fert_cd_2014.csv"),
                         sep=";", header = T)
hbs2014hane <- read.table(paste0(path,"2014/downloads/hane_cd_2014.csv"),
                          sep=";", header = T,fileEncoding = 'latin1')
#2015
hbs2015fert <- read.csv(paste0(path,"2015/downloads/fert_cd_2015.csv"),
                         sep=",", header = T,fileEncoding = 'latin1')
hbs2015hane <- read.table(paste0(path,"2015/downloads/hane_cd_2015.csv"),
                          sep=",", header = T,fileEncoding = 'latin1')
#2016
hbs2016fert <- read.csv(paste0(path,"2016/downloads/fert_cd_2016.csv"),
                         sep=",", header = T)
hbs2016hane <- read.table(paste0(path,"2016/downloads/hane_cd_2016.csv"),
                          sep=",", header = T,fileEncoding = 'latin1')

colnames(hbs2015fert) <- colnames(hbs2016fert)
colnames(hbs2015hane) <- colnames(hbs2016hane)
```

```
#04.Loop for Save them####
hbs <- ls()[grep('hbs',ls(),fixed=T)]

####Değişkenlerin adlarını küçük harf yap####
for(i in hbs){
  sil <- get(i)
  names(sil) <- tolower(colnames(sil))
  assign(i,sil)
  save(sil,file = paste0(path,'RData/',i,'.rdata'))
  rm(sil)
}
```

3.2 R Codes Used to Merge IL Part of HBSs

```
## -*- coding: utf-8 -*-
#"""
#Created on Wed Dec 20 23:11:04 2017

#######################################################################
#                          _          _              _          _     #
#     _   _   __ _   ___  (_)  _ __   | | __ _    _ | |_  _   _ | | __ #
#    | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ / #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <  #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\ #
#     |___/                                                         #
#     ____                                 _  _                     #
#    / __ \    __ _  _ __ ___     __ _ (_)| |     ___  ___    _ __ ___   #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_   _ \  #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK          ######################################
#@web   : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
#"""
#

# Yasin Kütük
# HBS FERT Soru ve yanıtları uyumlaştırma sistemi
print("HBS Soru ve yanıtları uyumlaştırma sistemi")
# R Kodları




####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,"systemfit",
              "foreign", "dplyr", "sampleSelection",
              "systemfit", "openxlsx")
paketkurulum(paketler)

####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
}
loadlib(x)



####03.Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle

path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/RData/"
prevpath <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"

veriler <- list.files(path=path)

for(i in veriler){
  load(paste0(path,i))
  assign(substr(i,1,nchar(i)-6),sil)
  rm(sil)
```

```
}

####Integer ve Numeric Dönüşümü####
numer <- function(x){as.numeric(as.character(x))}

for(i in fullf){
  k <- get(i)
  for(j in 1:dim(k)[2]){
    k[,j] <- sapply(k[,j],as.character)
  }
  assign(i,k)
  rm(k)
}

integ <- function(x){as.integer(as.character(x))}
chars <- function(x){as.character(x)}
chars <- function(x) {transform(x,
                                class=as.numeric(as.character(x) ) ) }



#Kontrol Parametresi####
hbsf <- ls()[grep('fert',ls(),fixed=T)]
vector <- list()
for(j in hbsf){
  dim <- dim(get(j))[1]
  print(dim)
  vector[match(j ,hbsf)] <- as.integer(dim)
}


#Tüm değişkenleri char olarak kodla, factor gelenler hata verdiriyor####
hbsf <- ls()[grep('fert',ls(),fixed=T)]
for(i in hbsf){
  k <- get(i)
  for(j in 1:dim(k)[2]){
    k[,j] <- sapply(k[,j],as.character)
  }
  assign(i,k)
  rm(k)
}







#___UZUNLAYICILAR####
h16 <- dim(hbs2016fert)[1]
h15 <- dim(hbs2015fert)[1]
h14 <- dim(hbs2014fert)[1]
h13 <- dim(hbs2013fert)[1]
h12 <- dim(hbs2012fert)[1]
h11 <- dim(hbs2011fert)[1]
h10 <- dim(hbs2010fert)[1]
h9 <-  dim(hbs2009fert)[1]
h8 <-  dim(hbs2008fert)[1]
h7 <-  dim(hbs2007fert)[1]
h6 <-  dim(hbs2006fert)[1]
h5 <-  dim(hbs2005fert)[1]
h4 <-  dim(hbs2004fert)[1]
h3 <-  dim(hbs2003fert)[1]
h2 <-  dim(hbs2002fert)[1]

#Row dimension#####
toplamuzunluk <- sum(h16,h15,h14,h13,h12,h11,h10,h9,h8,h7,h6,h5,h4,h3,h2)


#___Yıllar####
fullf <- data.frame(matrix(NA, nrow = toplamuzunluk))
fullf[,1] <- data.frame(c(rep(2016,length(hbs2016fert$birimno)),
                          rep(2015,length(hbs2015fert$birimno)),
                          rep(2014,length(hbs2014fert$bulten)),
```

```
                          rep(2013,length(hbs2013fert$bulten)),
                          rep(2012,length(hbs2012fert$bulten)),
                          rep(2011,length(hbs2011fert$bulten)),
                          rep(2010,length(hbs2010fert$bulten)),
                          rep(2009,length(hbs2009fert$bulten)),
                          rep(2008,length(hbs2008fert$bulten)),
                          rep(2007,length(hbs2007fert$bulten)),
                          rep(2006,length(hbs2006fert$bulten)),
                          rep(2005,length(hbs2005fert$bulten)),
                          rep(2004,length(hbs2004fert$bulten)),
                          rep(2003,length(hbs2003fert$bulten)),
                          rep(2002,length(hbs2002fert$bulten))))

colnames(fullf)[1] <- "year"
fullf[,1] <- integ(fullf[,1])

#Kontrol
length(fullf[,1])==toplamuzunluk



#___Bulten No####
fullf[,2] <- data.frame(c(hbs2016fert$birimno,
                          hbs2015fert$birimno,
                          hbs2014fert$bulten,
                          hbs2013fert$bulten,
                          hbs2012fert$bulten,
                          hbs2011fert$bulten,
                          hbs2010fert$bulten,
                          hbs2009fert$bulten,
                          hbs2008fert$bulten,
                          hbs2007fert$bulten,
                          hbs2006fert$bulten,
                          hbs2005fert$bulten,
                          hbs2004fert$bulten,
                          hbs2003fert$bulten,
                          hbs2002fert$bulten))

colnames(fullf)[2] <- "sheet"
fullf[,2] <- integ(fullf[,2])

#___Fert No####
fullf[,3] <- data.frame(c(hbs2016fert$fertno,
                          hbs2015fert$fertno,
                          hbs2014fert$fertno,
                          hbs2013fert$fertno,
                          hbs2012fert$fertno,
                          hbs2011fert$fertno,
                          hbs2010fert$fertno,
                          hbs2009fert$fertno,
                          hbs2008fert$fertno,
                          hbs2007fert$fertno,
                          hbs2006fert$fertno,
                          hbs2005fert$fertno,
                          hbs2004fert$fertno,
                          hbs2003fert$fertno,
                          hbs2002fert$fertno))

colnames(fullf)[3] <- "indiv"
fullf[,3] <- integ(fullf[,3])

#Kontrol
length(fullf[,3])==toplamuzunluk

#___Cinsiyet####
fullf[,4] <- data.frame(c(hbs2016fert$cinsiyet,
                          hbs2015fert$cinsiyet,
                          hbs2014fert$cinsiyet,
                          hbs2013fert$cinsiyet,
                          hbs2012fert$cinsiyet,
                          hbs2011fert$cinsiyet,
                          hbs2010fert$cinsiyet,
                          hbs2009fert$cinsiyet,
                          hbs2008fert$cinsiyet,
```

```
                          hbs2007fert$cinsiyet,
                          hbs2006fert$cinsiyet,
                          hbs2005fert$cinsiyet,
                          hbs2004fert$cinsiyet,
                          hbs2003fert$cinsiyet,
                          hbs2002fert$cinsiyet))

colnames(fullf)[4] <- "gender"
fullf[,4] <- integ(fullf[,4])

#Kontrol
length(fullf[,4])==toplamuzunluk

#___Yaş####
fullf[,5] <- data.frame(c(hbs2016fert$yas,
                          hbs2015fert$yas,
                          hbs2014fert$yas,
                          hbs2013fert$yas,
                          hbs2012fert$yas,
                          hbs2011fert$yas,
                          hbs2010fert$yas,
                          hbs2009fert$yas,
                          hbs2008fert$yas,
                          hbs2007fert$yas,
                          hbs2006fert$yas,
                          hbs2005fert$yas,
                          hbs2004fert$yas,
                          hbs2003fert$yas,
                          hbs2002fert$yas))

colnames(fullf)[5] <- "age"
fullf[,5] <- integ(fullf[,5])



#Kontrol
length(fullf[,5])==toplamuzunluk


#Düzeltmeler####
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & fullf$age<=5] <- 1
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=6 & fullf$age<=14) ] <- 2
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=15 & fullf$age<=19) ] <- 3
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=20 & fullf$age<=24) ] <- 4
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=25 & fullf$age<=29) ] <- 5
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=30 & fullf$age<=34) ] <- 6
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=35 & fullf$age<=39) ] <- 7
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=40 & fullf$age<=44) ] <- 8
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=45 & fullf$age<=49) ] <- 9
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=50 & fullf$age<=54) ] <- 10
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=55 & fullf$age<=59) ] <- 11
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>=60 & fullf$age<=64) ] <- 12
fullf$age[(fullf$year<2006 | fullf$year>2010)
          & (fullf$age>64 ) ] <- 13


#Tabulation Control####
table(fullf$age,fullf$year)
```

```
#___Yakınlık####
fullf[,6] <- data.frame(c(hbs2016fert$yakinlik,
                          hbs2015fert$yakinlik,
                          hbs2014fert$yakinlik,
                          hbs2013fert$yakinlik,
                          hbs2012fert$yakinlik,
                          hbs2011fert$yakinlik,
                          hbs2010fert$yakinlik,
                          hbs2009fert$yakinlik,
                          hbs2008fert$yakinlik,
                          hbs2007fert$yakinlik,
                          hbs2006fert$yakinlik,
                          hbs2005fert$yakinlik,
                          hbs2004fert$yakinlik,
                          hbs2003fert$yakinlik,
                          hbs2002fert$yakinlik))

colnames(fullf)[6] <- "relation"

fullf[,6] <- integ(fullf$relation)



#Kontrol
length(fullf[,6])==toplamuzunluk

#Düzeltmeler####
fullf$relation[(fullf$year==2002
                & fullf$relation==9)] <- 10

fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==9] <- 10
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==8] <- 9
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==7] <- 8
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==6] <- 7
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==5] <- 6
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==4] <- 5


fullf$relation[(fullf$year>=2012
                & fullf$year<=2014) & fullf$relation==4] <- 44
fullf$relation[(fullf$year>=2012
                & fullf$year<=2014) & fullf$relation==5] <- 55
fullf$relation[(fullf$year>=2012
                & fullf$year<=2014) & fullf$relation==6] <- 66
fullf$relation[(fullf$year>=2012
                & fullf$year<=2014) & fullf$relation==44] <- 5
fullf$relation[(fullf$year>=2012
                & fullf$year<=2014) & fullf$relation==55] <- 6
fullf$relation[(fullf$year>=2012
                & fullf$year<=2014) & fullf$relation==66] <- 4




fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==9] <- 10
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==8] <- 9
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==7] <- 8
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==6] <- 7
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==5] <- 6
fullf$relation[(fullf$year>=2009
                & fullf$year<=2011) & fullf$relation==4] <- 5
```

```
fullf$relation[(fullf$year>=2015
                & fullf$relation==1)] <- 0
fullf$relation[(fullf$year>=2015
                & fullf$relation==2)] <- 1
fullf$relation[(fullf$year>=2015
                & fullf$relation==3)] <- 2
fullf$relation[(fullf$year>=2015
                & fullf$relation==4)] <- 3
fullf$relation[(fullf$year>=2015
                & fullf$relation==5)] <- 4
fullf$relation[(fullf$year>=2015
                & fullf$relation==6)] <- 5
fullf$relation[(fullf$year>=2015
                & fullf$relation==7)] <- 6
fullf$relation[(fullf$year>=2015
                & fullf$relation==8)] <- 7
fullf$relation[(fullf$year>=2015
                & fullf$relation==9)] <- 8
fullf$relation[(fullf$year>=2015
                & fullf$relation==10)] <- 9
fullf$relation[(fullf$year>=2015
                & fullf$relation==11)] <- 10


table(fullf$relation,fullf$year)


#___Aile Grup####
fullf[,7] <- data.frame(c(rep(NA,sum(h16,h15,
                                     h14,h13,h12,h11,h10,h9,h8)),
                          hbs2007fert$ailegrup,
                          hbs2006fert$ailegrup,
                          hbs2005fert$ailegrup,
                          hbs2004fert$ailegrup,
                          hbs2003fert$ailegrup,
                          hbs2002fert$ailegrup))

colnames(fullf)[7] <- "familygroup"
fullf[,7] <- integ(fullf[,7])

#Kontrol
length(fullf[,7])==toplamuzunluk



#___Sağlık Sigortası####
fullf[,8] <- data.frame(c(hbs2016fert$saglik_sigorta_1,
                          hbs2015fert$saglik_sigorta_1,
                          hbs2014fert$sagliks,
                          hbs2013fert$sagliks,
                          hbs2012fert$sagliks,
                          hbs2011fert$sagliks,
                          hbs2010fert$sagliks,
                          hbs2009fert$sagliks,
                          hbs2008fert$sagliks,
                          hbs2007fert$sagliks,
                          hbs2006fert$sagliks,
                          hbs2005fert$sagliks,
                          hbs2004fert$sagliks,
                          hbs2003fert$sagliks,
                          hbs2002fert$sagliks))


colnames(fullf)[8] <- "health.insurance"
fullf[,8] <- integ(fullf[,8])

#Kontrol
length(fullf[,8])==toplamuzunluk


#Düzeltmeler####
fullf$health.insurance[(fullf$year>=2002
```

```
                        & fullf$year<=2004 ) & (fullf$health.insurance==3)] <- 2
fullf$health.insurance[(fullf$year>=2002
                        & fullf$year<=2004 ) & (fullf$health.insurance==4)] <- 3
fullf$health.insurance[(fullf$year>=2002
                        & fullf$year<=2004 ) & (fullf$health.insurance==5)] <- 4


fullf$health.insurance[(fullf$year==2008 )
                       & (fullf$health.insurance==2)] <- 1
fullf$health.insurance[(fullf$year==2008 )
                       & (fullf$health.insurance==3)] <- 1
fullf$health.insurance[(fullf$year==2008 )
                       & (fullf$health.insurance==4)] <- 2
fullf$health.insurance[(fullf$year==2008 )
                       & (fullf$health.insurance==5)] <- 2
fullf$health.insurance[(fullf$year==2008 )
                       & (fullf$health.insurance==6)] <- 3
fullf$health.insurance[(fullf$year==2008 )
                       & (fullf$health.insurance==7)] <- 4


fullf$health.insurance[(fullf$year>=2009
                        & fullf$year<=2011 ) & (fullf$health.insurance==3)] <- 2
fullf$health.insurance[(fullf$year>=2009
                        & fullf$year<=2011 ) & (fullf$health.insurance==4)] <- 3
fullf$health.insurance[(fullf$year>=2009
                        & fullf$year<=2011 ) & (fullf$health.insurance==5)] <- 4


fullf$health.insurance[(fullf$year>=2012
                        & fullf$year<=2014 ) & (fullf$health.insurance==3)] <- 2
fullf$health.insurance[(fullf$year>=2012
                        & fullf$year<=2014 ) & (fullf$health.insurance==4)] <- 2
fullf$health.insurance[(fullf$year>=2012
                        & fullf$year<=2014 ) & (fullf$health.insurance==5)] <- 3
fullf$health.insurance[(fullf$year>=2012
                        & fullf$year<=2014 ) & (fullf$health.insurance==6)] <- 4

fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==2)] <- 4
fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==11)] <- 1
fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==12)] <- 1
fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==13)] <- 1
fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==14)] <- 2
fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==15)] <- 2
fullf$health.insurance[(fullf$year>=2015 )
                       & (fullf$health.insurance==16)] <- 3



table(fullf$health.insurance,fullf$year)


#___Sağlık Sigortası2####
fullf[,9] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12,h11,h10,h9)),
                          hbs2008fert$sagliks2,
                          rep(NA,h7),
                          hbs2006fert$sagliks2,
                          hbs2005fert$sagliks2,
                          rep(NA,sum(h4,h3,h2))))

colnames(fullf)[9] <- "health.insurance2"
fullf[,9] <- integ(fullf[,9])

#Kontrol
length(fullf[,9])==toplamuzunluk


#___Engellilik Günlük####
```

```
fullf[,10] <- data.frame(c(hbs2016fert$sakatlik_gunluk,
                           hbs2015fert$sakatlik_gunluk,
                           hbs2014fert$gunengel,
                           hbs2013fert$gunengel,
                           hbs2012fert$gunengel,
                           hbs2011fert$gunengel,
                           hbs2010fert$gunengel,
                           hbs2009fert$gunengel,
                           hbs2008fert$gunengel,
                           hbs2007fert$gunengel,
                           hbs2006fert$gunengel,
                           hbs2005fert$gunengel,
                           hbs2004fert$gunengel,
                           rep(NA,sum(h3,h2))))

colnames(fullf)[10] <- "disability.daily"
fullf[,10] <- integ(fullf[,10])

#Kontrol
length(fullf[,10])==toplamuzunluk



#___Engellilik Çalışma####
fullf[,11] <- data.frame(c(hbs2016fert$sakatlik_gunluk,
                           hbs2015fert$sakatlik_gunluk,
                           hbs2014fert$calengel,
                           hbs2013fert$calengel,
                           hbs2012fert$calengel,
                           hbs2011fert$calengel,
                           hbs2010fert$calengel,
                           hbs2009fert$calengel,
                           hbs2008fert$calengel,
                           hbs2007fert$calengel,
                           hbs2006fert$calengel,
                           hbs2005fert$calengel,
                           hbs2004fert$calengel,
                           rep(NA,sum(h3,h2))))

colnames(fullf)[11] <- "disability.working"
fullf[,11] <- integ(fullf[,11])

#Kontrol
length(fullf[,11])==toplamuzunluk



#___Eğitim####
fullf[,12] <- data.frame(c(hbs2016fert$okul_biten,
                           hbs2015fert$okul_biten,
                           hbs2014fert$egitim,
                           hbs2013fert$egitim,
                           hbs2012fert$egitim,
                           hbs2011fert$egitim,
                           hbs2010fert$egitim,
                           hbs2009fert$egitim,
                           hbs2008fert$egitim,
                           hbs2007fert$egitim,
                           hbs2006fert$egitim,
                           hbs2005fert$egitim,
                           hbs2004fert$egitim,
                           hbs2003fert$egitim,
                           hbs2002fert$egitim))

colnames(fullf)[12] <- "education"
fullf[,12] <- integ(fullf[,12])

#Kontrol
length(fullf[,12])==toplamuzunluk

#Düzeltmeler####
fullf$education[(fullf$year>=2015 ) & (fullf$education==62)] <- 11
fullf$education[(fullf$year>=2015 ) & (fullf$education==7)] <- 11
fullf$education[(fullf$year>=2015 ) & (fullf$education==61)] <- 10
```

```
fullf$education[(fullf$year>=2015 ) & (fullf$education==5)] <- 10
fullf$education[(fullf$year>=2015 ) & (fullf$education==4)] <- 9
fullf$education[(fullf$year>=2015 ) & (fullf$education==32)] <- 8
fullf$education[(fullf$year>=2015 ) & (fullf$education==31)] <- 7
fullf$education[(fullf$year>=2015 ) & (fullf$education==22)] <- 5
fullf$education[(fullf$year>=2015 ) & (fullf$education==21)] <- 5
fullf$education[(fullf$year>=2015 ) & (fullf$education==23)] <- 4
fullf$education[(fullf$year>=2015 ) & (fullf$education==1)] <- 3
fullf$education[(fullf$year>=2015 ) & (fullf$education==0)] <- 1


table(fullf$education,fullf$year)




#___Eğitim Devam####
fullf[,13] <- data.frame(c(hbs2016fert$okul_devam,
                           hbs2015fert$okul_devam,
                           hbs2014fert$ogrenci,
                           hbs2013fert$ogrenci,
                           hbs2012fert$ogrenci,
                           hbs2011fert$ogrenci,
                           hbs2010fert$ogrenci,
                           hbs2009fert$ogrenci,
                           rep(NA,h8),
                           hbs2007fert$ogrenci,
                           hbs2006fert$ogrenci,
                           hbs2005fert$ogrenci,
                           hbs2004fert$ogrenci,
                           hbs2003fert$ogrenci,
                           hbs2002fert$ogrenci))

colnames(fullf)[13] <- "education.contd"
fullf[,13] <- integ(fullf[,13])

#Kontrol
length(fullf[,13])==toplamuzunluk

#Düzeltmeler####
fullf$education.contd[(fullf$year==2002 ) & (fullf$education.contd==2)] <- 1
fullf$education.contd[(fullf$year==2002 ) & (fullf$education.contd==3)] <- 2

fullf$education.contd[(fullf$year>=2015 ) & (fullf$education.contd>=2)] <- 2




#___Eğitim Devam2####
fullf[,14] <- data.frame(c(rep(NA,sum(h16,h15,h14)),
                           hbs2013fert$egitim_devam,
                           hbs2012fert$egitim_devam,
                           hbs2011fert$egitim_devam,
                           hbs2010fert$egitim_d,
                           hbs2009fert$egitim_devam,
                           hbs2008fert$egitim_devam,
                           hbs2007fert$egitim_devam,
                           rep(NA,sum(h6,h5,h4,h3,h2))
))

colnames(fullf)[14] <- "education.contd2"
fullf[,14] <- integ(fullf[,14])

#Kontrol
length(fullf[,14])==toplamuzunluk


#Düzeltmeler, geriye dönük uyumlaştırma####
fullf$education.contd[(fullf$year==2008 ) ] <- 2
fullf$education.contd[(fullf$year==2008 ) & (fullf$education.contd2>=0)] <- 1
```

```
table(fullf$education.contd,fullf$year)


#___Medeni Durum####
fullf[,15] <- data.frame(c(hbs2016fert$medeni_durum,
                           hbs2015fert$medeni_durum,
                           hbs2014fert$medenidr,
                           hbs2013fert$medenidr,
                           hbs2012fert$medenidr,
                           hbs2011fert$medenidr,
                           hbs2010fert$medenidr,
                           hbs2009fert$medenidr,
                           hbs2008fert$medenidr,
                           hbs2007fert$medenidr,
                           hbs2006fert$medenidr,
                           hbs2005fert$medenidr,
                           hbs2004fert$medenidr,
                           hbs2003fert$medenidr,
                           hbs2002fert$medenidr))

colnames(fullf)[15] <- "maritial.status"
fullf[,15] <- integ(fullf[,15])

#Kontrol
length(fullf[,15])==toplamuzunluk


#Düzeltmeler####
#HBS Anket düzeltmesinde 2008 yanlış kodlanmış, 2008 ile 2009 uyumlu####
fullf$maritial.status[(fullf$year<=2007 ) & (fullf$maritial.status==3)] <- 2
fullf$maritial.status[(fullf$year<=2007 ) & (fullf$maritial.status==4)] <- 3
fullf$maritial.status[(fullf$year<=2007 ) & (fullf$maritial.status==5)] <- 4
fullf$maritial.status[(fullf$year<=2007 ) & (fullf$maritial.status==6)] <- 4


fullf$maritial.status[(fullf$year>=2015 ) & (fullf$maritial.status==3)] <- 44
fullf$maritial.status[(fullf$year>=2015 ) & (fullf$maritial.status==4)] <- 33
fullf$maritial.status[(fullf$year>=2015 ) & (fullf$maritial.status==33)] <- 3
fullf$maritial.status[(fullf$year>=2015 ) & (fullf$maritial.status==44)] <- 4


table(fullf$maritial.status,fullf$year)



#___Çalışma Durumu####
fullf[,16] <- data.frame(c(hbs2016fert$calisma_durum,
                           hbs2015fert$calisma_durum,
                           hbs2014fert$calis_ay,
                           hbs2013fert$calis_ay,
                           hbs2012fert$calis_ay,
                           hbs2011fert$calis_ay,
                           hbs2010fert$calis_ay,
                           hbs2009fert$calis_ay,
                           hbs2008fert$calis_ay,
                           hbs2007fert$calis_ay,
                           hbs2006fert$calis_ay,
                           hbs2005fert$calis_ay,
                           hbs2004fert$calis_ay,
                           hbs2003fert$calis_ay,
                           hbs2002fert$calis_ay))

colnames(fullf)[16] <- "work.status"
fullf[,16] <- integ(fullf[,16])

#Kontrol
length(fullf[,16])==toplamuzunluk

#Düzeltmeler####
fullf$work.status[(fullf$year>=2015 ) & (fullf$work.status==2)] <- 1
fullf$work.status[(fullf$year>=2015 ) & (fullf$work.status==3)] <- 2
```

```
#Tabulation Kontrol####
table(fullf$work.status,fullf$year)



#___İş Arama####
fullf[,17] <- data.frame(c(hbs2016fert$is_arama,
                           hbs2015fert$is_arama,
                           hbs2014fert$is_arama,
                           hbs2013fert$is_arama,
                           hbs2012fert$is_arama,
                           hbs2011fert$is_arama,
                           hbs2010fert$is_arama,
                           hbs2009fert$is_arama,
                           hbs2008fert$is_arama,
                           hbs2007fert$is_arama,
                           hbs2006fert$is_arama,
                           hbs2005fert$is_arama,
                           hbs2004fert$is_arama,
                           hbs2003fert$is_arama,
                           hbs2002fert$is_arama))

colnames(fullf)[17] <- "work.search"
fullf[,17] <- integ(fullf[,17])

#Kontrol
length(fullf[,17])==toplamuzunluk



table(fullf$work.search,fullf$year)



#___İş aramama Nedeni####
fullf[,18] <- data.frame(c(hbs2016fert$isaramama_neden,
                           hbs2015fert$isaramama_neden,
                           hbs2014fert$isneden,
                           hbs2013fert$isneden,
                           hbs2012fert$isneden,
                           hbs2011fert$isneden,
                           hbs2010fert$isneden,
                           hbs2009fert$isneden,
                           hbs2008fert$isneden,
                           hbs2007fert$isneden,
                           hbs2006fert$isneden,
                           hbs2005fert$isneden,
                           hbs2004fert$isneden,
                           hbs2003fert$isneden,
                           hbs2002fert$isneden))

colnames(fullf)[18] <- "work.nosearch"
fullf[,18] <- integ(fullf[,18])

#Kontrol
length(fullf[,18])==toplamuzunluk



#Düzeltmeler####
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==3)] <- NA
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==4)] <- NA

fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==5)] <- 22
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==6)] <- 33
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==7)] <- 44
fullf$work.nosearch[(fullf$year>=2015 )
```

```
                    & (fullf$work.nosearch==8)] <- 77
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==9)] <- 66
fullf$work.nosearch[(fullf$year>=2015 ) & (fullf$work.nosearch==10)] <- 55



fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==2)] <- 10
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==98)] <- 11


fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==22)] <- 2
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==33)] <- 3
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==44)] <- 4
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==55)] <- 5
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==66)] <- 6
fullf$work.nosearch[(fullf$year>=2015 )
                    & (fullf$work.nosearch==77)] <- 7

fullf$work.nosearch[(fullf$year<=2014 )
                    & (fullf$work.nosearch==8)] <- 7
fullf$work.nosearch[(fullf$year<=2014 )
                    & (fullf$work.nosearch==9)] <- NA

#Tabulation Kontrol####
table(fullf$work.nosearch,fullf$year)


#___İşbaşı Durumu####
fullf[,19] <- data.frame(c(hbs2016fert$isbasi,
                           hbs2015fert$isbasi,
                           hbs2014fert$isbasi,
                           hbs2013fert$isbasi,
                           hbs2012fert$isbasi,
                           hbs2011fert$isbasi,
                           hbs2010fert$isbasi,
                           hbs2009fert$isbasi,
                           hbs2008fert$isbasi,
                           hbs2007fert$isbasi,
                           hbs2006fert$isbasi,
                           hbs2005fert$isbasi,
                           hbs2004fert$isbasi,
                           hbs2003fert$isbasi,
                           hbs2002fert$isbasi))

colnames(fullf)[19] <- "work.ready"
fullf[,19] <- integ(fullf[,19])

#Kontrol
length(fullf[,19])==toplamuzunluk



#___Çalışma Durumu####
fullf[,20] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$calis_yl,
                           hbs2010fert$calis_yl,
                           hbs2009fert$calis_yl,
                           hbs2008fert$calis_yl,
                           hbs2007fert$calis_yl,
                           hbs2006fert$calis_yl,
                           hbs2005fert$calis_yl,
                           hbs2004fert$calis_yl,
                           hbs2003fert$calis_yl,
                           hbs2002fert$calis_yl))
```

```
colnames(fullf)[20] <- "work.experience"
fullf[,20] <- integ(fullf[,20])

#Kontrol
length(fullf[,20])==toplamuzunluk



#___Sektör Kodu####
fullf[,21] <- data.frame(c(hbs2016fert$ikfa_1,
                           hbs2015fert$ikfa_1,
                           hbs2014fert$sekkod,
                           hbs2013fert$sekkod,
                           hbs2012fert$sekkod,
                           hbs2011fert$sekkod,
                           hbs2010fert$sekkod,
                           hbs2009fert$sekkod,
                           hbs2008fert$sekkod,
                           hbs2007fert$sekkod,
                           hbs2006fert$sekkod,
                           hbs2005fert$sekkod,
                           hbs2004fert$sekkod,
                           hbs2003fert$sekkod,
                           hbs2002fert$sekkod))

colnames(fullf)[21] <- "sector.code"
fullf[,21] <- integ(fullf[,21])

#Kontrol
length(fullf[,21])==toplamuzunluk



#Düzeltmeler, NaceRev2'yi NaceRev1'e indirgiyorum####
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==2)] <- 1
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==3)] <- 2
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==4)] <- 3
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==5)] <- 4
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==6)] <- 5
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==7)] <- 6
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==8)] <- 7
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==9)] <- 8
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==10)] <- 9
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==11)] <- 10
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==12)] <- 11
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==13)] <- 12
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==14)] <- 13
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==15)] <- 14
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==16)] <- 14
fullf$sector.code[(fullf$year<=2003 )
                  & (fullf$sector.code==17)] <- 14


fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==5)] <- 4
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==6)] <- 5
fullf$sector.code[(fullf$year>=2015)
```

```
                  & (fullf$sector.code==7)] <- 6
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==9)] <- 7
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==10)] <- 8
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==11)] <- 9
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==12)] <- 10
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==14)] <- 11
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==15)] <- 11
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==16)] <- 12
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==17)] <- 13
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==18)] <- 14
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==19)] <- 14
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==20)] <- 14
fullf$sector.code[(fullf$year>=2015)
                  & (fullf$sector.code==21)] <- 14



#2008'de hatalı kodlama var, oranlarına göre yeniden kodluyorum
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==3)] <- 2
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==4)] <- 3
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==5)] <- 4
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==6)] <- 5
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==7)] <- 6
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==8)] <- 7
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==9)] <- 8
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==10)] <- 9
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==11)] <- 10
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==12)] <- 11
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==13)] <- 12
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==14)] <- 13
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==15)] <- 14
fullf$sector.code[(fullf$year==2008)
                  & (fullf$sector.code==16)] <- 14



fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==9)] <- 8
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==10)] <- 9
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==11)] <- 10
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==12)] <- NA
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==13)] <- 11
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==14)] <- 11
```

```
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==15)] <- 12
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==16)] <- 13
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==17)] <- 14
fullf$sector.code[(fullf$year>=2009
                   & fullf$year<=2014) & (fullf$sector.code==18)] <- 14


#Tabulation Kontrol####
table(fullf$sector.code,fullf$year)


#___Meslek Kodu####
fullf[,22] <- data.frame(c(hbs2016fert$meslek_1,
                           hbs2015fert$meslek_1,
                           hbs2014fert$meslek,
                           hbs2013fert$meslek,
                           hbs2012fert$meslek,
                           hbs2011fert$meslek,
                           hbs2010fert$meslek,
                           hbs2009fert$meslek,
                           hbs2008fert$meslek,
                           hbs2007fert$meslek,
                           hbs2006fert$meslek,
                           hbs2005fert$meslek,
                           hbs2004fert$meslek,
                           hbs2003fert$meslek,
                           hbs2002fert$meslek))

colnames(fullf)[22] <- "occupation"
fullf[,22] <- integ(fullf[,22])

#Kontrol
length(fullf[,22])==toplamuzunluk



#Tabulation Kontrol####
table(fullf$occupation,fullf$year)




#___Şu anki İşindeki Deneyimi####
fullf[,23] <- data.frame(c(hbs2016fert$calisma_sure_yil,
                           hbs2015fert$calisma_sure_yil,
                           hbs2014fert$sure_yil,
                           hbs2013fert$sure_yil,
                           hbs2012fert$sure_yil,
                           hbs2011fert$sure_yil,
                           hbs2010fert$sure_yil,
                           hbs2009fert$sure_yil,
                           hbs2008fert$sure_yil,
                           hbs2007fert$sure_yil,
                           hbs2006fert$sure_yil,
                           hbs2005fert$sure_yil,
                           hbs2004fert$sure_yil,
                           hbs2003fert$sure_yil,
                           hbs2002fert$sure_yil))

colnames(fullf)[23] <- "work.experience2"
fullf[,23] <- integ(fullf[,23])

#Kontrol
length(fullf[,23])==toplamuzunluk



#___Çalışma Saati####
fullf[,24] <- data.frame(c(hbs2016fert$calisma_sure_saat,
                           hbs2015fert$calisma_sure_saat,
```

```
                            hbs2014fert$calsur,
                            hbs2013fert$calsur,
                            hbs2012fert$calsur,
                            hbs2011fert$calsur,
                            hbs2010fert$calsur,
                            hbs2009fert$calsur,
                            hbs2008fert$calsur,
                            hbs2007fert$calsur,
                            hbs2006fert$calsur,
                            hbs2005fert$calsur,
                            hbs2004fert$calsur,
                            hbs2003fert$calsur,
                            hbs2002fert$calsur))

colnames(fullf)[24] <- "work.hours"
fullf[,24] <- integ(fullf[,24])

#Kontrol
length(fullf[,24])==toplamuzunluk




#___Çalışan Kişi Sayısı####
fullf[,25] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$cal_kisi,
                            hbs2010fert$cal_kisi,
                            hbs2009fert$cal_kisi,
                            hbs2008fert$cal_kisi,
                            hbs2007fert$cal_kisi,
                            hbs2006fert$cal_kisi,
                            hbs2005fert$cal_kisi,
                            hbs2004fert$cal_kisi,
                            hbs2003fert$cal_kisi,
                            hbs2002fert$cal_kisi))

colnames(fullf)[25] <- "work.population"
fullf[,25] <- integ(fullf[,25])

#Kontrol
length(fullf[,25])==toplamuzunluk


#Tabulation Kontrol####
table(fullf$work.population,fullf$year)




#___Çalışan Kişi Sayısı2####
fullf[,26] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12,h11,h10,h9,h8)),
                            hbs2007fert$kac_kisi,
                            hbs2006fert$kac_kisi,
                            hbs2005fert$kac_kisi,
                            hbs2004fert$kac_kisi,
                            hbs2003fert$kac_kisi,
                            hbs2002fert$kac_kisi))

colnames(fullf)[26] <- "work.population2"
fullf[,26] <- integ(fullf[,26])

#Kontrol
length(fullf[,26])==toplamuzunluk



#___SGK Bilgisi####
fullf[,27] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
```

```
                            hbs2011fert$sgk,
                            hbs2010fert$sgk,
                            hbs2009fert$sgk,
                            hbs2008fert$sgk,
                            hbs2007fert$sgk,
                            hbs2006fert$sgk,
                            hbs2005fert$sgk,
                            hbs2004fert$sgk,
                            hbs2003fert$sgk,
                            hbs2002fert$sgk))

colnames(fullf)[27] <- "social.insurance"
fullf[,27] <- integ(fullf[,27])

#Kontrol
length(fullf[,27])==toplamuzunluk



#Düzeltmeler####
fullf$social.insurance[(fullf$year<=2008 ) & (fullf$social.insurance<=3)] <- 1
fullf$social.insurance[(fullf$year<=2008 ) & (fullf$social.insurance==4)] <- 2
fullf$social.insurance[(fullf$year<=2008 ) & (fullf$social.insurance==5)] <- 3


fullf$social.insurance[(fullf$year>=2009 ) & (fullf$social.insurance==3)] <- 22
fullf$social.insurance[(fullf$year>=2009 ) & (fullf$social.insurance==2)] <- 33
fullf$social.insurance[(fullf$year>=2009 ) & (fullf$social.insurance==22)] <- 2
fullf$social.insurance[(fullf$year>=2009 ) & (fullf$social.insurance==33)] <- 3



#Tabulation Kontrol####
table(fullf$social.insurance,fullf$year)




#___İş Statüsü####
fullf[,28] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$is_statu,
                            hbs2010fert$is_statu,
                            hbs2009fert$is_statu,
                            hbs2008fert$is_statu,
                            hbs2007fert$is_statu,
                            hbs2006fert$is_statu,
                            hbs2005fert$is_statu,
                            hbs2004fert$is_statu,
                            hbs2003fert$is_statu,
                            hbs2002fert$is_statu))

colnames(fullf)[28] <- "workplace.status"
fullf[,28] <- integ(fullf[,28])

#Kontrol
length(fullf[,28])==toplamuzunluk



#___Hukuki Durum####
fullf[,29] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12,h11,h10,h9)),
                            hbs2008fert$huk_dur,
                            hbs2007fert$huk_dur,
                            hbs2006fert$huk_dur,
                            hbs2005fert$huk_dur,
                            hbs2004fert$huk_dur,
                            hbs2003fert$huk_dur,
                            hbs2002fert$huk_dur))

colnames(fullf)[29] <- "justice"
fullf[,29] <- integ(fullf[,29])

#Kontrol
length(fullf[,29])==toplamuzunluk
```

```
#___Kooperatif Bilgisi####
fullf[,30] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$koopera,
                           hbs2010fert$koopera,
                           hbs2009fert$koopera,
                           rep(NA,sum(h8,h7,h6,h5,h4,h3,h2))))

colnames(fullf)[30] <- "cooperative"
fullf[,30] <- integ(fullf[,30])

#Kontrol
length(fullf[,30])==toplamuzunluk



#___İşyeri Ortaklı veya wage####
fullf[,31] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$sir_kon,
                           hbs2010fert$sir_kon,
                           hbs2009fert$sir_kon,
                           hbs2008fert$sir_kon,
                           hbs2007fert$sir_kon,
                           hbs2006fert$sir_kon,
                           hbs2005fert$sir_kon,
                           hbs2004fert$sir_kon,
                           hbs2003fert$sir_kon,
                           hbs2002fert$sir_kon))

colnames(fullf)[31] <- "workplace.partner"
fullf[,31] <- integ(fullf[,31])

#Kontrol
length(fullf[,31])==toplamuzunluk



fullf[,32] <- NA
#___İşteki Durumu####
fullf[,32] <- data.frame(c(hbs2016fert$isteki_durum,
                           hbs2015fert$isteki_durum,
                           hbs2014fert$isdur,
                           hbs2013fert$isdur,
                           hbs2012fert$isdur,
                           hbs2011fert$isdur,
                           hbs2010fert$isdur,
                           hbs2009fert$isdur,
                           hbs2008fert$isdur,
                           hbs2007fert$isdur,
                           hbs2006fert$isdur,
                           hbs2005fert$isdur,
                           hbs2004fert$isdur,
                           hbs2003fert$isdur,
                           hbs2002fert$isdur))

colnames(fullf)[32] <- "working.status"
fullf[,32] <- integ(fullf[,32])

#Kontrol
length(fullf[,32])==toplamuzunluk


#Düzeltmeler####
fullf$working.status[(fullf$year<=2004 ) & (fullf$working.status==3)] <- 2
fullf$working.status[(fullf$year<=2004 ) & (fullf$working.status==4)] <- 3
fullf$working.status[(fullf$year<=2004 ) & (fullf$working.status==5)] <- 4
fullf$working.status[(fullf$year<=2004 ) & (fullf$working.status==6)] <- 5

fullf$working.status[(fullf$year>=2015 ) & (fullf$working.status==4)] <- 5
fullf$working.status[(fullf$year>=2015 ) & (fullf$working.status==3)] <- 4
fullf$working.status[(fullf$year>=2015 ) & (fullf$working.status==2)] <- 3
```

```
fullf$working.status[(fullf$year>=2015 ) & (fullf$working.status==12)] <- 2
fullf$working.status[(fullf$year>=2015 ) & (fullf$working.status==11)] <- 1



#Tabulation Kontrol####
table(fullf$working.status,fullf$year)




#___Çalışma ve Süreklilik####
fullf[,33] <- data.frame(c(hbs2016fert$is_sureklilik,
                           hbs2015fert$is_sureklilik,
                           hbs2014fert$ist_sek,
                           hbs2013fert$ist_sek,
                           hbs2012fert$ist_sek,
                           hbs2011fert$ist_sek,
                           hbs2010fert$ist_sek,
                           hbs2009fert$ist_sek,
                           hbs2008fert$ist_sek,
                           hbs2007fert$ist_sek,
                           hbs2006fert$ist_sek,
                           hbs2005fert$ist_sek,
                           hbs2004fert$ist_sek,
                           hbs2003fert$ist_sek,
                           hbs2002fert$ist_sek))

colnames(fullf)[33] <- "workplace.contd"
fullf[,33] <- integ(fullf[,33])



#Kontrol
length(fullf[,33])==toplamuzunluk



#Düzeltmeler####
fullf$workplace.contd[(fullf$year>=2015 ) & (fullf$workplace.contd==21)] <- 2
fullf$workplace.contd[(fullf$year>=2015 ) & (fullf$workplace.contd>=22)] <- 3


#Tabulation Kontrol####
table(fullf$workplace.contd,fullf$year)



#___İşyerinde güç kullanımı####
fullf[,34] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$guc_kul,
                           hbs2010fert$guc_kul,
                           hbs2009fert$guc_kul,
                           hbs2008fert$guc_kul,
                           hbs2007fert$guc_kul,
                           hbs2006fert$guc_kul,
                           hbs2005fert$guc_kul,
                           hbs2004fert$guc_kul,
                           hbs2003fert$guc_kul,
                           rep(NA,h2)))

colnames(fullf)[34] <- "workplace.workforce"
fullf[,34] <- integ(fullf[,34])

#Kontrol
length(fullf[,34])==toplamuzunluk

#Tabulation Kontrol####
table(fullf$workplace.workforce,fullf$year)



#___Sendika Üyeliği####
```

```
fullf[,35] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$sendika,
                           hbs2010fert$sendika,
                           hbs2009fert$sendika,
                           hbs2008fert$sendika,
                           hbs2007fert$sendika,
                           hbs2006fert$sendika,
                           hbs2005fert$sendika,
                           hbs2004fert$sendika,
                           hbs2003fert$sendika,
                           hbs2002fert$sendika))

colnames(fullf)[35] <- "workplace.syndicate"
fullf[,35] <- integ(fullf[,35])

#Kontrol
length(fullf[,35])==toplamuzunluk

#Tabulation Kontrol####
table(fullf$workplace.syndicate,fullf$year)


#___Gelir - Nakdi####
fullf[,36] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ucrn_ay,
                           hbs2010fert$ucrn_ay,
                           hbs2009fert$ucrn_ay,
                           hbs2008fert$ucrn_ay,
                           hbs2007fert$ucrn_ay,
                           hbs2006fert$ucrn_ay,
                           hbs2005fert$ucrn_ay,
                           hbs2004fert$ucrn_ay,
                           hbs2003fert$ucrn_ay,
                           hbs2002fert$ucrn_ay))

colnames(fullf)[36] <- "wage.cash"
fullf[,36] <- numer(fullf[,36])

#Kontrol
length(fullf[,36])==toplamuzunluk



#___Gelir - Nakdı####
fullf[,37] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ucra_ay,
                           hbs2010fert$ucra_ay,
                           hbs2009fert$ucra_ay,
                           hbs2008fert$ucra_ay,
                           hbs2007fert$ucra_ay,
                           hbs2006fert$ucra_ay,
                           hbs2005fert$ucra_ay,
                           hbs2004fert$ucra_ay,
                           hbs2003fert$ucra_ay,
                           hbs2002fert$ucra_ay))

colnames(fullf)[37] <- "wage.supp.monthly"
fullf[,37] <- numer(fullf[,37])

#Kontrol
length(fullf[,37])==toplamuzunluk


#___WAge Yearly####
fullf[,38] <- data.frame(c(hbs2016fert$gelir_nakdi_ucret_yevmiye,
                           hbs2015fert$gelir_nakdi_ucret_yevmiye,
                           hbs2014fert$ucrn_yil,
                           hbs2013fert$ucrn_yil,
                           hbs2012fert$ucrn_yil,
                           hbs2011fert$ucrn_yil,
                           hbs2010fert$ucrn_yil,
                           hbs2009fert$ucrn_yil,
                           hbs2008fert$ucrn_yil,
                           hbs2007fert$ucrn_yil,
```

```
                            hbs2006fert$ucrn_yil,
                            hbs2005fert$ucrn_yil,
                            hbs2004fert$ucrn_yil,
                            hbs2003fert$ucrn_yil,
                            hbs2002fert$ucrn_yil))

colnames(fullf)[38] <- "wage.yearly"
fullf[,38] <- numer(fullf[,38])

#Kontrol
length(fullf[,38])==toplamuzunluk



#___İkramiye Yıllık####
fullf[,39] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$ikr_yil,
                            hbs2010fert$ikr_yil,
                            hbs2009fert$ikr_yil,
                            hbs2008fert$ikr_yil,
                            hbs2007fert$ikr_yil,
                            hbs2006fert$ikr_yil,
                            hbs2005fert$ikr_yil,
                            hbs2004fert$ikr_yil,
                            hbs2003fert$ikr_yil,
                            hbs2002fert$ikr_yil))

colnames(fullf)[39] <- "bonus.yearly"
fullf[,39] <- numer(fullf[,39])

#Kontrol
length(fullf[,39])==toplamuzunluk




#___İkramiye Yıllık####
fullf[,40] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13)),
                            hbs2012fert$ikrprim_yil,
                            hbs2011fert$prim_yil,
                            hbs2010fert$prim_yil,
                            hbs2009fert$prim_yil,
                            hbs2008fert$prim_yil,
                            hbs2007fert$prim_yil,
                            hbs2006fert$prim_yil,
                            hbs2005fert$prim_yil,
                            hbs2004fert$prim_yil,
                            hbs2003fert$prim_yil,
                            hbs2002fert$prim_yil))

colnames(fullf)[40] <- "bonus2.yearly"
fullf[,40] <- numer(fullf[,40])

#Kontrol
length(fullf[,40])==toplamuzunluk




#___Ayni Ücret Yıllıki####
fullf[,41] <- data.frame(c(hbs2016fert$gelir_ayni_ucret_yevmiye,
                            hbs2015fert$gelir_ayni_ucret_yevmiye,
                            hbs2014fert$ucra_yil,
                            hbs2013fert$ucra_yil,
                            hbs2012fert$ucra_yil,
                            hbs2011fert$ucra_yil,
                            hbs2010fert$ucra_yil,
                            hbs2009fert$ucra_yil,
                            hbs2008fert$ucra_yil,
```

```
                            hbs2007fert$ucra_yil,
                            hbs2006fert$ucra_yil,
                            hbs2005fert$ucra_yil,
                            hbs2004fert$ucra_yil,
                            hbs2003fert$ucra_yil,
                            hbs2002fert$ucra_yil))

colnames(fullf)[41] <- "wage.supp.yearly"
fullf[,41] <- numer(fullf[,41])

#Kontrol
length(fullf[,41])==toplamuzunluk



#___Konut Yardımı####
fullf[,42] <- data.frame(c(hbs2016fert$gelir_ayni_konut,
                            hbs2015fert$gelir_ayni_konut,
                            hbs2014fert$konayni,
                            hbs2013fert$konayni,
                            hbs2012fert$konayni,
                            hbs2011fert$konayni,
                            hbs2010fert$konayni,
                            hbs2009fert$konayni,
                            hbs2008fert$konayni,
                            hbs2007fert$konayni,
                            hbs2006fert$konayni,
                            hbs2005fert$konayni,
                            hbs2004fert$konayni,
                            hbs2003fert$konayni,
                            hbs2002fert$konayni))

colnames(fullf)[42] <- "housing.supp"
fullf[,42] <- numer(fullf[,42])

#Kontrol
length(fullf[,42])==toplamuzunluk



#___Ulaşım Yardımı####
fullf[,43] <- data.frame(c(hbs2016fert$gelir_ayni_ulasim,
                            hbs2015fert$gelir_ayni_ulasim,
                            hbs2014fert$ulsayni,
                            hbs2013fert$ulsayni,
                            hbs2012fert$ulsayni,
                            hbs2011fert$ulsayni,
                            hbs2010fert$ulsayni,
                            hbs2009fert$ulsayni,
                            hbs2008fert$ulsayni,
                            hbs2007fert$ulsayni,
                            hbs2006fert$ulsayni,
                            hbs2005fert$ulsayni,
                            hbs2004fert$ulsayni,
                            hbs2003fert$ulsayni,
                            hbs2002fert$ulsayni))

colnames(fullf)[43] <- "transport.supp"
fullf[,43] <- numer(fullf[,43])

#Kontrol
length(fullf[,43])==toplamuzunluk


#___Elektrik Yardımı####
fullf[,44] <- data.frame(c(hbs2016fert$gelir_ayni_elektrik_gaz,
                            hbs2015fert$gelir_ayni_elektrik_gaz,
                            hbs2014fert$egsayni,
                            hbs2013fert$egsayni,
                            hbs2012fert$egsayni,
                            hbs2011fert$egsayni,
                            hbs2010fert$egsayni,
                            hbs2009fert$egsayni,
```

```
                            hbs2008fert$egsayni,
                            hbs2007fert$egsayni,
                            hbs2006fert$egsayni,
                            hbs2005fert$egsayni,
                            hbs2004fert$egsayni,
                            hbs2003fert$egsayni,
                            hbs2002fert$egsayni))

colnames(fullf)[44] <- "electricity.supp"
fullf[,44] <- numer(fullf[,44])

#Kontrol
length(fullf[,44])==toplamuzunluk



#___Yemek Yardımı####
fullf[,45] <- data.frame(c(hbs2016fert$gelir_ayni_yemek,
                            hbs2015fert$gelir_ayni_yemek,
                            hbs2014fert$yemayni,
                            hbs2013fert$yemayni,
                            hbs2012fert$yemayni,
                            hbs2011fert$yemayni,
                            hbs2010fert$yemayni,
                            hbs2009fert$yemayni,
                            hbs2008fert$yemayni,
                            hbs2007fert$yemayni,
                            hbs2006fert$yemayni,
                            hbs2005fert$yemayni,
                            hbs2004fert$yemayni,
                            hbs2003fert$yemayni,
                            hbs2002fert$yemayni))

colnames(fullf)[45] <- "eating.supp"
fullf[,45] <- numer(fullf[,45])

#Kontrol
length(fullf[,45])==toplamuzunluk



#___Kreş Yardımı####
fullf[,46] <- data.frame(c(hbs2016fert$gelir_ayni_kres,
                            hbs2015fert$gelir_ayni_kres,
                            hbs2014fert$kresayni,
                            hbs2013fert$kresayni,
                            hbs2012fert$kresayni,
                            hbs2011fert$kresayni,
                            hbs2010fert$kresayni,
                            hbs2009fert$kresayni,
                            hbs2008fert$kresayni,
                            hbs2007fert$kresayni,
                            hbs2006fert$kresayni,
                            hbs2005fert$kresayni,
                            hbs2004fert$kresayni,
                            hbs2003fert$kresayni,
                            hbs2002fert$kresayni))

colnames(fullf)[46] <- "transport.supp"
fullf[,46] <- numer(fullf[,46])

#Kontrol
length(fullf[,46])==toplamuzunluk



#___Giyim Yardımı####
fullf[,47] <- data.frame(c(hbs2016fert$gelir_ayni_giyecek,
                            hbs2015fert$gelir_ayni_giyecek,
                            hbs2014fert$giyayni,
                            hbs2013fert$giyayni,
                            hbs2012fert$giyayni,
```

```
                            hbs2011fert$giyayni,
                            hbs2010fert$giyayni,
                            hbs2009fert$giyayni,
                            hbs2008fert$giyayni,
                            hbs2007fert$giyayni,
                            hbs2006fert$giyayni,
                            hbs2005fert$giyayni,
                            hbs2004fert$giyayni,
                            hbs2003fert$giyayni,
                            hbs2002fert$giyayni))

colnames(fullf)[47] <- "clothing.supp"
fullf[,47] <- numer(fullf[,47])

#Kontrol
length(fullf[,47])==toplamuzunluk



#___Gıda Yardımı####
fullf[,48] <- data.frame(c(hbs2016fert$gelir_ayni_gida,
                            hbs2015fert$gelir_ayni_gida,
                            hbs2014fert$gidaayni,
                            hbs2013fert$gidaayni,
                            hbs2012fert$gidaayni,
                            hbs2011fert$gidaayni,
                            hbs2010fert$gidaayni,
                            hbs2009fert$gidaayni,
                            hbs2008fert$gidaayni,
                            hbs2007fert$gidaayni,
                            hbs2006fert$gidaayni,
                            hbs2005fert$gidaayni,
                            hbs2004fert$gidaayni,
                            hbs2003fert$gidaayni,
                            hbs2002fert$gidaayni))

colnames(fullf)[48] <- "food.supp"
fullf[,48] <- numer(fullf[,48])

#Kontrol
length(fullf[,48])==toplamuzunluk



#___Diğer Yardımlar####
fullf[,49] <- data.frame(c(hbs2016fert$gelir_ayni_diger,
                            hbs2015fert$gelir_ayni_diger,
                            hbs2014fert$digayni,
                            hbs2013fert$digayni,
                            hbs2012fert$digayni,
                            hbs2011fert$digayni,
                            hbs2010fert$digayni,
                            hbs2009fert$digayni,
                            hbs2008fert$digayni,
                            hbs2007fert$digayni,
                            hbs2006fert$digayni,
                            hbs2005fert$digayni,
                            hbs2004fert$digayni,
                            hbs2003fert$digayni,
                            hbs2002fert$digayni))

colnames(fullf)[49] <- "other.supp"
fullf[,49] <- numer(fullf[,49])

#Kontrol
length(fullf[,49])==toplamuzunluk



#___Girişimcilik####
fullf[,50] <- data.frame(c(hbs2016fert$mutesebbis_calisma,
                            hbs2015fert$mutesebbis_calisma,
```

```
                            hbs2014fert$isfaal,
                            hbs2013fert$isfaal,
                            hbs2012fert$isfaal,
                            hbs2011fert$isfaal,
                            hbs2010fert$isfaal,
                            hbs2009fert$isfaal,
                            hbs2008fert$isfaal,
                            hbs2007fert$isfaal,
                            hbs2006fert$isfaal,
                            hbs2005fert$isfaal,
                            hbs2004fert$isfaal,
                            hbs2003fert$isfaal,
                            hbs2002fert$isfaal))

colnames(fullf)[50] <- "workplace.activity"
fullf[,50] <- numer(fullf[,50])

#Kontrol
length(fullf[,50])==toplamuzunluk



#Düzeltmeler####
fullf$workplace.activity[(fullf$year>=2012 )
                         & (fullf$workplace.activity==11)] <- 1
fullf$workplace.activity[(fullf$year>=2012 )
                         & (fullf$workplace.activity>=2)] <- 2



#Tabulation Kontrol####
table(fullf$workplace.activity,fullf$year)









#___İşyerinin Mülk Durumu####
fullf[,51] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$mulk_dur,
                            hbs2010fert$mulk_dur,
                            hbs2009fert$mulk_dur,
                            hbs2008fert$mulk_dur,
                            hbs2007fert$mulk_dur,
                            hbs2006fert$mulk_dur,
                            hbs2005fert$mulk_dur,
                            hbs2004fert$mulk_dur,
                            hbs2003fert$mulk_dur,
                            hbs2002fert$mulk_dur))

colnames(fullf)[51] <- "workplace.owner"
fullf[,51] <- numer(fullf[,51])

#Kontrol
length(fullf[,51])==toplamuzunluk



#___Girişimcilik geliri####
fullf[,52] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$mutsn_ay,
                            hbs2010fert$mutsn_ay,
                            hbs2009fert$mutsn_ay,
                            hbs2008fert$mutsn_ay,
                            hbs2007fert$mutsn_ay,
                            hbs2006fert$mutsn_ay,
                            hbs2005fert$mutsn_ay,
                            hbs2004fert$mutsn_ay,
                            hbs2003fert$mutsn_ay,
                            hbs2002fert$mutsn_ay))
```

```
colnames(fullf)[52] <- "entrepreneur.cash.income.monthly"
fullf[,52] <- numer(fullf[,52])

#Kontrol
length(fullf[,52])==toplamuzunluk



#___Girişimcilik nakdi geliri####
fullf[,53] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$mutsa_ay,
                           hbs2010fert$mutsa_ay,
                           hbs2009fert$mutsa_ay,
                           hbs2008fert$mutsa_ay,
                           hbs2007fert$mutsa_ay,
                           hbs2006fert$mutsa_ay,
                           hbs2005fert$mutsa_ay,
                           hbs2004fert$mutsa_ay,
                           hbs2003fert$mutsa_ay,
                           hbs2002fert$mutsa_ay))

colnames(fullf)[53] <- "entrepreneur.supp.income.monthly"
fullf[,53] <- numer(fullf[,53])

#Kontrol
length(fullf[,53])==toplamuzunluk



#___Girişimcilik geliri####
fullf[,54] <- data.frame(c(hbs2016fert$gelir_nakdi_mutesebbis,
                           hbs2015fert$gelir_nakdi_mutesebbis,
                           hbs2014fert$mutsn_yl,
                           hbs2013fert$mutsn_yl,
                           hbs2012fert$mutsn_yl,
                           hbs2011fert$mutsn_yl,
                           hbs2010fert$mutsn_yl,
                           hbs2009fert$mutsn_yl,
                           hbs2008fert$mutsn_yl,
                           hbs2007fert$mutsn_yl,
                           hbs2006fert$mutsn_yl,
                           hbs2005fert$mutsn_yl,
                           hbs2004fert$mutsn_yl,
                           hbs2003fert$mutsn_yl,
                           hbs2002fert$mutsn_yl))

colnames(fullf)[54] <- "entrepreneur.cash.income"
fullf[,54] <- numer(fullf[,54])

#Kontrol
length(fullf[,54])==toplamuzunluk



#___Girişimcilik geliri####
fullf[,55] <- data.frame(c(hbs2016fert$gelir_ayni_mutesebbis,
                           hbs2015fert$gelir_ayni_mutesebbis,
                           hbs2014fert$mutsa_yl,
                           hbs2013fert$mutsa_yl,
                           hbs2012fert$mutsa_yl,
                           hbs2011fert$mutsa_yl,
                           hbs2010fert$mutsa_yl,
                           hbs2009fert$mutsa_yl,
                           hbs2008fert$mutsa_yl,
                           hbs2007fert$mutsa_yl,
                           hbs2006fert$mutsa_yl,
                           hbs2005fert$mutsa_yl,
                           hbs2004fert$mutsa_yl,
                           hbs2003fert$mutsa_yl,
                           hbs2002fert$mutsa_yl))
```

```
colnames(fullf)[55] <- "entrepreneur.supp.income"
fullf[,55] <- numer(fullf[,55])

#Kontrol
length(fullf[,55])==toplamuzunluk



#___Tarım geliri####
fullf[,56] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$targ_ay,
                           hbs2010fert$targ_ay,
                           hbs2009fert$targ_ay,
                           hbs2008fert$targ_ay,
                           hbs2007fert$targ_ay,
                           hbs2006fert$targ_ay,
                           hbs2005fert$targ_ay,
                           hbs2004fert$targ_ay,
                           hbs2003fert$targ_ay,
                           hbs2002fert$targ_ay))

colnames(fullf)[56] <- "agricult.income.monthly"
fullf[,56] <- numer(fullf[,56])

#Kontrol
length(fullf[,56])==toplamuzunluk



#___Tarım Yıllık geliri####
fullf[,57] <- data.frame(c(hbs2016fert$gelir_tarim_net,
                           hbs2015fert$gelir_tarim_net,
                           hbs2014fert$targ_yl,
                           hbs2013fert$targ_yl,
                           hbs2012fert$targ_yl,
                           hbs2011fert$targ_yl,
                           hbs2010fert$targ_yl,
                           hbs2009fert$targ_yl,
                           hbs2008fert$targ_yl,
                           hbs2007fert$targ_yl,
                           hbs2006fert$targ_yl,
                           hbs2005fert$targ_yl,
                           hbs2004fert$targ_yl,
                           hbs2003fert$targ_yl,
                           hbs2002fert$targ_yl))

colnames(fullf)[57] <- "agricult.income.yearly"
fullf[,57] <- numer(fullf[,57])

#Kontrol
length(fullf[,57])==toplamuzunluk



#___Yıllık Ek iş####
fullf[,58] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ekis_yil,
                           hbs2010fert$ekis_yil,
                           hbs2009fert$ekis_yil,
                           hbs2008fert$ekis_yil,
                           hbs2007fert$ekis_yil,
                           hbs2006fert$ekis_yil,
                           hbs2005fert$ekis_yil,
                           hbs2004fert$ekis_yil,
                           hbs2003fert$ekis_yil,
                           hbs2002fert$ekis_yil))

colnames(fullf)[58] <- "add.job.yearly"
fullf[,58] <- numer(fullf[,58])

#Kontrol
```

```
length(fullf[,58])==toplamuzunluk




#Tabulation Kontrol####
table(fullf$add.job.yearly,fullf$year)


#___Aylık Ek iş####
fullf[,59] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ekis_ay,
                           hbs2010fert$ekis_ay,
                           hbs2009fert$ekis_ay,
                           hbs2008fert$ekis_ay,
                           hbs2007fert$ekis_ay,
                           hbs2006fert$ekis_ay,
                           hbs2005fert$ekis_ay,
                           hbs2004fert$ekis_ay,
                           hbs2003fert$ekis_ay,
                           hbs2002fert$ekis_ay))

colnames(fullf)[59] <- "job.add.monthly"
fullf[,59] <- numer(fullf[,59])

#Kontrol
length(fullf[,59])==toplamuzunluk




#Tabulation Kontrol####
table(fullf$job.add.monthly,fullf$year)



#___Çalışma süresi####
fullf[,60] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$calsur_e,
                           hbs2010fert$calsur_e,
                           hbs2009fert$calsur_e,
                           hbs2008fert$calsur_e,
                           hbs2007fert$calsur_e,
                           hbs2006fert$calsur_e,
                           hbs2005fert$calsur_e,
                           hbs2004fert$calsur_e,
                           hbs2003fert$calsur_e,
                           hbs2002fert$calsur_e))

colnames(fullf)[60] <- "job.duration"
fullf[,60] <- numer(fullf[,60])

#Kontrol
length(fullf[,60])==toplamuzunluk



#___Sektör Kodu####
fullf[,61] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$sekkod_e,
                           hbs2010fert$sekkod_e,
                           hbs2009fert$sekkod_e,
                           hbs2008fert$sekkod_e,
                           hbs2007fert$sekkod_e,
                           hbs2006fert$sekkod_e,
                           hbs2005fert$sekkod_e,
                           hbs2004fert$sekkod_e,
                           hbs2003fert$sekkod_e,
                           hbs2002fert$sekkod_e))

colnames(fullf)[61] <- "job.sector"
fullf[,61] <- numer(fullf[,61])
```

```
#Kontrol
length(fullf[,61])==toplamuzunluk



#Düzeltmeler, NaceRev2'yi NaceRev1'e indirgiyorum####
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==2)] <- 1
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==3)] <- 2
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==4)] <- 3
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==5)] <- 4
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==6)] <- 5
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==7)] <- 6
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==8)] <- 7
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==9)] <- 8
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==10)] <- 9
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==11)] <- 10
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==12)] <- 11
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==13)] <- 12
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==14)] <- 13
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==15)] <- 14
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==16)] <- 14
fullf$job.sector[(fullf$year<=2003 ) & (fullf$job.sector==17)] <- 14


fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==5)] <- 4
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==6)] <- 5
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==7)] <- 6
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==9)] <- 7
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==10)] <- 8
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==11)] <- 9
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==12)] <- 10
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==14)] <- 11
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==15)] <- 11
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==16)] <- 12
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==17)] <- 13
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==18)] <- 14
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==19)] <- 14
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==20)] <- 14
fullf$job.sector[(fullf$year>=2015) & (fullf$job.sector==21)] <- 14



#2008'de hatalı kodlama var, oranlarına göre yeniden kodluyorum
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==3)] <- 2
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==4)] <- 3
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==5)] <- 4
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==6)] <- 5
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==7)] <- 6
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==8)] <- 7
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==9)] <- 8
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==10)] <- 9
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==11)] <- 10
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==12)] <- 11
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==13)] <- 12
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==14)] <- 13
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==15)] <- 14
fullf$job.sector[(fullf$year==2008) & (fullf$job.sector==16)] <- 14



fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==9)] <- 8
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==10)] <- 9
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==11)] <- 10
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==12)] <- NA
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==13)] <- 11
```

```
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==14)] <- 11
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==15)] <- 12
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==16)] <- 13
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==17)] <- 14
fullf$job.sector[(fullf$year>=2009 & fullf$year<=2014)
                 & (fullf$job.sector==18)] <- 14



#Tabulation Kontrol####
table(fullf$job.sector,fullf$year)



#___Meslek####
fullf[,62] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$meslek_e,
                           hbs2010fert$meslek_e,
                           hbs2009fert$meslek_e,
                           hbs2008fert$meslek_e,
                           hbs2007fert$meslek_e,
                           hbs2006fert$meslek_e,
                           hbs2005fert$meslek_e,
                           hbs2004fert$meslek_e,
                           hbs2003fert$meslek_e,
                           hbs2002fert$meslek_e))

colnames(fullf)[62] <- "job.occupation"
fullf[,62] <- numer(fullf[,62])

#Kontrol
length(fullf[,62])==toplamuzunluk



#Tabulation Kontrol####
table(fullf$job.occupation,fullf$year)


#___İş durumu####
fullf[,63] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$isdur_e,
                           hbs2010fert$isdur_e,
                           hbs2009fert$isdur_e,
                           hbs2008fert$isdur_e,
                           hbs2007fert$isdur_e,
                           hbs2006fert$isdur_e,
                           hbs2005fert$isdur_e,
                           hbs2004fert$isdur_e,
                           hbs2003fert$isdur_e,
                           hbs2002fert$isdur_e))

colnames(fullf)[63] <- "job.status"
fullf[,63] <- numer(fullf[,63])

#Kontrol
length(fullf[,63])==toplamuzunluk



fullf$job.status[(fullf$year<=2004 ) & (fullf$job.status==3)] <- 2
fullf$job.status[(fullf$year<=2004 ) & (fullf$job.status==4)] <- 3
fullf$job.status[(fullf$year<=2004 ) & (fullf$job.status==5)] <- 4
fullf$job.status[(fullf$year<=2004 ) & (fullf$job.status==6)] <- 5

#Tabulation Kontrol####
table(fullf$job.status,fullf$year)
```

```
#___Nakdi Ek gelir####
fullf[,64] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ekgln_ay,
                           hbs2010fert$ekgln_ay,
                           hbs2009fert$ekgln_ay,
                           hbs2008fert$ekgln_ay,
                           hbs2007fert$ekgln_ay,
                           hbs2006fert$ekgln_ay,
                           hbs2005fert$ekgln_ay,
                           hbs2004fert$ekgln_ay,
                           hbs2003fert$ekgln_ay,
                           hbs2002fert$ekgln_ay))

colnames(fullf)[64] <- "income.add.cash.monthly"
fullf[,64] <- numer(fullf[,64])

#Kontrol
length(fullf[,64])==toplamuzunluk



#___Ayni Ek gelir####
fullf[,65] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ekgla_ay,
                           hbs2010fert$ekgla_ay,
                           hbs2009fert$ekgla_ay,
                           hbs2008fert$ekgla_ay,
                           hbs2007fert$ekgla_ay,
                           hbs2006fert$ekgla_ay,
                           hbs2005fert$ekgla_ay,
                           hbs2004fert$ekgla_ay,
                           hbs2003fert$ekgla_ay,
                           hbs2002fert$ekgla_ay))

colnames(fullf)[65] <- "income.add.supp.monthly"
fullf[,65] <- numer(fullf[,65])

#Kontrol
length(fullf[,65])==toplamuzunluk



#___Nakdi Ek gelir####
fullf[,66] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ekgln_yl,
                           hbs2010fert$ekgln_yl,
                           hbs2009fert$ekgln_yl,
                           hbs2008fert$ekgln_yl,
                           hbs2007fert$ekgln_yl,
                           hbs2006fert$ekgln_yl,
                           hbs2005fert$ekgln_yl,
                           hbs2004fert$ekgln_yl,
                           hbs2003fert$ekgln_yl,
                           hbs2002fert$ekgln_yl))

colnames(fullf)[66] <- "income.add.cash.yearly"
fullf[,66] <- numer(fullf[,66])

#Kontrol
length(fullf[,66])==toplamuzunluk



#___Ayni Ek gelir####
fullf[,67] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$ekgla_yl,
                           hbs2010fert$ekgla_yl,
                           hbs2009fert$ekgla_yl,
                           hbs2008fert$ekgla_yl,
                           hbs2007fert$ekgla_yl,
                           hbs2006fert$ekgla_yl,
                           hbs2005fert$ekgla_yl,
```

```
                            hbs2004fert$ekgla_yl,
                            hbs2003fert$ekgla_yl,
                            hbs2002fert$ekgla_yl))

colnames(fullf)[67] <- "income.add.supp.yearly"
fullf[,67] <- numer(fullf[,67])

#Kontrol
length(fullf[,67])==toplamuzunluk



#___Son 12 ay çalışılan süre####
fullf[,68] <- data.frame(c(hbs2016fert$calisma_sure_ay,
                            hbs2015fert$calisma_sure_ay,
                            hbs2014fert$calay_yil,
                            hbs2013fert$calay_yil,
                            rep(NA,h12),
                            hbs2011fert$calay_yil,
                            hbs2010fert$calay_yi,
                            hbs2009fert$calay_yil,
                            hbs2008fert$calay_yil,
                            hbs2007fert$calay_yil,
                            hbs2006fert$calay_yi,
                            hbs2005fert$calay_yi,
                            hbs2004fert$calay_yil,
                            hbs2003fert$calay_yi,
                            hbs2002fert$calay_yil))

colnames(fullf)[68] <- "work.duration.12month"
fullf[,68] <- numer(fullf[,68])

#Kontrol
length(fullf[,68])==toplamuzunluk



#___Çalışmama Nedeni####
fullf[,69] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$cl_neden,
                            hbs2010fert$cl_neden,
                            hbs2009fert$cl_neden,
                            hbs2008fert$cl_neden,
                            hbs2007fert$cl_neden,
                            hbs2006fert$cl_neden,
                            hbs2005fert$cl_neden,
                            hbs2004fert$cl_neden,
                            hbs2003fert$cl_neden,
                            hbs2002fert$cl_neden))

colnames(fullf)[69] <- "notworking.reason"
fullf[,69] <- numer(fullf[,69])

#Kontrol
length(fullf[,69])==toplamuzunluk


#Tabulation Kontrol####
table(fullf$notworking.reason,fullf$year)

#___Aynı iş yapma nedeni####
fullf[,70] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$ayni_is,
                            hbs2010fert$ayni_is,
                            hbs2009fert$ayni_is,
                            hbs2008fert$ayni_is,
                            hbs2007fert$ayni_is,
                            hbs2006fert$ayni_is,
                            hbs2005fert$ayni_is,
                            hbs2004fert$ayni_is,
                            hbs2003fert$ayni_is,
                            hbs2002fert$ayni_is))

colnames(fullf)[70] <- "job.same.reason"
```

```
fullf[,70] <- numer(fullf[,70])

#Kontrol
length(fullf[,70])==toplamuzunluk



#___Çalışma Süresi####
fullf[,71] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$calsur_d,
                           hbs2010fert$calsur_d,
                           hbs2009fert$calsur_d,
                           hbs2008fert$calsur_d,
                           hbs2007fert$calsur_d,
                           hbs2006fert$calsur_d,
                           hbs2005fert$calsur_d,
                           hbs2004fert$calsur_d,
                           hbs2003fert$calsur_d,
                           hbs2002fert$calsur_d))

colnames(fullf)[71] <- "job.prev.workhours"
fullf[,71] <- numer(fullf[,71])

#Kontrol
length(fullf[,71])==toplamuzunluk




#___Sektör Kodu####
fullf[,72] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$sekkod_d,
                           hbs2010fert$sekkod_d,
                           hbs2009fert$sekkod_d,
                           hbs2008fert$sekkod_d,
                           hbs2007fert$sekkod_d,
                           hbs2006fert$sekkod_d,
                           hbs2005fert$sekkod_d,
                           hbs2004fert$sekkod_d,
                           hbs2003fert$sekkod_d,
                           hbs2002fert$sekkod_d))

colnames(fullf)[72] <- "job.prev.nace"
fullf[,72] <- numer(fullf[,72])

#Kontrol
length(fullf[,72])==toplamuzunluk




#Düzeltmeler, NaceRev2'yi NaceRev1'e indirgiyorum####
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==2)] <- 1
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==3)] <- 2
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==4)] <- 3
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==5)] <- 4
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==6)] <- 5
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==7)] <- 6
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==8)] <- 7
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==9)] <- 8
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==10)] <- 9
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==11)] <- 10
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==12)] <- 11
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==13)] <- 12
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==14)] <- 13
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==15)] <- 14
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==16)] <- 14
fullf$job.prev.nace[(fullf$year<=2003 ) & (fullf$job.prev.nace==17)] <- 14


fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==5)] <- 4
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==6)] <- 5
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==7)] <- 6
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==9)] <- 7
```

```
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==10)] <- 8
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==11)] <- 9
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==12)] <- 10
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==14)] <- 11
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==15)] <- 11
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==16)] <- 12
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==17)] <- 13
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==18)] <- 14
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==19)] <- 14
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==20)] <- 14
fullf$job.prev.nace[(fullf$year>=2015) & (fullf$job.prev.nace==21)] <- 14



#2008'de hatalı kodlama var, oranlarına göre yeniden kodluyorum
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==3)] <- 2
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==4)] <- 3
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==5)] <- 4
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==6)] <- 5
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==7)] <- 6
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==8)] <- 7
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==9)] <- 8
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==10)] <- 9
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==11)] <- 10
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==12)] <- 11
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==13)] <- 12
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==14)] <- 13
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==15)] <- 14
fullf$job.prev.nace[(fullf$year==2008) & (fullf$job.prev.nace==16)] <- 14



fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==9)] <- 8
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==10)] <- 9
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==11)] <- 10
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==12)] <- NA
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==13)] <- 11
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==14)] <- 11
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==15)] <- 12
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==16)] <- 13
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==17)] <- 14
fullf$job.prev.nace[(fullf$year>=2009 & fullf$year<=2014)
                    & (fullf$job.prev.nace==18)] <- 14



#Tabulation Kontrol####
table(fullf$job.prev.nace,fullf$year)


#___Meslek kodu####
fullf[,73] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$meslek_d,
                           hbs2010fert$meslek_d,
                           hbs2009fert$meslek_d,
                           hbs2008fert$meslek_d,
                           hbs2007fert$meslek_d,
                           hbs2006fert$meslek_d,
                           hbs2005fert$meslek_d,
                           hbs2004fert$meslek_d,
                           hbs2003fert$meslek_d,
                           hbs2002fert$meslek_d))
```

```
colnames(fullf)[73] <- "job.prev.occupation"
fullf[,73] <- numer(fullf[,73])

#Kontrol
length(fullf[,73])==toplamuzunluk


#Tabulation Kontrol####
table(fullf$job.prev.occupation,fullf$year)


#___Öceki iş durumu####
fullf[,74] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$isdur_d,
                           hbs2010fert$isdur_d,
                           hbs2009fert$isdur_d,
                           hbs2008fert$isdur_d,
                           hbs2007fert$isdur_d,
                           hbs2006fert$isdur_d,
                           hbs2005fert$isdur_d,
                           hbs2004fert$isdur_d,
                           hbs2003fert$isdur_d,
                           hbs2002fert$isdur_d))

colnames(fullf)[74] <- "job.prev.status"
fullf[,74] <- numer(fullf[,74])

#Kontrol
length(fullf[,74])==toplamuzunluk




fullf$job.prev.status[(fullf$year<=2004 ) & (fullf$job.prev.status==3)] <- 2
fullf$job.prev.status[(fullf$year<=2004 ) & (fullf$job.prev.status==4)] <- 3
fullf$job.prev.status[(fullf$year<=2004 ) & (fullf$job.prev.status==5)] <- 4
fullf$job.prev.status[(fullf$year<=2004 ) & (fullf$job.prev.status==6)] <- 5

#Tabulation Kontrol####
table(fullf$job.status,fullf$year)



#___Önceki iş nakdi geliri####
fullf[,75] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$digln_yl,
                           hbs2010fert$digln_yl,
                           hbs2009fert$digln_yl,
                           hbs2008fert$digln_yl,
                           hbs2007fert$digln_yl,
                           hbs2006fert$digln_yl,
                           hbs2005fert$digln_yl,
                           hbs2004fert$digln_yl,
                           hbs2003fert$digln_yl,
                           hbs2002fert$digln_yl))

colnames(fullf)[75] <- "job.prev.cash.income"
fullf[,75] <- numer(fullf[,75])

#Kontrol
length(fullf[,75])==toplamuzunluk


#___Önceki iş ayni geliri####
fullf[,76] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$digla_yl,
                           hbs2010fert$digla_yl,
                           hbs2009fert$digla_yl,
                           hbs2008fert$digla_yl,
                           hbs2007fert$digla_yl,
                           hbs2006fert$digla_yl,
```

```
                            hbs2005fert$digla_yl,
                            hbs2004fert$digla_yl,
                            hbs2003fert$digla_yl,
                            hbs2002fert$digla_yl))

colnames(fullf)[76] <- "job.prev.supp.income"
fullf[,76] <- numer(fullf[,76])

#Kontrol
length(fullf[,76])==toplamuzunluk



#___Aylık gayrimenkul nakdi kira geliri####
fullf[,77] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$gmkn_ay,
                            hbs2010fert$gmkn_ay,
                            hbs2009fert$gmkn_ay,
                            hbs2008fert$gmkn_ay,
                            hbs2007fert$gmkn_ay,
                            hbs2006fert$gmkn_ay,
                            hbs2005fert$gmkn_ay,
                            hbs2004fert$gmkn_ay,
                            hbs2003fert$gmkn_ay,
                            hbs2002fert$gmkn_ay))

colnames(fullf)[77] <- "estate.cash.income.monthly"
fullf[,77] <- numer(fullf[,77])

#Kontrol
length(fullf[,77])==toplamuzunluk



#___Aylık gayrimenkul ayni kira geliri####
fullf[,78] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$gmka_ay,
                            hbs2010fert$gmka_ay,
                            hbs2009fert$gmka_ay,
                            hbs2008fert$gmka_ay,
                            hbs2007fert$gmka_ay,
                            hbs2006fert$gmka_ay,
                            hbs2005fert$gmka_ay,
                            hbs2004fert$gmka_ay,
                            hbs2003fert$gmka_ay,
                            hbs2002fert$gmka_ay))

colnames(fullf)[78] <- "estate.supp.income.monthly"
fullf[,78] <- numer(fullf[,78])

#Kontrol
length(fullf[,78])==toplamuzunluk



#___Yıllık Nakdi Gayrımenkul Geliri####
fullf[,79] <- data.frame(c(hbs2016fert$gelir_nakdi_gayrimenkul,
                            hbs2015fert$gelir_nakdi_gayrimenkul,
                            hbs2014fert$gmkn_yl,
                            hbs2013fert$gmkn_yl,
                            hbs2012fert$gmkn_yl,
                            hbs2011fert$gmkn_yl,
                            hbs2010fert$gmkn_yl,
                            hbs2009fert$gmkn_yl,
                            hbs2008fert$gmkn_yl,
                            hbs2007fert$gmkn_yl,
                            hbs2006fert$gmkn_yl,
                            hbs2005fert$gmkn_yl,
                            hbs2004fert$gmkn_yl,
                            hbs2003fert$gmkn_yl,
                            hbs2002fert$gmkn_yl))

colnames(fullf)[79] <- "estate.cash.income.yearly"
```

```
fullf[,79] <- numer(fullf[,79])

#Kontrol
length(fullf[,79])==toplamuzunluk



#___Yıllık Ayni Gayrımenkul Geliri####
fullf[,80] <- data.frame(c(hbs2016fert$gelir_nakdi_gayrimenkul,
                           hbs2015fert$gelir_nakdi_gayrimenkul,
                           hbs2014fert$gmka_yl,
                           hbs2013fert$gmka_yl,
                           hbs2012fert$gmka_yl,
                           hbs2011fert$gmka_yl,
                           hbs2010fert$gmka_yl,
                           hbs2009fert$gmka_yl,
                           hbs2008fert$gmka_yl,
                           hbs2007fert$gmka_yl,
                           hbs2006fert$gmka_yl,
                           hbs2005fert$gmka_yl,
                           hbs2004fert$gmka_yl,
                           hbs2003fert$gmka_yl,
                           hbs2002fert$gmka_yl))

colnames(fullf)[80] <- "estate.supp.income.yearly"
fullf[,80] <- numer(fullf[,80])

#Kontrol
length(fullf[,80])==toplamuzunluk


#___Aylık menkul ayni kira geliri####
fullf[,81] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$mulkn_ay,
                           hbs2010fert$mulkn_ay,
                           hbs2009fert$mulkn_ay,
                           hbs2008fert$mulkn_ay,
                           hbs2007fert$mulkn_ay,
                           hbs2006fert$mulkn_ay,
                           hbs2005fert$mulkn_ay,
                           hbs2004fert$mulkn_ay,
                           hbs2003fert$mulkn_ay,
                           hbs2002fert$mulkn_ay))

colnames(fullf)[81] <- "property.cash.income.monthly"
fullf[,81] <- numer(fullf[,81])

#Kontrol
length(fullf[,81])==toplamuzunluk


#___Aylık menkul ayni kira geliri####
fullf[,82] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$mulka_ay,
                           hbs2010fert$mulka_ay,
                           hbs2009fert$mulka_ay,
                           hbs2008fert$mulka_ay,
                           hbs2007fert$mulka_ay,
                           hbs2006fert$mulka_ay,
                           hbs2005fert$mulka_ay,
                           hbs2004fert$mulka_ay,
                           hbs2003fert$mulka_ay,
                           hbs2002fert$mulka_ay))

colnames(fullf)[82] <- "property.supp.income.monthly"
fullf[,82] <- numer(fullf[,82])

#Kontrol
length(fullf[,82])==toplamuzunluk



#___Yıllık Banka Geliri####
fullf[,83] <- data.frame(c(hbs2016fert$gelir_banka_tahvil,
```

```
                            hbs2015fert$gelir_banka_tahvil,
                            hbs2014fert$banka_yl,
                            hbs2013fert$banka_yl,
                            hbs2012fert$banka_yl,
                            hbs2011fert$banka_yl,
                            hbs2010fert$banka_yl,
                            hbs2009fert$banka_yl,
                            hbs2008fert$banka_yl,
                            hbs2007fert$banka_yl,
                            hbs2006fert$banka_yl,
                            hbs2005fert$banka_yl,
                            hbs2004fert$banka_yl,
                            hbs2003fert$banka_yl,
                            hbs2002fert$banka_yl))

colnames(fullf)[83] <- "bank.income"
fullf[,83] <- numer(fullf[,83])

#Kontrol
length(fullf[,83])==toplamuzunluk




#___Yıllık döviz geliri####
fullf[,84] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$doviz_yl,
                            hbs2010fert$doviz_yl,
                            hbs2009fert$doviz_yl,
                            hbs2008fert$doviz_yl,
                            hbs2007fert$doviz_yl,
                            hbs2006fert$doviz_yl,
                            hbs2005fert$doviz_yl,
                            hbs2004fert$doviz_yl,
                            hbs2003fert$doviz_yl,
                            hbs2002fert$doviz_yl))

colnames(fullf)[84] <- "exchange.income"
fullf[,84] <- numer(fullf[,84])

#Kontrol
length(fullf[,84])==toplamuzunluk




#___Yıllık nema geliri####
fullf[,85] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$nema_yl,
                            hbs2010fert$nema_yl,
                            hbs2009fert$nema_yl,
                            hbs2008fert$nema_yl,
                            hbs2007fert$nema_yl,
                            hbs2006fert$nema_yl,
                            hbs2005fert$nema_yl,
                            hbs2004fert$nema_yl,
                            hbs2003fert$nema_yl,
                            hbs2002fert$nema_yl))

colnames(fullf)[85] <- "interest.income"
fullf[,85] <- numer(fullf[,85])

#Kontrol
length(fullf[,85])==toplamuzunluk



#___Kar payı geliri####
fullf[,86] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$temet_yl,
                            hbs2010fert$temet_yl,
                            hbs2009fert$temet_yl,
                            hbs2008fert$temet_yl,
```

```
                            hbs2007fert$temet_yl,
                            hbs2006fert$temet_yl,
                            hbs2005fert$temet_yl,
                            hbs2004fert$temet_yl,
                            hbs2003fert$temet_yl,
                            hbs2002fert$temet_yl))

colnames(fullf)[86] <- "dividend.income"
fullf[,86] <- numer(fullf[,86])

#Kontrol
length(fullf[,86])==toplamuzunluk



#___Kar geliri####
fullf[,87] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$kar_yl,
                            hbs2010fert$kar_yl,
                            hbs2009fert$kar_yl,
                            hbs2008fert$kar_yl,
                            hbs2007fert$kar_yl,
                            hbs2006fert$kar_yl,
                            hbs2005fert$kar_yl,
                            hbs2004fert$kar_yl,
                            hbs2003fert$kar_yl,
                            hbs2002fert$kar_yl))

colnames(fullf)[87] <- "profit.income"
fullf[,87] <- numer(fullf[,87])

#Kontrol
length(fullf[,87])==toplamuzunluk



#___Toprak Nakdi geliri####
fullf[,88] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$toprn_yl,
                            hbs2010fert$toprn_yl,
                            hbs2009fert$toprn_yl,
                            hbs2008fert$toprn_yl,
                            hbs2007fert$toprn_yl,
                            hbs2006fert$toprn_yl,
                            hbs2005fert$toprn_yl,
                            hbs2004fert$toprn_yl,
                            hbs2003fert$toprn_yl,
                            hbs2002fert$toprn_yl))

colnames(fullf)[88] <- "land.cash.income"
fullf[,88] <- numer(fullf[,88])

#Kontrol
length(fullf[,88])==toplamuzunluk



#___Toprak Ayni geliri####
fullf[,89] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                            hbs2011fert$topra_yl,
                            hbs2010fert$topra_yl,
                            hbs2009fert$topra_yl,
                            hbs2008fert$topra_yl,
                            hbs2007fert$topra_yl,
                            hbs2006fert$topra_yl,
                            hbs2005fert$topra_yl,
                            hbs2004fert$topra_yl,
                            hbs2003fert$topra_yl,
                            hbs2002fert$topra_yl))

colnames(fullf)[89] <- "land.supp.income"
fullf[,89] <- numer(fullf[,89])

#Kontrol
```

```
length(fullf[,89])==toplamuzunluk



#___Toprak Nakdi geliri####
fullf[,90] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$trsn_ay,
                           hbs2010fert$trsn_ay,
                           hbs2009fert$trsn_ay,
                           hbs2008fert$trsn_ay,
                           hbs2007fert$trsn_ay,
                           hbs2006fert$trsn_ay,
                           hbs2005fert$trsn_ay,
                           hbs2004fert$trsn_ay,
                           hbs2003fert$trsn_ay,
                           hbs2002fert$trsn_ay))

colnames(fullf)[90] <- "land.cash.income"
fullf[,90] <- numer(fullf[,90])



#___Toprak Ayni geliri####
fullf[,91] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                           hbs2011fert$trsa_ay,
                           hbs2010fert$trsa_ay,
                           hbs2009fert$trsa_ay,
                           hbs2008fert$trsa_ay,
                           hbs2007fert$trsa_ay,
                           hbs2006fert$trsa_ay,
                           hbs2005fert$trsa_ay,
                           hbs2004fert$trsa_ay,
                           hbs2003fert$trsa_ay,
                           hbs2002fert$trsa_ay))

colnames(fullf)[91] <- "land.cash.income"
fullf[,91] <- numer(fullf[,91])


#___Yıllık Emekli Geliri####
fullf[,92] <- data.frame(c(hbs2016fert$gelir_devlet_emekli_maas,
                           hbs2015fert$gelir_devlet_emekli_maas,
                           hbs2014fert$emekl_yl,
                           hbs2013fert$emekl_yl,
                           hbs2012fert$emekl_yl,
                           hbs2011fert$emekl_yl,
                           hbs2010fert$emekl_yl,
                           hbs2009fert$emekl_yl,
                           hbs2008fert$emekl_yl,
                           hbs2007fert$emekl_yl,
                           hbs2006fert$emekl_yl,
                           hbs2005fert$emekl_yl,
                           hbs2004fert$emekl_yl,
                           hbs2003fert$emekl_yl,
                           hbs2002fert$emekl_yl))

colnames(fullf)[92] <- "pension.income"
fullf[,92] <- numer(fullf[,92])

#Kontrol
length(fullf[,92])==toplamuzunluk



#___Vergi İadesi####
fullf[,93] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12,h11,h10)),
                           hbs2009fert$viade_yl,
                           hbs2008fert$viade_yl,
                           hbs2007fert$viade_yl,
                           hbs2006fert$viade_yl,
                           hbs2005fert$viade_yl,
                           hbs2004fert$viade_yl,
                           hbs2003fert$viade_yl,
                           hbs2002fert$viade_yl))
```

```
colnames(fullf)[93] <- "tax.return"
fullf[,93] <- numer(fullf[,93])


#___Yıllık Yaşlılık Geliri####
fullf[,94] <- data.frame(c(hbs2016fert$gelir_devlet_yaslilik_maas,
                           hbs2015fert$gelir_devlet_yaslilik_maas,
                           hbs2014fert$yasli_yl,
                           hbs2013fert$yasli_yl,
                           hbs2012fert$yasli_yl,
                           hbs2011fert$yasli_yl,
                           hbs2010fert$yasli_yl,
                           hbs2009fert$yasli_yl,
                           hbs2008fert$yasli_yl,
                           hbs2007fert$yasli_yl,
                           hbs2006fert$yasli_yl,
                           hbs2005fert$yasli_yl,
                           hbs2004fert$yasli_yl,
                           hbs2003fert$yasli_yl,
                           hbs2002fert$yasli_yl))

colnames(fullf)[94] <- "old.income"
fullf[,94] <- numer(fullf[,94])

#Kontrol
length(fullf[,94])==toplamuzunluk



#___Yıllık Yardım Geliri####
fullf[,95] <- data.frame(c(hbs2016fert$gelir_devlet_aile_konut,
                           hbs2015fert$gelir_devlet_aile_konut,
                           hbs2014fert$sosy_yl,
                           hbs2013fert$sosy_yl,
                           hbs2012fert$sosy_yl,
                           hbs2011fert$sosy_yl,
                           hbs2010fert$sosy_yl,
                           hbs2009fert$sosy_yl,
                           hbs2008fert$sosy_yl,
                           hbs2007fert$sosy_yl,
                           hbs2006fert$sosy_yl,
                           hbs2005fert$sosy_yl,
                           hbs2004fert$sosy_yl,
                           rep(NA,h3),
                           hbs2002fert$issiz_yl))

colnames(fullf)[95] <- "old.income"
fullf[,95] <- numer(fullf[,95])

#Kontrol
length(fullf[,95])==toplamuzunluk



#___Hastalık Yardımı Geliri####
fullf[,96] <- data.frame(c(hbs2016fert$gelir_devlet_hastalik,
                           hbs2015fert$gelir_devlet_hastalik,
                           hbs2014fert$hasta_yl,
                           hbs2013fert$hasta_yl,
                           hbs2012fert$hasta_yl,
                           rep(NA,sum(h11,h10,h9,h8,h7,h6,h5,h4,h3,h2))))

colnames(fullf)[96] <- "patient.income"
fullf[,96] <- numer(fullf[,96])

#Kontrol
length(fullf[,96])==toplamuzunluk
```

```
#___Dul  Geliri####
fullf[,97] <- data.frame(c(hbs2016fert$gelir_devlet_dul_yetim,
                           hbs2015fert$gelir_devlet_dul_yetim,
                           hbs2014fert$dul_yl,
                           hbs2013fert$dul_yl,
                           hbs2012fert$dul_yl,
                           hbs2011fert$dul_yl,
                           hbs2010fert$dul_yl,
                           hbs2009fert$dul_yl,
                           hbs2008fert$dul_yl,
                           hbs2007fert$dul_yl,
                           hbs2006fert$dul_yl,
                           hbs2005fert$dul_yl,
                           hbs2004fert$dul_yl,
                           hbs2003fert$dul_yl,
                           hbs2002fert$dul_yl))

colnames(fullf)[97] <- "widow.income"
fullf[,97] <- numer(fullf[,97])

#Kontrol
length(fullf[,97])==toplamuzunluk



#___Gazi  Geliri####
fullf[,98] <- data.frame(c(hbs2016fert$gelir_devlet_gazilik,
                           hbs2015fert$gelir_devlet_gazilik,
                           hbs2014fert$gazi_yl,
                           hbs2013fert$gazi_yl,
                           hbs2012fert$gazi_yl,
                           hbs2011fert$gazi_yl,
                           hbs2010fert$gazi_yl,
                           hbs2009fert$gazi_yl,
                           hbs2008fert$gazi_yl,
                           hbs2007fert$gazi_yl,
                           hbs2006fert$gazi_yl,
                           hbs2005fert$gazi_yl,
                           hbs2004fert$gazi_yl,
                           hbs2003fert$gazi_yl,
                           hbs2002fert$gazi_yl))

colnames(fullf)[98] <- "veteran.income"
fullf[,98] <- numer(fullf[,98])

#Kontrol
length(fullf[,98])==toplamuzunluk


#___Burs  Geliri####
fullf[,99] <- data.frame(c(hbs2016fert$gelir_devlet_burs,
                           hbs2015fert$gelir_devlet_burs,
                           hbs2014fert$burs_yl,
                           hbs2013fert$burs_yl,
                           hbs2012fert$burs_yl,
                           hbs2011fert$burs_yl,
                           hbs2010fert$burs_yl,
                           hbs2009fert$burs_yl,
                           hbs2008fert$burs_yl,
                           hbs2007fert$burs_yl,
                           hbs2006fert$burs_yl,
                           hbs2005fert$burs_yl,
                           hbs2004fert$burs_yl,
                           hbs2003fert$burs_yl,
                           hbs2002fert$burs_yl))

colnames(fullf)[99] <- "bursary.income"
fullf[,99] <- numer(fullf[,99])

#Kontrol
length(fullf[,99])==toplamuzunluk
```

```
#___İşsizlik  Geliri####
fullf[,100] <- data.frame(c(hbs2016fert$gelir_devlet_issizlik,
                            hbs2015fert$gelir_devlet_issizlik,
                            hbs2014fert$issiz_yl,
                            hbs2013fert$issiz_yl,
                            hbs2012fert$issiz_yl,
                            hbs2011fert$issiz_yl,
                            hbs2010fert$issiz_yl,
                            hbs2009fert$issiz_yl,
                            hbs2008fert$issiz_yl,
                            hbs2007fert$issiz_yl,
                            hbs2006fert$issiz_yl,
                            hbs2005fert$issiz_yl,
                            hbs2004fert$issiz_yl,
                            hbs2003fert$issiz_yl,
                            hbs2002fert$issiz_yl))

colnames(fullf)[100] <- "unemp.income"
fullf[,100] <- numer(fullf[,100])

#Kontrol
length(fullf[,100])==toplamuzunluk



#___Tarımsal Destek  Geliri####
fullf[,101] <- data.frame(c(hbs2016fert$gelir_devlet_tarimsal_destek,
                            hbs2015fert$gelir_devlet_tarimsal_destek,
                            hbs2014fert$ddestek_yl,
                            hbs2013fert$ddestek_yl,
                            hbs2012fert$ddestek_yl,
                            hbs2011fert$ddestek_yl,
                            hbs2010fert$ddestek,
                            hbs2009fert$ddestek_yl,
                            hbs2008fert$ddestek_yl,
                            hbs2007fert$ddestek_yl,
                            hbs2006fert$ddestek,
                            hbs2005fert$ddestek_,
                            rep(NA,sum(h4,h3,h2))))

colnames(fullf)[101] <- "agricult.income"
fullf[,101] <- numer(fullf[,101])

#Kontrol
length(fullf[,101])==toplamuzunluk



#___Ayni Devlet Desteği  Geliri####
fullf[,102] <- data.frame(c(hbs2016fert$gelir_ayni_devlet,
                            hbs2015fert$gelir_ayni_devlet,
                            hbs2014fert$dayni_yl,
                            hbs2013fert$dayni_yl,
                            hbs2012fert$dayni_yl,
                            hbs2011fert$dayni_yl,
                            hbs2010fert$dayni_yl,
                            hbs2009fert$dayni_yl,
                            hbs2008fert$dayni_yl,
                            hbs2007fert$dayni_yl,
                            hbs2006fert$dayni_yl,
                            hbs2005fert$dayni_yl,
                            hbs2004fert$dayni_yl,
                            hbs2003fert$dayni_yl,
                            hbs2002fert$dayni_yl))

colnames(fullf)[102] <- "supp.income"
fullf[,102] <- numer(fullf[,102])

#Kontrol
length(fullf[,102])==toplamuzunluk
```

```
#___Yurtdışı Emekli Maaşı Geliri####
fullf[,103] <- data.frame(c(hbs2016fert$gelir_ydisi_emekli_maas,
                            hbs2015fert$gelir_ydisi_emekli_maas,
                            hbs2014fert$ydemk_yl,
                            hbs2013fert$ydemk_yl,
                            hbs2012fert$ydemk_yl,
                            hbs2011fert$ydemk_yl,
                            hbs2010fert$ydemk_yl,
                            hbs2009fert$ydemk_yl,
                            hbs2008fert$ydemk_yl,
                            hbs2007fert$ydemk_yl,
                            hbs2006fert$ydemk_yl,
                            hbs2005fert$ydemk_yl,
                            hbs2004fert$ydemk_yl,
                            hbs2003fert$ydemk_yl,
                            hbs2002fert$ydemk_yl))

colnames(fullf)[103] <- "veteran.foreign.income"
fullf[,103] <- numer(fullf[,103])

#Kontrol
length(fullf[,103])==toplamuzunluk



#___Yurtdışı dövizli Maaşı Geliri####
fullf[,104] <- data.frame(c(hbs2016fert$gelir_ydisi_doviz_burs,
                            hbs2015fert$gelir_ydisi_doviz_burs,
                            hbs2014fert$yddov_yl,
                            hbs2013fert$yddov_yl,
                            hbs2012fert$yddov_yl,
                            hbs2011fert$yddov_yl,
                            hbs2010fert$yddov_yl,
                            hbs2009fert$yddov_yl,
                            hbs2008fert$yddov_yl,
                            hbs2007fert$yddov_yl,
                            hbs2006fert$yddov_yl,
                            hbs2005fert$yddov_yl,
                            hbs2004fert$yddov_yl,
                            hbs2003fert$yddov_yl,
                            hbs2002fert$yddov_yl))

colnames(fullf)[104] <- "bursary.foreign.income"
fullf[,104] <- numer(fullf[,104])

#Kontrol
length(fullf[,104])==toplamuzunluk



#___Yurtdışı Ayni Geliri####
fullf[,105] <- data.frame(c(hbs2016fert$gelir_ayni_ydisi,
                            hbs2015fert$gelir_ayni_ydisi,
                            hbs2014fert$ydayn_yl,
                            hbs2013fert$ydayn_yl,
                            hbs2012fert$ydayn_yl,
                            hbs2011fert$ydayn_yl,
                            hbs2010fert$ydayn_yl,
                            hbs2009fert$ydayn_yl,
                            hbs2008fert$ydayni_yl,
                            hbs2007fert$ydayn_yl,
                            hbs2006fert$ydayn_yl,
                            hbs2005fert$ydayn_yl,
                            hbs2004fert$ydayn_yl,
                            hbs2003fert$ydayn_yl,
                            hbs2002fert$ydayn_yl))

colnames(fullf)[105] <- "supp.foreign.income"
fullf[,105] <- numer(fullf[,105])

#Kontrol
```

```
length(fullf[,105])==toplamuzunluk



#___Yurtdışı Özel Geliri####
fullf[,106] <- data.frame(c(hbs2016fert$gelir_nakdi_ozel,
                            hbs2015fert$gelir_nakdi_ozel,
                            hbs2014fert$dgnfk_yl,
                            hbs2013fert$dgnfk_yl,
                            hbs2012fert$dgnfk_yl,
                            hbs2011fert$dgnfk_yl,
                            hbs2010fert$dgnfk_yl,
                            hbs2009fert$dgnfk_yl,
                            hbs2008fert$dgnfk_yl,
                            hbs2007fert$dgnfk_yl,
                            hbs2006fert$dgnfk_yl,
                            hbs2005fert$dgnfk_yl,
                            hbs2004fert$dgnfk_yl,
                            hbs2003fert$dgnfk_yl,
                            hbs2002fert$dgnfk_yl))

colnames(fullf)[106] <- "private.foreign.income"
fullf[,106] <- numer(fullf[,106])

#Kontrol
length(fullf[,106])==toplamuzunluk




#___Devlet dışı burs ve yardımlar Maaşı Geliri####
fullf[,107] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12,h11,h10,h9,h8,h7)),
                            hbs2006fert$dgbrs_yl,
                            hbs2005fert$dgbrs_yl,
                            hbs2004fert$dgbrs_yl,
                            hbs2003fert$dgbrs_yl,
                            hbs2002fert$dgbrs_yl))

colnames(fullf)[107] <- "supp.other.income"
fullf[,107] <- numer(fullf[,107])

#Kontrol
length(fullf[,107])==toplamuzunluk




#___Diğer Türden Özel Emeklilik Gelirleri####
fullf[,108] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13)),
                            hbs2012fert$dsemk_yl,
                            hbs2011fert$dsemk_yl,
                            hbs2010fert$dsemk_yl,
                            hbs2009fert$dsemk_yl,
                            hbs2008fert$dsemk_yl,
                            hbs2007fert$dsemk_yl,
                            hbs2006fert$dsemk_yl,
                            hbs2005fert$dsemk_yl,
                            hbs2004fert$dsemk_yl,
                            rep(NA,sum(h3,h2))))

colnames(fullf)[108] <- "veteran.other.income"
fullf[,108] <- numer(fullf[,108])

#Kontrol
length(fullf[,108])==toplamuzunluk



#___Diğer Özel Teşebbüs Geliri####
fullf[,109] <- data.frame(c(hbs2016fert$gelir_ayni_ozel,
                            hbs2015fert$gelir_ayni_ozel,
```

```
                              hbs2014fert$dgayn_yl,
                              hbs2013fert$dgayn_yl,
                              hbs2012fert$dgayn_yl,
                              hbs2011fert$dgayn_yl,
                              hbs2010fert$dgayn_yl,
                              hbs2009fert$dgayn_yl,
                              hbs2008fert$dgayn_yl,
                              hbs2007fert$dgayn_yl,
                              hbs2006fert$dgayn_yl,
                              hbs2005fert$dgayn_yl,
                              hbs2004fert$dgayn_yl,
                              hbs2003fert$dgayn_yl,
                              hbs2002fert$dgayn_yl))

colnames(fullf)[109] <- "supp.private.income"
fullf[,109] <- numer(fullf[,109])

#Kontrol
length(fullf[,109])==toplamuzunluk



#___Toplam Gelir Nakdi Aylık####
fullf[,110] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                              hbs2011fert$topgn_ay,
                              hbs2010fert$topgn_ay,
                              hbs2009fert$topgn_ay,
                              hbs2008fert$topgn_ay,
                              hbs2007fert$topgn_ay,
                              hbs2006fert$topgn_ay,
                              hbs2005fert$topgn_ay,
                              hbs2004fert$topgn_ay,
                              hbs2003fert$topgn_ay,
                              hbs2002fert$topgn_ay))

colnames(fullf)[110] <- "total.cash.income.monthly"
fullf[,110] <- numer(fullf[,110])

#Kontrol
length(fullf[,110])==toplamuzunluk



#___Toplam Gelir Ayni Aylık####
fullf[,111] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                              hbs2011fert$topga_ay,
                              hbs2010fert$topga_ay,
                              hbs2009fert$topga_ay,
                              hbs2008fert$topga_ay,
                              hbs2007fert$topga_ay,
                              hbs2006fert$topga_ay,
                              hbs2005fert$topga_ay,
                              hbs2004fert$topga_ay,
                              hbs2003fert$topga_ay,
                              hbs2002fert$topga_ay))

colnames(fullf)[111] <- "total.supp.income.monthly"
fullf[,111] <- numer(fullf[,111])

#Kontrol
length(fullf[,111])==toplamuzunluk



#___Toplam Gelir Nakdi ve Ayni Aylık####
fullf[,112] <- data.frame(c(rep(NA,sum(h16,h15,h14,h13,h12)),
                              hbs2011fert$topgelay,
                              hbs2010fert$topgelay,
                              hbs2009fert$topgelay,
                              hbs2008fert$topgelay,
                              hbs2007fert$topgelay,
                              hbs2006fert$topgelay,
```

```
                              hbs2005fert$topgelay,
                              hbs2004fert$topgelay,
                              hbs2003fert$topgelay,
                              hbs2002fert$topgelay))

colnames(fullf)[112] <- "total.income.monthly"
fullf[,112] <- numer(fullf[,112])

#Kontrol
length(fullf[,112])==toplamuzunluk



#___Toplam Gelir Nakdi Yıllık####
fullf[,113] <- data.frame(c(hbs2016fert$gelir_nakdi_toplam,
                              hbs2015fert$gelir_nakdi_toplam,
                              hbs2014fert$topgn_yl,
                              hbs2013fert$topgn_yl,
                              hbs2012fert$topgn_yl,
                              hbs2011fert$topgn_yl,
                              hbs2010fert$topgn_yl,
                              hbs2009fert$topgn_yl,
                              hbs2008fert$topgn_yl,
                              hbs2007fert$topgn_yl,
                              hbs2006fert$topgn_yl,
                              hbs2005fert$topgn_yl,
                              hbs2004fert$topgn_yl,
                              hbs2003fert$topgn_yl,
                              hbs2002fert$topgn_yl))

colnames(fullf)[113] <- "total.cash.income"
fullf[,113] <- numer(fullf[,113])

#Kontrol
length(fullf[,113])==toplamuzunluk



#___Toplam Gelir Ayni Yıllık####
fullf[,114] <- data.frame(c(hbs2016fert$gelir_ayni_toplam,
                              hbs2015fert$gelir_ayni_toplam,
                              hbs2014fert$topga_yl,
                              hbs2013fert$topga_yl,
                              hbs2012fert$topga_yl,
                              hbs2011fert$topga_yl,
                              hbs2010fert$topga_yl,
                              hbs2009fert$topga_yl,
                              hbs2008fert$topga_yl,
                              hbs2007fert$topga_yl,
                              hbs2006fert$topga_yl,
                              hbs2005fert$topga_yl,
                              hbs2004fert$topga_yl,
                              hbs2003fert$topga_yl,
                              hbs2002fert$topga_yl))

colnames(fullf)[114] <- "total.supp.income"
fullf[,114] <- numer(fullf[,114])



#___Toplam Gelir Yıllık####
fullf[,115] <- data.frame(c(hbs2016fert$gelir_toplam,
                              hbs2015fert$gelir_toplam,
                              hbs2014fert$topgelyl,
                              hbs2013fert$topgelyl,
                              hbs2012fert$topgelyl,
                              hbs2011fert$topgelyl,
                              hbs2010fert$topgelyl,
                              hbs2009fert$topgelyl,
                              hbs2008fert$topgelyl,
                              hbs2007fert$topgelyl,
                              hbs2006fert$topgelyl,
                              hbs2005fert$topgelyl,
```

```
                              hbs2004fert$topgelyl,
                              hbs2003fert$topgelyl,
                              hbs2002fert$topgelyl))

colnames(fullf)[115] <- "total.income"
fullf[,115] <- numer(fullf[,115])

#Kontrol
length(fullf[,115])==toplamuzunluk



#___Endeks Değerik####
fullf[,116] <- data.frame(c(hbs2016fert$indeks_arl,
                              hbs2015fert$indeks_arl,
                              hbs2014fert$indeks_arl,
                              hbs2013fert$indeks_arl,
                              hbs2012fert$indeks_arl,
                              hbs2011fert$indeks_arl,
                              hbs2010fert$indeks_a,
                              hbs2009fert$indeks_arl,
                              hbs2008fert$indeks_arl,
                              hbs2007fert$indeks_arl,
                              hbs2006fert$indeks_a,
                              hbs2005fert$indeks_a,
                              hbs2004fert$indeks_arl,
                              hbs2003fert$indeks,
                              hbs2002fert$indeks))

colnames(fullf)[116] <- "index"
fullf[,116] <- numer(fullf[,116])



cols <- openxlsx::read.xlsx(
  xlsxFile="/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/HBS-DataAtlas.xlsx", sheet = 2, colName
colnames(fullf) <- cols$VARIABLE

save(fullf,file=paste0(prevpath,'FullFert.Rdata'))



#Düzeltmeler ve Uyumlaştırmalar####
load(file=paste0(prevpath,'FullFert.Rdata'))
```

3.3 R Codes Used to Merge HL Part of HBSs

```
## -*- coding: utf-8 -*-
#"""
#Created on Wed Dec 20 23:11:04 2017

#######################################################################
#                                                                     #
#     _    _    __ _   ___  (_)  _ __    | | __ _    _ | |_  _    _ | | __     #
#    | |  | |  / _  |/ __|| || '_ \  | |/ /| |  | || __|| |  | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    <  | |_| || |_ | |_| ||    <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                                 _  _                       #
#    / __ \    __ _   _ __  ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_    _ \   / _  || || |    / __|/ _ \  | '_    _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                    #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web    : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
#"""
#

# Yasin Kütük
# HBS HANE Soru ve yanıtları uyumlaştırma sistemi
print("HBS␣Soru␣ve␣yanıtları␣uyumlaştırma␣sistemi")
# R Kodları




####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,"systemfit",
              "foreign", "dplyr", "sampleSelection",
              "systemfit", "openxlsx")
paketkurulum(paketler)

####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm␣kütüphaneler␣yüklendi.")
}
loadlib(x)



####03.Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle

path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/RData/"
prevpath <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"


veriler <- list.files(path=path)


for(i in veriler){
  load(paste0(path,i))
```

```
  assign(substr(i,1,nchar(i)-6),sil)
  rm(sil)
}

####Integer ve Numeric Dönüşümü####
numer <- function(x){as.numeric(as.character(x))}
integ <- function(x){as.integer(as.character(x))}

#2012 ve 2014 HH YILKULGEL'de "," kullanılmış decimal olarak.
hbs2012hane$yilkulgel <- gsub("," ,".",hbs2012hane$yilkulgel)
hbs2014hane$yilkulgel <- gsub("," ,".",hbs2014hane$yilkulgel)
hbs2012hane$yilkulgel <- numer(hbs2012hane$yilkulgel)
hbs2014hane$yilkulgel <- numer(hbs2014hane$yilkulgel)

#2012 ve 2014 HH HARCAMA'da "," kullanılmış decimal olarak.
hbs2012hane$harcama <- gsub("," ,".",hbs2012hane$harcama)
hbs2014hane$harcama <- gsub("," ,".",hbs2014hane$harcama)
hbs2012hane$harcama <- numer(hbs2012hane$harcama)
hbs2014hane$harcama <- numer(hbs2014hane$harcama)


#Kontrol Parametresi####
hbsh <- ls()[grep('hane',ls(),fixed=T)]
vector <- list()
for(j in hbsh){
  dim <- dim(get(j))[1]
  print(dim)
  vector[match(j ,hbsh)] <- as.integer(dim)
}


toplamuzunluk <- sum(as.data.frame(vector))


#___UZUNLAYICILAR####
h16 <- dim(hbs2016hane)[1]
h15 <- dim(hbs2015hane)[1]
h14 <- dim(hbs2014hane)[1]
h13 <- dim(hbs2013hane)[1]
h12 <- dim(hbs2012hane)[1]
h11 <- dim(hbs2011hane)[1]
h10 <- dim(hbs2010hane)[1]
h9 <-  dim(hbs2009hane)[1]
h8 <-  dim(hbs2008hane)[1]
h7 <-  dim(hbs2007hane)[1]
h6 <-  dim(hbs2006hane)[1]
h5 <-  dim(hbs2005hane)[1]
h4 <-  dim(hbs2004hane)[1]
h3 <-  dim(hbs2003hane)[1]
h2 <-  dim(hbs2002hane)[1]
```

```
vars <- openxlsx::read.xlsx(paste0(prevpath,'Butce_Hane_Match.xlsx'),
                            sheet='Sheet5')
vars <- vars[vars$Year==1,]
vars <- data.frame(sapply(vars,tolower))
colnames(vars) <- gsub('X','',colnames(vars))
vars <- vars[,-1]

yatay <- dim(vars)[2]
dikey <- dim(vars)[1]
years <- 2002:2016

for(i in 1:2){
  for(j in 5:7){
    print(paste0(as.character(i),as.character(j)))
    #Dolayısıyla önce içerideki loop'u döndürüyor
  }
}
```

```
#
#for(i in 1:15){
#  row <- dim(get(paste0('hbs',years[i],'hane')))[1]
#  col <- dim(get(paste0('hbs',years[i],'hane')))[2]
#  narep <- rep(NA,row)
#  var <- paste0('hbs',years[i],'hane[,',col+1,']')
#  assign(var,rep(NA,row))
## k <- rep(NA,dim(eval(parse(text=paste0('hbs',years[j],'hane'))))[1])
## get(parse(text=paste0('hbs',years[j],'hane')))
#[,dim(eval(parse(text=paste0('hbs',years[j],'hane'))))[2]+1] <- k
#}
#


hbs2002hane[,dim(hbs2002hane)[2]+1] <- rep(NA,dim(hbs2002hane)[1])
hbs2003hane[,dim(hbs2003hane)[2]+1] <- rep(NA,dim(hbs2003hane)[1])
hbs2004hane[,dim(hbs2004hane)[2]+1] <- rep(NA,dim(hbs2004hane)[1])
hbs2005hane[,dim(hbs2005hane)[2]+1] <- rep(NA,dim(hbs2005hane)[1])
hbs2006hane[,dim(hbs2006hane)[2]+1] <- rep(NA,dim(hbs2006hane)[1])
hbs2007hane[,dim(hbs2007hane)[2]+1] <- rep(NA,dim(hbs2007hane)[1])
hbs2008hane[,dim(hbs2008hane)[2]+1] <- rep(NA,dim(hbs2008hane)[1])
hbs2009hane[,dim(hbs2009hane)[2]+1] <- rep(NA,dim(hbs2009hane)[1])
hbs2010hane[,dim(hbs2010hane)[2]+1] <- rep(NA,dim(hbs2010hane)[1])
hbs2011hane[,dim(hbs2011hane)[2]+1] <- rep(NA,dim(hbs2011hane)[1])
hbs2012hane[,dim(hbs2012hane)[2]+1] <- rep(NA,dim(hbs2012hane)[1])
hbs2013hane[,dim(hbs2013hane)[2]+1] <- rep(NA,dim(hbs2013hane)[1])
hbs2014hane[,dim(hbs2014hane)[2]+1] <- rep(NA,dim(hbs2014hane)[1])
hbs2015hane[,dim(hbs2015hane)[2]+1] <- rep(NA,dim(hbs2015hane)[1])
hbs2016hane[,dim(hbs2016hane)[2]+1] <- rep(NA,dim(hbs2016hane)[1])
colnames(hbs2002hane)[dim(hbs2002hane)[2]] <- c('narep')
colnames(hbs2003hane)[dim(hbs2003hane)[2]] <- c('narep')
colnames(hbs2004hane)[dim(hbs2004hane)[2]] <- c('narep')
colnames(hbs2005hane)[dim(hbs2005hane)[2]] <- c('narep')
colnames(hbs2006hane)[dim(hbs2006hane)[2]] <- c('narep')
colnames(hbs2007hane)[dim(hbs2007hane)[2]] <- c('narep')
colnames(hbs2008hane)[dim(hbs2008hane)[2]] <- c('narep')
colnames(hbs2009hane)[dim(hbs2009hane)[2]] <- c('narep')
colnames(hbs2010hane)[dim(hbs2010hane)[2]] <- c('narep')
colnames(hbs2011hane)[dim(hbs2011hane)[2]] <- c('narep')
colnames(hbs2012hane)[dim(hbs2012hane)[2]] <- c('narep')
colnames(hbs2013hane)[dim(hbs2013hane)[2]] <- c('narep')
colnames(hbs2014hane)[dim(hbs2014hane)[2]] <- c('narep')
colnames(hbs2015hane)[dim(hbs2015hane)[2]] <- c('narep')
colnames(hbs2016hane)[dim(hbs2016hane)[2]] <- c('narep')


fullvec <- vector()

for(i in 1:dikey){
  for(j in 1:yatay){
    veri <- paste0(vars[i,j])
    print(paste0(years[j],'␣yılının␣', veri,'␣değişkeni!'))
    if(veri!=999){
      set <- as.vector(eval(parse(text=paste0('hbs',years[j],'hane$',veri))))
      fullvec <-c(fullvec, set)
    }
    else{
      na.rep <- rep(NA,dim(as.vector(eval(
        parse(text=paste0('hbs',years[j],'hane')))))[1])
      fullvec <-c(fullvec, na.rep)
    }
  }
}

length(fullvec)==toplamuzunluk*dim(vars)[1]



#save(fullvec,file=paste0(prevpath,'FullVec.Rdata'))
#
#load(paste0(prevpath,'FullVec.Rdata'))
#
#fullset <- data.frame(fullvec[1:toplamuzunluk] )
```

```
#fullset[,2] <- fullvec[(1*toplamuzunluk)+1 : (2*toplamuzunluk)+1]
#
#
#for(i in 2:5){
#print(i)
#fullset[,i] <- data.frame(fullvec[i*toplamuzunluk:(i+1)*toplamuzunluk])
#}
#
```

```
##___Bulten No####
#fullh[,dim(fullh)[2]+1] <- data.frame(c(hbs2016hane$birimno,
#                           hbs2015hane$birimno,
#                           hbs2014hane$bulten,
#                           hbs2013hane$bulten,
#                           hbs2012hane$bulten,
#                           hbs2011hane$bulten,
#                           hbs2010hane$bulten,
#                           hbs2009hane$bulten,
#                           hbs2008hane$bulten,
#                           hbs2007hane$bulten,
#                           hbs2006hane$bulten,
#                           hbs2005hane$bulten,
#                           hbs2004hane$bulten,
#                           hbs2003hane$bulten,
#                           hbs2002hane$bulten))

#colnames(fullh)[dim(fullh)[2]] <- "sheet"
#fullh[,dim(fullh)[2]] <- integ(fullh[,dim(fullh)[2]])
```

```
fullh <- data.frame(matrix(data=fullvec,
                           nrow = toplamuzunluk,ncol=dikey,byrow = F))
#Çünkü dikeydeki kadar veri olacak
```

```
#___Yıllar####
fullh[,dim(fullh)[2]+1] <- data.frame(c(rep(2002,length(hbs2002hane$bulten)),
                                        rep(2003,length(hbs2003hane$bulten)),
                                        rep(2004,length(hbs2004hane$bulten)),
                                        rep(2005,length(hbs2005hane$bulten)),
                                        rep(2006,length(hbs2006hane$bulten)),
                                        rep(2007,length(hbs2007hane$bulten)),
                                        rep(2008,length(hbs2008hane$bulten)),
                                        rep(2009,length(hbs2009hane$bulten)),
                                        rep(2010,length(hbs2010hane$bulten)),
                                        rep(2011,length(hbs2011hane$bulten)),
                                        rep(2012,length(hbs2012hane$bulten)),
                                        rep(2013,length(hbs2013hane$bulten)),
                                        rep(2014,length(hbs2014hane$bulten)),
                                        rep(2015,length(hbs2015hane$birimno)),
                                        rep(2016,length(hbs2016hane$birimno))))

colnames(fullh)[dim(fullh)[2]] <- "yil"
fullh[,dim(fullh)[2]] <- integ(fullh[,dim(fullh)[2]])

#Kontrol
length(fullh[,dim(fullh)[2]])==toplamuzunluk
```

```
colnames(fullh) <- tolower(
  data.frame((openxlsx::read.xlsx(paste0(prevpath,'hbshanevars.xlsx')))[1])[,1])

fullh <- data.frame(lapply(fullh,numer))
```

```
#___HH Tipi####
#Düzeltmeler####
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==2] <- 1
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==3] <- 2
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==4] <- 2
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==5] <- 3
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==6] <- 3
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==7] <- 4
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==8] <- 5
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==9] <- 5
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==10] <- 5
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==11] <- 6
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==12] <- 6
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==13] <- 6
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==14] <- 7
fullh$hhtipi[(fullh$yil<=2007) & fullh$hhtipi==15] <- 7


#Tabulation Control####
table(fullh$hhtipi,fullh$yil)



#___Konuttipi####
#Düzeltmeler####
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==2)] <- 1
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==9)] <- 1
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==3)] <- 2
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==4)] <- 3
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==5)] <- 3
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==6)] <- 3
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==7)] <- 3
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==8)] <- 3
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==9)] <- 3
fullh$konuttip[(fullh$yil<=2007)  & (fullh$konuttip==10)] <- 4


fullh$konuttip[(fullh$yil==2008 | fullh$yil==2009) & fullh$konuttip==4] <- 3
fullh$konuttip[(fullh$yil==2008 | fullh$yil==2009) & fullh$konuttip==5] <- 3
fullh$konuttip[(fullh$yil==2008 | fullh$yil==2009) & fullh$konuttip==6] <- 4

fullh$konuttip[(fullh$yil>=2010 & fullh$yil<=2015) & fullh$konuttip==4] <- 3
fullh$konuttip[(fullh$yil>=2010 & fullh$yil<=2015) & fullh$konuttip==5] <- 3
fullh$konuttip[(fullh$yil>=2010 & fullh$yil<=2015) & fullh$konuttip==6] <- 3
fullh$konuttip[(fullh$yil>=2010 & fullh$yil<=2015) & fullh$konuttip==7] <- 4
fullh$konuttip[(fullh$yil>=2010 & fullh$yil<=2015) & fullh$konuttip==98] <- 4


fullh$konuttip[(fullh$yil>=2016) & fullh$konuttip==4] <- 3
fullh$konuttip[(fullh$yil>=2016) & fullh$konuttip==98] <- 4


#Tabulation Control####
table(fullh$konuttip,fullh$yil)



#___Bina Tarihi####
#Düzeltmeler####
fullh$bina_tar[(fullh$yil==2002)
               & (fullh$bina_tar>=10 & fullh$bina_tar<=1945)] <- 1
fullh$bina_tar[(fullh$yil==2002)
               & (fullh$bina_tar>=1946 & fullh$bina_tar<=1960)] <- 2
fullh$bina_tar[(fullh$yil==2002)
```

```
              & (fullh$bina_tar>=1961 & fullh$bina_tar<=1980)] <- 3
fullh$bina_tar[(fullh$yil==2002)
              & (fullh$bina_tar>=1981 & fullh$bina_tar<=1995)] <- 4
fullh$bina_tar[(fullh$yil==2002)
              & (fullh$bina_tar>=1996 & fullh$bina_tar<=2017)] <- 5


fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)  & (fullh$bina_tar==4)] <- 3
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)  & (fullh$bina_tar==5)] <- 4
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)  & (fullh$bina_tar==6)] <- 4
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)  & (fullh$bina_tar==7)] <- 5
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)  & (fullh$bina_tar==8)] <- 5


fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)
              & (fullh$bina_tar>=10 & fullh$bina_tar<=1945)] <- 1
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)
              & (fullh$bina_tar>=1946 & fullh$bina_tar<=1960)] <- 2
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)
              & (fullh$bina_tar>=1961 & fullh$bina_tar<=1980)] <- 3
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)
              & (fullh$bina_tar>=1981 & fullh$bina_tar<=1995)] <- 4
fullh$bina_tar[(fullh$yil>=2003 & fullh$yil<=2011)
              & (fullh$bina_tar>=1996 & fullh$bina_tar<=2017)] <- 5


fullh$bina_tar[(fullh$yil>=2012)  & (fullh$bina_tar==4)] <- 3
fullh$bina_tar[(fullh$yil>=2012)  & (fullh$bina_tar==5)] <- 4
fullh$bina_tar[(fullh$yil>=2012)  & (fullh$bina_tar==6)] <- 4
fullh$bina_tar[(fullh$yil>=2012)  & (fullh$bina_tar==7)] <- 5
fullh$bina_tar[(fullh$yil>=2012)  & (fullh$bina_tar==8)] <- 5

#Tabulation Kontrol#####
table(fullh$bina_tar,fullh$yil)



#___Isıtma Sistemi####
#Düzeltmeler####
fullh$isit_sis[(fullh$yil>=2010 & fullh$yil<=2014)
              & (fullh$isit_sis==4)] <- NA
#Klima sistemini kaldırıyorum, uyum için
fullh$isit_sis[(fullh$yil>=2010 & fullh$yil<=2014)
              & (fullh$isit_sis==5)] <- 4

fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==1)] <- 11
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==2)] <- 22
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==3)] <- 33
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==4)] <- 44
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==5)] <- 55
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==6)] <- 4
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==7)] <- 4
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==90)] <- 4
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==98)] <- 4

fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==11)] <- 2
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==22)] <- 2
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==33)] <- 3
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==44)] <- 1
fullh$isit_sis[(fullh$yil>=2015) & (fullh$isit_sis==55)] <- 1


#Tabulation Kontrol#####
table(fullh$isit_sis,fullh$yil)
```

```
#___Yakıt 1, 2 ve 3####
#Düzeltmeler####
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==4)] <- 6
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==5)] <- 6
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==6)] <- 6
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==7)] <- 6
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==10)] <- 6
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==8)] <- 4
fullh$yakit_1[(fullh$yil<=2011) & (fullh$yakit_1==9)] <- 5


fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==2)] <- 6
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==4)] <- 6
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==6)] <- 6
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==7)] <- 6
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==8)] <- 6
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==98)] <- 6
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==11)] <- 1
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==12)] <- 2
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==5)] <-  4
fullh$yakit_1[(fullh$yil>=2015) & (fullh$yakit_1==13)] <- 5


#Tabulation Kontrol#####
table(fullh$yakit_1,fullh$yil)


#Düzeltmeler####
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==4)] <- 6
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==5)] <- 6
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==6)] <- 6
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==7)] <- 6
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==10)] <- 6
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==8)] <- 4
fullh$yakit_2[(fullh$yil<=2011) & (fullh$yakit_2==9)] <- 5


fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==2)] <- 6
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==4)] <- 6
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==6)] <- 6
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==7)] <- 6
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==8)] <- 6
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==98)] <- 6
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==11)] <- 1
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==12)] <- 2
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==5)] <-  4
fullh$yakit_2[(fullh$yil>=2015) & (fullh$yakit_2==13)] <- 5


#Tabulation Kontrol#####
table(fullh$yakit_2,fullh$yil)



#Düzeltmeler####
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==4)] <- 6
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==5)] <- 6
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==6)] <- 6
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==7)] <- 6
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==10)] <- 6
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==8)] <- 4
fullh$yakit_3[(fullh$yil<=2011) & (fullh$yakit_3==9)] <- 5


fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==2)] <- 6
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==4)] <- 6
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==6)] <- 6
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==7)] <- 6
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==8)] <- 6
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==98)] <- 6
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==11)] <- 1
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==12)] <- 2
```

```
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==5)] <-  4
fullh$yakit_3[(fullh$yil>=2015) & (fullh$yakit_3==13)] <- 5


#Tabulation Kontrol#####
table(fullh$yakit_3,fullh$yil)



#___Yakıt MUT ve SU####
#Düzeltmeler####
fullh$yakit_mut[(fullh$yil==2009) &
                  (fullh$yakit_mut==4 | fullh$yakit_mut==5
                   | fullh$yakit_mut==6)] <- 88
fullh$yakit_mut[(fullh$yil==2009) & (fullh$yakit_mut==7)] <- 4
fullh$yakit_mut[(fullh$yil==2009) & (fullh$yakit_mut==8)] <- 5
fullh$yakit_mut[(fullh$yil==2009) & (fullh$yakit_mut==9)] <- 6
fullh$yakit_mut[(fullh$yil==2009) & (fullh$yakit_mut==10)] <- 7
fullh$yakit_mut[(fullh$yil==2009) & (fullh$yakit_mut==88)] <- 8

fullh$yakit_mut[(fullh$yil>=2010 &  fullh$yil<=2011)
                & (fullh$yakit_mut==4 | fullh$yakit_mut==5
                   | fullh$yakit_mut==6 | fullh$yakit_mut==11)] <- 88
fullh$yakit_mut[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_mut==7)] <- 4
fullh$yakit_mut[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_mut==8)] <- 5
fullh$yakit_mut[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_mut==9)] <- 6
fullh$yakit_mut[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_mut==10)] <- 7
fullh$yakit_mut[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_mut==88)] <- 8

fullh$yakit_mut[(fullh$yil>=2015 ) & (fullh$yakit_mut==2)] <- 8
fullh$yakit_mut[(fullh$yil>=2015 ) & (fullh$yakit_mut==11)] <- 1
fullh$yakit_mut[(fullh$yil>=2015 ) & (fullh$yakit_mut==12)] <- 2
fullh$yakit_mut[(fullh$yil>=2015 ) & (fullh$yakit_mut==13)] <- 5
fullh$yakit_mut[(fullh$yil>=2015 ) & (fullh$yakit_mut==98)] <- 8

#Tabulation Kontrol#####
table(fullh$yakit_mut,fullh$yil)


#Düzeltmeler####
fullh$yakit_su[(fullh$yil==2009) &
                 (fullh$yakit_su==4 | fullh$yakit_su==5 | fullh$yakit_su==6)] <- 88
fullh$yakit_su[(fullh$yil==2009) & (fullh$yakit_su==7)] <- 4
fullh$yakit_su[(fullh$yil==2009) & (fullh$yakit_su==8)] <- 5
fullh$yakit_su[(fullh$yil==2009) & (fullh$yakit_su==9)] <- 6
fullh$yakit_su[(fullh$yil==2009) & (fullh$yakit_su==10)] <- 7
fullh$yakit_su[(fullh$yil==2009) & (fullh$yakit_su==88)] <- 8

fullh$yakit_su[(fullh$yil>=2010 &  fullh$yil<=2011)
               & (fullh$yakit_su==4 | fullh$yakit_su==5
                  | fullh$yakit_su==6 | fullh$yakit_su==11)] <- 88
fullh$yakit_su[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_su==7)] <- 4
fullh$yakit_su[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_su==8)] <- 5
fullh$yakit_su[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_su==9)] <- 6
fullh$yakit_su[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_su==10)] <- 7
fullh$yakit_su[(fullh$yil>=2010 &  fullh$yil<=2011) & (fullh$yakit_su==88)] <- 8

fullh$yakit_su[(fullh$yil>=2015 ) & (fullh$yakit_su==2)] <- 8
fullh$yakit_su[(fullh$yil>=2015 ) & (fullh$yakit_su==11)] <- 1
fullh$yakit_su[(fullh$yil>=2015 ) & (fullh$yakit_su==12)] <- 2
fullh$yakit_su[(fullh$yil>=2015 ) & (fullh$yakit_su==13)] <- 5
fullh$yakit_su[(fullh$yil>=2015 ) & (fullh$yakit_su==98)] <- 8

#Tabulation Kontrol#####
table(fullh$yakit_su,fullh$yil)



#___Anket şıkları düzeltmeleri####
anket <- c('zor_alis', 'zor_banka', 'zor_posta', 'zor_ulas', 'zor_saglik', 'zor_egiti')
anket <- which(colnames(fullh) %in% anket )
```

```
for(i in anket){

  fullh[,i][fullh$yil>=2015 & fullh[,i]==5] <- 11
  fullh[,i][fullh$yil>=2015 & fullh[,i]==4] <- 22
  fullh[,i][fullh$yil>=2015 & fullh[,i]==3] <- NA
  fullh[,i][fullh$yil>=2015 & fullh[,i]==2] <- 33
  fullh[,i][fullh$yil>=2015 & fullh[,i]==1] <- 44

  fullh[,i][fullh$yil>=2015 & fullh[,i]==44] <- 4
  fullh[,i][fullh$yil>=2015 & fullh[,i]==33] <- 3
  fullh[,i][fullh$yil>=2015 & fullh[,i]==22] <- 2
  fullh[,i][fullh$yil>=2015 & fullh[,i]==11] <- 1

  #Tabulation Control
  print(table(fullh[,i],fullh$yil))
}
```

```
#___Dichotomous Vars düzeltmeleri,2=H olan 0 olacak####
anket <- c('banyo','sauna','jakuzi','tuvalet','mutfak','copogut','kalorif'
           ,'tabanisi','su_sis','dog_gaz','sicaksu','kabloyay','asansor','garaj'
           ,'havuz','jenrator','apartman','yazlik','tarla','bag','sera'
           ,'arsa','dukkan','otel')
anket <- which(colnames(fullh) %in% anket )

for(i in anket){

  fullh[,i][fullh$yil>=2015 & fullh[,i]==2] <- 0


  #Tabulation Control
  print(table(fullh[,i],fullh$yil))
}
```

```
sink(file=paste0(prevpath,'Tabulations.txt'))

for(i in 27:dim(fullh)[2]){
  print(colnames(fullh[i]))
  print(table(fullh[,i],fullh$yil))
  print('_____')
  print('_____')
}

sink()
```

```
save(fullh,file=paste0(prevpath,'FullHane.Rdata'))
```

```
load(file=paste0(prevpath,'FullHane.Rdata'))
```

```
##Veri var mı yok mu
#hbs2016hane$
#hbs2015hane$
#hbs2014hane$
#hbs2013hane$
#hbs2012hane$
#hbs2011hane$
#hbs2010hane$
#hbs2009hane$
#hbs2008hane$
#hbs2007hane$
#hbs2006hane$
#hbs2005hane$
#hbs2004hane$
#hbs2003hane$
#hbs2002hane$


#1            bulten
#2            hhtipi
#3          konuttip
#4          mulkiyet
#5           kira_ay
#6          kira_sek
#7          loj_fert
#8           izafi_k
#9          konutsah
#10         kon_sure
#11         kon_borc
#12         kondeger
#13         otur_sur
#14         bina_tar
#15         yapituru
#16          oda_say
#17         kon_alan
#18        zemin_sal
#19        zemin_oda
#20        zemin_mut
#21        zemin_ban
#22         isit_sis
#23          yakit_1
#24          yakit_2
#25          yakit_3
#26        yakit_mut
#27         yakit_su
#28         zor_alis
#29        zor_banka
#30        zor_posta
#31         zor_ulas
#32       zor_saglik
#33        zor_egiti
#34        oda_say.1
#35       kon_alan.1
#36         konutiki
#37         konut2_n
#38            banyo
#39            sauna
#40           jakuzi
#41          tuvalet
#42           mutfak
#43          copogut
#44          kalorif
#45        tabanisi
#46            sudepo
#47            su_sis
#48          elektri
#49          dog_gaz
#50          sicaksu
#51         kabloyay
#52          asansor
#53            garaj
#54            havuz
#55         jenrator
```

```
#56          telefon
#57           ceptel
#58         bilgisay
#59           intern
#60          televiz
#61            video
#62          lcdtele
#63          dvd_vcd
#64           kamera
#65          uyduant
#66         buzdolab
#67         derindon
#68          bulasik
#69          m_firin
#70          camasir
#71          kurutma
#72         haliyika
#73            klima
#74         otomobil
#75          otoadet
#76         isveroto
#77         isotoadt
#78          motosik
#79         motoadet
#80         deniztas
#81         dentadet
#82         muskonut
#83           musmik
#84          muskira
#85          muskdeg
#86         apartman
#87           aptmik
#88          aptkira
#89          aptkdeg
#90           yazlik
#91           yazmik
#92          yazkira
#93          yazkdeg
#94            tarla
#95           tarmik
#96          tarkira
#97          tarkdeg
#98              bag
#99           bagmik
#100         bagkira
#101         bagkdeg
#102            sera
#103         seramik
#104        serakira
#105        serakdeg
#106            arsa
#107         arsamik
#108        arsakira
#109        arsakdeg
#110         arsadeg
#111          dukkan
#112          dukmik
#113         dukkira
#114         dukkdeg
#115          dukdeg
#116            otel
#117         otelmik
#118        otelkira
#119        otelkdeg
#120         oteldeg
#121           diger
#122          digmik
#123         digkira
#124         digkdeg
#125          digdeg
#126       yilkulgel
#127         efb_oecd
#128         harcama
#129             hhb
```

```
#130        kirkntkd
#131          faktor
```

3.4 R Codes Used to Merge IL with HL

```
## -*- coding: utf-8 -*-
#"""
#Created on Wed Dec 20 23:11:04 2017

#######################################################################
#                          _                 _                _       #
#     _   _    __ _   ___ (_) _ __    | | __ _   _ | |_  _   _  _ | | __    #
#    | | | |  / _  |/ __|| || '_ \  | |/ /| | | || __|| | | || |/ /    #
#    | |_| || (_| |\__ \| || | | ||    < | |_| || |_ | |_| ||   <     #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\    #
#     |___/                                                           #
#     ____                                 _  _                       #
#    / __ \    __ _   _ __ ___     __ _ (_)| |     ___   ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_   _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                  #
#######################################################################
#@author: Yasin KÜTÜK          ######################################
#@web   : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
#"""
#

# Yasin Kütük
# HBS Soru ve yanıtları uyumlaştırma sistemi
print("HBS birleştirme")
# R Kodları




####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table"
,"systemfit",  "foreign", "dplyr",
 "sampleSelection","systemfit", "openxlsx") #2000-03 arasındaki STATA dosyalarını okumak için
paketkurulum(paketler)

####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
}
loadlib(x)



####03.Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle

path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/RData/"
prevpath <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"



load(paste0(prevpath,'FullFert.Rdata'))
load(paste0(prevpath,'FullHane.Rdata'))
```

```
# Dot (.) ve <NA> Replacements
nareplacer <- function(x){
  lapply(x, ifelse(as.character(x)=='.',NA,x))
}



nareplacer <- function(x){
  x[x==.] <- NA
}


numer <- function(x){as.numeric(as.character(x))}
integ <- function(x){as.integer(as.character(x))}









full <- merge(fullf,fullh, by = 'bulten'  , all = T)

full$yil <- NULL



##Varsayimlarim (Bunlar mutlaka reason edilecek)

full <- subset(full,full$yilkulgel>0 | !is.na(full$yilkulgel) )
#1 Yillik kullanilabilir gelir bilgisi NA olanlar atilacak
#ve negatif kullanilabilir geliri olanlar çikarilacak
dim(full)
full <- subset(full,full$yilkulgel>(full$harcama*12))
#Harcamasi yillik kullanilabilir gelirden fazla olanlari da çikariyorum
dim(full)
full$tasarruf <- (full$yilkulgel-(full$harcama*12))
head(full$tasarruf)
full$tasarruf_orani <- (full$tasarruf/full$yilkulgel)
head(full$tasarruf_orani)




#Deflator,base(2009)####
deflator <- openxlsx::read.xlsx(
xlsxFile="/media/DRIVE/Dropbox/_VeriTabanlarim/Deflator_WDI.xlsx",
sheet = 2, colNames=T)

full$deflator[full$year==2000] <- deflator$deflator[1]
full$deflator[full$year==2001] <- deflator$deflator[2]
full$deflator[full$year==2002] <- deflator$deflator[3]
full$deflator[full$year==2003] <- deflator$deflator[4]
full$deflator[full$year==2004] <- deflator$deflator[5]
full$deflator[full$year==2005] <- deflator$deflator[6]
full$deflator[full$year==2006] <- deflator$deflator[7]
full$deflator[full$year==2007] <- deflator$deflator[8]
full$deflator[full$year==2008] <- deflator$deflator[9]
full$deflator[full$year==2009] <- deflator$deflator[10]
full$deflator[full$year==2010] <- deflator$deflator[11]
full$deflator[full$year==2011] <- deflator$deflator[12]
full$deflator[full$year==2012] <- deflator$deflator[13]
full$deflator[full$year==2013] <- deflator$deflator[14]
full$deflator[full$year==2014] <- deflator$deflator[15]
full$deflator[full$year==2015] <- deflator$deflator[16]
full$deflator[full$year==2016] <- deflator$deflator[17]
```

```
save(full,file=paste0(prevpath,'FullHBS_base.Rdata'))
```

3.5 R Codes for Descriptive Statistics

```
## -*- coding: utf-8 -*-
#"""
#Created on Wed Feb 28 18:47:52 2018

#######################################################################
#                          _                 _                _       #
#     _    _    __ _    ___  (_)  _ __    | |  __ _    _ |_   _    _ | |  __     #
#    | |  | |  / _  |/ __|| || '_ \  | |/ /| |  | || __|| |  | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    <  | |_| || |_ | |_| ||    <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                          _  _                              #
#    / __ \    __ _  _ __ ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \  | '_   _ \   #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
#"""
#

# Yasin Kütük
# HLFS Soru ve yanıtları uyumlaştırma sistemi
print("HBS Statistical Computations")
# R Kodları


#Initials#####
options(digits = 4)
options(scipen=999)
options(max.print = 500)

if(.Platform$OS.type=="windows"){
  path="d://Dropbox//_Dissertation//Yazim//Paper1_ML//Analyses//"
  respath="d://Dropbox//_Dissertation//Yazim//Paper1_ML//Analyses//Results//"
  print("Hocam Windows'dasın!")
} else {
  path="/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper2_DL/Analyses/"
  respath="/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper2_DL/Analyses/Results/"
  print("Abi Linux bu!")
}


####01.Gerekli Olan Package Kurulumları####

paketkurulum <- function(paket){
  paketkur <- paket[!(paket %in% installed.packages()[, "Package"])]
  if (length(paketkur))
    install.packages(paketkur, dependencies = TRUE)
  sapply(paket, require, character.only = TRUE)
}
paketler <- c("xlsx","data.table" ,
  "systemfit",  "foreign", "Hmisc",
  "dplyr", "sampleSelection",
  "systemfit","psych",
  "openxlsx", "stargazer", "reporttools",
   "tab" , "xtable", "stargazer", "doBy",
    "quantreg", "tidyverse") #2000-03 arasındaki STATA dosyalarını okumak için
paketkurulum(paketler)

####02.Gerekli Kütüphanelerin Yüklenmesi#####
loadlib=function(x){
  library(foreign)
  library(car)
  library(systemfit)
  print("Tüm kütüphaneler yüklendi.")
```

```
}
loadlib(x)

####03.Verilerin Rdata Dönüşümleri ve Yüklenmesi####
rm(list=ls()) #Tüm environment'ı temizle


path <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/RData/"
prevpath <- "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"
respath="/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper2_DL/Analyses/Results/"

#Load the Data####
load(file=paste0(prevpath,'FullHBS_base.Rdata'))




#Nominal var deflator and 6 digit remover
noms <- c('topgna_yl' ,'ucrn_ay','ucra_ay','ucrn_yil',
  'ikr_yil','prim_yil','ucra_yil','konayni','ulsayni',
  'egsayni','yemayni','kresayni','giyayni','gidaayni','digayni','mutsn_ay',
          'mutsa_ay','mutsn_yl','mutsa_yl','targ_ay',
          'targ_yl','ekgln_ay','ekgla_ay','ekgln_yl',
          'ekgla_yl','digln_yl','digla_yl','gmkn_ay',
          'gmka_ay','gmkn_yl','gmka_yl',
          'mulkn_ay','mulka_ay','banka_yl','doviz_yl',
          'nema_yl','temet_yl','kar_yl','toprn_yl',
          'topra_yl','trsn_ay','trsa_ay','emekl_yl',
          'viade_yl','yasli_yl','sosy_yl',
          'hasta_yl','dul_yl','gazi_yl','burs_yl',
          'issiz_yl','ddestek_yl','dayni_yl','ydemk_yl',
          'yddav_yl','ydayn_yl','dgnfk_yl','dgbrs_yl',
          'dsemk_yl','dgayn_yl','topgn_yl',
          'topga_ay','topgelay','topga_yl','topgelyl',
          'kira_ay','izafi_k','kondeger','muskdeg',
          'aptkdeg','yazkdeg','tarkdeg','bagkdeg',
          'serakdeg','arsakdeg','arsadeg',
          'dukkdeg','dukdeg','otelkdeg','oteldeg',
          'digmik','digkira','digkdeg','digdeg',
          'yilkulgel','harcama', 'tasarruf')

full[full$year<=2004,noms] <-  full[full$year<=2004,noms]/1000000

full[,noms] <-  full[,noms]/full[,"deflator"]



save(full,file=paste0(prevpath,'FullHBSreal.Rdata'))


load(file=paste0(prevpath,'FullHBSreal.Rdata'))




#Qualitative Vars
qual <- c('cinsiyet','yakinlik','sagliks','gunengel',
  'calengel','ogrenci','egitim_devam','medenidr',
'calis_ay','is_arama','isneden','isbasi',
'calis_yl','sekkod','meslek','cal_kisi',
'sgk','is_statu','huk_dur','koopera',
'sir_kon','isdur','ist_tesk','guc_kul',
'sendika','isfaal','mulk_dur','ekis_yl','
ekis_ay','sekkod_e','meslek_e','isdur_e',
'cl_neden','ayni_is','sekkod_d','meslek_d',
'isdur_d','hhtipi','konuttip','mulkiyet','kira_sek',
'yapituru','zemin_sal','zemin_oda',
'zemin_mut','zemin_ban','isit_sis','yakit_1','yakit_2',
'yakit_3','yakit_mut','yakit_su',
'zor_alis','zor_banka','zor_posta','zor_ulas','zor_saglik',
'zor_egiti','banyo','sauna','jakuzi',
```

```
'tuvalet','mutfak','copogut','kalorif',
'tabanisi','su_sis','elektri','dog_gaz',
'sicaksu','kabloyay','asansor',
'garaj','havuz','jenrator','apartman','yazlik',
'tarla','bag','sera','arsa',
'dukkan','otel','diger', 'kirkntkd')


k1 <- as.list(noms)
k2 <- as.list(qual)
k3<- colnames(full)

q <- setdiff(k3,k2)

#Quantitative Vars
quan <- setdiff(q, c('year','bulten', 'fertno', 'deflator'))



#Descriptive Statistics####
sink(file=paste0(respath,"01.Savings.txt"))

collapse <- summaryBy(tasarruf_orani  ~
  bulten + year , FUN=c(mean,sd), data=full)
collapse <- summaryBy(tasarruf_orani.mean  ~
 year , FUN=mean, data=collapse)

wi <-6
he <- 3.5
pdf(paste0(respath,'savingsratio.pdf'), width = wi, height = he)
ggplot(collapse, aes(year, tasarruf_orani.mean.mean)) +
geom_line()  + xlab("") + ylab("Savings␣Ratio␣at␣HH␣Level")
dev.off()
sink()



sink(file=paste0(respath,"02.DataFrameStr.txt"))

str(full,list.len=ncol(full))

sink()




summaryfunction= function (x){
  if( is.numeric(x)!=TRUE) {stop("Supplied␣X␣is␣not␣numeric")}
  mysummary = data.frame(
    "Min." =as.numeric( min(x)),
    "1st␣Qu." = quantile(x,na.rm = T)[2],
    "Median" = median(x),
    "Mean" = mean(x),
    "3rd␣Qu." = quantile(x,na.rm = T)[4],
    "Max." = max(x),
    "St.Dev." = sd(x),
    row.names=""

  )
  names(mysummary) = c("Min.","1st␣Qu.","Median","Mean","3rd␣Qu.","Max.", "St.Dev.")
  return( mysummary )
}



desc <- full[,quan]

tabu <- full[,qual]


sink(file=paste0(respath,"03.Descriptives.txt"))
stargazer(desc)
sink()
```

```
sink(file=paste0(respath,"04.Tabulates.txt"))
byvars=4
for(i in seq(1,81,byvars)){
  if(i<82){
  print(xtable(summary(tabu[,i:(i+byvars)])))
  } else {
  print(xtable(summary(tabu[,82:ncol(tabu)])))
    }
}

sink()
```

```
#Labeling####
full$year <- factor(full$year,
                    levels = seq(2004,2016),
                    labels = c('2004', '2005', '2006', '2007', '2008', '2009', '2010',
                               '2011', '2012', '2013', '2014', '2015', '2016'))


full$hhtipi <- factor(full$hhtipi,
                          levels = seq(1,7),
                          labels = c('Nuclear␣household␣with␣a␣child',
                            'Nuclear␣household␣with␣two␣children',
                            'Nuclear␣household␣with␣three␣or␣more␣children',
                            'Couple␣without␣children',
                            'Patriarchal␣or␣extensive␣household',
                            'One␣adult␣household',
                            'Persons␣who␣live␣together'))


full$konuttip <- factor(full$konuttip,
                          levels = seq(1,4),
                          labels = c('Detached␣house␣   ␣Independent',
                           'Detached␣house␣   ␣Attached␣to␣a␣single␣wall',
                                     'Apartment␣house','Other'))


full$mulkiyet <- factor(full$mulkiyet,
                          levels = seq(1,4),
                          labels = c('Owner', 'Tenant', 'Lodging', 'Other'))


full$kira_sek <- factor(full$kira_sek,
                          levels = seq(1,2),
                          labels = c('With␣furnishings', 'Without␣furnishings'))


full$kon_borc <- factor(full$kon_borc,
                          levels = seq(1,2),
                          labels = c('Yes', 'No'))


full$bina_tar <- factor(full$bina_tar,
                          levels = seq(1,5),
                          labels = c('1945␣-', '1946␣-␣1960',
                           '1961␣-␣1980' -'1981␣-␣1995', '1996␣+'))


full$yapituru <- factor(full$yapituru,
                          levels = seq(1,5),
```

```
                        labels = c('Reinforced concrete',
                         'Steel', 'Wood', 'Briquette', ' Stone',
                                  'Bricks', 'Adobe', 'Other'))


full$zemin_sal <- factor(full$zemin_sal,
                        levels = seq(1,9),
                        labels = c('Parquet', 'Wood',
                          'Floor tile (ceramic)', 'Vinyl floor covering (pvc)',
                          'Carpet floor covering','Coarse concrete', 'Mosaic',
                          'Marble', 'Other'))


full$zemin_oda <- factor(full$zemin_oda,
                         levels = seq(1,9),
                         labels = c('Parquet', 'Wood',
                          'Floor tile (ceramic)', 'Vinyl floor covering (pvc)',
                                   'Carpet floor covering',
                                   'Coarse concrete', 'Mosaic',
                                   'Marble', 'Other'))


full$zemin_mut <- factor(full$zemin_mut,
                      levels = seq(1,9),
                     labels = c('Parquet', 'Wood',
                       'Floor tile (ceramic)', 'Vinyl floor covering (pvc)',
                       'Carpet floor covering',
                       'Coarse concrete', 'Mosaic',
                       'Marble', 'Other'))


full$zemin_ban <- factor(full$zemin_ban,
                         levels = seq(1,9),
                         labels = c('Parquet', 'Wood',
                          'Floor tile (ceramic)',
                          'Vinyl floor covering (pvc)',
                          'Carpet floor covering',
                          'Coarse concrete', 'Mosaic',
                              'Marble', 'Other'))


full$isit_sis <- factor(full$isit_sis,
                 levels = seq(1,4),
                    labels = c('Stove',
              'Wood', 'Common or central heating',
             'Central heating boiler/Wall hung gas boiler',
              'Other'))


full$yakit_1 <- factor(full$yakit_1,
                         levels = seq(1,6),
                         labels = c('Wood', 'Coal',
                          'Natural gas',
                                   'Electric',
                                   'Dried cow dung', 'Other'))


full$yakit_2 <- factor(full$yakit_2,
                        levels = seq(1,6),
                        labels = c('Wood', 'Coal',
                         'Natural gas',
                         'Electric',
                         'Dried cow dung', 'Other'))
```

```
full$yakit_3 <- factor(full$yakit_3,
                       levels = seq(1,6),
                       labels = c('Wood', 'Coal', 'Natural gas',
                                  'Electric', 'Dried cow dung',
                                   'Other'))


full$yakit_mut <- factor(full$yakit_mut,
                         levels = seq(1,8),
                         labels = c('Wood', 'Coal',
                           'Natural gas', 'LPG',
                            'Electric', 'Dried cow dung',
                             'Electric', 'Solar energy system',
                              'Other'))


full$yakit_su <- factor(full$yakit_su,
                          levels = seq(1,8),
                          labels = c('Wood', 'Coal',
                           'Natural gas', 'LPG',
                           'Electric', 'Dried cow dung',
                           'Electric', 'Solar energy system',
                           'Other'))


full$zor_alis <- factor(full$zor_alis,
                          levels = seq(1,4),
                          labels = c('With a great difficulty',
                           'With difficulty', 'Easily',
                                    'Very easily'))



full$zor_banka <- factor(full$zor_banka,
                         levels = seq(1,4),
                         labels = c('With a great difficulty',
                          'With difficulty', 'Easily',
                                    'Very easily'))


full$zor_posta <- factor(full$zor_posta,
                         levels = seq(1,4),
                         labels = c('With a great difficulty',
                          'With difficulty', 'Easily',
                                    'Very easily'))


full$zor_ulas <- factor(full$zor_ulas,
                         levels = seq(1,4),
                         labels = c('With a great difficulty',
                          'With difficulty', 'Easily',
                                    'Very easily'))



full$zor_saglik <- factor(full$zor_saglik,
                         levels = seq(1,4),
                         labels = c('With a great difficulty',
                          'With difficulty', 'Easily',
                                    'Very easily'))



full$zor_egiti <- factor(full$zor_egiti,
                         levels = seq(1,4),
                         labels = c('With a great difficulty',
                          'With difficulty', 'Easily',
                                    'Very easily'))
```

```
full$konutiki <- factor(full$konutiki,
                        levels = seq(1,2),
                        labels = c('Yes', 'No'))


full$konut2_n <- factor(full$onut2_n,
                        levels = seq(1,3),
                        labels = c('Abroad',
                         'Another province in Turkey',
                          'The same province'))



full$banyo <- factor(full$banyo,
                       levels = seq(0,1),
                       labels = c('Not available', 'Available'))



full$sauna <- factor(full$sauna,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$jakuzi <- factor(full$jakuzi,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$tuvalet <- factor(full$tuvalet,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))



full$mutfak <- factor(full$mutfak,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$copogut <- factor(full$cop,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$kalorif <- factor(full$kalorif,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))




full$tabanisi <- factor(full$tabanisi,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))



full$su_sis <- factor(full$su_sis,
                       levels = seq(0,1),
                       labels = c('Not available', 'Available'))
```

```
full$elektri <- factor(full$elektri,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))


full$dog_gaz <- factor(full$dog_gaz,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))

full$sicaksu <- factor(full$sicaksu,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))

full$kabloyay <- factor(full$kabloyay,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))

full$asansor <- factor(full$asansor,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))


full$garaj <- factor(full$garaj,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))

full$havuz <- factor(full$havuz,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))


full$jenrator <- factor(full$jenrator,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))

full$telefon <- factor(full$telefon,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))


full$muskonut <- factor(full$muskonut,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))



full$apartman <- factor(full$apartman,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))



full$yazlik <- factor(full$yazlik,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))

full$telefon <- factor(full$telefon,
                       levels = seq(0,1),
                       labels = c('Not␣available', 'Available'))
```

```
full$tarla <- factor(full$tarla,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$bag <- factor(full$bag,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$sera <- factor(full$sera,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))

full$arsa <- factor(full$arsa,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$dukkan <- factor(full$dukkan,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))


full$otel <- factor(full$otel,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))




full$diger <- factor(full$diger,
                     levels = seq(0,1),
                     labels = c('Not available', 'Available'))




full$kirkntkd <- factor(full$kirkntkd,
                    levels = seq(1,2),
                    labels = c('Rural', 'Urban'))



full$cinsiyet <- factor(full$cinsiyet,
                      levels = seq(1,2),
                      labels = c('Male', 'Female'))




full$yas <- factor(full$yas,
                    levels = seq(1,13),
                    labels = c('0-5 years',
                     '6-14 years', '15-19 years',
                     '20-24 years', '25-29 years',
                      '30-34 years', '35-39 years',
                      '40-44 years', '45-49 years',
                      '50-54 years','55-59 years',
                      '60-64 years', '65+years'))
```

```
full$yakinlik <- factor(full$yakinlik,
                    levels = seq(1,11),
                    labels = c('Reference␣person',
                     'Spouse', 'Son/Doughter',
                     'Father/Mother',
                     'Father-in-law/Mother-in-law',
                     'Son-in-law/Doughter-in-law',
                     'Brother/Sister',
                     'Grandchild',
                     'Other␣relative',
                     'Housemaid␣lives␣together␣with␣household',
                     'Non-relative'))
```

```
full$sagliks <- factor(full$sagliks,
                        levels = seq(1,4),
                        labels = c('Yes␣-␣SGK␣All',
    'Yes␣-␣Other',
    'Yes␣-␣GSS␣(General␣Health␣Insurance,␣Green␣Health␣Card)', 'No'))
```

```
full$gunengel <- factor(full$gunengel,
                        levels = seq(1,2),
                        labels = c('Yes', 'No'))
```

```
full$calengel <- factor(full$calengel,
                        levels = seq(1,2),
                        labels = c('Yes', 'No'))
```

```
full$egitim <- factor(full$egitim,
                        levels = seq(1,11),
                        labels = c('Illiterate',
                          'Literate␣  ␣not␣completed␣a␣school',
                           'Primary␣school',
                           'Primary␣education',
                           'Secondary␣school',
                           'Junior␣Vocational␣High␣School',
                           'High␣School',
                           'Senior␣Vocational␣High␣School',
                           '2-3␣year-College',
                           '4-year-College␣or␣University',
                           'Post␣Graduate/PhD'))
```

```
full$ogrenci <- factor(full$ogrenci,
                        levels = seq(1,2),
                        labels = c('Yes', 'No'))
```

```
full$egitim_devam <- factor(full$egitim_devam,
                        levels = seq(1,11),
                        labels = c('Illiterate',
                          'Literate␣  ␣not␣completed␣a␣school',
                           'Primary␣school',
                           'Primary␣education',
                           'Secondary␣school',
                           'Junior␣Vocational␣High␣School',
                           'High␣School',
                           'Senior␣Vocational␣High␣School',
                           '2-3␣year-College',
                           '4-year-College␣or␣University',
                           'Post␣Graduate/PhD'))


full$medenidr <- factor(full$medenidr,
                       levels = seq(1,4),
                       labels = c('Never␣married', 'Married',
                                  'Widowed,␣Green␣Health␣Card',
                                  'Divorced'))




full$calis_ay <- factor(full$calis_ay,
                        levels = seq(1,3),
                        labels = c('Worked',
                        'Not␣worked␣but␣work␣connection␣is␣going␣on',
                        'Not␣worked'))




full$is_arama<- factor(full$is_arama,
                        levels = seq(1,2),
                                  labels = c('Yes', 'No'))




full$isneden <- factor(full$egitim_devam,
levels = seq(1,11),
labels = c('Found␣a␣job,␣established␣his/her␣own␣business␣and␣waiting␣to␣get␣started',
'Continuing␣education␣and␣training',
'Busy␣with␣housework','Retired','Property␣income␣earners',
'Elderly␣(not␣retired,␣however,␣thinking␣him/herself␣too␣old␣to␣work,␣aged␣65+)',
'Disabled␣or␣ill', 'Seasonal␣worker',
'Other␣(PS:There␣is␣no␣jump␣in␣order,␣it␣is␣encoded␣as␣above␣inevitably)'))



#Qualitative Data, NA is a factor
for(i in 1:ncol(tabu)){
  tabu[, i] <- addNA(tabu[, i])
}

#Quantitative Data NA <- Median Replacer
for(i in 1:ncol(desc)){
  desc[is.na(desc[,i]), i] <- median(desc[,i], na.rm = TRUE)
}


#Quantile Regression
quantiles <- c(.1, .25, .50,.75 )
qregdata <- data.frame(tabu,desc,full[, c('year','deflator')])
qregdata[,c('tasarruf', 'harcama', 'kirkntkd.1')] <- NULL #Dropping of identical vars that u
```

```r
#Sample seç
qregdata <- qregdata[sample(nrow(qregdata),10000,replace=T),]

desc.noise <- data.frame(lapply(desc, function(x) jitter(x)))
desc.noise <- desc.noise[sample(nrow(qregdata),100000,replace=T),]


qreg1<- rq(tasarruf_orani~ 1,
 tau=quantiles, data=desc.noise)
qreg2<- rq(tasarruf_orani~
  yazkira + digkira+deflator+, tau=quantiles, data=qregdata)
qreg2<- rq(tasarruf_orani~ .,
 tau=quantiles, data=desc.noise)

res2 <- sum((qreg2$residuals)^2)
res2

rho <- function(u,tau=quantiles)u*(tau - (u < 0))
V1 <- sum(rho(qreg1$resid, qreg1$tau))
rho <- function(u,tau=quantiles)u*(tau - (u < 0))
V2 <- sum(rho(qreg2$resid, qreg2$tau))

R1 <- 1-(V2/V1)



sink(file=paste0(respath,"05.Quantreg.txt"))
print(summary(qreg2))
print(R1)
sink()



desc %>%
  select_if(is.numeric) %>%
  cor() %>%
  round(digits = 2) %>%
  as.data.frame() %>%
  filter_all(all_vars(abs(.) > 0.8))
```

3.6 Python Codes for QREG

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 08 18:47:52 2017

#######################################################################
#                      _                 _                 _        #
#     _    _    __ _   ___  (_)  _ __    | | __ _    _ | |_  _    _ | | __    #
#    | |  | |  / _  |/ __|| || || '_ \  | |/ /| |  | || __|| |  | || |/ /    #
#    | |_| || (_| |\__ \| || | | ||    < | |_| || |_ | |_| ||    <     #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\    #
#     |___/                                                           #
#     ____                              _  _                          #
#    / __ \    __ _  _ __ ___     __ _ (_)| |    ___  ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |   / __|/ _ \ | '_   _ \   #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
"""




from __future__ import print_function
import patsy
import numpy as np
import pandas as pd
import seaborn as sns

import os
import time


import statsmodels.api as sm
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
from statsmodels.regression.quantile_regression import QuantReg

from scipy.stats import pointbiserialr, pearsonr


import plotly.offline as py


py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls



from rpy2.robjects import r
from rpy2.robjects import pandas2ri
pandas2ri.activate()
import matplotlib.pyplot as plt




#Logger file for systdout and a txt file
class Logger(object):
    """
    Write log messages to the console (sys.stdout) and to a log file,
    all managed as a context manager:
    >>> with Logger('log_file.txt'):
    ...   print "Hello, World!"  # goes to stdout and to the log file
```

```
    ...
    Hello, World!
    >>> print open('log_file.txt').read()
    Hello, World!
    """
    def write(self, message):
        self.console.write(message)
        self.console.flush()
        self.log_file.write(message)
        self.log_file.flush()

    def close(self):
        self.console.flush()
        self.log_file.flush()
        self.log_file.close()

    def __init__(self, log_file):
        self.console = sys.stdout
        self.log_file_name = log_file
        self.log_file = None

    def __enter__(self):
        self.log_file = open(self.log_file_name, 'w')
        sys.stdout = self
        return self

    def __exit__(self, *args):
        self.close()
        sys.stdout = self.console

        return not any(args)


# Çok ciddi uyarılar yoksa, uyarı gelmesin!
import warnings
warnings.filterwarnings('ignore')

# HLFS Datalarını R'den alıyorum
fileloc = "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"
resloc = "/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper2_DL/Analyses/Results/"


# Klasördeki dosyalar
p = sns.color_palette()
print('# File sizes')
for f in os.listdir(fileloc):
    if 'zip' not in f:
        print(f.ljust(30) +
              str(round(os.path.getsize(fileloc + f) / 1000000, 2)) + 'MB')



# HLFS'deki Rdata yüklenecek
rfile = r['load'](fileloc + os.listdir(fileloc)[
    os.listdir(fileloc).index('FullHBSreal.Rdata')])

hbs = pd.DataFrame(r['full'])



# Intro
hbs.shape
hbs.describe()
hbs.info()
len(hbs.index) - hbs.count()


# NA replacement for RData
tic = time.clock()
hbs = hbs.replace(-2147483648, 999)
toc = time.clock()
toc - tic
```

```
# Imputer or Median filler uygulama
feats = list(hbs.columns.values)
for col in hbs:
    hbs[col].fillna(hbs[col].median(), inplace=True)

len(hbs.index) - hbs.count()




# Short Description of Data
hbs.tasarruf_orani.describe()




#Index Resetting
hbs = hbs.reset_index()



####Initial runlar buraya kadar Hocam!######
##############################################################################




qual=["year","cinsiyet","yakinlik","sagliks","gunengel","calengel","ogrenci",
      "egitim_devam","medenidr","calis_ay","is_arama","isneden","isbasi",
      "calis_yl",
"sekkod","meslek","cal_kisi","sgk","is_statu","huk_dur","koopera","sir_kon",
      "isdur","ist_tesk","guc_kul","sendika","isfaal",
"mulk_dur","ekis_yl","ekis_ay","sekkod_e","meslek_e",
      "isdur_e","cl_neden","ayni_is","sekkod_d",
      "meslek_d","isdur_d","hhtipi","konuttip",
"mulkiyet","kira_sek","yapituru","zemin_sal",
      "zemin_oda","zemin_mut","zemin_ban","isit_sis",
      "yakit_1","yakit_2","yakit_3","yakit_mut","yakit_su",
"zor_alis","zor_banka","zor_posta","zor_ulas","zor_saglik",
      "zor_egiti","banyo","sauna","jakuzi","tuvalet","mutfak","copogut",
      "kalorif",
"tabanisi","su_sis","elektri","dog_gaz","sicaksu","kabloyay",
      "asansor","garaj","havuz","jenrator","apartman","yazlik","tarla",
"bag","sera","arsa","dukkan","otel","diger","kirkntkd"]

quan = ["yas","ailegrup","sagliks2","egitim","sure_yil","calsur","kac_kisi",
        "ucrn_ay","ucra_ay","ucrn_yil","ikr_yil",
"prim_yil","ucra_yil","konayni","ulsayni","egsayni","yemayni","kresayni",
"giyayni","gidaayni","digayni","mutsn_ay",
"mutsa_ay","mutsn_yl","mutsa_yl","targ_ay","targ_yl","calsur_e","ekgln_ay",
"ekgla_ay","ekgln_yl","ekgla_yl","calay_yil",
"calsur_d","digln_yl","digla_yl","gmkn_ay","gmka_ay","gmkn_yl","gmka_yl",
"mulkn_ay","mulka_ay","banka_yl","doviz_yl",
"nema_yl","temet_yl","kar_yl","toprn_yl","topra_yl","trsn_ay","trsa_ay",
"emekl_yl","viade_yl","yasli_yl","sosy_yl",
"hasta_yl","dul_yl","gazi_yl","burs_yl","issiz_yl","ddestek_yl","dayni_yl",
"ydemk_yl","yddav_yl","ydayn_yl","dgnfk_yl",
"dgbrs_yl","dsemk_yl","dgayn_yl","topgn_yl","topga_ay","topgelay","topgna_yl",
"topga_yl","topgelyl","indeks_arl","kira_ay",
"loj_fert","izafi_k","konutsah","kon_sure","kon_borc","kondeger","otur_sur",
"bina_tar","oda_say","kon_alan",
"konutiki","konut2_n","sudepo","telefon","ceptel","bilgisay","intern",
"televiz","video","lcdtele",
"dvd_vcd","kamera","uyduant","buzdolab","derindon","bulasik","m_firin",
"camasir","kurutma","haliyika","klima",
"otomobil","otoadet","isveroto","isotoadt","motosik","motoadet","deniztas",
"dentadet","muskonut","musmik","muskira",
"muskdeg","aptmik","aptkira","aptkdeg","yazmik","yazkira","yazkdeg","tarmik",
"tarkira","tarkdeg","bagmik",
```

```
"bagkira","bagkdeg","seramik","serakira","serakdeg","arsamik","arsakira",
"arsakdeg","arsadeg","dukmik","dukkira",
"dukkdeg","dukdeg","otelmik","otelkira","otelkdeg","oteldeg","digmik",
"digkira","digkdeg","digdeg","yilkulgel",
"efb_oecd","hhb","faktor"]



#Point BiSerial Correlation Matrix
w, h = len(qual), len(qual)
corr = [[0 for x in range(w)] for y in range(h)]
pval = [[0 for x in range(w)] for y in range(h)]

for i in range(len(qual)):
    for j in range(i+1,len(qual)):
        res = pointbiserialr(hbs[qual[i]], hbs[qual[j]])
        corr[i][j] = res[0]
        pval[i][j] = res[1]

#Point BiSerial Correlation List, Corr>.80 alerted, 4 digit decimals
biser = []
biser70 = []
biser70vars= []
for i in range(len(qual)):
    for j in range(i+1,len(qual)):
        res = pointbiserialr(hbs[qual[i]], hbs[qual[j]])
        res2=[qual[i],qual[j],\
              '{:.4f}'.format(res[0]), '{:.4f}'.format(res[1])]
        biser70vars.append(qual[j])
        biser.append(res2)
        if res[0]>.7:
            biser70.append(res2)

full=pd.DataFrame(biser)
full.to_csv(resloc+'06.BiSerial_All.csv', sep=';', encoding='utf-8')

full=pd.DataFrame(biser70)
full.to_csv(resloc+'06.BiSerial_70.csv', sep=';', encoding='utf-8')




#Final Qualitative Set
qual=["year","cinsiyet","yakinlik","sagliks","ogrenci",
      "egitim_devam","medenidr","is_arama","isneden",
"sekkod","sgk","is_statu","koopera","sir_kon",
      "isdur","ist_tesk","guc_kul","isfaal",
"mulk_dur","ekis_yl","ekis_ay","sekkod_e","meslek_e",
      "isdur_e","cl_neden","ayni_is","sekkod_d",
      "meslek_d","isdur_d","hhtipi","konuttip",
"mulkiyet","kira_sek","yapituru","zemin_sal",
"zemin_mut","zemin_ban","isit_sis",
      "yakit_1","yakit_2","yakit_3","yakit_mut","yakit_su",
      "zor_egiti","banyo","sauna","jakuzi","tuvalet","mutfak","copogut",
"tabanisi","su_sis","elektri","sicaksu","kabloyay",
      "asansor","garaj","havuz","jenrator","apartman","yazlik","tarla",
"bag","sera","arsa","dukkan","otel","diger","kirkntkd"]


hbs[qual] = hbs[qual].astype('category')




#Pearson Correlation Matrix
w, h = len(quan), len(quan)
corrp = [[0 for x in range(w)] for y in range(h)]
pvalp = [[0 for x in range(w)] for y in range(h)]
```

```
for i in range(len(quan)):
    for j in range(i+1,len(quan)):
        res = pearsonr(hbs[quan[i]], hbs[quan[j]])
        corrp[i][j] = res[0]
        pvalp[i][j] = res[1]


#Point BiSerial Correlation List, Corr>.80 alerted, 4 digit decimals
pear = []
pear60 = []
pear60vars= []
for i in range(len(quan)):
    for j in range(i+1,len(quan)):
        res = pointbiserialr(hbs[quan[i]], hbs[quan[j]])
        res2=[quan[i],quan[j],\
              '{:.4f}'.format(res[0]), '{:.4f}'.format(res[1])]
        pear60vars.append(quan[j])
        pear.append(res2)
        if res[0]>.6:
            pear60.append(res2)

full=pd.DataFrame(pear)
full.to_csv(resloc+'07.Pearson_All.csv', sep=';', encoding='utf-8')

full=pd.DataFrame(pear60)
full.to_csv(resloc+'07.Pearson_60.csv', sep=';', encoding='utf-8')



#final Quantitative Set
quan = ["yas","ailegrup","egitim","sure_yil","kac_kisi","ucrn_ay","ucra_ay",
        "ucrn_yil","ikr_yil",
"prim_yil","konayni","ulsayni","egsayni","yemayni","kresayni","giyayni",
"gidaayni","digayni",
"calsur_e","ekgla_ay","ekgln_yl","ekgla_yl","calay_yil",
"digln_yl","digla_yl","gmkn_ay","gmka_ay","gmkn_yl","gmka_yl","mulka_ay",
"banka_yl","doviz_yl",
"nema_yl","temet_yl","kar_yl","toprn_yl","topra_yl","trsn_ay","emekl_yl",
"viade_yl","yasli_yl","sosy_yl",
"hasta_yl","dul_yl","gazi_yl","burs_yl","issiz_yl","ddestek_yl","dayni_yl",
"ydemk_yl","yddav_yl","ydayn_yl","dgnfk_yl",
"dgbrs_yl","dsemk_yl","dgayn_yl","topga_ay","topgelay","topga_yl","topgelyl",
"indeks_arl","kira_ay",
"loj_fert","izafi_k","konutsah","kon_sure","kon_borc","kondeger","otur_sur",
"bina_tar","oda_say","kon_alan",
"konutiki","konut2_n","sudepo","telefon","ceptel","bilgisay","intern",
"televiz","video","lcdtele",
"dvd_vcd","kamera","uyduant","buzdolab","derindon","bulasik","m_firin",
"camasir","kurutma","haliyika","klima",
"otomobil","otoadet","isotoadt","motoadet","dentadet","muskonut",
"musmik","muskira",
"muskdeg","aptkira","aptkdeg","yazmik","yazkira","yazkdeg","tarmik",
"tarkira","tarkdeg","bagmik",
"bagkira","bagkdeg","seramik","serakira","serakdeg","arsamik",
"arsakira","arsakdeg","arsadeg","dukkira",
"dukkdeg","dukdeg","otelmik","otelkira","otelkdeg","oteldeg",
"digmik","digkira","digkdeg","digdeg","yilkulgel",
"hhb","faktor"]



# Imputer or Median filler uygulama
feats = list(hbs[quan].columns.values)
for col in feats:
    hbs[col].fillna(hbs[col].median(), inplace=True)



#FINAL QREG Estimation in StatModels
with Logger(resloc+'08QREG.txt'):
    pred = []
    quantiles = [.1, .2, .4, .8 ]
```

```
    for q in quantiles:
        hbs[qual] = hbs[qual].astype('category')
        n=hbs[quan].columns
        l=hbs[qual].columns
        nl = []
        for i in l:
            nl.append(i)
        for i in n:
            nl.append(i)
        all_columns = "+".join(nl)
        from statsmodels.regression.quantile_regression import QuantReg
        my_formula = 'tasarruf_orani~' + all_columns
        mod = smf.quantreg(formula=my_formula, data=hbs)
        res = mod.fit(q=q)
        pred.append(res.predict())
        print(res.summary().as_latex())



full=pd.DataFrame(pred[0])
full.to_csv(resloc+'10.QREG_Predictions-10.csv', sep=';', encoding='utf-8')
full=pd.DataFrame(pred[1])
full.to_csv(resloc+'10.QREG_Predictions-20.csv', sep=';', encoding='utf-8')
full=pd.DataFrame(pred[2])
full.to_csv(resloc+'10.QREG_Predictions-40.csv', sep=';', encoding='utf-8')
full=pd.DataFrame(pred[3])
full.to_csv(resloc+'10.QREG_Predictions-80.csv', sep=';', encoding='utf-8')




#Labeling Example
d={0:'aaa',1:'a',2:'b',3:'a',4:'b',5:'a',6:'b',7:'a',8:'b',9:'a',10:'b'}
hbs.yakinlik.apply(lambda x: d[x])




########################################################################
# Feature Extraction with Feature Selector Module (if not clear this) #
########################################################################
from feature_selector import FeatureSelector

fs = FeatureSelector(data = hbs[nl], labels = hbs.tasarruf_orani)

# Missing Value Analysis
fs.identify_missing(missing_threshold = 0.6) #Missings Threshold is 0.60
fs.missing_stats.head()
missing_features = fs.ops['missing']
fs.plot_missing()

#Save Missing Values Analysis as data
missings=pd.DataFrame(fs.missing_stats)
#Export
missings.to_csv(resloc+'20.FeatureExtracion_Missings.csv', sep=';')



# Colinear Variable Analysis
fs.identify_collinear(correlation_threshold = 0.80)
fs.plot_collinear()
collinear_features = fs.ops['collinear']
fs.record_collinear.head()

#Save Collinear Variables Analysis as data
collinears=pd.DataFrame(fs.record_collinear)
#Export
collinears.to_csv(resloc+'21.FeatureExtracion_Collinears.csv', sep=';')



# Feature Imjportance Analysis
```

```
# Pass in the appropriate parameters
fs.identify_zero_importance(task = 'regression',
                            eval_metric = 'auc',
                            n_iterations = 10,
                             early_stopping = True)

# list of zero importance features
zero_importance_features = fs.ops['zero_importance']

fs.plot_feature_importances(threshold = 0.90, plot_n = 32)
fs.identify_low_importance(cumulative_importance = 0.90)
fs.feature_importances.head(50)




#Save Importance of Variables as data
importance=pd.DataFrame(fs.feature_importances)
#Export
importance.to_csv(resloc+'22.FeatureExtracion_Importance.csv', sep=';')



# Remove Variables in data after FeatureSelector #
nl = fs.remove(methods = 'all')
nl = nl.columns
#####################


#FINAL QREG Estimation in StatModels
with Logger(resloc+'08QREGwFE.txt'):
    pred = []
    quantiles = [.1, .2, .4, .8 ]
    for q in quantiles:
        hbs[qual] = hbs[qual].astype('category')
        all_columns = "+".join(nl)
        from statsmodels.regression.quantile_regression import QuantReg
        my_formula = 'tasarruf_orani~' + all_columns
        mod = smf.quantreg(formula=my_formula, data=hbs)
        res = mod.fit(q=q)
        pred.append(res.predict())
        print(res.summary().as_latex())



full=pd.DataFrame(pred[0])
full.to_csv(resloc+'10.QREG_Predictions-10wFE.csv', sep=';', encoding='utf-8')
full=pd.DataFrame(pred[1])
full.to_csv(resloc+'10.QREG_Predictions-20wFE.csv', sep=';', encoding='utf-8')
full=pd.DataFrame(pred[2])
full.to_csv(resloc+'10.QREG_Predictions-40wFE.csv', sep=';', encoding='utf-8')
full=pd.DataFrame(pred[3])
full.to_csv(resloc+'10.QREG_Predictions-80wFE.csv', sep=';', encoding='utf-8')
```

3.7 Python Codes for Deep Learning

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 08 18:47:52 2017

######################################################################
#                          _          _              _          _     #
#     _   _    __ _   ___ (_) _ __   | | __ _    _ | |_  _   _ | | __ #
#    | | | |  / _  |/ __|| || '_ \  | |/ /| |  | || __|| | | || |/ /  #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <     #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\    #
#     |___/                                                          #
#     ____                                 _  _                      #
#    / __ \    __ _   _ __ ___     __ _ (_)| |     ___  ___    _ __ ___   #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_   _ \  #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                  #
######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
######################################################################
"""

#Backward Compatibility
from __future__ import print_function

#ScikitLearn Modules
from sklearn.feature_extraction import DictVectorizer
from sklearn.model_selection import StratifiedShuffleSplit,\
 StratifiedKFold, train_test_split, KFold, GroupKFold
from sklearn.preprocessing import MinMaxScaler
from sklearn.utils import shuffle

#Keras Modules
from keras import utils
from keras import backend as K

#Other Modules
from tabulate import tabulate
import numpy as np
import numpy.random as rng
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
import sys
import time

#R 3.2 Dataset Import Tools
from rpy2.robjects import r
from rpy2.robjects import pandas2ri
#Activation of R in Python
pandas2ri.activate()
import matplotlib.pyplot as plt

# import pydot
# import graphviz

#Tensorflow Import
import tensorflow as tf



#graphical Modules
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
```

```
#Logger file for deep learning implementation and a txt file
class Logger(object):
    """
    Write log messages to the console (sys.stdout) and to a log file,
    all managed as a context manager:
    >>> with Logger('log_file.txt'):
    ...   print "Hello, World!"  # goes to stdout and to the log file
    ...
    Hello, World!
    >>> print open('log_file.txt').read()
    Hello, World!
    """
    def write(self, message):
        self.console.write(message)
        self.console.flush()
        self.log_file.write(message)
        self.log_file.flush()

    def close(self):
        self.console.flush()
        self.log_file.flush()
        self.log_file.close()

    def __init__(self, log_file):
        self.console = sys.stdout
        self.log_file_name = log_file
        self.log_file = None

    def __enter__(self):
        self.log_file = open(self.log_file_name, 'w')
        sys.stdout = self
        return self

    def __exit__(self, *args):
        self.close()
        sys.stdout = self.console

        return not any(args)


# Çok ciddi uyarılar yoksa, uyarı gelmesin!
import warnings
warnings.filterwarnings('ignore')


# Import HBS Dataset from R
fileloc = "/media/DRIVE/Dropbox/_VeriTabanlarim/HBS/"
resloc = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper2_DL/Analyses/Results/'


# Seeing the datasets in the folder named "_Veritabanlarim"
p = sns.color_palette()
print('#␣File␣sizes')
for f in os.listdir(fileloc):
    if 'zip' not in f:
        print(f.ljust(30) +
              str(round(os.path.getsize(fileloc + f) / 1000000, 2)) + 'MB')



#Load HBS files as pandas dataframe
rfile = r['load'](fileloc + os.listdir(fileloc)[
    os.listdir(fileloc).index('FullHBSreal.Rdata')])
```

```
hbs = pd.DataFrame(r['full'])


# Pandas dataframe setting to see more row
pd.set_option('display.max_rows', 500)
print(len(hbs.index) - hbs.count())


# Intro
hbs.shape
hbs.describe()
hbs.info()
len(hbs.index) - hbs.count()


# NA replacement for RData
tic = time.clock()
hbs = hbs.replace(-2147483648, 999)
toc = time.clock()
toc - tic



# Short Description of Data
hbs.tasarruf_orani.describe()




#Index Resetting
hbs = hbs.reset_index()







####Initial runlar buraya kadar Hocam!######
################################################################################


#Final Qualitative Set
qual=["year","cinsiyet","yakinlik","sagliks","ogrenci",
      "egitim_devam","medenidr","is_arama","isneden",
"sekkod","sgk","is_statu","koopera","sir_kon",
      "isdur","ist_tesk","guc_kul","isfaal",
"mulk_dur","ekis_yl","ekis_ay","sekkod_e","meslek_e",
      "isdur_e","cl_neden","ayni_is","sekkod_d",
      "meslek_d","isdur_d","hhtipi","konuttip",
"mulkiyet","kira_sek","yapituru","zemin_sal",
"zemin_mut","zemin_ban","isit_sis",
      "yakit_1","yakit_2","yakit_3","yakit_mut","yakit_su",
      "zor_egiti","banyo","sauna","jakuzi","tuvalet","mutfak","copogut",
"tabanisi","su_sis","elektri","sicaksu","kabloyay",
      "asansor","garaj","havuz","jenrator","apartman","yazlik","tarla",
"bag","sera","arsa","dukkan","otel","diger","kirkntkd"]


hbs[qual] = hbs[qual].astype('category')


#Final Quantitative Set
quan = ["yas","ailegrup","egitim","sure_yil","kac_kisi","ucrn_ay",
        "ucra_ay","ucrn_yil","ikr_yil",
"prim_yil","konayni","ulsayni","egsayni","yemayni","kresayni",
"giyayni","gidaayni","digayni",
"calsur_e","ekgla_ay","ekgln_yl","ekgla_yl","calay_yil",
"digln_yl","digla_yl","gmkn_ay","gmka_ay","gmkn_yl","gmka_yl",
```

```
"mulka_ay","banka_yl","doviz_yl",
"nema_yl","temet_yl","kar_yl","toprn_yl","topra_yl","trsn_ay",
"emekl_yl","viade_yl","yasli_yl","sosy_yl",
"hasta_yl","dul_yl","gazi_yl","burs_yl","issiz_yl","ddestek_yl",
"dayni_yl","ydemk_yl","yddav_yl","ydayn_yl","dgnfk_yl",
"dgbrs_yl","dsemk_yl","dgayn_yl","topga_ay","topgelay","topga_yl",
"topgelyl","indeks_arl","kira_ay",
"loj_fert","izafi_k","konutsah","kon_sure","kon_borc","kondeger",
"otur_sur","bina_tar","oda_say","kon_alan",
"konutiki","konut2_n","sudepo","telefon","ceptel","bilgisay",
"intern","televiz","video","lcdtele",
"dvd_vcd","kamera","uyduant","buzdolab","derindon","bulasik",
"m_firin","camasir","kurutma","haliyika","klima",
"otomobil","otoadet","isotoadt","motoadet","dentadet","muskonut",
"musmik","muskira",
"muskdeg","aptkira","aptkdeg","yazmik","yazkira","yazkdeg",
"tarmik","tarkira","tarkdeg","bagmik",
"bagkira","bagkdeg","seramik","serakira","serakdeg","arsamik",
"arsakira","arsakdeg","arsadeg","dukkira",
"dukkdeg","dukdeg","otelmik","otelkira","otelkdeg","oteldeg",
"digmik","digkira","digkdeg","digdeg","yilkulgel",
"hhb","faktor"]

n = hbs[quan].columns
l = hbs[qual].columns

# Selecting only Quantiative and Qualitative Vars
nl = []
for i in l:
    nl.append(i)
for i in n:
    nl.append(i)




# Imputer or Median filler uygulama
feats = list(hbs[n].columns.values)
for col in feats:
    hbs[col].fillna(hbs[col].median(), inplace=True)







##X ve Label ayrımı, grup stratifiedkfold için
#X = X.iloc[0:100000,]
#y = y.iloc[0:100000,]
#groups = hbs.year



##Normalization
#yil=X.yil
#X = X.drop('yil',1)
#min_max_scaler = preprocessing.MinMaxScaler()
#X = pd.DataFrame(min_max_scaler.fit_transform(X))
#X['yil'] = yil




################################################
########## STRATIFIED K-FOLD ALGORITHM ##########
################################################
X = hbs.loc[:, hbs.columns != 'tasarruf_orani']

groups = X.year
#Her yıl'ı fold alıp, fold'lardan rassal veri seçtiriyorum
# Train and Test Sampling, Stratified Yearly
kfold_splits = 13  # i determined geneal k-fold
```

```
skf = KFold(n_splits=kfold_splits, shuffle=True)
# Shuffle false since i do not want to use them again.
skf.get_n_splits(X, y, groups=X.year)

print(skf)

for train_index, test_index in skf.split(X, y, groups=groups):
   print("TRAIN:", train_index, "TEST:", test_index)
   X_train, X_test = X.iloc[train_index], X.iloc[test_index]
   y_train, y_test = y.iloc[train_index], y.iloc[test_index]


X_train, y_train = shuffle(X_train, y_train, random_state=0)
X_test, y_test = shuffle(X_test, y_test, random_state=0)









############ Continue with ALL VARs ##############
################################################


X = hbs[nl]
y = hbs.tasarruf_orani


#Deneme Sample
#X = X_train.iloc[0:10000,]
#y = y_train.iloc[0:10000,]


X_train = X_train.reset_index(drop=True)
X_test = X_test.reset_index(drop=True)
y_train = y_train.reset_index(drop=True)
y_test = y_test.reset_index(drop=True)



# Min Max Scaler
mms=MinMaxScaler()

#Remove cols after FeatureSelector
#n = list(set(n).intersection(train_removed.columns))+['year']

pd.isnull(X_train[n]).sum() > 0

X_train[n] = mms.fit_transform(X_train[n])
X_test = mms.fit_transform(X_test[n])



#Hyperparameters
alpha = 0.003
datasize = X_train.shape[0]
batch_size = 10000
epoch = 100




## Loss /w Quantile Regression

with Logger(resloc+'90.DL_training.txt'):
    pred = []
    loss_history = []
    quantiles = [.1, .2, .4, .8 ]
```

```
    for quantile in quantiles:
        # Initialize session, Minibatches Gradient Descent
        sess = tf.Session()
        # Placeholders
        x_data = tf.placeholder(shape=[None, X_train.shape[1]],
                                dtype=tf.float32)
        y_target = tf.placeholder(shape=[None], dtype=tf.float32)
        # parameters
        W1 = tf.Variable(tf.random_normal(shape=[X_train.shape[1], 250]))
        b1 = tf.Variable(tf.random_normal(shape=[250]))
        hidden_1 = tf.nn.tanh(tf.add(tf.matmul(x_data, W1), b1))

        W2 = tf.Variable(tf.random_normal(shape=[250, 20]))
        b2 = tf.Variable(tf.random_normal(shape=[20]))
        hidden_2 = tf.nn.relu(tf.add(tf.matmul(hidden_1, W2), b2))

        W3 = tf.Variable(tf.random_normal(shape=[20, 4]))
        b3 = tf.Variable(tf.random_normal(shape=[4]))
        hidden_3 = tf.nn.softmax(tf.add(tf.matmul(hidden_2, W3), b3))

        W4 = tf.Variable(tf.random_normal(shape=[4, 1]))
        b4 = tf.Variable(tf.random_normal(shape=[1]))
        output = tf.nn.tanh(tf.add(tf.matmul(hidden_3, W4), b4))

        # loss
        error = tf.subtract(y_target, output)
        loss = tf.reduce_mean(
                tf.maximum(quantile * error, (quantile - 1) * error), axis=-1)
        #tf.summary.scalar('loss', loss)

        # optimize
        optimizer = tf.train.AdamOptimizer(learning_rate=alpha,  beta1=0.9,
        beta2=0.999, epsilon=1e-08, use_locking=False, name='Adam')
        train_step = optimizer.minimize(loss)



        # Writer Definition
        #logfile = resloc + '90.DL_Quantile_'+str(quantile)
        #writer = tf.summary.FileWriter(logfile, sess.graph)

        #Initialization of TF Session
        init = tf.global_variables_initializer()
        sess.run(init)


        for e in range(epoch):
            epoch_losses = []
            epoch_preds = []
            for i in range(0,datasize,batch_size):
                merge = tf.summary.merge_all()
                if datasize == batch_size:
                    batch_start_idx = 0
                elif datasize < batch_size:
                    raise ValueError(
                        "datapoint_size: %d, must be greater than batch_size: \
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣%d" % (datasize, batch_size))
                else:
                    batch_start_idx = (i) % (datasize - batch_size)
                    #batch_start_idx = (i* batch_size) % (datasize - batch_size)
                batch_end_idx = batch_start_idx + batch_size
                batch_xs = X_train.iloc[batch_start_idx:batch_end_idx]
                batch_ys = y_train.iloc[batch_start_idx:batch_end_idx]
                xs = np.array(batch_xs)
                ys = np.array(batch_ys)
                if i % 10000 == 0:
                    preds,result, lossf =\
                    sess.run([output,train_step, loss],  \
                             feed_dict={x_data: xs, y_target: ys})
                    epoch_losses.append(lossf)
                    epoch_preds.append(preds)
                    print("For quantile {}, Loss function is {} \
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣where Minibatches are in {} and {}".\
                          format(quantile,lossf[0], batch_start_idx,\
```

```
                                batch_end_idx))
            epoch_loss = np.mean(epoch_losses)
            loss_history = np.append(loss_history, epoch_loss)
            print("For␣quantile␣{},␣Loss␣function␣is␣{}␣in␣EPOCH␣{},␣".\
                  format(quantile, epoch_loss,e + 1))
    predictions = sess.run(output,\
                           feed_dict={x_data:X_train, y_target:y_train})
    pred.append(predictions)
    #pred.append(predict)

    lossq10=list([loss_history[0:epoch], range(1,epoch+1)][0])
    lossq20=list([loss_history[epoch:epoch*2], range(1,epoch+1)][0])
    lossq40=list([loss_history[epoch*2:epoch*3], range(1,epoch+1)][0])
    lossq80=list([loss_history[epoch*3:epoch*4], range(1,epoch+1)][0])


    AvLossQ10 = sum(lossq10) / len(lossq10)
    AvLossQ20 = sum(lossq20) / len(lossq20)
    AvLossQ40 = sum(lossq40) / len(lossq40)
    AvLossQ80 = sum(lossq80) / len(lossq80)

    AvLossQL = sum(loss_history) / len(loss_history)
    print("Quantile␣Regression␣Average␣Q10␣Loss␣Function␣is␣{0:.8f}.".\
        format(AvLossQ10))
    print("Quantile␣Regression␣Q10␣Last␣Loss␣Function␣is␣{0:.8f}.".\
        format(lossq10[len(lossq10)-1]))

    print("Quantile␣Regression␣Average␣Q20␣Loss␣Function␣is␣{0:.8f}.".\
        format(AvLossQ20))
    print("Quantile␣Regression␣Q20␣Last␣Loss␣Function␣is␣{0:.8f}.".\
        format(lossq20[len(lossq20)-1]))

    print("Quantile␣Regression␣Average␣Q40␣Loss␣Function␣is␣{0:.8f}.".\
        format(AvLossQ40))
    print("Quantile␣Regression␣Q40␣Last␣Loss␣Function␣is␣{0:.8f}.".\
        format(lossq40[len(lossq40)-1]))

    print("Quantile␣Regression␣Average␣Q80␣Loss␣Function␣is␣{0:.8f}.".\
        format(AvLossQ80))
    print("Quantile␣Regression␣Q80␣Last␣Loss␣Function␣is␣{0:.8f}.".\
        format(lossq80[len(lossq80)-1]))

    print("Quantile␣Regression␣Average␣Loss␣Function␣is␣{0:.8f}.".\
        format(AvLossQL))
    print("Quantile␣Regression␣Average␣Last␣Loss␣Function␣is␣{0:.8}.".format(\
          (lossq10[len(lossq10)-1]+ lossq20[len(lossq20)-1]+ \
           lossq40[len(lossq40)-1]+lossq80[len(lossq80)-1]) /4))


## Loss /w Normal Regression




with Logger(resloc+'99.DL_training_normalloss.txt'):
    pred = []
    # Initialize session, Minibatches Gradient Descent
    sess = tf.Session()
    # Placeholders
    x_data = tf.placeholder(shape=[None, X_train.shape[1]],
                            dtype=tf.float32)
    y_target = tf.placeholder(shape=[None], dtype=tf.float32)
    # parameters
    W1 = tf.Variable(tf.random_normal(shape=[X_train.shape[1], 250]))
    b1 = tf.Variable(tf.random_normal(shape=[250]))
    hidden_1 = tf.nn.tanh(tf.add(tf.matmul(x_data, W1), b1))

    W2 = tf.Variable(tf.random_normal(shape=[250, 20]))
    b2 = tf.Variable(tf.random_normal(shape=[20]))
    hidden_2 = tf.nn.relu(tf.add(tf.matmul(hidden_1, W2), b2))

    W3 = tf.Variable(tf.random_normal(shape=[20, 4]))
    b3 = tf.Variable(tf.random_normal(shape=[4]))
```

```
    hidden_3 = tf.nn.softmax(tf.add(tf.matmul(hidden_2, W3), b3))

    W4 = tf.Variable(tf.random_normal(shape=[4, 1]))
    b4 = tf.Variable(tf.random_normal(shape=[1]))
    output = tf.nn.tanh(tf.add(tf.matmul(hidden_3, W4), b4))


    # loss
    loss = tf.reduce_mean(tf.square(y_target - output))
    #tf.summary.scalar('loss', loss)
    init = tf.global_variables_initializer()
    sess.run(init)
    # Declare an optimizer: here i use gradient descent
    my_opt = tf.train.GradientDescentOptimizer(learning_rate=alpha)

    #Create the train step
    train_step = my_opt.minimize(loss)


    loss_vec = []

    for i in range(epoch):
        rand_index = np.random.choice(len(X_train), size=batch_size)
        X = X_train.iloc[rand_index]
        Y = y_train.iloc[rand_index]
        sess.run(train_step, feed_dict={x_data: X, y_target:Y})
        o = sess.run(output, feed_dict={x_data: X, y_target:Y})
        temp_loss = sess.run(loss, feed_dict={x_data: X, y_target: Y})
        loss_vec.append(temp_loss)
        pred.append(o)


    AvLossNL = sum(loss_vec) / len(loss_vec)

    print("Normal Regression Average Loss Function is {0:.8f}.".\
        format(AvLossNL))
    print("Normal Regression Last Loss Function is {0:.8f}.".\
        format(loss_vec[len(loss_vec)-1]))




text_file = open(resloc+'90.DL_training.txt', "r")
lines = text_file.readlines()
epocloss=[]
for line in lines:
    if "EPOCH" in line:
        line=line.replace('For quantile ', '')
        line = line.replace(' in EPOCH ', '')
        line = line.replace('Loss function is ', '')
        line = line.replace('  \n',' ')
        ########line = line.split(',')
        ########epocloss.append(line)

epocloss = epocloss.apply(pd.to_numeric, errors='coerce')

epocloss10=pd.DataFrame(epocloss[0:100],
                        columns= ('Quantile', 'Epoch', 'Loss'))
epocloss10= epocloss10.apply(pd.to_numeric, errors='coerce')
epocloss20=pd.DataFrame(epocloss[100:200],
                        columns= ('Quantile', 'Epoch', 'Loss'))
epocloss20= epocloss20.apply(pd.to_numeric, errors='coerce')
epocloss40=pd.DataFrame(epocloss[200:300],
                        columns= ('Quantile', 'Epoch', 'Loss'))
epocloss40= epocloss40.apply(pd.to_numeric, errors='coerce')
epocloss80=pd.DataFrame(epocloss[300:400],
                        columns= ('Quantile', 'Epoch', 'Loss'))
epocloss80= epocloss80.apply(pd.to_numeric,
                             errors='coerce')


import seaborn as sns
import matplotlib.pyplot as plt
```

```
import␣pylab

sns.set_style("darkgrid")
plt.figure()
sns10=sns.tsplot(epocloss10['Loss'],epocloss10['Epoch'],␣)
sns10.set(xlabel='Number of EPOCH',␣ylabel='QREG Loss (q=0.10)')
pylab.savefig(resloc+'epoch10.pdf')
plt.close()

sns20=sns.tsplot(epocloss20['Loss'],epocloss10['Epoch'],␣)
sns20.set(xlabel='Number of EPOCH',␣ylabel='QREG Loss (q=0.20)')
pylab.savefig(resloc+'epoch20.pdf')
plt.close()

sns40=sns.tsplot(epocloss40['Loss'],epocloss10['Epoch'],␣)
sns40.set(xlabel='Number of EPOCH',␣ylabel='QREG Loss (q=0.40)')
pylab.savefig(resloc+'epoch40.pdf')
plt.close()

sns80=sns.tsplot(epocloss80['Loss'],epocloss10['Epoch'],␣)
sns80.set(xlabel='Number of EPOCH',␣ylabel='QREG Loss (q=0.80)')
pylab.savefig(resloc+'epoch80.pdf')
plt.close()
```

```
#########################################################################
#␣Feature␣Extraction␣with␣Feature␣Selector␣Module␣(if␣not␣clear␣this)␣#
#########################################################################
from␣feature_selector␣import␣FeatureSelector

fs␣=␣FeatureSelector(data␣=␣hbs[nl],␣labels␣=␣hbs.tasarruf_orani)

#␣Missing␣Value␣Analysis
fs.identify_missing(missing_threshold␣=␣0.6)␣#Missings␣Threshold␣is␣0.60
fs.missing_stats.head()
missing_features␣=␣fs.ops['missing']
fs.plot_missing()

#Save␣Missing␣Values␣Analysis␣as␣data
missings=pd.DataFrame(fs.missing_stats)
#Export
missings.to_csv(resloc+'20.FeatureExtracion_Missings.csv',␣sep=';')



#␣Colinear␣Variable␣Analysis
fs.identify_collinear(correlation_threshold␣=␣0.80)
fs.plot_collinear()
collinear_features␣=␣fs.ops['collinear']
fs.record_collinear.head()


#Save␣Collinear␣Variables␣Analysis␣as␣data
collinears=pd.DataFrame(fs.record_collinear)
#Export
collinears.to_csv(resloc+'21.FeatureExtracion_Collinears.csv',␣sep=';')



#␣Feature␣Imjportance␣Analysis

#␣Pass␣in␣the␣appropriate␣parameters
fs.identify_zero_importance(task␣=␣'regression',
␣␣␣
```

```
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣n_iterations␣=␣10,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣early_stopping␣=␣True)

#␣list␣of␣zero␣importance␣features
zero_importance_features␣=␣fs.ops['zero_importance']

fs.plot_feature_importances(threshold␣=␣0.90,␣plot_n␣=␣32)
fs.identify_low_importance(cumulative_importance␣=␣0.90)
fs.feature_importances.head(50)




#Save␣Importance␣of␣Variables␣as␣data
importance=pd.DataFrame(fs.feature_importances)
#Export
importance.to_csv(resloc+'22.FeatureExtracion_Importance.csv',␣sep=';')



#␣Remove␣Variables␣in␣data␣after␣FeatureSelector␣#
train_removed␣=␣fs.remove(methods␣=␣'all')
X␣=␣hbs[train_removed.columns]
y␣=␣hbs.tasarruf_orani
#####################




################################################
##########␣STRATIFIED␣K-FOLD␣ALGORITHM␣###########
################################################
X␣=␣X.loc[:,␣X.columns␣!=␣'tasarruf_orani']

groups␣=␣X.year
#Her␣yıl'ı fold alıp, fold'lardan␣rassal␣veri␣seçtiriyorum
#␣Train␣and␣Test␣Sampling,␣Stratified␣Yearly
kfold_splits␣=␣13␣␣#␣i␣determined␣geneal␣k-fold
skf␣=␣KFold(n_splits=kfold_splits,␣shuffle=True)
#␣Shuffle␣false␣since␣i␣do␣not␣want␣to␣use␣them␣again.
skf.get_n_splits(X,␣y,␣groups=X.year)

print(skf)

for␣train_index,␣test_index␣in␣skf.split(X,␣y,␣groups=groups):
␣␣␣print("TRAIN:",␣train_index,␣"TEST:",␣test_index)
␣␣␣X_train,␣X_test␣=␣X.iloc[train_index],␣X.iloc[test_index]
␣␣␣y_train,␣y_test␣=␣y.iloc[train_index],␣y.iloc[test_index]


X_train,␣y_train␣=␣shuffle(X_train,␣y_train,␣random_state=0)
X_test,␣y_test␣=␣shuffle(X_test,␣y_test,␣random_state=0)



#Deneme␣Sample
#X␣=␣X_train.iloc[0:10000,]
#y␣=␣y_train.iloc[0:10000,]


X_train␣=␣X_train.reset_index(drop=True)
X_test␣=␣X_test.reset_index(drop=True)
y_train␣=␣y_train.reset_index(drop=True)
y_test␣=␣y_test.reset_index(drop=True)



#␣Min␣Max␣Scaler
mms=MinMaxScaler()
```

```
#Remove␣cols␣after␣FeatureSelector
n␣=␣list(set(n).intersection(train_removed.columns))+['year']

pd.isnull(X_train[n]).sum()␣>␣0

X_train[n]␣=␣mms.fit_transform(X_train[n])
X_test␣=␣mms.fit_transform(X_test[n])



#Hyperparameters
alpha␣=␣0.003
datasize␣=␣X_train.shape[0]
batch_size␣=␣10000
epoch␣=␣100



##␣Loss␣/w␣Quantile␣Regression

with␣Logger(resloc+'90.DL_training.wFE.txt'):
␣␣␣␣pred␣=␣[]
␣␣␣␣loss_history␣=␣[]
␣␣␣␣quantiles␣=␣[.1,␣.2,␣.4,␣.8␣]
␣␣␣␣for␣quantile␣in␣quantiles:
␣␣␣␣␣␣␣␣#␣Initialize␣session,␣Minibatches␣Gradient␣Descent
␣␣␣␣␣␣␣␣sess␣=␣tf.Session()
␣␣␣␣␣␣␣␣#␣Placeholders
␣␣␣␣␣␣␣␣x_data␣=␣tf.placeholder(shape=[None,␣X_train.shape[1]],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣dtype=tf.float32)
␣␣␣␣␣␣␣␣y_target␣=␣tf.placeholder(shape=[None],␣dtype=tf.float32)
␣␣␣␣␣␣␣␣#␣parameters
␣␣␣␣␣␣␣␣W1␣=␣tf.Variable(tf.random_normal(shape=[X_train.shape[1],␣250]))
␣␣␣␣␣␣␣␣b1␣=␣tf.Variable(tf.random_normal(shape=[250]))
␣␣␣␣␣␣␣␣hidden_1␣=␣tf.nn.tanh(tf.add(tf.matmul(x_data,␣W1),␣b1))

␣␣␣␣␣␣␣␣W2␣=␣tf.Variable(tf.random_normal(shape=[250,␣20]))
␣␣␣␣␣␣␣␣b2␣=␣tf.Variable(tf.random_normal(shape=[20]))
␣␣␣␣␣␣␣␣hidden_2␣=␣tf.nn.relu(tf.add(tf.matmul(hidden_1,␣W2),␣b2))

␣␣␣␣␣␣␣␣W3␣=␣tf.Variable(tf.random_normal(shape=[20,␣4]))
␣␣␣␣␣␣␣␣b3␣=␣tf.Variable(tf.random_normal(shape=[4]))
␣␣␣␣␣␣␣␣hidden_3␣=␣tf.nn.softmax(tf.add(tf.matmul(hidden_2,␣W3),␣b3))

␣␣␣␣␣␣␣␣W4␣=␣tf.Variable(tf.random_normal(shape=[4,␣1]))
␣␣␣␣␣␣␣␣b4␣=␣tf.Variable(tf.random_normal(shape=[1]))
␣␣␣␣␣␣␣␣output␣=␣tf.nn.tanh(tf.add(tf.matmul(hidden_3,␣W4),␣b4))

␣␣␣␣␣␣␣␣#␣loss
␣␣␣␣␣␣␣␣error␣=␣tf.subtract(y_target,␣output)
␣␣␣␣␣␣␣␣loss␣=␣tf.reduce_mean(
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣tf.maximum(quantile␣*␣error,␣(quantile␣-␣1)␣*␣error),␣axis=-1)
␣␣␣␣␣␣␣␣#tf.summary.scalar('loss',␣loss)

␣␣␣␣␣␣␣␣#␣optimize
␣␣␣␣␣␣␣␣optimizer␣=␣tf.train.AdamOptimizer(learning_rate=alpha,␣␣beta1=0.9,
␣␣␣␣␣␣␣␣beta2=0.999,␣epsilon=1e-08,␣use_locking=False,␣name='Adam')
␣␣␣␣␣␣␣␣train_step␣=␣optimizer.minimize(loss)



␣␣␣␣␣␣␣␣#␣Writer␣Definition
␣␣␣␣␣␣␣␣#logfile␣=␣resloc␣+␣'90.DL_Quantile_'+str(quantile)
␣␣␣␣␣␣␣␣#writer␣=␣tf.summary.FileWriter(logfile,␣sess.graph)

␣␣␣␣␣␣␣␣#Initialization␣of␣TF␣Session
␣␣␣␣␣␣␣␣init␣=␣tf.global_variables_initializer()
␣␣␣␣␣␣␣␣sess.run(init)


␣␣␣␣␣␣␣␣for␣e␣in␣range(epoch):
␣␣␣␣␣␣␣␣␣␣␣␣epoch_losses␣=␣[]
```

```
␣␣␣␣␣␣␣␣␣␣␣␣epoch_preds␣=␣[]
␣␣␣␣␣␣␣␣␣␣␣␣for␣i␣in␣range(0,datasize,batch_size):
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣merge␣=␣tf.summary.merge_all()
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣if␣datasize␣==␣batch_size:
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣batch_start_idx␣=␣0
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣elif␣datasize␣<␣batch_size:
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣raise␣ValueError(
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣"datapoint_size:␣%d,␣must␣be␣greater␣than␣batch_size:␣\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣%d"␣%␣(datasize,␣batch_size))
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣else:
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣batch_start_idx␣=␣(i)␣%␣(datasize␣-␣batch_size)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣#batch_start_idx␣=␣(i*␣batch_size)␣%␣(datasize␣-␣batch_size)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣batch_end_idx␣=␣batch_start_idx␣+␣batch_size
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣batch_xs␣=␣X_train.iloc[batch_start_idx:batch_end_idx]
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣batch_ys␣=␣y_train.iloc[batch_start_idx:batch_end_idx]
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣xs␣=␣np.array(batch_xs)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣ys␣=␣np.array(batch_ys)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣if␣i␣%␣10000␣==␣0:
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣preds,result,␣lossf␣=\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣sess.run([output,train_step,␣loss],␣␣\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣feed_dict={x_data:␣xs,␣y_target:␣ys})
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣epoch_losses.append(lossf)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣epoch_preds.append(preds)
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣print("For␣quantile␣{},␣Loss␣function␣is␣{}␣\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣where␣Minibatches␣are␣in␣{}␣and␣{}".\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣format(quantile,lossf[0],␣batch_start_idx,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣batch_end_idx))
␣␣␣␣␣␣␣␣␣␣␣␣epoch_loss␣=␣np.mean(epoch_losses)
␣␣␣␣␣␣␣␣␣␣␣␣loss_history␣=␣np.append(loss_history,␣epoch_loss)
␣␣␣␣␣␣␣␣␣␣␣␣print("For␣quantile␣{},␣Loss␣function␣is␣{}␣in␣EPOCH␣{},␣".\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣format(quantile,␣epoch_loss,e␣+␣1))
␣␣␣␣predictions␣=␣sess.run(output,\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣feed_dict={x_data:X_train,␣y_target:y_train})
␣␣␣␣pred.append(predictions)
␣␣␣␣#pred.append(predict)

␣␣␣␣lossq10=list([loss_history[0:epoch],␣range(1,epoch+1)][0])
␣␣␣␣lossq20=list([loss_history[epoch:epoch*2],␣range(1,epoch+1)][0])
␣␣␣␣lossq40=list([loss_history[epoch*2:epoch*3],␣range(1,epoch+1)][0])
␣␣␣␣lossq80=list([loss_history[epoch*3:epoch*4],␣range(1,epoch+1)][0])


␣␣␣␣AvLossQ10␣=␣sum(lossq10)␣/␣len(lossq10)
␣␣␣␣AvLossQ20␣=␣sum(lossq20)␣/␣len(lossq20)
␣␣␣␣AvLossQ40␣=␣sum(lossq40)␣/␣len(lossq40)
␣␣␣␣AvLossQ80␣=␣sum(lossq80)␣/␣len(lossq80)

␣␣␣␣AvLossQL␣=␣sum(loss_history)␣/␣len(loss_history)
␣␣␣␣print("Quantile␣Regression␣Average␣Q10␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(AvLossQ10))
␣␣␣␣print("Quantile␣Regression␣Q10␣Last␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(lossq10[len(lossq10)-1]))

␣␣␣␣print("Quantile␣Regression␣Average␣Q20␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(AvLossQ20))
␣␣␣␣print("Quantile␣Regression␣Q20␣Last␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(lossq20[len(lossq20)-1]))

␣␣␣␣print("Quantile␣Regression␣Average␣Q40␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(AvLossQ40))
␣␣␣␣print("Quantile␣Regression␣Q40␣Last␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(lossq40[len(lossq40)-1]))

␣␣␣␣print("Quantile␣Regression␣Average␣Q80␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(AvLossQ80))
␣␣␣␣print("Quantile␣Regression␣Q80␣Last␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(lossq80[len(lossq80)-1]))

␣␣␣␣print("Quantile␣Regression␣Average␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(AvLossQL))
␣␣␣␣print("Quantile␣Regression␣Average␣Last␣Loss␣Function␣is␣{0:.8}.".format(\
␣␣␣␣␣␣␣␣␣␣(lossq10[len(lossq10)-1]+␣lossq20[len(lossq20)-1]+␣\
␣␣␣␣␣␣␣␣␣␣lossq40[len(lossq40)-1]+lossq80[len(lossq80)-1])␣/4))
```

```
##␣Loss␣/w␣Normal␣Regression



with␣Logger(resloc+'99.DL_training_normalloss.wFE.txt'):
␣␣␣␣pred␣=␣[]
␣␣␣␣#␣Initialize␣session,␣Minibatches␣Gradient␣Descent
␣␣␣␣sess␣=␣tf.Session()
␣␣␣␣#␣Placeholders
␣␣␣␣x_data␣=␣tf.placeholder(shape=[None,␣X_train.shape[1]],
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣dtype=tf.float32)
␣␣␣␣y_target␣=␣tf.placeholder(shape=[None],␣dtype=tf.float32)
␣␣␣␣#␣parameters
␣␣␣␣W1␣=␣tf.Variable(tf.random_normal(shape=[X_train.shape[1],␣250]))
␣␣␣␣b1␣=␣tf.Variable(tf.random_normal(shape=[250]))
␣␣␣␣hidden_1␣=␣tf.nn.tanh(tf.add(tf.matmul(x_data,␣W1),␣b1))

␣␣␣␣W2␣=␣tf.Variable(tf.random_normal(shape=[250,␣20]))
␣␣␣␣b2␣=␣tf.Variable(tf.random_normal(shape=[20]))
␣␣␣␣hidden_2␣=␣tf.nn.relu(tf.add(tf.matmul(hidden_1,␣W2),␣b2))

␣␣␣␣W3␣=␣tf.Variable(tf.random_normal(shape=[20,␣4]))
␣␣␣␣b3␣=␣tf.Variable(tf.random_normal(shape=[4]))
␣␣␣␣hidden_3␣=␣tf.nn.softmax(tf.add(tf.matmul(hidden_2,␣W3),␣b3))

␣␣␣␣W4␣=␣tf.Variable(tf.random_normal(shape=[4,␣1]))
␣␣␣␣b4␣=␣tf.Variable(tf.random_normal(shape=[1]))
␣␣␣␣output␣=␣tf.nn.tanh(tf.add(tf.matmul(hidden_3,␣W4),␣b4))


␣␣␣␣#␣loss
␣␣␣␣loss␣=␣tf.reduce_mean(tf.square(y_target␣-␣output))
␣␣␣␣#tf.summary.scalar('loss',␣loss)
␣␣␣␣init␣=␣tf.global_variables_initializer()
␣␣␣␣sess.run(init)
␣␣␣␣#␣Declare␣an␣optimizer:␣here␣i␣use␣gradient␣descent
␣␣␣␣my_opt␣=␣tf.train.GradientDescentOptimizer(learning_rate=alpha)

␣␣␣␣#Create␣the␣train␣step
␣␣␣␣train_step␣=␣my_opt.minimize(loss)


␣␣␣␣loss_vec␣=␣[]

␣␣␣␣for␣i␣in␣range(epoch):
␣␣␣␣␣␣␣␣rand_index␣=␣np.random.choice(len(X_train),␣size=batch_size)
␣␣␣␣␣␣␣␣X␣=␣X_train.iloc[rand_index]
␣␣␣␣␣␣␣␣Y␣=␣y_train.iloc[rand_index]
␣␣␣␣␣␣␣␣sess.run(train_step,␣feed_dict={x_data:␣X,␣y_target:Y})
␣␣␣␣␣␣␣␣o␣=␣sess.run(output,␣feed_dict={x_data:␣X,␣y_target:Y})
␣␣␣␣␣␣␣␣temp_loss␣=␣sess.run(loss,␣feed_dict={x_data:␣X,␣y_target:␣Y})
␣␣␣␣␣␣␣␣loss_vec.append(temp_loss)
␣␣␣␣␣␣␣␣pred.append(o)


␣␣␣␣AvLossNL␣=␣sum(loss_vec)␣/␣len(loss_vec)

␣␣␣␣print("Normal␣Regression␣Average␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(AvLossNL))
␣␣␣␣print("Normal␣Regression␣Last␣Loss␣Function␣is␣{0:.8f}.".\
␣␣␣␣␣␣␣␣format(loss_vec[len(loss_vec)-1]))




text_file␣=␣open(resloc+'90.DL_training.wFE.txt',␣"r")
lines␣=␣text_file.readlines()
epocloss=[]
for␣line␣in␣lines:
␣␣␣␣if␣"EPOCH"␣in␣line:
```

```
        line=line.replace('For quantile ', '')
        line = line.replace(' in EPOCH ', '')
        line = line.replace('Loss function is ', '')
        line = line.replace('  \n', '')
        line = line.split(',')
        epocloss.append(line)

epocloss = epocloss.apply(pd.to_numeric, errors='coerce')

epocloss10=pd.DataFrame(epocloss[0:100],
                        columns= ('Quantile', 'Loss', 'Epoch'))
epocloss10.Epoch = list(range(1,101))

epocloss10= epocloss10.apply(pd.to_numeric, errors='coerce')
epocloss20=pd.DataFrame(epocloss[100:200],
                        columns= ('Quantile', 'Loss', 'Epoch'))
epocloss20.Epoch = list(range(1,101))

epocloss20= epocloss20.apply(pd.to_numeric, errors='coerce')

epocloss40=pd.DataFrame(epocloss[200:300],
                        columns= ('Quantile', 'Loss', 'Epoch'))
epocloss40.Epoch = list(range(1,101))
epocloss40= epocloss40.apply(pd.to_numeric, errors='coerce')

epocloss80=pd.DataFrame(epocloss[300:400],
                        columns= ('Quantile', 'Loss', 'Epoch'))
epocloss80.Epoch = list(range(1,101))
epocloss80= epocloss80.apply(pd.to_numeric,
                             errors='coerce')


import seaborn as sns
import matplotlib.pyplot as plt
import pylab

sns.set_style("darkgrid")
plt.figure()
sns10=sns.tsplot(epocloss10['Loss'],epocloss10['Epoch'], )
sns10.set(xlabel='Number of EPOCH', ylabel='QREG Loss (q=0.10)')
pylab.savefig(resloc+'epoch10wFE.pdf')
plt.close()

sns20=sns.tsplot(epocloss20['Loss'],epocloss10['Epoch'], )
sns20.set(xlabel='Number of EPOCH', ylabel='QREG Loss (q=0.20)')
pylab.savefig(resloc+'epoch20wFE.pdf')
plt.close()

sns40=sns.tsplot(epocloss40['Loss'],epocloss10['Epoch'], )
sns40.set(xlabel='Number of EPOCH', ylabel='QREG Loss (q=0.40)')
pylab.savefig(resloc+'epoch40wFE.pdf')
plt.close()

sns80=sns.tsplot(epocloss80['Loss'],epocloss10['Epoch'], )
sns80.set(xlabel='Number of EPOCH', ylabel='QREG Loss (q=0.80)')
pylab.savefig(resloc+'epoch80wFE.pdf')
plt.close()









#R1 Calculations (Pseduo R2)
#rho <- function(u,tau=.5) u*(tau - (u < 0))

pred<-predict(fit1,engel)
```

```
cor(engel$foodexp, pred)^2



def rho(u,tau)):
    return u*(tau-(u[u<0]))




merged

combined_loss = tf.reduce_mean(tf.add_n(losses))
train_step = tf.train.AdamOptimizer().minimize(combined_loss)



#optimize
optimizer = tf.train.GradientDescentOptimizer(learning_rate=alpha)
train_step = optimizer.minimize(loss)




log_file = "/media/DRIVE/Dropbox/deneme.tmp"
writer = tf.summary.FileWriter(log_file, sess.graph_def)



#Initialize session, Minibatches Gradient Descent
loss_history=[]
sess = tf.Session()
init = tf.global_variables_initializer()
sess.run(init)

for e in range(epoch):
    epoch_losses = []
    for i in range(datasize):
        if datasize == batch_size:
            batch_start_idx = 0
        elif datasize < batch_size:
            raise ValueError("datapoint_size: %d, must be greater than \
                              batch_size: %d" % (datapoint_size, batch_size))
        else:
            batch_start_idx = (i * batch_size) % (datasize - batch_size)
        batch_end_idx = batch_start_idx + batch_size
        batch_xs = X.iloc[batch_start_idx:batch_end_idx]
        batch_ys = y.iloc[batch_start_idx:batch_end_idx]
        xs = np.array(batch_xs)
        ys = np.array(batch_ys)
        if i % batch_size == 0:
            result,lossf = sess.run([train_step,combined_loss], \
                                    feed_dict={x_data: xs, y_target: ys})
            epoch_losses.append(lossf)
            #print("Loss function is {} where Minibatches are in {} and {}".
            #format(lossf, batch_start_idx, batch_end_idx))
    epoch_loss = np.mean(epoch_losses)
    loss_history = np.append(loss_history, epoch_loss)
    print("in EPOCH {}, Loss function is {} ".format(e+1,epoch_loss))
```

Appendix D : Chapter 5 Appendices

Table D.1 : InD Chapters N-grams of SMPCM - 1-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	prices, 69	prices, 169	prices, 215	prices, 196	prices, 187	prices, 156	inflation, 141
TOP.2	inflation, 61	inflation, 144	inflation, 170	inflation, 173	inflation, 137	inflation, 142	prices, 139
TOP.3	annual, 32	annual, 84	percent, 107	percent, 111	annual, 107	annual, 103	annual, 97
TOP.4	food, 29	food, 70	food, 105	annual, 82	food, 77	percent, 87	food, 70
TOP.5	increases, 26	services, 59	annual, 94	food, 76	percent, 62	food, 78	percent, 60
Order	2013	2014	2015	2016	2017	2018	
TOP.1	inflation, 175	inflation, 186	inflation, 224	inflation, 237	inflation, 132	inflation, 80	
TOP.2	prices, 129	annual, 119	prices, 172	prices, 141	prices, 82	prices, 58	
TOP.3	annual, 111	prices, 109	annual, 143	annual, 126	annual, 80	annual, 46	
TOP.4	percent, 101	percent, 105	percent, 113	percent, 114	percent, 63	percent, 34	
TOP.5	food, 65	food, 69	food, 85	food, 84	food, 63	food, 31	

Table D.2 : FAI Chapters N-grams of SMPCM - 1-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	demand, 61	demand, 139	quarter, 125	demand, 105	demand, 116	demand, 78	employment, 75
TOP.2	inflation, 31	quarter, 117	demand, 95	quarter, 86	recovery, 66	committee, 46	quarter, 59
TOP.3	increase, 29	growth, 80	growth, 73	production, 63	production, 62	employment, 44	demand, 55
TOP.4	decline, 25	production, 60	production, 69	terms, 58	committee, 59	quarter, 41	domestic, 52
TOP.5	quarter, 24	domestic, 57	domestic, 68	seasonally, 56	quarter, 58	would, 39	economic, 48
Order	2013	2014	2015	2016	2017	2018	
TOP.1	employment, 69	quarter, 103	quarter, 97	quarter, 93	demand, 60	demand, 27	
TOP.2	demand, 67	demand, 101	demand, 96	employment, 73	growth, 51	quarter, 26	
TOP.3	quarter, 57	production, 79	employment, 96	production, 71	quarter, 48	growth, 25	
TOP.4	data, 56	goods, 66	production, 95	demand, 66	economic, 47	activity, 24	
TOP.5	increase, 40	employment, 56	growth, 80	growth, 60	exports, 44	exports, 23	

Table D.3 : MPR Chapters N-grams of SMPCM - 1-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	inflation, 69	inflation, 230	inflation, 190	policy, 119	inflation, 140	policy, 120	inflation, 135
TOP.2	policy, 39	policy, 106	policy, 113	inflation, 95	committee, 132	inflation, 102	policy, 97
TOP.3	risk, 33	outlook, 75	prices, 88	global, 85	policy, 121	committee, 99	committee, 95
TOP.4	medium-term, 31	prices, 73	committee, 82	committee, 81	would, 114	monetary, 65	monetary, 80
TOP.5	monetary, 30	monetary, 70	monetary, 77	monetary, 78	demand, 74	would, 63	demand, 55
Order	2013	2014	2015	2016	2017	2018	
TOP.1	policy, 103	inflation, 179	inflation, 175	inflation, 202	inflation, 156	inflation, 89	
TOP.2	monetary, 94	policy, 96	policy, 127	policy, 129	policy, 88	policy, 58	
TOP.3	inflation, 89	committee, 90	monetary, 111	monetary, 83	monetary, 77	monetary, 55	
TOP.4	committee, 75	monetary, 87	global, 78	global, 82	economic, 69	growth, 44	
TOP.5	fiscal, 66	stance, 70	growth, 69	developments, 80	fiscal, 66	global, 43	

Table D.4 : InD Chapters N-grams of SMPCR - 2-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	foreign demand, 14	seasonally adjusted, 36	seasonally adjusted, 60	seasonally adjusted, 56	foreign demand, 37	domestic demand, 31	economic activity, 43
TOP.2	domestic demand, 13	industrial production, 35	industrial production, 29	adjusted terms, 50	domestic demand, 32	economic activity, 22	domestic demand, 30
TOP.3	third quarter, 12	domestic demand, 31	first quarter, 27	second quarter, 34	pre-crisis levels, 30	external demand, 21	current account, 19
TOP.4	manufacturing industry, 11	economic activity, 29	adjusted terms, 27	foreign demand, 26	economic activity, 27	leading indicators, 18	first quarter, 15
TOP.5	strong course, 11	third quarter, 28	domestic demand, 25	economic activity, 26	seasonally adjusted, 26	unemployment rates, 16	non-farm employment, 15

Order	2013	2014	2015	2016	2017	2018	
TOP.1	current account, 34	domestic demand, 31	industrial production, 43	current account, 29	economic activity, 34	economic activity, 18	
TOP.2	domestic demand, 29	second quarter, 23	domestic demand, 37	domestic demand, 27	domestic demand, 27	domestic demand, 14	
TOP.3	non-farm employment, 22	economic activity, 23	external demand, 25	economic activity, 26	current account, 20	fourth quarter, 11	

Table D.4 (Continued) : InD Chapters N-grams of SMPCR - 2-grams.

Order	2013	2014	2015	2016	2017	2018	
TOP.4	account deficit, 18	industrial production, 19	first quarter, 22	first quarter, 25	exports goods, 16	first quarter, 9	
TOP.5	industrial production, 16	first quarter, 17	employment growth, 22	third quarter, 25	labor market, 15	data indicate, 8	

Table D.5 : FAI Chapters N-grams of SMPCM - 2-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	annual inflation, 18	annual inflation, 57	annual inflation, 57	food prices, 39	annual inflation, 43	annual inflation, 44	annual inflation, 70
TOP.2	unprocessed food, 18	food prices, 38	food prices, 47	annual inflation, 37	food prices, 39	food prices, 30	food prices, 31
TOP.3	exchange rates, 12	unprocessed food, 35	processed food, 38	percent year-on-year, 28	percentage points, 34	unprocessed food, 29	core goods, 31
TOP.4	food prices, 10	services inflation, 27	energy prices, 29	unprocessed food, 28	unprocessed food, 29	core goods, 26	unprocessed food, 27
TOP.5	food products, 9	inflation rate, 17	unprocessed food, 27	rate increase, 28	meat prices, 27	processed food, 21	processed food, 26

Order	2013	2014	2015	2016	2017	2018	
TOP.1	annual inflation, 68	annual inflation, 66	annual inflation, 81	annual inflation, 76	annual inflation, 38	annual inflation, 26	
TOP.2	inflation indicators, 27	core goods, 44	core goods, 47	core goods, 41	core goods, 31	core goods, 15	
TOP.3	core inflation, 27	food prices, 26	food prices, 36	unprocessed food, 34	food prices, 22	food inflation, 12	
TOP.4	unprocessed food, 27	goods inflation, 24	underlying trend, 30	services inflation, 32	unprocessed food, 21	percent annual, 10	

Table D.5 (Continued) : FAI Chapters N-grams of SMPCR - 2-grams.

Order	2013	2014	2015	2016	2017	2018	
TOP.5	core goods, 26	services inflation, 22	goods inflation, 25	goods inflation, 30	percent annual, 19	processed food, 9	

Table D.6 : MPR Chapters N-grams of SMPCM - 2-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	medium-term inflation, 20	inflation outlook, 54	monetary policy, 59	monetary policy, 67	monetary policy, 55	monetary policy, 49	medium term, 38
TOP.2	monetary policy, 20	medium term, 48	medium term, 42	economic activity, 38	economic activity, 36	price stability, 34	financial stability, 34
TOP.3	inflation expectations, 15	monetary policy, 44	structural reforms, 39	inflation report, 24	price stability, 34	measures taken, 27	inflation outlook, 30
TOP.4	central bank, 14	inflation expectations, 39	inflation outlook, 38	committee members, 23	committee members, 30	domestic demand, 27	domestic demand, 28
TOP.5	inflation outlook, 13	structural reforms, 33	energy prices, 33	fiscal discipline, 22	domestic demand, 29	medium term, 27	

Order	2013	2014	2015	2016	2017	2018	
TOP.1	monetary policy, 65	monetary policy, 56	monetary policy, 89	monetary policy, 69	monetary policy, 47	monetary policy, 33	
TOP.2	inflation outlook, 37	inflation outlook, 50	inflation outlook, 56	policy stance, 65	economic activity, 46	inflation outlook, 16	
TOP.3	policy stance, 33	policy stance, 42	policy stance, 54	inflation outlook, 51	inflation outlook, 38	economic activity, 16	
TOP.4	medium term, 33	tight monetary, 27	current account, 33	fiscal discipline, 36	fiscal policy, 34	fiscal policy, 16	

Table D.6 (Continued) : MPR Chapters N-grams of SMPCR - 2-grams.

Order	2013	2014	2015	2016	2017	2018	
TOP.5	fiscal discipline, 24	medium term, 27	fiscal discipline, 31	upcoming period, 32	policy stance, 28	global growth, 15	

Table D.7 : InD Chapters N-grams of SMPCR - 3-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	slowdown total demand, 6	last quarter 2006, 14	seasonally adjusted terms, 27	seasonally adjusted terms, 50	private consumption demand, 14	capacity utilization rates, 8	current account deficit, 14
TOP.2	seasonally adjusted data, 5	private consumption demand, 13	seasonally adjusted figures, 11	demand remains weak, 11	capacity utilization rates, 13	recent data releases, 7	orders manufacturing industry, 10
TOP.3	business tendency survey, 5	seasonally adjusted terms, 12	imports capital goods, 11	recent data releases, 9	seasonally adjusted terms, 13	unit labor costs, 7	last quarter 2011, 9
TOP.4	net foreign demand, 5	aggregate demand conditions, 9	domestic sales automobiles, 9	recovery economic activity, 8	utilization rates return, 11	conditions continue improve, 7	slowdown domestic demand, 9
TOP.5	second half year, 5	domestic sales automobiles, 8	heavy commercial vehicles, 9	foreign demand remains, 8	return pre-crisis levels, 11	employment conditions continue, 7	restrain employment growth, 8

Order	2013	2014	2015	2016	2017	2018	
TOP.1	current account deficit, 18	industrial production index, 10	current account balance, 20	current account balance, 21	current account deficit, 12	recently released data, 7	

Table D.7 (Continued) : InD Chapters N-grams of SMPCR - 3-grams.

Order	2013	2014	2015	2016	2017	2018	
TOP.2	final domestic demand, 15	first quarter 2014, 9	improvement current account, 12	growing eu demand, 11	recovery economic activity, 8	indicate economic activity, 7	
TOP.3	current account balance, 15	seasonally adjusted terms, 9	data expenditure side, 12	moderate course consumer, 10	increases exports goods, 8	released data indicate, 7	
TOP.4	trade current account, 11	final domestic demand, 9	moderate course consumer, 9	course consumer loans, 10	released data indicate, 7	data indicate economic, 7	
TOP.5	foreign trade current, 10	growth upcoming period, 8	course consumer loans, 9	eu demand continues, 10	recently released data, 7	activity maintains strength, 6	

Table D.8 : FAI Chapters N-grams of SMPCR - 3-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	energy unprocessed food, 7	unprocessed food prices, 20	processed food prices, 23	unprocessed food prices, 20	annual rate increase, 21	unprocessed food prices, 20	unprocessed food prices, 20
TOP.2	unprocessed food products, 6	annual inflation rate, 15	unprocessed food prices, 15	annual rate increase, 15	unprocessed food prices, 21	core inflation indicators, 11	core inflation indicators, 15
TOP.3	unprocessed food prices, 6	monetary policy committee, 11	food energy prices, 12	monetary policy committee, 13	core inflation indicators, 16	annual services inflation, 11	front annual inflation, 14
TOP.4	alcoholic beverages tobacco, 6	annual services inflation, 10	monetary policy committee, 11	prices durable goods, 13	monetary policy committee, 11	annual rate increase, 11	percent annual inflation, 14
TOP.5	tobacco products gold, 6	annual rate increase, 9	goods excluding energy, 11	processed food prices, 13	policy committee committee, 11	percent annual inflation, 11	core goods group, 12

Order	2013	2014	2015	2016	2017	2018	
TOP.1	core inflation indicators, 27	core goods inflation, 23	core goods inflation, 25	core goods inflation, 30	core goods inflation, 16	percent annual inflation, 9	
TOP.2	unprocessed food prices, 15	depreciation turkish lira, 16	depreciation turkish lira, 15	unprocessed food prices, 23	unprocessed food prices, 12	core goods inflation, 7	

Table D.8 (Continued) : FAI Chapters N-grams of SMPCR - 3-grams.

Order	2013	2014	2015	2016	2017	2018	
TOP.3	percent annual inflation, 13	group annual inflation, 16	underlying trend core, 15	annual core goods, 18	annual core goods, 10	inflation inflation expectations, 5	
TOP.4	inflation core goods, 12	annual core goods, 13	annual food inflation, 14	annual inflation food, 13	percent annual inflation, 10	levels inflation inflation, 5	
TOP.5	annual energy inflation, 11	core inflation indicators, 12	unprocessed food prices, 14	annual services inflation, 12	annual food inflation, 9	annual core goods, 5	

Table D.9 : MPR Chapters N-grams of SMPCR - 3-grams.

Order	2006	2007	2008	2009	2010	2011	2012
TOP.1	medium-term inflation expectations, 12	medium term inflation, 21	food energy prices, 26	monetary policy decisions, 14	medium term program, 19	regarding global economy, 17	medium term program, 21
TOP.2	medium-term inflation outlook, 6	medium-term inflation outlook, 13	medium term inflation, 19	union accession process, 12	levels long period, 15	flexibility monetary policy, 13	flexibility monetary policy, 21
TOP.3	union accession process, 6	risk inflation outlook, 13	second round effects, 13	european union accession, 12	low levels long, 15	sustaining fiscal discipline, 13	regarding global economy, 18
TOP.4	implementation structural reforms, 6	union accession process, 12	union accession process, 12	economy global crisis, 12	envisaged medium term, 13	domestic external demand, 13	monetary policy stance, 17
TOP.5	hesitate tighten monetary, 6	reforms envisaged economic, 12	implementation structural reforms, 12	prudent monetary policy, 12	price stability respect, 12	monetary policy continue, 12	stability financial stability, 16

Order	2013	2014	2015	2016	2017	2018	
TOP.1	monetary policy stance, 30	monetary policy stance, 41	monetary policy stance, 54	monetary policy stance, 55	monetary policy stance, 24	monetary policy stance, 11	

Table D.9 (Continued) : MPR Chapters N-grams of SMPCR - 3-grams.

Order	2013	2014	2015	2016	2017	2018	
TOP.2	medium term program, 24	tight monetary policy, 21	current account balance, 27	current account balance, 19	monitored closely regard, 14	risks pricing behavior, 7	
TOP.3	outlined medium term, 16	expectations pricing behavior, 14	improvement current account, 19	tight monetary policy, 19	expectations pricing behavior, 12	coordination monetary fiscal, 6	
TOP.4	deviate significantly framework, 12	improvement inflation outlook, 14	tight monetary policy, 17	expectations pricing behavior, 14	inflation expectations pricing, 11	major central banks, 6	
TOP.5	securities persistently low, 12	effects inflation outlook, 13	cautious monetary policy, 13	economy external shocks, 14	closely regard effects, 9	global growth outlook, 6	

4.1 Python Web Crawler for SMPCM Links

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jul  7 12:22:55 2018

#######################################################################
#                          _          _             _           _     #
#     _    _    __ _   ____ (_)  _ __   | | __ _    _ | |_   _    _ | | __     #
#    | | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    < | |_| || |_ | |_| ||    <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                                _  _                        #
#    / __ \    __ _  _ __ ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_   _ \   / _  || || |    / __|/ _ \ | '_    _ \   #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com ######################################
#######################################################################
"""

#Turkish CentralBank news and ppk crawler


import urllib, urllib3, math, json, re, requests, time, random, shutil
from lxml import html
from bs4 import BeautifulSoup as bs
import pandas as pd



path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'


#Mimicing Browser
hdr = {"Host": "www.google.com",
"User-Agent": "Mozilla/5.0␣(Windows␣NT␣10.0;␣WOW64;␣rv:47.0)␣Gecko/20100101␣\
Firefox/47.0",
"Accept": "*/*",
"Accept-Language": "tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3",
"Content-Type":"text/html;␣charset=utf-8",
"Accept-Encoding": "identity",
"X-Requested-With": "XMLHttpRequest",
"Referer": "www.google.com",
"Cookie": "__utma=45412152.1424915812.1465638666.1467996321.1468056341.11;␣\
__utmz=45412152.1465638666.1.1.utmcsr=google|utmccn=(organic)|utmcmd=organic|\
utmctr=(not%20provided);␣__auc=75312a441553edf3e92f4d51d8c;␣\
__gfp_64b=tltrSz7UZyUj2wJ29_zcTNbDRlwxCDxRj5w13WFiRmD._7;␣\
facebooklike=fbmodalviewed;␣s_nr=1467985988780-Repeat;␣OPTAW_gaCookie=GA1.2.\
1424915812.1465638666;␣ASPSESSIONIDSQTSCDTS=CKCAMKODJBLOCCGKGMHEIEEM;␣\
__utmc=45412152;␣s_cc=true;␣s_ppv=-%2C100%2C100%2C947;␣s_sq=%5B%5BB%5D%5D;␣\
__asc=5edba925155cefa07b1b07e9797;␣__utmb=45412152.2.10.1468056341;␣__utmt=1",
"Connection": "keep-alive"}


#Bulk Process
url0='http://www.tcmb.gov.tr'
url1='http://www.tcmb.gov.tr/wps/wcm/connect/EN/TCMB+EN/Main+Menu/\
Announcements/Press+Releases/'


yillar=range(2006,2014,1)
yillar=[str(i) for i in yillar]
#last='2000 ve Oncesi'
#yillar.insert(0,last)
```

```
urllist=[]
for yil in yillar:
    urllist.append(url1+yil)



http = urllib3.PoolManager()


#2013 and before
urldata=[]
urltext=[]
for url in urllist:
    try:
        print(url + '␣indiriliyor')
        time.sleep(random.randint(1, 10))
        response = http.request('GET', url, headers=hdr)
        soup = bs(response.data, 'lxml')
        k1 = soup.findAll('a', attrs={'class': 'collection-title' ,
                                      'href': re.compile('/wps/')})
        for k in k1:
            urldata.append(url0+k.get('href'))
            urltext.append(k.text)
    except:
        continue
    #k2 = soup.findAll('div',attrs={'class':'collection-tag'})
    #for j in k2:
        #urltext.append(j.text)



df=pd.DataFrame({'text':urltext, 'url':urldata})
dataname='01.tcmb.news.eng.2006-2013'
df.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)



#2014 and after
yillar=range(2014,2019,1)
yillar=[str(i) for i in yillar]
#last='2000 ve Oncesi'
#yillar.insert(0,last)

urllist=[]
for yil in yillar:
    urllist.append(url1+yil)



http = urllib3.PoolManager()

urldata=[]
urltext=[]
urlpdf=[]
for url in urllist:
    try:
        print(url + '␣indiriliyor')
        time.sleep(random.randint(1, 10))
        response = http.request('GET', url, headers=hdr)
        soup = bs(response.data, 'lxml')
        k1 = soup.findAll('a', attrs={'class': 'collection-title' ,
                                      'href': re.compile('/wps/')})
        for k in k1:
            response = http.request('GET', url0+k.get('href'), headers=hdr)
            soup = bs(response.data, 'lxml')
            l1 = soup.findAll('a', attrs={'class': 'pdf␣type-link'})
            for l in l1:
                urldata.append(url0+l.get('href'))
                urltext.append(l.text.replace('\n',''))
␣␣␣␣except:
␣␣␣␣␣␣␣␣continue
```

```
    #k2 = soup.findAll('div',attrs={'class':'collection-tag'})
    #for j in k2:
        #urltext.append(j.text)


df=pd.DataFrame({'text':urltext, 'url':urldata})
dataname='01.tcmb.news.eng.2014-2018'
df.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

4.2 Python Web Crawler for SMPCM PDFs

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jul  7 12:22:55 2018

#####################################################################
#                          _          _              _          _    #
#     _    _    __ _   ___ (_) _ __   | | __ _    _ | |_   _    _ | | __     #
#    | |  | |  / _  |/ __|| || '_ \ | |/ /| |  | || __|| |  | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    < | |_| || |_ | |_| ||    <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                                 _  _                       #
#    / __ \    __ _   _ __ ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_    _ \   / _  || || |    / __|/ _ \ | '_    _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                    #
#####################################################################
#@author: Yasin KÜTÜK           ######################################
#@web    : yasinkutuk.com       ######################################
#@email : yasinkutuk@gmail.com ######################################
#####################################################################
"""

#Turkish CentralBank news and ppk crawler


import urllib, urllib3, math, json, re, requests, time, random, shutil
from lxml import html
from bs4 import BeautifulSoup as bs
import pandas as pd



path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'


#Mimicing Browser
hdr = {"Host": "www.google.com",
"User-Agent": "Mozilla/5.0␣(Windows␣NT␣10.0;␣WOW64;␣rv:47.0)␣Gecko/20100101␣\
Firefox/47.0",
"Accept": "*/*",
"Accept-Language": "tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3",
"Content-Type":"text/html;␣charset=utf-8",
"Accept-Encoding": "identity",
"X-Requested-With": "XMLHttpRequest",
"Referer": "www.google.com",
"Cookie": "__utma=45412152.1424915812.1465638666.1467996321.1468056341.11;␣\
__utmz=45412152.1465638666.1.1.utmcsr=google|utmccn=(organic)|utmcmd=organic|\
utmctr=(not%20provided);␣___auc=75312a441553edf3e92f4d51d8c;␣\
__gfp_64b=tltrSz7UZyUj2wJ29_zcTNbDRlwxCDxRj5w13WFiRmD._7;␣\
facebooklike=fbmodalviewed;␣s_nr=1467985988780-Repeat;␣OPTAW_gaCookie=GA1.2.\
1424915812.1465638666;␣ASPSESSIONIDSQTSCDTS=CKCAMKODJBLOCCGKGMHEIEEM;␣\
__utmc=45412152;␣s_cc=true;␣s_ppv=-%2C100%2C100%2C947;␣s_sq=%5B%5BB%5D%5D;␣\
__asc=5edba925155cefa07b1b07e9797;␣___utmb=45412152.2.10.1468056341;␣___utmt=1",
"Connection": "keep-alive"}


#Bulk Process
url0='http://www.tcmb.gov.tr'
url1='http://www.tcmb.gov.tr/wps/wcm/connect/EN/TCMB+EN/Main+Menu/\
Announcements/Press+Releases/'



df=pd.read_csv(path+'01.tcmb.news.eng.2006-2013.csv', sep=';',
               header='infer')
df=df.append(pd.read_csv(path+'01.tcmb.news.eng.2014-2018.csv',
                         sep=';', header='infer'),  ignore_index=True)
```

```
urldata = []
urltext = []
#filter only PPK
for k in range(len(df)):
    if('Monetary␣Policy␣Committee' in df.text[k]):
        urldata.append(df.url[k])
        urltext.append(df.text[k])


#urldata= df['url']
#urltext= df['text']

#There are two types of links in CB, one is standard text, other is pdf.
#https://stackoverflow.com/questions/24844729/download-pdf-using-urllib
for i in range(len(urldata)):
    if ('pdf' in urldata[i]):
        try:
            print(urldata[i]+ '␣linki␣indiriliyor')
            pdfin = urldata[i]
            urlname=pdfin.split('/')
            pdfname=urlname[len(urlname)-1]
            pdfname=re.search(r'(.*).pdf?', pdfname).group(0)
            if ('Monetary␣Policy␣Committee' in urltext[i]):
                output= ('/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/\
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Analysis/ReportsENG/PPK/'+ pdfname)
                with urllib.request.urlopen(pdfin) as response, \
                open(output, 'wb') as out_file:
                    shutil.copyfileobj(response, out_file)
            else:
                output= (\
'/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/ReportsENG/'+ pdfname)
                with urllib.request.urlopen(pdfin) as response, \
                open(output, 'wb') as out_file:
                    shutil.copyfileobj(response, out_file)
        except:
            continue
        time.sleep(random.randint(1,10))



pdfin = urldata[10]
urlname=pdfin.split('/')
pdfname=urlname[len(urlname)-1]
pdfname=re.search(r'(.*).pdf?', pdfname).group(0)
output= ('/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/\
␣␣␣␣␣␣␣␣␣ReportsENG/'+ pdfname)
with urllib.request.urlopen(pdfin) as response, open(output, 'wb') as out_file:
    shutil.copyfileobj(response, out_file)
```

4.3 Python PDF to Text Extractor

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jul  7 12:22:55 2018

######################################################################
#                          _          _              _          _    #
#     _   _   __ _  ___ (_) _ __   | | __ _   _ | |_  _   _ | | __    #
#    | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                          #
#     ____                              _  _                         #
#    / __ \   __ _  _ __ ___     __ _ (_)| |    ___  ___    _ __ ___   #
#   / / _  | / _  || '_   _ \   / _  || || |   / __|/ _ \ | '_   _ \  #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                     #
######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
######################################################################
"""

#Turkish CentralBank news and ppk crawler


import urllib, urllib3, math, json, re, requests, time, random, shutil, os
from lxml import html
from bs4 import BeautifulSoup as bs
import pandas as pd
import seaborn as sns
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
from io import StringIO



path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'


#Mimicing Browser
hdr = {"Host": "www.google.com",
"User-Agent": "Mozilla/5.0␣(Windows␣NT␣10.0;␣WOW64;␣rv:47.0)␣Gecko/20100101␣\
Firefox/47.0",
"Accept": "*/*",
"Accept-Language": "tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3",
"Content-Type":"text/html;␣charset=utf-8",
"Accept-Encoding": "identity",
"X-Requested-With": "XMLHttpRequest",
"Referer": "www.google.com",
"Cookie": "__utma=45412152.1424915812.1465638666.1467996321.1468056341.11;␣\
__utmz=45412152.1465638666.1.1.utmcsr=google|utmccn=(organic)|utmcmd=organic|\
utmctr=(not%20provided);␣___auc=75312a441553edf3e92f4d51d8c;␣\
__gfp_64b=tltrSz7UZyUj2wJ29_zcTNbDRlwxCDxRj5w13WFiRmD._7;␣\
facebooklike=fbmodalviewed;␣s_nr=1467985988780-Repeat;␣OPTAW_gaCookie=GA1.2.\
1424915812.1465638666;␣ASPSESSIONIDSQTSCDTS=CKCAMKODJBLOCCGKGMHEIEEM;␣\
__utmc=45412152;␣s_cc=true;␣s_ppv=-%2C100%2C100%2C947;␣s_sq=%5B%5BB%5D%5D;␣\
__asc=5edba925155cefa07b1b07e9797;␣___utmb=45412152.2.10.1468056341;␣___utmt=1",
"Connection": "keep-alive"}


#Bulk Process
url0='http://www.tcmb.gov.tr'
url1='http://www.tcmb.gov.tr/wps/wcm/connect/TR/TCMB+TR/Main+Menu/Duyurular/\
Basin/'
```

```
df=pd.read_csv(path+'02.report_meeting_dates.csv', sep=';', header='infer')

textloc = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/\
ReportsENG/TEXT/'
ppkex = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/\
ReportsENG/PPKEX/'


# Klasördeki dosyalar
p = sns.color_palette()
print('# File sizes')
for f in os.listdir(textloc):
    if 'zip' not in f:
        print(f.ljust(30) +
              str(round(os.path.getsize(textloc + f) / 1000000, 2)) + 'MB')


files = os.listdir(textloc)




no = []
inflation = []
factorsinf = []
monetary = []
for i in range(0,len(files)):
    mno=files[i].replace('ANO','').replace('.txt','')
    no.append(mno)

    tex = open(textloc+files[i],'r')
    tex = tex.read()
    tex = tex.replace('\n\n\x0c',' ')
    tex = tex.replace('\n',' ')
    if(('Inflation Developments' in tex)==True and \
       ('Factors Affecting Inflation' in tex)==True):
        inflat = re.search('Inflation Developments(.*)\
                            Factors Affecting Inflation', tex)
        inflat1 = inflat.group(1)
    elif(('Inflation Developments' in tex)==True and \
         ('Outlook and Factors Affecting Inflation' in tex)==True):
        inflat = re.search('Inflation Developments(.*)\
                            Outlook and Factors Affecting Inflation', tex)
        inflat1 = inflat.group(1)
    inflation.append(inflat1)

    if(('Factors Affecting Inflation' in tex)==True and \
       ('Risks' in tex)==True):
        factors = re.search('Factors Affecting Inflation(.*)Risks', tex)
        factors1 = factors.group(1)
    if(('Factors Affecting Inflation' in tex)==True and \
       ('Risks and Monetary Policy' in tex)==True):
        factors = re.search('Factors Affecting Inflation\
                             (.*)Risks and Monetary Policy', tex)
        factors1 = factors.group(1)
    if(('Factors Affecting Inflation' in tex)==True and \
       ('Risks and the Monetary Policy' in tex)==True):
        factors = re.search('Factors Affecting Inflation(.*)\
                             Risks and the Monetary Policy', tex)
        factors1 = factors.group(1)
    if(('Factors Affecting Inflation' in tex)==True and \
       ('Monetary Policy and Risks' in tex)==True):
        factors = re.search('Factors Affecting Inflation(.*)\
                             Monetary Policy and Risks', tex)
        factors1 = factors.group(1)
    factorsinf.append(factors1)

    if(('Risks' in tex)==True):
        monet = re.search('Risks (.*)', tex)
        monet1 = monet.group(1)
    if(('Risks and Monetary Policy' in tex)==True):
```

```
        monet = re.search('Risks and Monetary Policy (.*)', tex)
        monet1 = monet.group(1)
    if(('Monetary Policy and Risks' in tex)==True):
        monet = re.search('Monetary Policy and Risks (.*)', tex)
        monet1 = monet.group(1)
    if(('Risks and the Monetary Policy' in tex)==True):
        monet = re.search('Risks and the Monetary Policy (.*)', tex)
        monet1 = monet.group(1)
    monetary.append(monet1)



df = pd.DataFrame(df)

df2 = pd.DataFrame({'No':no})
df2['InflationDevelopments'] = inflation
df2['FactorsAffectingInflation'] = factorsinf
df2['MonetaryPolicyandRisks'] = monetary

df3 = pd.merge(df, df2, on='No')

dataname='02.tcmb.smpcm'
df3.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

4.4 Python Word Cloud Generator

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 25 21:30:14 2018

#######################################################################
#                          _          _              _          _     #
#    _   _   __ _  ____ (_) _ __   | | __ _   _ | |_  _   _ | | __    #
#   | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#   | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#    \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#    |___/                                                            #
#    _____                                 _  _                       #
#   / __ \    __ _  _ __ ____     __ _ (_)| |    ___  ____   _ __ ___   #
#  / / _  |  / _  || '_   _ \  / _  || || |   / __|/ _ \ | '_   _ \  #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
#######################################################################
"""

import sys,re
from os import path
import numpy as np
from PIL import Image
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
from collections import Iterable
from itertools import chain
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
from ggplot import *

path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'

results = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/Results/'

df=pd.read_csv(path+'02.tcmb.smpcm.csv', sep=';', header='infer')



flattened  = [i for i in df.FactorsAffectingInflation]
flatten = flattened.pop()


# create numpy araay for wordcloud mask image
mask = np.array(Image.open(results + 'cloud.png'))

# create set of stopwords
stopwords = set(STOPWORDS)

# create wordcloud object
wc = WordCloud(background_color="white",        max_words=200, mask=mask,
               pwords=stopwords)

# generate wordcloud
wc.generate(flatten)

# save wordcloud
wc.to_file(results+ "01fai.png")



flattened  = [i for i in df.InflationDevelopments]
```

```
flatten = flattened.pop()


# create numpy araay for wordcloud mask image
mask = np.array(Image.open(results + 'cloud.png'))

# create set of stopwords
stopwords = set(STOPWORDS)

# create wordcloud object
wc = WordCloud(background_color="white",        max_words=200, mask=mask,
               pwords=stopwords)

# generate wordcloud
wc.generate(flatten)

# save wordcloud
wc.to_file(results+ "02ind.png")



flattened  = [i for i in df.MonetaryPolicyandRisks]
flatten = flattened.pop()


# create numpy araay for wordcloud mask image
mask = np.array(Image.open(results + 'cloud.png'))

# create set of stopwords
stopwords = set(STOPWORDS)

# create wordcloud object
wc = WordCloud(background_color="white",        max_words=200, mask=mask,
               pwords=stopwords)

# generate wordcloud
wc.generate(flatten)

# save wordcloud
wc.to_file(results+ "03mpr.png")



#Find maximum item number in texts
numbers=[]
for i in range(len(df)):
    numb1 = re.findall('\d+[.]␣', df.InflationDevelopments[i])
    numb1 = [x.replace('␣','')  for x in numb1]
    numb1 = [int(x.replace('.',''))  for x in numb1]
    numb1 = [x for x in numb1 if x < 50]
    numb2 = re.findall('\d+[.]␣', df.FactorsAffectingInflation[i])
    numb2 = [x.replace('␣','')  for x in numb2]
    numb2 = [int(x.replace('.',''))  for x in numb2]
    numb2 = [x for x in numb2 if x < 50]
    numb3 = re.findall('\d+[.]␣', df.MonetaryPolicyandRisks[i])
    numb3 = [x.replace('␣','')  for x in numb3]
    numb3 = [int(x.replace('.',''))  for x in numb3]
    numb3 = [x for x in numb3 if x < 50]
    number=numb1+numb2+numb3
    number2=[number[0]]
    #Robustness check, if not sequantial, drop it!
    for l in range(1,len(number)):
        if(number[l-1]==(number[l]-1)):
            number2.append(number[l])
    numbers.append(number2)




df2 = pd.DataFrame({'No':df.No ,'ItemNos':numbers})
df2 = df2.sort_values(['No'])
```

```
dataname='03.tcmb.smpcm.itemno'
df2.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)



#Plotting number of items  in R
items=[]
for k in range(len(df2)):
    items.append(max(df2.ItemNos[k]))

df3=pd.DataFrame({'No':df.No ,'Items':items})
df3 = df3.sort_values(['No'])
df3=df3.reset_index(drop=True)

dataname='04.tcmb.itemnumbers'
df3.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

4.5 Python Word Counter

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 25 21:30:14 2018

#######################################################################
#                          _          _              _          _     #
#     _    _    __ _   ___ (_) _ __   | | __ _    _ | |_   _    _ | | __     #
#    | |  | |  / _  |/ __|| || '_ \  | |/ /| |  | || __|| |  | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||    < | |_| || |_ | |_| ||    <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                                 _  _                       #
#    / __ \    __ _   _ __ ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_    _ \   / _  || || |    / __|/ _ \  | '_    _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                  #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
#######################################################################
"""

import sys,re
from os import path
import numpy as np
from PIL import Image
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
from collections import Iterable
from itertools import chain
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns


path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'

results = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/Results/'

df=pd.read_csv(path+'02.tcmb.smpcm.csv', sep=';', header='infer')

factors=[]
factorcount=[]
for i in range(len(df.FactorsAffectingInflation)):
    d=df.FactorsAffectingInflation[i].split()
    factors.append(d)
    factorcount.append(len(d))


infla=[]
inflacount=[]
for i in range(len(df.InflationDevelopments)):
    d=df.InflationDevelopments[i].split()
    infla.append(d)
    inflacount.append(len(d))

monet=[]
monetcount=[]
for i in range(len(df.MonetaryPolicyandRisks)):
    d=df.MonetaryPolicyandRisks[i].split()
    monet.append(d)
    monetcount.append(len(d))



df['InflaWords'] = infla
```

```
df['InflaCounts'] = inflacount
df['FactorWords'] = factors
df['FactorCounts'] = factorcount
df['MonetaryWords'] = monet
df['MonetaryCounts'] = monetcount

df=(df.sort_values('No', ascending=True)).reset_index(drop=True)

dinfla=[0]
dinflacount=[0]
dfactors=[0]
dfactorscount=[0]
dmonet=[0]
dmonetcount=[0]
for m in range(1,len(infla)):
    words = list(set(infla[m]) - set(infla[m-1]) )
    counts = len(words)
    dinfla.append(words)
    dinflacount.append(counts)

for m in range(1,len(factors)):
    words = list(set(factors[m]) - set(factors[m-1]) )
    counts = len(words)
    dfactors.append(words)
    dfactorscount.append(counts)

for m in range(1,len(monet)):
    words = list(set(monet[m]) - set(monet[m-1]) )
    counts = len(words)
    dmonet.append(words)
    dmonetcount.append(counts)

df['dInflaWords'] = dinfla
df['dInflaCounts'] = dinflacount
df['dFactorWords'] = dfactors
df['dFactorCounts'] = dfactorscount
df['dMonetaryWords'] = dmonet
df['dMonetaryCounts'] = dmonetcount

dataname='05.tcmb.wordcounts'
df.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

4.6 Python SMPCM Item Reader

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Aug  9 16:00:44 2018

#######################################################################
#                          _          _              _          _     #
#    _   _   __ _  ___ (_) _ __   | | __ _    _ | |_  _   _ | | __    #
#   | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /    #
#   | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <     #
#    \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\    #
#    |___/                                                          #
#    _____                                 _  _                     #
#   / __ \   __ _  _ __ ___     __ _ (_)| |    ___  ___    _ __ ___   #
#  / / _  | / _  || '_   _ \   / _  || || |   / __|/ _ \ | '_   _ \  #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                     #
#######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
#######################################################################
"""



import sys, re, math, collections
from os import path
import numpy as np
from PIL import Image
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
from collections import Iterable
from itertools import chain
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
from nltk.tokenize import sent_tokenize as st



path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'

results = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/Results/'

df = pd.read_csv(path+'02.tcmb.smpcm.csv', sep=';', header='infer')
df = df.sort_values(['No'], ascending = True)
df = df.reset_index(drop=True)

indlist=[]
for i in range(0,len(df.InflationDevelopments)):
    ind = st(df.InflationDevelopments[i])
    for l in ind:
        if len(l)==1:
            ind.remove(l)
    orders = []
    inddict = {}
    for j in range(len(ind)):
        if(len(ind[j])<4):
            orders.append(j)
    for k in range(len(orders)):
        if (k+1< len(orders)):
            sira = str(ind[orders[k]]).replace('␣','').replace('.','')
            inddict[sira] = ind[orders[k]+1:orders[k+1]]
        else:
            inddict[str(int(sira)+1)]  = ind[orders[k]+1:]
    indlist.append(inddict)
```

```
failist=[]
for i in range(0,len(df.FactorsAffectingInflation)):
    ind = st(df.FactorsAffectingInflation[i])
    for l in ind:
        if len(l)==1:
            ind.remove(l)
    orders = []
    inddict = {}
    for j in range(len(ind)):
        if(len(ind[j])<4):
            orders.append(j)
    for k in range(len(orders)):
        if (k+1< len(orders)):
            sira = str(ind[orders[k]]).replace('␣','').replace('.','')
            inddict[sira] = ind[orders[k]+1:orders[k+1]]
        else:
            inddict[str(int(sira)+1)]  = ind[orders[k]+1:]
    failist.append(inddict)


mprlist=[]
for i in range(0,len(df.MonetaryPolicyandRisks)):
    ind = st(df.MonetaryPolicyandRisks[i])
    for l in ind:
        if len(l)==1:
            ind.remove(l)
    orders = []
    inddict = {}
    for j in range(len(ind)):
        if(len(ind[j])<4):
            orders.append(j)
    for k in range(len(orders)):
        if (k+1< len(orders)):
            sira = str(ind[orders[k]]).replace('␣','').replace('.','')
            inddict[sira] = ind[orders[k]+1:orders[k+1]]
        else:
            inddict[str(int(sira)+1)]  = ind[orders[k]+1:]
    mprlist.append(inddict)




df['ind'] = indlist
df['fai'] = failist
df['mpr'] = mprlist

dataname = '06.tcmb.smpcm.clean.sentences'
df.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

4.7 Python N-Gram Extractor

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Aug  9 16:00:44 2018

#####################################################################
#                          _          _              _          _    #
#     _   _   __ _  ___ (_) _ __   | | __ _   _ | |_  _   _ | | __    #
#    | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                           #
#     ____                                 _  _                       #
#    / __ \    __ _   _ __ ___     __ _ (_)| |     ___  ___    _ __ ___    #
#   / / _  |  / _  || '_    _ \   / _  || || |    / __|/ _ \ | '_    _ \   #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                    #
#####################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
#####################################################################
"""



import sys, re, math, collections
from os import path
import numpy as np
from PIL import Image
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
from collections import Iterable
from itertools import chain
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
from nltk.tokenize import sent_tokenize as st
from nltk import FreqDist as freq
from nltk.util import ngrams
from nltk.corpus import stopwords
import nltk
import ast



path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'

results = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/Results/'

df=pd.read_csv(path+'06.tcmb.smpcm.clean.sentences.csv', sep=';', header='infer')

df=(df.sort_values('No', ascending=True)).reset_index(drop=True)

ma = max(df.Year)
mi = min(df.Year)

years = list(range(mi,ma+1))

tab = pd.crosstab(index=df.Year, columns="count")

counts =tab['count'].tolist()
counts.insert(0,0)
counts.insert(0,0)
for i in range(1,len(counts)):
    counts[i]+= counts[i-1]
```

```
yind = []
for c in range(1,len(counts)):
    li = []
    for k in range(counts[c-1], counts[c]):
        li += list(ast.literal_eval(df.ind[k]).values())
        if(counts[c-1]==k):
            yind.append(li)

yfai = []
for c in range(1,len(counts)):
    li = []
    for k in range(counts[c-1], counts[c]):
        li += list(ast.literal_eval(df.fai[k]).values())
        if(counts[c-1]==k):
            yfai.append(li)

ympr = []
for c in range(1,len(counts)):
    li = []
    for k in range(counts[c-1], counts[c]):
        li += list(ast.literal_eval(df.mpr[k]).values())
        if(counts[c-1]==k):
            ympr.append(li)

#Get yearly corpuses
yind2 = []
for i in range(len(yind)):
    yind2.append([item for sublist in yind[i] for item in sublist])

yind = []
for i in range(len(yind2)):
    li = "␣".join(yind2[i])
    yind.append(li)




yfai2 = []
for i in range(len(yind)):
    yfai2.append([item for sublist in yfai[i] for item in sublist])

yfai = []
for i in range(len(yfai2)):
    li = "␣".join(yfai2[i])
    yfai.append(li)




ympr2 = []
for i in range(len(yind)):
    ympr2.append([item for sublist in ympr[i] for item in sublist])


ympr = []
for i in range(len(ympr2)):
    li = "␣".join(ympr2[i])
    ympr.append(li)




#Set stopwords
default_stopwords = set(nltk.corpus.stopwords.words('english'))

#Set topwords
topwords = 5

ind1 = []
ind2 = []
ind3 = []
for year in range(13):
    words = nltk.word_tokenize(yind[year])
```

```
    words = [word for word in words if len(word) > 1]
    words = [token.lower() for token in words]
    words = [w for w in words if not w in default_stopwords]
    fdist = freq(words)
    all_counts = dict()
    ngram = []
    for size in 1, 2, 3:
        all_counts[size] = freq(ngrams(words, size))
        a = all_counts[size].most_common(topwords)
        a2 = []
        for i in range(topwords):
            li = "␣".join(a[i][:][0])
            count = a[i][:][1]
            a2.append((li,count))
        ngram.append(a2)
    ind1.append(ngram[0])
    ind2.append(ngram[1])
    ind3.append(ngram[2])


fai1 = []
fai2 = []
fai3 = []
for year in range(13):
    words = nltk.word_tokenize(yfai[year])
    words = [word for word in words if len(word) > 1]
    words = [token.lower() for token in words]
    words = [w for w in words if not w in default_stopwords]
    fdist = freq(words)
    all_counts = dict()
    ngram = []
    for size in 1, 2, 3:
        all_counts[size] = freq(ngrams(words, size))
        a = all_counts[size].most_common(topwords)
        a2 = []
        for i in range(topwords):
            li = "␣".join(a[i][:][0])
            count = a[i][:][1]
            a2.append((li,count))
        ngram.append(a2)
    fai1.append(ngram[0])
    fai2.append(ngram[1])
    fai3.append(ngram[2])


mpr1 = []
mpr2 = []
mpr3 = []
for year in range(13):
    words = nltk.word_tokenize(ympr[year])
    words = [word for word in words if len(word) > 1]
    words = [token.lower() for token in words]
    words = [w for w in words if not w in default_stopwords]
    fdist = freq(words)
    all_counts = dict()
    ngram = []
    for size in 1, 2, 3:
        all_counts[size] = freq(ngrams(words, size))
        a = all_counts[size].most_common(topwords)
        a2 = []
        for i in range(topwords):
            li = "␣".join(a[i][:][0])
            count = a[i][:][1]
            a2.append((li,count))
        ngram.append(a2)
    mpr1.append(ngram[0])
    mpr2.append(ngram[1])
    mpr3.append(ngram[2])


years = list(range(2006,2019))

df = pd.DataFrame({'Years':years, 'indgram1': ind1, 'indgram2': ind2,
                   'indgram3': ind3,
```

```
                    'faigram1': fai1, 'faigram2': fai2, 'faigram3': fai3,
                    'mprgram1': mpr1, 'mprgram2': mpr2, 'mprgram3': mpr3})

dataname = '07.ngrams.yearly'
df.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

4.8 Python Sentiment Neuron

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Aug 12 16:25:52 2018

######################################################################
#                          _          _              _          _    #
#     _   _   __ _  ___ (_) _ __   | | __ _    _ | |_  _   _ | | __    #
#    | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#    | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#     \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#     |___/                                                          #
#     ____                                 _  _                      #
#    / __ \   __ _  _ __ ___     __ _ (_)| |    ___  ___    _ __ ___   #
#   / / _  | / _  || '_   _ \   / _  || || |   / __|/ _ \ | '_   _ \  #
#  | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#   \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#    \____/  |___/                                                   #
######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com  ######################################
######################################################################
"""


import sys, re, math, collections, ast
from os import path
import numpy as np
from PIL import Image
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
from collections import Iterable
from itertools import chain, repeat
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
from nltk.tokenize import sent_tokenize as st
from nltk import FreqDist as freq
from nltk.util import ngrams
from nltk.corpus import stopwords
import nltk
import nltk.classify.util
from nltk.classify import NaiveBayesClassifier
from nltk.corpus import names
from nltk.corpus import sentiwordnet as swn


path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'

results = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/Results/'

df=pd.read_csv(path+'06.tcmb.smpcm.clean.sentences.csv', sep=';', header='infer')

df=(df.sort_values('No', ascending=True)).reset_index(drop=True)

diction = pd.read_csv(path+'LoughranMcDonald_MasterDictionary_2016.csv',
                      sep=',', header='infer')

dneg = [token.lower() for token in diction.Word[diction.Negative!=0]]
dpos = [token.lower() for token in diction.Word[diction.Positive!=0]]
dneu = diction.Word[(diction.Negative==0) & (diction.Positive==0)]



# Since it is impossible to run a loop on dneu, i seperate the nonlowerable words.
dneu2 = []
dneu3 = []
for i in dneu:
    try:
```

```
            dneu2.append(i.lower())
        except:
            dneu3.append(i)
    dneu = dneu2

    indpassagescore = []
    for i in range(len(df)):
        passa = list(ast.literal_eval(df.ind[i]).values())
       # all sentences in all items as one passage
        passa = [item for sublist in passa for item in sublist]
        for doc in passa:
            sentences = nltk.sent_tokenize(doc)
            stokens = [nltk.word_tokenize(sent) for sent in sentences]
            taggedlist=[]
            for stoken in stokens:
                 taggedlist.append(nltk.pos_tag(stoken))
            wnl = nltk.WordNetLemmatizer()
            score_list=[]
            for idx,taggedsent in enumerate(taggedlist):
                score_list.append([])
                for idx2,t in enumerate(taggedsent):
                    newtag=''
                    lemmatized=wnl.lemmatize(t[0])
                    if t[1].startswith('NN'):
                        newtag='n'
                    elif t[1].startswith('JJ'):
                        newtag='a'
                    elif t[1].startswith('V'):
                        newtag='v'
                    elif t[1].startswith('R'):
                        newtag='r'
                    else:
                        newtag=''
                    if(newtag!=''):
                        synsets = list(swn.senti_synsets(lemmatized, newtag))
                        #tüm olası sentimentlerin ortalamasını al
                        score=0
                        if(len(synsets)>0):
                            for syn in synsets:
                                score+=syn.pos_score()-syn.neg_score()
                            score_list[idx].append(score/len(synsets))
        print(score_list)
        sentence_sentiment=[]
        for score_sent in score_list:
            sentence_sentiment.append(sum([word_score for word_score in \
                                           score_sent])/len(score_sent))
        print(sentence_sentiment)
        indpassagescore.append(sentence_sentiment)




    faipassagescore = []
    for i in range(len(df)):
        passa = list(ast.literal_eval(df.fai[i]).values()) # cümleler tek pasajda
        passa = [item for sublist in passa for item in sublist]
        for doc in passa:
            sentences = nltk.sent_tokenize(doc)
            stokens = [nltk.word_tokenize(sent) for sent in sentences]
            taggedlist=[]
            for stoken in stokens:
                 taggedlist.append(nltk.pos_tag(stoken))
            wnl = nltk.WordNetLemmatizer()
            score_list=[]
            for idx,taggedsent in enumerate(taggedlist):
                score_list.append([])
                for idx2,t in enumerate(taggedsent):
                    newtag=''
                    lemmatized=wnl.lemmatize(t[0])
                    if t[1].startswith('NN'):
                        newtag='n'
                    elif t[1].startswith('JJ'):
```

```
                    newtag='a'
                elif t[1].startswith('V'):
                    newtag='v'
                elif t[1].startswith('R'):
                    newtag='r'
                else:
                    newtag=''
                if(newtag!=''):
                    synsets = list(swn.senti_synsets(lemmatized, newtag))
                    score=0
                    if(len(synsets)>0):
                        for syn in synsets:
                            score+=syn.pos_score()-syn.neg_score()
                        score_list[idx].append(score/len(synsets))
    print(score_list)
    sentence_sentiment=[]
    for score_sent in score_list:
        sentence_sentiment.append(sum([word_score for word_score in \
                                       score_sent])/len(score_sent))
    print(sentence_sentiment)
    faipassagescore.append(sentence_sentiment)



mprpassagescore = []
for i in range(len(df)):
    passa = list(ast.literal_eval(df.mpr[i]).values())
    passa = [item for sublist in passa for item in sublist]
    for doc in passa:
        sentences = nltk.sent_tokenize(doc)
        stokens = [nltk.word_tokenize(sent) for sent in sentences]
        taggedlist=[]
        for stoken in stokens:
             taggedlist.append(nltk.pos_tag(stoken))
        wnl = nltk.WordNetLemmatizer()
        score_list=[]
        for idx,taggedsent in enumerate(taggedlist):
            score_list.append([])
            for idx2,t in enumerate(taggedsent):
                newtag=''
                lemmatized=wnl.lemmatize(t[0])
                if t[1].startswith('NN'):
                    newtag='n'
                elif t[1].startswith('JJ'):
                    newtag='a'
                elif t[1].startswith('V'):
                    newtag='v'
                elif t[1].startswith('R'):
                    newtag='r'
                else:
                    newtag=''
                if(newtag!=''):
                    synsets = list(swn.senti_synsets(lemmatized, newtag))

                    score=0
                    if(len(synsets)>0):
                        for syn in synsets:
                            score+=syn.pos_score()-syn.neg_score()
                        score_list[idx].append(score/len(synsets))

    print(score_list)
    sentence_sentiment=[]
    for score_sent in score_list:
        sentence_sentiment.append(sum([word_score for word_score \
                                       in score_sent])/len(score_sent))
    print(sentence_sentiment)
    mprpassagescore.append(sentence_sentiment)




indpassagescore = [score for sublist in indpassagescore for score in sublist]
```

```
#unlisting
faipassagescore = [score for sublist in faipassagescore for score in sublist]
mprpassagescore =[score for sublist in mprpassagescore for score in sublist]

df['indsem']= indpassagescore
df['faisem']= faipassagescore
df['mprsem']= mprpassagescore

dataname = '08.semantics.yearly'
df.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

```
day = []
year = []
month = []
number = []
items = []
sentno = []
sents = []
title = []
sentencebase = []

for i in range(len(df)):
    itemno = list(ast.literal_eval(df.ind[i]).keys())
    passa = list(ast.literal_eval(df.ind[i]).values())
    passa = [item for sublist in passa for item in sublist]

    bg = ast.literal_eval(df.ind[i])
    counts= []
    keys = []
    for key, value in bg.items():
        counts.append(sum(1 for v in value if v))
        keys.append(key)

    iteming = []
    for i in range(len(keys)):
        iteming.extend(repeat(keys[i], counts[i]))
    items.extend(iteming)


    for doc in range(len(passa)):
        sentences = nltk.sent_tokenize(passa[doc])
        stokens = [nltk.word_tokenize(sent) for sent in sentences]
        taggedlist=[]
        for stoken in stokens:
             taggedlist.append(nltk.pos_tag(stoken))
        wnl = nltk.WordNetLemmatizer()
        score_list=[]
        for idx,taggedsent in enumerate(taggedlist):
            score_list.append([])
            for idx2,t in enumerate(taggedsent):
                newtag=''
                lemmatized=wnl.lemmatize(t[0])
                if t[1].startswith('NN'):
                    newtag='n'
                elif t[1].startswith('JJ'):
                    newtag='a'
                elif t[1].startswith('V'):
                    newtag='v'
                elif t[1].startswith('R'):
                    newtag='r'
                else:
                    newtag=''
                if(newtag!=''):
                    synsets = list(swn.senti_synsets(lemmatized, newtag))

                    score=0
                    if(len(synsets)>0):
                        for syn in synsets:
                            score+=syn.pos_score()-syn.neg_score()
```

```
                    score_list[idx].append(score/len(synsets))
        sentence_sentiment=[]
        for score_sent in score_list:
            try:
                sentence_sentiment.append(sum([word_score for word_score \
                                               in score_sent])/len(score_sent))
            except:
                sentence_sentiment.append(0)
        sentencebase.append(passa[doc])
        sents.append(sentence_sentiment)
        day.append(df.Day[i])
        year.append(df.Year[i])
        month.append(df.Month[i])
        number.append(df.No[i])
        sentno.append(doc+1)
        title.append('Inflation_Developments')



for i in range(len(df)):
    itemno = list(ast.literal_eval(df.fai[i]).keys())
    passa = list(ast.literal_eval(df.fai[i]).values())
    passa = [item for sublist in passa for item in sublist]

    bg = ast.literal_eval(df.fai[i])
    counts= []
    keys = []
    for key, value in bg.items():
        counts.append(sum(1 for v in value if v))
        keys.append(key)

    iteming = []
    for i in range(len(keys)):
        iteming.extend(repeat(keys[i], counts[i]))
    items.extend(iteming)

    for doc in range(len(passa)):
        sentences = nltk.sent_tokenize(passa[doc])
        stokens = [nltk.word_tokenize(sent) for sent in sentences]
        taggedlist=[]
        for stoken in stokens:
             taggedlist.append(nltk.pos_tag(stoken))
        wnl = nltk.WordNetLemmatizer()
        score_list=[]
        for idx,taggedsent in enumerate(taggedlist):
            score_list.append([])
            for idx2,t in enumerate(taggedsent):
                newtag=''
                lemmatized=wnl.lemmatize(t[0])
                if t[1].startswith('NN'):
                    newtag='n'
                elif t[1].startswith('JJ'):
                    newtag='a'
                elif t[1].startswith('V'):
                    newtag='v'
                elif t[1].startswith('R'):
                    newtag='r'
                else:
                    newtag=''
                if(newtag!=''):
                    synsets = list(swn.senti_synsets(lemmatized, newtag))

                    score=0
                    if(len(synsets)>0):
                        for syn in synsets:
                            score+=syn.pos_score()-syn.neg_score()
                        score_list[idx].append(score/len(synsets))
        sentence_sentiment=[]
        for score_sent in score_list:
            try:
                sentence_sentiment.append(sum([word_score for word_score in \
                                               score_sent])/len(score_sent))
            except:
                sentence_sentiment.append(0)
```

```
            sentencebase.append(passa[doc])
            sents.append(sentence_sentiment)
            day.append(df.Day[i])
            year.append(df.Year[i])
            month.append(df.Month[i])
            number.append(df.No[i])
            sentno.append(doc+1)
            title.append('Factors␣Affecting␣Inflation')



for i in range(len(df)):
    itemno = list(ast.literal_eval(df.mpr[i]).keys())
    passa = list(ast.literal_eval(df.mpr[i]).values())
    passa = [item for sublist in passa for item in sublist]

    bg = ast.literal_eval(df.mpr[i])
    counts= []
    keys = []
    for key, value in bg.items():
        counts.append(sum(1 for v in value if v))
        keys.append(key)

    iteming = []
    for i in range(len(keys)):
        iteming.extend(repeat(keys[i], counts[i]))
    items.extend(iteming)

    for doc in range(len(passa)):
        sentences = nltk.sent_tokenize(passa[doc])
        stokens = [nltk.word_tokenize(sent) for sent in sentences]
        taggedlist=[]
        for stoken in stokens:
             taggedlist.append(nltk.pos_tag(stoken))
        wnl = nltk.WordNetLemmatizer()
        score_list=[]
        for idx,taggedsent in enumerate(taggedlist):
            score_list.append([])
            for idx2,t in enumerate(taggedsent):
                newtag=''
                lemmatized=wnl.lemmatize(t[0])
                if t[1].startswith('NN'):
                    newtag='n'
                elif t[1].startswith('JJ'):
                    newtag='a'
                elif t[1].startswith('V'):
                    newtag='v'
                elif t[1].startswith('R'):
                    newtag='r'
                else:
                    newtag=''
                if(newtag!=''):
                    synsets = list(swn.senti_synsets(lemmatized, newtag))

                    score=0
                    if(len(synsets)>0):
                        for syn in synsets:
                            score+=syn.pos_score()-syn.neg_score()
                        score_list[idx].append(score/len(synsets))
        sentence_sentiment=[]
        for score_sent in score_list:
            try:
                sentence_sentiment.append(sum([word_score for word_score in \
                                               score_sent])/len(score_sent))
            except:
                sentence_sentiment.append(0)
        sentencebase.append(passa[doc])
        sents.append(sentence_sentiment)
        day.append(df.Day[i])
        year.append(df.Year[i])
        month.append(df.Month[i])
        number.append(df.No[i])
        sentno.append(doc+1)
        title.append('Monetary␣Policy␣and␣Risks')
```

```
sents = [s for sublist in sents for s in sublist]
df1 = pd.DataFrame({'Day':day, 'Year': year, 'month': month, 'ItemNo': items,
                    'SentenceNo': sentno, 'SentSemanticScore': sents,
                    'Title': title,
                    'SentenceBase' : sentencebase, 'ReportNo' : number})

dataname = '09.semantics.sentencebased'
df1.to_csv(path+dataname+'.csv', sep=';', encoding='utf-8', index=False)
```

```
bg = ast.literal_eval(df.mpr[0])
counts= []
keys = []
for key, value in bg.items():
    counts.append(sum(1 for v in value if v))
    keys.append(key)

iteming = []
for i in range(len(keys)):
    iteming.extend(repeat(keys[i], counts[i]))
```

4.9 Python LSTM Network Model Implementation in Keras

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Aug  6 11:14:58 2018

######################################################################
#                          _         _              _          _     #
#    _   _   __ _  ___ (_) _ __  | | __ _   _ | |_  _   _ | | __     #
#   | | | | / _  |/ __|| || '_ \ | |/ /| | | || __|| | | || |/ /     #
#   | |_| || (_| |\__ \| || | | ||   < | |_| || |_ | |_| ||   <      #
#    \__, | \__,_||___/|_||_| |_||_|\_\ \__,_| \__| \__,_||_|\_\     #
#    |___/                                                           #
#    _____                                 _  _                      #
#   / __ \    __ _  _ __ ___     __ _ (_)| |    ___  ___    _ __ ___    #
#  / / _  |  / _  || '_   _ \   / _  || || |   / __|/ _ \ | '_   _ \  #
# | | (_| || (_| || | | | | || (_| || || | _| (__| (_) || | | | | | | #
#  \ \__,_| \__, ||_| |_| |_| \__,_||_||_|(_)\___|\___/ |_| |_| |_| #
#   \____/  |___/                                                    #
######################################################################
#@author: Yasin KÜTÜK           ######################################
#@web   : yasinkutuk.com        ######################################
#@email : yasinkutuk@gmail.com ######################################
######################################################################
"""

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 25 21:30:14 2018

@author: yasin
"""

import sys,re,math
from os import path
import numpy as np
from PIL import Image
import pandas as pd
from wordcloud import WordCloud, STOPWORDS
from collections import Iterable
from itertools import chain
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns


from sklearn.feature_extraction.text import CountVectorizer
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical



path='/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/'

results = '/media/DRIVE/Dropbox/_Dissertation/Yazim/Paper3_NLP/Analysis/Results/'

df=pd.read_csv(path+'09.semantics.sentencebased.csv', sep=';', header='infer')

data = df[['SentSemanticScore', 'SentenceBase']]

data['sentiment'] = 0
data = data[data.SentSemanticScore!=0]
data.sentiment[data.SentSemanticScore>0] = 'Positive'
data.sentiment[data.SentSemanticScore<0] = 'Negative'
```

```
data=data.rename(columns={'SentSemanticScore': 'score', 'SentenceBase': 'text',
                          'sentiment': 'sentiment'})

data['text'] = data['text'].apply(lambda x: x.lower())
data['text'] = data['text'].apply((lambda x: re.sub('[^a-zA-z0-9\s]','',x)))

print(data[data['sentiment'] == 'Positive'].size)
print(data[data['sentiment'] == 'Negative'].size)



max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, split=' ')
tokenizer.fit_on_texts(data['text'].values)
X = tokenizer.texts_to_sequences(data['text'].values)
X = pad_sequences(X)


embed_dim = 128
lstm_out = 196
testsize = 0.10


model = Sequential()
model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))
model.add(SpatialDropout1D(0.4))
model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(2,activation='softmax'))
model.compile(loss = 'categorical_crossentropy', optimizer='adam',
              metrics = ['accuracy'])
print(model.summary())

Y = pd.get_dummies(data['sentiment']).values
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = testsize,
                                                    random_state = 42)
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)



batch_size = 32
model.fit(X_train, Y_train, epochs = 7, batch_size=batch_size, verbose = 10)


validation_size = int((X_train.shape[0] * 0.10) // 1)



X_validate = X_test[-validation_size:]
Y_validate = Y_test[-validation_size:]
X_test = X_test[:-validation_size]
Y_test = Y_test[:-validation_size]
score,acc = model.evaluate(X_test, Y_test, verbose = 2, batch_size = batch_size)
print("score: %.2f" % (score))
print("acc: %.3f" % (acc))


pos_cnt, neg_cnt, pos_correct, neg_correct = 0, 0, 0, 0
for x in range(len(X_validate)):

    result = model.predict(X_validate[x].reshape(1,X_test.shape[1]),
                           batch_size=1,verbose = 2)[0]

    if np.argmax(result) == np.argmax(Y_validate[x]):
        if np.argmax(Y_validate[x]) == 0:
            neg_correct += 1
        else:
            pos_correct += 1

    if np.argmax(Y_validate[x]) == 0:
        neg_cnt += 1
    else:
        pos_cnt += 1
```

```
print("pos_acc", pos_correct/0.1*100, "%")
print("neg_acc", neg_correct/0.1*100, "%")
```

CURRICULUM VITAE

Name Surname: YASİN KÜTÜK

Place and Date of Birth: ANKARA / 09.07.1982

E-Mail: yasinkutuk@gmail.com, yasinkutuk@itu.edu.tr

EDUCATION:

* **B.Sc.:** 2007, Ankara University, Philosophy
* **B.Sc.:** 2019, Anadolu University, Economics
* **M.Sc.:** 2011, ITU, Economics

PROFESSIONAL EXPERIENCE AND REWARDS:

* 2009-2018, Sabancı University (TUBITAK).
* 2011, Turkish Economic Association - Graduate Thesis Award.
* 2013-Present, Altınbaş University.
* 2017, Best Poster in Twelfth International Conference on Interdisciplinary Social Sciences at Hiroshima.
* 2019, Completed Doctorate at Istanbul Technical University.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

* **Kütük, Y.**, Güloğlu, B. (2019). Prediction of Transition Probabilities from Unemployment to Employment for Turkey Via Machine Learning and Econometrics: A Comparative Study, Journal of Research in Economics, 3 (1), 58-75.
* **Kütük, Y.** (2018). Prediction of Transition Probabilities from Unemployment to Employment for Turkey Via Machine Learning and Econometrics: A Comparative Study, VI. International Conference on Economics, Turkish Economic Association.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

* **Kütük, Y.** (2011). The Relationship Between Economic And Social Indicators And Suicides In Turkey, İstanbul Technical University, Institute of Social Sciences.
* İlkkaracan, A., **Kütük, Y.** (2013). Türkiye'de intiharların ekonomik dalgalanmalarla ilişkisi üzerine bir uygulama, Ümit Şenesen'e Armağan Paylaşımlar, Literatür Yayıncılık.
* Yıldırım, M., Belen, M., **Kütük, Y.** (2014). Küresel Emtia Fiyatları ile Hisse Senedi Getirileri Arasındaki İlişkinin İncelenmesi: Kardemir ve İzdemir Üzerine Bir Uygulama. *Finansal Araştırmalar ve Çalışmalar Dergisi, 5(10)*, 107-138.
* Yıldırım, M., Belen, M., **Kütük, Y.** (2014). Mamul Fiyatları İle Hisse Senetleri Getirileri Arasındaki İlişkinin İncelenmesi: Erdemir Üzerine Bir Uygulama. *Ekonomik ve Sosyal Araştırmalar Dergisi*.
* **Kütük, Y.** (2014). İşgücü Piyasasında Mezuniyet Alanında İş Bulma Olasılığını Belirleyen Faktörlerin İncelenmesi: 2009 -2012 Türkiye Örneği, 4th International Economics Conference, Turkish Economic Association.
* **Kütük, Y.** (2015). Fertlerin ve Hanehalklarının Tasarruf Davranışını Belirleyen Faktörlerin İncelenmesi: 2002 -2013 Türkiye Örneği, TEK 18. Ulusal İktisat Sempozyumu, Konya.
* **Kütük, Y.** (2016). Is there any externality in core family via the effects of working same industry?, 5th International Economics Conference, Turkish Economic Association.
* İlhan, B.Y., **Kütük, Y.** (2016). Is there any externality in core family via the effects of working same industry?, 19th. EBES Conference, Istanbul.
* İlhan, B.Y., **Kütük, Y.** (2016). Structural Change in Wage Differentiation Patterns for Turkey in Terms of Working in the Same Industry, *AURUM Sosyal Bilimler Dergisi, 1 (2)*, 7-32.
* Usturalı, A., **Kütük, Y.** (2017). Demokratik Olan ve Omayan Ülkelerde Seçim Dışı siyasal Katılıma Kurumsal Güvenin Etkisi Üzerine Bir Uygulama, 15. Ulusal Sosyal Bilimler Konferansı, ODTÜ, Ankara.
* **Kütük, Y.** (2017). The Benefits of Working in the Same Occupation for the Total Household Income, Twelfth International Conference on Interdisciplinary Social Sciences, Hiroshima.