

MULTIRESOLUTION ANALYSIS OF S&P500 TIME SERIES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENİZ KENAN KILIÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
FINANCIAL MATHEMATICS

JUNE 2015

Approval of the thesis:

MULTIRESOLUTION ANALYSIS OF S&P500 TIME SERIES

submitted by **DENİZ KENAN KILIÇ** in partial fulfillment of the requirements for
the degree of **Master of Science in Department of Financial Mathematics, Middle
East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics**

Assoc. Prof. Dr. Ali Devin Sezer
Head of Department, **Financial Mathematics**

Assoc. Prof. Dr. Ömür Uğur
Supervisor, **Scientific Computing**

Examining Committee Members:

Assoc. Prof. Dr. Yeliz Yolcu Okur
Financial Mathematics, METU

Assoc. Prof. Dr. Ömür Uğur
Scientific Computing, METU

Assoc. Prof. Dr. Ümit Aksoy
Mathematics, Atılım University

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: DENİZ KENAN KILIÇ

Signature : 

ABSTRACT

MULTIRESOLUTION ANALYSIS OF S&P500 TIME SERIES

Kılıç, Deniz Kenan

M.S., Department of Financial Mathematics

Supervisor : Assoc. Prof. Dr. Ömür Uğur

June 2015, 168 pages

Time series analysis is an essential research area for almost all people who are dealing with scientific and engineering problems. Main aim is to understand the underlying characteristics of the time series by using time as well as frequency domain analyses. Then one can make a prediction for the desired system to forecast observations ahead. Time series modeling, frequency domain analysis and some descriptive statistical analysis are main subjects of this thesis. Choosing an appropriate model is the main focus of all analysis in order to make a good prediction. In this thesis financial time series are focused, particularly S&P500 daily closing prices and it's return values are handled. Fourier transform and wavelet transform are creatively at the center of the frequency domain analysis. Knowing the fact that financial time series are complex data sets to sufficiently predict the future, multiresolution analysis is handled in this thesis using the wavelet transforms to figure out specialties of S&P500 data. Also, apparently, models that are appropriate for the financial time series are discussed in the application part.

Keywords : Time series analysis, wavelets, multiresolution analysis, statistical analysis

ÖZ

S&P500 ZAMAN SERİSİNİN ÇOKLU ÇÖZÜNÜRLÜK ANALİZİ

Kılıç, Deniz Kenan

Yüksek Lisans, Finansal Matematik Bölümü

Tez Yöneticisi : Doç. Dr. Ömür Uğur

Haziran 2015, 168 sayfa

Zaman serisi analizi hemen hemen tüm bilim ve mühendislik problemleri ile uğraşan kişiler için gerekli bir araştırma alanıdır. Temel amacı zaman uzayı ve aynı zamanda frekans uzayı analizini kullanarak zaman serisinin altında yatan özelliklerini anlamaktır. Sonrasında, zaman serisinin ileriye dönük verileri tahmin edilebilir. Zaman serisi modellemesi, frekans uzayı analizi ve bazı tanımlayıcı istatistiksel analizler bu tezin ana konularıdır. Uygun bir model seçmek, iyi bir tahminleme yapabilmek için analizin ana odaklılığını oluşturmaktadır. Bu çalışmada finansal zaman serileri üzerine odaklanılmıştır ve özellikle S&P500 günlük kapanış fiyatları ve getiri değerleri ele alınmıştır. Fourier dönüşümü ve dalgacık dönüşümü frekans analizinin merkezini oluşturmaktadır. Finansal zaman serilerinin, yeterli düzeyde geleceği tahmin etmek için karmaşık veri setleri olduğu bilindiği için, bu çalışmada S&P500 verisinin özelliklerini ortaya çıkarmak için dalgacık dönüşümleri kullanılarak çoklu çözünürlük analizi ele alınmıştır. Aynı zamanda, finansal zaman serileri için uygun olan modeller uygulama kısmında tartışılmıştır.

Anahtar Kelimeler: Zaman serisi analizi, dalgacıklar, çoklu çözünürlük analizi, istatistiksel analizler

To My Family

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor Assoc. Prof. Dr. Ömür Uğur for introducing me to the subject ‘time series analysis’ and his patient guidance, enthusiastic encouragement and valuable advices during the development and preparation of this thesis. His willingness to give his time and to share his experiences has brightened my path.

And, I thank all members of the Institute of Applied Mathematics of Middle East Technical University for their kindness and help.

Furthermore, I would like to specially thank to my friend Dilek Aydoğan for his friendship, support and help.

Finally, I would like to express my pleasure to having such a great family and thank each of them for their endless love and existence.

TABLE OF CONTENTS

ABSTRACT	vii
ÖZ	ix
ACKNOWLEDGMENTS	xiii
TABLE OF CONTENTS	xv
LIST OF FIGURES	xix
LIST OF TABLES	xxv
LIST OF ABBREVIATIONS	xxvii
CHAPTERS	
1 INTRODUCTION	1
2 LINEAR TIME SERIES AND HETEROSCEDASTIC MODELS . .	3
2.1 Introduction	3
2.2 Autoregressive Moving Average (ARMA) Models	5
2.3 Autoregressive Integrated Moving Average (ARIMA) Models	7
2.4 Multiplicative Seasonal Autoregressive Integrated Moving Average (SARIMA) Models	22
2.5 Forecasting	22
Note.	24
2.6 Heteroscedastic Models	25
2.6.1 ARCH and GARCH Models	26

2.6.2	Other Heteroscedastic Models	29
2.7	Other Methods: Overview	29
2.7.1	Threshold Models	29
2.7.2	Neural Networks	30
2.7.3	Other Methods	32
3	FUNDAMENTALS OF FOURIER AND WAVELET THEORY: FREQUENCY DOMAIN ANALYSIS	35
3.1	Fourier Theory	36
3.1.1	Introduction	36
3.1.2	Continuous Fourier Transform	36
3.1.3	Discrete Fourier Transform	36
3.1.4	Spectral Analysis and Periodogram	37
3.2	Wavelet Theory	50
3.2.1	Introduction	50
3.2.2	Continuous Wavelet Transform	55
3.2.3	Discrete Wavelet Transform	56
3.2.4	Maximal Overlap Discrete Wavelet Transform	63
3.2.5	Multiresolution Analysis	65
3.2.6	Wavelet Variance Analysis	67
4	EMPIRICAL RESULTS	71
4.1	Characteristic of S&P500 Data	71
4.2	Exponential Smoothing	94
4.3	Frequency Domain Analysis	97

4.3.1	Fourier Transform Analysis	97
4.3.2	Wavelet Analysis	101
4.4	Time Domain Analysis	119
4.4.1	SARIMA Fitting	119
4.4.2	GARCH Methods	135
4.4.3	Other Methods for Time Series Analysis	135
5	CONCLUSION AND OUTLOOK	143
	REFERENCES	147
A	Some of the <i>R</i> -Codes	153

LIST OF FIGURES

Figure 2.1 Simulated AR(1) Processes.	9
Figure 2.2 ACF and PACF Plots of AR(1) Process with First 1000 Observations where $\phi = +0.8$	10
Figure 2.3 ACF and PACF Plots of AR(1) Process with First 100 Observations where $\phi = +0.8$	11
Figure 2.4 ACF and PACF Plots of AR(1) Process with First 100 Observations where $\phi = -0.8$	12
Figure 2.5 Simulated AR(2) Process.	13
Figure 2.6 ACF and PACF Plots of AR(2) Process with First 100 Observations where $\phi_1 = 0.5$ and $\phi_2 = 0.4$	14
Figure 2.7 Simulated MA(1) Process.	15
Figure 2.8 ACF and PACF Plots of MA(1) Process with First 100 Observations where $\theta = 0.5$	16
Figure 2.9 Simulated MA(2) Process.	17
Figure 2.10 ACF and PACF Plots of MA(2) Process with First 100 Observations where $\theta_1 = -0.5$ and $\theta_2 = -0.9$	18
Figure 2.11 Simulated ARMA(2, 2) Process.	19
Figure 2.12 ACF and PACF Plots of ARMA(2, 2) Process with First 100 Observations where Coefficients are Given as in Figure 2.11.	20
Figure 2.13 Simulated ARIMA(1, 1, 1) Processes and related ACF plots.	21
Figure 2.14 Daily Closing Prices of DAX data and its return values.	26
Figure 2.15 GARCH(1, 1) Model Fitted to DAX Return Data.	28
Figure 2.16 Typical Structure of Neural Network System	31
Figure 2.17 Forecasting by Fitting NNAR(2, 1) Model.	31
Figure 3.1 Relation between power spectrum and autocovariance.	38

Figure 3.2 The Spectral Densities of White Noise, Moving Average and Autoregressive Process separately in Example 3.1, Example 3.2 and Example 3.3.	43
Figure 3.3 The spectral density of ARMA(2, 1) in Example 3.3.	44
Figure 3.4 Three periodic functions with different frequencies and their sum.	45
Figure 3.5 Periodogram of sum function by using FFT.	46
Figure 3.6 Periodogram where frequency of 3rd function is 0.75 at left, 0.5 at right.	47
Figure 3.7 Data of DAX and Related ACF, PACF, Histogram and Quantile Graphs.	48
Figure 3.8 More detailed histogram of DAX.	49
Figure 3.9 DAX data, spectral density by smoothed periodogram, spectral density and spectral density using AR method.	51
Figure 3.10 Periodogram of DAX and its smoothed versions using Daniell smoothers by spanning.	52
Figure 3.11 Periodogram of DAX and its smoothed versions using Daniell smoothers by kernel.	53
Figure 3.12 Flowchart of the pyramid algorithm for decomposing. In every iteration each coefficients divided by 2 which is called downsampling as mentioned by Gençay et al. in [28]. Frequency decreases as j increases.	61
Figure 3.13 Flowchart of the pyramid algorithm for synthesis. In every iteration each coefficients multiplied by 2 which is called upsampling as mentioned by Gençay et al. in [28].	61
Figure 4.1 S&P500 Daily Closing Price Data and Return Data of Daily Closing Prices	72
Figure 4.2 Recurrence Plots of S&P500 Daily Closing Prices.	73
Figure 4.3 Recurrence Plots of S&P500 Return Values.	74
Figure 4.4 Histogram of Daily Closing Prices of S&P500 and Q-Q Plot of Daily Closing Prices of S&P500.	75
Figure 4.5 Histogram of Return Values of S&P500 and Q-Q Plot of Return Values of S&P500.	76
Figure 4.6 Q-Q Plots of Related Distributions for S&P500 Return Data.	77

Figure 4.7 Grid of scatterplots of daily closing prices (t -lag) versus daily closing prices for lag = 1, 2, ..., 9.	78
Figure 4.8 Grid of scatterplots of return values (t -lag) versus return values for lag = 1, 2, ..., 9.	79
Figure 4.9 S&P500 Return Series and its Absolute and Squared Types.	80
Figure 4.10 ACF of Absolute and Squared Types of S&P500 Return Series.	81
Figure 4.11 ACF and PACF of S&P500 Daily Closing Prices and its Return Values.	82
Figure 4.12 ACF and PACF of Differenced S&P500 Return Values.	83
Figure 4.13 ACF and PACF of S&P500 Daily Closing Prices with Frequency 1.	84
Figure 4.14 ACF and PACF of S&P500 Daily Closing Prices with Frequency 260.	85
Figure 4.15 ACF and PACF of S&P500 Return Values with Frequency 1.	86
Figure 4.16 ACF and PACF of S&P500 Return Values with Frequency 260.	87
Figure 4.17 ACF and PACF of Differenced S&P500 Return Values with Frequency 1.	88
Figure 4.18 Linear Filtering of Daily Closing Prices with $a = 2, 5$ and 65.	90
Figure 4.19 Decomposition of Logarithmic Closing Prices of S&P500 with Frequency 260 by Using <i>Loess</i> Method.	91
Figure 4.20 Decomposition of Logarithmic Closing Prices of S&P500 with Frequency 260 by Using MA Method.	92
Figure 4.21 S&P500 Daily Closing Prices without Trend Component.	93
Figure 4.22 S&P500 Data with Holt-Winters Smoothing.	94
Figure 4.23 Predicted 200 Observations by Using Holt-Winters Method.	95
Figure 4.24 Predicted 200 Observations by Using Holt-Winters with 75% and 95% Confidence Intervals.	96
Figure 4.25 Forecasting of S&P500 Data by STLF Method.	97
Figure 4.26 S&P500 Daily Closing Prices Data and Different Type of Periodograms.	98
Figure 4.27 Periodograms of S&P500 Data with Given Smoothness Parameters.	99
Figure 4.28 AR(3) Fitted Spectrum of S&P500 Data with Yule-Walker and Burg Methods.	100

Figure 4.29 DWT of S&P500 Daily Closing Prices with Least Asymetric(8) Filter by Reflected Data of Wavelet Scales up to 5.	102
Figure 4.30 DWT of S&P500 Daily Returns with Least Asymetric(8) Filter by Reflected Data of Wavelet Scales up to 6.	103
Figure 4.31 MODWT of S&P500 Daily Closing Prices with $J = 8$ by Using Haar Filter.	104
Figure 4.32 MODWT of S&P500 Daily Closing Prices with $J = 8$ by Using Least Asymetric Filter with Length 8.	105
Figure 4.33 MODWT of Daily Closing Prices up to Level 10.	105
Figure 4.34 MODWT of S&P500 Daily Returns with $J = 8$ by Using Haar Filter.	106
Figure 4.35 MODWT of S&P500 Daily Returns with $J = 8$ by Using Least Asymetric Filter with Length 8.	107
Figure 4.36 MODWT of S&P500 Daily Returns with levels 5, 6, 7, 8 by Using Haar Filter.	108
Figure 4.37 MODWT of S&P500 Daily Returns with levels 5, 6, 7, 8 by Using Least Asymetric Filter with Length 8.	109
Figure 4.38 MODWT of S&P500 Daily Returns with levels 8, 9 by Using Least Asymetric Filter with Length 8.	110
Figure 4.39 MRA of Absolute S&P500 Return Values with Scale Level 4.	110
Figure 4.40 Wavelet Power Spectrum by using “white noise” method of generating surrogate time series.	112
Figure 4.41 Reconstruction of the de-trended (0.75 spanned) series by the “white noise” method.	113
Figure 4.42 Wavelet Power Spectrum by using “shuffle” method of generating surrogate time series.	114
Figure 4.43 Reconstruction of the de-trended (0.75 spanned) series by the “shuffle” method.	115
Figure 4.44 Wavelet Power Spectrum by using AR(1) method of generating surrogate time series.	116
Figure 4.45 Reconstruction of the de-trended (0.75 spanned) series by AR(1) method.	117
Figure 4.46 <i>Bias-corrected</i> and <i>Biased</i> Wavelet Spectra of S&P500 Daily Closing Prices.	118

Figure 4.47 ARIMA(1, 1, 1) Fitted S&P500 Daily Closing Prices.	119
Figure 4.48 SARIMA(1, 1, 1) \times (0, 1, 1) ₄₂ Fitted S&P500 Daily Closing Prices with Frequency 260.	120
Figure 4.49 ARIMA(1, 1, 1) Fitted S&P500 Return Values.	121
Figure 4.50 PACF of S&P500 Daily Closing Prices and ACF of S&P500 Return Values.	122
Figure 4.51 Daily Closing Prices of S&P500 versus Simulated ARIMA(0, 1, 2) Related to Original Data.	123
Figure 4.52 100 Step Prediction of ARIMA(2, 1, 0) Model which is Suggested by AIC.	124
Figure 4.53 Forecasts for S&P500 Data and its First Difference with Auto Selected ARIMA Models.	125
Figure 4.54 Residuals of Forecasts in Figure 4.53 Respectively, i.e. Time Plot of Forecast Errors.	126
Figure 4.55 ACF Plots of Residuals (forecast errors) of Auto Selected Models given in Figure 4.53.	127
Figure 4.56 Histogram Plots of Forecast Errors Related to Auto Selected Models Respectively in Figure 4.53.	128
Figure 4.57 Prediction without Drift and with Drift modelled by ARIMA(0, 1, 2) Model.	129
Figure 4.58 Zoomed Version of Prediction without Drift and with Drift modelled by ARIMA(0, 1, 2) Model.	130
Figure 4.59 Predictions of S&P500 Daily Closing Prices of with Given SARIMA Models.	131
Figure 4.60 Forecasts of Daily Closing Prices with Selected SARIMA Models.	133
Figure 4.61 Zoomed Versions of Forecasts of Daily Closing Prices with Selected SARIMA Models.	134
Figure 4.62 Plots According to GARCH(4, 1) Fitting with Skewed Generalized Error Distribution.	141

LIST OF TABLES

Table 2.1 Coefficients of Fitted AR(1)–ARCH(1) DAX Data	27
Table 2.2 Error Analysis	27
Table 2.3 Error Analysis of Fitted GARCH(1, 1) DAX Data	33
Table 2.4 Information Criterion Statistics of Fitted GARCH(1, 1) DAX Data . .	33
 Table 3.1 Basic Statistical Results of DAX Data	 50
 Table 4.1 Shapiro-Wilk and Kolmogorov-Smirnov Tests Results for Normality for Daily Closing Prices (DCP), Return Values (RV), and First Difference of Daily Closing Prices (FD-DCP).	 74
Table 4.2 Descriptive Statistics of Data Sets Related to S&P500 Daily Closing Prices	85
Table 4.3 Augmented Dickey-Fuller Test for Stationary and KPSS Test for Trend Stationary	89
Table 4.4 Results of SARIMA Fitting to Related Data	132
Table 4.5 Accuracy Values of Training Set and Test Set with respect to Related ARIMA Model	137
Table 4.6 Accuracy Values of Training Set and Test Set with respect to Related SARIMA Model	138
Table 4.7 Accuracy Values of Training Set and Test Set of Selected Models . .	139
Table 4.8 Information Criterion Statistics for Selected GARCH Models with given Distributions	140
Table 4.9 Accuracy Values of Test Set of AAR and LSTAR Models	140
Table 4.10 AIC Values of Given Models	142

LIST OF ABBREVIATIONS

AAR	Additive Nonlinear Autoregressive
ACF	Autocorrelation Function
ACF1	Autocorrelation of Errors at Lag 1
AIC	Akaike Information Criterion
AICc	Akaike Information Criterion with a correction
AR	Autoregressive
ARCH	Autoregressive Conditionally Heteroscedastic
ARIMA	Autoregressive Integrated Moving Average
BIC	Bayesian Information Criterion
CAC	Cotation Assistee en Continu
CFT	Continuous Fourier Transform
CHARMA	Conditional Heteroscedastic Autoregressive Moving Average
CI	Confidence Interval
cov	Covariance
CWT	Continuous Wavelet Transform
D	Daubechies
DAX	Deutscher Aktien Index
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
\mathbb{E}	Expectation
ESTAR	Exponential Smooth Transition Autoregressive
ETS	Exponential Smoothing State Space Model
FFT	Fast Fourier Transform
FTSE	Financial Times Stock Exchange
FWT	Fast Wavelet Transform
GARCH	Generalized Autoregressive Conditionally Heteroscedastic
GARCH-M	Generalized Autoregressive Conditionally Heteroscedastic in the Mean
HQIC	Hannan-Quinn Information Criterion
IGARCH	Integrated Generalized Autoregressive Conditionally Heteroscedastic

iid	Independent and identically distributed
KPSS	Kwiatkowski–Phillips–Schmidt–Shin
log	Natural Logarithm (\ln)
LA	Least Asymmetric
LSTAR	Logistic Smooth Transition Autoregressive
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
ME	Mean Error
MLE	Maximum-Likelihood Estimation
MODWT	Maximal Overlap Discrete Wavelet Transform
MPE	Mean Percentage Error
MRA	Multiresolution Analysis
NNAR	Neural Network Autoregression
\mathcal{N}	Normal Distribution
ODE	Ordinary Differential Equation
OLS	Ordinary Least Squares
\mathbb{P}	Probability
PACF	Partial Autocorrelation Function
PDE	Partial Differential Equation
PDWT	Partial Discrete Wavelet Transform
RMSE	Root Mean Absolute Error
TGARCH	Threshold Generalized Autoregressive Conditionally Heteroscedastic
QMF	Quadrature Mirror Filter
Q-Q	Quantile-Quantile
\mathbb{R}	Real Numbers
S&P500	Standard & Poor's 500
SARIMA	Seasonal Autoregressive Integrated Moving Average
SDF	Spectral Density Function
SETAR	Self-exciting Threshold Autoregressive
SIC	Schwarz Information Criterion
SMI	Swiss Market Index
STAR	Smooth Transition Autoregressive
STFT	Short-Time Fourier Transform

STL	Seasonal Trend Decomposition by Loess
SVM	Support Vector Machine
TAR	Threshold Autoregressive
WFT	Windowed Fourier Transform
var	Variance

CHAPTER 1

INTRODUCTION

Time series can be defined as recording of observations of selected variable with respect to different time points. One can assume a time series as a signal. There are two domains which are named frequency and time to analyze the series. In some sense one needs to look both frequency and time domain to catch real characteristic of data.

Time series analysis is very important issue according to many scientific fields such as biology, geology, astronomy, economy, meteorology, medicine, finance etc. . In other words, time series analysis is used by engineers and scientists who are dealing with applied sciences. Especially financial time series like stock market data are in the area of our interest in this thesis. Tsay in [68] points out that financial theory includes undetermined factors. Therefore we need to understand some statistical theories and methods to analyze the data. Then one can make a forecast of time series whose underlying mechanism is understood.

Moreover, correlation analysis is linked to the time-domain analysis. Using correlations of adjacent points in time and parametric functions such as SARIMA, researchers try to predict the future values of linear time series. Using the SARIMA function is also explained as multiplicative model where data set is modeled by differential equation operators. Additive model is also used to analyze time series, especially in financial and economic time series. Then seasonal, trend and noise parts are separated to understand time series in basic components. In order to analyze nonlinear time series, one can use again parametric methods such as LSTAR. Variance analysis includes the method of GARCH models.

Spectral analysis and wavelet analysis are example of the frequency-domain analysis. Following the wavelet analysis, multiresolution analysis (MRA) can be done as well. Frequency-domain analysis is basically used to detect the periodicity of the time series. *Fourier* transform is very famous in frequency domain analysis and it has been applying for several years. On the other hand, wavelet transforms can be regarded as new issue since modern wavelet theory has taken shape after middle 1950's. Wavelets are recently popular issue due to providing time-scale analysis. Therefore one can analyze any time series with desired scales and time intervals.

There are many fields where wavelet theory is used. Some of the applications can be counted as: applying scale based decomposition to understand real characteristic of

data, signal de-noising and smoothing, data compression etc.. Financial time series analysis by using multiresolution analysis is one of scopes of this thesis.

In [17] and in [64] wavelet analysis is handled with dilation equations to illustrate construction of multiresolution analysis. Lyubushin studied in [41] multidimensional wavelet analysis applied on geophysical monitoring time series in the Moscow region. Aim of study is illustrating that wavelet transform is better than Fourier transform such that wavelet analysis catches both high and low frequency components in selected time intervals. Similarly, study of Masset in [46] also shows that wavelet transform has several advantages compared to Fourier transform. Crowley clearly summarized fundamentals of wavelets and applications to economic and finance areas in [20]. MRA is applied to Nokia share volatility by Vuorenmaa in [71]. In [76] In [5] Aloui and Nguyen combine the global Hurst exponent and Morlet wavelet multi-resolution analysis in order to understand behavior of six different weekly stock markets data in Mediterranean region. In [56] Ramsey et al. analyzed stock market data which is complex time series. They concluded that wavelets are useful to detect non-detectable structures in system. Yousefi, Weinreich and Reinarz applied multiresolution analysis in order to understand dynamics of wavelet-based prediction by using oil prices. Ababneh, Al Wadi and Ismail showed that ARIMA modeling with wavelet transform gives better accuracy results compared to original ARIMA fitting by using Amman stock market (Jordan) in [1]. Similarly, Al Wadi et al. shows same result in [72]. In [49] Nouri, Oryoie and Fallahi use ARIMA–GARCH model to calculate one-step-ahead forecasts of monthly gold returns. Data is separated into different scales and forecasts of each part is added together. Then forecasting with separated data increases forecasting performance compared to forecasting without wavelets. Rocha, Paredes, Carvalho, Henriques and Harris in [58] studied how to combine wavelet analysis and neural network to predict acute hypotensive episodes data.

In Chapter 2, some terminologies used in time series analysis will be defined with some examples. Chapter will give to reader clear definitions to understand time series concept.

In Chapter 3, subjects that are related to frequency domain analysis will be covered. Two main title are Fourier transform and wavelet transform in frequency domain. We will discuss why do we need any other method than Fourier transform in frequency domain analysis.

In Chapter 4 empirical results will be given by using S&P500 historical data. Descriptive statistics, results of linear model fittings, spectrums, multiresolution analysis and some other modeling methods will be covered. Models will be compared according to information criteria values and results of accuracy measures.

Finally in Chapter 5 consequences of this study will be given with possible future works.

CHAPTER 2

LINEAR TIME SERIES AND HETEROSCEDASTIC MODELS

In this chapter firstly time series basic issues and definitions of linear time series models like $\text{AR}(p)$, $\text{MA}(q)$, $\text{ARMA}(p, q)$, $\text{ARIMA}(p, d, q)$ and $\text{SARIMA}(p, d, q) \times (P, D, Q)_s$ will be considered. Numerical examples will be given to understand characteristic of models. Structure, properties and example of linear model forecasting will be covered.

Secondly, definitions and examples of heteroscedastic models like ARCH, GARCH etc., will be given to consider volatility issues.

Finally other methods which are used in time series analysis will be given shortly with definitions and some examples.

2.1 Introduction

Time series analysis is important for understanding the essence of the data. Correlations of observed data are used to catch the characteristic of series and then some operations are applied to series according to correlations.

Let's give some important definitions related to time series using the methodology of Shumway and Stoffer mentioned in [62].

Definition 2.1 (Sample Autocovariance). The sample autocovariance of x_t is

$$\hat{\gamma}(h) = \frac{1}{N} \sum_{t=1}^{N-h} (x_{t+h} - \bar{x})(x_t - \bar{x}), \quad (2.1)$$

where \bar{x} is the mean of x_t and $\hat{\gamma}(h) = \hat{\gamma}(-h)$ for $h = 0, 1, \dots, N - 1$.

Definition 2.2 (Theoretical Autocovariance). The theoretical autocovariance function is calculated by the sample autocovariance function [62] and written as

$$\gamma(h) = \mathbb{E}[(x_{t+h} - \mu)(x_t - \mu)], \quad (2.2)$$

where μ is the mean and $h = |s - t|$ is the lag for all s, t . Other notations for the autocovariance function can be listed as $\gamma_x(s, t)$, $\gamma(s, t)$, $\gamma(t, t-h)$, $\gamma(t+h, t)$, $\text{cov}(x_s, x_t)$, $\text{cov}(x_{t+h}, x_t)$ and $\text{cov}(x_h, x_0)$.

Definition 2.3 (Strictly Stationary Time Series). Strictly stationary time series is a time series which ensures the following condition,

$$\begin{aligned} \mathbb{P}\{x_{t_1} \leq c_1, x_{t_2} \leq c_1, \dots, x_{t_k} \leq c_k\} \\ = \mathbb{P}\{x_{t_1} + h \leq c_1, x_{t_2} + h \leq c_1, \dots, x_{t_k} + h \leq c_k\}. \end{aligned} \quad (2.3)$$

Definition 2.4 (Weakly Stationary Time Series). Weakly stationary time series is a finite variance process that has following conditions,

1. The mean of x_t is $\mu_t = \mathbb{E}[x_t] = \mu(\text{constant}) = \int_{-\infty}^{\infty} xf_t(x)dx$
2. The autocovariance $\gamma(s, t) = \mathbb{E}[(x_s - \mu)(x_t - \mu)]$ depends on only s and t through their difference $h = |s - t|$ which is called as lag.

Claim 2.1. If the time series x_t is stationary, then first difference of x_t is given as $y_t = \nabla x_t$.

Proof. Expectation of y_t is zero and autocovariance of y_t is independent of time as follows;

- i. $\mathbb{E}[y_t] = \mathbb{E}[x_t - x_{t-1}] = \mu - \mu = 0$
- ii. $\gamma_y(h) = \text{cov}(x_{t+h} - x_{t+h-1}, x_t - x_{t-1}) = \gamma_x(h) - \gamma_x(h+1) - \gamma_x(h-1) + \gamma_x(h)$

depends on only the lag h . □

If one is interested in the signal to be stationary, it is necessary to remove trend and seasonal part in the signal. In other words, time series is stationary if there is no systematic change in the mean and the variance with having no periodic variations. The correlogram that will be defined in Definition 2.5, cuts off after 2 or 3 lags if the time series is stationary.

Definition 2.5 (Autocorrelation Function). Autocorrelation function can be written as the autocovariance function that is divided by square root of multiplication of variances of series according to s and t .

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}} \quad (2.4)$$

Autocorrelation function satisfies that $-1 \leq \rho(s, t) \leq 1$. It can be proved by Cauchy-Schwartz inequality [62].

Definition 2.6 (Backward Shift Operator and Forward Shift Operator). Backward shift operator and forward shift operator are written respectively for first difference and for k th difference as

$$Bx_t = x_{t-1} \quad \text{and} \quad B^k x_t = x_{t-k}, \quad (2.5)$$

$$Fx_t = x_{t+1} \quad \text{and} \quad F^k x_t = x_{t+k}. \quad (2.6)$$

Then difference operator can be written by using the backward shift operator as $\nabla = (1 - B)$ proved by follows,

$$\nabla x_t = x_t - x_{t-1} = x_t - Bx_t = (1 - B)x_t. \quad (2.7)$$

It is also valid or higher dimensions such as $\nabla^d = (1 - B)^d$, $d \in \mathbb{N}_0$.

Definition 2.7 (Autoregressive Process (AR(p))). Autoregressive process is constructed on sum of past values of time series x_t and Gaussian white noise w_t . Autoregressive process of order p is

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \quad (2.8)$$

where x_t is stationary and $\phi_1, \phi_2, \dots, \phi_p$ are constant ($\phi_p \neq 0$) and $w_t \sim w_n(0, \sigma_w^2)$ is a white noise and is particularly assumed to be independent and identically distributed with $\mathcal{N}(0, \sigma_w^2)$.

One can rewrite Equation 2.8 by replacing x_t with $x_t - \mu$ in case the mean μ is different from zero as follows;

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + \cdots + \phi_p(x_{t-p} - \mu) + w_t. \quad (2.9)$$

After some arrangements, Equation 2.9 can be written as

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \quad (2.10)$$

where $\alpha = \mu(1 - \phi_1 - \phi_2 - \cdots - \phi_p)$. Finally we can write the p -order autoregressive process AR(p) in compact form as

$$\phi(B)x_t = w_t, \quad (2.11)$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ is AR operator of order p .

Definition 2.8 (Moving Average Model(MA(q))). Moving average model of order q is

$$x_t = \theta_0 w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}, \quad (2.12)$$

where θ_q is different from zero. In simple form it can be written as

$$x_t = \theta(B)w_t, \quad (2.13)$$

where $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$.

2.2 Autoregressive Moving Average (ARMA) Models

Time series x_t can be written as sum of autoregressive part and moving average part according to ARMA model of order p and q respectively.

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}. \quad (2.14)$$

Simply we can write Equation 2.14 as

$$\phi(B)x_t = \theta(B)w_t. \quad (2.15)$$

Definition 2.9 (Causal ARMA(p, q) Process). If the model predicts backward observations, then it is not causal but it can be stationary process. An ARMA(p, q) model is causal in case the time series can be written as

$$x_t = \sum_{i=0}^{\infty} \psi_i w_{t-i} = \psi(B)w_t, \quad (2.16)$$

where $\psi(B) = \sum_{i=0}^{\infty} \psi_i B^i$ and $\sum_{i=0}^{\infty} |\psi_i| < \infty$ for $\psi_0 = 1$.

Definition 2.10 (Invertible ARMA(p, q) Process). An ARMA(p, q) model is invertible in case the time series is written as

$$w_t = \sum_{i=0}^{\infty} \pi_i x_{t-i} = \pi(B)x_t, \quad (2.17)$$

where $\pi(B) = \sum_{i=0}^{\infty} \pi_i B^i$ and $\sum_{i=0}^{\infty} |\pi_i| < \infty$ for $\pi_0 = 1$.

Definition 2.11 (AR and MA Polynomials). AR and MA polynomials are given respectively as

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \quad (2.18)$$

and

$$\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q, \quad (2.19)$$

where z is complex number and ϕ_p and θ_p are different than zero.

Definition 2.12 (Causality of ARMA(p, q) Process). An ARMA(p, q) model is causal if and only if $\phi(z) \neq 0$ for $|z| \leq 1$, i.e. zeros of $\phi(z)$ lie outside of unit circle. One can achieve coefficients of Equation 2.16 by solving Equation 2.20,

$$\psi(z) = \sum_{i=0}^{\infty} \psi_i z^i = \frac{\theta(z)}{\phi(z)}, \quad (2.20)$$

where $|z| \leq 1$.

Definition 2.13 (Invertibility of ARMA(p, q) Process). An ARMA(p, q) model is invertible if and only if $\theta(z) \neq 0$ for $|z| \leq 1$, i.e. $\theta(z) = 0$ for $|z| > 1$. One can achieve coefficients of Equation 2.17 by solving Equation 2.21,

$$\pi(z) = \sum_{i=0}^{\infty} \pi_i z^i = \frac{\phi(z)}{\theta(z)}, \quad (2.21)$$

where $|z| \leq 1$.

Example 2.1 (ARMA(2, 2) Process). Let's write ARMA(2, 2) process with its coefficients according to autoregressive and moving average parts as

$$x_t - 0.2x_{t-1} - 0.15x_{t-2} = w_t + 0.7w_{t-1} + 0.12w_{t-2}.$$

After finding roots of $\phi(B)$ and $\theta(B)$, we can write AR and MA polynomials respectively as

$$\phi(z) = (1 - 0.5z)(1 + 0.3z),$$

and

$$\theta(z) = (1 + 0.4z)(1 + 0.3z).$$

It is seen that they have same factors as $(1 + 0.3z)$. As a result we can cancel out this factor which is mentioned as *parameter redundancy* by Shumway and Stoffer in [62]. Finally we get ARMA(1, 1) process as

$$(1 - 0.5B)x_t = (1 + 0.4B)w_t,$$

where $\phi(B) = (1 - 0.5B)$ and $\theta(z) = (1 + 0.4B)$. According to Definition 2.12 and Definition 2.13, process is causal and invertible. For $\phi(z) = 0$, we have $z = \frac{1}{0.5}$ and $\left|\frac{1}{0.5}\right| > 1$ (outside of the unit circle), so process is causal (as well as stationary). For $\theta(z) = 0$, we have $z = -\frac{1}{0.4}$ and $\left|-\frac{1}{0.4}\right| > 1$ (outside of the unit circle), so process is invertible. Hence, we can write the process in terms of ψ -coefficients and π -coefficients. Firstly, for ψ -coefficients we have

$$x_t = \frac{\theta(B)}{\phi(B)}w_t = \psi(B)w_t \Rightarrow \phi(B)\psi(B) = \theta(B),$$

and

$$(1 - 0.5B)(\psi_0 + \psi_1B + \psi_2B^2 + \dots) = (1 + 0.4B),$$

where $\psi_0 = 1$. If we equalize the coefficients of B as $\psi_1 - 0.5\psi_0 = 0.4$, ψ_1 is founded as 0.9. For other coefficients we have $\psi_j - 0.5\psi_{j-1} = 0$ and after j iterations we get $\psi_j = (0.9)(0.5)^{j-1}$ for $j = 1, 2, \dots$. So the time series x_t is written in the form as

$$x_t = w_t + (0.9) \sum_{i=1}^{\infty} (0.5)^{i-1} w_{t-i}.$$

Secondly, for π -coefficients we have

$$w_t = \frac{\phi(B)}{\theta(B)}x_t = \pi(B)x_t \Rightarrow \theta(B)\pi(B) = \phi(B),$$

and

$$(1 + 0.4B)(\pi_0 + \pi_1B + \pi_2B^2 + \dots) = (1 - 0.5B),$$

where $\pi_0 = 1$. Then applying similar steps as we did in ψ -coefficients, π -coefficients are written as $\pi_j = (-0.9)(-0.4)^{j-1}$ for $j = 1, 2, \dots$. Finally the time series x_t is written in the form as following,

$$w_t = x_t - (0.9) \sum_{i=1}^{\infty} (-0.4)^{i-1} x_{t-i}.$$

2.3 Autoregressive Integrated Moving Average (ARIMA) Models

In addition to autoregressive and moving average parts, model would involve integration part, i.e. differencing part. Then model is called as an autoregressive integrated moving average ARIMA(p, d, q) process and it is of the form;

$$\phi(B)\nabla^d x_t = \phi(B)(1 - B)^d x_t = \theta(B)w_t, \quad (2.22)$$

where $\phi(B)$ is stationary AR operator of order p , $\theta(B)$ is invertible MA operator of order q and $(1 - B)^d$ is difference operator of order d . Operators can be written explicitly as

$$\begin{aligned}\phi(B)(1 - B)^d &= \varphi(B) = \varphi_0 + \varphi_1 B + \cdots + \varphi_{p+d} B^{p+d} \\ &= (1 - \phi_1 B - \cdots - \phi_p B^p)(1 - B)^d,\end{aligned}$$

and

$$\theta(B) = \theta_0 + \theta_1 B + \cdots + \theta_q B^q.$$

If $\mathbb{E}[\nabla^d x_t] = \mu$, then we consider the model in more general form and it can be written as [62]

$$\phi(B)(1 - B)^d x_t = \beta_0 + \theta(B)w_t, \quad (2.23)$$

where $\beta_0 = (1 - \phi_1 - \cdots - \phi_p)\mu$. In addition we can define random shock form (MA(∞)) and inverted form (AR(∞)) respectively as [62]

$$x_t = \varphi^{-1}(B)\theta(B)w_t = \psi(B)w_t, \quad (2.24)$$

and

$$w_t = \theta^{-1}(B)\varphi(B)x_t = \pi(B)x_t, \quad (2.25)$$

where $\varphi^{-1} = \phi^{-1}(B)(1 - B)^{-d}$.

Taking difference with needed order of the non-stationary time series makes the time series stationary. Many analyses are done via using stationary time series according to see de-trended properties of the signal. It is better to use differenced time series for forecasting issue. AR part shows how strongly past observations affects the present observations. MA part shows model present using past errors of prediction.

Example 2.2 (ARIMA(2, 2, 2) Process). If we consider an ARIMA(2, 2, 2) process with general coefficients, operators can be written as follows;

$$\phi(B) = (1 - \phi_1 B - \phi_2 B^2),$$

$$\nabla^2 = (1 - B)^2 = (1 - 2B + B^2),$$

$$\theta(B) = (1 + \theta_1 B + \theta_2 B^2),$$

and

$$\varphi(B) = \phi(B)\nabla^2$$

$$= (1 - 2B + B^2) - \phi_1(B - 2B^2 + B^3) - \phi_2(B^2 - 2B^3 + B^4).$$

Explicitly the model can be written as;

$$\begin{aligned}x_t - 2x_{t-1} + x_{t-2} - \phi_1 x_{t-1} + 2\phi_1 x_{t-2} - \phi_1 x_{t-3} - \phi_2 x_{t-2} + 2\phi_2 x_{t-3} - \phi_2 x_{t-4} \\ = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2},\end{aligned}$$

or

$$\begin{aligned}x_t - x_{t-1}(2 + \phi_1) + x_{t-2}(1 + 2\phi_1 - \phi_2) - x_{t-3}(\phi_1 - 2\phi_2) - \phi_2 x_{t-4} \\ = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2}.\end{aligned}$$

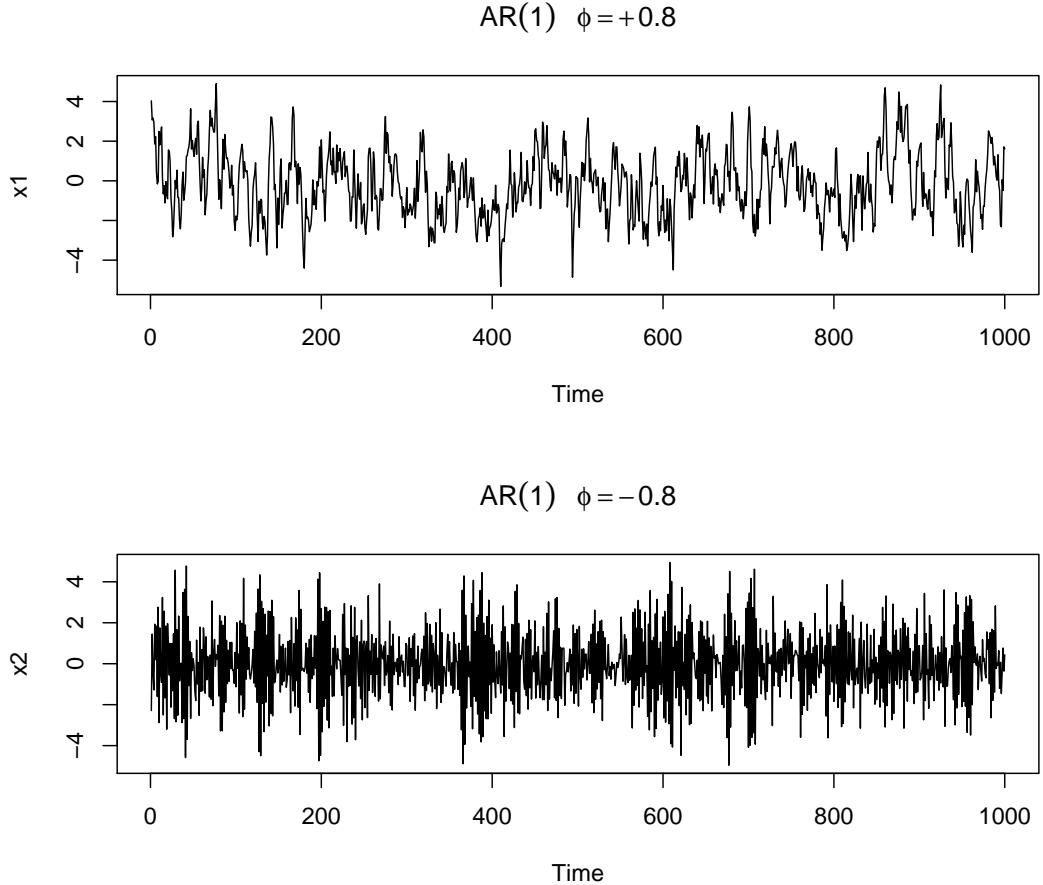


Figure 2.1: Simulated AR(1) Processes.

Random shock form and inverted form can be found respectively by setting following equations as,

$$\phi(B)(1 - B)^2\psi(B) = \varphi(B)\psi(B) = \theta(B),$$

and

$$\theta(B)\pi(B) = \varphi(B).$$

After setting the equalities, one can find the coefficients with respect to our time series in order to write random shock form and inverted form.

Example 2.3 (Simulated ARIMA Processes). Let's consider AR(1), AR(2), MA(1), MA(2), ARMA(1, 1), ARMA(2, 2), ARIMA(1, 1, 1) and ARIMA(2, 1, 1) with arbitrary coefficients given in the plots. ACF and PACF plots are also given for some processes.

AR(1) Processes

In Figure 2.1 first simulated series looks like financial data without trend, especially daily closing prices without trend. The one with negative coefficient looks like return of a financial data, especially first difference of logarithmic stock market data.

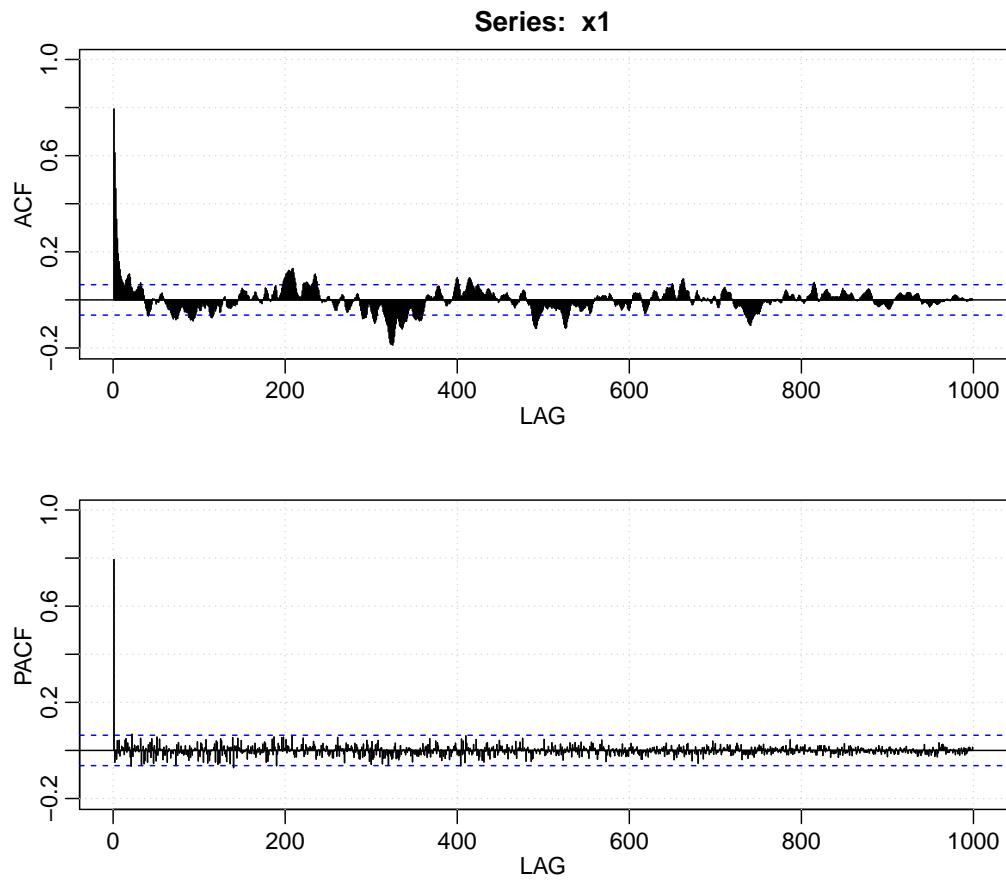


Figure 2.2: ACF and PACF Plots of AR(1) Process with First 1000 Observations where $\phi = +0.8$.

ACF and PACF plots of simulated AR(1) processes are given as follows in Figure 2.2, Figure 2.3 and Figure 2.4:

AR(2) Process

In Figure 2.5 simulated AR(2) process and in Figure 2.6 its ACF and PACF are given with respect to given coefficients.

MA(1) Process

In Figure 2.7 simulated MA(1) process and in Figure 2.8 its ACF and PACF are given with respect to given coefficient.

MA(2) Process

In Figure 2.9 simulated MA(2) process and in Figure 2.10 its ACF and PACF are given with respect to given coefficients.

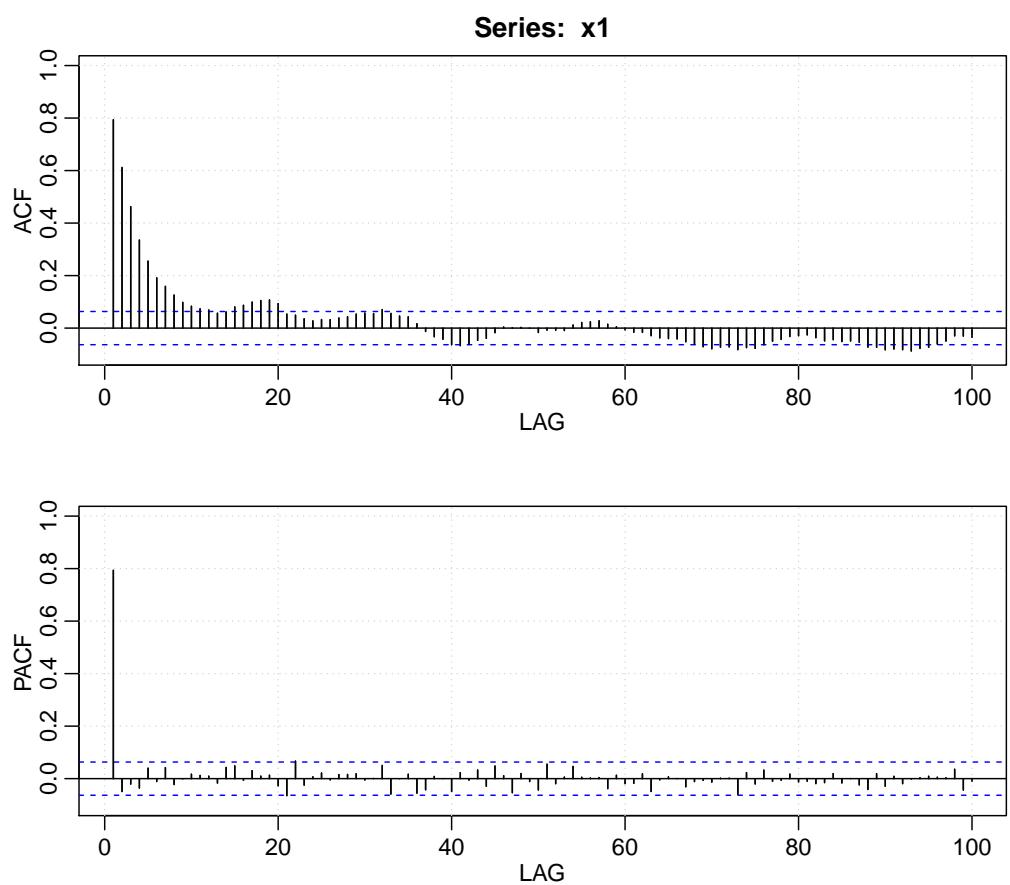


Figure 2.3: ACF and PACF Plots of AR(1) Process with First 100 Observations where $\phi = +0.8$.

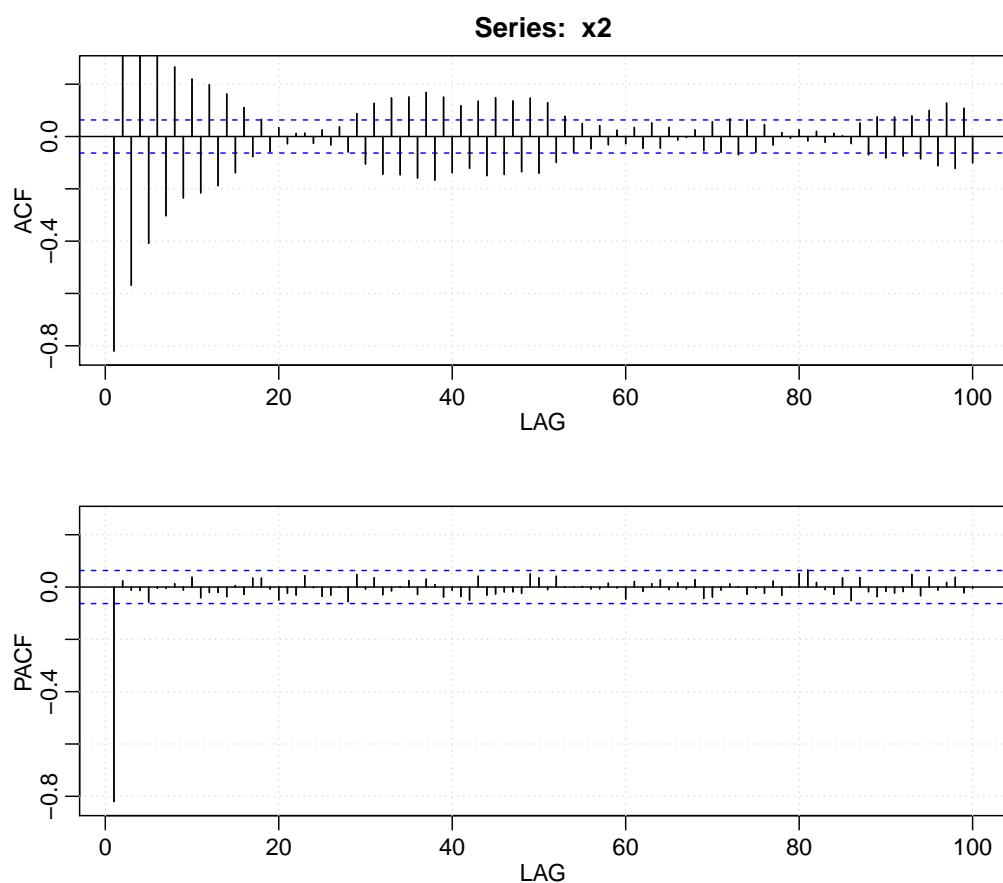


Figure 2.4: ACF and PACF Plots of AR(1) Process with First 100 Observations where $\phi = -0.8$.

$\text{AR}(2) \quad \phi_1 = 0.5 \quad \phi_2 = 0.4$

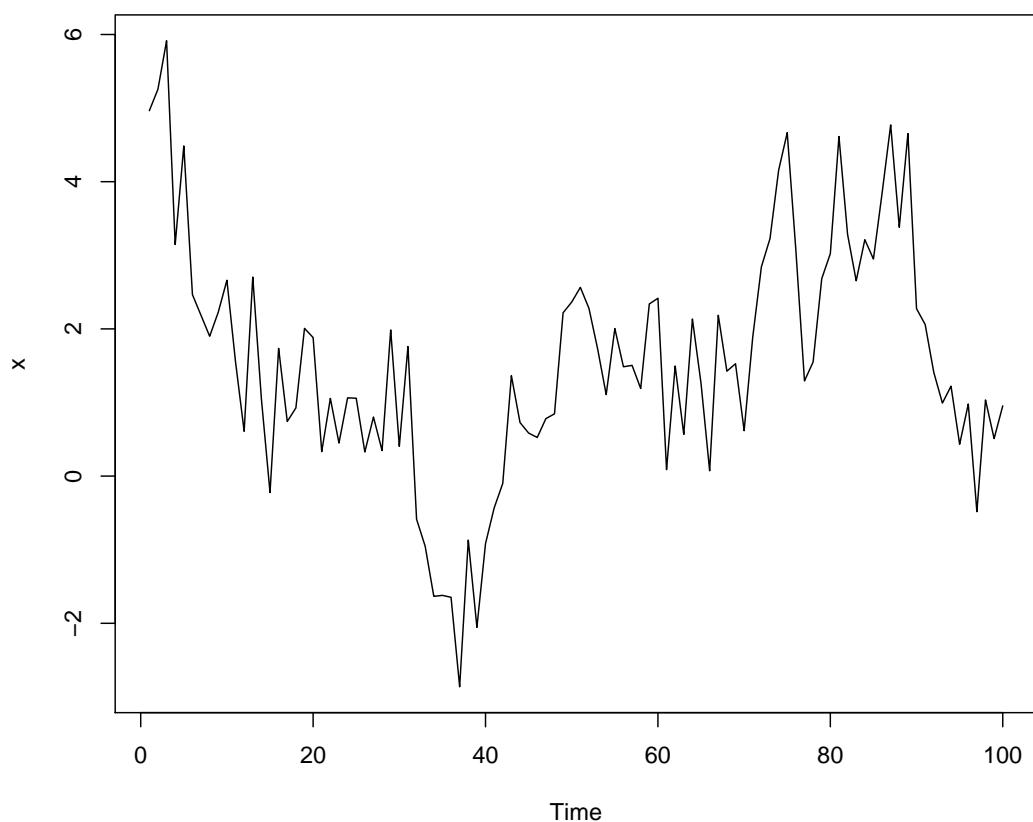


Figure 2.5: Simulated $\text{AR}(2)$ Process.

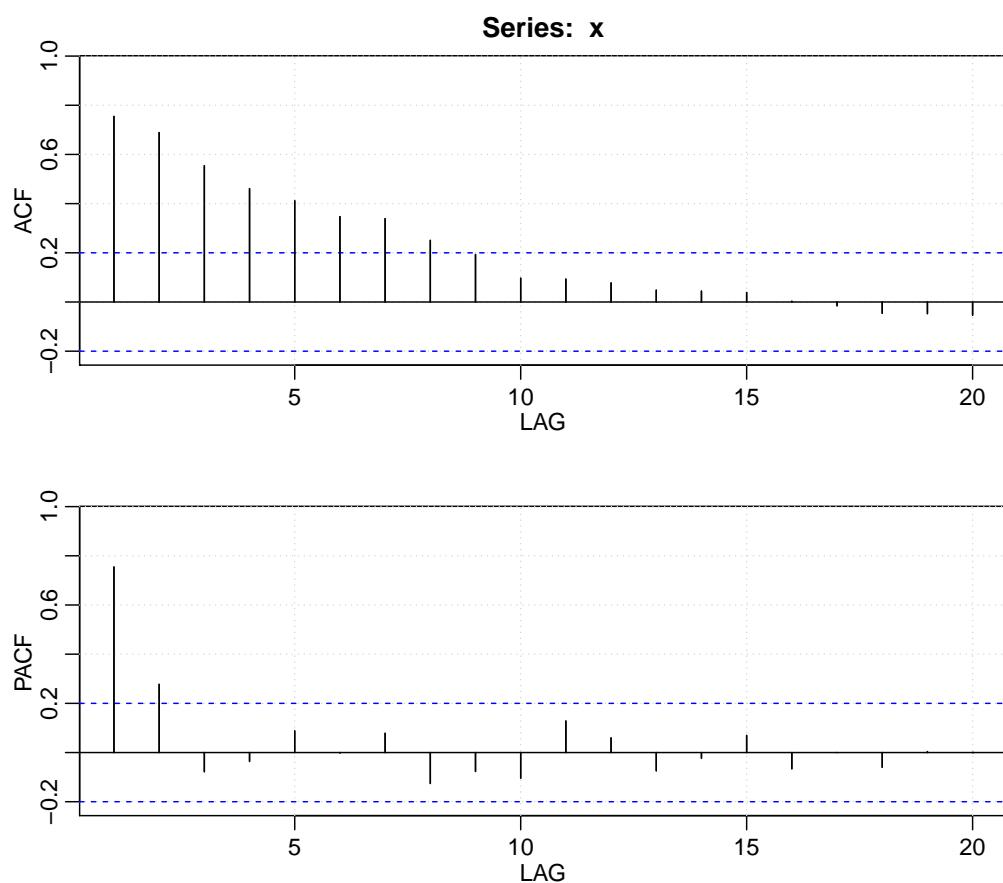


Figure 2.6: ACF and PACF Plots of AR(2) Process with First 100 Observations where $\phi_1 = 0.5$ and $\phi_2 = 0.4$.

MA(1) $\theta = 0.5$

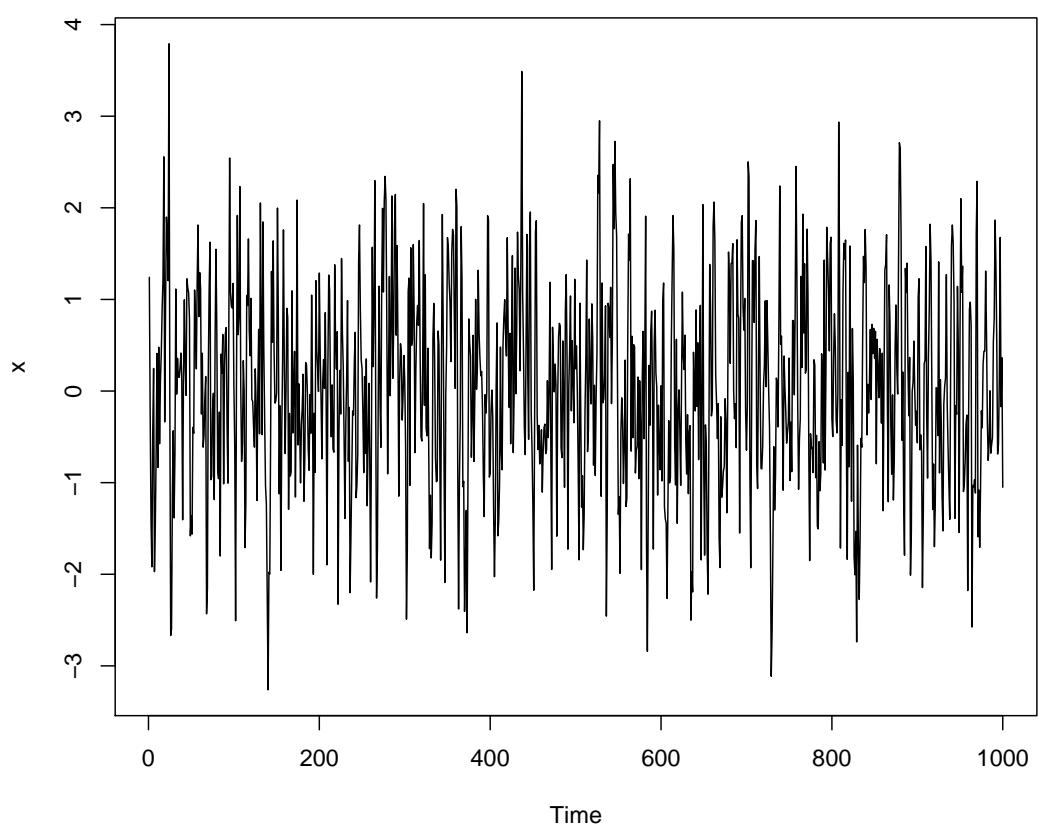


Figure 2.7: Simulated MA(1) Process.

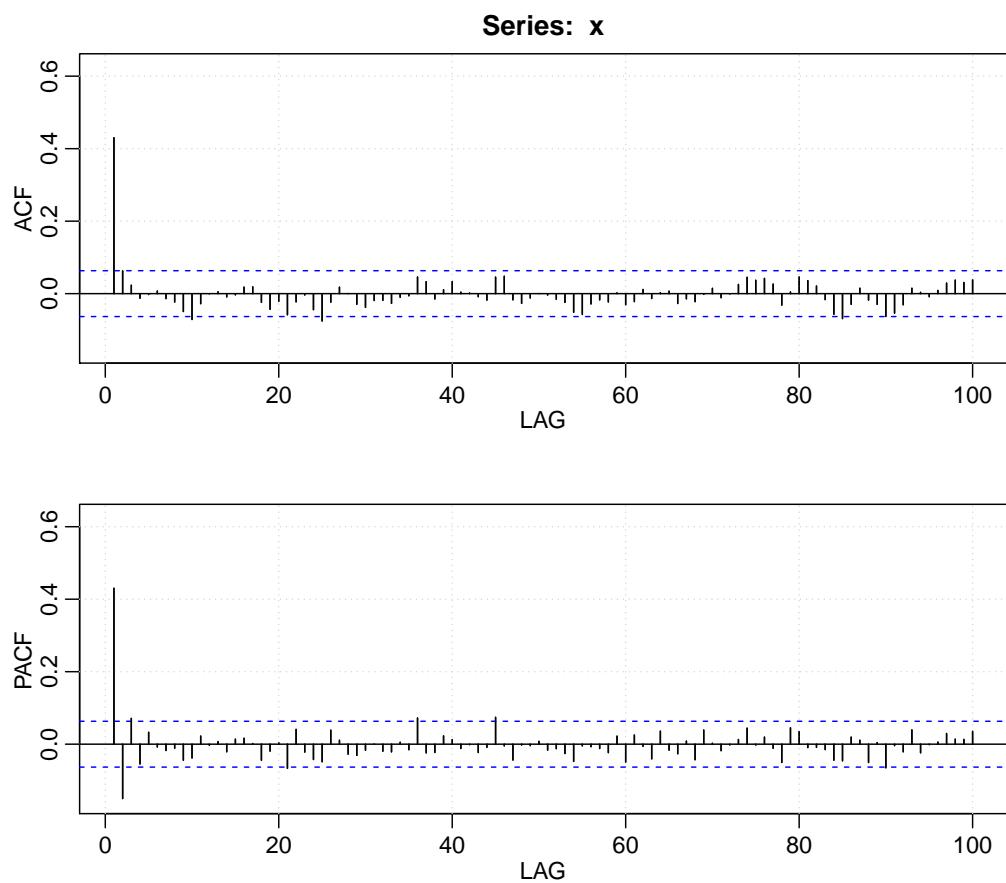


Figure 2.8: ACF and PACF Plots of MA(1) Process with First 100 Observations where $\theta = 0.5$.

$\text{MA}(2) \quad \theta_1 = -0.5 \quad \theta_2 = -0.9$

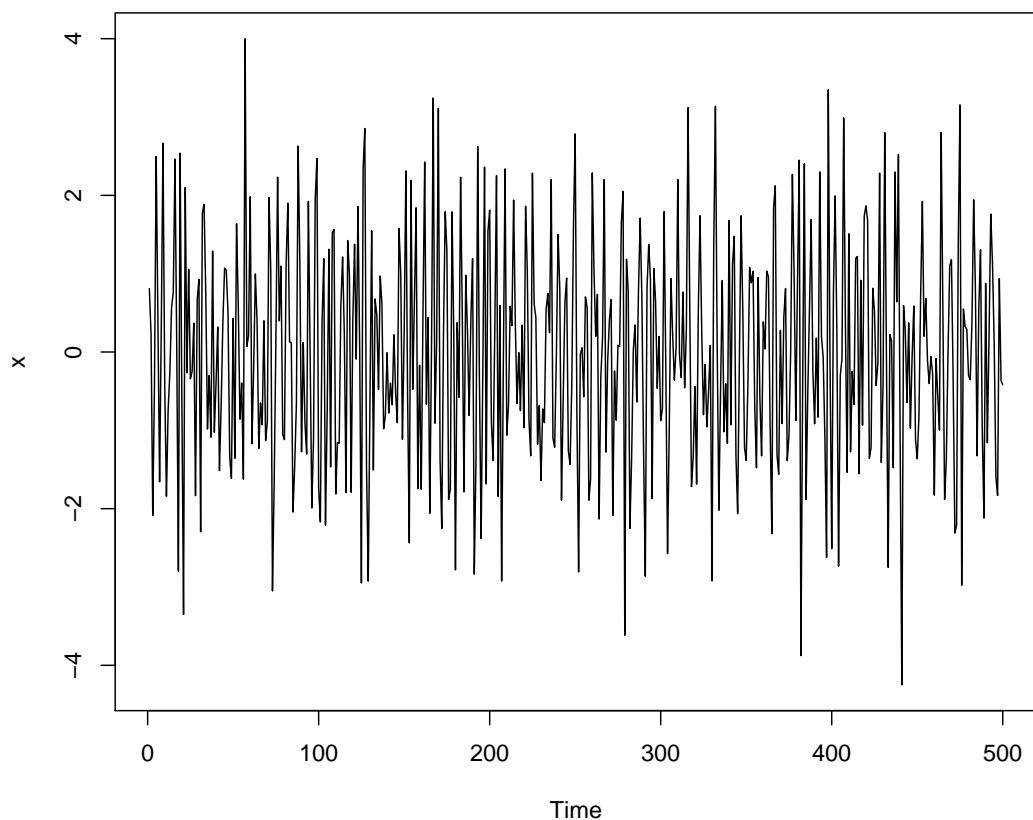


Figure 2.9: Simulated MA(2) Process.

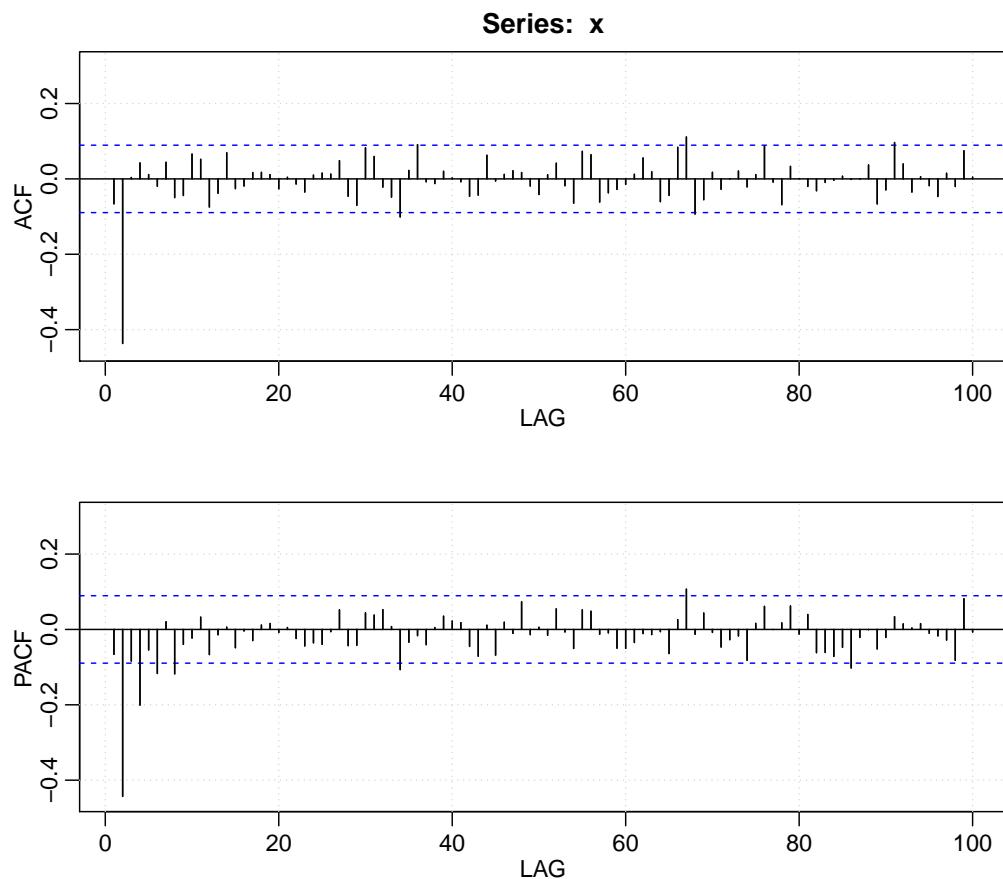


Figure 2.10: ACF and PACF Plots of MA(2) Process with First 100 Observations where $\theta_1 = -0.5$ and $\theta_2 = -0.9$.

ARMA(2, 2) $\phi_1 = 0.5$ $\phi_2 = 0.4$ $\theta_1 = -0.5$ $\theta_2 = -0.9$

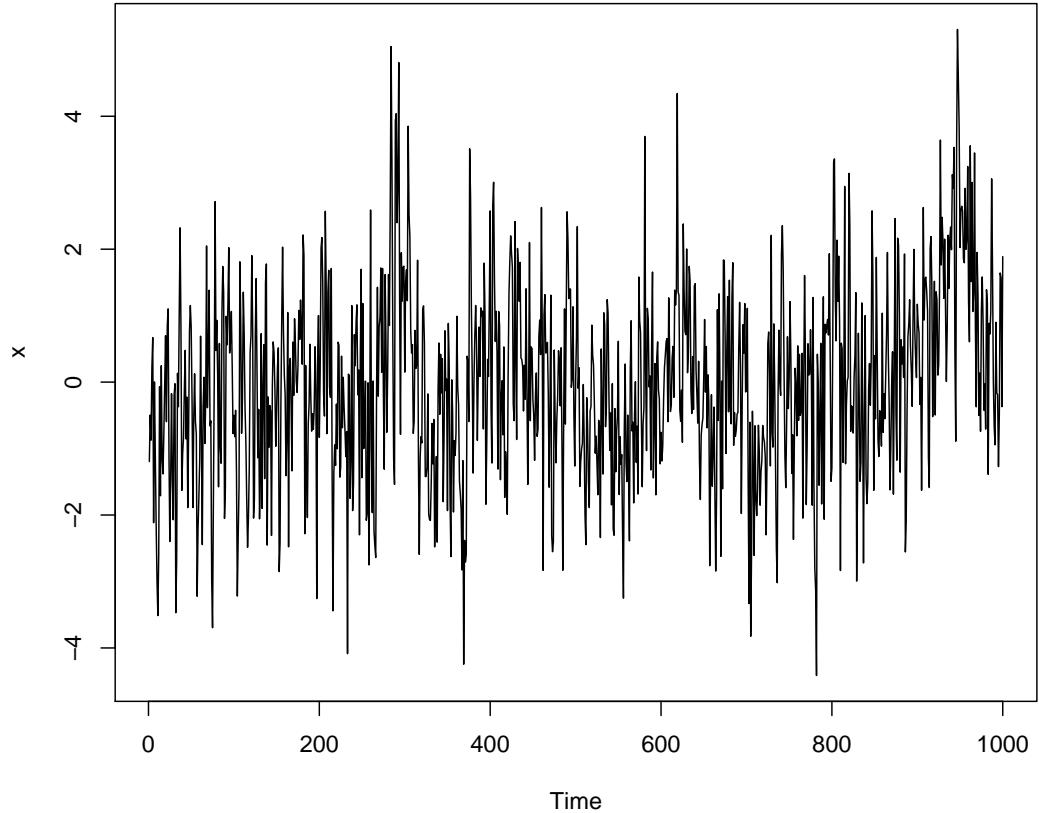


Figure 2.11: Simulated ARMA(2, 2) Process.

ARMA(2, 2) Process

In Figure 2.11 simulated ARMA(2, 2) process and in Figure 2.12 its ACF and PACF are given with respect to given coefficients.

ARIMA(1, 1, 1) Processes

In Figure 2.13 simulated ARIMA(1, 1, 1) processes and their ACF and PACF are given with respect to given coefficients.

After fitting the ARIMA model to simulated series where whose coefficients are given as $\phi = 0.5$ and $\theta = -0.4$, coefficients are calculated as $\phi = 0.5976$ and $\theta = -0.5040$ in R program. Sigma square, log likelihood and AIC are calculated as 0.9876, -1412.69 and 2831.39 respectively.

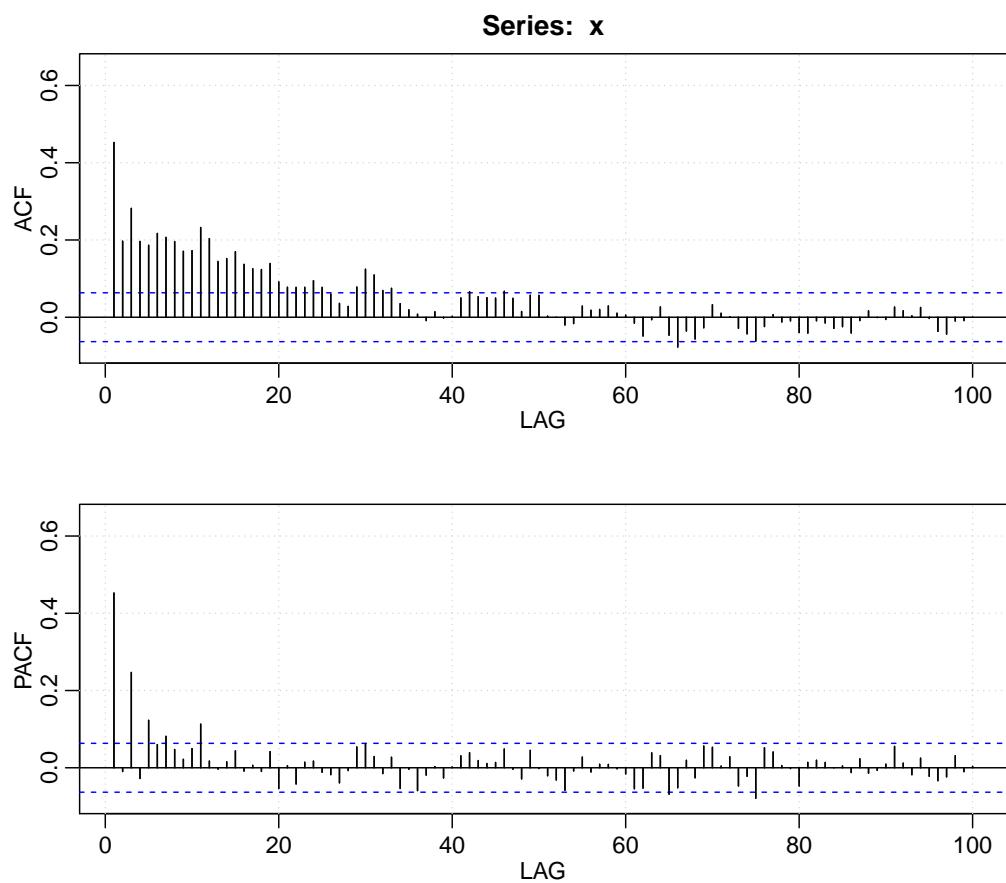


Figure 2.12: ACF and PACF Plots of ARMA(2, 2) Process with First 100 Observations where Coefficients are Given as in Figure 2.11.

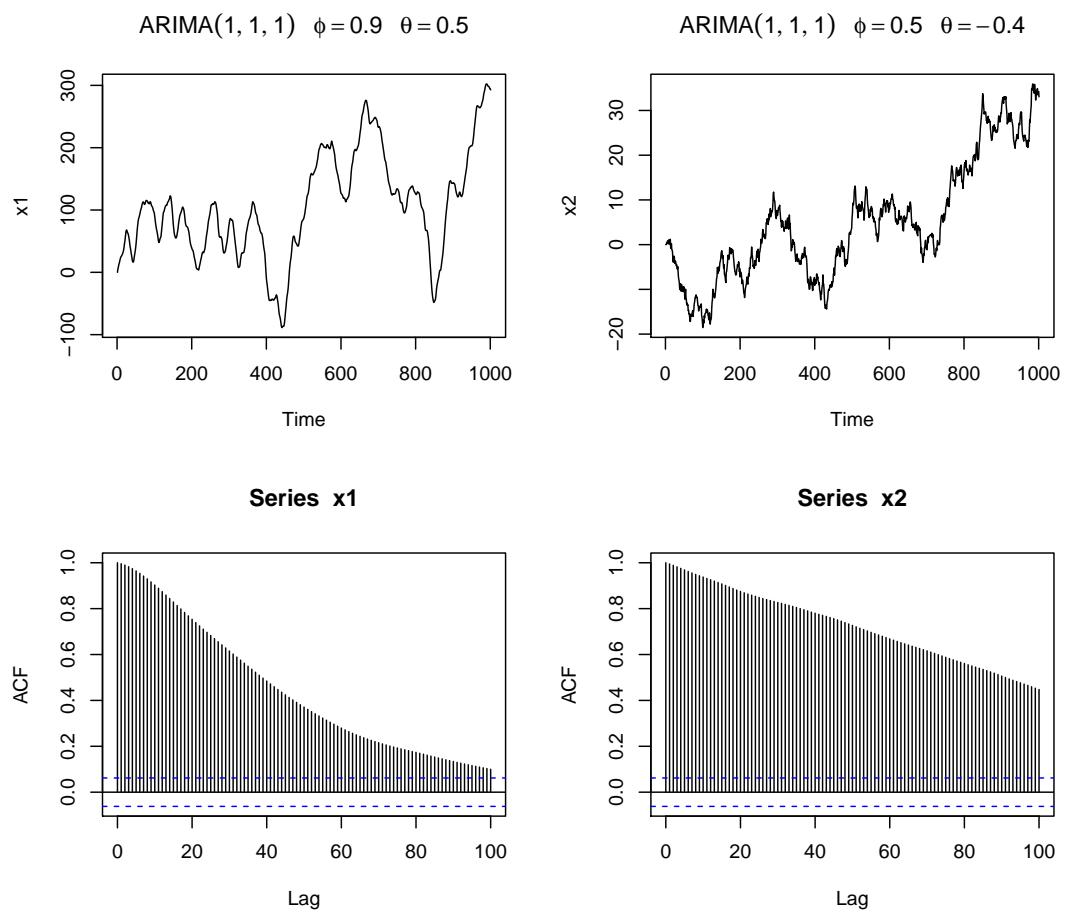


Figure 2.13: Simulated ARIMA(1, 1, 1) Processes and related ACF plots.

2.4 Multiplicative Seasonal Autoregressive Integrated Moving Average (SARIMA) Models

Seasonality is important subject for many time series such as economic and financial data sets. Forecasting depends on past values. If past values are in seasonal components, model can be modified to SARIMA model.

Definition 2.14 (Seasonal Operator). The seasonal autoregressive operator of order P and the seasonal moving average operator of order Q are given respectively as

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^2 s - \cdots - \Phi_P B^P s, \quad (2.26)$$

and

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^2 s + \cdots + \Theta_Q B^Q s, \quad (2.27)$$

where s is seasonal period.

Definition 2.15 (Seasonal Difference). The simplifying operator (or seasonal difference) of order D is defined as

$$\nabla_s^D = (1 - B^s)^D, \quad (2.28)$$

where s is seasonal period.

Definition 2.16 (SARIMA Model). By using Definition 2.14 and Definition 2.15, the SARIMA model, i.e. $\text{ARIMA}(p, d, q) \times (P, D, Q)_s$ model can be defined as follows

$$\Phi_P(B^s)\phi(B)\nabla_s^D \nabla^d x_t = \Theta_Q(B^s)\theta(B)w_t. \quad (2.29)$$

Let's consider $\text{ARIMA}(0, 0, 0) \times (1, 0, 1)_{12}$ model. If one writes it explicitly, it will be seen that model is same as $\text{ARMA}(12, 12)$.

Example 2.4 ($\text{ARIMA}(2, 3, 1) \times (1, 2, 1)_{12}$ Process).

$$(1 - \Theta_1 B^{12})(1 - \theta_1 B - \theta_2 B^2)(1 - B)^3(1 - B^{12})^2 x_t = (1 + \Phi_1 B^{12})(1 + \phi_1 B)w_t.$$

2.5 Forecasting

Basically aim of forecasting or prediction is calculating future values approximately by using given sample data set. We can liken the fog to noise in the time series. If the fog is less, visibility of the forward will be much better. Forecasting is getting more complex if the noise is higher than the original signal. Choosing the correct model and estimating correct parameters are very important for prediction.

There are several methods to predict future values of the time series by using observed data set. Seasonal moving average, exponential smoothing (or weighted moving average), ARIMA [18], state space [31] [37], neural network [8], Bayesian and wavelets are some methods which are used to forecast desired future values of the time series.

Most of them are applied on theoretical basis of minimum mean squared error to minimize the error between real values and predicted values.

Consider ARIMA(p, d, q) process and let's write observation at time $t + 1$ as

$$x_{t+1} = \varphi_1 x_t + \varphi_2 x_{t-1} + \cdots + \varphi_{p+d} x_{t-p-d+1} + w_{t+1} + \theta_1 w_t + \cdots + \theta_q w_{t-q+1}, \quad (2.30)$$

where $\varphi_0 = \Theta_0 = 0$. In general we can write Equation 2.30 at time $t + l$ (l-lead ahead) by writing $t = t + l - 1$ as

$$\begin{aligned} x_{t+l} &= \varphi_1 x_{t+l-1} + \varphi_2 x_{t+l-2} + \cdots \\ &\quad + \varphi_{p+d} x_{t+l-p-d} + w_{t+l} + \theta_1 w_{t+l-1} + \cdots + \theta_q w_{t+l-q}. \end{aligned} \quad (2.31)$$

In addition, infinite weighted sum of x_{t+l} can be written by putting $t + l - k$ instead of i as

$$x_{t+l} = \sum_{i=0}^{\infty} \psi_i w_{t+l-i} = \sum_{k=-\infty}^{t+l} \psi_{t+l-k} w_k \quad (2.32)$$

$$= \psi_0 w_{t+l} + \psi_1 w_{t+l-1} + \cdots + \psi_{l-1} w_{t+1} + \psi_l w_t + \psi_{l+1} w_{t+1} + \cdots$$

If we consider the π coefficients, we can write the process as

$$x_{t+l} = - \sum_{i=1}^{\infty} \pi_i x_{t+l-i} + w_{t+l}. \quad (2.33)$$

Let $\hat{x}_t(l)$ be the l -lead linear forecast of x_{t+l} based on observed data set and it is given as

$$\hat{x}_t(l) = \psi_l^* w_t + \psi_{l+1}^* w_{t-1} + \psi_{l+2}^* w_{t-2} + \cdots \quad (2.34)$$

One can find the mean squared error of the forecast by using Equation 2.32 as

$$\begin{aligned} \mathbb{E} [(x_{t+l} - \hat{x}_t(l))^2] &= \mathbb{E} \left[\left(\psi_0 w_{t+l} + \psi_1 w_{t+l-1} + \cdots + \psi_{l-1} w_{t+1} + \sum_{i=0}^{\infty} (\psi_{i+l} - \psi_{i+l}^*) w_{t-i} \right)^2 \right] \\ &= (1 + \psi_1^2 + \psi_2^2 + \cdots + \psi_{l-1}^2) \sigma_w^2, \end{aligned} \quad (2.35)$$

where $w_t \sim w_n(0, \sigma_w^2)$. In order to minimize Equation 2.35, the equality $\psi_{i+l} = \psi_{i+l}^*$ needs to hold for $i = 0, 1, 2, \dots$. Therefore forecast of the lead l which is made at origin t , which minimizes the mean squared error is,

$$\hat{x}_t(l) = \psi_l w_t + \psi_{l+1} w_{t-1} + \psi_{l+2} w_{t-2} + \cdots \quad (2.36)$$

where ψ_i 's are ψ -coefficients.

Note. One can write the value at time $t + l$ as sum of forecast and error as following,

$$\begin{aligned} x_{t+l} &= (w_{t+l} + \psi_1 w_{t+l-1} + \cdots + \psi_{l-1} w_{t+1}) + (\psi_l w_t + \psi_{l+1} w_{t-1} + \cdots) \\ &= e_t(l) + \hat{x}_t(l), \end{aligned} \quad (2.37)$$

where $\psi_0 = 1$ and $e_t(l)$ is mean squared error of lead l .

Consequently, $\hat{x}_t(l)$ can be written in the form of conditional expectation as

$$\hat{x}_t(l) = \mathbb{E}[x_{t+l}|w_t, w_{t-1}, \dots] = \mathbb{E}_t[x_{t+l}] = \mathbb{E}[x_{t+l}|x_t, x_{t-1}, \dots], \quad (2.38)$$

where $\mathbb{E}_t[e_t(l)] = 0$. In addition to expectation, variance of error part is calculated as

$$\text{var}_t(l) = \text{var}[e_t(l)] = \sum_{i=0}^{l-1} \psi_i^2 \sigma_w^2. \quad (2.39)$$

There are some important properties of forecasting as follows [55, 13];

- Both expectation of past x values and expectation of past w values are same as their own values. On the other hand expectation of future x values are predicted ones and expectation of future w values are zero. This property is called as rules of thumb in statistics.

$$\mathbb{E}_t[x_{t-i}] = x_{t-i} \quad \text{and} \quad \mathbb{E}_t[w_{t-i}] = w_{t-i} \quad \text{for } i \geq 0, \quad (2.40)$$

and

$$\mathbb{E}_t[x_{t+i}] = \hat{x}_t(l) \quad \text{and} \quad \mathbb{E}_t[w_{t+i}] = 0 \quad \text{for } i \geq 1. \quad (2.41)$$

- We can write forecasting function by using ψ -coefficients or π -coefficients as follows respectively by using Equation 2.32 and Equation 2.33,

$$\hat{x}_t(l) = \mathbb{E}_t[x_{t+l}] = \sum_{i=0}^{\infty} \psi_i \mathbb{E}_t[w_{t+l-i}], \quad (2.42)$$

and

$$\hat{x}_t(l) = \mathbb{E}_t[w_{t+l}] - \sum_{i=1}^{\infty} \pi_i \mathbb{E}_t[x_{t+l-i}]. \quad (2.43)$$

- For $l = 1$ we have $e_t(1) = \psi_0 w_{t+1} + \sum_{i=0}^{\infty} (\psi_i + 1 - \psi_{i+1}^*) w_{t-i}$. For minimizing error term, we write $\psi_{i+1} = \psi_{i+1}^*$. Then we have

$$w_{t+1} = x_{t+1} - \hat{x}_t(1) = e_t(1). \quad (2.44)$$

- Updating forecast is illustrated as following equation

$$\hat{x}_{t+1}(l) = \hat{x}_t(l+1) + \psi_l w_{t+1}. \quad (2.45)$$

Equation 2.45 can be found by writing $\hat{x}_{t+1}(l)$ and $\hat{x}_t(l+1)$ explicitly.

Example 2.5 (ARIMA(1, 0, 0) \times (1, 0, 1)₄ Forecasting). Consider the SARIMA process $(1 - \Phi B^4)(1 - \phi B)x_t = (1 + \Theta B^4)w_t$. We can rearrange the process as

$$(1 - \phi B - \Phi B^4 + \Phi\phi B^5)x_t = (1 + \Theta B^4)w_t,$$

or

$$x_t = \phi x_{t-1} + \Phi x_{t-4} - \Phi\phi x_{t-5} + w_t + \Theta w_{t-4}.$$

Then by taking conditional expectation we will get forecasting function for $l > 0$,

$$\hat{x}_t(l) = \phi \hat{x}_t(l-1) + \Phi \hat{x}_t(l-4) - \Phi\phi \hat{x}_t(l-5) + \mathbb{E}[w_{t+l}] + \Theta \mathbb{E}[w_{t+l-4}].$$

Using the convention $\hat{x}_t(-k) = x_{t-k}$ for $k \geq 0$, we can write the following forecasts,

$$\begin{aligned}\hat{x}_t(1) &= \phi x_t + \Phi x_{t-3} - \Phi\phi x_{t-4} + \Theta w_{t-3}, \\ \hat{x}_t(2) &= \phi \hat{x}_t(1) + \Phi x_{t-2} - \Phi\phi x_{t-3} + \Theta w_{t-2}, \\ \hat{x}_t(3) &= \phi \hat{x}_t(2) + \Phi x_{t-1} - \Phi\phi x_{t-2} + \Theta w_{t-1}, \\ \hat{x}_t(4) &= \phi \hat{x}_t(3) + \Phi x_t - \Phi\phi x_{t-1} + \Theta w_t, \\ \hat{x}_t(5) &= \phi \hat{x}_t(4) + \Phi \hat{x}_t(1) - \Phi\phi x_t, \\ \hat{x}_t(l) &= \phi \hat{x}_t(l-1) + \Phi \hat{x}_t(l-4) - \Phi\phi \hat{x}_t(l-5), \quad \text{for } l \geq 6.\end{aligned}$$

As a result $(1 - \Phi D^4)(1 - \phi D)\hat{x}_t(l) = 0$ for $l = 6, 7, 8, \dots$. Let Φ and ϕ be $1/16$ and $1/2$ respectively. We have roots of $2, \mp 2$ and $\mp 2i$ for equation $(1 - \Phi D^4)(1 - \phi D)\hat{x}_t(l) = 0$. Therefore the eventual forecast function has form of,

$$\hat{x}_t(l) = c_1^{(t)} \left(\frac{1}{2}\right)^l + c_2^{(t)} \left(\frac{1}{2}\right)^l l + c_3^{(t)} \left(-\frac{1}{2}\right)^l + c_4^{(t)} \left(\frac{1}{2}\right)^l \cos\left(\frac{\pi}{2}l\right) + c_5^{(t)} \left(\frac{1}{2}\right)^l \sin\left(\frac{\pi}{2}l\right),$$

where c_i^t are discovered from $\hat{x}_t(h)$ for $h = 1, 2, 3, 4, 5$.

2.6 Heteroscedastic Models

In perspective of finance, volatility is fluctuation of price of the financial instrument. Generally, higher volatility means that one will face with higher risk in the related finance instrument. Most of usages are seen in option trading and asset returns data. Volatility can be historical or implied. We call it historical if the volatility is extracted from past market prices. On the other hand it is called implied if the volatility is obtained by the market price of a market traded derivative.

Shumway and Stoffer stated that volatility is considered as constant in ARMA models while the autoregressive conditionally heteroscedastic model (ARCH) stands for detecting the changes in variation [62]. There are several types of ARCH model by modifications. We will cover the autoregressive conditionally heteroscedastic model (GARCH) and the generalized autoregressive conditionally heteroscedastic (GARCH) model.

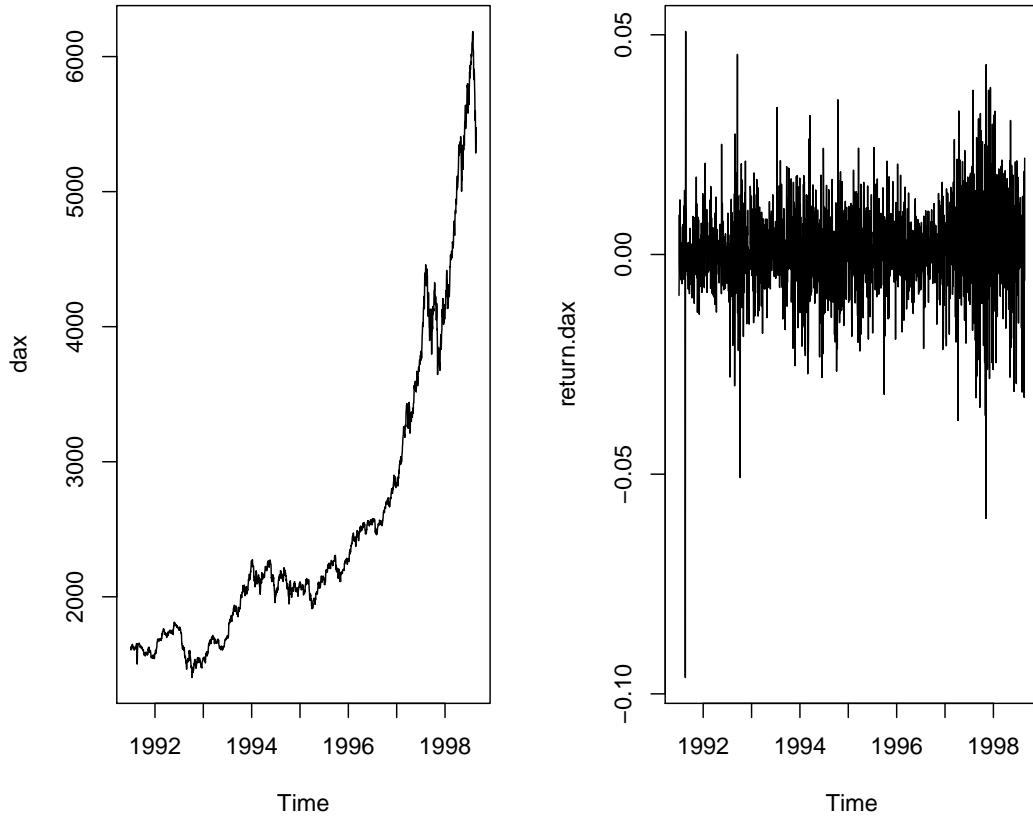


Figure 2.14: Daily Closing Prices of DAX data and its return values.

2.6.1 ARCH and GARCH Models

ARCH model is used to do variance analysis. Method was firstly introduced by Engle in [23] to model volatility. Assume that we have a time series x_t which gives the stock values at related time t . Return of x_t at time t , i.e. relative gain [62], is given as

$$r_t = \frac{x_t - x_{t-1}}{x_{t-1}} \approx \nabla [\log(x_t)]. \quad (2.46)$$

Return of a financial time series is used for variance analyses with using volatility models. Versions of ARCH model and some state-space models are example for volatility or variance analysis.

ARCH(1) model for return r_t is given by following equations

$$r_t = \sigma_t \epsilon_t \quad (2.47)$$

and

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2, \quad (2.48)$$

Table 2.1: Coefficients of Fitted AR(1)–ARCH(1) DAX Data

μ	ϕ_1	ω	α_1
6.9409e-04	3.3857e-02	9.4651e-05	1.0752e-01

where ϵ_t is standard Gaussian white noise that is distributed as $\epsilon_t \sim \text{iid}N(0, 1)$. By using Equation 2.47 and Equation 2.48, it is also possible to write the ARCH(1) model as non-Gaussian AR(1) model as

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + v_t, \quad (2.49)$$

where $v_t = \sigma_t^2(\epsilon_t^2 - 1)$. ARCH(1) model has the following properties [62],

- For the condition $0 \leq \alpha_1 < 1$, r_t is white noise and its distribution is leptokurtic.
- If $3\alpha_1^2 < 1$, r_t is causal AR(1) model and if $3\alpha_1 > 1$, r_t^2 is strictly stationary process with infinite variance.

Parameters α_0 and α_1 are estimated by conditional maximum likelihood estimation (MLE). Details can be found in Shumway and Stoffer [62] in page 283.

Example 2.6 (DAX Return Data with AR(1)–ARCH(1) Fitting). EuStockMarkets data was taken from ‘datasets’ package in R program. It includes major European stock indices: Germany DAX (Ibis), Switzerland SMI, France CAC, and UK FTSE with 1860 observations on 4 variables. Only DAX data was used in the following examples. Some results of fitting AR(1)–ARCH(1) model to return of DAX data are given in Table 2.1 and in Table 2.2. In Table 2.2 standard errors are calculated based on Hessian and significant codes ‘* **’, ‘* *’, ‘*’, ‘.’ and ‘’ stands for 0.001, 0.01, 0.05, 0.1 and 1 respectively.

Table 2.2: Error Analysis

	Estimate	Std. Error	t-value	Pr(> t)
μ	6.941e-04	2.344e-04	2.962	0.00306**
ϕ_1	3.386e-02	2.669e-02	1.268	0.20466
ω	9.465e-05	3.740e-06	25.309	< 2e-16 ***
α_1	1.075e-01	2.718e-02	3.955	7.65e-05***

Estimation of ARCH(m) is done via using ARCH(1) and conditional likelihood of returns r_{m+1}, \dots, r_n given r_1, \dots, r_m . Then ARCH(m) can be illustrated by following equations;

$$r_t = \sigma_t \epsilon_t, \quad (2.50)$$

and

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \dots + \alpha_m r_{t-m}^2. \quad (2.51)$$

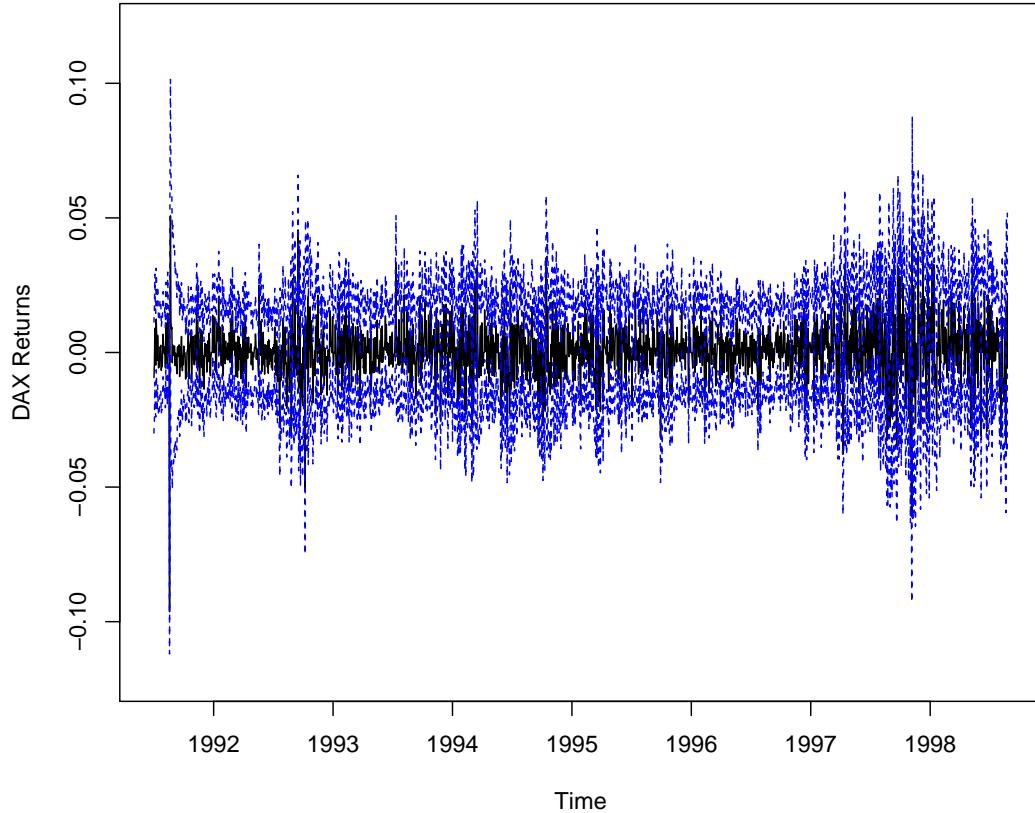


Figure 2.15: GARCH(1, 1) Model Fitted to DAX Return Data.

On the other hand generalized ARCH(m, f) or GARCH(m, f) model which is introduced by Bollerslev in [12], is given as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^m \alpha_i r_{t-i}^2 + \sum_{i=1}^f \beta_i \sigma_{t-i}^2. \quad (2.52)$$

Then the one-step-ahead forecasts of the volatility is written as

$$\hat{\sigma}_{t+1}^2 = \hat{\alpha}_0 + \sum_{i=1}^m \hat{\alpha}_i \hat{r}_{t+1-i}^2 + \sum_{i=1}^f \hat{\beta}_i \hat{\sigma}_{t+1-i}^2. \quad (2.53)$$

Example 2.7 (DAX Return Data with GARCH(1, 1) Fitting). In Table 2.3, error analysis of DAX data with fitted model GARCH(1, 1) is seen and significant codes are same as in Example 2.6. In Table 2.4 related information criterion statistics are given for fitted model. In Figure 2.15 GARCH(1, 1) model which is fitted to DAX return data is shown and prediction of volatility is with $\mp 2\hat{\sigma}_t$.

2.6.2 Other Heteroscedastic Models

There are many types of ARCH model according to related data. For example Equation 2.52 can be rewritten as

$$r_t^2 = \alpha_0 + \sum_i^{\max(m,f)} (\alpha_i + \beta_i)r_{t-i}^2 + \eta_t - \sum_{i=1}^f \beta_i \eta_{t-i}, \quad (2.54)$$

where $\sigma_{t-i}^2 = r_{t-i}^2 - \eta_{t-i}$ for $i = 0, 1, \dots, f$. Then one can have integrated GARCH model (IGARCH) in case the AR polynomial of the Equation 2.54 has a unit root. GARCH in the mean (GARCH-M) model can be used for detect the serial correlations in the financial time series with respect to related volatility. Another modified version of GARCH model is the exponential GARCH model. It provides a usage in asymmetric return dynamics of positive and negative assets. In order to manage with leverage effect, the threshold GARCH (TGARCH) model is used. Moreover, CHARMA model can be used for modeling the volatility of time series. Details for ARCH and GARCH models are given by Tsay in [68] of Chapter 3.

2.7 Other Methods: Overview

There are many other modeling types for time series analysis. One can use linear or nonlinear method to model selected time series. For example, threshold methods, neural networks, hidden Markov model and state space models can be candidate for time series which will be analyzed.

2.7.1 Threshold Models

Threshold methods are included in nonlinear modeling issues. Let's consider the simple AR(p) model given in Equation 2.10 for time series x_t as,

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + \epsilon_t \sigma, \quad (2.55)$$

where $\epsilon_t \sim WN(0, 1)$ and $\sigma > 0$ is standard deviation of disturbance term. In the sense of *Threshold AutoRegressive (TAR) model*, related to threshold variable z_t model parameters would be changeable. We can write the related formula as follows,

$$x_t = \alpha^{(j)} + X_t \phi_{(i)}^{(j)} + \sigma^{(j)} \epsilon_t \quad if \quad r_{j-i} < z_t < r_j, \quad (2.56)$$

where $X_t = (1, x_{t-1}, x_{t-2}, \dots, x_{t-p})$ for $j = 1, 2, \dots, k$, $i = 1, 2, \dots, p$ and $-\infty = r_0 < r_1 < \dots < r_k = \infty$. The domain of threshold variable z_t is separated into k different order by the $k - 1$ non-trivial thresholds that are given as $(r_1, r_2, \dots, r_{k-1})$. In each order, time series meets with different AR(p) model.

Rule of the *self-exciting TAR (SETAR) model* is that the threshold variable is lagged value of the time series. We can write the formula of SETAR as,

$$x_t = \alpha^{(j)} + \phi_1^{(j)} x_{t-1} + \phi_2^{(j)} x_{t-2} + \cdots + \phi_p^{(j)} x_{t-p} + \sigma^{(j)} \epsilon_t \quad \text{if } r_{j-i} < z_t < r_j, \quad (2.57)$$

where k and d are positive integers and $j = 1, 2, \dots, k$, $i = 1, 2, \dots, p$ and $-\infty = r_0 < r_1 < \cdots < r_k = \infty$. The threshold variable z_t is one of $\{x_t, x_{t-d}, \dots, x_{t-(m-1)d}\}$ where d is the delay parameter.

If the binary indicator function is replaced by smooth transition function $G(z_t)$ which is dependent on transition variable z_t , we get the *smooth transition autoregressive model (STAR)* [66] as

$$x_t = \alpha^{(j)} + X_t \phi_{(i)}^{(1)} (1 - G(z_t)) + X_t \phi_{(i)}^{(2)} G(z_t) + \sigma^{(j)} \epsilon_t. \quad (2.58)$$

Smooth transition function is given for logistic and exponential equations respectively as,

$$G(z_t; \gamma, c) = \frac{1}{1 + e^{-\gamma(z_t - c)}}, \quad (2.59)$$

and

$$G(z_t; \gamma, c) = 1 - e^{-\gamma(z_t - c)^2} \quad (2.60)$$

where c can be said as threshold and $\gamma > 0$ shows the speed and smoothness of transition [77]. Logistic smooth transition autoregressive model and exponential smooth transition autoregressive model are given as LSTAR and ESTAR in short terms. We have also threshold version of GARCH model as we mentioned in Subsection 2.6.2.

2.7.2 Neural Networks

Neural network system is one of learning algorithm which also includes linear regression, logistic regression, bagging, boosting, decision trees, naive Bayes, support vector machine (SVM) etc. . It is actually based on artificial learning which is inspired by brain of animals and humans. There are nodes that have similar tasks with neurons in a brain. Most system is formed of input, hidden and output nodes that are each interconnected with previous and next ones. In other words each input and each hidden nodes are connected each other with some weights. Same connection structures exist between hidden nodes and output nodes. In Figure 2.16, the information flow direction is to right. A numerical example for IBM stock is given by Tsay in [68].

Neural networks are used also for nonlinear time series forecasting [9]. It provides convenience for solving complex problems like nonlinear data sets. In Figure 2.16 structure of neural network is given.

Example 2.8 (Forecasting of DAX Data by Using Feed-Forward Neural Network). DAX daily stock prices data between middle of year 1991 and middle of year 1994, is used to find one year forecast by using neural network model. In Figure 2.17 it catches the fractal structure of stock data. The fitted model NNAR(2, 1) is analogous to an SARIMA(2, 0, 0) \times (1, 0, 0)₁ model with nonlinear functions and one hidden layer.

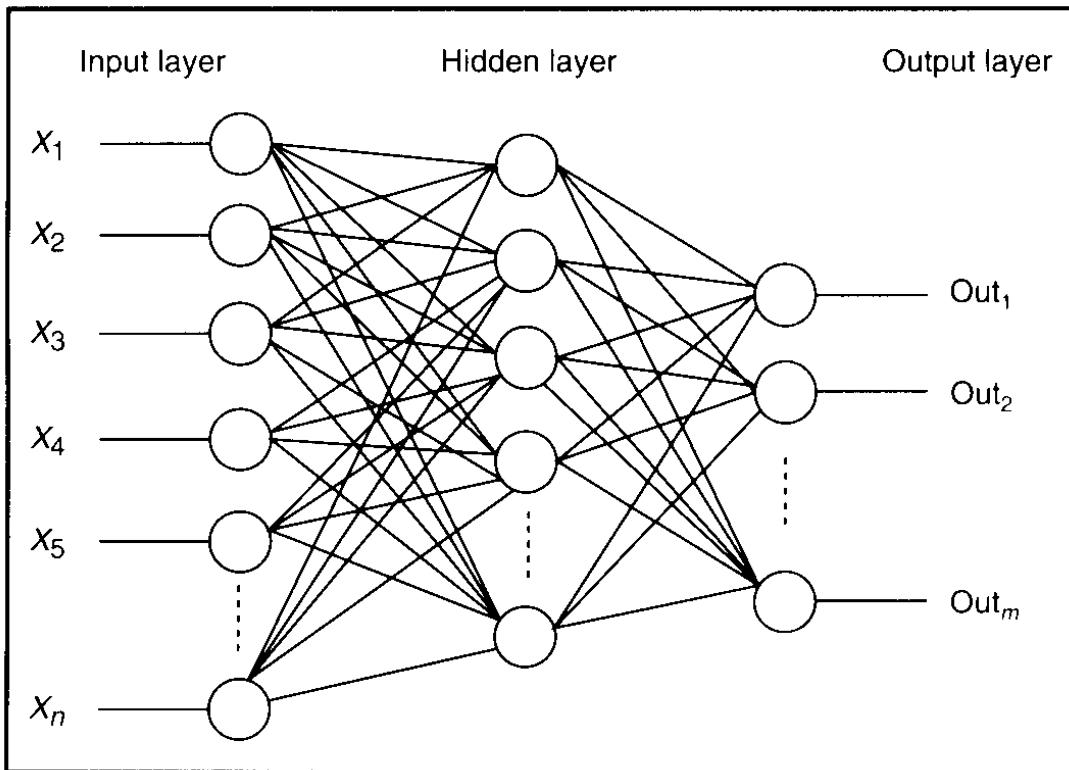
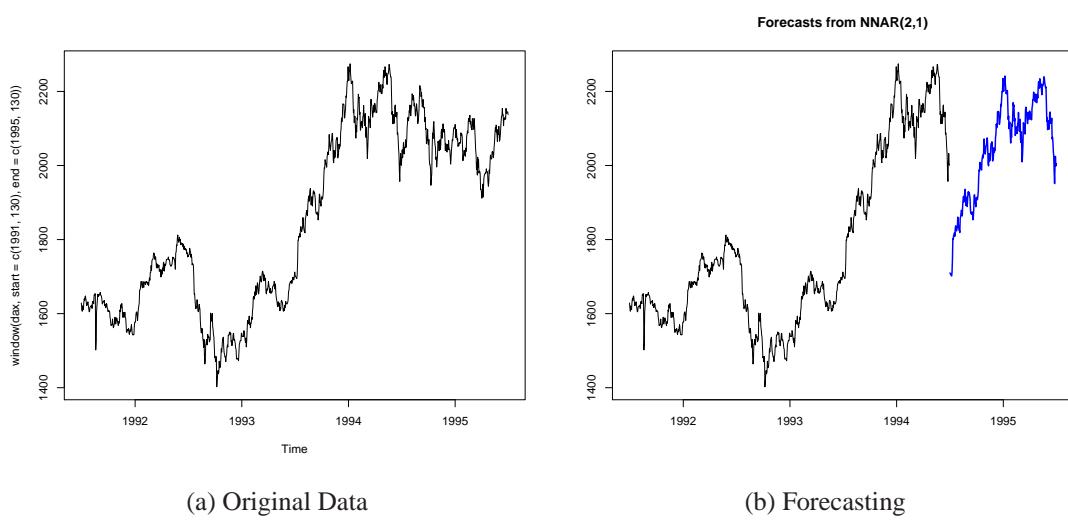


Figure 2.16: Typical Structure of a Neural Network System.
<http://mechanicalforex.com/wp-content/uploads/2011/06/NN.png>



(a) Original Data

(b) Forecasting

Figure 2.17: Forecasting by Fitting NNAR(2, 1) Model.

2.7.3 Other Methods

According to time series analysis, we have some other nonlinear models like Bilinear Model, Markov Switching Model, Nonlinear Additive AR Model, Nonlinear State-Space Model etc. . In addition there are nonparametric models for time series analysis. Moreover there are such models for volatility, multivariate time series and principal component analysis. After statistical analysis one can decide which model fits better the related data set [68, 31, 25].

Table 2.3: Error Analysis of Fitted GARCH(1, 1) DAX Data

	Estimate	Std. Error	t-value	Pr(> t)
μ	6.535e-04	2.158e-04	3.029	0.00245 **
ω	4.754e-06	1.264e-06	3.760	0.00017***
α_1	6.842e-02	1.478e-02	4.630	3.66e-06***
β_1	8.876e-01	2.356e-02	37.677	< 2e-16 ***

Table 2.4: Information Criterion Statistics of Fitted GARCH(1, 1) DAX Data

Information Criterion Statistics			
AIC	BIC	SIC	HQIC
-6.414432	-6.402538	-6.414441	-6.410049

CHAPTER 3

FUNDAMENTALS OF FOURIER AND WAVELET THEORY: FREQUENCY DOMAIN ANALYSIS

In this chapter, firstly theoretical background of Fourier theory and wavelet theory will be covered. Non-complex signals or time series can be analyzed with using time domain analysis. When data is getting more complex, frequency-domain analysis is also used to reach other characteristics of data. Masset pointed out in [46] that if the system relies on more than one frequency components, time-domain analysis is inadequate to reveal important information of real data.

Both Fourier and wavelet transform are linked to the frequency domain analysis, but Fourier analysis has several drawbacks compared to the wavelet analysis. Vuorenmaa [71] says that wavelet analysis provide non-parametric multiscale technique which is useful in both time and frequency domain where Fourier transform is inadequate for time information. In addition location parameter is held in phase and it is too hard to reach it. Also, function and the Fourier coefficients are affected each other in very sensitive way. In order to get away from some drawbacks, Windowed Fourier Transform (WFT) was used for analysis. However it doesn't still provide different levels of resolution for different time and frequencies regions. In other words WFT cannot handle physical space and Fourier space at the same time. Developed version of Fourier theory, which is called wavelet theory, has fixed many problems faced in Fourier analysis.

Variance of the financial time series is analyzed by taking the difference of the consecutive observations which are smoothed by logarithm. It is also called as volatility which is known as standard deviation. Since the variance is the square of the standard deviation, volatility is used in variance analysis for both Fourier and wavelet analysis. Volatility can be given as absolute value of the daily returns [42];

$$y_t = |\log(x_t) - \log(x_{t-1})|. \quad (3.1)$$

We use the log difference without absolute value for some empirical results to make the data stationary.

After giving basic properties and theoretical backgrounds of both Fourier and wavelet transform, applied examples will be covered. In technical explanations and in examples we are using signal, function and time series alternately.

3.1 Fourier Theory

3.1.1 Introduction

According to Joseph Fourier [10], any continuous or discontinuous function can be written as sum combination of infinite sine and cosine functions. Actually this is not the case that some discontinuous functions are not Fourier series. For instance, the convergent trigonometric series $\sum_{n=1}^{\infty} \frac{\sin(nx)}{n^\alpha}$ where $0 < \alpha \leq 1/2$, is not a Fourier series as given by Gelbaum and Olmsted in [27]. Apart from some exceptions, Fourier transforms are used to alter the domain of the wave from time to frequency.

Each *sine* and *cosine* function has a frequency and amplitude. Later on we will call the plot of frequency versus amplitude of these *sine* and *cosine* functions as power spectrum or periodogram.

On the other hand, many time series can be explained in different frequency components using Fourier transform. As well as, it can be thought as a filter so that the original time series can be decomposed into different frequency parts. Specially, financial time series are showing different behaviors according to different frequencies and different time scales [63]. After applying Fourier transform, no time information will be left to occur. Fourier transform filters the concerned time series to only frequency domain. Then from different frequency zones, different component analysis can be done.

3.1.2 Continuous Fourier Transform

Let $f \in L_1(\mathbb{R})$ and $\omega = 2\pi k$ where $k \in \mathbb{R}$ stays for frequency component, then Continuous Fourier Transform of f is defined as

$$\mathfrak{C}[f](\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt, \quad (3.2)$$

and Inverse Continuous Fourier Transform is given as

$$f(t) = \int_{-\infty}^{\infty} \mathfrak{C}[f](\omega)e^{i\omega t} d\omega. \quad (3.3)$$

If the part $e^{i2\pi kt}$ is rewritten by using *Euler's Theorem*¹ [33], then original signal can be represented onto a set of sinusoidal functions that are related to a specific frequency component.

3.1.3 Discrete Fourier Transform

In order to solve problems with computers, we need to make functions suitable for numerical evolution and implementation. Continuous numerical calculations which

¹ $e^{-i2\pi kt} = \cos(2\pi kt) - i \sin(2\pi kt)$.

are done with computers are using actually approximated estimations. Due to infinite number of elements, we need to do sampling before implementing Fourier transform. Then transform with discrete function is called discrete Fourier transform. For integrable function f which has N samples, $k/N = \omega_k$ is frequency component, Discrete Fourier Transform of f is defined as in [62, 33],

$$\begin{aligned}\mathfrak{D}[k/N] &= N^{-1/2} \sum_{t=1}^N f(t) e^{-2\pi itk/N} \\ &= N^{-1/2} \left(\sum_{t=1}^N f(t) \cos\left(\frac{2\pi itk}{N}\right) - i \sum_{t=1}^N f(t) \sin\left(\frac{2\pi itk}{N}\right) \right)\end{aligned}\quad (3.4)$$

for $k = 0, 1, \dots, n$.

If we consider DFT then amplitude and phase are given respectively as

$$|\mathfrak{D}[k/N]| / N = \frac{\sqrt{Re(\mathfrak{D}[k/N])^2 + Im(\mathfrak{D}[k/N])^2}}{N} \quad (3.5)$$

and

$$\arg(\mathfrak{D}[k/N]) = \text{atan2}(Im(\mathfrak{D}[k/N]), Re(\mathfrak{D}[k/N])) = -i * \ln\left(\frac{\mathfrak{D}[k/N]}{|\mathfrak{D}[k/N]|}\right), \quad (3.6)$$

where atan2 is arctan representation for two variables.²

3.1.4 Spectral Analysis and Periodogram

Distribution of the variance of the signal $x(t)$ over the frequency ingredients can be represented by the power spectrum. In addition spectrum gives information about periodicity of the data [13]. In other words, cyclical behaviors of the time series can be viewed in the frequency domain [32]. However, as we mentioned in Subsection 3.1.1, all time knowledge will be lost after Fourier transform. Moreover, Fourier analysis is effective for only periodic and stationary time series [55].

The Short-time Fourier transform (STFT) and the Gabor transform are designed to divide the time series into little pieces to add new parameter as time shifting near the frequency [28]. However it brings a problem about the length of the slice that is not fit with both time and frequency at the same time. If a window function has wide interval, then the consequence gives better frequency resolution beside the bad time resolution and vice versa. In addition STFT doesn't give information about different resolutions.

The spectrum of the windowed time series can be given as [55];

$$\text{STFT}_x(n, \omega) = \sum_{t=1}^N x(t) h^*(t-n) e^{-i\omega t}, \quad (3.7)$$

² See http://en.wikipedia.org/wiki/Discrete_Fourier_transform [Retrieved 23.11.2014]

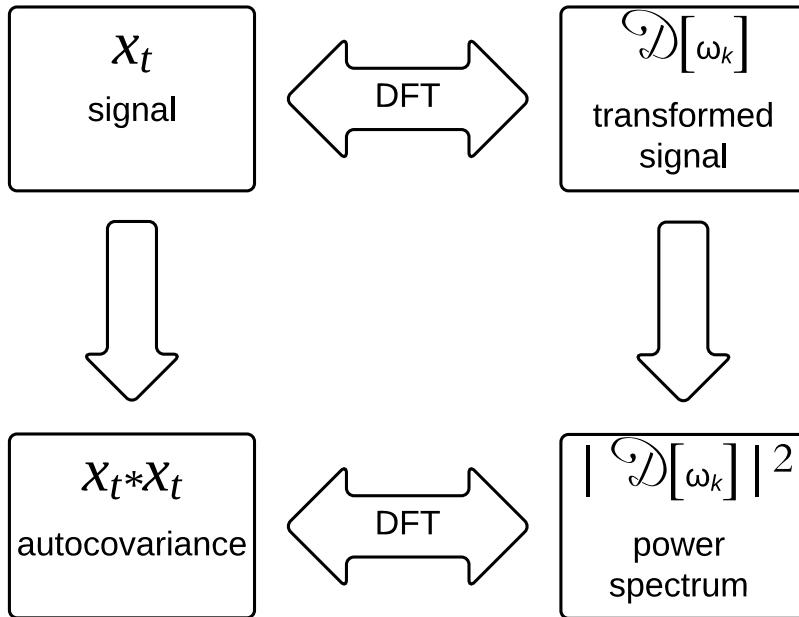


Figure 3.1: Relation between power spectrum and autocovariance.

where $h(t)$ is a window function centered at a time point n , ω is the angular frequency and h^* is the complex conjugate of a window function. In order to visualize the energy spectral density of $x(t)$, *spectrogram* of the time series is written as

$$\text{Spectrogram}_x(n, \omega) = \left| \sum_{t=1}^N x(t)h^*(t-n)e^{-i\omega t} \right|^2. \quad (3.8)$$

Power spectrum is obtained by taking the Fourier transform of an autocovariance function. The connection between autocovariance and the power spectrum is that inverse Fourier transform of the power spectrum is autocovariance function. Loop for the signal, autocovariance and power spectrum can be shown as in the below graph for discrete Fourier transform.

In general form, *periodogram* of the time series can be written as

$$\begin{aligned} I_x(\omega_k) &= |\mathfrak{D}[k/N]|^2 = \frac{1}{N} \left| \sum_{t=1}^N x_t e^{-2\pi i \omega_k t} \right|^2 \\ &= \frac{1}{N} \left(\sum_{t=1}^N x_t \cos \left(\frac{2\pi i t k}{N} \right) \right)^2 + \frac{1}{N} \left(\sum_{t=1}^N x_t \sin \left(\frac{2\pi i t k}{N} \right) \right)^2. \end{aligned} \quad (3.9)$$

If we consider the sample autocovariances, we can write the periodogram for $\omega_k = 0$ as $I(0) = N\bar{x}^2$ and ω_k for $k \neq 0$ by following equation,

$$I_x(\omega_k) = \sum_{h=-(N-1)}^{N-1} \hat{\gamma}(h) e^{-2\pi i \omega_k h} = \hat{\gamma}(0) + 2 \sum_{h=1}^{N-1} \hat{\gamma}(h) \cos(2\pi \omega_k h), \quad (3.10)$$

where $\hat{\gamma}(h)$ is the sample autocovariance and for large N , $\hat{\gamma}(h)$ is approximately unbiased for $\gamma(h)$. In other words we have the equality $\mathbb{E}[\hat{\gamma}(h)] \approx \gamma(h)$. Then we can edit Equation 3.10 as

$$\begin{aligned} \mathbb{E}[I_x(\omega_k)] &\approx \sum_{h=-(N-1)}^{N-1} \gamma(h) e^{-2\pi i \omega_k h} \approx \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega_k h} \\ &= \gamma(0) + 2 \sum_{h=1}^{\infty} \gamma(h) \cos(2\pi \omega_k h) = f(\omega_k), \end{aligned} \quad (3.11)$$

where $f(\omega_k)$ is *spectral density*.

On the other hand the *scaled periodogram* is given as

$$P_x(\omega_k) = \left(\frac{4}{n} \right) I_x(\omega_k). \quad (3.12)$$

The part $\left| \sum_{t=1}^N x_t e^{-2\pi i \omega_k t} \right|^2$ in Equation 3.9 which is linked to discrete Fourier transform of the time series is called as *power spectrum*. All the definitions according to spectral analysis above are also valid for continues case. Initially, the idea of periodogram was created to identify and measure the amplitude of sine and cosine factors whose frequencies are known [13]. Many researchers used periodogram to analyze the data in frequency domain. For instance, Box and Jenkins in [13] used periodogram for detecting the randomness of the time series and finding periodic parts of unknown frequencies.

Spectral Density

If a number of processes in a function repeat itself in a certain gap, we consider the periodicity issue in this function. Sine and cosine functions are periodic functions. They are used to define periodic processes. Then periodic function can be written as follows

$$x_t = B_1 \sin(2\pi \omega t) + B_2 \cos(2\pi \omega t), \quad (3.13)$$

where $B_1 = A \sin \phi$ and $B_2 = -A \cos \phi$ are random variables. Then the amplitude is $A = \sqrt{B_1^2 + B_2^2}$ and the starting point which is called as phase is $\phi = \tan^{-1}(-B_1/B_2)$. In addition, B_1 and B_2 are independent if and only if A and ϕ are independent random variables with some conditions [62]. Then the autocovariance function of x_t in Equation 3.13 for independent B_1 and B_2 is

$$\begin{aligned} & \text{cov}(x_{t+h}, x_t) \\ &= \sigma^2 \{ \sin(2\pi\omega(t+h)) \sin(2\pi\omega t) + \cos(2\pi\omega(t+h)) \cos(2\pi\omega t) \}. \end{aligned} \quad (3.14)$$

From the trigonometric equations which are sum and difference formulas [16], we can write Equation 3.14 as

$$\text{cov}(x_{t+h}, x_t) = \sigma^2 \cos(2\pi\omega(t+h-t)) = \sigma^2 \cos(2\pi\omega h), \quad (3.15)$$

where $h = |s - t|$ is the lag for every s and t .

In general form, as Schuster pointed out in [59] that periodic series can be written as the sum of the several number of sine and cosine functions such as [13]

$$x_t = \sum_{i=1}^m [B_{i1} \sin(2\pi\omega_i t) + B_{i2} \cos(2\pi\omega_i t)], \quad (3.16)$$

where B_{i1} and B_{i2} are independent zero-mean random variables with variances σ_i^2 for $i = 1, 2, \dots, m$. In addition ω_i 's are distinct frequencies [28]. Then the autocovariance function of x_t that is defined in Equation 3.16 can be given as for independent random variables B_i 's

$$\gamma(t+h, t) = \sum_{i=1}^m \sigma_i^2 \cos(2\pi\omega_i t). \quad (3.17)$$

The autocovariance in Equation 3.17 depends on only the lag, not depend on s and t . Therefore x_t is a mean-zero stationary function with variance [62]

$$\gamma(0) = \gamma_x(t, t) = \mathbb{E}(x_t^2) = \text{cov}(x_t, x_t) = \sum_{i=1}^m \sigma_i^2. \quad (3.18)$$

Shumway and Stoffer mentioned in [62] that Equation 3.16 is valid for almost all stationary time series [62]. Then one can link the spectral analysis of stationary time series with variance analysis. If one fix the frequency ω in Equation 3.13 as ω' , then process complete ω' cycles for any $t = 0, \pm 1, \pm 2, \dots$. One can write the autocovariance function in Equation 3.15 by using the Euler's theorem [33] and the Riemann Stieltjes integration [7] as

$$\begin{aligned} \text{cov}(x_{t+h}, x_t) &= \frac{\sigma^2}{2} (e^{-2\pi i \omega' h} + e^{2\pi i \omega' h}) \\ &= \int_{-1/2}^{1/2} e^{2\pi i \omega h} dF(\omega) = 2 \int_0^{1/2} \cos(2\pi\omega h) dF(\omega), \end{aligned} \quad (3.19)$$

where $F(\omega)$ is spectral distribution function and given by

$$F(\omega) = \begin{cases} 0 & \text{if } \omega < -\omega', \\ \sigma^2/2 & \text{if } -\omega' \leq \omega < \omega', \\ \sigma^2 & \text{if } \omega \geq \omega'. \end{cases} \quad (3.20)$$

If the absolute sum of the autocovariance function with respect to Equation 3.13 is finite where h is between $-\infty$ and ∞ , then we can write Riemann Stieltjes for $h = 0, \pm 1, \pm 2, \dots$ as

$$\text{cov}(x_{t+h}, x_t) = \int_{-1/2}^{1/2} e^{2\pi i \omega h} f(\omega) d\omega = 2 \int_0^{1/2} \cos(2\pi \omega h) f(\omega) d\omega, \quad (3.21)$$

with the inverse transform of the *spectral density* as

$$f(\omega) = \sum_{h=-\infty}^{\infty} \text{cov}(x_{t+h}, x_t) e^{-2\pi i \omega h} = \gamma(0) + 2 \sum_{h=1}^{\infty} \gamma(h) \cos(2\pi \omega h), \quad (3.22)$$

where $-1/2 < \omega < 1/2$. In other words, $F'(\omega) = f(\omega)$ in case $F(\omega)$ is absolutely continuous. The spectral density is even function of period one. As a result one can verify that $f(\omega)$ is illustrated only for $\omega \geq 0$. In addition, $f(\omega) = f(-\omega)$ and $f(\omega) = f(1 - \omega)$ as Shumway and Stoffer explained in [62]. Another property is being non-negative, i.e. $f(\omega) \geq 0$. The variance of x_t can be written by choosing h as 0,

$$\text{var}(x_t) = \int_{-1/2}^{1/2} f(\omega) d\omega < \infty. \quad (3.23)$$

The autocovariance function and the spectral density function catch the same properties of the signal in terms of lag and cycle respectively [62].

The periodogram chooses strong frequencies in the finite sample. On the other hand, for a stationary x_t , the spectral density can be used to define population model. The spectral density is approximation of expected value of the periodogram as seen in Equation 3.11. As a result plotting expectation of the periodogram gives similar result with the spectral density. On the other hand, estimator of the spectral density would be the periodogram. Drawback of choosing the periodogram as an estimator of the spectral density is providing poor estimate due to having only two degree of freedom.

Example 3.1 (Examples of Spectral Density: White Noise). Take into consideration a series of uncorrelated random variables w_t 's with variance σ_w^2 [68]. The mean is $\mu_{w_t} = \mathbb{E}[w_t] = 0$ and the autocovariance is $\gamma_w(h) = \text{cov}(w_{t+h}, w_t)$ is equal to σ_w^2 for $h = 0$ and equal to 0 for $h \neq 0$. White noises w_{t+h} and w_t are uncorrelated. We find out that w_t is a stationary (weak) process. Moreover it has a constant power spectral density such that

$$f_w(\omega) = \sigma_w^2, \quad (3.24)$$

where ω is between $-1/2$ and $1/2$. It means that the power in each frequency is equal.

Example 3.2 (Examples of Spectral Density: Moving Average). Assume we have such a process as $v_t = \frac{1}{5}(w_{t-1} + w_t + w_{t+1})$. It has zero mean and the autocovariance of the process is computed by $\gamma(h) = \frac{1}{25}\text{cov}(w_{t-1} + w_t + w_{t+1}, w_{t+h-1} + w_{t+h}, w_{t+h+1})$. Then we get autocovariance function as

$$\gamma(h) = \begin{cases} \frac{3}{25}\sigma_w^2 & \text{if } h = 0, \\ \frac{2}{25}\sigma_w^2 & \text{if } h = \mp 1, \\ \frac{1}{25}\sigma_w^2 & \text{if } h = \mp 2, \\ 0 & \text{if } |h| > 2. \end{cases}$$

Process is stationary since it does not depend on s and t . It only depends on lag h . The spectral density of the process is estimated from Equation 3.22 and given as

$$f_v(\omega) = \frac{\sigma_w^2}{25}[3 + 4\cos(2\pi\omega) + 2\cos(4\pi\omega)].$$

Example 3.3 (Examples of Spectral Density: ARMA Process). Let x_t be ARMA(p, q) process such that $\phi(B)x_t = \theta(B)w_t$ where B is backshift operator and ϕ, θ are operators for autoregression and moving average respectively. Then spectral density of ARMA(p, q) is given by in [62]

$$f_x(\omega) = \sigma_w^2 \frac{|\theta(e^{-2\pi i\omega})|^2}{|\phi(e^{-2\pi i\omega})|^2}, \quad (3.25)$$

where $\phi(z) = 1 - \sum_{l=1}^p \phi_l z^l$ and $\theta(z) = 1 + \sum_{l=1}^q \theta_l z^l$.

Assume that we have a ARMA(2, 1) process such that $x_t = -0.4x_{t-1} + 0.6x_{t-2} + w_t + 0.5w_{t-1}$. It is seen that $\phi_1 = -0.4$, $\phi_2 = 0.6$ and $\theta_1 = 0.5$. Then we have,

$$\begin{aligned} |\theta(e^{-2\pi i\omega})|^2 &= |1 + 0.5e^{-2\pi i\omega}|^2 = (1 + 0.5e^{-2\pi i\omega})(1 + 0.5e^{2\pi i\omega}) \\ &= 1.25 + 0.5(e^{-2\pi i\omega} + e^{2\pi i\omega}) = 1.25 + \cos(2\pi\omega). \\ |\phi(e^{-2\pi i\omega})|^2 &= (1 + 0.4e^{-2\pi i\omega} - 0.6e^{-4\pi i\omega})(1 + 0.4e^{2\pi i\omega} - 0.6e^{4\pi i\omega}) \\ &= 1.52 - 0.2(e^{-2\pi i\omega} + e^{2\pi i\omega}) - 0.6(e^{-4\pi i\omega} + e^{4\pi i\omega}) \\ &= 1.52 - 0.4\cos(2\pi\omega) - 1.2\cos(4\pi\omega). \end{aligned}$$

Finally by using Equation 3.25 spectral density of our ARMA(p, q) process is written as,

$$f_x(\omega) = \sigma_w^2 \frac{1.25 + \cos(2\pi\omega)}{1.52 - 0.4\cos(2\pi\omega) - 1.2\cos(4\pi\omega)}.$$

However AR part is not satisfy the causality, because $1/|\phi_2|$ is not outside of the unit circle. So let's change ϕ_2 as 0.3 and write the spectral density after same calculations as

$$f_x(\omega) = \sigma_w^2 \frac{1.25 + \cos(2\pi\omega)}{1.25 - 0.56\cos(2\pi\omega) - 0.6\cos(4\pi\omega)}.$$

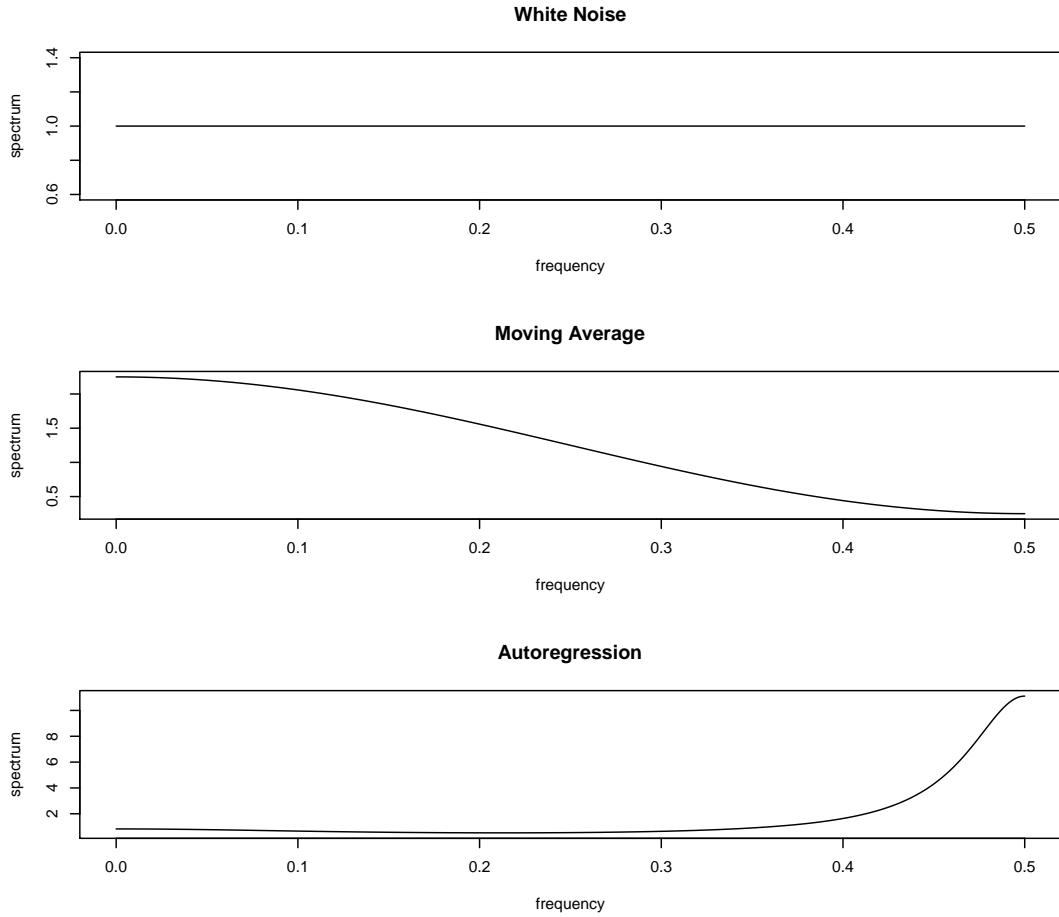


Figure 3.2: The Spectral Densities of White Noise, Moving Average and Autoregressive Process separately in Example 3.1, Example 3.2 and Example 3.3.

Theoretical spectra of white noise, moving average, autoregressive process and ARMA process are given in Figure 3.2 and in Figure 3.3. The ARMA(2, 1) process has the highest spectrum as 2.7777778 at frequency 0.5 and it has a spectrum of 1.8595041 at frequency 0. As we see in Figure 3.2, AR process has also the highest spectrum at frequency 0.5 as 11.111111. As a result it has highest spectrum at a period at two cycles per point.

If we have the same example with only AR(2) process instead of ARMA(2, 1), then we would get the spectral density as

$$f_x(\omega) = \sigma_w^2 \frac{1}{1.25 - 0.56 \cos(2\pi\omega) - 0.6 \cos(4\pi\omega)}.$$

Example 3.4 (Simulated Series Example). Let's define three sine functions x_1, x_2 and x_3 with frequencies $\omega_1 = \frac{5}{100}$, $\omega_2 = \frac{25}{100}$ and $\omega_3 = \frac{85}{100}$ respectively and with each has 100 observations, i.e. $t = 1, 2, \dots, 100$. Series are given as

$$x_1(t) = 3 \sin\left(\frac{2\pi t 5}{100}\right) + 4 \cos\left(\frac{2\pi t 5}{100}\right),$$

ARMA(2,1)

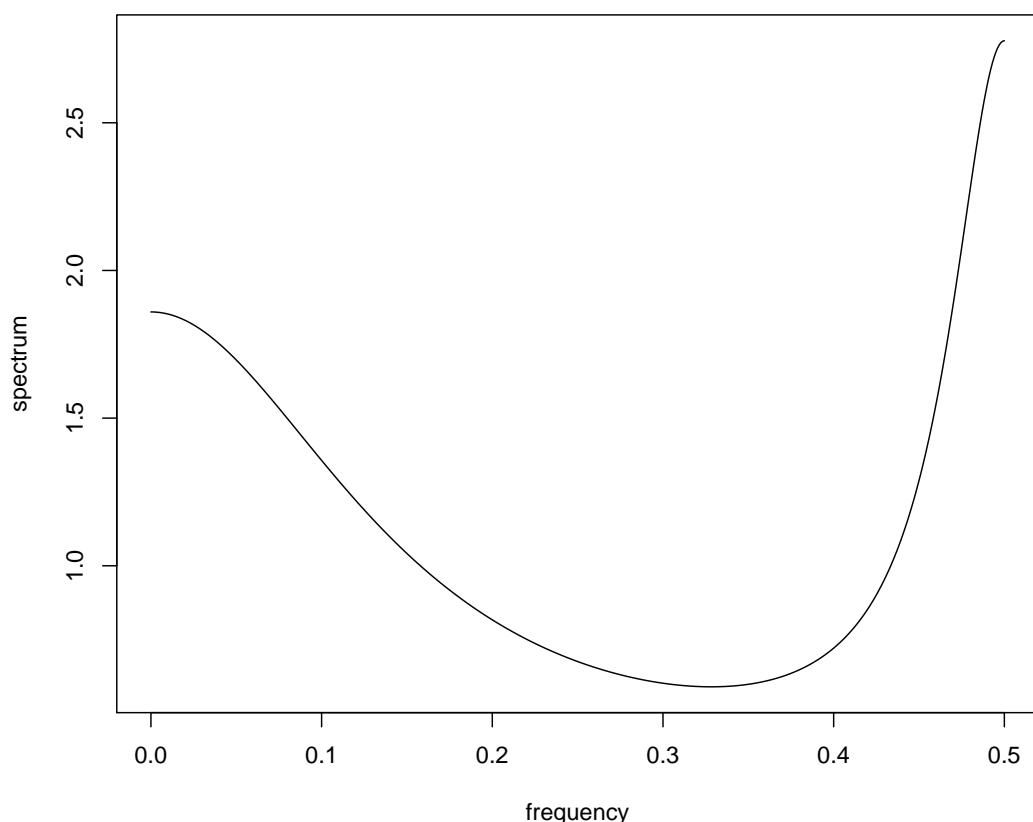


Figure 3.3: The spectral density of ARMA(2, 1) in Example 3.3.

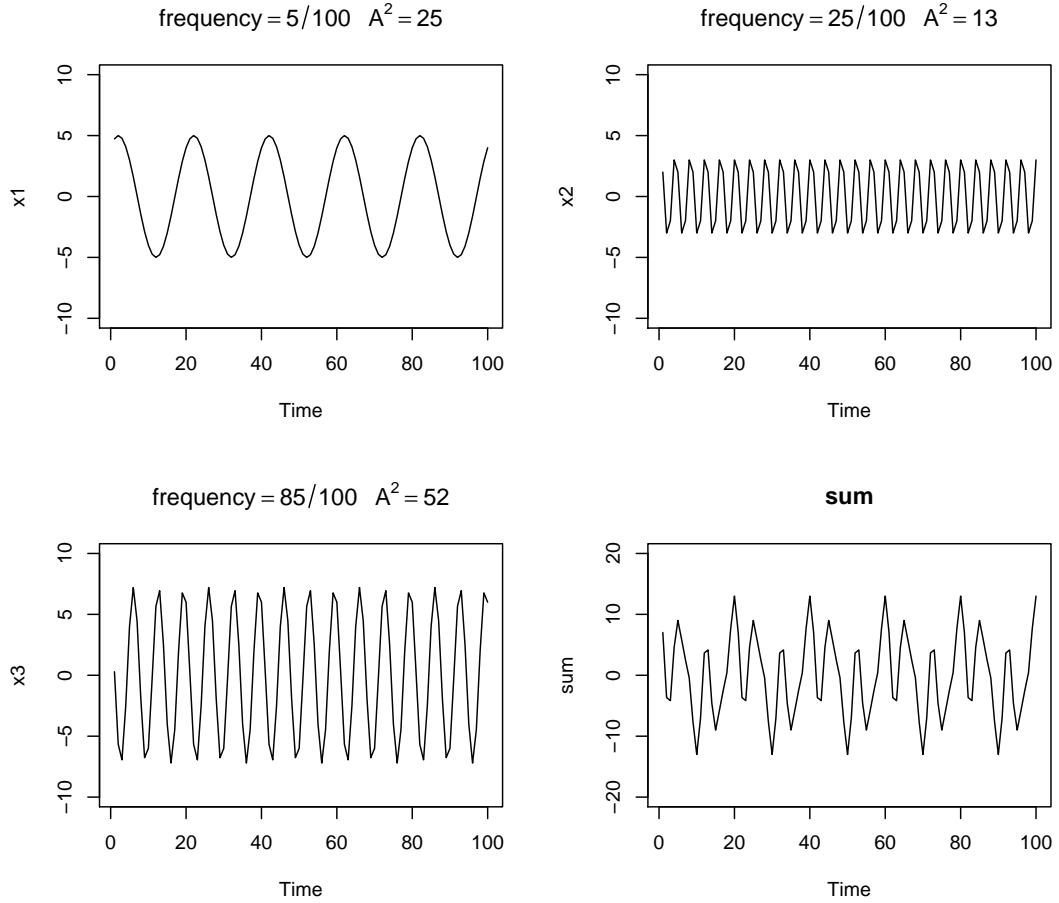


Figure 3.4: Three periodic functions with different frequencies and their sum.

$$x_2(t) = 2 \sin\left(\frac{2\pi t 25}{100}\right) + 3 \cos\left(\frac{2\pi t 25}{100}\right),$$

$$x_3(t) = 4 \sin\left(\frac{2\pi t 85}{100}\right) + 6 \cos\left(\frac{2\pi t 85}{100}\right).$$

Sum of above three functions includes three different frequency components with respect to sine and cosine components. For equal number of observations, function with higher frequency includes denser signal due to reserving more number of cycles per unit time. Due to the fact that all three functions are cyclical, the sum of these functions is also cyclical. In addition amplitude of the sum is total amplitude of three functions. In Figure 3.4 three simulated functions and the sum are illustrated.

Shumway and Stofer in [62] gives the formula of the scaled periodogram as

$$P\left(\frac{j}{n}\right) = \left(\frac{2}{n} \sum_{t=1}^n x_t \sin\left(\frac{2\pi t j}{n}\right) \right)^2 + \left(\frac{2}{n} \sum_{t=1}^n x_t \cos\left(\frac{2\pi t j}{n}\right) \right)^2, \quad (3.26)$$

where j/n represents the j th frequency. The peaks values in Figure 3.5 and in Figure 3.6 are the values which are the square amplitudes of the simulated functions by

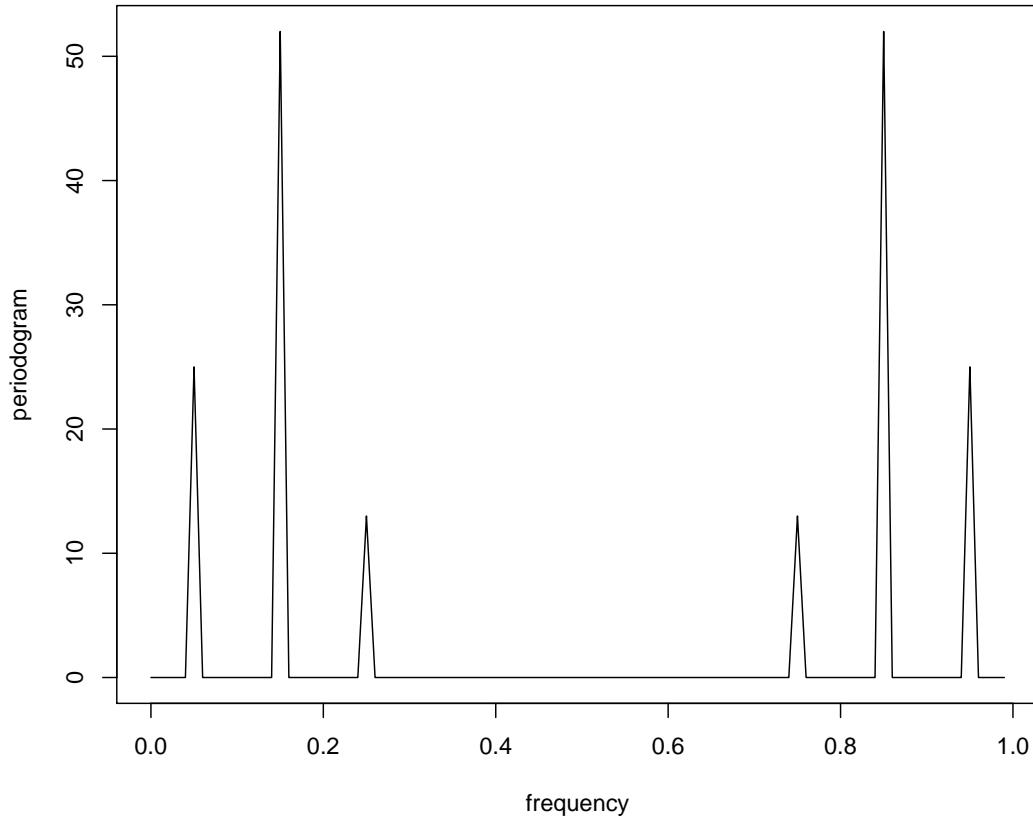


Figure 3.5: Periodogram of sum function by using FFT.

using the Fast Fourier Transform. We get the mirrors of peaks with respect to frequency 0.5 for each function since $P(j/n) = P(1 - j/n)$. Figure 3.5 shows the periodogram of sum function. If we change the third frequency as $\omega_3 = 75/100$, we will see 4 peaks instead of 6 peaks since the periodogram of the second and the third function will be the same. If we choose the 1st and 2nd functions same, but 3rd one by changing the frequency $\omega_3 = 50/100$ then we will get 5 peaks.

Example 3.5 (DAX Data). In the below analysis, DAX data from R program database is used and it has 1860 observations with frequency 260.

It is seen from ACF plot of the data DAX that the time series is not stationary since all autocorrelation values exceed the confidence interval.

Histogram is skewed right (positive). In Figure 3.8, histogram on the right has more number of cells and so details are more visible. Data is distributed non-normally. In general normal distributed data is symmetric and it has a center, i.e. mean, median, and mode are all same. On the contrary, mean, median, and mode are different each other for the non-symmetric distributions. In order to have an idea of centerness issue for data which has a skewed histogram, it is better to look all of three measures. It is seen

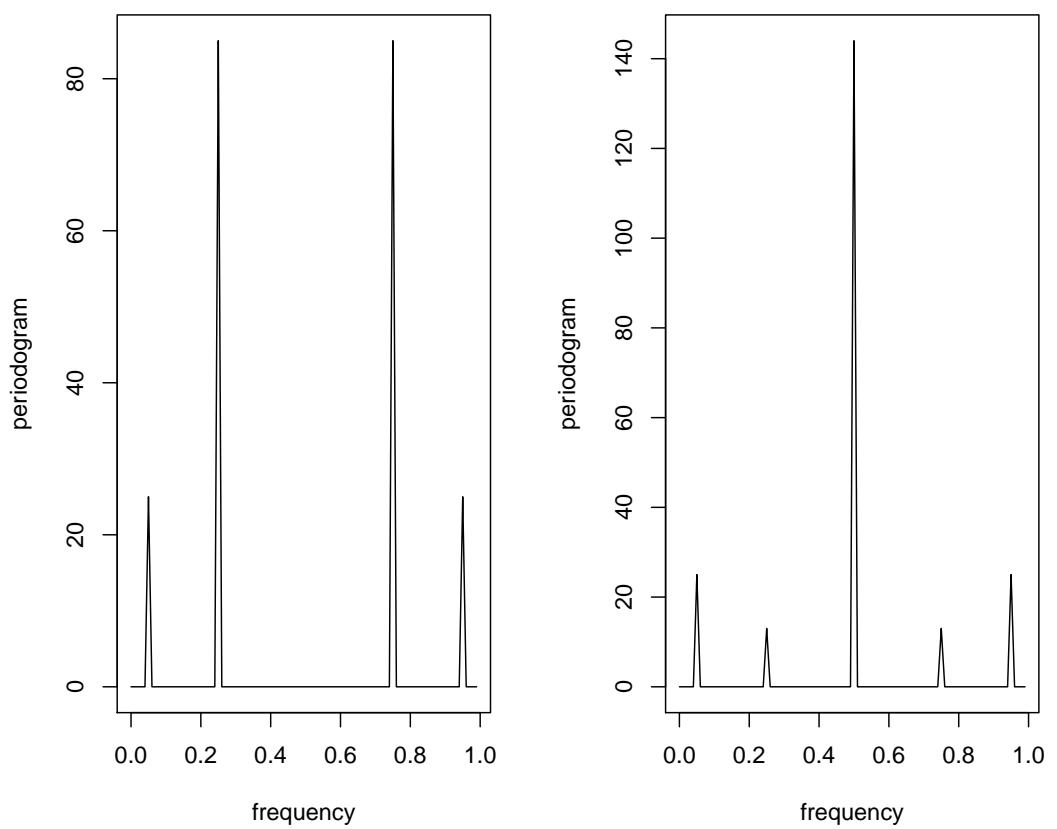


Figure 3.6: Periodogram where frequency of 3rd function is 0.75 at left, 0.5 at right.

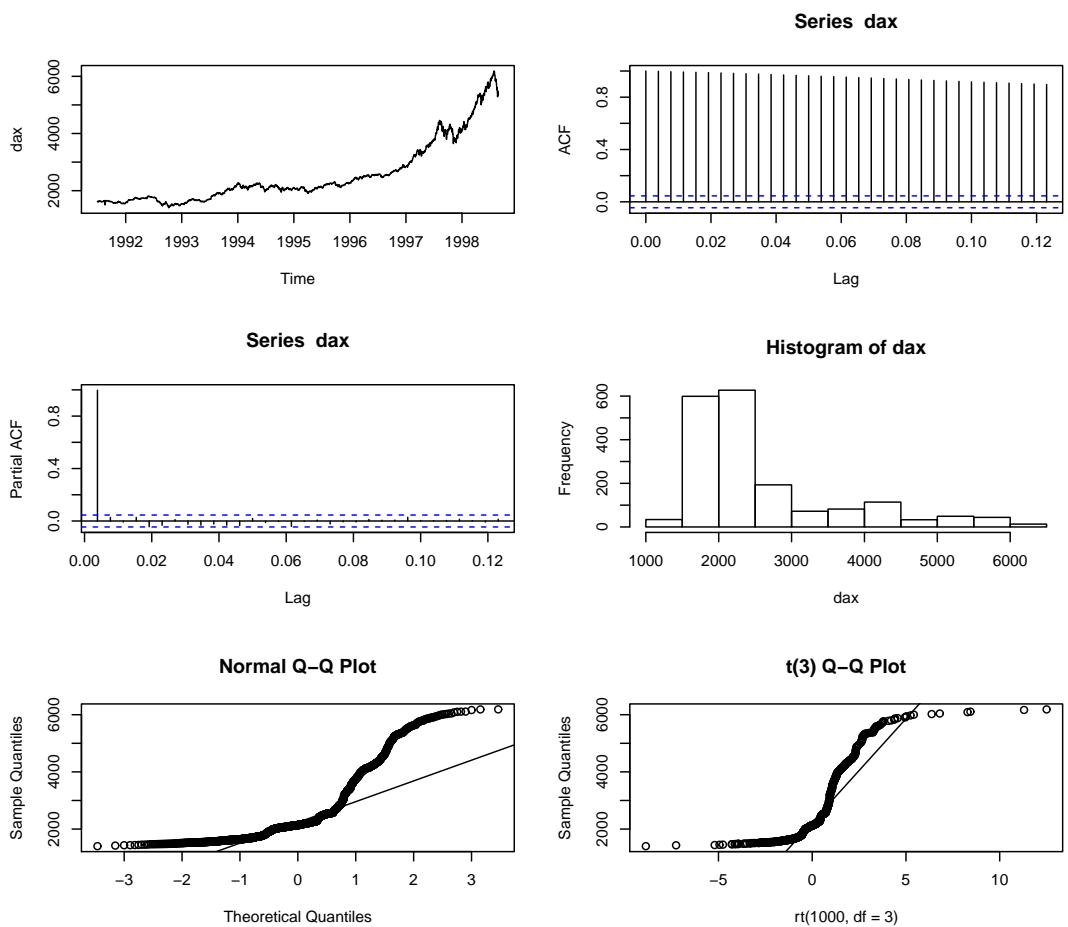


Figure 3.7: Data of DAX and Related ACF, PACF, Histogram and Quantile Graphs.

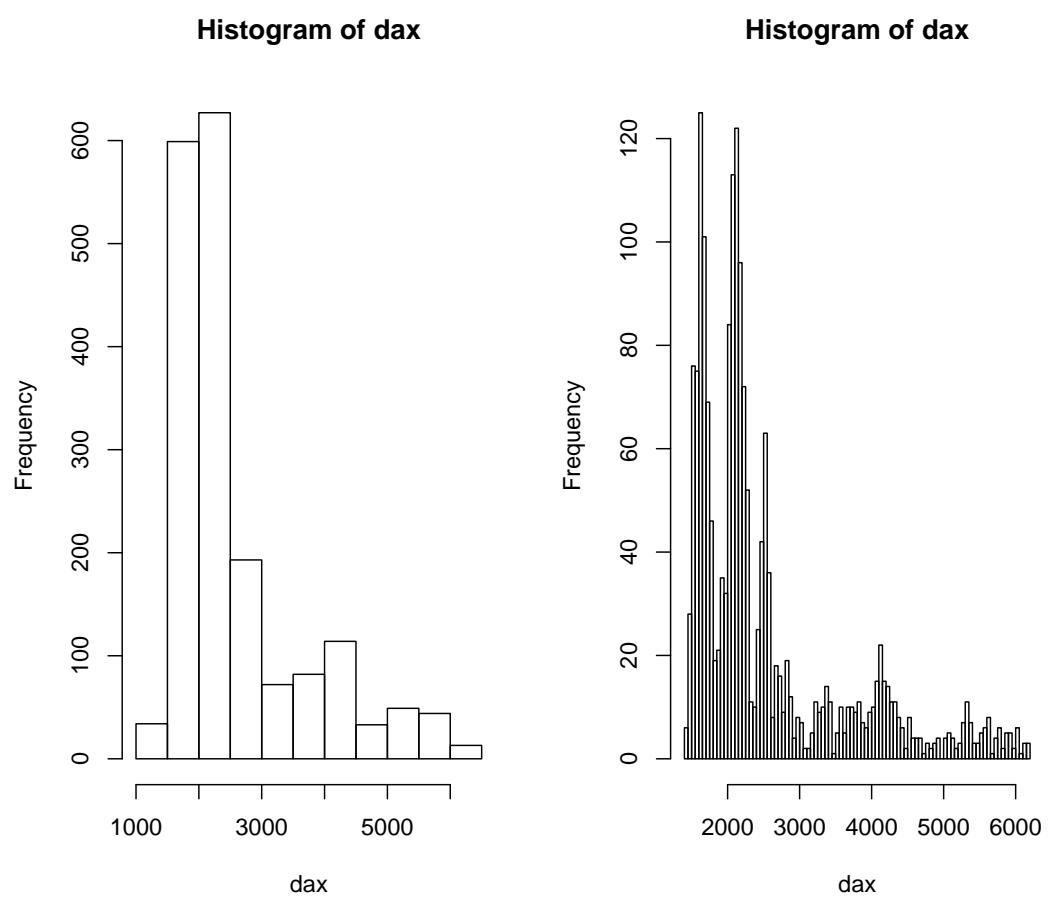


Figure 3.8: More detailed histogram of DAX.

Table 3.1: Basic Statistical Results of DAX Data

Min.	1st Qu.	Median	Mean
1402	1744	2141	2531
3rd Qu.	Max	Mode	Kurtosis
2722	6186	1542.77	4.565346

in Table 3.1 that median, mean and mode values are different each other for DAX data which means that distribution is non-normal. Having lower bounds cause also skewed distributions. Last two figures in Figure 3.7 show the Quantile-Quantile plot that is fitting sample data to the normal distribution. We see the non-normal distribution from the S type of the line which is not fitting the normality line.

In Figure 3.9 the right top graph shows the estimated spectral density of DAX data by smoothed periodogram using FFT. Algorithm uses taper argument to link relation between the series and taper by applying a split cosine bell taper to the beginning and the end of the time series. It should be between 0 and 0.5. Here taper argument is 0 and there is no character to qualify log scale. The figure on the left bottom has a log scale characterization and it is seem that it is hard to read the significant spectrums which are related to seasonality. The AR fitted spectrum figure consists of four methods. OLS method and Burg method almost follow each other. Other than YuleWalker and MLE methods, OLS and Burg methods are giving better spectrum results, since we can realize peaks better.

In Figure 3.10, first graph is the periodogram of time series DAX. Other graphs are the smoothed ones with using modified Daniell smoother and widths of smoothing values are given in graphs. It is seen that, after some values of widths it is easier to catch the important peaks. In addition to peaks between frequencies 20-40, 40-60 and frequency about 40 in the AR fitted spectrum, we catch the peak between frequencies 110-120.

In Figure 3.11, again Daniell smoother is used with the kernel argument. It gives almost same periodograms with spanning argument, but kernel argument makes the periodogram smoother than span argument does.

3.2 Wavelet Theory

3.2.1 Introduction

Wavelet theory is a nearly recent and popular issue in almost all engineering and science areas. It is applied in many areas like finance, economy, image processing, signal analysis etc. One of its important properties is providing a multiresolution analysis to get decomposed parts of signal for analysis, processing, de-noising or compression [15]. Specific examples of usage areas can be given as seismic data for geologists, voice and fingerprint data for FBI, image processing like JPEG 2000 standard [2] for

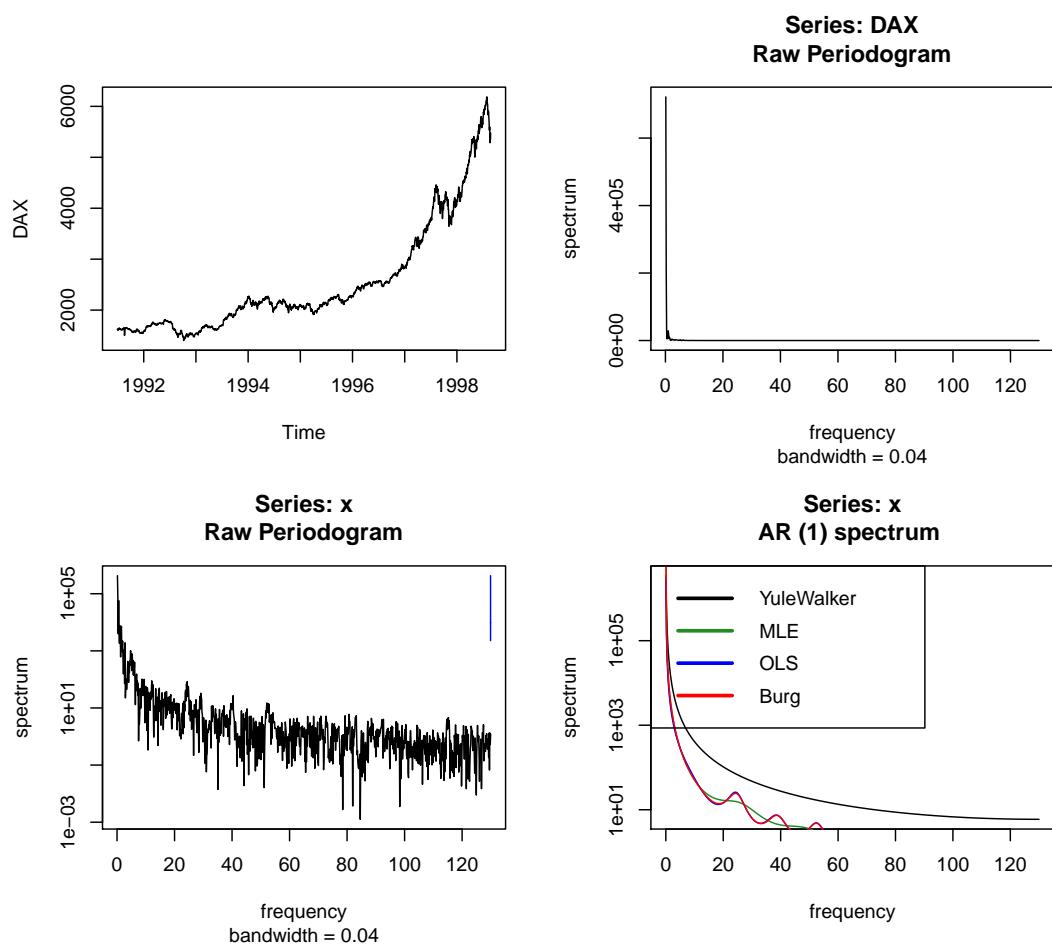


Figure 3.9: DAX data, spectral density by smoothed periodogram, spectral density and spectral density using AR method.

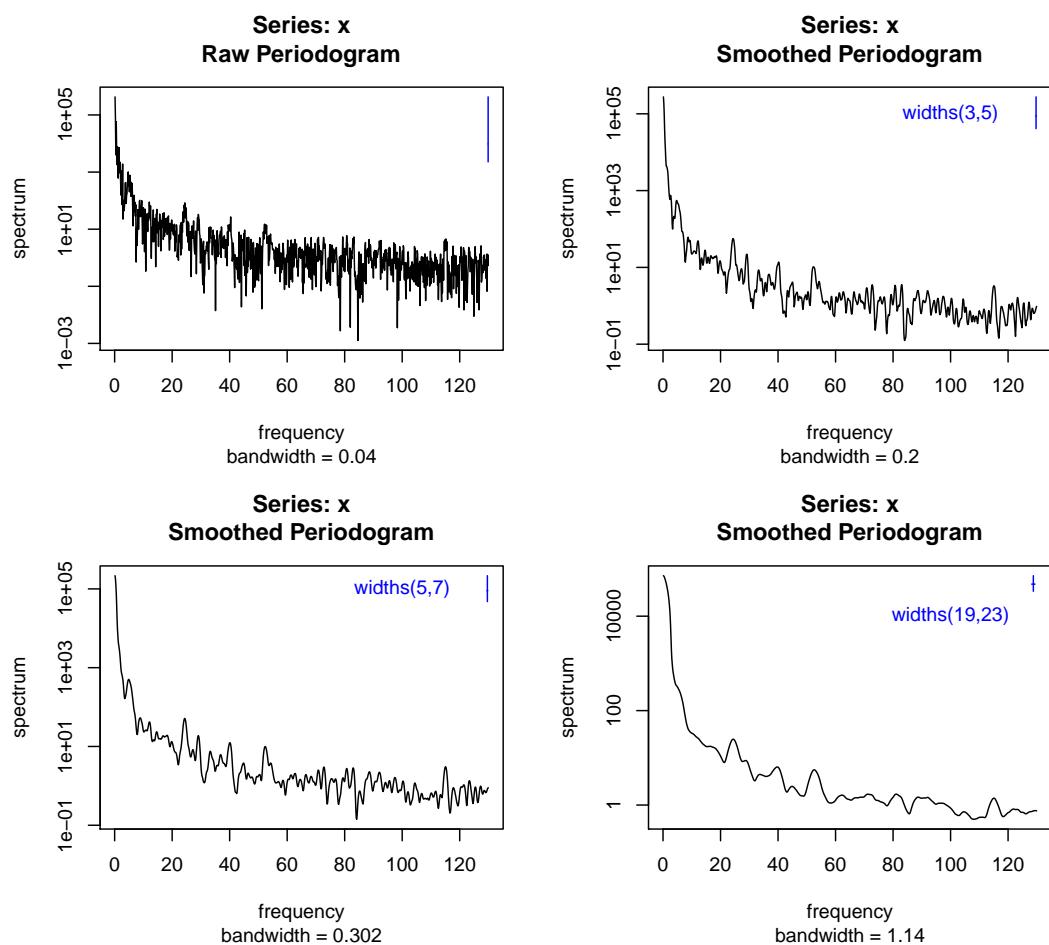


Figure 3.10: Periodogram of DAX and its smoothed versions using Daniell smoothers by spanning.

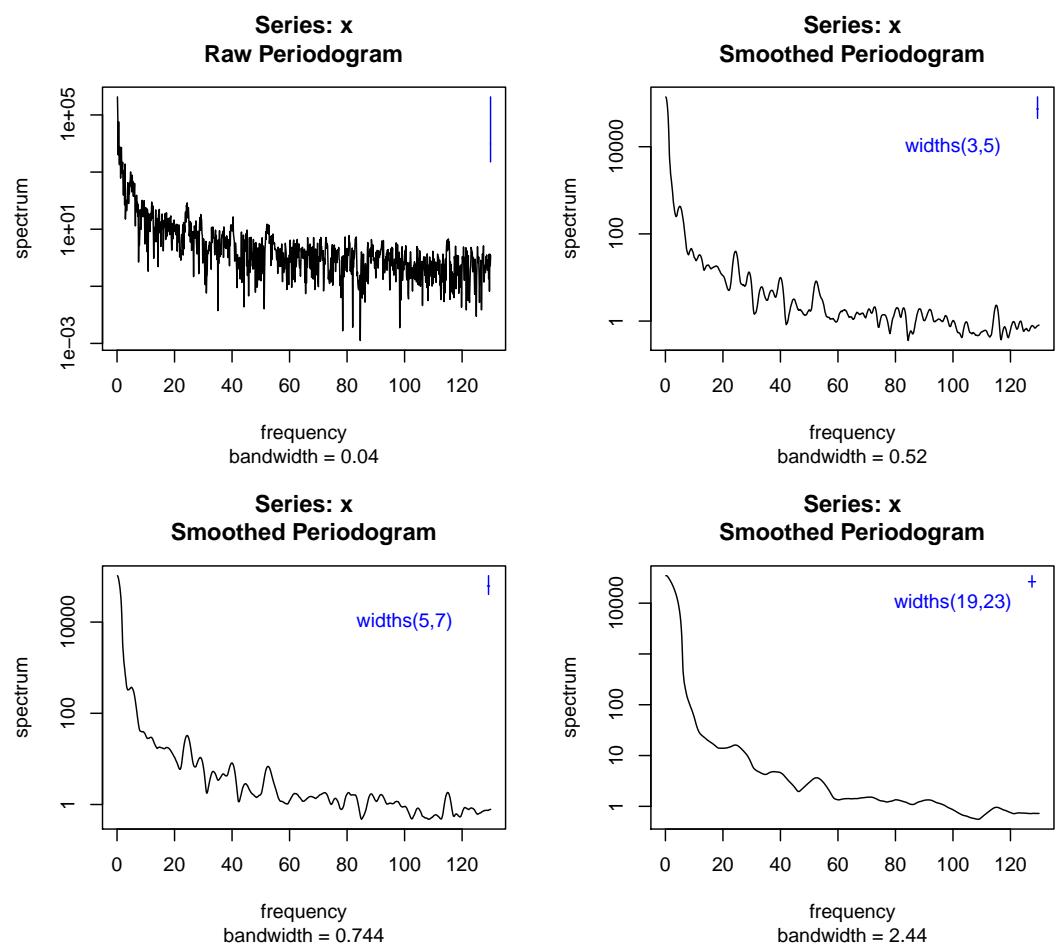


Figure 3.11: Periodogram of DAX and its smoothed versions using Daniell smoothers by kernel.

NASA, pattern recognition, filter design, de-noising [30] etc. .

For any independent scale, data can be analyzed by continuous or discrete ways. Tong says that wavelet analysis is very significant tool for financial market data for short term prediction and the calculation of the variance in relation to particular time scales [54]. In financial markets, it is very important to look time-scale decomposition with multiresolution analysis for traders. For example short-term traders are interested in the detailed parts which are included in wavelet coefficients which are constructed by high-frequency filters and for long-term traders vice versa.

People have used Fourier transform for many years to understand the characteristic of functions or signals in the view of frequency. However, Fourier transform has some drawbacks compared to wavelet transform. Specifically, financial time series like stock market prices includes high-frequency data and Fourier transform is insufficient to analyze these types of data, since it does not have the flexible and detailed parts. On the other hand wavelets ,which is called as mathematical microscope by Burke in [14], have resolution information.

Here are some properties which make us to use wavelet transform rather than Fourier transform:

- Fourier method requires stationary data while wavelet transforms does not requires [45]. In addition to capturing stationary and non-stationary signals, wavelets also catch signals that are aperiodic. Especially these advantages are very important in analyzing financial time series.
- Projections of a data onto Fourier space are fundamentally global where projections of a data onto wavelet space are fundamentally local. Hence, one can go into desired details of data with wavelet transform like using a microscope [57, 14].
- Fourier transform only provides frequency analysis while wavelet transforms ensure that data can be analyzed for any times with the related frequency details. Time-scale information has importance for complex data sets like financial time series.
- Most of wavelet algorithms are faster than Fourier algorithms, even faster than FFT (Fast Fourier Transform).
- Wavelets provide catching breakdown points, discontinuities and self-similarities if there exist.

Both continuous and discrete transforms will be given in some technical and mathematical details in the next subsections. Relation between wavelet transforms and multiresolution analysis will be covered to understand how to cover a function with detail and smooth parts.

3.2.2 Continuous Wavelet Transform

The continuous wavelet transform $X_{u,v}$ of signal $x(t)$ is actually done by dilation and translation of the mother wavelet (ψ) in order to measure the difference in the signal for specific frequency and specific time [46]. Equation 3.27 shows continuous wavelet transform.

$$X_{u,v} = \int_{-\infty}^{\infty} x(t)\psi_{u,v}(t)dt, \quad (3.27)$$

where

$$\psi_{u,v}(t) = |u|^{-\frac{1}{2}} \psi\left(\frac{t-v}{u}\right). \quad (3.28)$$

The parameters u and v are dilation (or scale) and translation (or location) parameters respectively. These parameters make wavelets flexible and easy adapted tools. According to their flexible structure, one can analyze different components of $x(t)$ easily. Original function has only one variable t while new one has two variables u and v . The signal $x(t)$ can be also written from its wavelet coefficients. This implementation is called “synthesizing” and signal can be written as:

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X_{u,v} \psi_{u,v}(t) (u^{-2}) dv du, \quad (3.29)$$

where C_ψ is given in Equation 3.30.

There are some conditions for constructing desirable transforms as Gençay et al. mentioned in [28] according to the mother wavelet ψ . Firstly, $\psi(t)$ needs to satisfy the *admissibility condition* which is given as

$$\int_0^{\infty} \frac{|\mathcal{C}(\omega)|^2}{\omega} d\omega < \infty, \quad (3.30)$$

where ω is frequency and \mathcal{C} is the Fourier transform of $\psi(t)$. Admissibility condition makes the CWT invertible [45]. In addition it signifies that the Fourier transform of $\psi(t)$ disappears where the frequency is zero such as

$$|\mathcal{C}(\omega)|^2 |_{\omega=0} = 0. \quad (3.31)$$

Valens says in [69] that since the Fourier transform at frequency zero is zero, this ensures that the wavelets have a band-pass similar to spectrum.

Condition of being zero for $\psi(0)$ or following condition in Equation 3.32 provides the *finiteness* of C_ψ [28]. So it can be said that wavelets have zero average and therefore it has a shape of wave or wave form.

$$\int_{-\infty}^{\infty} \psi(t) dt = 0, \quad (3.32)$$

Another condition stays for that wavelet function has unit energy (according to its domain integrated squared function) and it can be shown as

$$\int_{-\infty}^{\infty} |\psi(t)|^2 = 1. \quad (3.33)$$

Under the construction of above conditions, the wavelet function must have at least some nonzero coefficients and these nonzero coefficients must cancel out [28, 60].

One can analyze or decompose $x(t)$ by wavelet basis using the above structures. In addition, if Equation 3.30 holds then $x(t)$ can be reconstructed from its wavelet coefficients as given in Equation 3.29.

In practice it is nearly unachievable to use all wavelet coefficients to analyze the signal. One cannot compute almost infinite number of coefficients by hand or by computer. In theory, it is more applicable to use CWT for continuous functions but not for time series or discrete signals as Masset pointed out in [46]. As a result it is better to work with sampled wavelets.

3.2.3 Discrete Wavelet Transform

Critical Sampling

Beside the calculation problem, CWT has also drawback coming from non-essential information. Because the CWT has two variables, they bring much more information than necessary. As a result one needs to struggle with excessive information as well. However we can eliminate the information that we will not use by the help of discretization of the CWT [28]. Two variables of CWT, i.e. u and v , can be reconstructed for any k and j as

$$u = a^{-j} \quad \text{and} \quad v = kba^{-j}. \quad (3.34)$$

For dyadic wavelet bases, a and b are chosen as 2 and 1 respectively [45]. Critical sampling is used for choosing minimum number of coefficients from the CWT without any missing information. The scalars $k = 1, 2, \dots, 2^j$ and $j = 1, 2, \dots, J$ are stays for again translation and scale factors respectively. The scale number can reach its maximum if the dyadic power of scale is equal to number of observations, i.e. $2^J \leq N$ or $J < \log_2(N) < J+1$ where N is number of observations in data set. Consequently, DWT requires dyadic length for transformation of $x(t)$. If the signal has not a dyadic length, one can take away some observations from data set or one can add some observations (generally zeros) to data set to make the length dyadic. First way invites loss of information, so compared to first one second way is more appropriate. Using zeros to complete data set is called “zero padding”. The sample mean or the last values in the series are other examples to pad the signal.

Then for $u = 2^{-j}$ and $v = k2^{-j}$, we use the CWT to write new wavelet transform and mother wavelet such as

$$X_{j,k} = \int_{-\infty}^{\infty} x(t)\psi_{j,k}(t)dt, \quad (3.35)$$

where

$$\psi_{j,k}(t) = 2^{(\frac{j}{2})}\psi(2^jt - k). \quad (3.36)$$

The function $\psi(t)$ is also called as basis function as a result of reproducing all $\psi_{j,k}(t)$ functions. Resolution of the DWT is described in both time and frequency by *critical sampling* with the set of discrete dilations and discrete translations. The wavelet basis

is consists of the mother wavelet and the father wavelet. The father wavelet is given also in the same form of the mother wavelet as

$$\phi_{j,k}(t) = 2^{(\frac{j}{2})} \phi(2^j t - k). \quad (3.37)$$

The DWT is constructed by high-pass (or wavelet) filter and low-pass (or scale) filter that are shown as $h_l = (h_0, h_1, \dots, h_{L-1})$ and $g_l = (g_0, g_1, \dots, g_{L-1})$ respectively where L is length of filters. As a result, the wavelet filter captures high-frequency (i.e. short scales) components and the scale filter captures low-frequency (i.e. long scales) components. These two filters create a *Quadrature Mirror Filter (QMF)* pair [73] and if wavelets are orthonormal, they can be written in the following equality for $l = 0, 1, \dots, L-1$ as [28, 73]

$$h_l = (-1)^l g_{L-1-l} \quad \text{and} \quad g_l = (-1)^{l+1} h_{L-1-l}. \quad (3.38)$$

QMF reduces the aliasing effects which appear while the convolution is done [74].

For desirable frequency gap, it is possible to construct band-pass filter from high-frequency and low-frequency filters. Details are covered by Gençay et al. in [28] between pages 106 and 110.

Wavelet filter has three properties given as below

1. summation to zero:

$$\sum_{l=0}^{L-1} h_l = 0 \quad (3.39)$$

2. unit energy:

$$\sum_{l=0}^{L-1} h_l^2 = 1 \quad (3.40)$$

3. orthogonality to even shifts:

$$\sum_{l=0}^{L-1} h_l h_{l+2n} = \sum_{l=-\infty}^{\infty} h_l h_{l+2n} = 0 \quad \text{for all integers except } n \neq 0 \quad (3.41)$$

Wavelet filter is familiar with the difference operator according to first and second properties [46]. In addition, second property shows that energy is protected in the series [40]. Both second and third property implies the orthogonality, i.e. wavelet coefficients are independent or uncorrelated to construct wavelet coefficient bases [52].

Then according to h_l , the scaling filter g_l has following properties:

1. summation to $\pm\sqrt{2}$:

$$\sum_{l=0}^{L-1} g_l = \sqrt{2} \quad (3.42)$$

2. unit energy:

$$\sum_{l=0}^{L-1} g_l^2 = 1 \quad (3.43)$$

3. orthogonality to even shifts:

$$\sum_{l=0}^{L-1} g_l g_{l+2n} = \sum_{l=-\infty}^{\infty} g_l g_{l+2n} = 0 \quad (3.44)$$

4. orthogonality to wavelet filter at even shifts:

$$\sum_{l=0}^{L-1} g_l h_{l+2n} = \sum_{l=-\infty}^{\infty} g_l h_{l+2n} = 0 \quad (3.45)$$

In the form of wavelet and scaling filters, the father and the mother wavelet can be written as below equations respectively by using dilation equation that is clearly explained in Strang's paper [64]

$$\phi(t) = 2^{1/2} \sum_l g_l \phi(2t - l) \quad \text{and} \quad \psi(t) = 2^{1/2} \sum_l h_l \phi(2t - l), \quad (3.46)$$

where $l = 0, 1, \dots, L - 1$.

Crowley mentioned in [20] that the mother wavelet stands for cyclical part of data and the father wavelet stands for the trend. By the help of flexible structure of wavelets, one can go into details for high-frequency and low-frequency components for any time.

Construction of the DWT

Assume we have time series $x(t)$ with dyadic length $N = 2^J$. Then we can write wavelet and scaling coefficients as

$$\mathbf{w} = \mathbf{W}x, \quad (3.47)$$

where \mathbf{W} is an $N \times N$ orthonormal matrix of DWT and the length of the vector $\mathbf{w} = [w_1, w_2, \dots, w_J, v_J]^T$ is $J + 1$. In order to satisfy Equation 3.47, \mathbf{w} must have length of N . Hence we set the length of w_j as $N/2^j$ and the length of v_J as $N/2^J$ according to scale lengths as $\lambda_j = 2^{j-1}$ and $\lambda_J = 2^{J-1}$ respectively where $j = 1, 2, \dots, J$. Let $h_j^{(0)}$ be unit scale wavelet coefficients. If the length of the filter is L , then elements that are between L and N , namely $L < t < N$, are zero. So the filter vector is given as

$$h_1^{(0)} = [h_1(N-1), h_1(N-2), \dots, h_1(1), h_1(0)]^T. \quad (3.48)$$

Then, we define the orthonormal wavelet coefficients $[h_1^{(2)}, h_4^{(0)}, \dots, h_1^{(\frac{N}{2}-1)}]$ by shifting each element of $h_1^{(0)}$ with length of 2^j in each step. As a result next step gives the vector $h_1^{(2)} = [h_1(1), h_1(0), h_1(N-1), h_1(N-2), \dots, h_1(3), h_1(2)]^T$ and so on. After $\frac{N}{2} - 1$ steps we find the matrix W_1 as given below

$$W_1 = \left[h_1^{(0)}, h_1^{(2)}, h_4^{(0)}, \dots, h_1^{(\frac{N}{2}-1)} \right]^T \quad (3.49)$$

Other matrices W_j 's are set up similarly where the scale factor is $j = 1, 2, \dots, J$. One can write the matrix \mathbf{W} with submatrices as

$$\mathbf{W} = [W_1 \ W_2 \ \dots \ W_J \ V_J]^T, \quad (3.50)$$

where W_j is $(N/2^j) \times N$ and V_J is $(N/2^J) \times N$ matrices where $j = 1, 2, \dots, J$. The matrices W_j and V_J can be written by using the high-pass and low-pass filters. Construction of \mathbf{W} can be also done in different ways like defining submatrices from high-pass and low-pass filters again where sorting of the filters can be different from Equation 3.48. Each submatrix is defined from first quarter of previous submatrix [4]. Let's show a simple construction example of DWT with using *Mallat's pyramid algorithm* which will be illustrated in details after Equation 3.6 [43].

Example 3.6 (Construction Example of DWT). Assume that we are dealing with time series $x(t)$ where number of observations is $N = 16$ and the length of the filter is $L = 4$. Then 8×16 matrices $\{H_1, G_1\}$, 4×8 matrices $\{H_2, G_2\}$, 2×4 matrices $\{H_3, G_3\}$ and 1×2 matrices $\{H_4, G_4\}$ are written as follows by taking first quarter part from previous matrix [6],

$$H_1 = \begin{bmatrix} h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 \\ h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 \end{bmatrix},$$

$$H_2 = \begin{bmatrix} h_1 & h_0 & 0 & 0 & 0 & 0 & h_3 & h_2 \\ h_3 & h_2 & h_1 & h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & h_2 & h_1 & h_0 \end{bmatrix},$$

$$H_3 = \begin{bmatrix} h_1 & h_0 & h_3 & h_2 \end{bmatrix},$$

$$H_4 = \begin{bmatrix} h_1 & h_2 \end{bmatrix},$$

$$G_1 = \begin{bmatrix} g_1 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_3 & g_2 \\ g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 \end{bmatrix},$$

$$G_2 = \begin{bmatrix} g_1 & g_0 & 0 & 0 & 0 & 0 & g_3 & g_2 \\ g_3 & g_2 & g_1 & g_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_3 & g_2 & g_1 & g_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_3 & g_2 & g_1 & g_0 \end{bmatrix},$$

$$G_3 = \begin{bmatrix} g_1 & g_0 & g_3 & g_2 \\ g_3 & g_2 & g_1 & g_0 \end{bmatrix},$$

$$G_4 = [g_1 \ g_2].$$

The relation between the high-pass and low-pass filters is given in the Equation 3.38. Then the matrix \mathbf{W} is written by using *Mallat's pyramid algorithm* as

$$\mathbf{W} = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ V_4 \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 G_1 \\ H_3 G_2 G_1 \\ H_4 G_3 G_2 G_1 \\ G_4 G_3 G_2 G_1 \end{bmatrix}.$$

Then we can get the vector \mathbf{w} by multiplying the original time series by \mathbf{W} . In general, coefficients w_j 's have length of $N/2^j$ and v_J has length of $N/2^J$. In Example 3.6, w_1, w_2, w_3, w_4 and v_4 has 8,4,2,1 and 1 observations respectively.

Pyramid Algorithm

Now we consider the construction of wavelet coefficients for DWT. Let $x(t)$ be time series with N observations: $\{x(1), x(2), \dots, x(N)\}$. The mother and the father wavelet filters convolute with the time series to get first level of the decomposition of the wavelet and the scaling coefficients. Before passing to the '*Pyramid Algorithm*' [43] that describe a iteration algorithm to construct the wavelet and scaling coefficients [43, 46], let's write the first coefficients for both the wavelet and the scaling coefficients as

$$w_1(t) = \sum_{l=0}^{L-1} h_l x(t') \quad \text{and} \quad v_1(t) = \sum_{l=0}^{L-1} g_l x(t'), \quad (3.51)$$

where $t = 0, 1, \dots, (\frac{N}{2}) - 1$ and $t' = 2t + 1 - l \pmod{N}$. Wavelet coefficients at the first scale includes the highest frequency components of x . It is seen that the number of samples is divided into half of its value for $w_1(t)$ and $v_1(t)$. It is the preliminary of the first step of pyramid algorithm. The variable t' ensures the subsampling and positiveness of step of x . Masset point out that modulus stays for solving boundary problems [46]. Construction of the pyramid algorithm takes shape from iteration of the scaling coefficients in the Equation 3.51. In each step, sample size is decreases as the half of the previous step (downsampling by 2). For example in the second step time variables are given as $t = 0, 1, \dots, \frac{N}{4} - 1$ and $t' = 2t + 1 - l \pmod{N}$ for $w_2(t)$ and $v_2(t)$.

After J step iterations where $J = [\log_2 N]$ is the scale, one can get the wavelet and the scaling coefficients as

$$w_J(t) = \sum_{l=0}^{L-1} h_l v_{J-1}(t') \quad \text{and} \quad v_J(t) = \sum_{l=0}^{L-1} g_l v_{J-1}(t'), \quad (3.52)$$

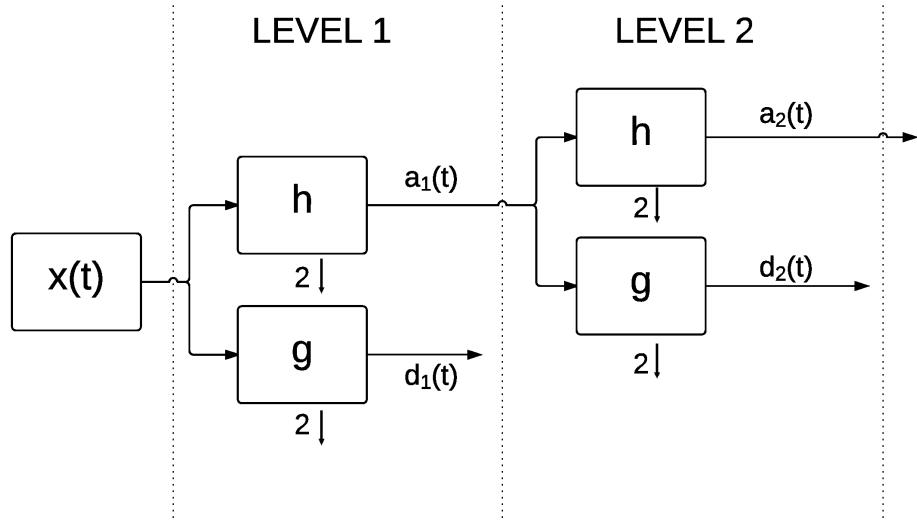


Figure 3.12: Flowchart of the pyramid algorithm for decomposing. In every iteration each coefficients divided by 2 which is called downsampling as mentioned by Gençay et al. in [28]. Frequency decreases as j increases.

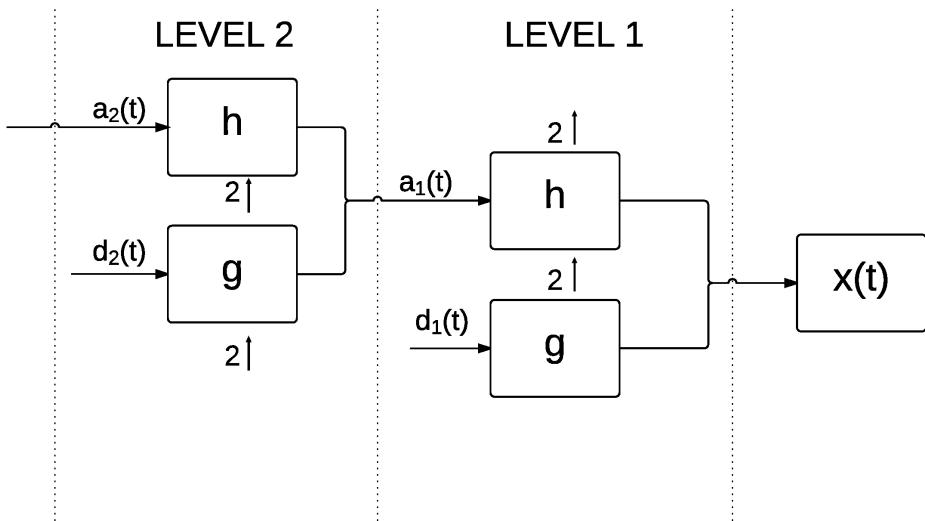


Figure 3.13: Flowchart of the pyramid algorithm for synthesis. In every iteration each coefficients multiplied by 2 which is called upsampling as mentioned by Gençay et al. in [28].

where $t = 0, 1, \dots, (\frac{N}{2^J}) - 1$ and $t' = 2t + 1 - l \bmod N$. At the end of iterations, vector space with wavelet and scaling coefficients occurs as below,

$$\mathbf{w} = [w_1 \ w_2 \ \dots \ w_J \ v_J]^T. \quad (3.53)$$

This vector contains the all frequency components according to our choice of scale band in related times. It is also known as the filter bank approach. It is not necessary to implement all scales, i.e. scale can be less than J . Then it is called partial discrete wavelet transform (PDWT).

In the above constructions we used high-pass and low-pass filters for the decomposition of the time series $x(t)$. Reverse operation for the DWT is used for reconstruction of the original time series $x(t)$. Reconstruction is done by opposite iteration of the last level of the wavelet and scaling coefficients using the related filters [28]. Down-sampling is used for decomposition. On the contrary, upsampling is employed in each iteration for reconstruction of time series. Then the vectors w_J and v_J are given as

$$w_J^0 = [0, w_J(0)]^T \quad \text{and} \quad v_J^0 = [0, v_J(0)]^T. \quad (3.54)$$

Then the scaling coefficient v_{J-1} is

$$v_{J-1}(t) = \sum_{l=0}^{L-1} h_l w_J^0(t') + \sum_{l=0}^{L-1} g_l v_J^0(t'), \quad (3.55)$$

where $t = 0, 1$ and $t' = t + l \bmod 2$. Then second step is similarly done with upsampling such that

$$v_{J-2}(t) = \sum_{l=0}^{L-1} h_l w_{J-2}^0(t') + \sum_{l=0}^{L-1} g_l v_{J-2}^0(t'), \quad (3.56)$$

where $t = 0, 1, 2, 3$ and $t' = t + l \bmod 4$. If the iteration procedure is keeping until the first level of wavelet and scaling coefficients by using upsampling, the original data set can be obtained as

$$x(t) = \sum_{l=0}^{L-1} h_l w_1^0(t') + \sum_{l=0}^{L-1} g_l v_1^0(t'), \quad (3.57)$$

where $t = 0, 1, 2, \dots, N - 1$ and $t' = t + l \bmod N$.

Vuorenmaa points out in [71] that Mallat's pyramid algorithm is also called as Fast Wavelet Transform (FWT), because the algorithm requires multiplications at most of order N rather than DWT's multiplications which is at most of order N^2 . He is also adding that Fast Fourier Transform requires at most $N \log_2 N$ multiplications which has larger calculation steps than FWT as well. As a result, FWT is very effective algorithm for big data sets.

3.2.4 Maximal Overlap Discrete Wavelet Transform

Discrete wavelet transform (DWT) has some drawbacks due to skeleton of the construction. For instance, DWT need a data which has dyadic length according to using downsampling and upsampling in analysis and synthesis respectively. Otherwise one needs to add some observations or remove some observations to make the length of data dyadic. MODWT, which does not employ upsampling and downsampling, does not require dyadic length whereas DWT restrict the data for length 2^{J_k} where $0 < J_k < J + 1$. Moreover, Gençay et al. states in [28] that the MODWT is unvarying to circularly shifting the data set where the DWT does not hold that property. In addition, it is mentioned that the variance analysis of MODWT is more effective than the variance analysis of DWT [53].

One can get the MODWT by setting u and v as 2^{-j} and k respectively. It means that scaling parameter is same as DWT's parameter, but translating parameter provides that in each scale there are same number of coefficients in the MODWT while parameter is $k2^{-j}$ in DWT. In other words number of wavelet and scaling coefficients are equal to length of the original time series in at every step of the transform since subsampling is not implemented in the MODWT differently from the DWT [16]. Other steps are done in similar ways as done in the DWT.

Consider the same time series that is used in the DWT construction and let $\tilde{\mathbf{w}}$ be wavelet and scaling coefficients of the MODWT,

$$\tilde{\mathbf{w}} = [\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_J, \tilde{v}_J]^T, \quad (3.58)$$

where length of \tilde{w}_j is $N/2^j$ and the length of \tilde{v}_J is $N/2^J$ according to scale lengths as $\lambda_j = 2^{j-1}$ and $\lambda_J = 2^{J-1}$ respectively where $j = 1, 2, \dots, J$. DWT wavelet and scaling coefficients can be captured by subsampling and rescaling the MODWT coefficients, mentioned by Gençay et al. in [28]. Relations for wavelet coefficients can be written as

$$w_j(t) = 2^{j/2} \tilde{w}_j(2^j(t+1) - 1), \quad (3.59)$$

where $t = 0, 1, \dots, \frac{N}{2^j} - 1$ and relation for scaling coefficient can be written as

$$v_J(t) = 2^{J/2} \tilde{v}_J(2^J(t+1) - 1), \quad (3.60)$$

where $t = 0, 1, \dots, \frac{N}{2^J} - 1$.

In similar way to the DWT, the vector $\tilde{\mathbf{w}}$ can be constructed using high-pass and low-pass filters. Details can be found in Gençay et al. between pages 135-137 [28]. The vector of coefficients is written such as

$$\tilde{\mathbf{w}} = \tilde{\mathbf{W}}x, \quad (3.61)$$

where $\tilde{\mathbf{W}} = [\tilde{W}_1, \tilde{W}_2, \dots, \tilde{W}_J, \tilde{V}_J]^T$ is $(J+1)N \times N$ matrix which implies that each \tilde{W}_j and \tilde{V}_J are $N \times N$ matrices. It ensures that in each scale level number of the coefficients is equal to number of the observations in the time series that is N . It shows that the MODWT can handle with any size of data set.

In order to compute the MODWT, ‘*pyramid algorithm*’ that is showed in Figure 3.12 and in Figure 3.13 is applied without using downsampling (or upsampling in the opposite direction). Algorithm again starts with convolving the time series with high-pass and low-pass filters to get first level wavelet and scaling coefficients as following

$$\tilde{w}_1(t) = \sum_{l=0}^{L-1} \tilde{h}_l x(\dot{t}) \quad \text{and} \quad \tilde{v}_1(t) = \sum_{l=0}^{L-1} \tilde{g}_l x(\dot{t}), \quad (3.62)$$

where $t = 0, 1, \dots, N - 1$ and $\dot{t} = t - l \bmod N$. In order to get second level of wavelet and scaling coefficients, $\tilde{v}_1(t)$ is convolved with the high-pass filter \tilde{h}_l and low-pass filter \tilde{g}_l . After J iterations, where $J = \log_2 N$, wavelet and scaling coefficients are written as

$$\tilde{w}_J(t) = \sum_{l=0}^{L-1} \tilde{h}_l \tilde{v}_{J-1}(\dot{t}) \quad \text{and} \quad \tilde{v}_J(t) = \sum_{l=0}^{L-1} \tilde{g}_l \tilde{v}_{J-1}(\dot{t}), \quad (3.63)$$

where $t = 0, 1, \dots, N - 1$ and $\dot{t} = t - 2^{J-1}l \bmod N$. Then decomposition gives the vector $\tilde{\mathbf{w}}$ as $\tilde{\mathbf{w}} = [\tilde{w}_1 \ \tilde{w}_2 \ \dots \ \tilde{w}_J \ \tilde{v}_J]^T$.

Reverse iteration is also done with similar way used in DWT. Last level of wavelet and scaling coefficients are convolved with high-pass filter and low-pass filter respectively. Afterward, summation of two convolving parts gives the scaling coefficient of previous level as given below [28],

$$\tilde{v}_{J-1}(t) = \sum_{l=0}^{L-1} \tilde{h}_l \tilde{w}_J(\dot{t}) + \sum_{l=0}^{L-1} \tilde{g}_l \tilde{v}_J(\dot{t}), \quad (3.64)$$

where $t = 0, 1, \dots, N - 1$ and $\dot{t} = t + l \bmod N$. Iteration can be continued up to first level of wavelet and scaling coefficients to get the original time series as

$$x(t) = \sum_{l=0}^{L-1} \tilde{h}_l \tilde{w}_1(\dot{t}) + \sum_{l=0}^{L-1} \tilde{g}_l \tilde{v}_1(\dot{t}), \quad (3.65)$$

where $t = 0, 1, \dots, N - 1$ and $\dot{t} = t + l \bmod N$.

Example 3.7 (MODWT Example). Assume that we are dealing with time series $x(t)$ where number of observations is $N = 8$ and the length of the filter is $L = 2$. Then 8×8 matrix \tilde{H}_1 is written as follows,

$$\tilde{H}_1 = \begin{bmatrix} \tilde{h}_1 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{h}_2 \\ \tilde{h}_2 & \tilde{h}_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{h}_2 & \tilde{h}_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{h}_2 & \tilde{h}_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{h}_2 & \tilde{h}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \tilde{h}_2 & \tilde{h}_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \tilde{h}_2 & \tilde{h}_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{h}_2 & \tilde{h}_1 \end{bmatrix}.$$

\tilde{H}_2 is calculated by replacing \tilde{h}_1 by \tilde{h}_2 in \tilde{H}_1 and \tilde{H}_3 is calculated by replacing \tilde{h}_1 by \tilde{h}_3 in \tilde{H}_1 [71]. Similarly 8×8 matrices \tilde{H}_1 , \tilde{H}_2 and \tilde{H}_3 are written in the same way with filters \tilde{g}_1 , \tilde{g}_2 and \tilde{g}_3 .

Then the matrix $\tilde{\mathbf{W}}$ is written by using again *Mallat's pyramid algorithm* as

$$\tilde{\mathbf{w}} = \begin{bmatrix} \tilde{W}_1 \\ \tilde{W}_2 \\ \tilde{W}_3 \\ \tilde{V}_3 \end{bmatrix} = \begin{bmatrix} \tilde{H}_1 \\ \tilde{H}_2 \tilde{G}_1 \\ \tilde{H}_3 \tilde{G}_2 \tilde{G}_1 \\ \tilde{G}_3 \tilde{G}_2 \tilde{G}_1 \end{bmatrix}.$$

It is seen that \tilde{W}_1 , \tilde{W}_2 , \tilde{W}_3 and \tilde{V}_3 have 8×8 entries which shows that there is no downsampling.

The MODWT is called by other names as “translation invariant DWT”, “time invariant DWT”, “redundant DWT”, “stationary DWT” and “undecimated DWT” [53, 28]. Both DWT and MODWT provide a scale based multiresolution analysis and scale based variance analysis. Pyramid algorithm can be applied to both transform. However they have some differences due to the construction of transforms. The DWT requires time series with dyadic length and it is affected by circular shifts to time series. On the other hand, the MODWT can handle with time series with any number of observations. In opposition to the DWT, the MODWT is not influenced by circular shifts. The MODWT gives always same coefficients although there is a time shift in original time series. In addition the MODWT is not orthonormal where it causes being highly redundant as Percival and Walden pointed out in [53].

3.2.5 Multiresolution Analysis

Definition 3.1 (Multiresolution Analysis). MRA algorithm divides data into vectors of coefficients which are linked to the specific time scales. MRA is a series of closed nested subspaces $\{V_j; j \in \mathbb{Z}\}$ in $L_2(\mathbb{R})$ with the conditions below, [26, 71, 70]

1. $\{\phi(x - k); k \in \mathbb{Z}\}$ is an orthonormal basis for V_0 where ϕ is scaling function of the MRA
2. $\{0\} \subset \dots \subset V_j \subset V_{j+1} \subset \dots \subset L_2(\mathbb{R})$
3. Closure of $(\cup_{j \in \mathbb{Z}} V_j) = L_2(\mathbb{R})$ such that any union is dense in $L(\mathbb{R})$
4. $\cap_{j \in \mathbb{Z}} V_j = \{0\}$ so zero element is the only common object
5. $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$ so V -spaces are self-similar
6. $V_{j+1} = V_j \oplus W_j$ where the space W_j is the j th resolution level of the MRA [71] and $V_j \cap W_j = \{0\}$

In first property, it is seen that if we translate any function in V_0 then we get new function again in V_0 . Similarly, wavelet function (or mother wavelet) ψ creates an orthonormal basis for V_j that is orthogonal complement of V_j in V_{j+1} [26]. This implies that in general form, scaling (ϕ) and wavelet (ψ) functions are basis for V_j and W_j subspaces respectively where W_j is difference of V_{j+1} and V_j . In practice we start from subspace V_0 and then add other subspaces to enlarge space. Second property shows the nested space. Third and fourth properties shows the limit of space.

We are using dilation and translation parameters to cover the $L_2(\mathbb{R})$ space. Main idea of the MRA is writing $L_2(\mathbb{R})$ space or any function in it by combination of detail and smooth parts.

One can get V_{j+1} from last property in Definition 3.1 as

$$V_{j+1} = V_0 \oplus (W_0 \oplus W_1 \oplus \cdots \oplus W_j). \quad (3.66)$$

Fryzlewicz pointed out in [26] that one can take limit in Equation 3.66 depending on the third property in Definition 3.1.

$$L_2(\mathbb{R}) = V_0 \oplus \left(\bigoplus_{i=0}^{i=\infty} W_i \right) = V_{j_0} \oplus \left(\bigoplus_{i=j_0}^{i=\infty} W_i \right) \quad (3.67)$$

for any j_0 . Equation 3.67 shows that any data or signal can be written as large scale features described by V_j subspaces and small scale features (details) can be described by W_j subspaces.

According to constructions of the DWT and the MODWT, original time series can be written by the combination of the wavelet and scaling coefficients. In Equation 3.47 x can be written by multiplying both sides by \mathbf{W}^T . Since \mathbf{W} is orthogonal matrix, its transpose is the inverse of itself, i.e. $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. Then we get $x = \mathbf{W}^T\mathbf{w}$. In Equation 3.61, $\tilde{\mathbf{W}}$ is not orthogonal matrix but we can find the parts of original time series in each scale level by multiplying transpose of j th level transform matrix by related coefficient matrix. For example we can find j th detail level of $x(t)$ as

$$\tilde{d}_j = \tilde{\mathbf{W}}_j^T \tilde{\mathbf{w}}_j.$$

Firstly let's define the reconstruction of the MRA for the DWT, and then do it for MODWT. In Example 3.6, we defined the \mathbf{W} by the pyramid algorithm. If we write it in general form, x can be written as

$$\begin{aligned} x = & H_1 w_1 + G_1 H_2 w_2 + G_1 G_2 H_3 w_3 + \cdots \\ & + G_1 \cdots G_{J-1} H_J w_J + G_1 \cdots G_{J-1} G_J v_J. \end{aligned} \quad (3.68)$$

In the additive decomposition components are called as detail and smooth by Percival and Walden in [53]. Detail and smooth components are given for $j = 1, 2, \dots, J$ as

$$d_j = G_1 G_2 \dots G_{j-1} H_j w_j, \quad (3.69)$$

and

$$s_J = G_1 G_2 \dots G_{J-1} H_J v_J. \quad (3.70)$$

As a result the data x can be written as

$$x = d_1 + d_2 + \cdots + d_J + s_J = \sum_{j=1}^J d_j + s_J. \quad (3.71)$$

Equation 3.71 is illustration of the MRA. The last component s_J includes the average information (or trend) of the original data at the largest scale and it is associated with the scaling coefficients. Components d_j 's, from first scale to last scale, are related to wavelet coefficients. They are deployed for catching higher frequency information, i.e. they include detail coefficients [46].

All steps are similar in the MODWT. According to pyramid algorithm, x can be written as

$$\begin{aligned} x = & \tilde{H}_1 \tilde{w}_1 + \tilde{G}_1 \tilde{H}_2 \tilde{w}_2 + \tilde{G}_1 \tilde{G}_2 \tilde{H}_3 \tilde{w}_3 + \cdots \\ & + \tilde{G}_1 \cdots \tilde{G}_{J-1} \tilde{H}_J \tilde{w}_J + \tilde{G}_1 \cdots \tilde{G}_{J-1} \tilde{G}_J \tilde{v}_J. \end{aligned} \quad (3.72)$$

Using the variables of the MODWT and put those into Equation 3.69 and Equation 3.70 x can be decomposed as

$$x = \tilde{d}_1 + \tilde{d}_2 + \cdots + \tilde{d}_J + \tilde{s}_J = \sum_{j=1}^J \tilde{d}_j + \tilde{s}_J. \quad (3.73)$$

It is not necessary to use all scales for the resolution analysis for both DWT and MODWT. One can choose the scale which is less than J , i.e., $1 < J_p < J$. We have already mentioned choosing J_p as partial wavelet transform. Masset says in [46] that it is also possible to estimate the original time series by using the wavelet and scaling coefficients where the scale is between J_p and J . One of the objectives of doing the composition in that way is separating out the noise or the seasonality from the data.

3.2.6 Wavelet Variance Analysis

We mentioned that Equation 3.33 and Equation 3.43 stand for showing the unit energy in the wavelet and the scaling filters. Sample variance of a time series can be decomposed scale-by-scale as a result of energy protection in the DWT and the MODWT [28, 71]. It is seen easily that

$$\|x\|^2 = x^T x = (\mathbf{W}^T \mathbf{w})^T \mathbf{W}^T \mathbf{w} = \mathbf{w}^T \mathbf{W} \mathbf{W}^T \mathbf{w} = \mathbf{w}^T \mathbf{w} = \|\mathbf{w}\|^2, \quad (3.74)$$

where \mathbf{W} is $N \times N$ orthonormal matrix and \mathbf{w} is the vector which contains the wavelet and scaling coefficients as defined in construction of the DWT. As a result one can write energy of x as

$$\|\mathbf{w}\|^2 = \sum_{j=1}^J \sum_{t=0}^{(N/2^j)-1} w_j^2(t) + v_J^2(0) = \sum_{t=0}^{N-1} x^2(t) = \|x\|^2. \quad (3.75)$$

Simply, energy of x can be written as follows by using the orthonormality of elements with respect to the matrix \mathbf{W} as Gençay et al. [28] referred to

$$\|x\|^2 = \sum_{j=1}^J \|w_j\|^2 + \|v_J\|^2 = \sum_{j=1}^J \|d_j\|^2 + \|s_J\|^2. \quad (3.76)$$

Then decomposition of the sample variance for x can be written as

$$\hat{\sigma}_x^2 = \frac{1}{N} \sum_{t=0}^{N-1} (x(t) - \bar{x})^2 = \frac{1}{N} \sum_{j=1}^J \|w_j\|^2 + \frac{1}{N} \|v_J\|^2 - \bar{x}^2. \quad (3.77)$$

where \bar{x} denotes the sample mean of x and written as

$$\bar{x} = \frac{1}{N} \sum_{t=0}^{N-1} x(t). \quad (3.78)$$

Wavelet spectrum can be regarded as the energy of \mathbf{w} and given as

$$\|\mathbf{w}\|^2 = \sum_{j=1}^J \|w_j\|^2 + \|v_J\|^2, \quad (3.79)$$

and if we have the condition $\frac{1}{N} \|v_J\|^2 = \bar{x}^2$, the sample variance for x is

$$\hat{\sigma}_x^2 = \frac{1}{N} \sum_{j=1}^J \|w_j\|^2. \quad (3.80)$$

Let's consider the MODWT as a next step. Energy is protected as well in the MODWT like being in the DWT. As a result we get

$$\|x\|^2 = \sum_{j=1}^J \|\tilde{w}_j\|^2 + \|\tilde{v}_J\|^2. \quad (3.81)$$

Unlike the DWT, $N \times N$ matrix $\tilde{\mathbf{W}}$ is not orthonormal in the MODWT. Consequently Equation 3.76 does not hold for the MODWT and so [28, 40]

$$\|\tilde{w}_j\|^2 \neq \|\tilde{d}_j\|^2, \quad (3.82)$$

and

$$\|\tilde{d}_j\|^2 = \|\tilde{\mathbf{W}}^T \tilde{w}_j\|^2 = \tilde{w}_j^T \tilde{\mathbf{W}}_j \tilde{\mathbf{W}}_j^T \tilde{w}_j. \quad (3.83)$$

Gençay et al. [28] referenced Percival and Walden [53] that $\|\tilde{d}_j\|^2 \leq \|\tilde{w}_j\|^2$. Hence it is seen that the MODWT wavelet coefficients are important in variance analysis, not the wavelet details [28].

Estimation of Wavelet Variance

Consider the j th level MODWT coefficients as

$$\tilde{w}_j(t) = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} x_{t-l \mod N} \quad \text{and} \quad \bar{\tilde{w}}_j(\dot{t}) = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} x_{t-l}, \quad (3.84)$$

where $t = 0, 1, \dots, N-1$ and $\dot{t} = 0, \pm 1, \dots, \pm(N-1)$. Then $\tilde{w}_j(t)$ and $\bar{\tilde{w}}_j(\dot{t})$ are equal if the condition $L_j - 1 \leq t < N$ [52]. Unbiased estimator of $\sigma_x^2(\lambda_j)$ is calculated for $N - L_j \geq 0$ as

$$\hat{\sigma}_x^2(\lambda_j) = \frac{1}{\hat{N}_j} \sum_{t=L_j-1}^{N-1} \tilde{w}_j^2(t) = \frac{1}{\hat{N}_j} \sum_{t=L_j-1}^{N-1} \bar{\tilde{w}}_j^2(t), \quad (3.85)$$

where $\hat{N}_j = N - L_j + 1$ and $L_j = (2^j - 1)(L - 1) + 1$. As Crowley mentioned in [20] that \hat{N}_j is the number of coefficients which are not influenced by boundary effect. Beside that number of the MODWT coefficients that are computed with boundary is represented by L_j . Scale level and length of the filter are showed by λ_j and L respectively. Differently from the MODWT, t is between $L'_j = (L - 2)(1 - 2^{-j})$ and $N/2^j - 1$ and equation is normalized by the factor $2\lambda_j$ in the estimation of the unbiased estimator of $\sigma_x^2(\lambda_j)$ [28].

Referred to Percival [52], one can construct the biased estimator of $\sigma_x^2(\lambda_j)$ as (here the MODWT is used)

$$\tilde{\sigma}_x^2(\lambda_j) = \frac{1}{N} \sum_{t=0}^{N-1} \tilde{w}_j^2(t) = \frac{1}{N} \left(\sum_{t=0}^{L_j-2} \tilde{w}_j^2(t) + \sum_{t=L_j-1}^{N-1} \bar{\tilde{w}}_j^2(t) \right). \quad (3.86)$$

In Equation 3.86 the sum from zero to $L_j - 2$ is affected by the periodicity.

In general the wavelet variance is determined for both stationary and non-stationary processes. If the process is stationary then time-independent wavelet variance is considered for analysis. Time-dependent and time-independent wavelet variances are showed as $\sigma_{x,t}^2(\lambda_j)$ and $\sigma_x^2(\lambda_j)$ respectively. Sometimes they are called also as wavelet spectrum [71]. We can use the coefficients $\bar{\tilde{w}}_j^2(t)$ in Equation 3.84 for general purpose to write the wavelet spectrum as

$$\sigma_{x,t}^2(\lambda_j) = \text{var}(\bar{\tilde{w}}_j^2(t)). \quad (3.87)$$

Then the variance of x_t is given as

$$\text{var}(x_t) = \sum_{j=1}^{\infty} \sigma_{x,t}^2(\lambda_j). \quad (3.88)$$

For DWT scale is $\lambda_j = 2^{j-1}$ and it is in the frequency band of $[1/2^{j+1}, 1/2^j]$. Moreover the wavelet spectrum for the DWT is

$$\sigma_{x,t}^2(\lambda_j) = \frac{1}{2\lambda_j} \text{var}(w_j(t)). \quad (3.89)$$

Spectral density function (SDF) separated the variance in a way of frequency-by-frequency whereas the wavelet spectrum decomposes the variance on a scale-by-scale basis. The connection of two spectrums can be written as

$$\sigma_{x,t}^2(\lambda_j) \approx 2 \int_{1/2^{j+1}}^{1/2^J} S_x(f) df, \quad (3.90)$$

where $S_x(f)$ is called as periodogram and given as

$$S_x(f_k) = \frac{1}{N} \left| \sum_{t=0}^{N-1} x_t e^{-2\pi f_k t} \right|^2. \quad (3.91)$$

In Equation 3.91 $f_k = k/N$ and k represents the k th Fourier frequency for $k = 0, 1, \dots, N/2$.

Confidence Interval for the Wavelet Variance

Let x be our time series with N observations, $\{x_0, x_1, \dots, x_{N-1}\}$, with mean $\mu = \mathbb{E}(x_t)$. The variance of the sample mean is $\text{var}(\bar{x}) = \frac{\sigma^2}{N}$ with $\bar{x} = \frac{1}{N} \sum_{t=0}^{N-1} x_t$. Then $(1 - \alpha)$ confidence interval (CI) for the mean can be defined as [28]

$$\bar{x} \pm \xi_{\frac{\alpha}{2}} \left(\frac{\hat{\sigma}}{\sqrt{N}} \right), \quad (3.92)$$

for unknown σ^2 . Moreover unbiased estimator of σ is

$$\hat{\sigma}^2 = (N - 1)^{-1} \sum_t (x_t - \bar{x})^2. \quad (3.93)$$

Basicly we can write the confidence interval as

$$\mathbb{P} \left[\bar{x} - \xi_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \leq \mu \leq \bar{x} + \xi_{\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} \right] = (1 - \alpha). \quad (3.94)$$

CHAPTER 4

EMPIRICAL RESULTS

We will consider the daily closing prices of S&P500 data that is observed from *Yahoo Finance* web page (<http://ichart.finance.yahoo.com/table.csv>). Length of the data is 21 years which is between 01.01.1990 and 01.01.2011. There are 5444 observations in data set.

4.1 Characteristic of S&P500 Data

First of all we have looked recurrence plots, histograms, Q-Q plots, ACF and PACF plots and some statistical tests to understand both daily closing prices and return values of S&P500 data. In order to see details and to compare result, we will consider both S&P500 daily closing prices and its return values. As we mentioned before, return series is found by logarithmic difference of the original time series. Plots of daily closing prices and return values are given in Figure 4.1.

Recurrence Plots to See Characteristics of Time Series

Idea of recurrence plot is that if the phase space trajectory of the dynamical system passes through the same area in the phase space, then it gives the shortest distance. In other words, it compares two vector of the system and if the vector meets itself then the *Euclidean* distance will be zero. We can write vectors for time series x_i with N observations as,

$$v(i) = \{x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(m-1)\tau}\}^T,$$

where m is embedding dimension and τ is the time delay. The corresponding recurrence matrix of embedding vectors is given as,

$$R_{i,j}(\varepsilon) = \begin{cases} 0 & \text{if } \|v(i) - v(j)\| > \varepsilon \\ 1 & \text{if } \|v(i) - v(j)\| \leq \varepsilon \end{cases}, \quad \text{for } i, j = 1, \dots, n = N - (m-1)\tau,$$

where $\|\cdot\|$ is the *Euclidean* distance, and ε is threshold for distance [11]. So in our recurrence plots, lighter shades shows longer distances while darker shades represent shorter distances.

Recurrence plots gives some characteristics of the time series to show which type the time series is. Some types can be given as Wiener process (Brownian Motion), Gaus-

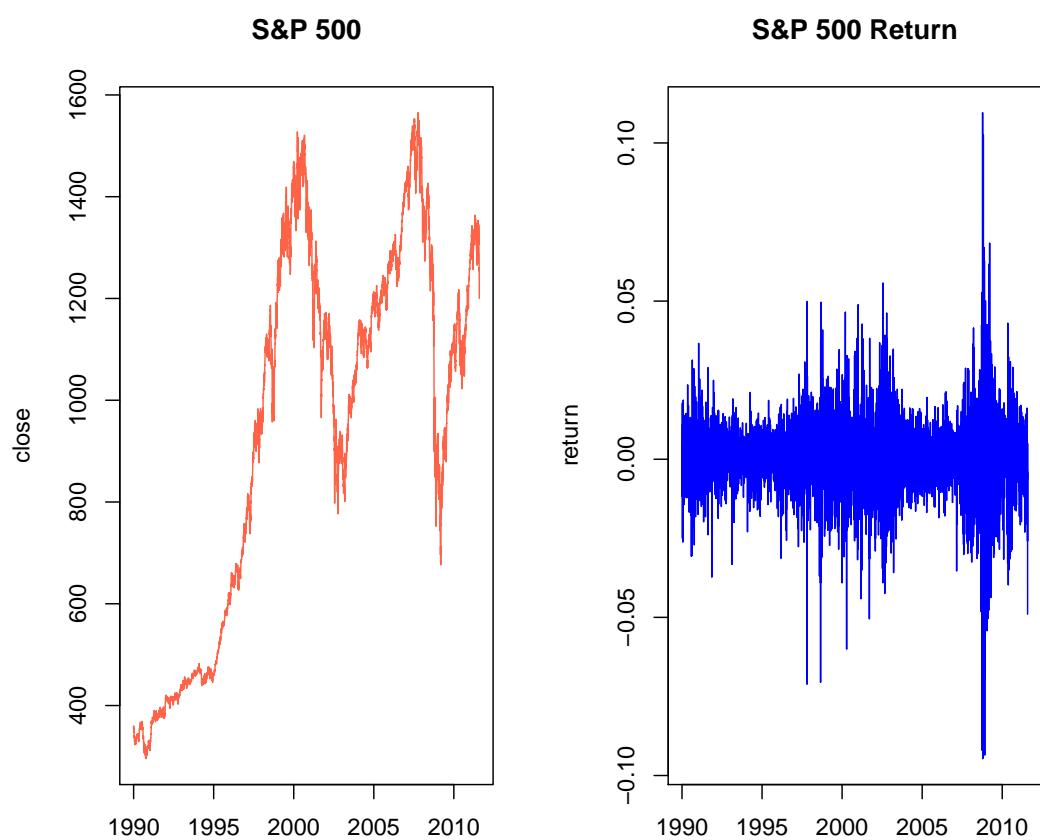
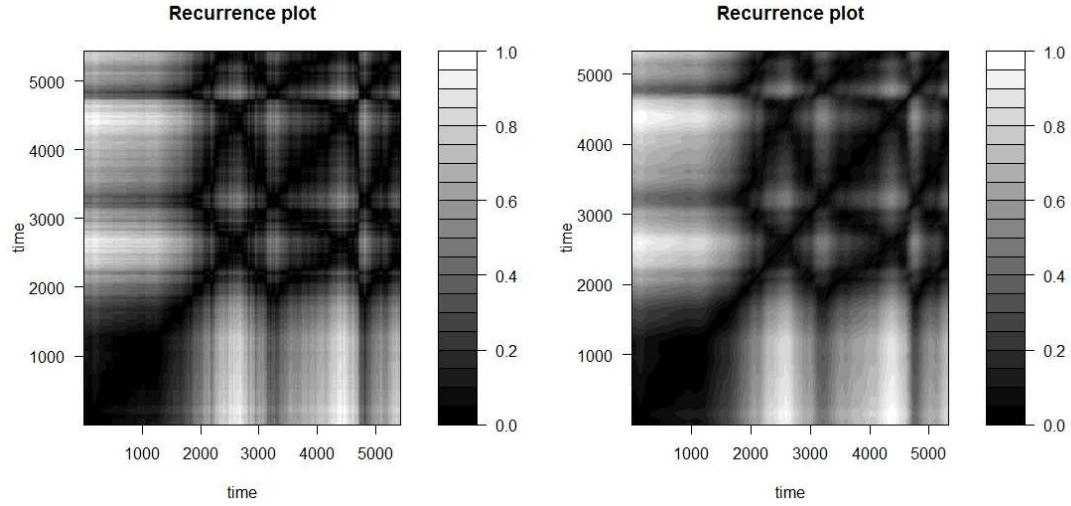


Figure 4.1: S&P500 Daily Closing Price Data and Return Data of Daily Closing Prices



(a) Embedding Dimension ($m = 12$) and Time Delay ($d = 1$) (b) Embedding Dimension ($m = 60$) and Time Delay ($d = 2$)

Figure 4.2: Recurrence Plots of S&P500 Daily Closing Prices.

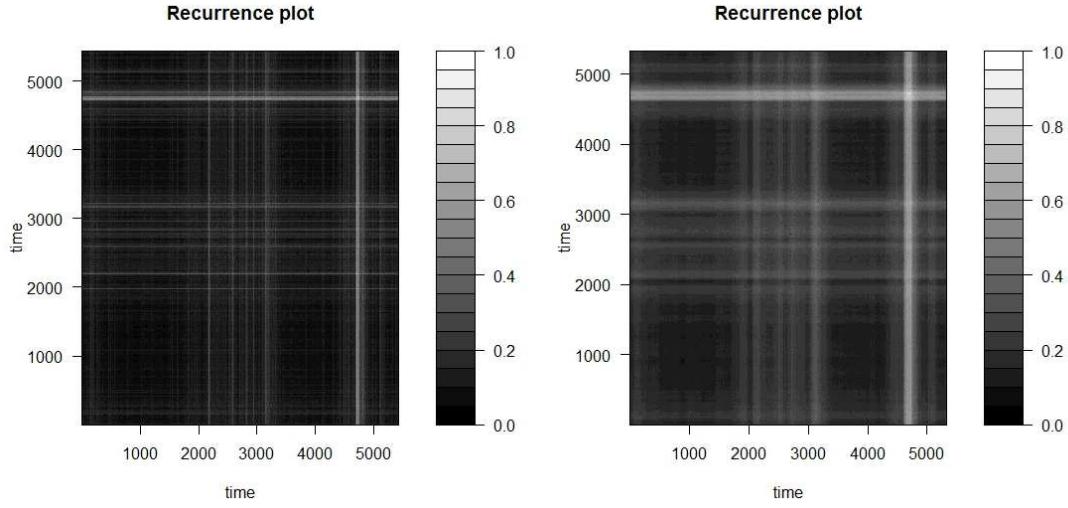
sian white noise, periodic signal (sine and cosine signals), pure trend signal, log periodic signal etc. [24].

In our recurrence graph lighter shades show longer distances. Fabretti and Ausloos in [24] state that homogeneity in recurrence plot shows that the signal has stationary characteristic. Therefore as we see in the Figure 4.3, return values are stationary. On the other hand, in Figure 4.2 white and light gray bands illustrate the non-stationary parts. Hence, daily closing prices show non-stationary characteristics and divided trend parts [24]. One can link S&P500 Daily Closing Prices with Wiener process (Brownian motion). Then again one can link the return values with Gaussian white noise [51, 50].

Histograms, Q-Q Plots, Normality Tests, ACF and PACF

There may be missing values in historical data sets. Therefore we have to fill the missing values in both closing prices data and return data to work with them. Simply we fill the missing values with the average of the observed values. In Figure 4.4 normal Q-Q plot shows that distribution is mixture of positive skewed distributions and negative skewed distributions. In Figure 4.5, it can be realized return values are not normally distributed. The bulk of the daily return series for S&P500 is between -4% and $+4\%$. However Q-Q plot of return series has clear tilted S-shape which deviates from linearity in both left and right tails of the distribution. The daily return of S&P500 has much fatter tails than the normal distribution. In Figure 4.6, Q-Q plots according to normal, log-normal, uniform and student-t distributions are given. It is seen that none of distributions exactly fit our return data. Student-t distribution gives the least deflection from the line.

We can apply some normality tests to our data set as well. For example Shapiro-Wilk normality test can be applied, but it restricts the sample size between 3 and 5000.



(a) Embedding Dimension ($m = 12$) and Time Delay ($d = 1$) (b) Embedding Dimension ($m = 60$) and Time Delay ($d = 2$)

Figure 4.3: Recurrence Plots of S&P500 Return Values.

Therefore we use the first 5000 observations in the test. Test rejects being normally distributed if $p\text{-value} \leq 0.05$. The results for daily closing prices, return prices and first difference of daily closing prices are given in Table 4.1.

Table 4.1: Shapiro-Wilk and Kolmogorov-Smirnov Tests Results for Normality for Daily Closing Prices (DCP), Return Values (RV), and First Difference of Daily Closing Prices (FD-DCP).

	Shapiro-Wilk normality test		Kolmogorov-Smirnov test	
	W	$p\text{-value}$	D	$p\text{-value}$
DCP	0.9186	$<2.2e^{-16}$	0.1299	$<2.2e^{-16}$
RV	0.9104	$<2.2e^{-16}$	0.082	$<2.2e^{-16}$
FD-DCP	0.8902	$<2.2e^{-16}$	0.1197	$<2.2e^{-16}$

According to results of Shapiro-Wilk normality test, all p -values reject the null hypothesis. Therefore none of above data is normally distributed. Another test for normality is Kolmogorov-Smirnov test which has a alternative hypothesis of two-sided. It also says that none of the following data set is normally distributed. The results are in Table 4.1.

Let's look the scatterplots of a series versus lagged values of the series of both daily closing prices and its return values up to lag 9.

It is seen that daily closing prices of S&P500 have the large positive correlation at each lag until lag 9. According to plots in Figure 4.7, we can say that our original data comes from an underlying autoregressive model with strong positive autocorrelation.

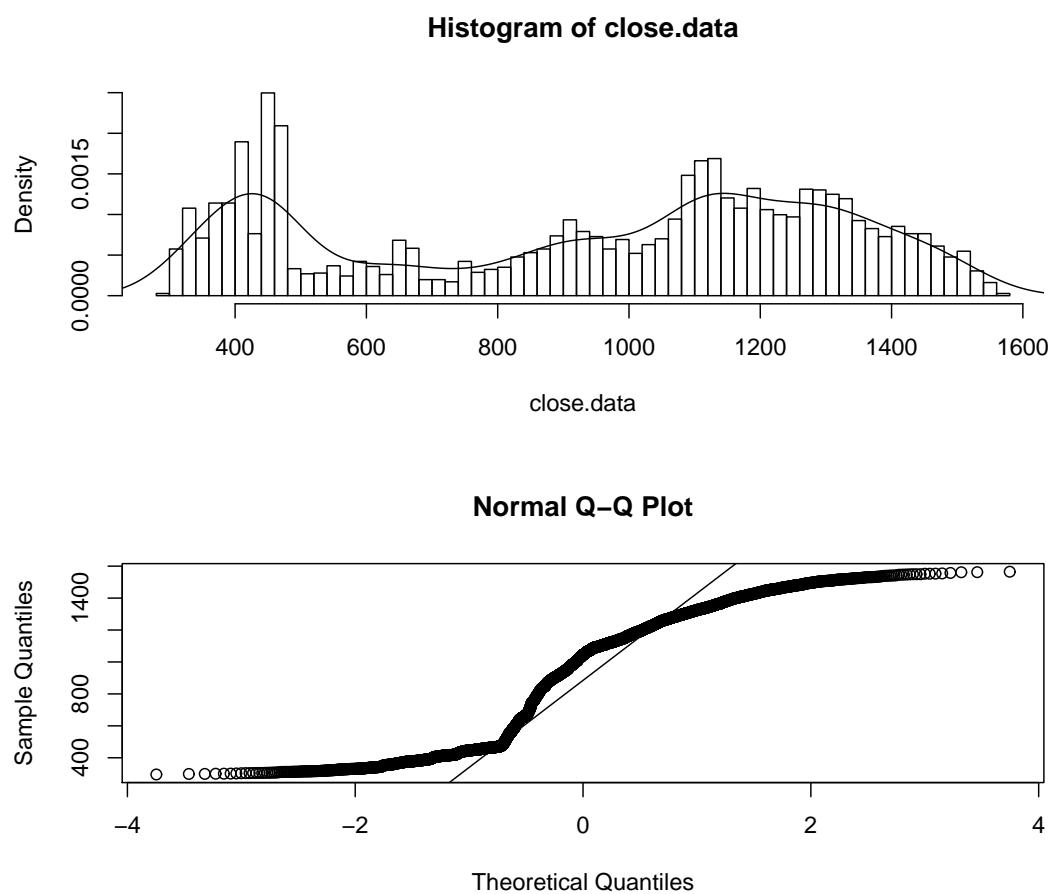


Figure 4.4: Histogram of Daily Closing Prices of S&P500 and Q-Q Plot of Daily Closing Prices of S&P500.

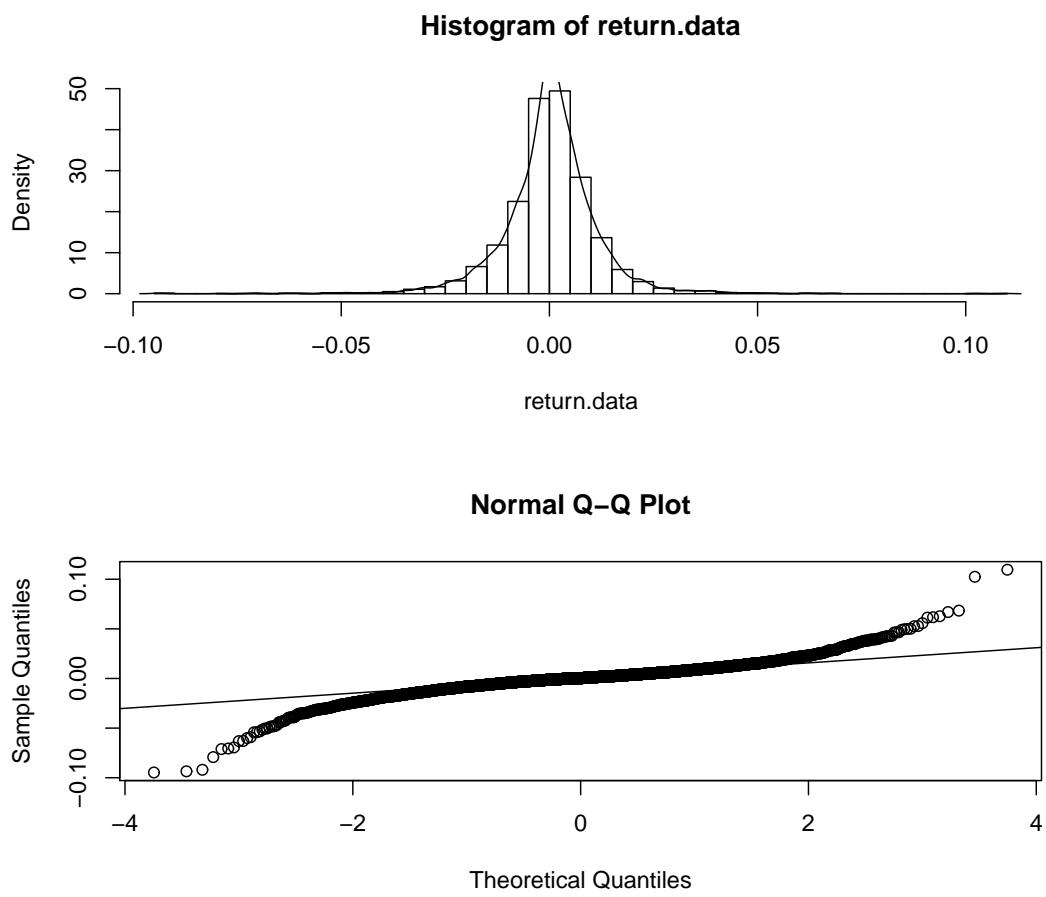


Figure 4.5: Histogram of Return Values of S&P500 and Q-Q Plot of Return Values of S&P500.

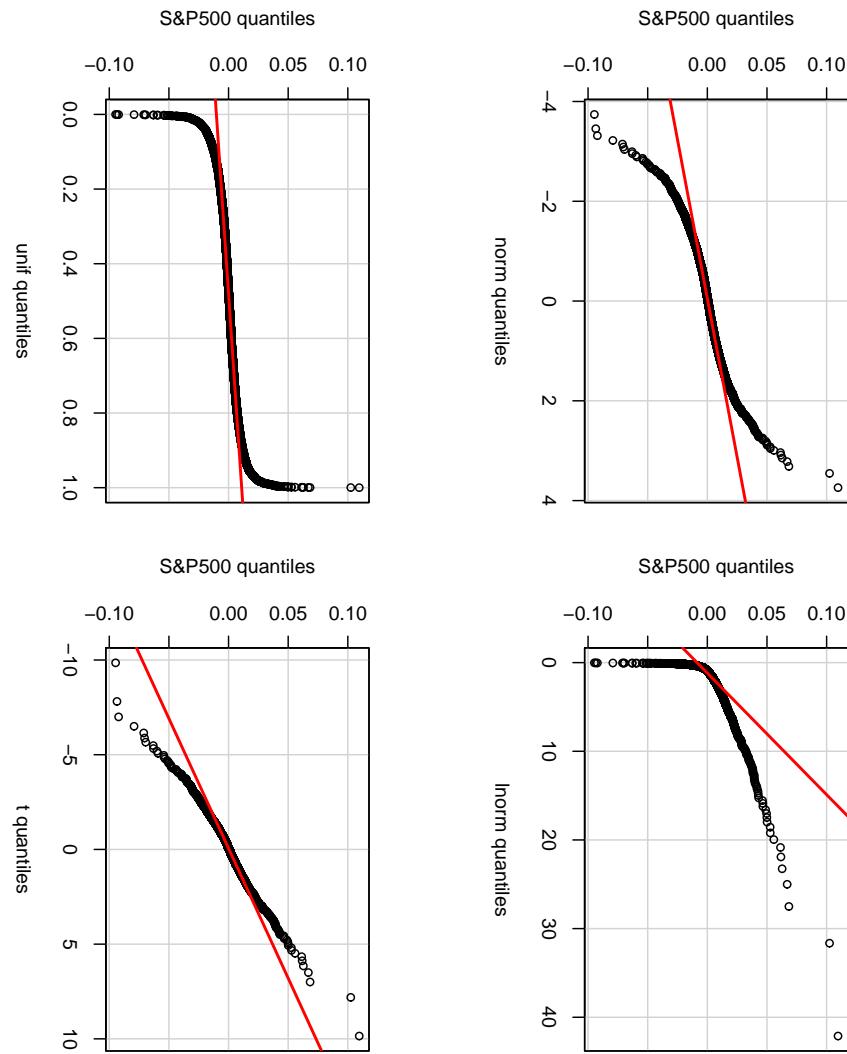


Figure 4.6: Q-Q Plots of Related Distributions for S&P500 Return Data.

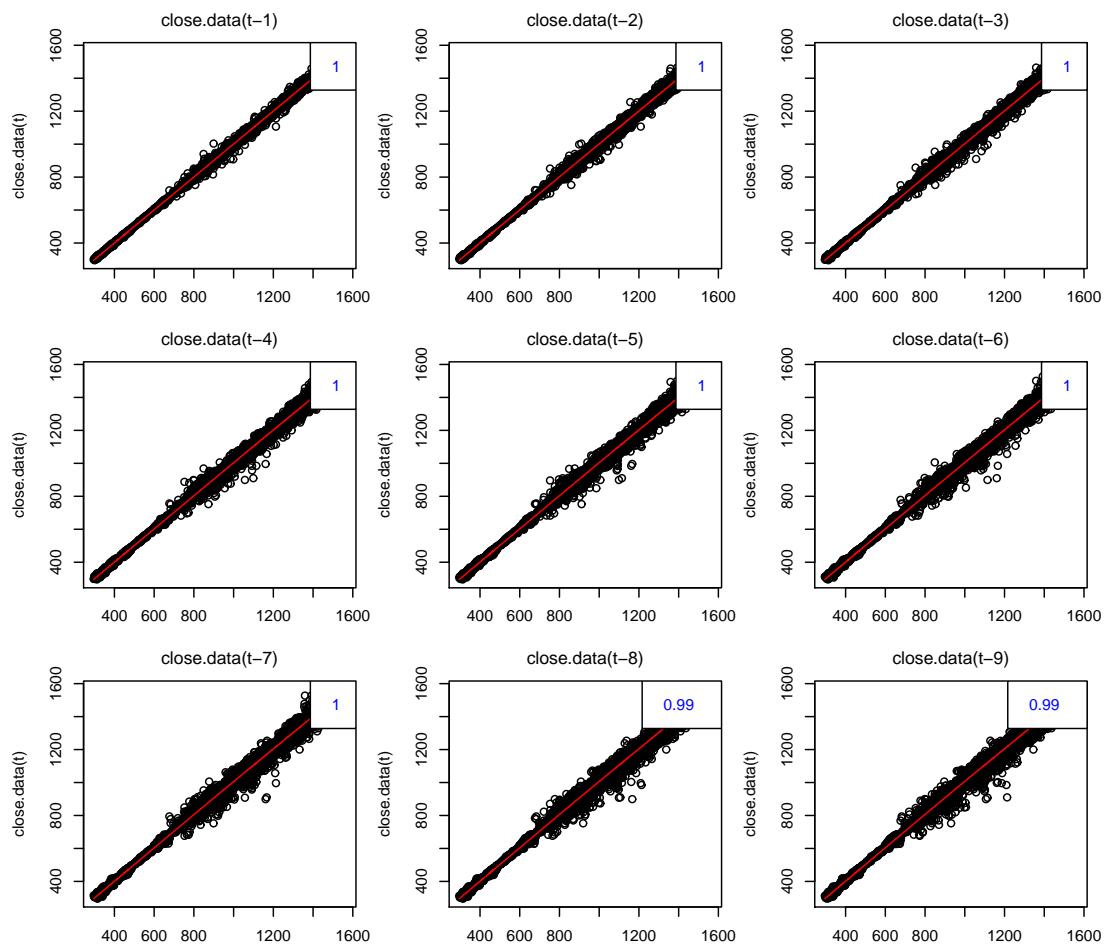


Figure 4.7: Grid of scatterplots of daily closing prices (t -lag) versus daily closing prices for lag = 1, 2, ..., 9.

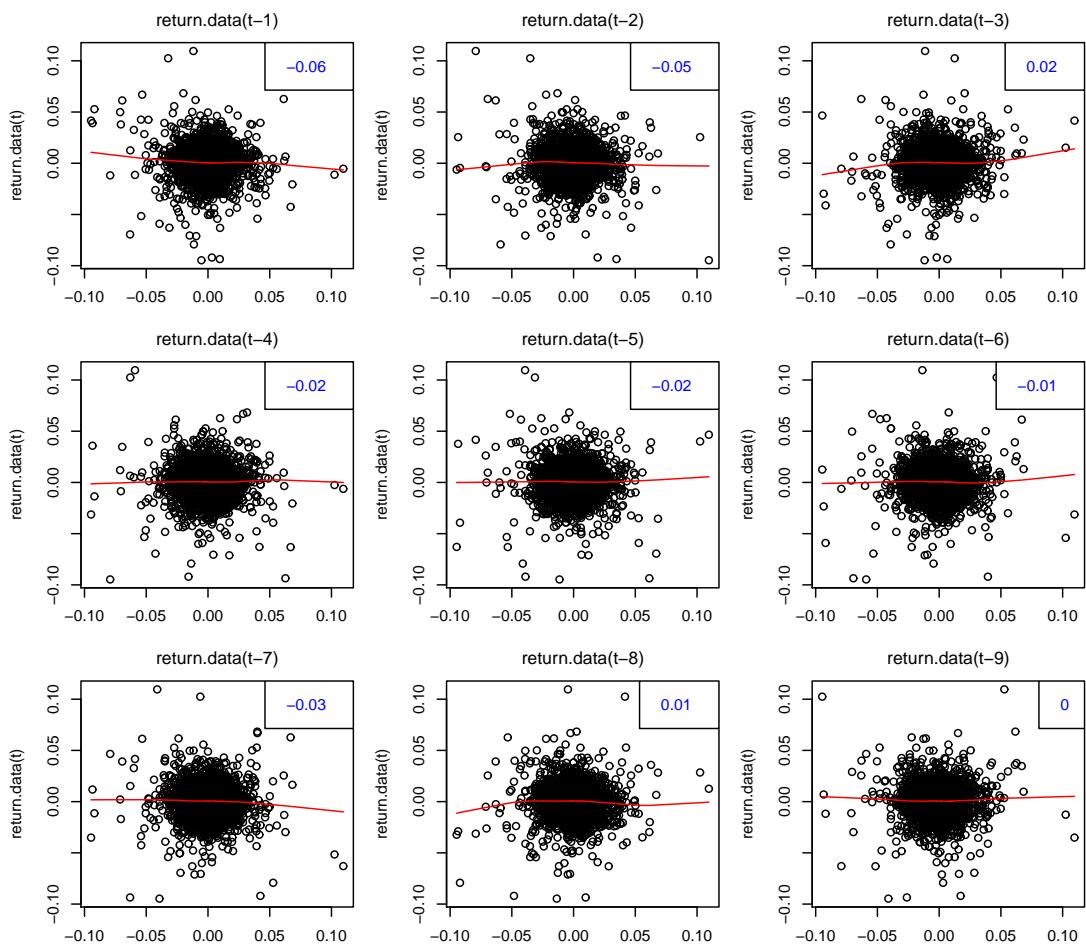


Figure 4.8: Grid of scatterplots of return values (t -lag) versus return values for lag = 1, 2, ..., 9.

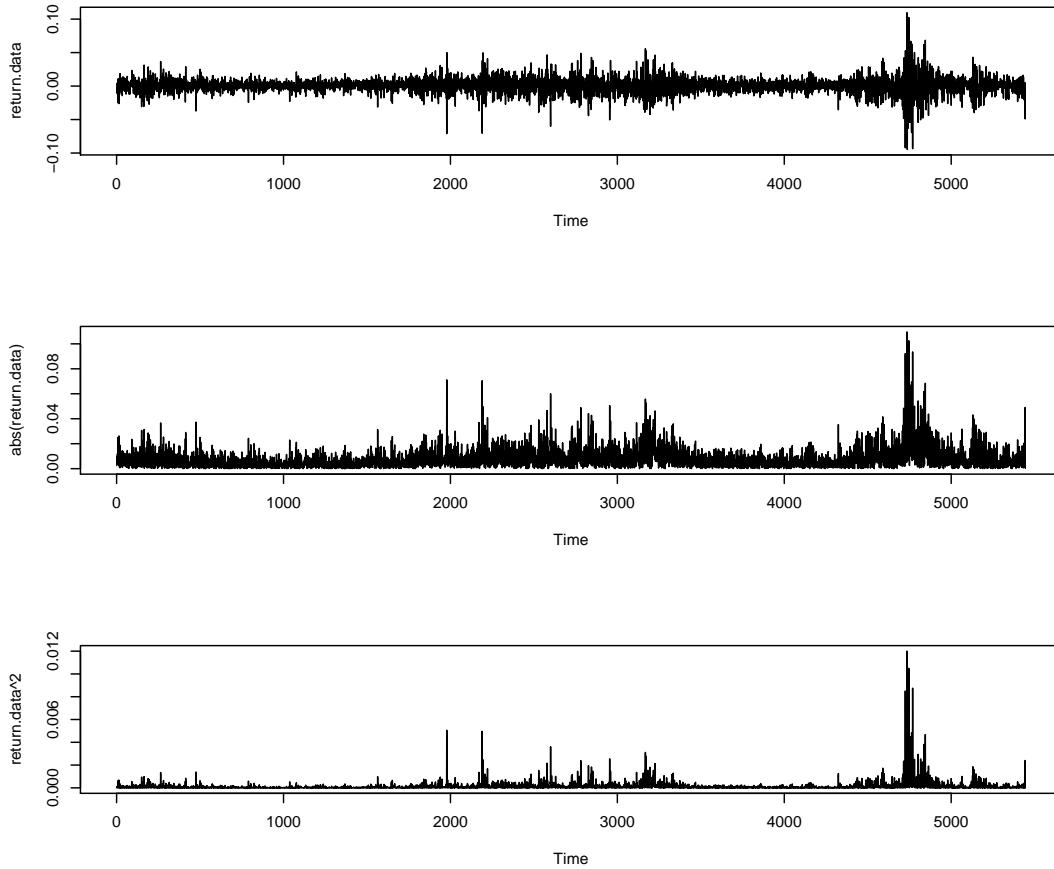


Figure 4.9: S&P500 Return Series and its Absolute and Squared Types.

Process is highly non-random. In other words there is strong relationship between an observation and next observation.

On the other hand, return values have approximately zero correlation at each lag as seen in Figure 4.8. Return values seems more random or we can say that data comes from an underlying autoregressive model with moderate autocorrelation. It has also almost zero autocorrelation value for higher lag values. Hence return data seems to be random. We can consider stock market returns as random walks as Seiler and Rom stated in [60]. Nevertheless, having almost zero autocorrelation does not mean that returns are independent over time. Daily returns of S&P500 show nonlinear time dependence according to changing volatility in time. In other words, although many price of stocks act independently, in crises periods they all tend to fall down [22]. In order to see the nonlinear time dependency in daily return volatility, we will plot the absolute and squared daily returns and their autocorrelation functions.

In Figure 4.10, graph on the left shows ACF of absolute value of S&P500 return values. Graph on the right shows squared of S&P500 return values. It is clear that there is time dependence in the daily absolute and squared returns. Volatility of the daily returns

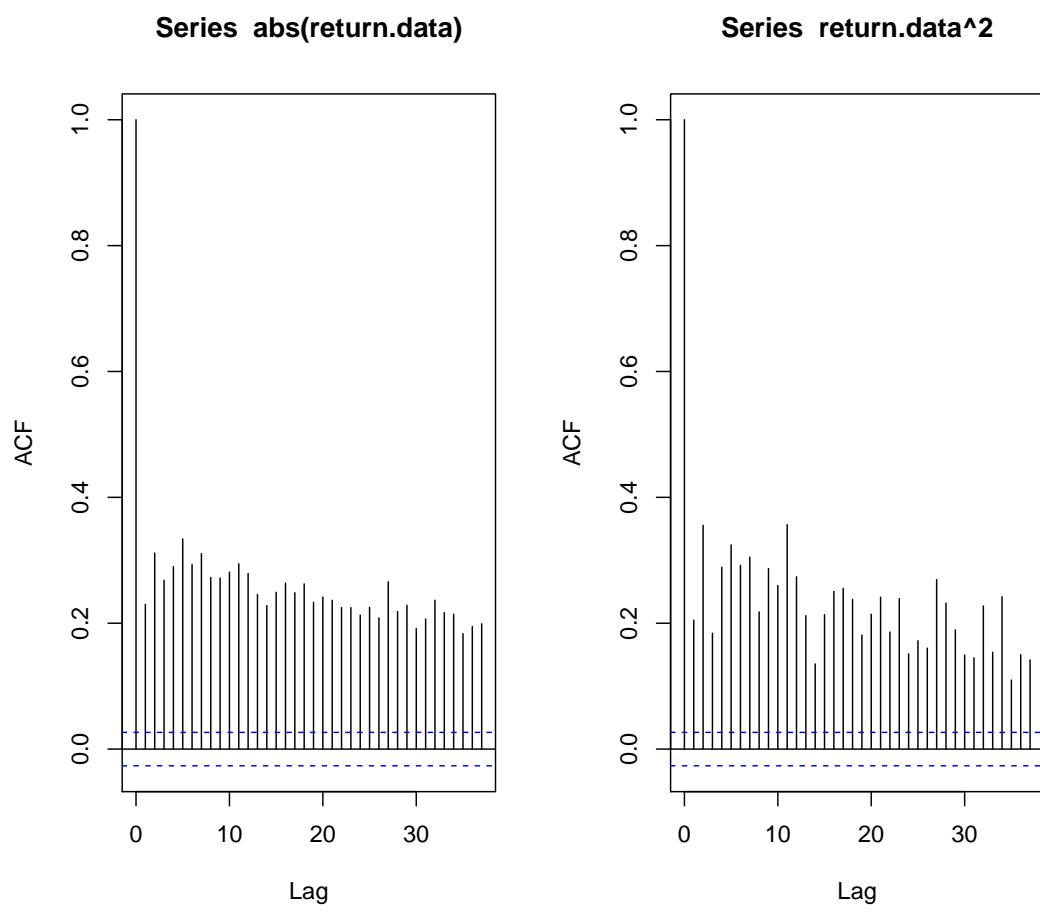


Figure 4.10: ACF of Absolute and Squared Types of S&P500 Return Series.

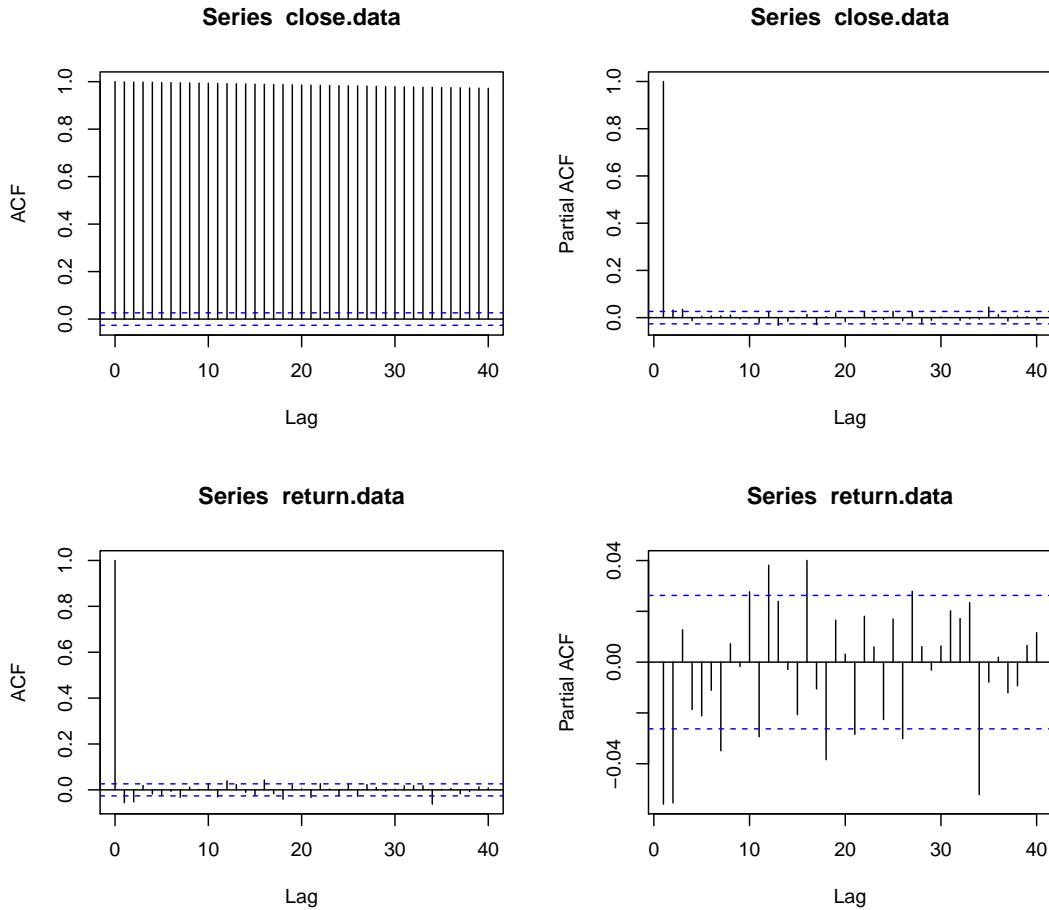


Figure 4.11: ACF and PACF of S&P500 Daily Closing Prices and its Return Values.

is expressed by the daily absolute and squared returns. As a result there is positive time dependence in daily return volatility. It is seen that volatility is autocorrelated and hence it is predictable.

In Figure 4.11 we plot autocorrelation functions and partial autocorrelation functions of closing prices and return values up to lag 40. The graphs on top shows ACF and PACF of daily closing price series and graphs on bottom shows ACF and PACF of return value series.

ACF of closing prices data shows that original data is completely non-stationary. Almost all ACF values are close to 1 and they exceed the confidence interval. The data comes from an underlying autoregressive model with strong positive autocorrelation. On the other hand ACF of return values exceed the confidence interval at lags 0, 1, 2, 7, 10, 11, 12, 16, 18, 21, 22, 26 and 34 with non important size.

ACF of differenced return values data exceeds confidence interval at lags 0, 1, 2, 3, 7, 8, 10, 11, 12, 15, 16, 18, 19, 21, 22, 24, 25, 26, 27, 33, 34 and 35 with non important size. First difference exceeds at 13 points while second difference exceeds 22 points

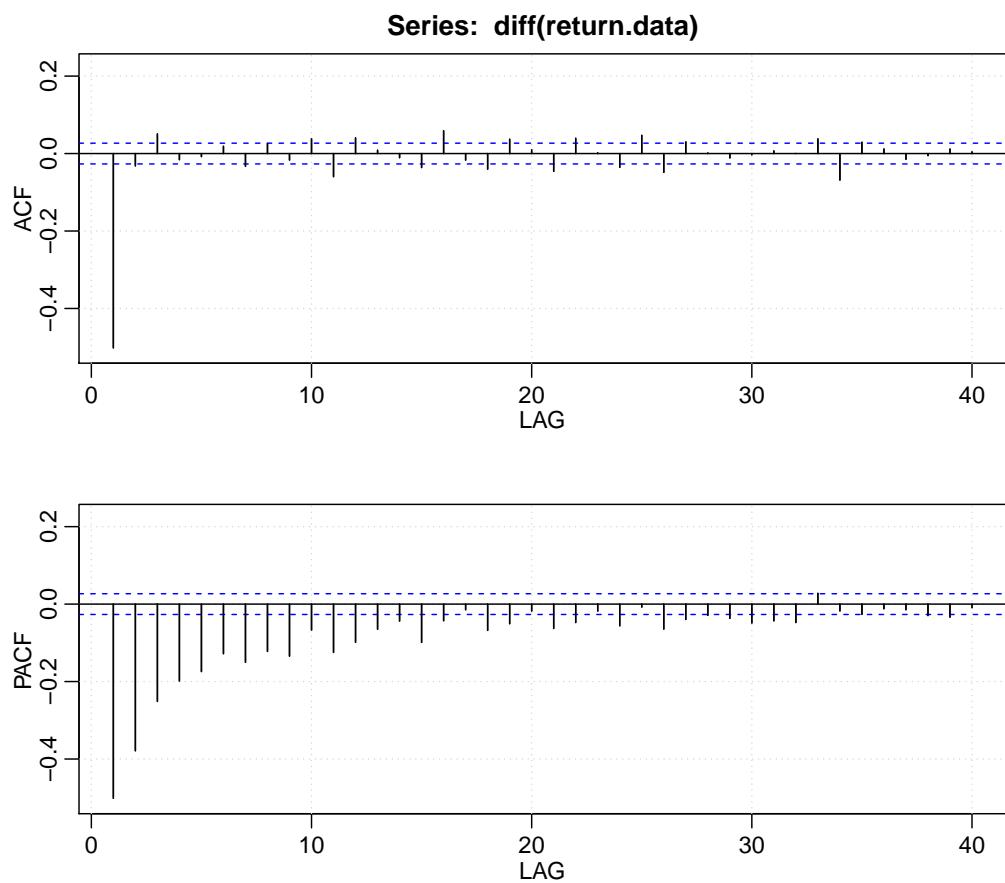


Figure 4.12: ACF and PACF of Differenced S&P500 Return Values.

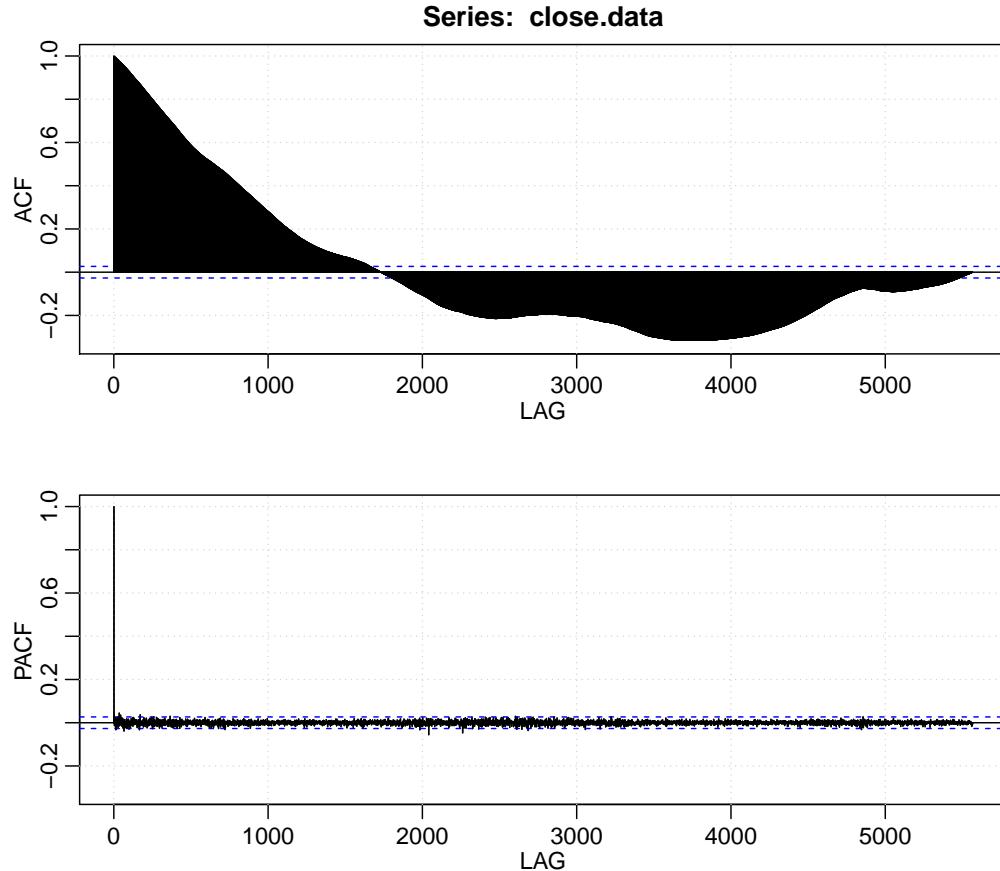


Figure 4.13: ACF and PACF of S&P500 Daily Closing Prices with Frequency 1.

between lag 0 and lag 40.

Actually lag axes are in terms of frequency. If we consider the frequency as 260, then each lag corresponds to lags that 260 times itself. In our plot, frequency is 1 and if we change the frequency as 260 than maximum lag can be found from number of observation divided by the frequency. Following plots in Figure 4.13, Figure 4.14, Figure 4.15 and Figure 4.16 explain frequency concepts more clearly.

In Table 4.2, there are some descriptive statistics of data sets that belong to S&P500 daily closing prices. Return data is the same as the first difference of the logarithmic closing data set as we see in the following table.

Kurtosis of return values is higher than the value of normal distribution with kurtosis=3 which means that the financial time series data has the fat-tail characteristic [38]. The skewness of both daily closing price data and return data are not zero, so that both are not symmetric.

Some Other Statistical Tests

Here we apply some tests to daily closing prices and its returns with both frequency 1

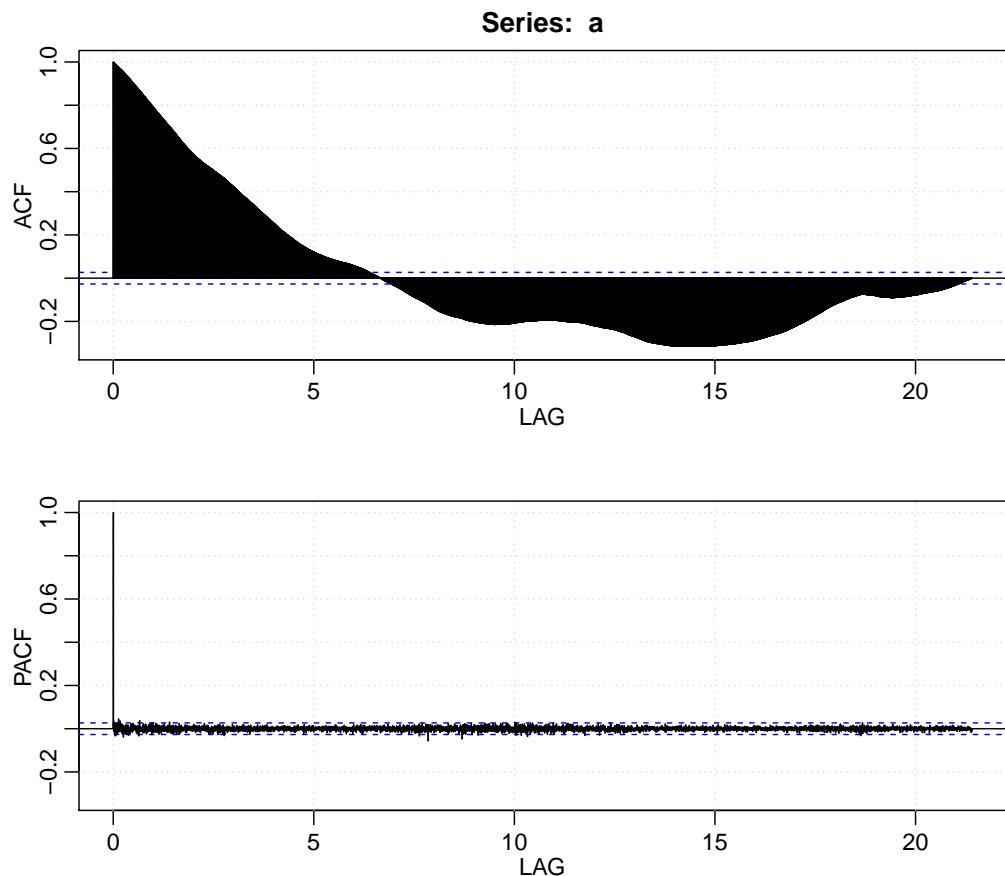


Figure 4.14: ACF and PACF of S&P500 Daily Closing Prices with Frequency 260.

Table 4.2: Descriptive Statistics of Data Sets Related to S&P500 Daily Closing Prices

Data	Mean	Standard Deviation	Skewness	Kurtosis
close.data	940.5007954	371.7583052	-0.2997497	1.6993896
return.data	0.0002213655	0.0116499799	-0.2137863793	11.8615840860
log(close.data)	6.7464941	0.4766666	-0.6841409	1.9951488
diff(close.data)	0.1543965	12.0781645	-0.3036438	10.8152632
diff(log(close.data))	0.0002213655	0.0116510502	-0.2137667433	11.8594052498
diff(diff(close.data))	-0.01090408	17.60439277	0.46129925	10.95511959
diff(return.data)	5.163097e-07	1.691974e-02	5.512029e-01	1.226257e+01

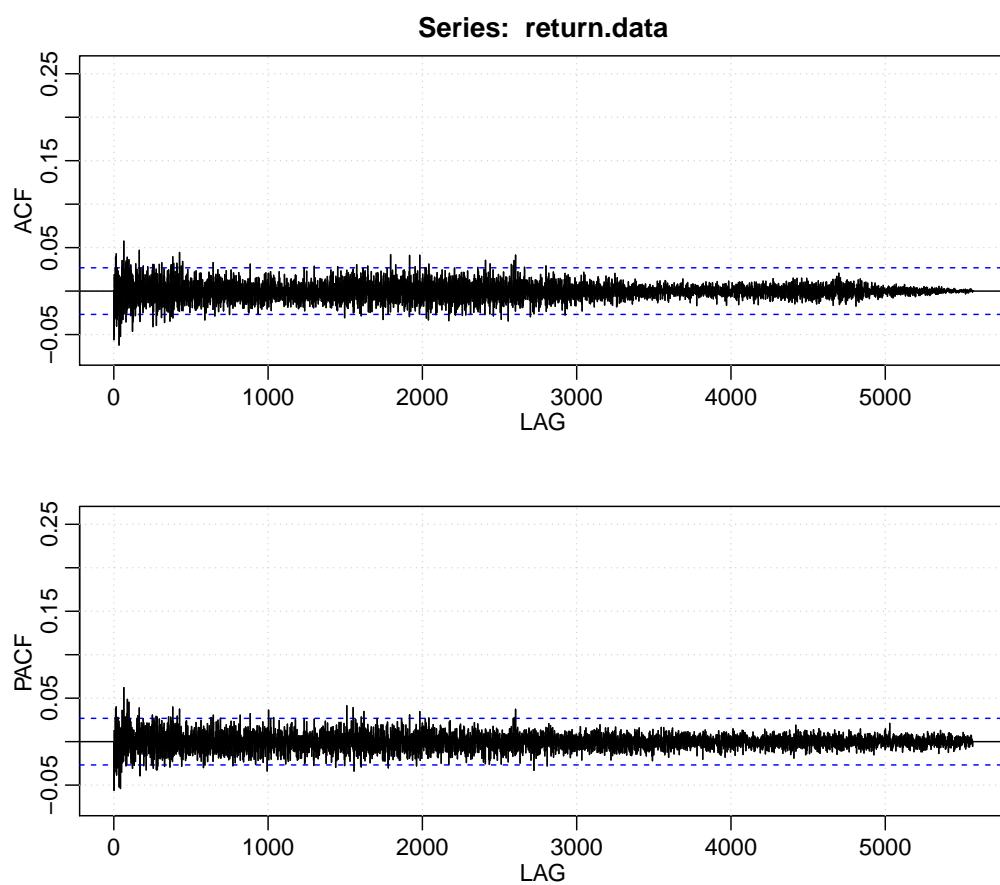


Figure 4.15: ACF and PACF of S&P500 Return Values with Frequency 1.

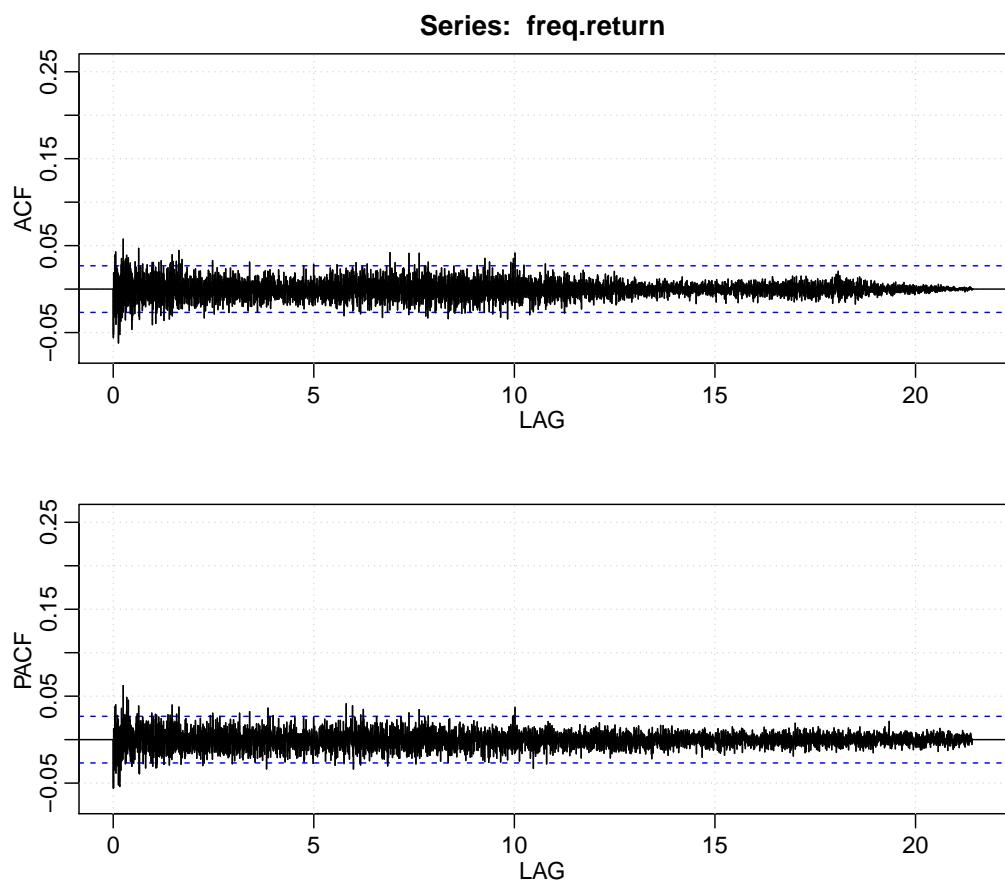


Figure 4.16: ACF and PACF of S&P500 Return Values with Frequency 260.

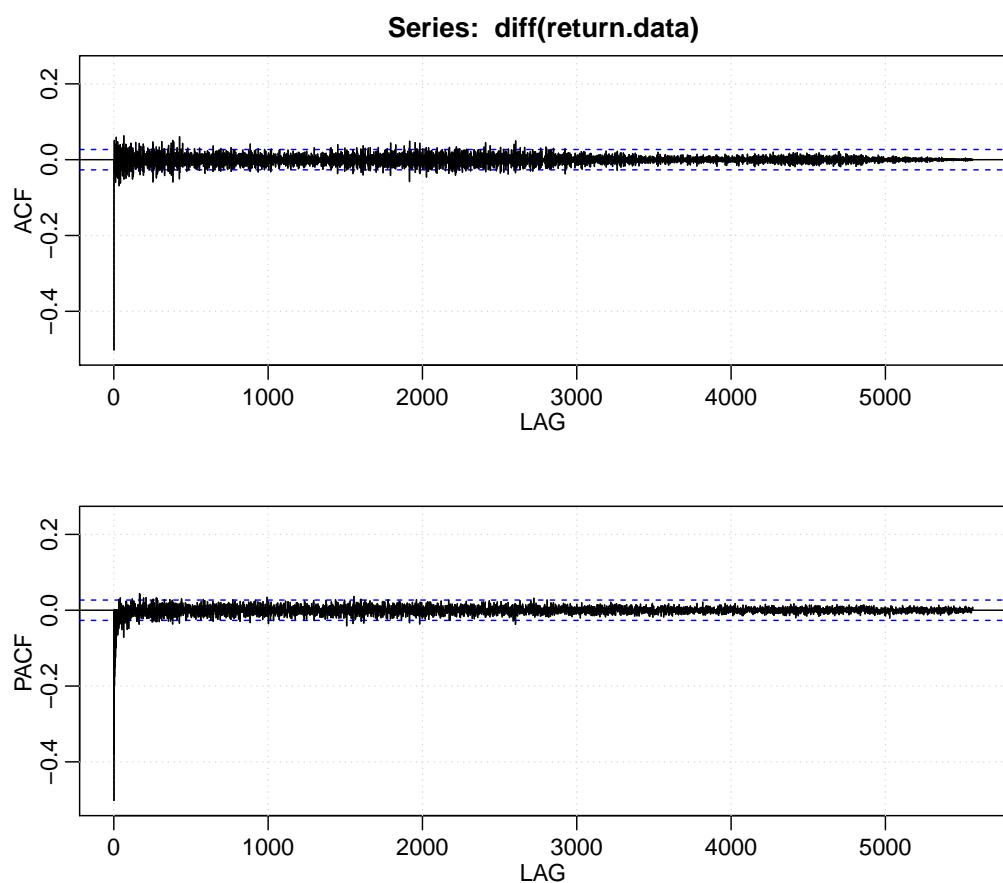


Figure 4.17: ACF and PACF of Differenced S&P500 Return Values with Frequency 1.

Table 4.3: Augmented Dickey-Fuller Test for Stationary and KPSS Test for Trend Stationary

	Augmented Dickey-Fuller Test			KPSS Test		
	Dickey-Fuller	Lag order	p-value	KPSS Trend	Truncation lag parameter	p-value
close.data	-1.6186	17	0.7399	3.9387	17	0.01
return.data	-17.8034	17	0.01	0.0709	17	0.1

and frequency 260. Results of frequency 260 are same with the same series with frequency 1. Tests are applied to see if series are stationary and if residuals are correlated. Firstly we handle with the Augmented Dickey-Fuller test for being stationary or not. Results are given in Table 4.3 for closing prices and return values.

Dickey-Fuller test says that we cannot reject H_0 for daily closing prices data. It means that these series are non-stationary and there exists unit root. On the other hand test says that we can reject H_0 for p-value which is 0.01 and so there is no unit root, i.e. return series is stationary.

We can also use KPSS test to consider that time series is stationary or not where results are given in Table 4.3. According to KPSS test we cannot reject H_0 for daily closing prices data set, i.e. series is non-stationary. On the other hand we can reject the null hypothesis for return series, i.e. return series is stationary.

Data Characteristic

According to daily closing prices of S&P500, it is seen that prices look like random walk non-stationary and return values look like mostly covariance stationary. Returns are nearly uncorrelated over time and there is little evidence of linear time dependence. Distributions of returns have much fatter tails than the normal distribution (excess kurtosis). It means that returns are not normally distributed. Moreover volatility seems to be autocorrelated.

Since daily closing prices of S&P500 data looks like random walk and it is non-stationary, it is better to look the first difference of logarithmic version of the original data to see if there is a predictable pattern.

Decomposition

In general time series is gathering observations from repeated measurements over time and it consists of three basic components that are given as trend, seasonality and irregularity (residuals). Trend shows long term direction, seasonality shows systematic and calendar related effects and irregularity shows unsystematic, short term fluctuations.

First of all let's look the linear filtering applied to daily closing prices of S&P500 for $a = 2, 12, 65$ in Figure 4.18. Coefficients for filtering are found by $\frac{1}{2a+1}$.

We cannot use the data with frequency 1, since it does not include periodicity with construction of frequency 1. Decomposition of data requires at least 2 periodic parts. Therefore we use basic decomposition on the data with frequency 260. In Figure 4.19, decomposition is done by using *loess* (Local Polynomial Regression Fitting) method.

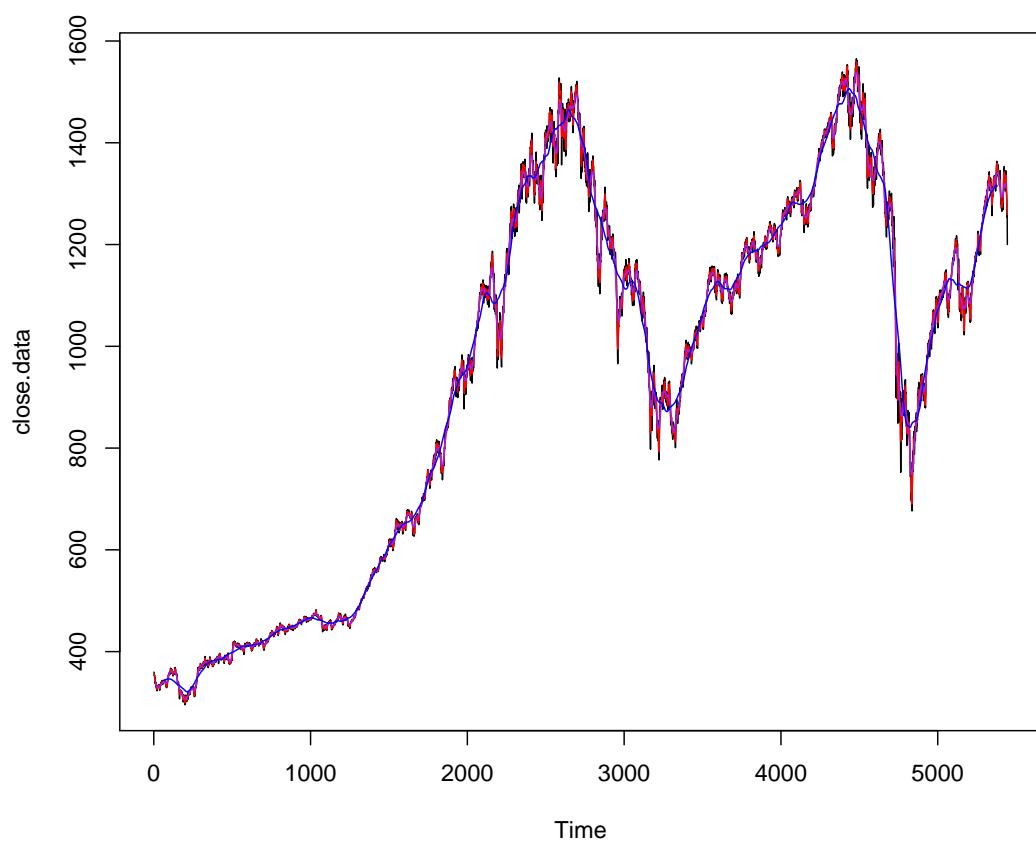


Figure 4.18: Linear Filtering of Daily Closing Prices with $a = 2, 5$ and 65 .

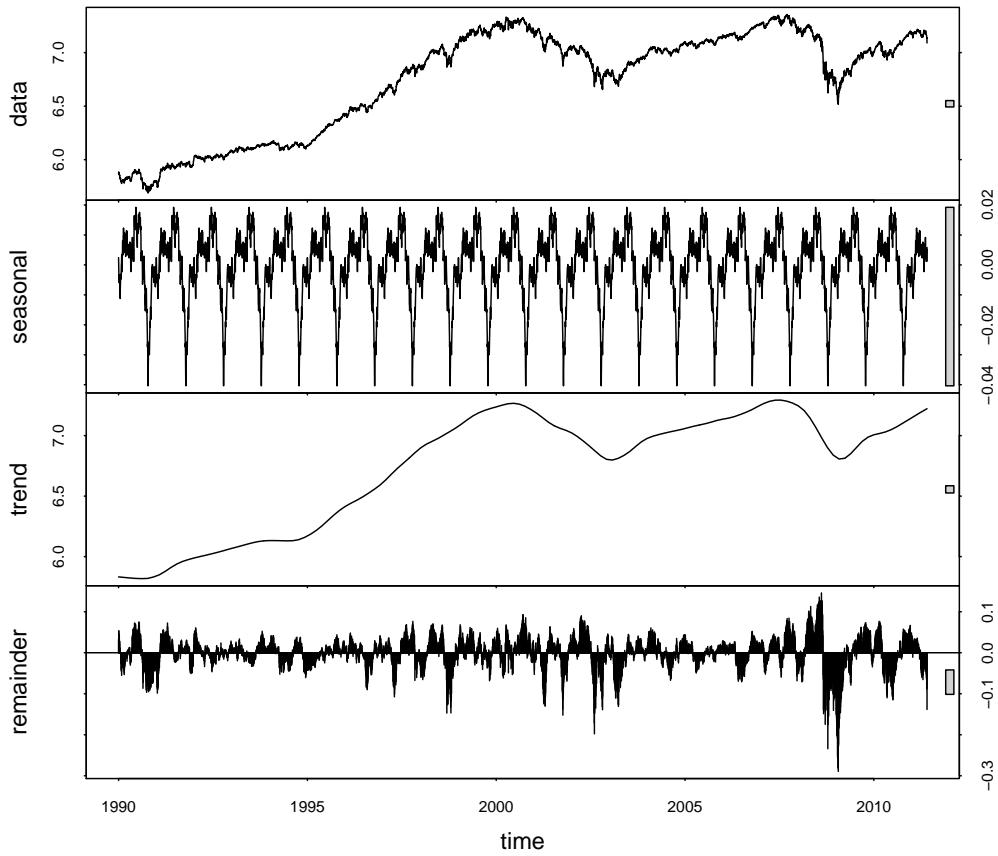


Figure 4.19: Decomposition of Logarithmic Closing Prices of S&P500 with Frequency 260 by Using *Loess* Method.

In other words it is linear regression ‘*plus*’ k-nearest neighbors. Method first derives the seasonal part from the difference of the original signal and its trend. It divides the data into three parts as seasonal, trend and remainder part. Sum of these three parts gives the original data, i.e. type is additive.

Moreover one can use classical seasonal decomposition by moving averages. Contrary to *loess* method, this method firstly derives the trend part by moving averages. In Figure 4.20, plot of decomposition of this method is seen.

We can reach values that are cleaned by trend component as given in Figure 4.21.

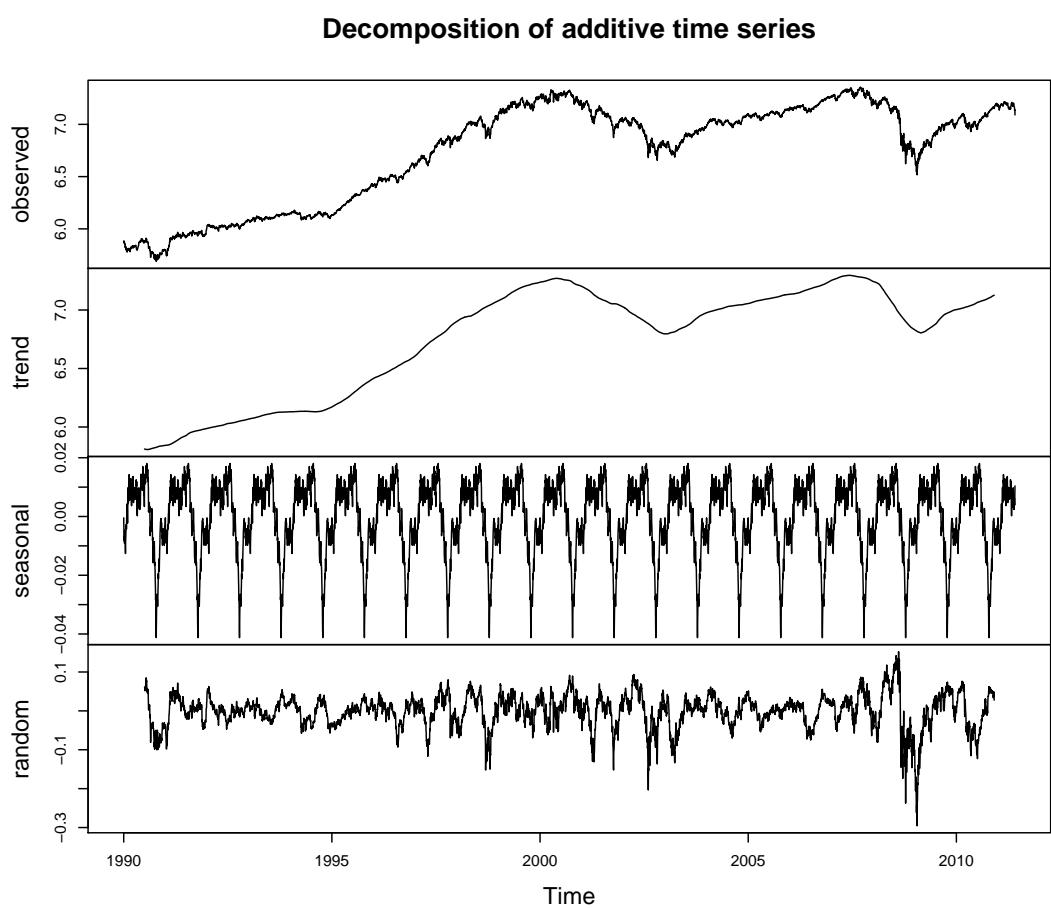


Figure 4.20: Decomposition of Logarithmic Closing Prices of S&P500 with Frequency 260 by Using MA Method.

signal without trend component

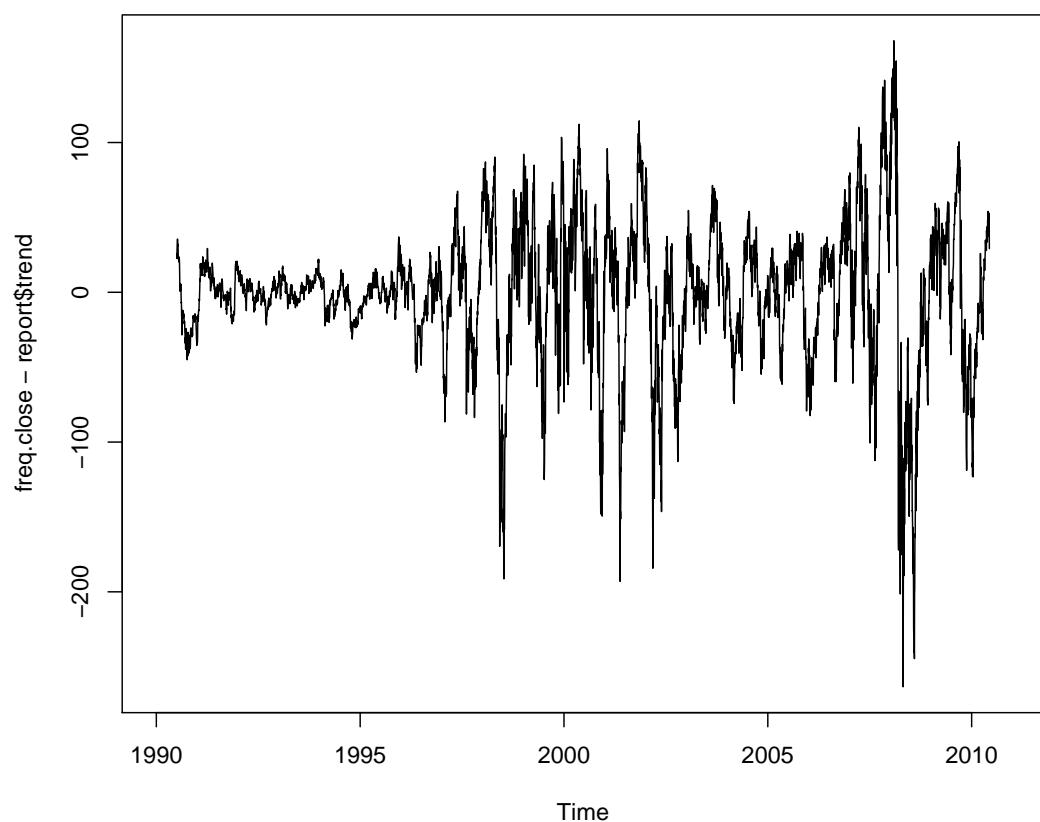


Figure 4.21: S&P500 Daily Closing Prices without Trend Component.

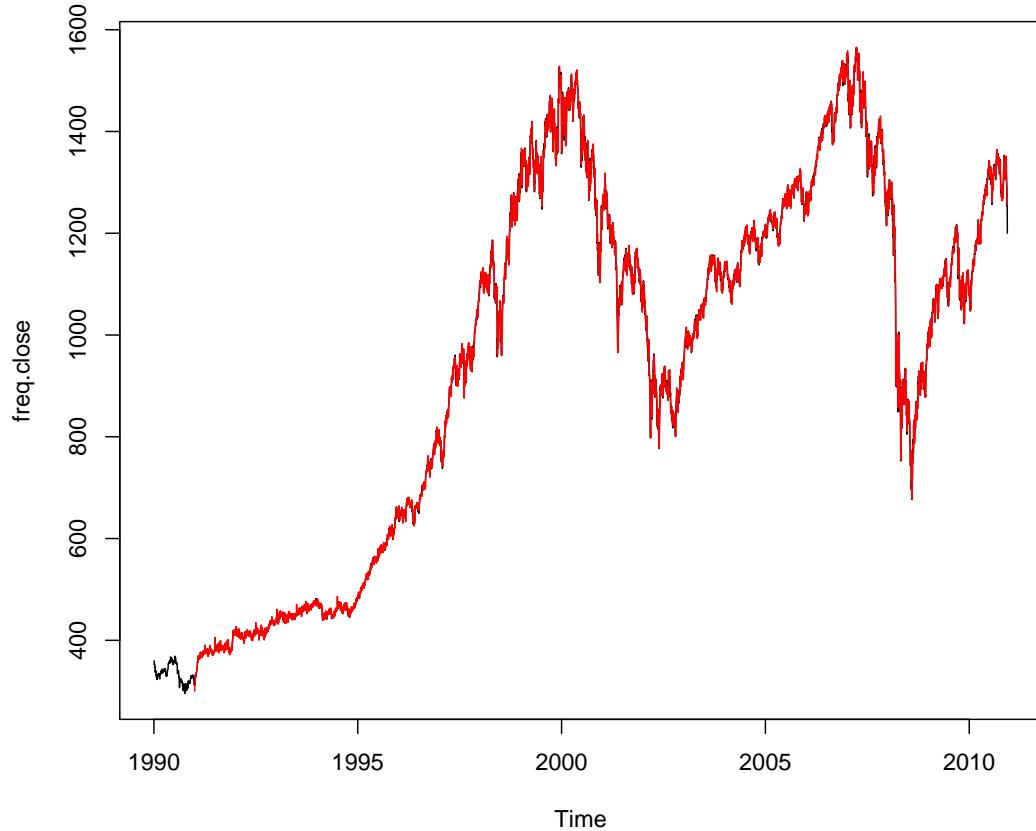


Figure 4.22: S&P500 Data with Holt-Winters Smoothing.

4.2 Exponential Smoothing

Exponential smoothing is done to estimate the next values of the time series with no trend or seasonal components. Basically it is written as,

$$\hat{x} = \lambda_1 x_{t-1} + \lambda_2 x_{t-2} + \dots ,$$

where $\lambda_i = \alpha(1 - \alpha)^{i-1}$ for $0 < \alpha < 1$. Extended version of this method is Holt-Winters smoothing which accept the time series with both trend and seasonal components. The S&P500 daily closing prices and its smoothed values are given in Figure 4.22. Then the next 200 predicted observations are given in Figure 4.23 by using Holt-Winters.

Forecast with 75% and 95% confidence intervals are given in Figure 4.24, again using Holt-Winters method.

Another forecasting method is short term load forecasting with 75% and 95% confidence intervals. It uses in here exponential smoothing state space model which is given by Hyndman et al. in [37] and in [34]. The characters, which are given in the

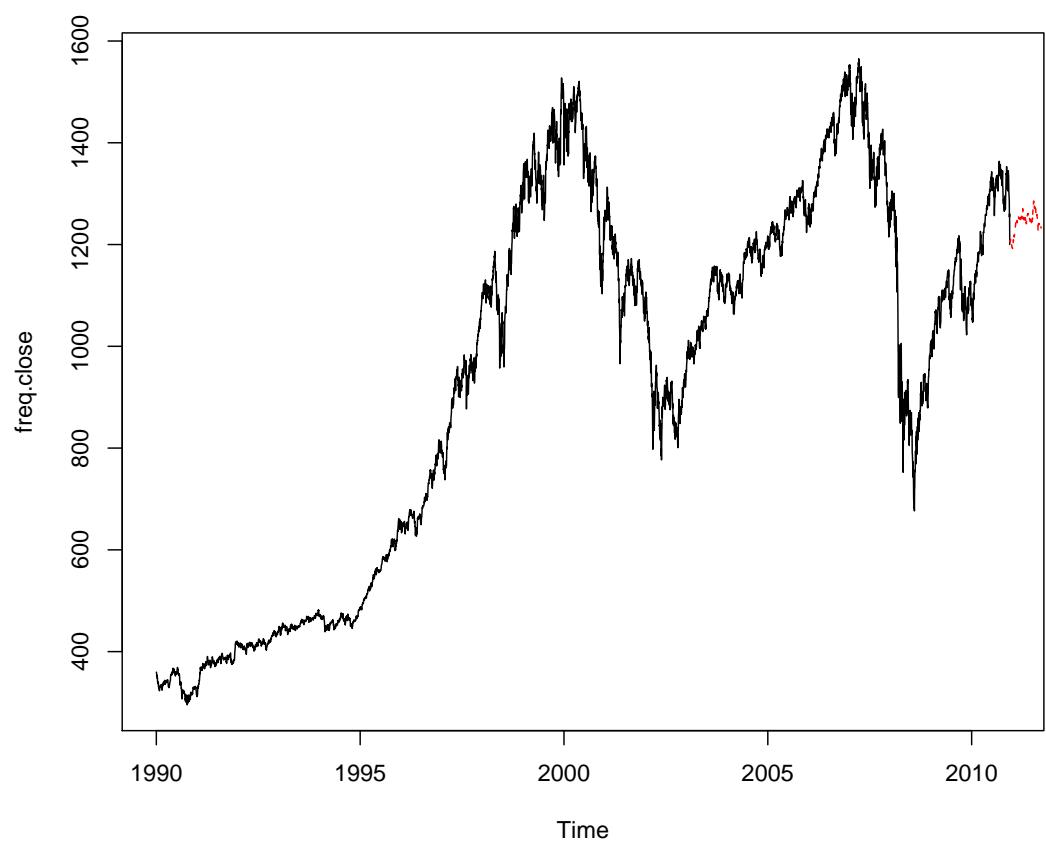


Figure 4.23: Predicted 200 Observations by Using Holt-Winters Method.

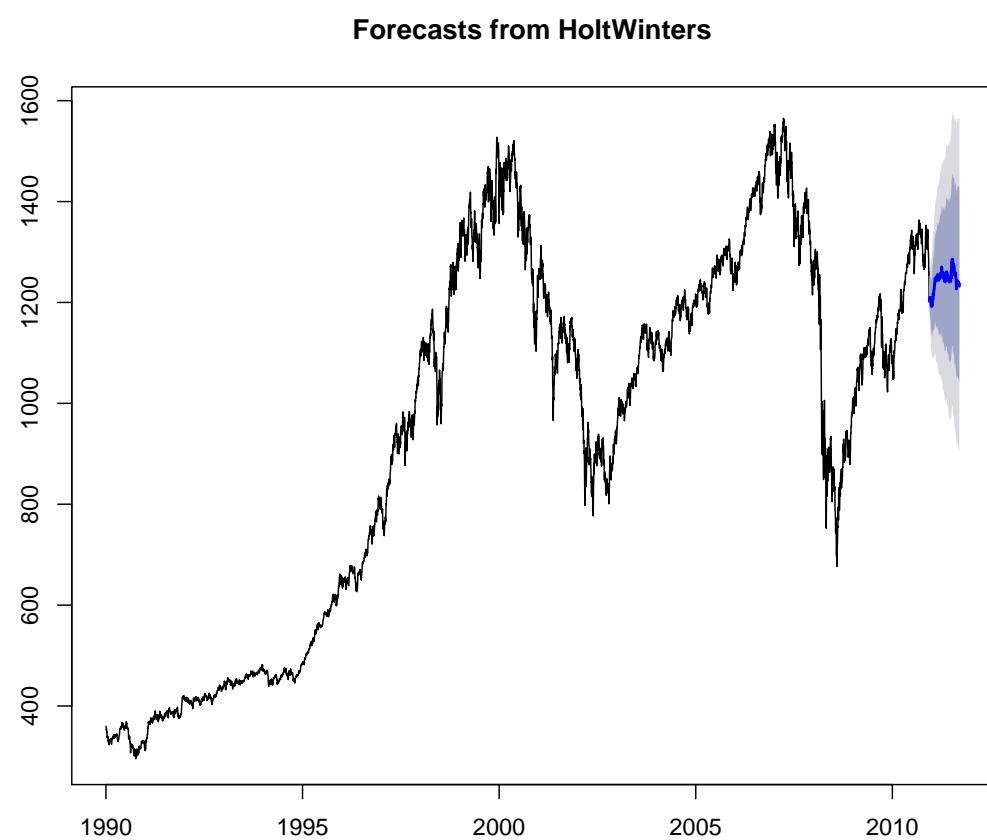


Figure 4.24: Predicted 200 Observations by Using Holt-Winters with 75% and 95% Confidence Intervals.

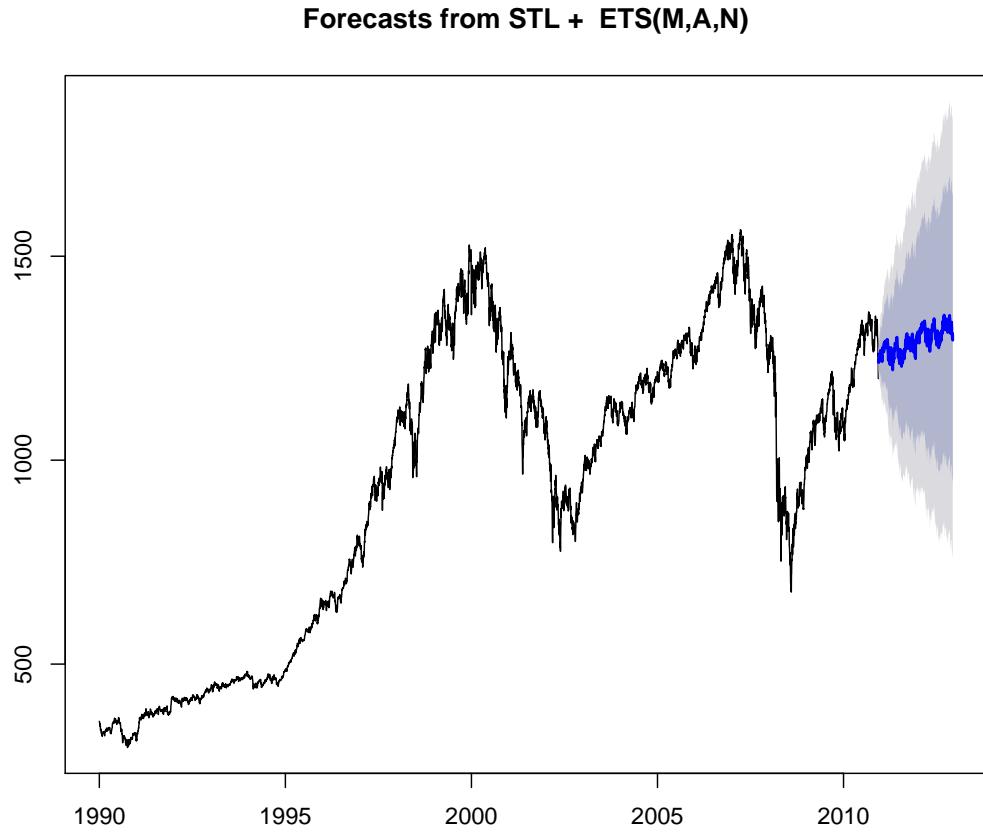


Figure 4.25: Forecasting of S&P500 Data by STLF Method.

ETS brackets, are the error type, the trend type and the season type, respectively. The trend type and the season type can be “N”=none, “A”=additive, “M”=multiplicative and “Z”=automatically selected. The error type can be A, M or Z. Therefore it is seen in the figure that our method goes with the model of $\text{ETS}(M, A, N)$ which means that it is simple exponential smoothing with multiplicative errors as given in Figure 4.25.

4.3 Frequency Domain Analysis

4.3.1 Fourier Transform Analysis

S&P500 data is very complex data as we regarded from the some tests and the properties of the data. It includes different type of frequency components in it and Fourier analysis is used to look the time series in the frequency domain. In Figure 4.26, original data and some related periodograms are given. The figure on the right top is the raw periodogram without smoothness. The figure on the left bottom is $\text{kernel}(10, 20)$ smoothed version of the raw periodogram. It is better than the original one but still

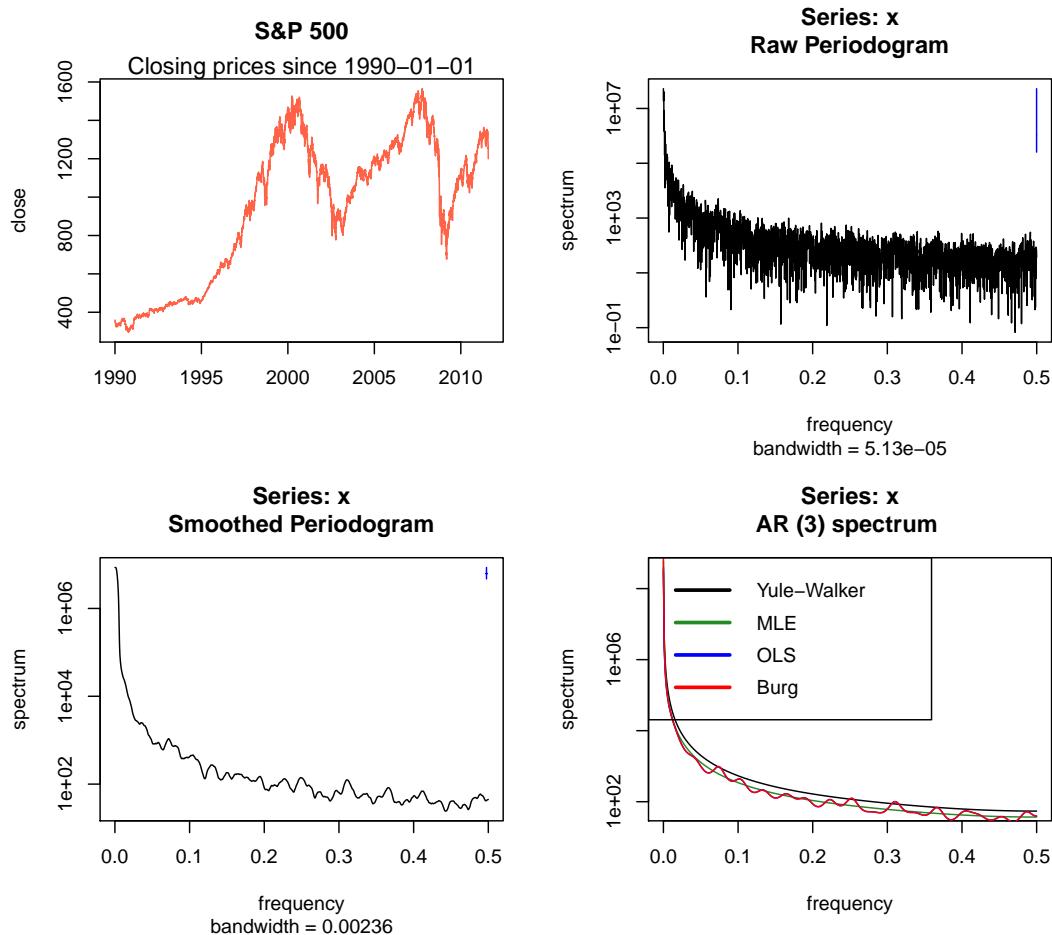


Figure 4.26: S&P500 Daily Closing Prices Data and Different Type of Periodograms.

there is nothing to analyze from the figure. The figure on the right bottom is AR fitted spectrum with four methods. As we mentioned in the DAX example, again MLE and Yule-Walker methods are similar in each other. Only difference is that MLE method has lower spectrum for the same frequencies. OLS method and Burg method are better than Yule-Walker and MLE, since one can choose clear picks from their results. Peak results are disappearing for Yule-Walker and MLE methods.

If we consider the periodogram which is AR fitted by Burg method gives the following peak numbers of frequencies. If one look at the number of the frequency and divide 1 by that frequency, the period or cycle related to that frequency can be found. Bold numbers of frequencies are selected and considered in the calculations. *Peaks: **75**, **103**, **135**, **166**, **190**, **225**, **254**, **312**, **368**, **407**, **454**, **488**.*

Bold numbers are chosen important frequencies and they are given as follows, respectively related to peak number; 0.074148297, 0.102204409, 0.134268537, 0.165330661, 0.224448898, 0.253507014, 0.311623246, 0.367735471 and 0.487975952. Some of these frequencies are also seen in Daniell smoothed periodograms in Figure 4.26 and in Figure 4.27. In Figure 4.28, chosen peak values are signed by dotted lines.

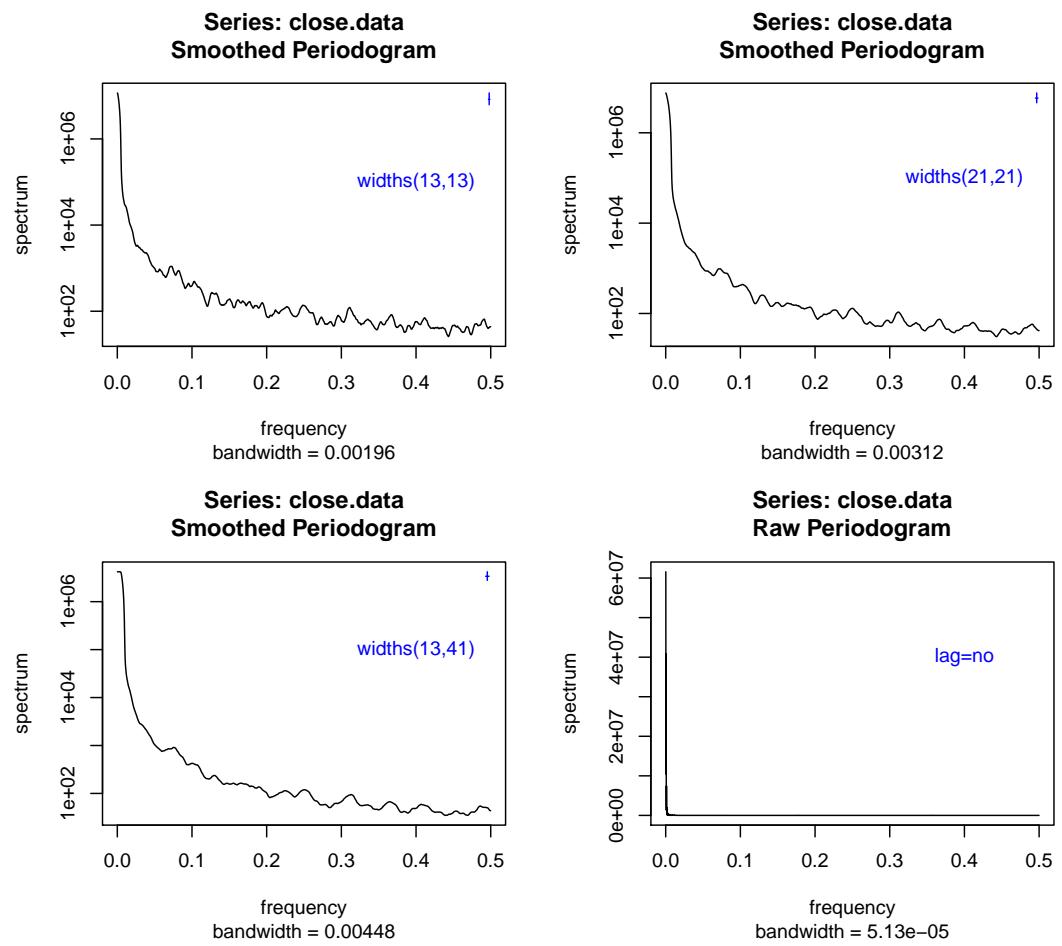


Figure 4.27: Periodograms of S&P500 Data with Given Smoothness Parameters.

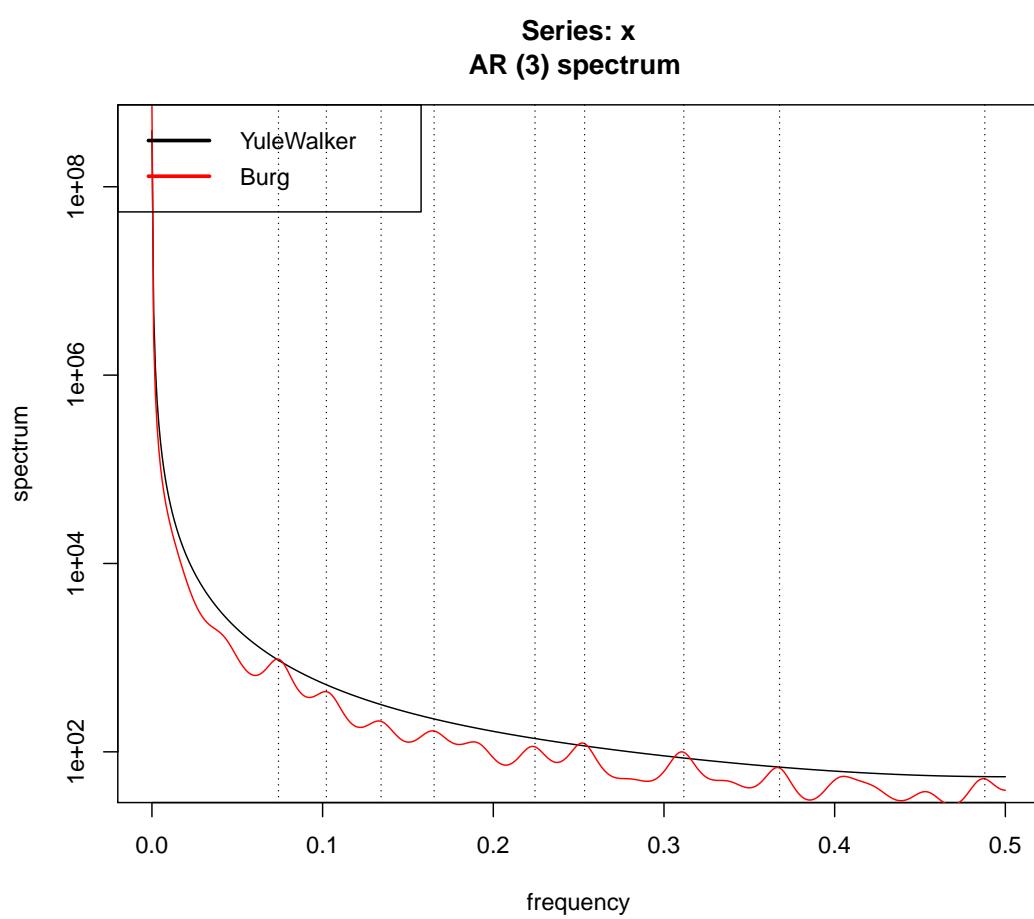


Figure 4.28: AR(3) Fitted Spectrum of S&P500 Data with Yule-Walker and Burg Methods.

If we divide 1 by the chosen frequencies, we will get periodicities in days as 13.4865, 9.7843, 7.4478, 6.0485, 4.4554, 3.9447, 3.2090, 2.7194 and 2.0493 respectively. However we cannot reach the long term periodicities with this method.

Spectrum graphs tell us that higher frequency parts have control over the spectrum. Other frequency components are almost lost under higher frequency part. That's why non-stationary time series like financial time series are not appropriate to be analyzed by Fourier analysis. Therefore trend component should be cleared from the time series. In R program, function '*detrend*' de-trends the time series by using a linear trend model by default option.

Not doing smoothing or doing too much smoothing result in non-visible frequency components in time series. On the other hand if one thinks that the smoothness is in moderation, then many frequency components can be missed as we see in the periodogram results. In addition, it is not clear to understand which frequency component belongs to which time slice which is important drawback of Fourier transform.

4.3.2 Wavelet Analysis

For practical issues we need to decide which wavelet filter and wavelet transform will be used. Time series are discrete data sets and so discrete transforms are used rather than continuous transforms. As we mentioned in Subsection 3.2.2, the CWT has some drawbacks according to the DWT like requiring more computations. Padding or removing the time series is needed for the DWT due to requiring dyadic length. The MODWT can be used instead of the DWT not to be forced to use dyadic length. Discrete transforms are discussed in this chapter with using R program to analyze the financial time series.

Length of the filter is another issue which one needs to consider. For instance, as the length of the filter is getting larger, it can catch better typical properties in signal. However it causes some results as Vuorenmaa in [71] mentioned such ad increasing number of coefficients causes more computation and results in increasing number of coefficients that are affected by the boundary condition. Vuorenmaa in [71] suggests the appropriate filter for the DWT as LA(8) in practice. According to Masset in [46], many researchers in finance choose Daubechies(D) or Least-Asymmetric(LA) filters of length 4 to 8 in wavelet analysis. Setz in [61] used Morlet wavelet, which has non-orthogonal and complex function, for wavelet transformations to analyze the financial data.

In each DWT and MODWT figures, level- j wavelet coefficients captures the differences of 2^{j-1} day averages of the time series. It means that, for example for the first level of the decomposition we see day-by-day fluctuations and it goes on like this.

As we mentioned in Section 3.2, DWT requires time series with dyadic length. Otherwise one has to reflect the original time series in order to use the DWT. In Figure 4.29, only wavelet scales of daily closing prices data are plotted up to scale 5 and in Figure 4.30 wavelet scales of return data are plotted up to scale 6 with smooth scale at

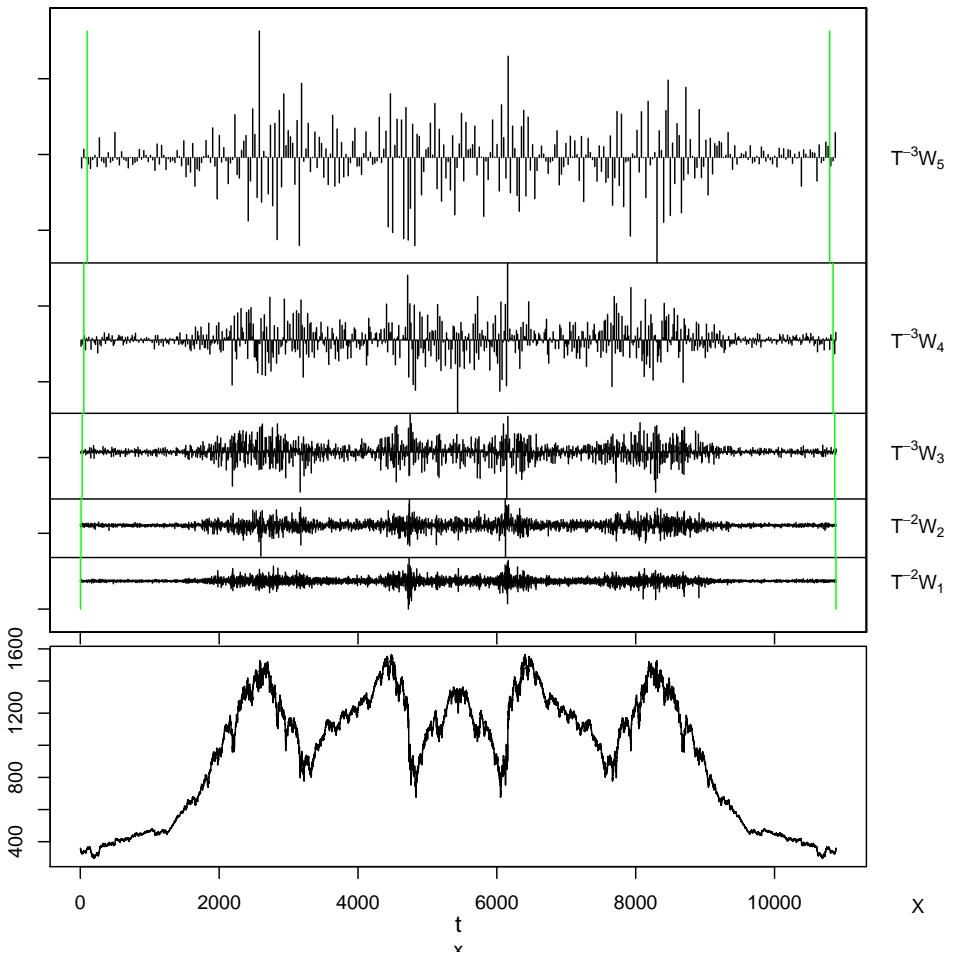


Figure 4.29: DWT of S&P500 Daily Closing Prices with Least Asymmetric(8) Filter by Reflected Data of Wavelet Scales up to 5.

level 6.

In MODWT analyses, decompositions has 8 levels and each level illustrates 2^{j-1} fluctuations up to level 8. From first level to last level, components illustrate fluctuations from 1-day to 128-days. Similarly, 10 level decomposition has fluctuation up to 1024 days.

In each figure of wavelet decomposition, both high and low-frequency fluctuations are seen in time-scale analysis from finest scale (first level) to coarser scales. After level-5, MODWT of the S&P500 daily closing prices show remarkable features. In addition, after 8-th level wavelet coefficients are look like quasi-periodic behavior which means that it has a fractal structure. According to Asian crisis, bubble net (or dot com bubble), and mortgage crisis which occurred respectively in 1997, 1999 and 2007, it is seen that stock prices were affected by unstable economic indicators. In higher scales there are low frequency oscillations and vice versa. MODWT of daily return series provides opportunity of trading for both long-term traders and short-term traders since it shows changes in different scales with related time intervals. It is hard to read day by day, 2

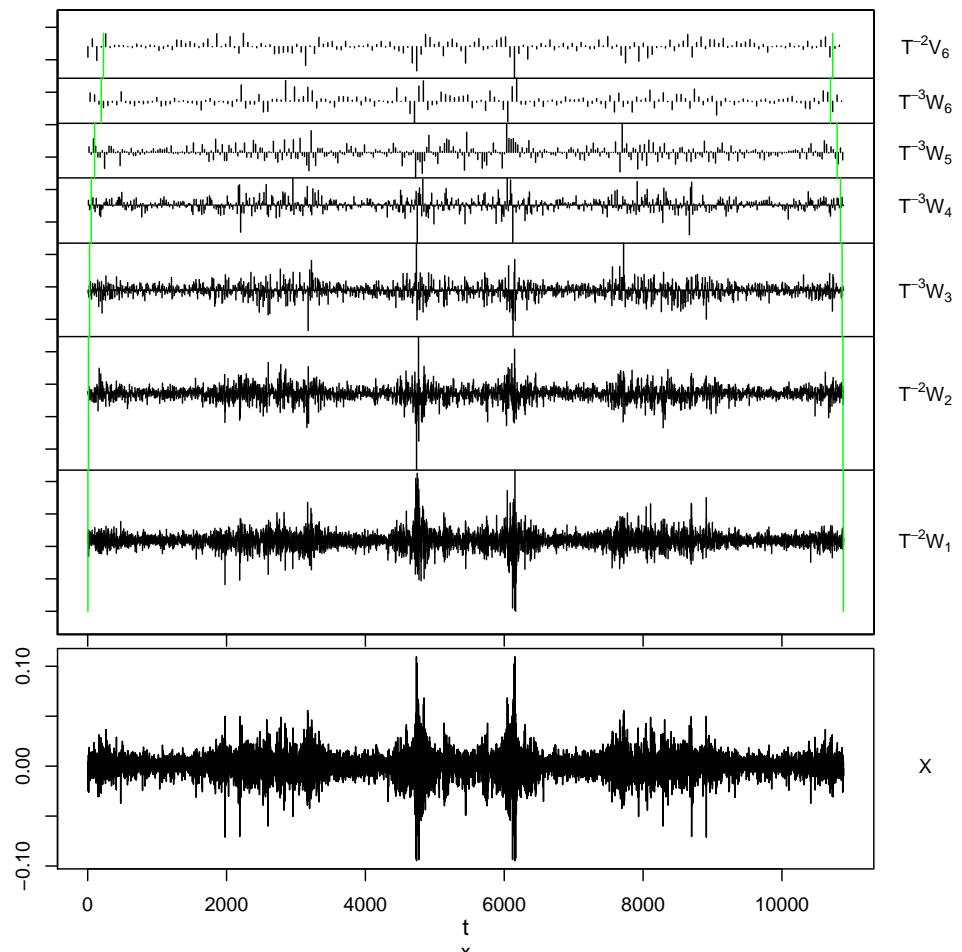


Figure 4.30: DWT of S&P500 Daily Returns with Least Asymmetric(8) Filter by Reflected Data of Wavelet Scales up to 6.

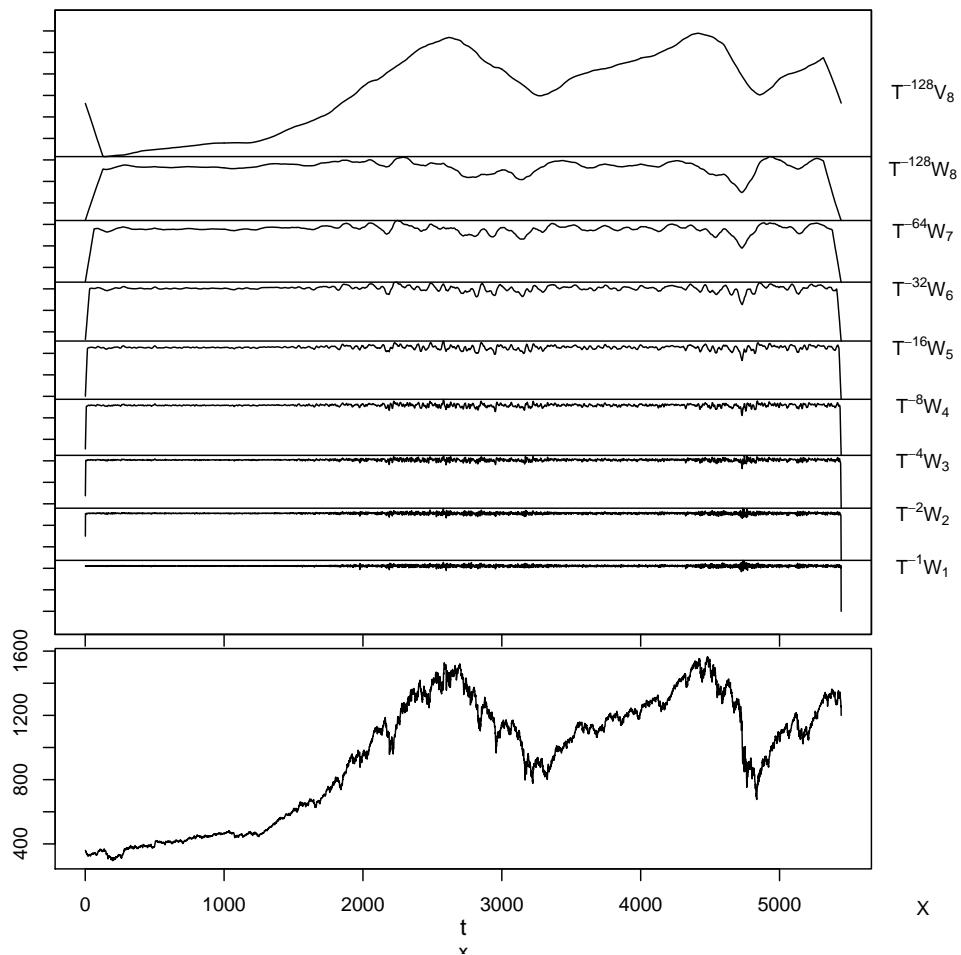


Figure 4.31: MODWT of S&P500 Daily Closing Prices with $J = 8$ by Using Haar Filter.

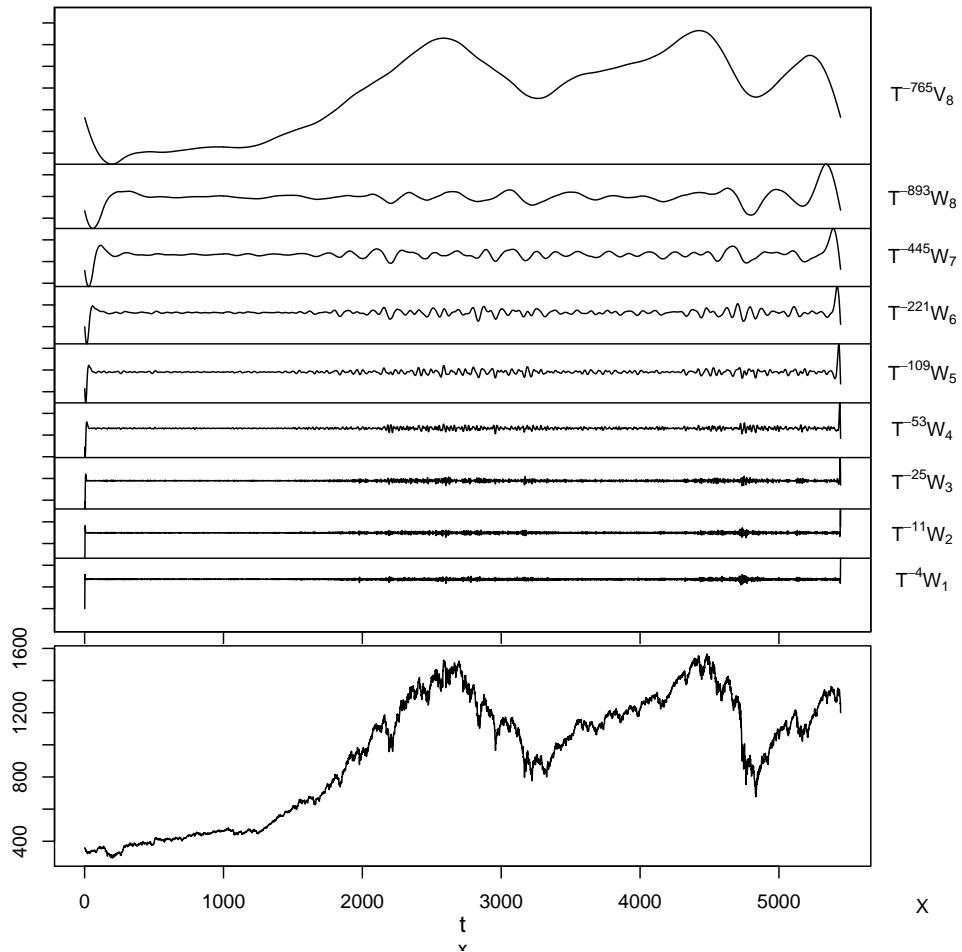
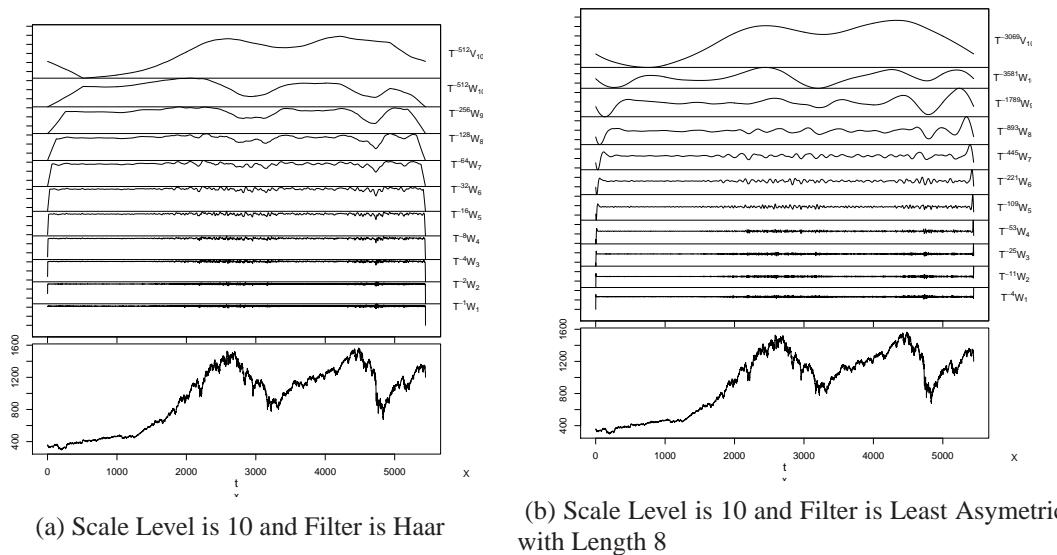


Figure 4.32: MODWT of S&P500 Daily Closing Prices with $J = 8$ by Using Least Asymmetric Filter with Length 8.



(a) Scale Level is 10 and Filter is Haar

(b) Scale Level is 10 and Filter is Least Asymmetric with Length 8

Figure 4.33: MODWT of Daily Closing Prices up to Level 10.

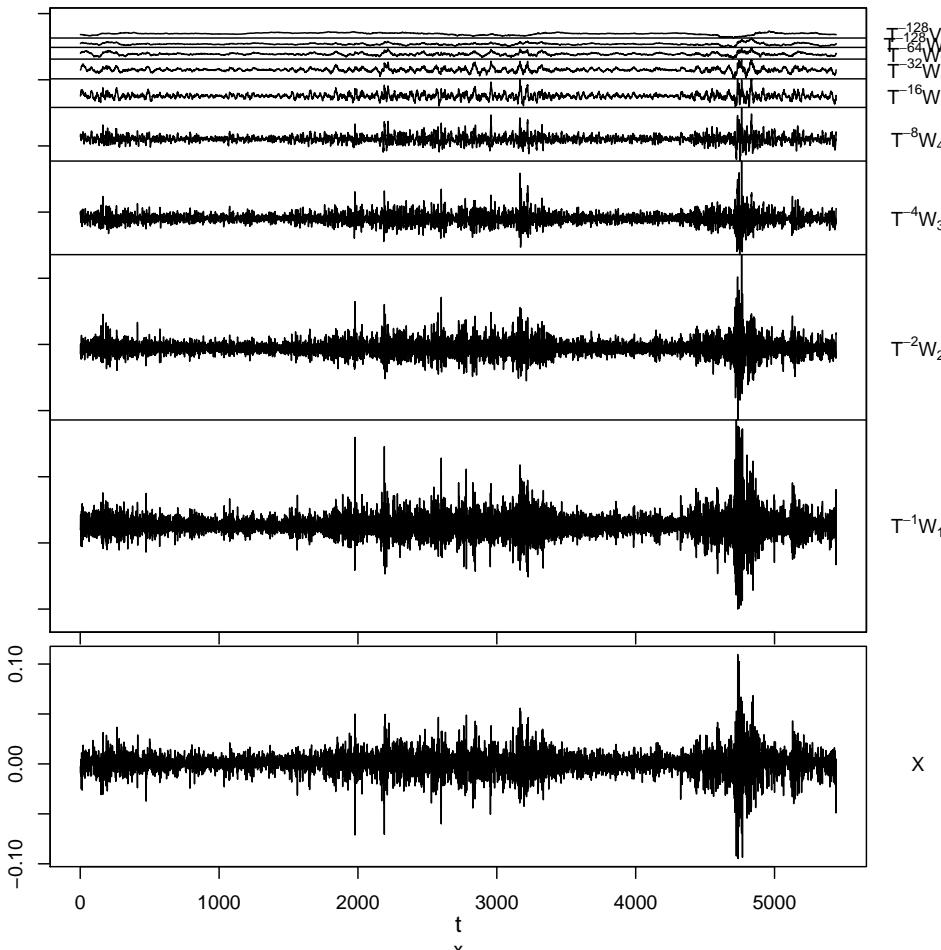


Figure 4.34: MODWT of S&P500 Daily Returns with $J = 8$ by Using Haar Filter.

days, 4 days and 8 days results. They look like noise and it is better to look them in return analysis.

Now on let's look the MODWT of return values.

It is more clear from Figure 4.36 and Figure 4.37 that LA(8) filter is smoother than Haar filter. Moreover, longer wavelet filters are better to eliminate the disorder of the data. Longer filters have higher order vanishing moments and they are used to separate regular and noisy parts from chaotic and fractal parts.

According to MODWT and MRA of return series, there are fluctuations between observations 1900 and 3300 in wavelet coefficients between level 1 and 6. There are short fluctuations between observations 100 and 200 in 2nd and 3rd level wavelet coefficients. Fluctuations around observation 1900 is result of *Asian crisis* which occurred in 1997 [22]. Fluctuations between observations 2100 and 2800 is related to collapse of *internet bubble* in between years 1999 and 2001. In addition there are huge fluctuations between observations in 4200-5100 in wavelet coefficients that has range between 1 and 6. There are high level of volatility between observations due to the

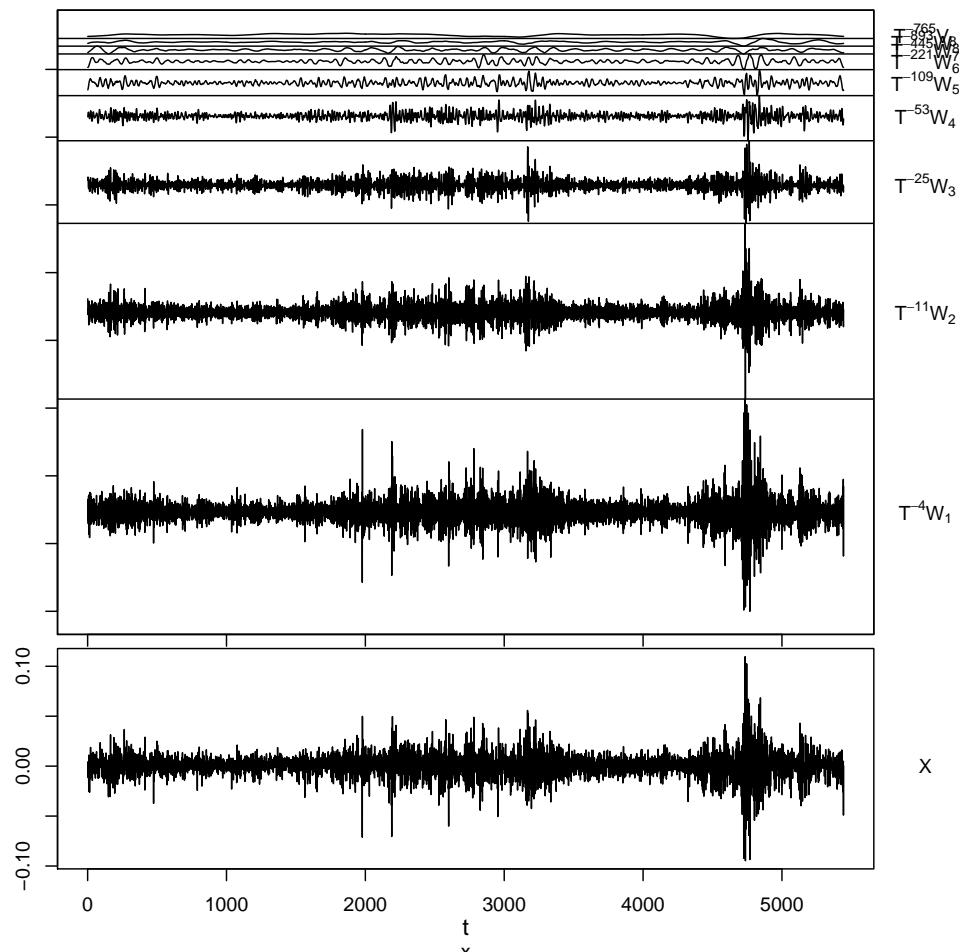


Figure 4.35: MODWT of S&P500 Daily Returns with $J = 8$ by Using Least Asymmetric Filter with Length 8.

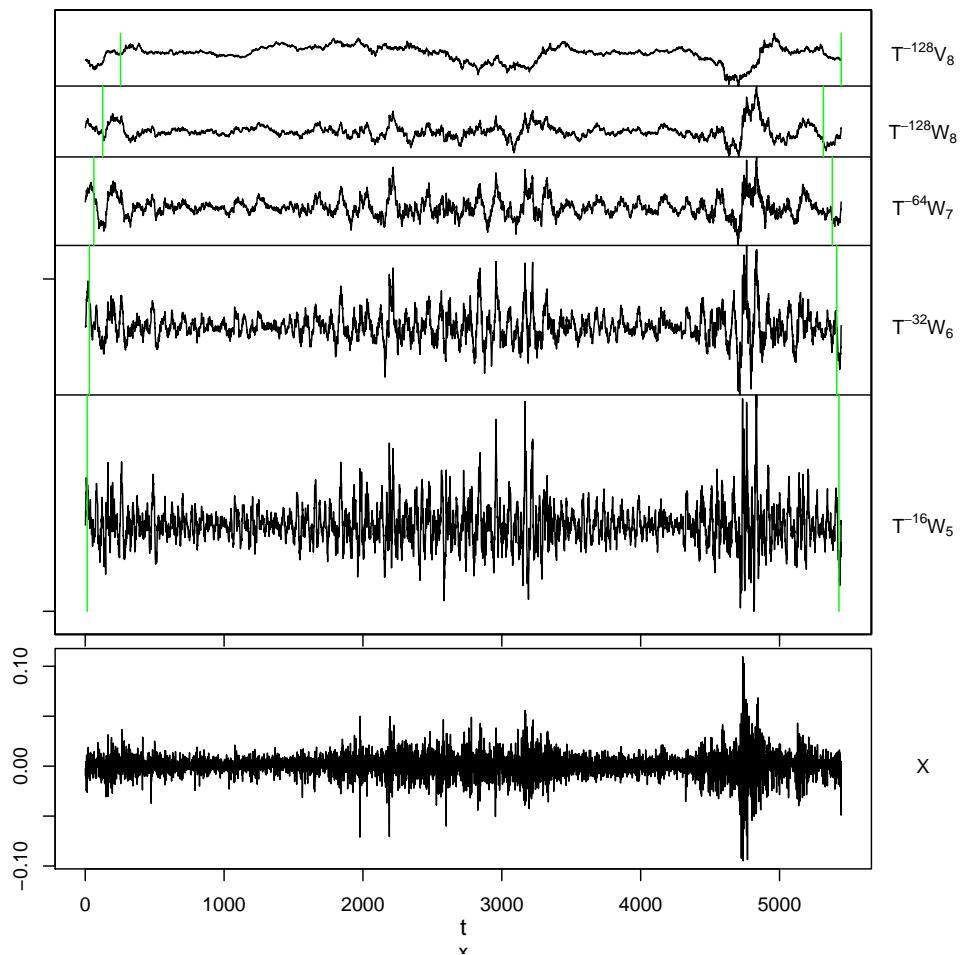


Figure 4.36: MODWT of S&P500 Daily Returns with levels 5, 6, 7, 8 by Using Haar Filter.

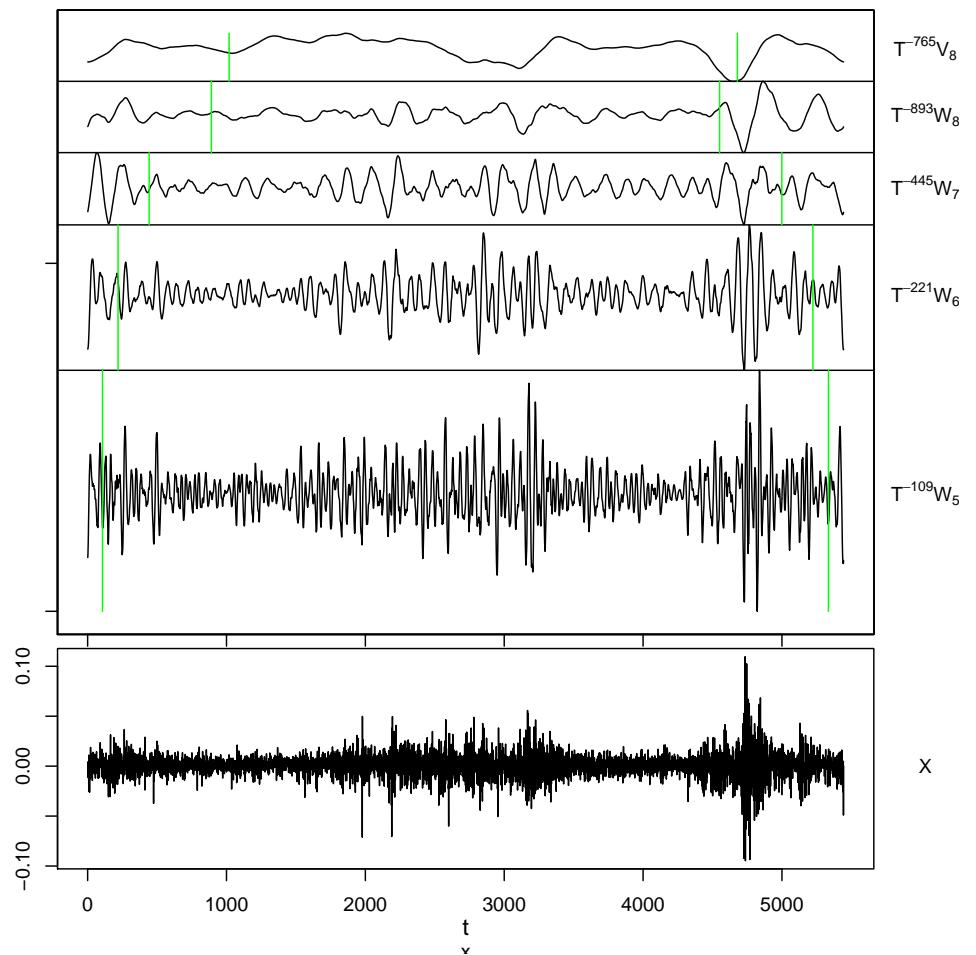


Figure 4.37: MODWT of S&P500 Daily Returns with levels 5, 6, 7, 8 by Using Least Asymmetric Filter with Length 8.

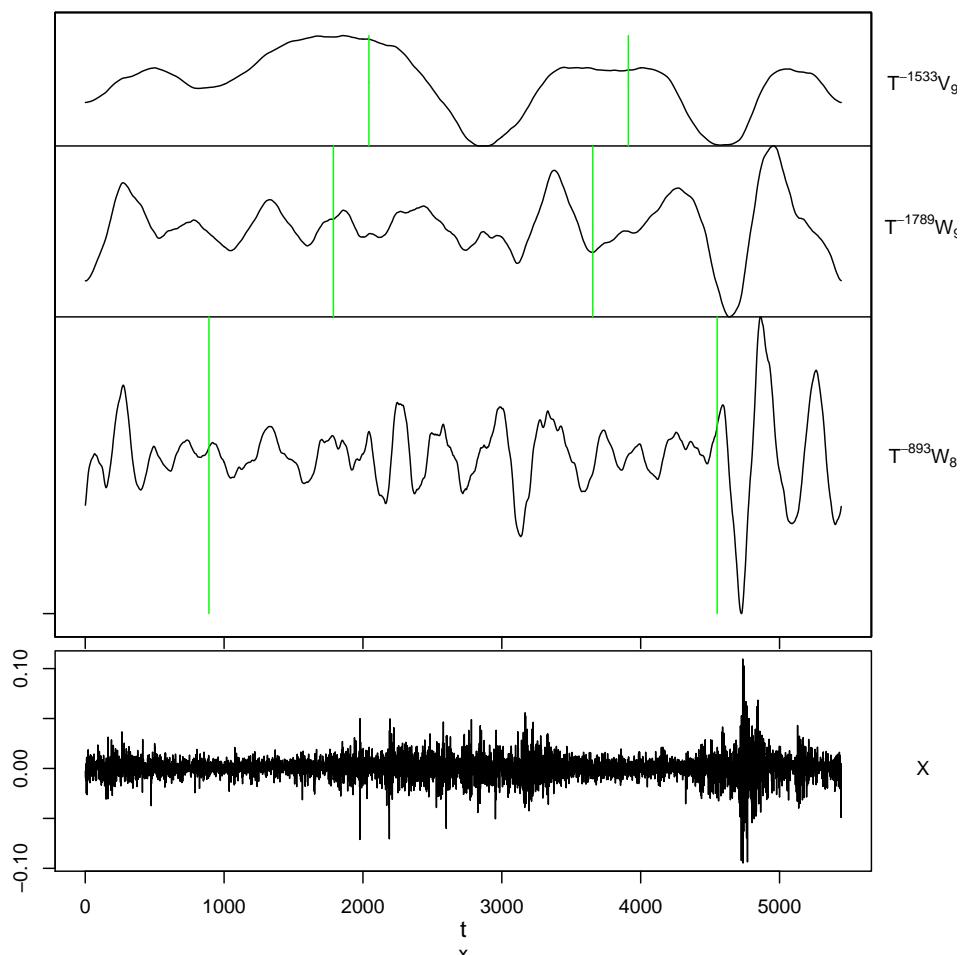


Figure 4.38: MODWT of S&P500 Daily Returns with levels 8, 9 by Using Least Asymmetric Filter with Length 8.

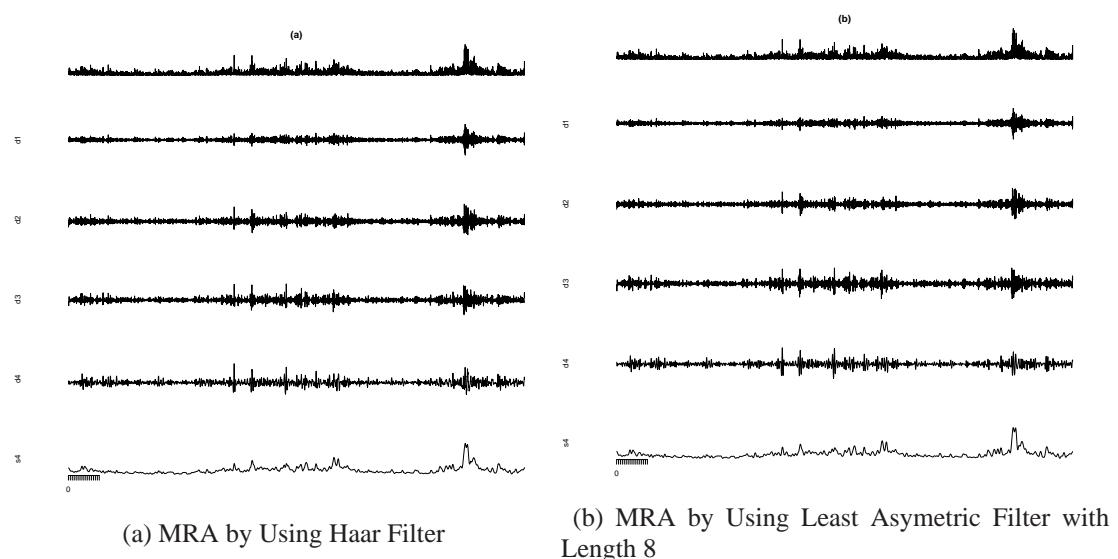


Figure 4.39: MRA of Absolute S&P500 Return Values with Scale Level 4.

global crises occurred in 2007. It happened because of mortgage issue which affected banks terribly and it has continued its bad effect until late 2010. There was a domino effect which influenced all world and all sectors resulted by banking sector. After level 6, we see that there are low frequency oscillations in the series. Wavelet coefficient of level 8 or level 9 show quasi-periodic behavior. We see fluctuations between observations 1900 and 3300 (in higher scales it is divided into more parts) and between observations 4200-5100 there are quasi-periodicities after scale level 7.

Having too many rapid fluctuations in some periods of return series means that there are economic problems in periods that have rapid fluctuations. In the bullish trend of economy, wavelet coefficients of return series are seems to be more stable. On the other hand, the bear trend part of the series gives high fluctuations in the high frequency wavelet coefficients of the return series. Moreover, smooth coefficients of the return series shows falling trend as we seen in V(9) in the MODWT of return series after late 2009.

Wavelet Power Spectrum

The wavelet power spectrum is computed by applying the Morlet wavelet in Figure 4.40, in Figure 4.42 and in Figure 4.44. The confidence intervals are shown by white lines and it is changeable due to the parameters used in function. The vertical axis shows the Fourier periods and the horizontal axis shows time step counts. Quantiles of power or equidistant breakpoints (covering the interval from 0 to maximum power) can describe the color levels, with the number of levels as a further parameter. We are not interested in transparent white parabolic region, i.e. cone of influence. It is seen that the highest power colors stands in periods between 64-128, around 260 and between 1300-1560. In long term, it is seen that original data shows periodicity around 4-6 years. As we see in the wavelet power spectrum, in periods between 4 and 6 years region shows high powers.

Figure 4.41 includes original detrended time series and reconstruction of (detrended) time series analyzed by wavelet transformation.

In Figure 4.42 method shuffles the given time series in estimations. It gives very similar result in reconstruction as we get by “*White Noise*” method.

AR(1) method is weak compared to *white noise* and *shuffle* method for reconstructing.

In Figure 4.46, the *bias-corrected* wavelet power and *biased* wavelet power, which are explained by Liu et al. in [39], are given. Basic algorithm was written by Christopher Torrence and Gibert P. Compo in Matlab [67]. Corrected version provides clearer plots by removing noises. Power increases from light blue to dark red. Significance contours are drawn where spectrum over percentile is equal or larger than tolerance level which is 95% in Figure 4.46. The mother wavelet is Morlet function that is used in the analysis. Between 2000th and 3000th observations, there are middle powered signs around periods 22, 66, 260, 512 and 1024. It shows that there are monthly, several monthly (i.e. 2-3 monthly), yearly or several yearly (i.e. 2, 4 years) seasonal effects in the S&P500 daily closing prices data.

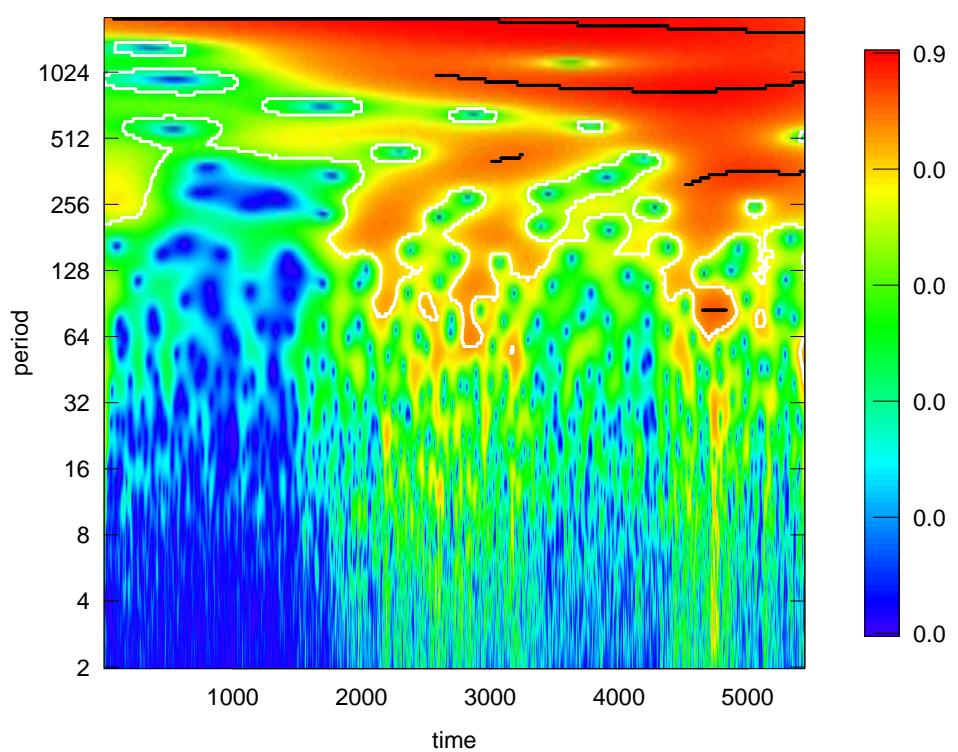
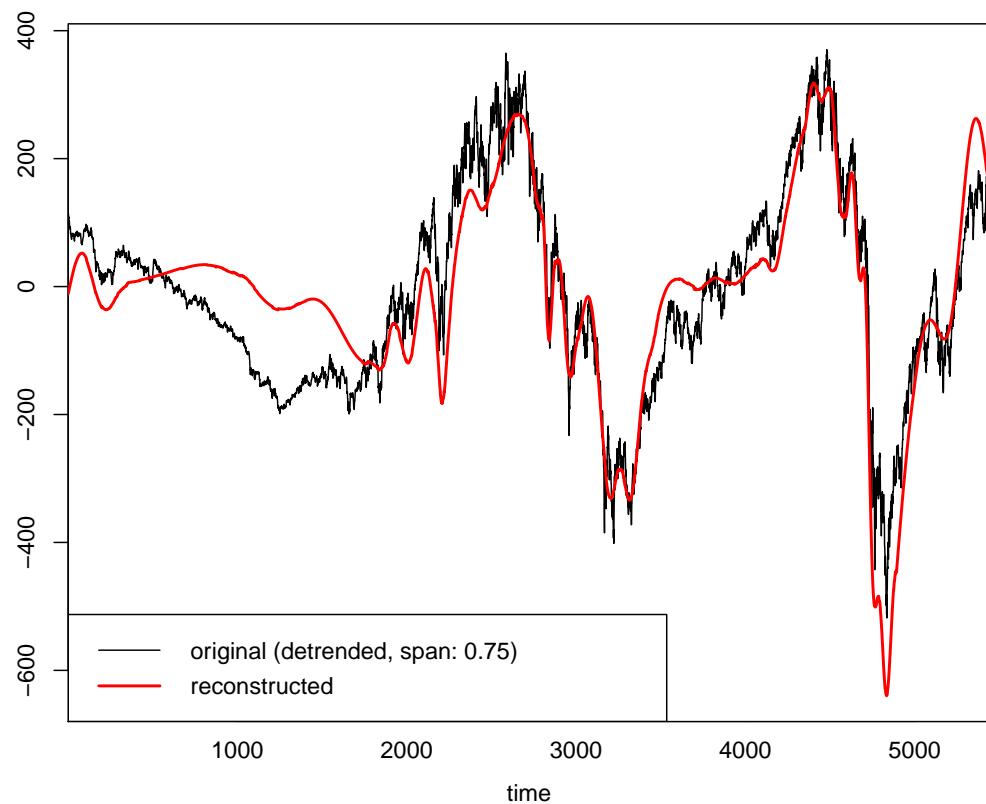


Figure 4.40: Wavelet Power Spectrum by using “white noise” method of generating surrogate time series.



minimum power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: all relevan

Figure 4.41: Reconstruction of the de-trended (0.75 spanned) series by the “white noise” method.

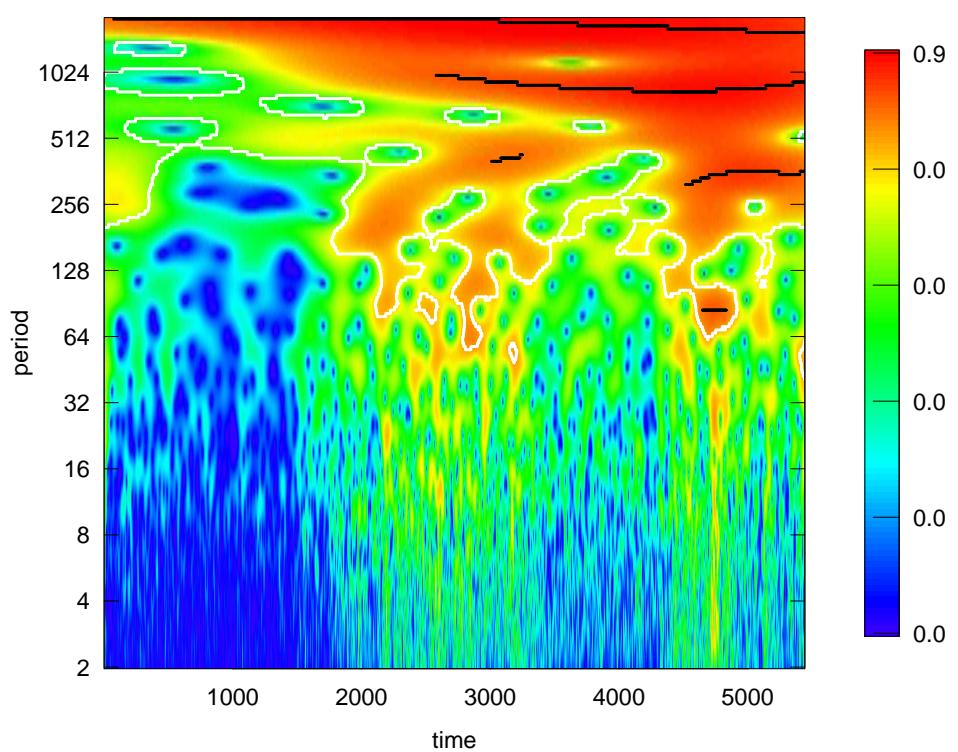
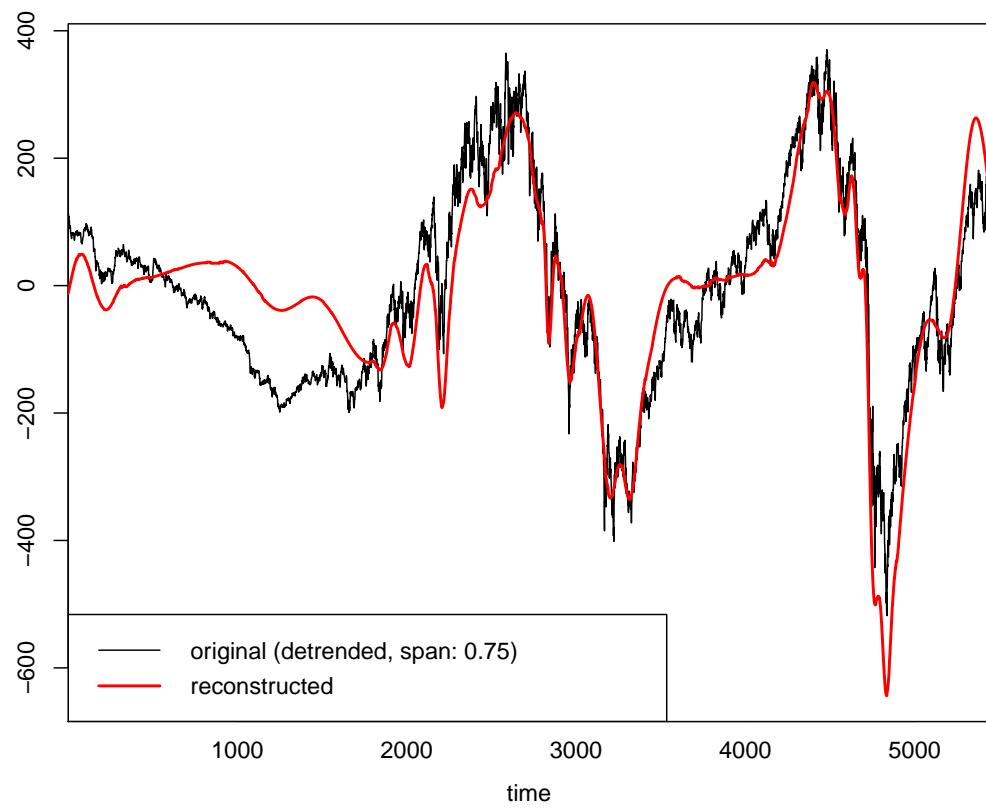


Figure 4.42: Wavelet Power Spectrum by using “shuffle” method of generating surrogate time series.



minimum power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: all relevan

Figure 4.43: Reconstruction of the de-trended (0.75 spanned) series by the “shuffle” method.

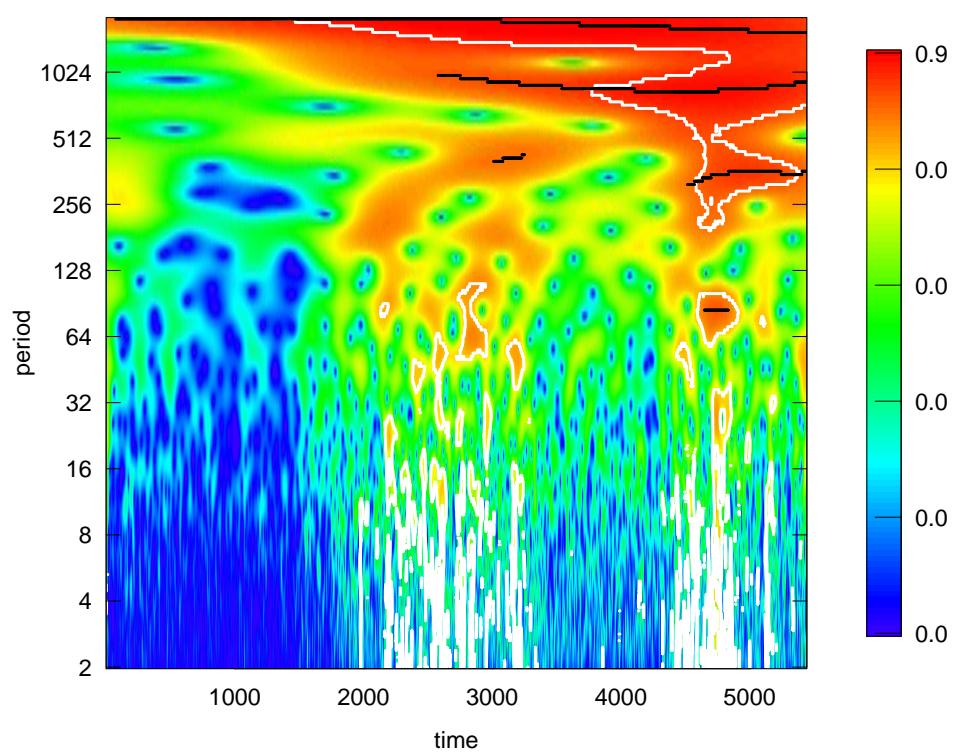
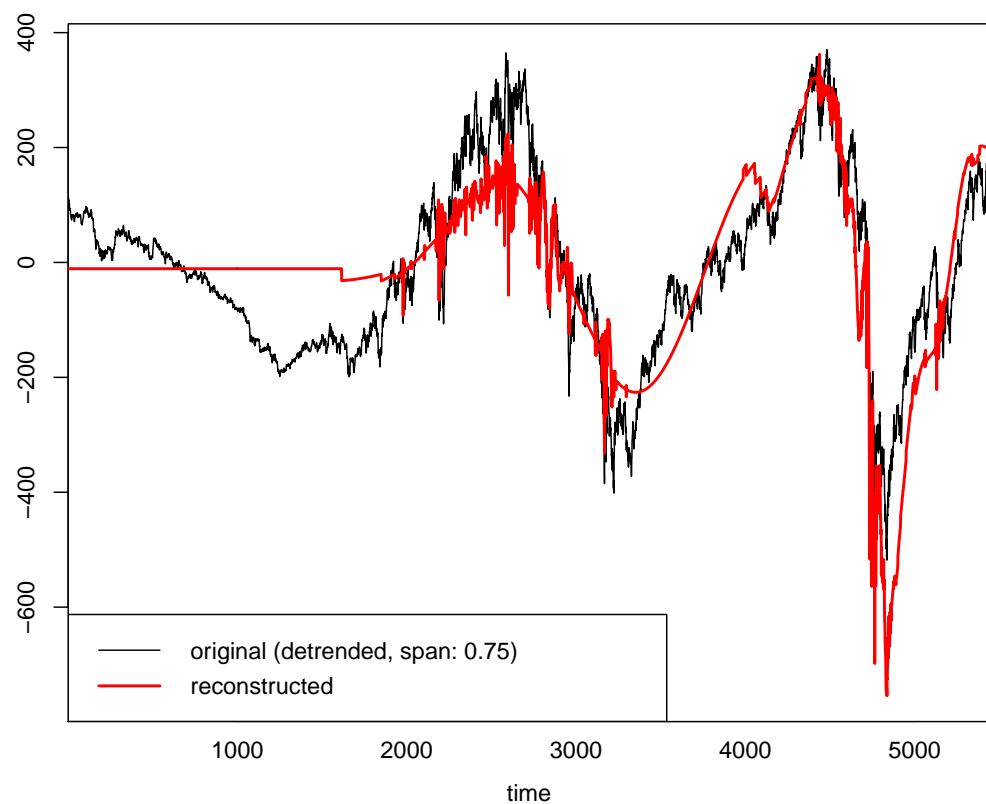


Figure 4.44: Wavelet Power Spectrum by using AR(1) method of generating surrogate time series.



minimum power level: 0, significance level: 0.05, only coi: FALSE, only ridge: FALSE, period: all relevar

Figure 4.45: Reconstruction of the de-trended (0.75 spanned) series by AR(1) method.

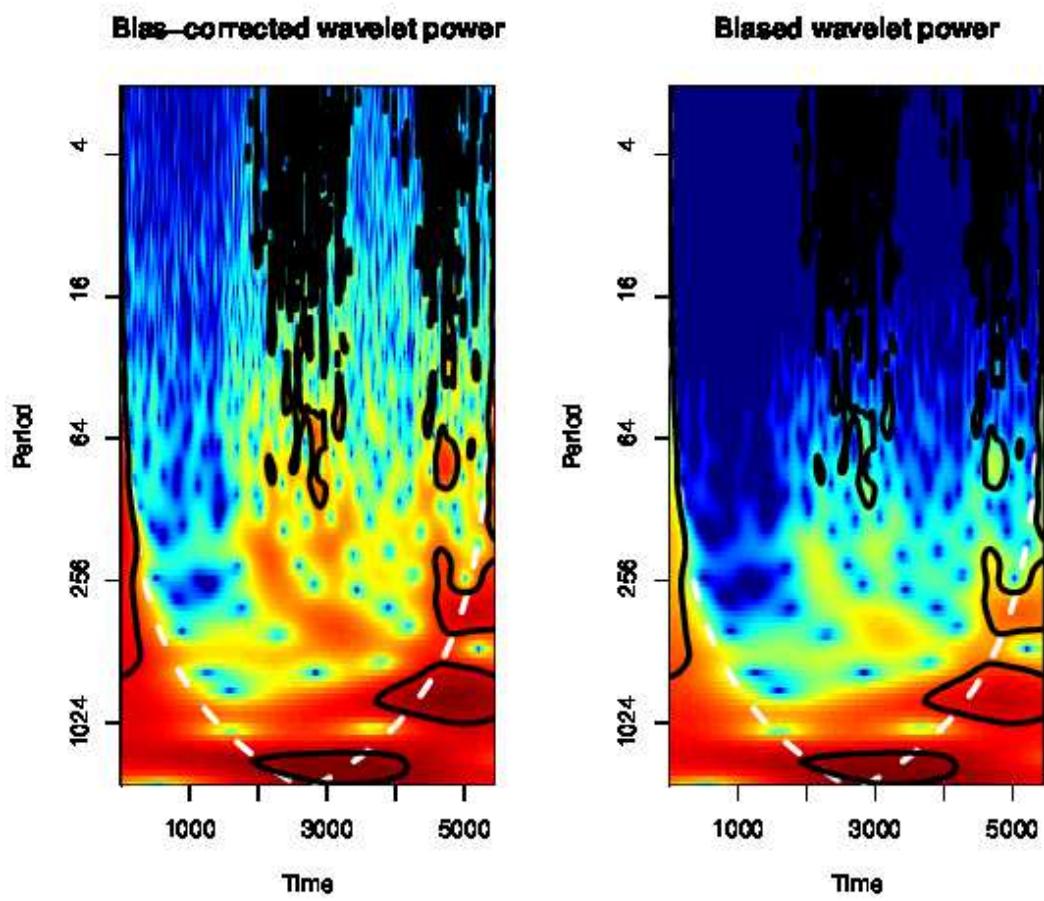


Figure 4.46: *Bias-corrected* and *Biased* Wavelet Spectra of S&P500 Daily Closing Prices.

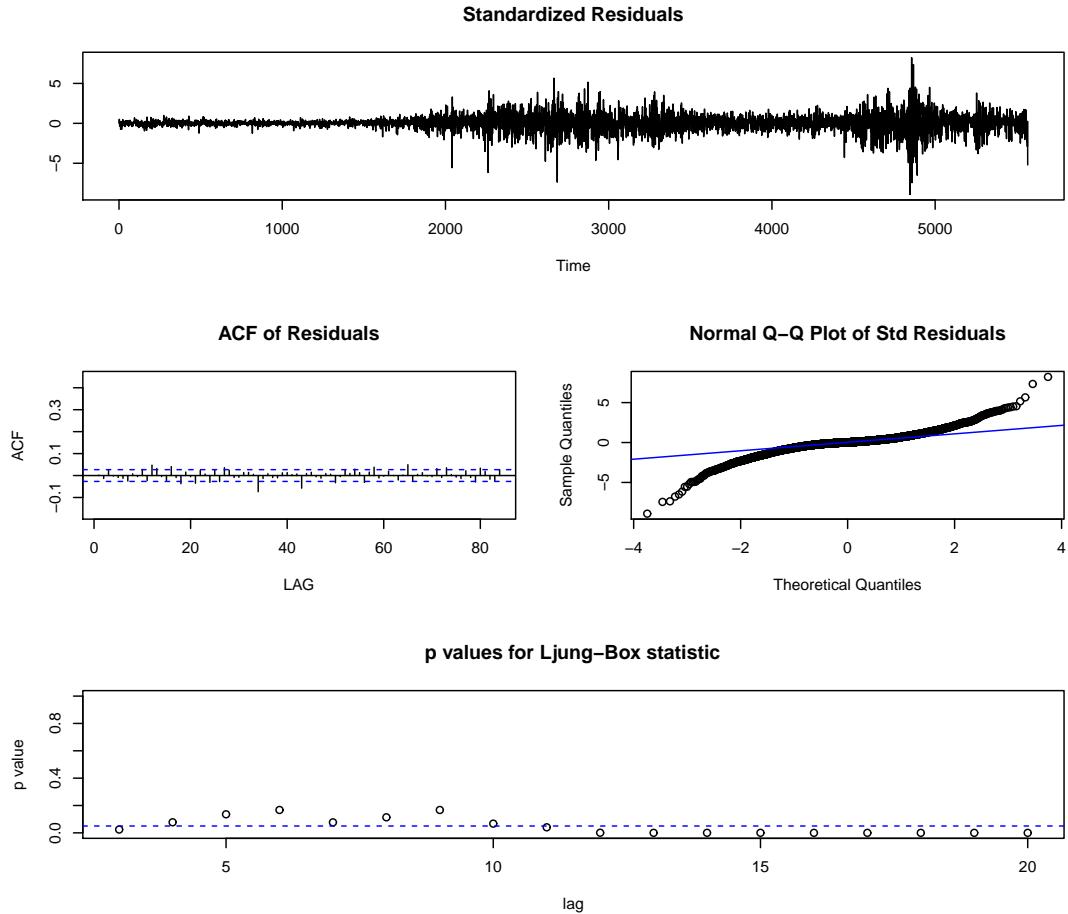


Figure 4.47: ARIMA(1, 1, 1) Fitted S&P500 Daily Closing Prices.

4.4 Time Domain Analysis

4.4.1 SARIMA Fitting

The Box-Jenkins model

Before defining the Box-Jenkins model let's give some example of SARIMA fitted data. In Figure 4.47, in Figure 4.48 and in Figure 4.49 SARIMA fitted data sets are given with related models given in figures.

According to Box-Jenkins model, sharp cut-off of PACF in the original time series shows the AR signature. ACF of our original time series decays more slowly than PACF. Cut-off number of ACF of first difference of logarithm of our original data shows the MA signature.

It is seen from ACF that MA(2) model is appropriate. In PACF graph, it cuts off at lag 1 and so AR(1) model can be appropriate. We can consider ARIMA(0, 1, 2), ARIMA(1, 1, 0) and ARIMA(1, 1, 2) models. One of them can be selected by using

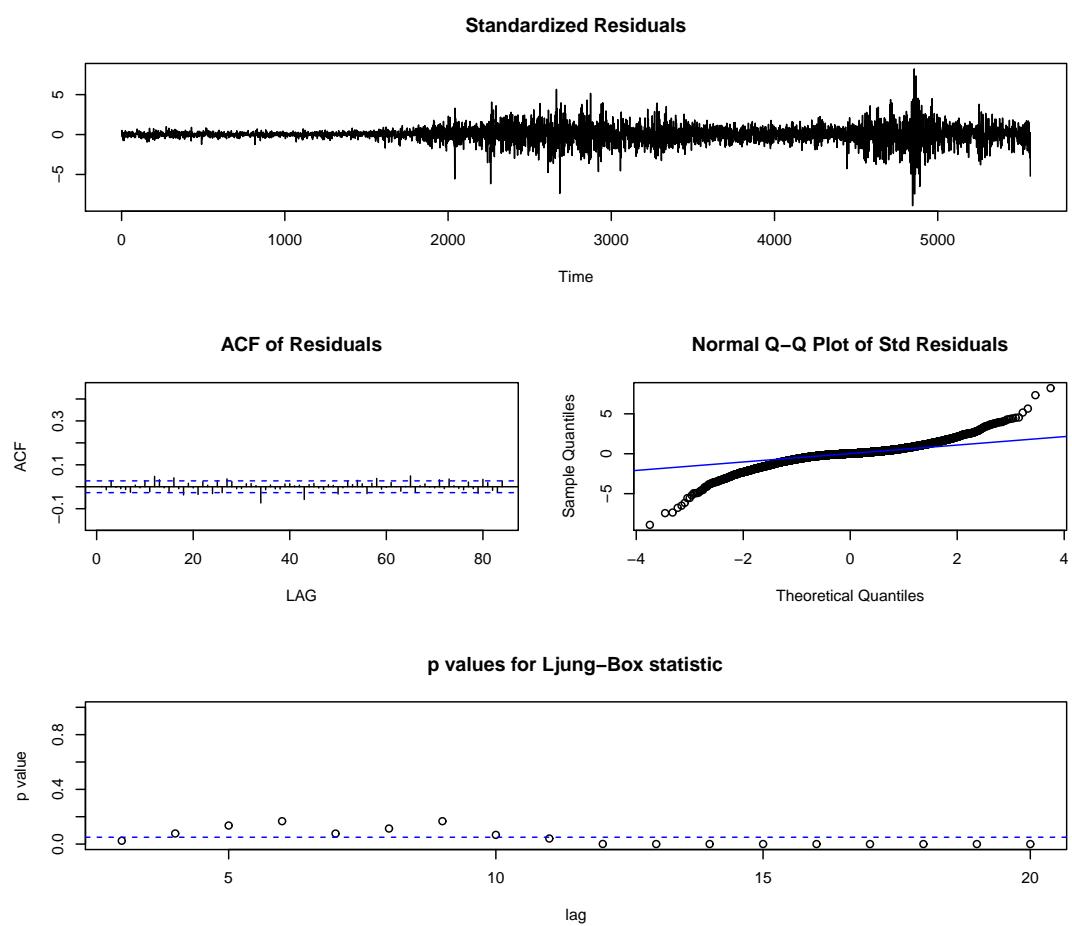


Figure 4.48: SARIMA(1, 1, 1) \times (0, 1, 1)₄₂ Fitted S&P500 Daily Closing Prices with Frequency 260.

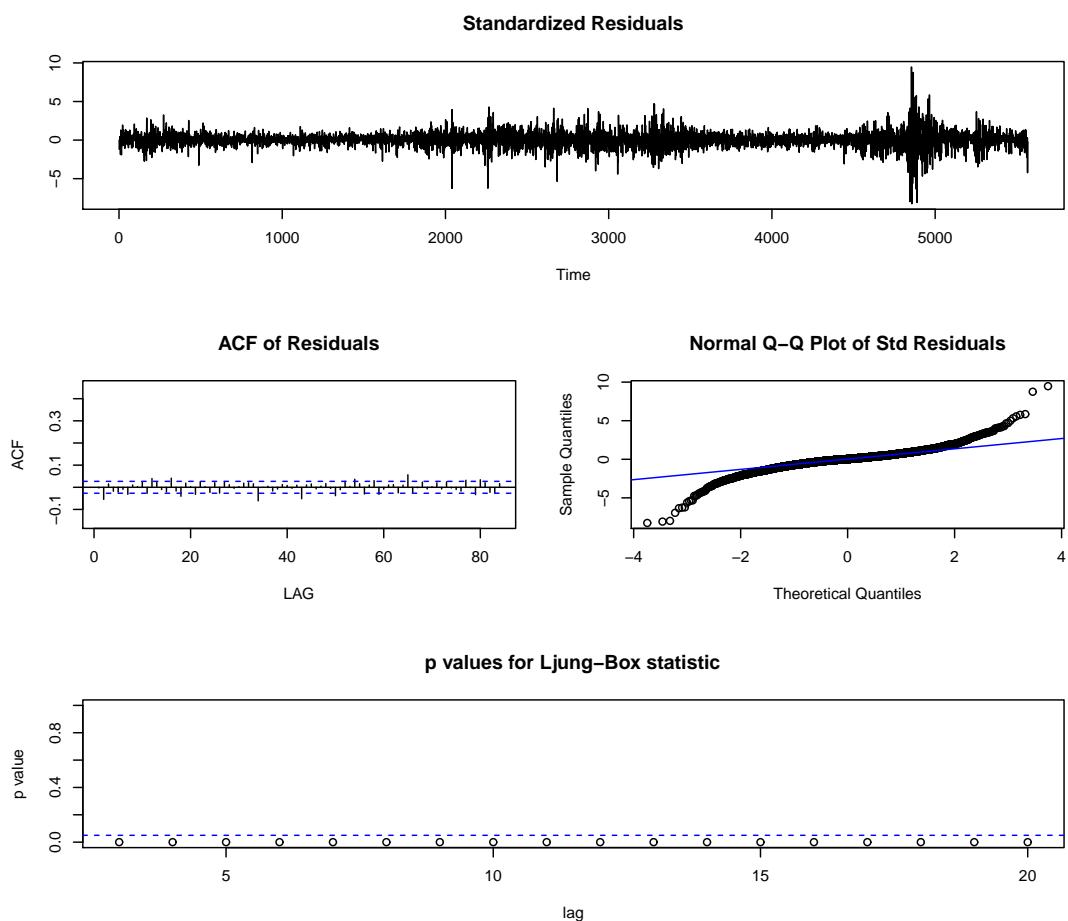


Figure 4.49: ARIMA(1, 1, 1) Fitted S&P500 Return Values.

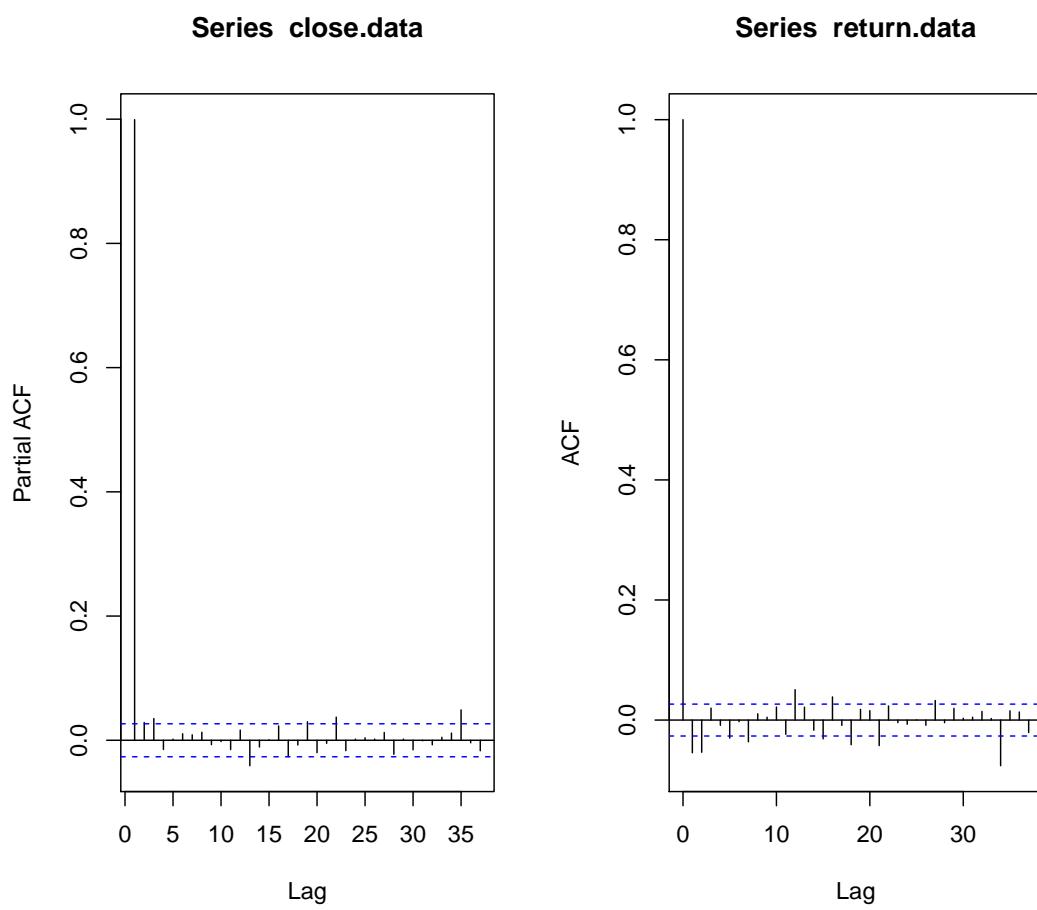


Figure 4.50: PACF of S&P500 Daily Closing Prices and ACF of S&P500 Return Values.

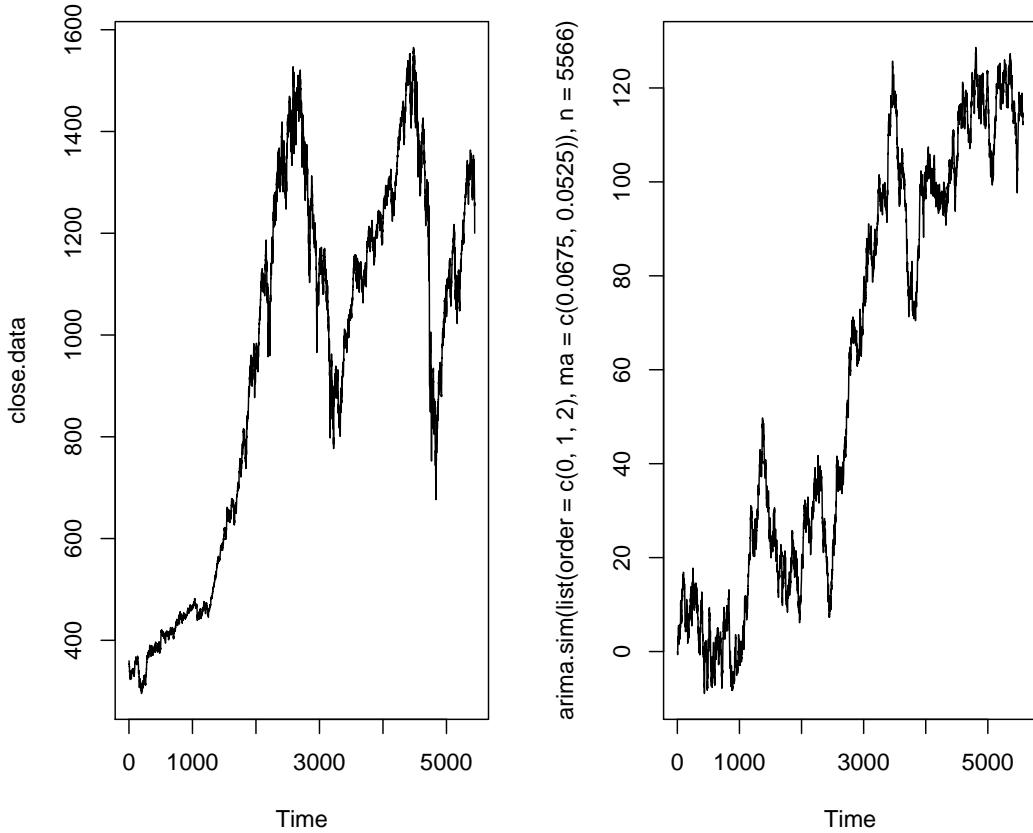


Figure 4.51: Daily Closing Prices of S&P500 versus Simulated ARIMA(0, 1, 2) Related to Original Data.

AIC.

One of the candidates of SARIMA models for S&P500 daily closing price data is ARIMA(0, 1, 2) given by Hyndman and Khandakar in [35]. Original data and the selected ARIMA process are given in Figure 4.51.

It doesn't fit quite well since the simulated ARIMA(0, 1, 2) process gives different graph in each simulation due to generating random values in each simulation. Fitted model can be written as $x_t - \mu = w_t - \theta_1 w_{t-1} - \theta_2 w_{t-2}$ where we apply the ARMA(0, 2) model to first differenced data. If we fit the ARMA(0, 2) model with non-zero mean to first difference of daily closing prices then we will get results with intercepts, i.e. intercept is the mean of the differenced data, which is the drift.

On the other hand, by using BIC, suggested model for daily closing prices turns out to be ARIMA(1, 1, 1). The results of ARIMA(1, 1, 1) model fitting to closing data and the results of ARIMA(1, 0, 1) model fitting to first differenced closing data are given in Table 4.4. Moreover results of ARIMA(0, 1, 2) fitted to daily closing prices and

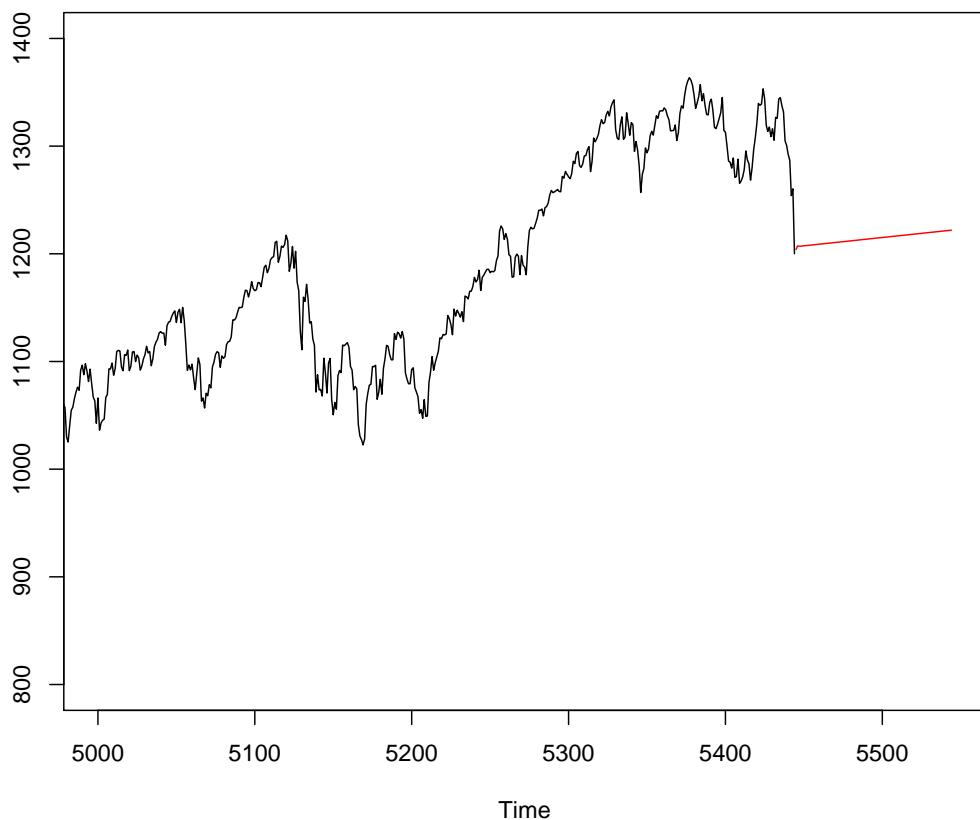


Figure 4.52: 100 Step Prediction of ARIMA(2, 1, 0) Model which is Suggested by AIC.

ARIMA(0, 0, 2) fitted to first difference of daily closing prices are given in Table 4.4.

Best ARIMA model selected by AIC gives ARIMA(2, 1, 0) model whose AIC value is 42533.76 that is smaller than any model in Table 4.4. In Figure 4.52 100 step prediction of S&P500 daily closing prices is seen according to model chosen by AIC.

Forecasts for Auto Selected Models

We will plot the forecast graphs for daily closing prices and its first difference in a 99.5% prediction interval with offered ARIMA models as given in Figure 4.53. Auto selected models are ARIMA(0, 1, 2) and ARIMA(1, 1, 1) by using AIC and BIC respectively.

In Figure 4.54 it is seen that variances of the forecast errors are almost constant. Moreover the forecast errors are not distributed normally and the means are approximately zero as seen in histogram plots. Their means are 0.177002, 0.001344022, 0.1846697 and 0.008642171 respectively. ACF plots of residuals show that the forecast errors do not correlated.

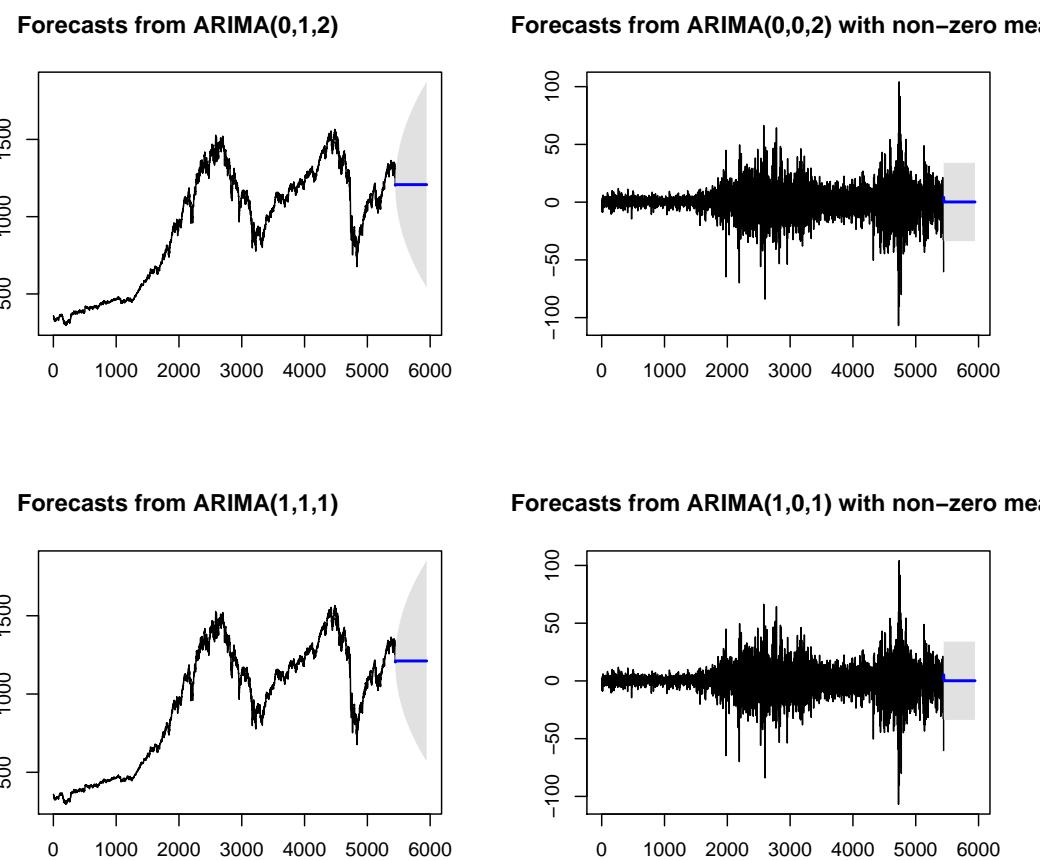


Figure 4.53: Forecasts for S&P500 Data and its First Difference with Auto Selected ARIMA Models.

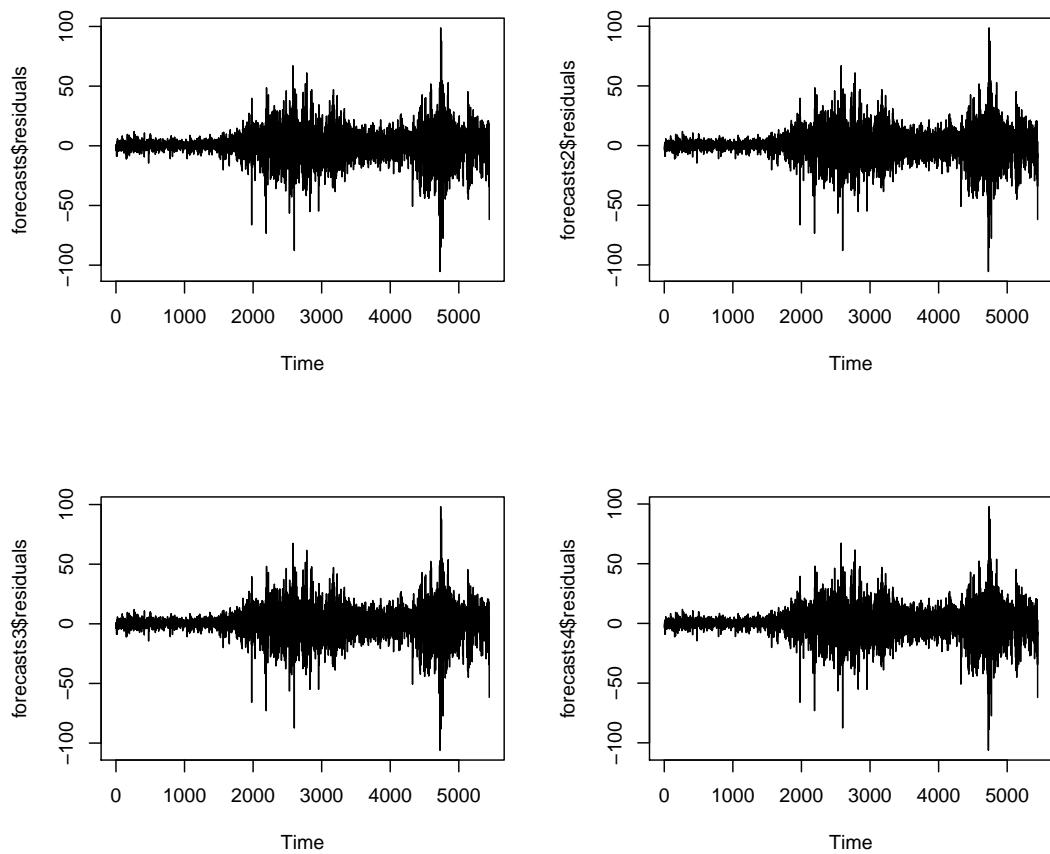


Figure 4.54: Residuals of Forecasts in Figure 4.53 Respectively, i.e. Time Plot of Forecast Errors.

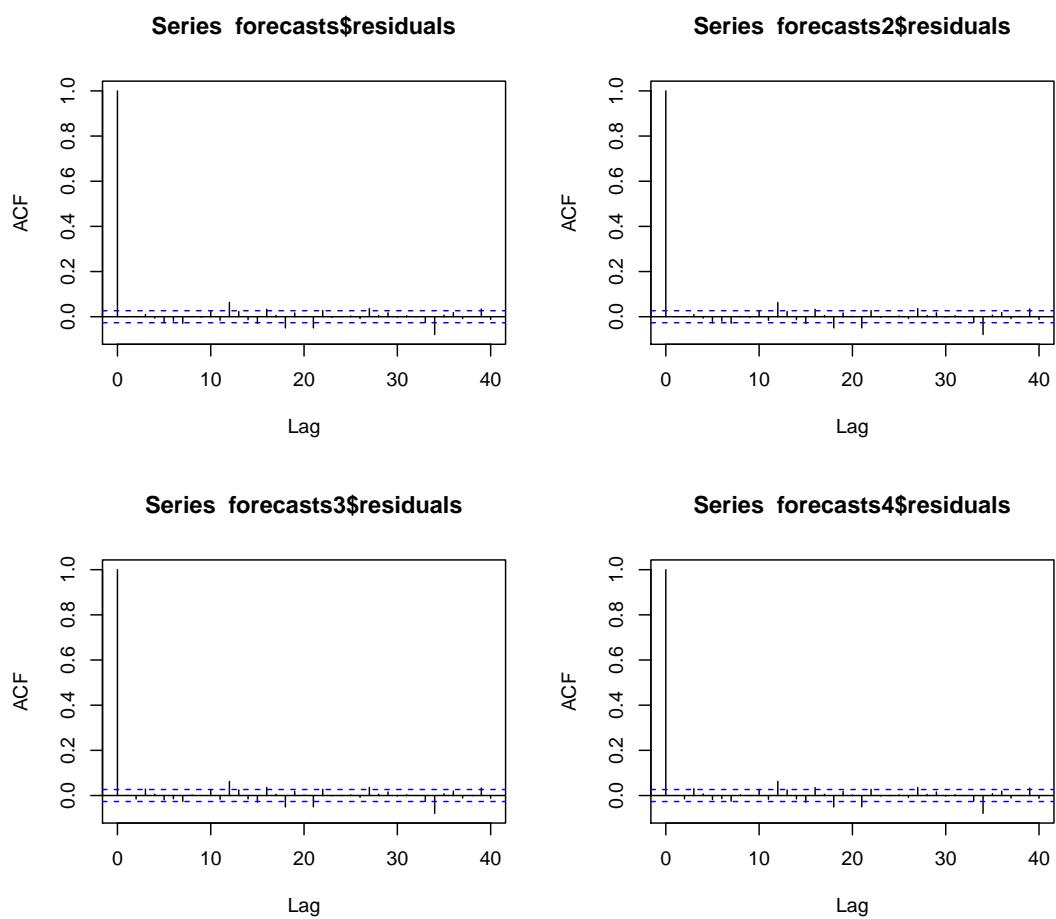


Figure 4.55: ACF Plots of Residuals (forecast errors) of Auto Selected Models given in Figure 4.53.

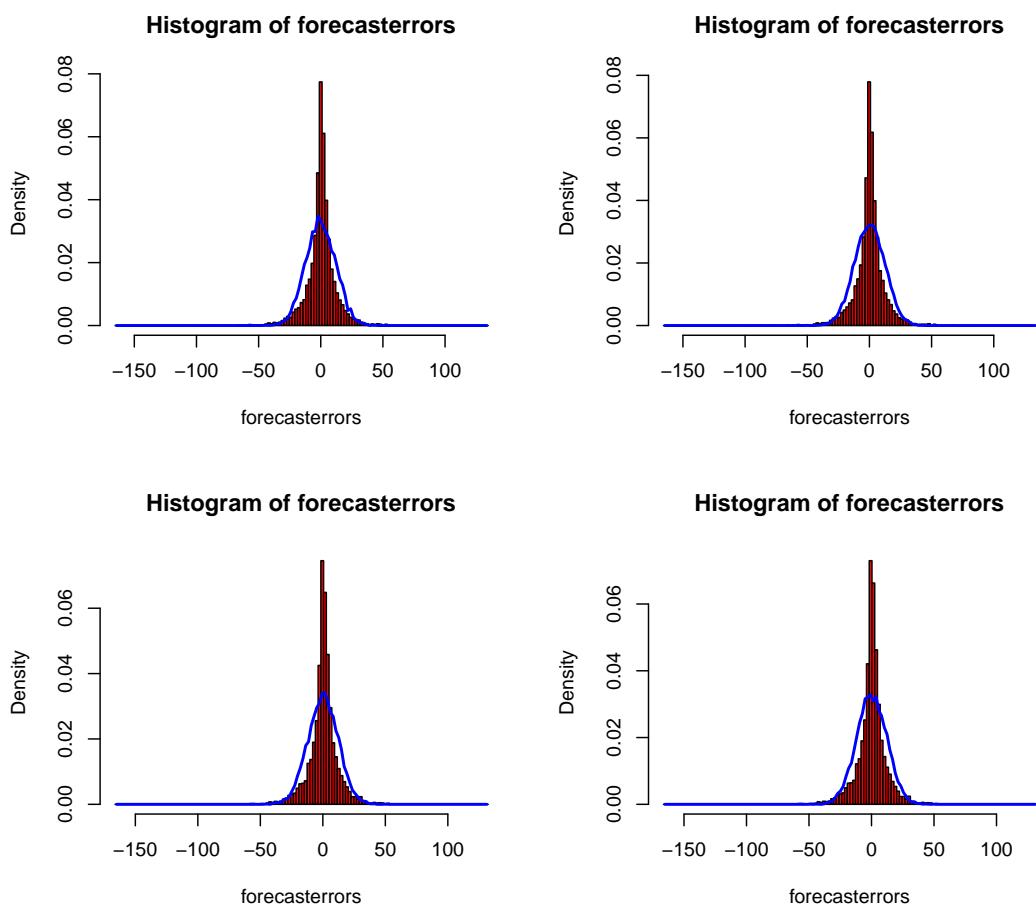


Figure 4.56: Histogram Plots of Forecast Errors Related to Auto Selected Models Respectively in Figure 4.53.

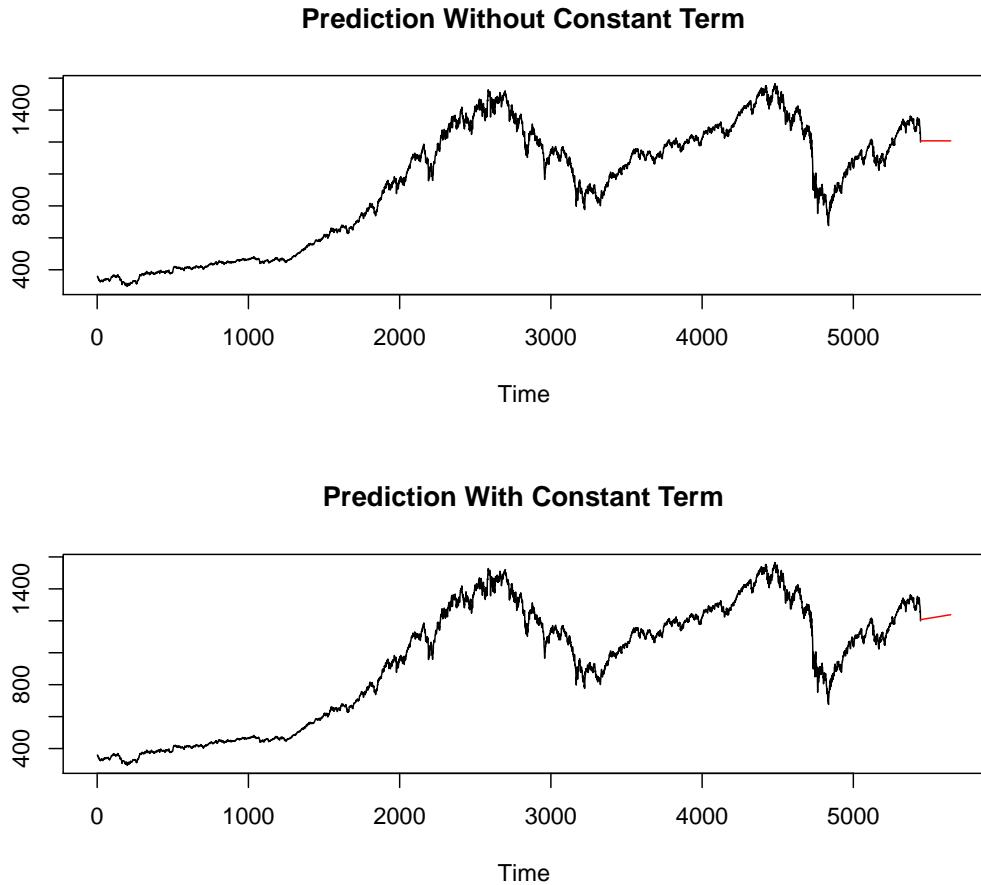


Figure 4.57: Prediction without Drift and with Drift modelled by ARIMA(0, 1, 2) Model.

The problem with forecasts in Figure 4.53 is not considering the constant term in models, which is also called drift. If we include the drift term in the forecasting than we will get trend term as well. In Figure 4.58 difference between prediction with constant term and prediction without constant term is seen.

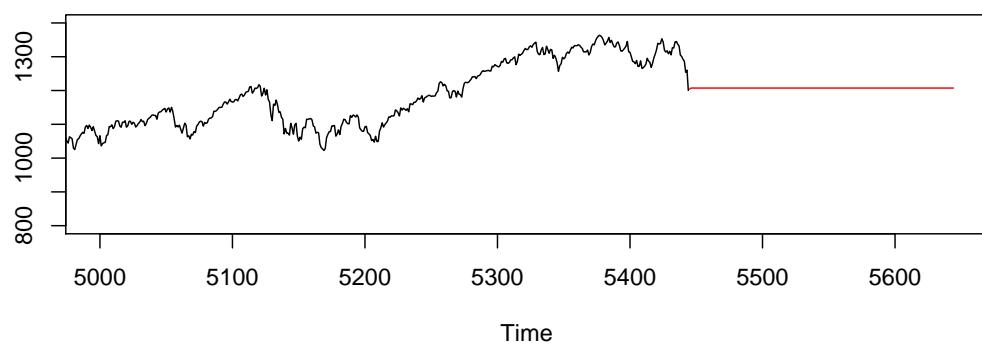
Prediction with constant term catches some trend compared to prediction without drift term. In Figure 4.59 following predictions include the drift terms.

We showed some results of some SARIMA model. Now we will show additional results of different model fitting in Table 4.4. In model $\text{SARIMA}(p, d, q) \times (P, D, Q)_s$, parameters p, d, q, P, D, Q and s are AR order, difference order, MA order, seasonal AR order, seasonal MA order, seasonal difference and seasonal period respectively. Parameters P, D, Q and s are used for only seasonal models.

In Table 4.4, `close.data`, `return.data` and `freq.data` refer to daily closing prices, return values and daily closing prices with frequency 260 respectively.

It is seen in Table 4.4 ARIMA(2, 1, 0) gives the smallest AIC if we don't use seasonal

Prediction Without Constant Term



Prediction With Constant Term

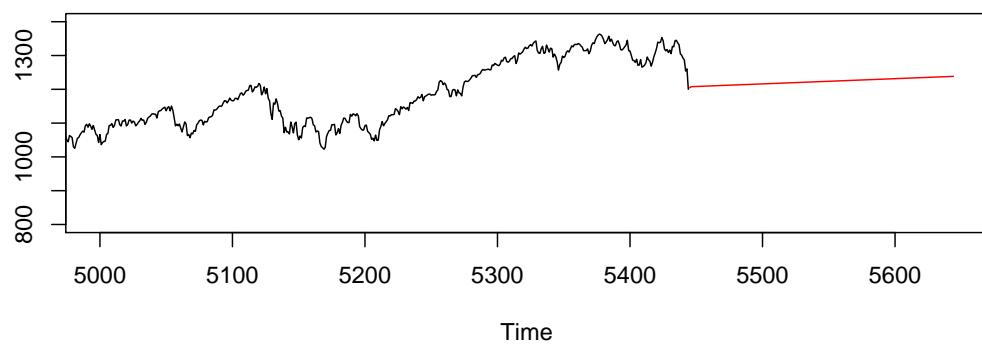


Figure 4.58: Zoomed Version of Prediction without Drift and with Drift modelled by ARIMA(0, 1, 2) Model.

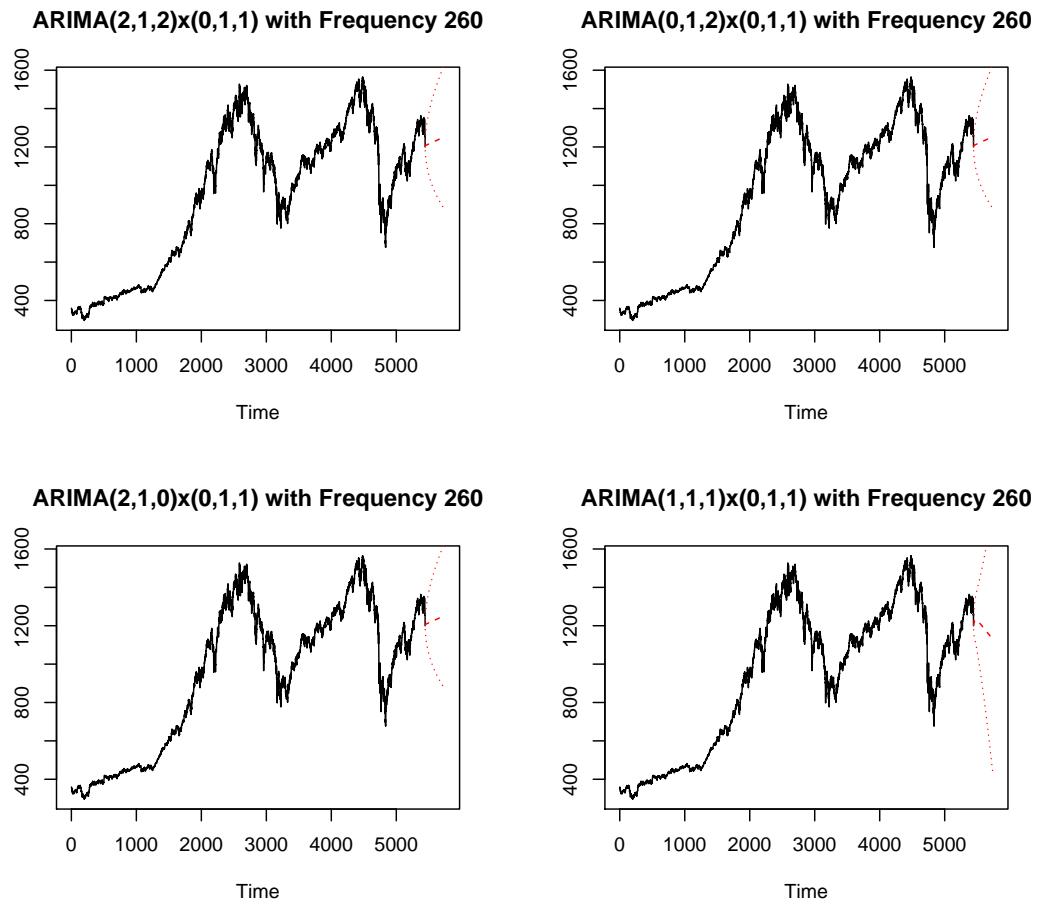


Figure 4.59: Predictions of S&P500 Daily Closing Prices of with Given SARIMA Models.

Table 4.4: Results of SARIMA Fitting to Related Data

	sigma^2	log likelihood	AIC	AICc	BIC
ARIMA(0,1,2) close.data	144.8	-21259.51	42527.02	42527.03	42553.43
ARIMA(0,0,2) diff(close.data)	144.8	-21263.42	42534.83	42534.84	42561.24
ARIMA (1,1,1) close.data	144.9	-21260.55	42529.1	42529.11	42555.51
ARIMA(1,0,1) diff(close.data)	144.8	-21264.46	42536.92	42536.92	42563.33
ARIMA(1,2,2) close.data	144.9	-21265.52	42539.04	42539.05	42565.45
ARIMA(1,1,2) close.data	144.8	-21259.35	42528.7	42528.71	42561.71
ARIMA(0,1,1) close.data	145.2	-21266.85	42539.7	42539.7	42559.5
ARIMA(1,1,0) close.data	145.3	-21268.23	42542.46	42542.46	42562.27
ARIMA(2,1,0) close.data	144.8	-21259.4	42526.8	42526.8	42553.2
ARIMA(2,1,1) close.data	144.8	-21259.38	42528.75	42528.77	42561.76
SARIMA(1,1,1,0,1,1,42) freq.close	144.9	-21203.98	42415.95	42415.96	42442.33
SARIMA(1,2,2,0,1,1,42) freq.close	146	-21226.21	42464.42	42464.44	42503.99
SARIMA(1,1,1,0,1,1,22) freq.close	145.5	-21239.84	42487.69	42487.69	42514.08
SARIMA(1,1,1,0,1,1,21) freq.close	145	-21243.66	42495.32	42495.33	42521.71
SARIMA(1,1,1,0,1,1,63) freq.close	144.9	-21159.62	42327.25	42327.26	42353.61
SARIMA(1,1,2,0,1,1,130) freq.close	144.8	-21001.95	42013.9	42013.92	42046.79
SARIMA(2,1,0,0,1,1,130) freq.close	144.8	-21002	42012.01	42012.02	42038.32
SARIMA(0,1,2,0,1,1,130) freq.close	144.8	-21001.97	42011.94	42011.95	42038.25

model part. ARIMA(1, 1, 1) gives the second smallest AIC value in ARIMA models. On the other hand if we use the seasonal parts SARIMA($0, 1, 2 \times (0, 1, 1)_{130}$) gives the smallest AIC. It means that seasonality is important in modeling financial time series. SARIMA models with frequencies which are also captured by wavelet analysis in Subsection 4.3.2, have smaller AIC values than ARIMA models which are not taking into account the seasonality. Therefore in order to make prediction, it is better to use SARIMA models compared to ARIMA for financial time series.

Forecasting Accuracies

We measures the forecast accuracy to understand which model gives better result. Here we choose 5344 observations out of 5444 observations to be training set. Remained observations form test set to compare real values with forecasting values. Details of accuracy measures are described by Hyndman and Koehler in [36].

In Table 4.5, in Table 4.6and in Table 4.7 definitions of abbreviations are;

- ME: Mean Error
- RMSE: Root Mean Absolute Error
- MAE: Mean Absolute Error
- MPE: Mean Percentage Error
- MAPE: Mean Absolute Percentage Error
- MASE: Mean Absolute Scaled Error
- ACF1: Autocorrelation of errors at lag 1.

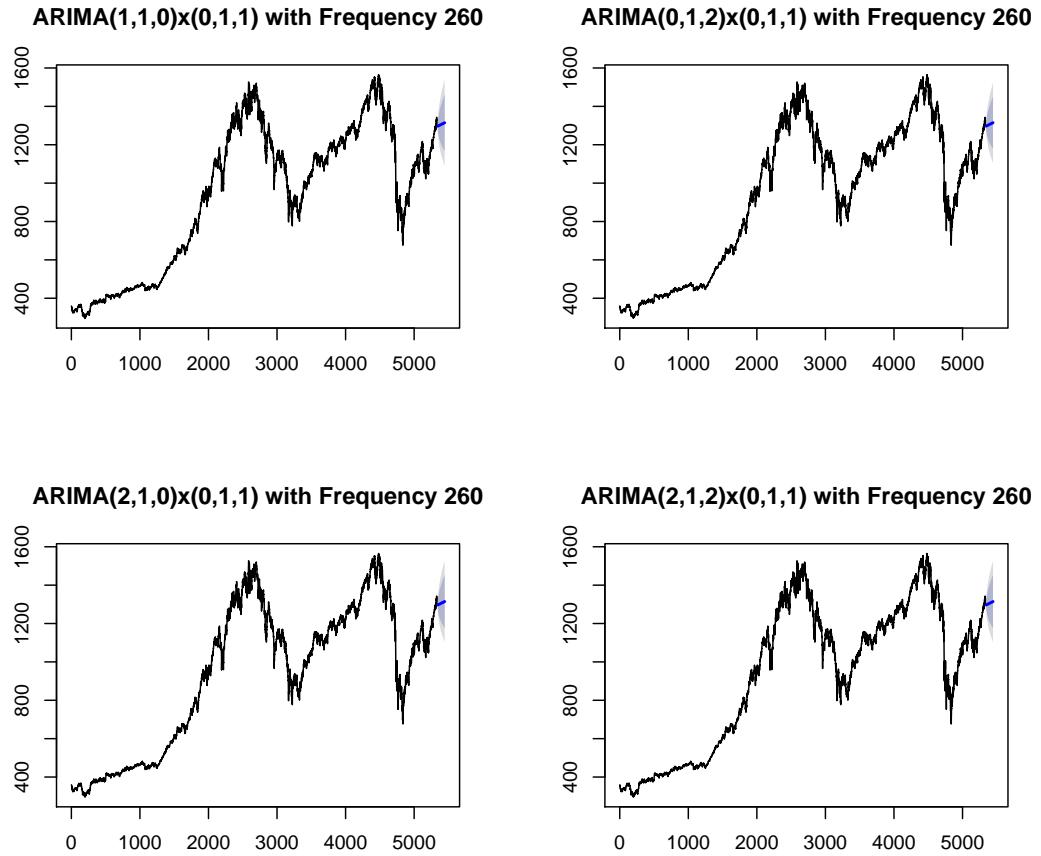


Figure 4.60: Forecasts of Daily Closing Prices with Selected SARIMA Models.

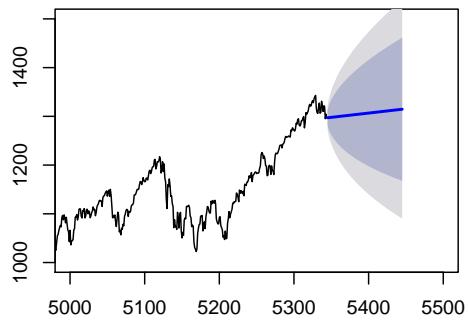
In Table 4.5 ARIMA(1, 1, 1), ARIMA(0, 1, 2) and ARIMA(2, 1, 0) have smaller test set errors. ARIMA(1, 1, 1) has smallest values in each measure of accuracy. In Table 4.6 SARIMA($(0, 1, 2) \times (0, 1, 1)_{260}$) model gives smallest test set errors except ACF1 method. In Table 4.7 results are given again for ARIMA(1, 1, 1), ARIMA(0, 1, 2), ARIMA(2, 1, 0) and SARIMA($(0, 1, 2) \times (0, 1, 1)_{260}$). It is seen that seasonal autoregressive integrated moving average model has better accuracy values than others. In Figure 4.60 four different forecasts are given with 100 ahead. In Figure 4.61 selected four models give smaller accuracies than other models in forecasting.

Wavelet Transform Based ARIMA Fitting

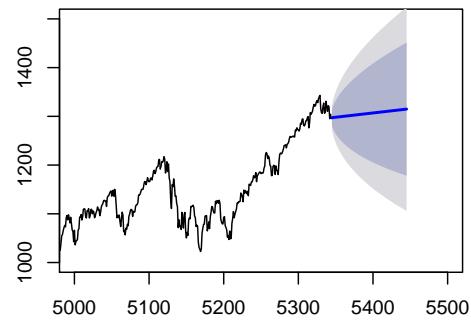
In Table 4.5, it is seen that ARIMA(1, 1, 1) model has better accuracy values compared to other ARIMA models. If we consider the ARIMA(1, 1, 1) model fitting to daily closing prices of S&P500 data with first 5344 observations, we get sum of residuals as 1121,65 for 100 step ahead forecasts. Residuals are differences between daily closing prices and fitted values.

Then we apply DWT and MODWT to S&P500 daily closing prices with first 4096 and 5344 observations respectively. We used 4096 observations for DWT, because

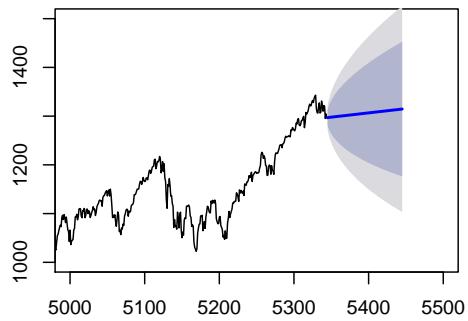
ARIMA(1,1,0)x(0,1,1) with Frequency 260



ARIMA(0,1,2)x(0,1,1) with Frequency 260



ARIMA(2,1,0)x(0,1,1) with Frequency 260



ARIMA(2,1,2)x(0,1,1) with Frequency 260

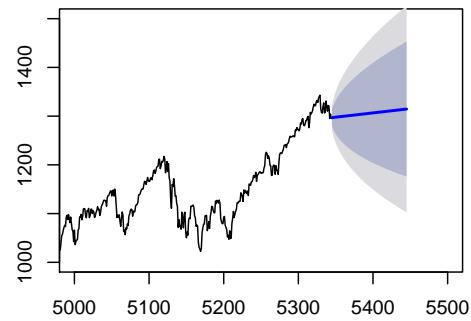


Figure 4.61: Zoomed Versions of Forecasts of Daily Closing Prices with Selected SARIMA Models.

transform requires time series with dyadic length. We used least asymmetric wavelet filter with length 8 in both transform. In addition, scale levels are chosen as 2 for both transformations, i.e. we have 2 detail parts and 1 approximation part for both transformations. ARIMA(1, 1, 1) model was fitted to all scales of DWT coefficients. Afterward, summation of residuals which are coming from 100 step ahead forecasts of each ARIMA(1, 1, 1) fitted scales, is -178,47.

ARIMA models are chosen according to AIC values for each scale level of MODWT. Then ARIMA(5, 1, 5), ARIMA(2, 0, 4) and ARIMA(0, 0, 5) models were fitted to smooth part, second detail part and first detail part respectively. For each fitted model, again 100 step ahead forecasts were found. Sum of residuals is 65,62 at this time. We can easily say that for model fitting and forecasting, using wavelet based ARIMA models is better than only applying ARIMA models according to related empirical results.

4.4.2 GARCH Methods

In Q-Q plots given in Section 4.1, we have seen that student-t distribution is more appropriate for S&P500 data. It is better to use skewed student-t distribution while data is fitted to GARCH model. Using skewed generalized error distribution gives also better results. In Table 4.8, GARCH models are fitted to return data which is multiplied by 100 and removed from trend. According to information criterion statistics, AIC and SIC say that GARCH(4, 1) model with skewed generalized error distribution is better than other models. BIC and HQIC suggest that GARCH(1, 1) model with skewed generalized error distribution is better than others.

In Figure 4.62, conditional standard deviation, 2 conditional standard deviation superimposed with return series, standardized residuals and it's ACF, ACF of squared standardized residuals and Q-Q plot are given. It is seen that residuals are not correlated each other. In addition, distribution of model is almost fit with used data.

In respect of information criterion statistics, ACF of residuals and Q-Q plot, GARCH(4, 1) can be used with generalized error distribution for volatility modeling.

4.4.3 Other Methods for Time Series Analysis

AAR (*Additive nonlinear autoregressive model*)

Nonparametric additive autoregressive model is given as,

$$x_{t+s} = \mu + \sum_{j=1}^m s_j (x_{t-(j-1)d}) \quad (4.1)$$

where s_j is nonparametric univariate function of lagged time series values that are described by cubic regression splines [75].

Optimal time delay and embedding dimension for S&P500 can be found by average mutual information and by false nearest neighbors with respect to optimal time delay respectively [19, 3, 47]. Results are given for embedding dimensions 3, 4, 10 and time delay 1 fitted to S&P500 daily closing prices. Prediction is done for 100 steps between observations 5345 and 5444. Accuracy results with given dimensions and time delays, are given in Table 4.9.

LSTAR (*Logistic Smooth Transition AutoRegressive Model*)

In Subsection 2.7.1, mathematical definitions are given. In order to find more details one can see Franses and Dijk [25].

Model is applied for $m=4$, $d=1$ and coefficients for the lagged time series as 0,1,2,3. Training set is observations between 1 and 5344 of S&P500 daily closing prices and test set is last 100 observations. Values of the accuracy measures are given in Table 4.9.

In Table 4.9, it is seen that LSTAR gives smaller error values than $\text{SARIMA}(0, 1, 2) \times (0, 1, 1)_{260}$ except ACF1 method. If we consider AAR model, accuracy is getting better in case embedding dimension is getting larger. AAR with $m=10$ and $d=1$, shows better accuracy results than $\text{SARIMA}(0, 1, 2) \times (0, 1, 1)_{260}$ in ME, MAE, MPE and MAPE measures.

AIC Values of Some Models

In Table 4.10, AIC values of SETAR, LSTAR, AAR and SARIMA models are given. It is seen that regime switching models and nonlinear models have much more smaller AIC values than SARIMA model which means that regime switching models and non-linear models are more appropriate for financial time series.

Table 4.5: Accuracy Values of Training Set and Test Set with respect to Related ARIMA Model

	ARIMA(0,1,2)			
	ME	RMSE	MAE	MPE
Training Set	0.2010764	12.01217	7.667793	0.01986503
Test Set	17.4081355	33.89432	28.947227	1.27504809
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7867608	0.9980797	-0.0007129374	NA
Test Set	2.1917985	3.7679214	0.8195796219	2.592994
	ARIMA(1,1,1)			
	ME	RMSE	MAE	MPE
Training Set	0.209889	12.01520	7.667983	0.02079551
Test Set	16.250759	33.29349	28.405878	1.18699826
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7866443	0.9981045	0.001995627	NA
Test Set	2.1520228	3.6974566	0.819534466	2.548832
	ARIMA(1,1,2)			
	ME	RMSE	MAE	MPE
Training Set	0.1996574	12.01178	7.66771	0.01971641
Test Set	17.5682628	33.97777	29.02159	1.28723402
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7868012	0.9980689	-0.0002997833	NA
Test Set	2.1972617	3.7776011	0.8195603581	2.599162
	ARIMA(0,1,1)			
	ME	RMSE	MAE	MPE
Training Set	0.1896198	12.03165	7.66648	0.01864417
Test Set	17.5471431	33.97076	29.01858	1.28561879
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7866316	0.9979089	0.003926237	NA
Test Set	2.1970606	3.7772090	0.819615759	2.598472
	ARIMA(1,1,0)			
	ME	RMSE	MAE	MPE
Training Set	0.1870117	12.03517	7.666008	0.01837357
Test Set	17.6074035	34.00220	29.046741	1.29020488
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7864954	0.9978474	-0.004347022	NA
Test Set	2.1991296	3.7808746	0.819618423	2.600783
	ARIMA(2,1,0)			
	ME	RMSE	MAE	MPE
Training Set	0.1985839	12.01202	7.667259	0.01960365
Test Set	17.6814261	34.03736	29.075111	1.29584538
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7867622	0.9980102	-7.238344e-05	NA
Test Set	2.2011958	3.7845674	8.195789e-01	2.603537
	ARIMA(2,1,1)			
	ME	RMSE	MAE	MPE
Training Set	0.1984932	12.01200	7.66724	0.01959421
Test Set	17.6907784	34.04224	29.07942	1.29655716
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7867626	0.9980078	-0.0001238208	NA
Test Set	2.2015119	3.7851279	0.8195779808	2.603898
	ARIMA(1,1,2)			
	ME	RMSE	MAE	MPE
Training Set	0.199948	12.01191	7.667711	0.01974665
Test Set	17.541211	33.96376	29.009330	1.28517509
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7867878	0.9980691	-0.000125071	NA
Test Set	2.1963617	3.7760051	0.819571858	2.598118

Table 4.6: Accuracy Values of Training Set and Test Set with respect to Related SARIMA Model

	ARIMA(0, 1, 1) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.001992195	13.10853	8.43074	-0.001456642
Test Set	214.983147651	241.74554	216.14151	16.354929617
	MAPE	MASE	ACF1	Theil's U
Training Set	0.8667416	1.097389	0.1431508	NA
Test Set	16.4464264	28.134101	0.9694050	18.65588
	ARIMA(1, 1, 0) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.0004665282	12.03372	7.656206	0.003001934
Test Set	8.7373979494	31.78757	26.713677	0.613206074
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7856908	0.9965715	-0.004285541	NA
Test Set	2.0340338	3.4771908	0.828029619	2.45375
	ARIMA(1, 1, 1) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	0.04554281	12.05698	7.672398	0.007762099
Test Set	-46.39757939	67.07280	50.462464	-3.592632772
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7869763	0.9986791	-0.04925805	NA
Test Set	3.8949318	6.5684563	0.90099761	5.27051
	ARIMA(0, 1, 2) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.0007803767	12.01051	7.655735	0.003270706
Test Set	8.5058085779	31.72167	26.633174	0.595588306
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7857506	0.9965102	-0.0004955254	NA
Test Set	2.0282102	3.4667122	0.8279689754	2.449416
	ARIMA(2, 1, 2) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.0007205613	12.01022	7.655616	0.00322816
Test Set	8.8252603242	31.80971	26.736386	0.61989985
	MAPE	MASE	ACF1	Theil's U
Training Set	0.785791	0.9964947	-0.0001506475	NA
Test Set	2.035645	3.4801468	0.8279449836	2.455462
	ARIMA(2, 1, 0) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.0007348941	12.01038	7.655437	0.00323189
Test Set	8.7903951016	31.80013	26.725720	0.61724594
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7857633	0.9964714	2.467592e-05	NA
Test Set	2.0348795	3.4787584	8.279648e-01	2.454785
	ARIMA(2, 1, 1) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	0.05037788	12.03905	7.658431	0.008766098
Test Set	-33.73441096	56.01539	41.955653	-2.627817482
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7859909	0.9968612	-0.003247522	NA
Test Set	3.2397870	5.4611656	0.891556915	4.408117
	ARIMA(1, 1, 2) × (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.0007028143	12.01042	7.656021	0.003228572
Test Set	8.7970774785	31.80189	26.726933	0.617754967
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7858302	0.9965474	-0.0001075208	NA
Test Set	2.0349635	3.4789164	0.8279161080	2.454943

Table 4.7: Accuracy Values of Training Set and Test Set of Selected Models

	ARIMA(2,1,0)			
	ME	RMSE	MAE	MPE
Training Set	0.1985839	12.01202	7.667259	0.01960365
Test Set	17.6814261	34.03736	29.075111	1.29584538
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7867622	0.9980102	-7.238344e-05	NA
Test Set	2.2011958	3.7845674	8.195789e-01	2.603537
	ARIMA(0,1,2)			
	ME	RMSE	MAE	MPE
Training Set	0.2010764	12.01217	7.667793	0.01986503
Test Set	17.4081355	33.89432	28.947227	1.27504809
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7867608	0.9980797	-0.0007129374	NA
Test Set	2.1917985	3.7679214	0.8195796219	2.592994
	ARIMA(1,1,1)			
	ME	RMSE	MAE	MPE
Training Set	0.209889	12.01520	7.667983	0.02079551
Test Set	16.250759	33.29349	28.405878	1.18699826
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7866443	0.9981045	0.001995627	NA
Test Set	2.1520228	3.6974566	0.819534466	2.548832
	ARIMA(0, 1, 2) \times (0, 1, 1)₂₆₀			
	ME	RMSE	MAE	MPE
Training Set	-0.0007803767	12.01051	7.655735	0.003270706
Test Set	8.5058085779	31.72167	26.633174	0.595588306
	MAPE	MASE	ACF1	Theil's U
Training Set	0.7857506	0.9965102	-0.0004955254	NA
Test Set	2.0282102	3.4667122	0.8279689754	2.449416

Table 4.8: Information Criterion Statistics for Selected GARCH Models with given Distributions

	Information Criterion Statistics			
	AIC	BIC	SIC	HQIC
GARCH(1,0) with normal distribution	3.046191	3.048617	3.046191	3.047038
GARCH(1,0) with skewed student-t distribution	2.859012	2.863864	2.859011	2.860705
GARCH(4,0) with skewed student-t distribution	2.747997	2.756488	2.747994	2.750960
GARCH(4,1) with skewed student-t distribution	2.677922	2.687625	2.677917	2.681308
GARCH(4,1) with skewed generalized error distribution	2.675907	2.685610	2.675902	2.679293
GARCH(1,1) with skewed student-t distribution	2.679437	2.685501	2.679435	2.681553
GARCH(1,1) with skewed generalized error distribution	2.677056	2.683120	2.677054	2.679172

Table 4.9: Accuracy Values of Test Set of AAR and LSTAR Models

	AAR with m=3 and d=1			
	ME	RMSE	MAE	MPE
Test Set	-3.821709	34.07534	26.95399	-0.3453562
	MAPE	ACF1	Theil's U	
Test Set	2.070063	0.8479687	2.667273	
	AAR with m=4 and d=1			
	ME	RMSE	MAE	MPE
Test Set	-3.467193	33.67968	26.6773	-0.3181473
	MAPE	ACF1	Theil's U	
Test Set	2.048477	0.8459106	2.635209	
	AAR with m=10 and d=1			
	ME	RMSE	MAE	MPE
Test Set	-3.667028	32.5765	25.64558	-0.3322671
	MAPE	ACF1	Theil's U	
Test Set	1.969833	0.8395296	2.550718	
	LSTAR with m=4 and d=1			
	ME	RMSE	MAE	MPE
Test Set	6.319223	31.05532	25.85419	0.429437
	MAPE	ACF1	Theil's U	
Test Set	1.971797	0.8280158	2.403215	

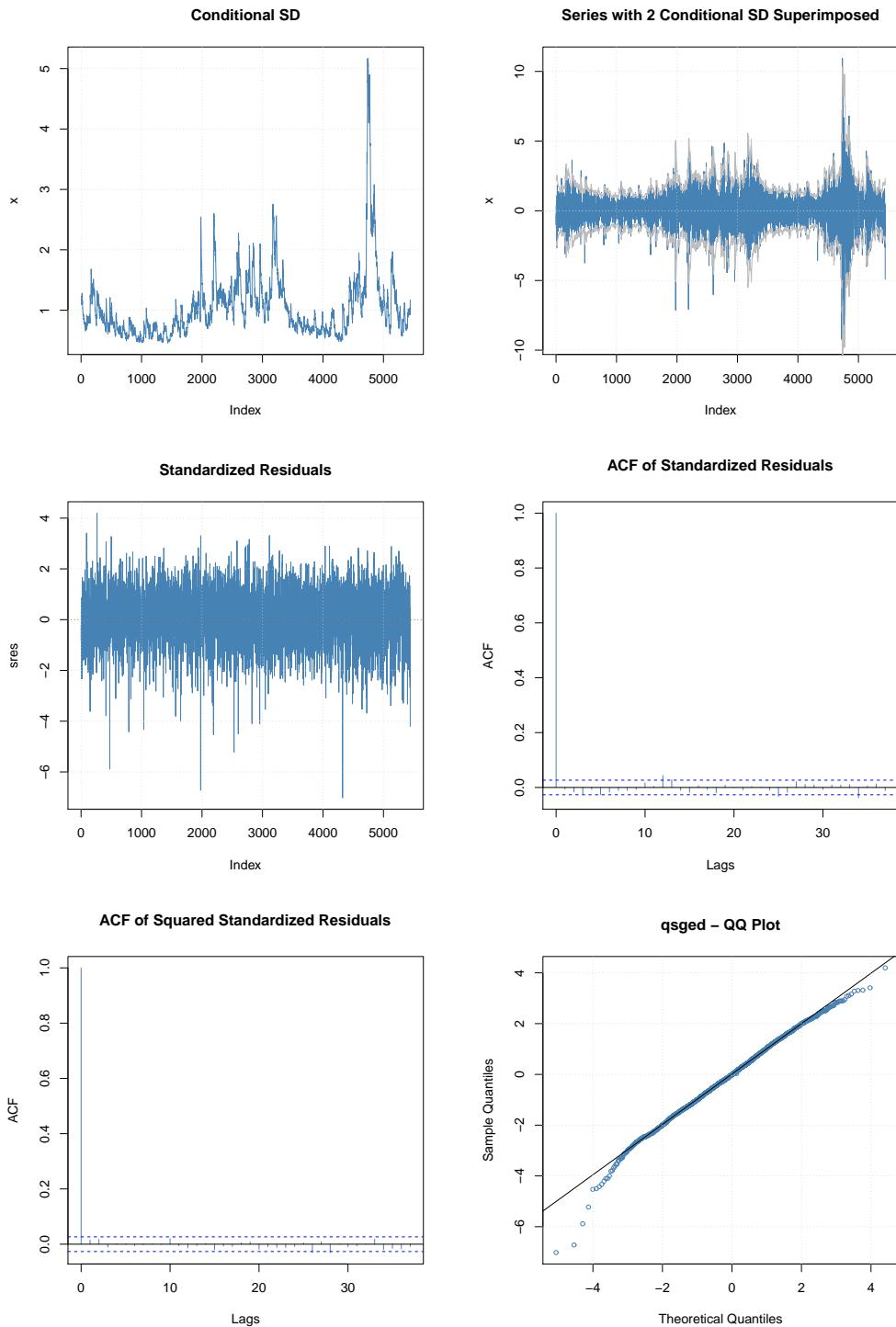


Figure 4.62: Plots According to GARCH(4, 1) Fitting with Skewed Generalized Error Distribution.

Table 4.10: AIC Values of Given Models

	SETAR ($m = 4, d = 1$)	LSTAR ($m = 4, d = 1$)	AAR ($m = 4$)	SARIMA ($0, 1, 2 \times (0, 1, 1)_{260}$)
AIC	27075.92	27078.05	27097.23	42536.88

CHAPTER 5

CONCLUSION AND OUTLOOK

In this thesis, time series concept is handled together with definitions as well as their extensions. Linear time series models, volatility and other models are explained through selected examples. Time series analysis is a really important issue for everyone to understand the real world issues as well as to predict future behaviors of systems in many areas of interest. However, usually time series analysis that includes only time domain is inadequate in general. Therefore, frequency domain analysis comes into prominence when time domain analysis is weak to catch some important characteristic properties. At this stage, Fourier and wavelet transforms come into play and are covered in details with definitions and carefully chosen examples.

Study of empirical results are handled to reach some decisions and conclusions. Time series of S&P500 daily closing prices, its logarithmic first difference as well as its first difference, are used in within the scope of this study. According to the recurrence plots, daily closing prices behave like a Wiener process (Brownian motion) and the values for the returns behave like Gaussian white noise. There are non-stationary parts in recurrence plots of daily closing prices while recurrence of returns shows stationary characteristics. ACF plots and results of Augmented Dickey-Fuller as well as KPSS tests support the results of recurrence plots: daily closing prices are non-stationary, but return values are. Moreover, Shapiro-Wilk and Kolmogorov-Smirnov tests results in that none of daily closing prices data and returns are normally distributed. As a matter of fact, Student-t distribution is the most appropriate one for returns (Figure 4.6).

On the other hand, for frequency domain analysis, we have seen that the use of wavelet transform produces better results than Fourier transform does. Smoothed raw periodograms and AR fitted spectrums in fact catch some periodicity and OLS and Burg methods used in AR fitted spectrum have shown peaks at some frequencies. These peaks give periodicities in terms of days as, such as, 13.4865, 9.7843, 7.4478, 6.0485, 4.4554, 3.9447, 3.2090, 2.7194 and 2.0493. Largest period is approximately 14 days which is far from low frequency ones. Higher frequency components overlap low frequency ones. Therefore, non-stationary time series, like financial time series in our case, are not suitable candidates for Fourier transformed analysis. Even further, time information is lost in applying Fourier transforms. STFT, which is a modified version of DFT, however, provides us with the time information; but still, it does not contain multiresolution analysis. In order to cover such drawbacks of Fourier transform, wavelet transform was developed.

It is found from the wavelet analysis that series of S&P500 daily closing prices shows a quasi-periodic behavior after the scale level 5. This means that for approximately 2 weeks, 1 month, 3 months, 6 months and for higher scales, daily closing prices have fractal structures [44]. Time-scale decomposition catches interest of both short-term and long-term traders: short-term traders are mainly interested with low scales, namely, the high frequency analysis, while the long-term traders are focused on high scales of the time series of interest. In fact, MRA of the values for the returns gives information about economic landmarks. MODWT of S&P500 returns captures the *Asian* crisis, *dot-com* collapse and *mortgage* crisis occurred in 1997, 1999 and 2007, respectively. Sizes of such fluctuations are observed in scales and in time intervals. Looking wavelet spectrums, it is found out that time series of S&P500 daily closing prices has periods of 22, 66, 260, 512 and 1024 days around *dot-com* crisis. In periods between 4 and 6 years, data has high power during high number of years. Hence data has periods in low frequency components for long years.

In order to analyse the S&P500 data with linear models, ARIMA and SARIMA models are used and their results are compared. Some periods we have found in wavelet analysis are also used in SARIMA. According to AIC, AICc and BIC, SARIMA $(0, 1, 2) \times (0, 1, 1)_{130}$ seems the best among such ARIMA and SARIMA models. Furthermore, SARIMA $(0, 1, 2) \times (0, 1, 1)_{130}$ has more accurate values for the test set than those of ARIMA and SARIMA models such that the first 5344 observations generate the training set and the last 100 observations form the test set. Evidence of long term seasonality found in wavelet transform supports that the results are accurate enough for the SARIMA $(0, 1, 2) \times (0, 1, 1)_{130}$ model. In addition, using multiresolution analysis provides a better fitting and forecasting. We have divided daily closing prices time series into 3 scales and modeled each part with different ARIMA models. Total sum of the residuals from each scale is less than the sum of the residuals which were found using only ARIMA model with the original data.

On the other hand, LSTAR model with $m = 4$ and $d = 1$ gives more accurate results, except ACF1 method. AAR model with $m = 10$ and $d = 1$ gives better results in terms of ME, MAE, MPE and MAPE measures. Besides accuracy, AIC values of SETAR, LSTAR and AAR models are much smaller than that of SARIMA $(0, 1, 2) \times (0, 1, 1)_{130}$. Likewise, values of the information criteria are compared in order to find out which GARCH model is more suitable for the volatility modeling. It has been found that GARCH(4, 1) model with skewed generalized error distribution has the smallest AIC, BIC, SIC and HQIC values. Student-t distributed GARCH(4, 1) model has very close values too. Because we have found in descriptive statistics that distribution of returns is not normal, Student-t distribution approximately fit the data.

In conclusion, financial time series have nonlinear and complex structures. Therefore, it is better to model financial time series with nonlinear models whenever possible. In addition, complex time series requires wavelet analysis instead of Fourier analysis when passing to the frequency domain. Wavelet transform provides a time domain analysis as well as a frequency domain analysis simultaneously. Particularly, SARIMA $(0, 1, 2) \times (0, 1, 1)_{130}$ model, which has better accuracy and information criteria values than other linear models, is most appropriate model for the S&P500 daily closing prices. Furthermore, it should always be preferable that modeling after MRA

gives better fitted models, especially for forecasting the time series. However, non-linear models fit better than linear ones do for the financial time series. For instance, AAR model with $m = 10$ and $d = 1$ has better values than other fitted models in our study. Thus it would be better to use nonlinear methods for modeling financial time series, such as S&P500 data.

Shortly, some of possible future works on financial time series would include

- multivariate time series analysis by using wavelets,
- applications of wavelets to ODEs and PDEs by using chaotic time series [21, 48],
- wavelet-based multi-fractal analysis, Hybrid forecasting models [29, 65].

REFERENCES

- [1] F. Ababneh, S. A. Wadi, and M. T. Ismail, Haar and Daubechies wavelet methods in modeling banking sector, International Mathematical Forum, 8(12), pp. 551–566, 2013.
- [2] T. Acharya and P. S. Tsai, *Source Coding Algorithms, in JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*, John Wiley & Sons, Inc., January 2005, ISBN 0-471-48422-9.
- [3] P. M. Addo, M. Billio, and D. Guegan, Nonlinear dynamics and recurrence plots for detecting financial crisis, The North American Journal of Economics and Finance, 26, pp. 416–435, December 2013.
- [4] J. Almerud, *World Business Cycles over Time and Scale, A study of their existence and causes*, Master's thesis, Lund University, January 2010.
- [5] C. Aloui and D. K. Nguyen, On the detection of extreme movements and persistent behavior in Mediterranean stock markets: a wavelet-based approach, IPAG Business School, Working Paper Series, 66, January 2014.
- [6] F. N. G. Andersson, *Wavelet Analysis of Economic Time Series*, Lund University, 2008.
- [7] D. Anevski, Riemann-Stieltjes integrals, Technical report, Mathematical Sciences, Lund University, October 07.01.2015.
- [8] E. M. Azoff, *Neural Network Time Series Forecasting of Financial Markets*, John Wiley & Sons, Inc., 1994, ISBN 0471943568.
- [9] P. P. Balestrassi, E. Popova, A. P. Paiva, and J. W. M. Lima, Design of experiments on neural network's training for nonlinear time series forecasting, Neurocomputing, 72, pp. 1160–1178, 2009.
- [10] J. B. J. baron Fourier, *Théorie analytique de la chaleur*, Chez Firmin Didot, père et fils, 1822.
- [11] J. A. Bastos and J. Caiado, Recurrence quantification analysis of global stock markets, Physica A, 390(7), pp. 1315–1325, 2011.
- [12] T. Bollerslev, Generalized autoregressive conditional heteroscedasticity, J.Econ., 31, pp. 307–327, 1986.
- [13] G. E. P. Box and G. M. Jenkins, *Time Series Analysis Forecasting and Control*, Holden-Day, Inc., 1976, ISBN 978-0-470-27284-8.

- [14] B. Burke, The mathematical microscope: waves, wavelets, and beyond, *Scientific Discovery at the Frontier*, chapter 7, pp. 196–235, 1994.
- [15] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to wavelets and wavelet transforms*, Prentice-Hall, Inc., 1998, ISBN 978-0134896007.
- [16] S. Butler, Notes from trigonometry, Technical report, Department of Mathematics, Iowa State University, 20.12.2014.
- [17] C. A. Cabrelli and U. M. Molter, Wavelet transform of the dilation equation, *Journal of the Australian Mathematical Society*, 37(4), pp. 474–489, 1989.
- [18] C. Chatfield, *Time-Series Forecasting*, Chapman & Hall/CRC, 2000, ISBN 1-58488-063-5.
- [19] T. Conradié, *Modelling of Nonlinear Dynamic systems: Using surrogate data methods*, Master's thesis, University of Stellenbosch, 2000.
- [20] P. M. Crowley, An intuitive guide to wavelets for economists, Bank of Finland Research Discussion Paper, 1, 2005.
- [21] W. Dahmen, Wavelet methods for pdes — some recent developments, *Journal of Computational and Applied Mathematics*, 128(2), pp. 133–185, March 2001.
- [22] J. Danielsson, *Financial Risk Forecasting*, John Wiley & Sons, Inc., 2011, ISBN 978-0-470-66943-3.
- [23] R. F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, *Econometrica*, 50, pp. 987–1007, 1982.
- [24] A. Fabretti and M. Ausloos, Recurrence plot and recurrence quantification analysis techniques for detecting a critical regime. Examples from financial market indices, *International Journal of Modern Physics C*, 16(5), pp. 671–706, 2011.
- [25] P. H. Franses and D. van Dijk, *Financial Risk Forecasting*, Cambridge University Press, 2003, ISBN 0 521 77041 6.
- [26] P. Z. Fryzlewicz, *Wavelet Techniques for Time Series and Poisson Data*, Master's thesis, The University of Bristol, September 2003.
- [27] B. R. Gelbaum and J. M. H. Olmsted, *Counterexamples in Analysis*, Dover Publications, Inc., 1992.
- [28] R. Gençay, F. Selçuk, and B. Whitcher, *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*, Academic Press, 2001, ISBN 978-0122796708.
- [29] K. Gol and Y. R. Suk, Hybrid forecasting of exchange rate by using chaos wavelet SVM-Markov model and grey relation degree, *CoRR*, abs/1207.1547, 2012.
- [30] D. Guegan and K. Houmyyia, De-noising with wavelets method in chaotic time series: application in climatology, energy and finance, in *Proceedings of SPIE*, pp. 174–185, October 2005.

- [31] J. D. Hamilton, State-space models, handbook of econometrics, Elsevier Science B.V, 4, pp. 3041–3080, 1994.
- [32] J. D. Hamilton, *Time Series Analysis*, Princeton University Press, 1994, ISBN 0-691-04289-6.
- [33] M. Hazewinkel, *Encyclopaedia of Mathematics: Coproduct—Hausdorff—Young Inequalities*, Springer, 2013, ISBN 1489937951, 9781489937957.
- [34] R. J. Hyndman and Y. Khandakar, The admissible parameter space for exponential smoothing models, *Annals of Statistical Mathematics*, 60(2), pp. 407–426, 2008.
- [35] R. J. Hyndman and Y. Khandakar, Automatic time series forecasting: The forecast package for R, *Journal of Statistical Software*, 27(3), July 2008.
- [36] R. J. Hyndman and A. B. Koehler, Another look at measures of forecast accuracy, *International Journal of Forecasting*, 22(4), pp. 679–688, 2006.
- [37] R. J. Hyndman, A. B. Koehler, J. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing, The State Space Approach*, Springer-Verlag, 2008, ISBN 978-3-540-71916-8.
- [38] W. Jiang, *Modeling and predicting of different stock markets with GARCH model*, Master's thesis, Uppsala University, June 2012.
- [39] Y. Liu, X. S. Liang, and R. H. Weisberg, Rectification of the bias in the wavelet power spectrum, *Journal of Atmospheric and Oceanic Technology*, 24, pp. 2093–2102, December 2007.
- [40] C. Lundberg, *Time scale Structure in the Cross Section of Portfolio Returns: A Wavelet-based Approach*, Ph.D. thesis, University of Washington, October 2013.
- [41] A. A. Lyubushin, Multidimensional wavelet analysis of geophysical monitoring time series, *Izvestiya, Physics of the Solid Earth*, 37(6), pp. 41–51, 2001.
- [42] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting Methods and Applications*, Wiley&Sons, Inc., 1998.
- [43] S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 674–693, 1989.
- [44] B. B. Mandelbrot, *Fractals and Scaling in Finance: Discontinuity, Concentration, Risk*, Springer, 1997, ISBN 0-387-98363.
- [45] P. S. Mashikian, *Multiresolution Models of Financial Time Series*, Master's thesis, Massachusetts Institute of Technology, June 1997.
- [46] P. Masset, Analysis of financial time-series using Fourier and wavelet methods, University of Fribourg, 2008.

- [47] E. S. Mayfield and B. Mizrach, On determining the dimension of real-time stock-price data, American Statistical Association, *Journal of Business & Economic Statistics*, 10(3), pp. 416–435, July 1992.
- [48] J. S. Murguía and E. Campos-Cantón, Wavelet analysis of chaotic time series, *Revista Mexicana de Física*, 52(2), pp. 155–162, April 2006.
- [49] M. Nouri, A. R. Oryoie, and S. Fallahi, Forecasting gold return using wavelet analysis, *World Applied Sciences Journal*, 19(2), pp. 276–280, 2012.
- [50] B. Pecar, The use of visual recurrence analysis and hurst exponents as qualitative tools for analysing financial time series, in *Proceedings of the 23rd SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, Cambridge, 2003.
- [51] B. Pecar, Visual recurrence analysis as an alternative framework for time series characterisation, *Computer Finance and its Applications*, pp. 241–252, 2004.
- [52] D. Percival, An introduction to the wavelet analysis of time series, Technical report, University of Washington, 2000.
- [53] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*, Cambridge University Press, 2000.
- [54] T. R., Wavelet analysis for financial market data, Technical report, The Numerical Algorithms Group (NAG), 14.10.2014.
- [55] S. M. Raihan, Y. Wen, and B. Zeng, Wavelet: A new tool for business cycle analysis, Working Paper, FEDERAL RESERVE BANK OF ST. LOUIS, <http://research.stlouisfed.org/wp/2005/2005-050.pdf>, 050-A, 2005.
- [56] J. B. Ramsey, An analysis of U.S. stock price behavior using wavelets, *Economic Research Reports*, 6, February 1994.
- [57] J. B. Ramsey, Wavelets in economics and finance: Past and future, *Studies in Nonlinear Dynamics & Econometrics*, 6(3), November 2002.
- [58] T. Rocha, S. Paredes, P. Carvalho, J. Henriques, and M. Harris, Wavelet based time series forecast with application to acute hypotensive episodes prediction, 32nd Annual International Conference of the IEEE EMBS, pp. 2403–2406, 2010.
- [59] A. Schuster, On the investigation of hidden periodicities, *Journal of Geophysical Research*, 3(1), pp. 13–41, March 1898.
- [60] M. J. Seiler and W. Rom, A historical analysis of market efficiency: Do historical returns follow a random walk?, *Journal Of Financial And Strategic Decisions*, 10(2), pp. 49–57, 1997.
- [61] T. Setz, *Wavelet Analysis on Stochastic Time Series, A visual introduction with an examination of long term financial time series*, Master's thesis, ETH Zürich, 2011.

- [62] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, Springer, 2010, ISBN 978-1-4419-7864-6.
- [63] H. Steehouwer, A frequency domain methodology for time series modelling, OFRC Methodological Paper 2008-02, ORTEC Finance, 2008.
- [64] G. Strang, Wavelets and dilation equations: A brief introduction, *SIAM Review*, 31(4), pp. 614–627, December 1989.
- [65] Y. Sul, S. Yao-yuan, and S. Heng, Stock market time series prediction method based on SVM and wavelet, *Science Technology and Engineering*, 8(12), pp. 1371–1374, 2008.
- [66] J. W. Taylor, Smooth transition exponential smoothing, *Journal of Forecasting*, 23, pp. 385–394, 2004.
- [67] C. Torrence and G. P. Compo, A practical guide to wavelet analysis, *Bulletin of the American Meteorological Society*, 79, pp. 61–78, 1998.
- [68] R. S. Tsay, *Analysis of Financial Time Series*, John Wiley & Sons, Inc., 2005, ISBN 978-0-471-74618-8.
- [69] C. Valens, A really friendly guide to wavelets, Technical report, The University of New Mexico, 08.11.2014.
- [70] B. Videkovic, Basics of wavelets, 12.12.2014, <http://www2.isye.gatech.edu/brani/isyebayes/bank/handout20.pdf>.
- [71] T. A. Vuorenmaa, *A Multiresolution Analysis of Stock Market Volatility Using Wavelet Methodology*, Master's thesis, Universtiyy of Helsinki, September 2004.
- [72] S. A. Wadi, M. T. Ismail, M. H. Alkhahazaleh, and S. A. A. Karim, Selecting wavelet transforms model in forecasting financial time series data based on arima model, *Applied Mathematical Sciences*, 5(7), pp. 315–326, 2011.
- [73] A. T. Walden, Wavelet analysis of discrete time series, 3rd European Congress of Mathematics (3ECM), 2, pp. 627–641, July 2001.
- [74] A. T. Walden, Statistical computational of volatility in financial time series data, *International Science Index*, 4(2), pp. 663–667, 2010.
- [75] S. N. Wood and N. H. Augustin, Gams with integrated model selection using penalized regression splines and applications to environmental modelling, *Eco-logical Modelling*, 157(3), pp. 157–177, 2002.
- [76] S. Yousefi, I. Weinreich, and D. Reinarz, Wavelet-based prediction of oil prices, *Chaos, Solitons and Fractals*, 25, pp. 265–275, 2005.
- [77] E. Zivot, Nonlinear time series models, 15.03.2015, <http://faculty.washington.edu/ezivot/econ584/notes/nonlinear.pdf>.

Appendix A

Some of the *R*-Codes

Simulated ARMA Codes of Section 2.3

```
# AR(1) Processes
x1 = arima.sim(list(order=c(1,0,0), ar=.8), n=1000)
x2 = arima.sim(list(order=c(1,0,0), ar=-.8), n=1000)
par(mfrow=c(2,1))
plot(x1, main=(expression(AR(1)~~~phi==+.8)))
plot(x2, main=(expression(AR(1)~~~phi==-.8)))
dev.new()
acf2(x1, 999)
acf2(x1, 100)
acf2(x2, 100)
#AR(2) Process
x = arima.sim(list(order=c(2,0,0), ar=c(.5,.4)), n=100)
dev.new()
plot(x,main=(expression(AR(2)~~~phi[1]==.5~~~phi[2]==.4)))
dev.new()
acf2(x)
#MA(1) Process
x = arima.sim(list(order=c(0,0,1), ma=.5), n=1000)
dev.new()
plot(x, main=(expression(MA(1)~~~theta==.5)))
dev.new()
acf2(x,100)
#MA(2) Process
x = arima.sim(list(order=c(0,0,2), ma=c(-.5,-.9)), n=500)
plot(x, main=(expression(MA(2)~~~theta[1]==-.5~~~
theta[2]==-.9)))
dev.new()
acf2(x,100)
#ARMA(2,2) Process
x = arima.sim(list(order=c(2,0,2), ar=c(.5,.4),ma=c(-.5,-.9)),
n=1000)
plot(x, main=(expression(ARMA(2,2)~~~phi[1]==.5~~~phi[2]==.4~~~
theta[1]==-.5~~~theta[2]==-.9)))
dev.new()
acf2(x,100)
#ARIMA(1,1,1) Process
x1 = arima.sim(list(order=c(1,1,1), ar=.9, ma=.5), n=1000)
x2 = arima.sim(list(order=c(1,1,1), ar=.5, ma=-.4), n=1000)
par(mfrow=c(2,2))
```

```

plot(x1, main=(expression(ARIMA(1,1,1)~~~phi==.9~~~
                           theta==.5)))
plot(x2, main=(expression(ARIMA(1,1,1)~~~phi==.5~~~
                           theta==-.4)))
acf(x1,100)
acf(x2,100)

```

GARCH Codes of Subsection 2.6.1

```

#plotting dax data and its returns
dax<-(EuStockMarkets)[,"DAX"]
return.dax<-diff(log(dax))
par(mfrow=c(1,2))
plot(dax)
plot(return.dax)
#summary of AR(1)-ARCH(1) of return of DAX
install.packages("fGarch")
library(fGarch)
summary(garchFit(~arma(1,0)+garch(1,0),return.dax))
summary(dax.g <- garchFit(~garch(1,1),return.dax))
#Forecast of GARCH(1,1) volatility of DAX return
u = dax.g@sigma.t
plot(window(return.dax), ylim=c(-.12,.12),
     ylab="DAX Returns")
lines(window(return.dax-2*u), lty=2, col=4)
lines(window(return.dax+2*u), lty=2, col=4)

```

Neural Network Codes of Subsection 2.7.2

```

library(forecast)
data(EuStockMarkets)
dax<-EuStockMarkets[, "DAX"]
a<- window(dax, start = c(1991, 130), end =
            c(1994, 130))
fit <- nnetar(a)
fore<-forecast(fit,h=260)
plot(fore)
dev.new()
plot(window(dax, start = c(1991, 130), end =
            c(1995, 130)))
accuracy(fore,window(dax, start = c(1994, 130),
                      end = c(1995, 130)))

```

Spectral Analysis Codes of Subsection 3.1.4

```

#Spectral Density of Noise and ARMA
par(mfrow=c(3,1))
arma.spec(log="no", main="White Noise")
arma.spec(ma=.5, log="no", main="Moving Average")
arma.spec(ar=c(-.4,.3), log="no", main="Autoregression")
arma.spec(ar=c(-.4,.3),ma=.5, log="no", main="Autoregression")
#Example Sine Series
x1<-3*sin(2*pi*1:100*5/100)+4*cos(2*pi*1:100*5/100)
x2<-2*sin(2*pi*1:100*25/100)+3*cos(2*pi*1:100*25/100)

```

```

x3<-4*sin(2*pi*1:100*85/100)+6*cos(2*pi*1:100*85/100)
sum=x1+x2+x3
par(mfrow=c(2,2))
plot.ts(x1, ylim=c(-10,10), main=expression(frequency
==5/100~~~A^2==25))
plot.ts(x2, ylim=c(-10,10), main=expression(frequency
==25/100~~~A^2==13))
plot.ts(x3, ylim=c(-10,10), main=expression(frequency
==85/100~~~A^2==52))
plot.ts(sum, ylim=c(-20,20), main="sum")
P = abs(2*fft(sum)/100)^2
f = 0:99/100
plot(f, P, type="l", xlab="frequency", ylab="periodogram")
x3<-4*sin(2*pi*1:100*75/100)+6*cos(2*pi*1:100*75/100)
sum=x1+x2+x3
par(mfrow=c(1,2))
P = abs(2*fft(sum)/100)^2
f = 0:99/100
plot(f, P, type="l", xlab="frequency", ylab="periodogram")
x3<-4*sin(2*pi*1:100*50/100)+6*cos(2*pi*1:100*50/100)
sum=x1+x2+x3
#periodogram of sum series
P = abs(2*fft(sum)/100)^2
f = 0:99/100
plot(f, P, type="l", xlab="frequency", ylab="periodogram")

```

DAX Example Codes of Subsection 3.1.4

```

#some descriptive analysis
dax<-(EuStockMarkets)[,"DAX"]
par(mfrow=c(3,2))
plot(dax)
acf(dax)
pacf(dax)
hist(dax)
qqnorm(dax)
qqline(dax)
qqplot(rt(1000,df=3), dax, main="t(3) Q-Q Plot",ylab=
      "Sample Quantiles")
qqline(dax)
par(mfrow=c(1,2))
#for comparing the histograms for different number of breakpoints
hist(dax)
hist(dax,100)
summary(dax)
kurtosis(dax)
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
Mode(dax)
#general look at spectrums of DAX
DAX<-ts(EuStockMarkets[0:1860],frequency=260,
         start=c(1991,130))

```

```

par(mfrow=c(2,2))
plot(DAX)
spec.pgram(DAX, taper=0, log="no")
spectrum(DAX)
spectrum(DAX,method="ar")
spectrum(DAX,method="ar",na.action=na.pass)
spec.ar(DAX, plot=TRUE, method = "mle",
        add = TRUE,col = "forest green")
spec.ar(DAX, plot=TRUE, method = "ols",
        add = TRUE,col = "blue")
spec.ar(DAX, plot=TRUE, method = "burg",
        add = TRUE,col = "red")
legend("topleft",c("YuleWalker","MLE","OLS","Burg"),
       lty=c(1,1,1,1),
       lwd=c(2.5,2.5,2.5,2.5),col=c("black",
                                      "forest green","blue","red"))
#spanned smooth periodograms(Daniell)
par(mfrow=c(2,2))
spectrum(DAX)
spectrum(DAX, spans = c(3,5))
text(100,1e+05,"widths(3,5)",col="blue")
spectrum(DAX, spans = c(5,7))
text(100,1e+05,"widths(5,7)",col="blue")
spectrum(DAX, spans = c(19,23))
text(100,10000,"widths(19,23)",col="blue")
#kernel smooth periodograms(Daniell)
par(mfrow=c(2,2))
spectrum(DAX)
spectrum(DAX,kernel("daniell", c(3,5)))
text(100,10000,"widths(3,5)",col="blue")
spectrum(DAX,kernel("daniell", c(5,7)))
text(100,10000,"widths(5,7)",col="blue")
spectrum(DAX,kernel("daniell", c(19,23)))
text(100,10000,"widths(19,23)",col="blue")

```

Empirical Results Codes of Chapter 4

```

#getting data form yahoo web page
str <- sprintf ("%s?s=%GSPC&d=7&e=4&f=2011&g=d&a=0&b=3&c=1950",
                "http://ichart.finance.yahoo.com/table.csv")
df <- tryCatch(read.csv(url(str)), error = function(e) NA)
names(df) <- tolower(names(df))
df$date <- as.Date(df$date)
df <- df[order(df$date), ]
start.date <- "1990-01-01"
end.date <- "2011-01-01"
close.data<-ts(subset(df,date>=start.date|date>=end.date)
               [,c("close")])
df$return <- c(diff(log(df$close)), NA)
return.data<-ts(subset(df,date>=start.date|date>=end.date)
                [,c("return")])
par(mfrow=c(1,2))
plot(subset(df,date>=start.date|date>=end.date)
      [,c("date","close")],

```

```

    type="l",main="S&P 500", xlab="", col="tomato")
plot(subset(df,date>=start.date|date>=end.date)
      [,c("date","return")],
    type="l",main="S&P 500 Return", xlab="", col="blue")
#recurrence plots to see characteristics of time series
install.packages("tseriesChaos")
recurr(close.data,m=12,d=1)
recurr(close.data,m=60,d=2)
recurr(return.data,m=12,d=1)
recurr(return.data,m=60,d=2)
#histogram and Q-Q plot
par(mfrow=c(2,1))
hist(close.data, prob=TRUE, 60)
lines(density(close.data,na.rm = TRUE))
qqnorm(close.data)
qqline(close.data)
par(mfrow=c(2,1))
hist(return.data, prob=TRUE, 60)
lines(density(return.data,na.rm = TRUE))
qqnorm(return.data)
qqline(return.data)
library(car)
par(mfrow=c(2,2))
qqPlot(return.data, distribution="norm",
       ylab="S&P500 quantiles",
       envelope=FALSE)
qqPlot(return.data, distribution="lnorm",
       ylab="S&P500 quantiles",
       envelope=FALSE)
qqPlot(return.data, distribution="unif",
       ylab="S&P500 quantiles",
       envelope=FALSE)
qqPlot(return.data, distribution="t", df=5,
       ylab="S&P500 quantiles",
       envelope=FALSE)
#Shapiro-Wilk normality test
shapiro.test(close.data[0:4999])
shapiro.test(return.data[0:4999])
shapiro.test(diff(close.data[0:4999]))
#Kolmogorov-Smirnov test
ks.test(close.data, "pnorm", mean(close.data),
        sd(close.data))
ks.test(return.data, "pnorm", mean(return.data),
        sd(return.data))
ks.test(diff(close.data), "pnorm", mean(diff(close.data)),
        sd(diff(close.data)))
#for looking the lag-lag plot fill the missing values with
#average values and get the lag plot
mean(close.data, na.rm=T)
#for missing values
close.data[is.na(close.data)]<-mean(close.data,
                                       na.rm=T)
lag1.plot(close.data, 9)

```

```

mean(return.data, na.rm=T)
return.data[is.na(return.data)]<-mean(return.data,
                                         na.rm=T)
lag1.plot(return.data, 9)
#Abs(return) and squared return
par(mfrow=c(3,1))
plot(return.data)
plot(abs(return.data))
plot(return.data^2)
dev.new()
par(mfrow=c(1,2))
acf(abs(return.data))
acf(return.data^2)
#Acf and Pacf analysis
dev.new()
par(mfrow=c(2,2))
acf(close.data,40)
pacf(close.data,40)
acf(return.data,40)
pacf(return.data,40)
acf2(diff(return.data),40)
#Big picture of the data with frequency 1 and frequency 260
acf2(close.data,5566)
freq.close<-ts(close.data,frequency=260,
                 start=c(1990,1))
acf2(freq.close, 5566)
acf2(return.data,5566)
freq.return<-ts(return.data,frequency=260,
                  start=c(1990,1))
acf2(freq.return, 5566)
acf2(diff(return.data),5565)
#descriptive statistics
library(moments)
close.stats <- c(mean(close.data), sd(close.data),
                 skewness(close.data), kurtosis(close.data))
names(close.stats) <- statNames
close.stats
return.stats <- c(mean(return.data),
                  sd(return.data), skewness(return.data),
                  kurtosis(return.data))
names(return.stats) <- statNames
return.stats
log.close.stats <- c(mean(log(close.data)),
                     sd(log(close.data)), skewness(log(close.data)),
                     kurtosis(log(close.data)))
names(log.close.stats) <- statNames
log.close.stats
diff.close.stats <- c(mean(diff(close.data)),
                     sd(diff(close.data)), skewness(diff(close.data)),
                     kurtosis(diff(close.data)))
names(diff.close.stats) <- statNames
diff.close.stats
diff.log.close.stats <- c(mean(diff(log(close.data))),
```

```

sd(diff(log(close.data))), skewness(diff(log(close.data))),
kurtosis(diff(log(close.data))))
names(diff.log.close.stats) <- statNames
diff.log.close.stats
diff.diff.close.stats <- c(mean(diff(diff(close.data))),
sd(diff(diff(close.data))), skewness(diff(diff(close.data))),
kurtosis(diff(diff(close.data)))))
names(diff.diff.close.stats) <- statNames
diff.diff.close.stats
diff.return.stats <- c(mean(diff(return.data)),
sd(diff(return.data)), skewness(diff(return.data)),
kurtosis(diff(return.data)))
names(diff.return.stats) <- statNames
diff.return.stats
#tests
adf.test(close.data, alternative="stationary")
adf.test(return.data, alternative="stationary")
library(tseries)
kpss.test(close.data, null = "Trend")
kpss.test(return.data, null = "Trend")
#linear filtering
plot(close.data, type = "l")
tui.1 <- filter(close.data, filter = rep(1/5, 5))
tui.2 <- filter(close.data, filter = rep(1/25, 25))
tui.3 <- filter(close.data, filter = rep(1/131, 131))
lines(tui.1, col = "red")
lines(tui.2, col = "purple")
lines(tui.3, col = "blue")
#decomposition
plot(decom <- stl(log(freq.close), "per")) #Loess method
plot(decom2 <- decompose(log(freq.close))) #MA method
decomposed.data <- decompose(freq.close,
                               type="multiplicative")
plot(freq.close - decomposed.data$trend,
     main="signal without trend component")
#Exponential smoothing
model <- HoltWinters(freq.close)
plot(freq.close)
lines(model$fitted[, "xhat"], col="red")
pred <- predict(model, n.ahead=200)
plot(freq.close)
lines(pred, col="red", lty=2)
plot(forecast(model, h=200, level=c(75,95)))
dev.new()
fit <- stlf(freq.close)
plot(forecast(fit, level=c(75,95)))
summary(fit)
#Fourier analysis
par(mfrow=c(2,2))
plot(subset(df, date >= start.date)[ , c("date", "close")],
     type="l",main="S&P 500", xlab="", col="tomato")
mtext(sprintf("Closing prices since %s", start.date))
spectrum(close.data,na.action=na.pass)

```

```

spectrum(close.data,kernel("daniell", c(10,20)),
         na.action=na.pass)
spectrum(close.data,method="ar",na.action=na.pass)
spec.ar(close.data, plot=TRUE, method = "mle",
        add = TRUE,col = "forest green")
spec.ar(close.data, plot=TRUE, method = "ols",
        add = TRUE,col = "blue")
spec.ar(close.data, plot=TRUE, method = "burg",
        add = TRUE,col = "red")
legend("topleft",c("YuleWalker","MLE","OLS","Burg"),
       lty=c(1,1,1,1),lwd=c(2.5,2.5,2.5,2.5),
       col=c("black","forest green","blue","red"))
#smoothed periodograms of daily closing prices
par(mfrow=c(2,2))
specvalues1 <- spec.pgram(close.data,
                           kernel("daniell", c(13,13)),taper=0)
text(0.4,100000,"widths(13,13)",col="blue")
specvalues2 <- spec.pgram(close.data,
                           kernel("daniell", c(21,21)),taper=0)
text(0.4,100000,"widths(21,21)",col="blue")
specvalues3 <- spec.pgram(close.data,
                           kernel("daniell", c(13,41)),taper=0)
text(0.4,100000,"widths(13,41)",col="blue")
specvalues <- spec.pgram(close.data,taper=0, log="no")
text(0.4,40000000,"lag=no",col="blue")
#peaks at periodogram
spectrum(close.data,method="ar",na.action=na.pass)
a<-spec.ar(close.data, plot=TRUE, method = "burg",
            add = TRUE,col = "red")
x<-a$spec
y<-a$freq
z<-ts(x,y)
install.packages("quantmod")
findPeaks(z, thresh=0) #thresh is for minimum peak/valley threshold
legend("topleft",c("YuleWalker","Burg"),lty=c(1,1),
       lwd=c(2.5,2.5), col=c("black","red"))
a$freq
abline(v=1/13.4865, lty="dotted")
abline(v=1/9.7843, lty="dotted")
abline(v=1/7.4478, lty="dotted")
abline(v=1/6.0485, lty="dotted")
abline(v=1/4.4554, lty="dotted")
abline(v=1/3.9447, lty="dotted")
abline(v=1/3.2090, lty="dotted")
abline(v=1/2.7194, lty="dotted")
abline(v=1/2.0493, lty="dotted")
#Wavelet analysis
install.packages("wavelets")
library(wavelets)
#DWT
wt <- dwt((close.data[1:5444]), boundary="reflection",
           fast=FALSE)
plot.dwt(wt, levels = list(c(1,2,3,4,5),c())),

```

```

draw.boundary = TRUE,col.boundary = "green")
wt2 <- dwt((return.data[1:5444]), boundary="reflection",
            fast=FALSE)
plot.dwt(wt2, levels=6, draw.boundary = TRUE,
          col.boundary = "green")
#MODWT
modwt.close.data<-modwt(close.data,filter="d2",
                           n.levels=8, boundary="periodic",fast=TRUE)
plot.modwt(modwt.close.data)
dev.new()
modwt.close.data2<-modwt(close.data,filter="d2",
                           n.levels=10,boundary="periodic",fast=TRUE)
plot.modwt(modwt.close.data2)
modwt.close.data3<-modwt(close.data,filter="la8",
                           n.levels=8,boundary="periodic",fast=TRUE)
plot.modwt(modwt.close.data3)
modwt.close.data4<-modwt(close.data,filter="la8",
                           n.levels=10,boundary="periodic",fast=TRUE)
plot.modwt(modwt.close.data4)
modwt.return.data<-modwt(return.data,filter="d2",
                           n.levels=8,boundary="periodic",fast=TRUE)
plot.modwt(modwt.return.data)
modwt.return.data2<-modwt(return.data,filter="la8",
                           n.levels=8,boundary="periodic",fast=TRUE)
plot.modwt(modwt.return.data2)
plot.modwt(modwt.return.data2, levels = list(c(5,6,7,8),c(8)),
            draw.boundary = TRUE,col.boundary = "green")
modwt.return.data3<-modwt(return.data,filter="la8",
                           n.levels=9,boundary="periodic",fast=TRUE)
plot.modwt(modwt.return.data3, levels = list(c(8,9),c(9)),
            draw.boundary = TRUE,col.boundary = "green")
library(wavelet)
SP500.volatility <- abs(return.data[1:5444])
SP500V.haar <- mra(SP500.volatility, "haar", 4, "modwt")
names(SP500V.haar) <- c("d1", "d2", "d3", "d4", "s4")
SP500V.la8 <- mra(SP500.volatility, "la8", 4, "modwt")
names(SP500V.la8) <- c("d1", "d2", "d3", "d4", "s4")
par(mfcol=c(6,1), pty="m", mar=c(5-2,4,4-2,2))
plot.ts(SP500.volatility, axes=FALSE, ylab="", main="(a)")
for(i in 1:5)
  plot.ts(SP500V.haar[[i]], axes=FALSE, ylab=names(SP500V.haar)[i])
axis(side=1, at=seq(0,368,by=23),
     + labels=c(0,"",46,"",92,"",138,"",184,"",230,"",276,
               ",322","",368))
par(mfcol=c(6,1), pty="m", mar=c(5-2,4,4-2,2))
plot.ts(SP500.volatility, axes=FALSE, ylab="", main="(b)")
for(i in 1:5)
  plot.ts(SP500V.la8[[i]], axes=FALSE, ylab=names(SP500V.la8)[i])
axis(side=1, at=seq(0,368,by=23),
     + labels=c(0,"",46,"",92,"",138,"",184,"",230,"",276,
               ",322","",368))
#Wavelet Power Spectrum
install.packages("WaveletComp")

```

```

library(WaveletComp)
my.data = data.frame(freq.close)
my.w = analyze.wavelet(my.data)
wt.image(my.w)
dev.new()
reconstruct(my.w, plot.waves = F, lwd = c(1,2),
            legend.coords = "bottomleft")
my.w1 = analyze.wavelet(my.data,method="shuffle")
wt.image(my.w1)
dev.new()
reconstruct(my.w1, plot.waves = F, lwd = c(1,2),
            legend.coords = "bottomleft")
my.w2 = analyze.wavelet(my.data,method="ARIMA")
wt.image(my.w2)
dev.new()
reconstruct(my.w2, plot.waves = F, lwd = c(1,2),
            legend.coords = "bottomleft")
library(biwavelet)
wt1=wt(cbind(1:5444,close.data[1:5444]))
par(mfrow=c(1,2))
plot(wt1, type="power.corr.norm",
      main="Bias-corrected wavelet power")
plot(wt1, type="power.norm",
      main="Biased wavelet power")
#SARIMA fitting
#The Box-Jenkins model
a<-pacf(close.data)
b<-acf(return.data)
par(mfrow=c(1,2))
plot(a)
plot(b)
#SARIMA models
sarima(close.data, 1, 1, 1)
sarima(return.data, 1, 1, 1)
sarima(freq.close,1,1,1,0,1,1,42)
#Autoselected models
library(forecast)
auto.arima(close.data) #gives ARIMA(0,1,2)
par(mfrow=c(1,2))
plot(close.data)
plot(arima.sim(list(order=c(0,1,2), ma=c(.0675,.0525)),
               n=5566))
#Data fitting
close.data.arima <- arima(close.data,
                           order=c(0,1,2))
close.data.arima
diff.close.data.arima <- arima(diff(close.data),
                                 order=c(0,0,2))
diff.close.data.arima
#Auto selection by bic criterion
auto.arima(close.data,ic="bic") #gives ARIMA(1,1,1)
diff.close.data.arima <- arima(diff(close.data),
                               order=c(1,0,1))

```

```

diff.close.data.arima
#Best ARIMA model by AIC
get.best.arima <- function(x.ts, maxord = c(1,1,1))
{
  best.aic <- 1e8
  n <- length(x.ts)
  for (p in 0:maxord[1]) for(d in 0:maxord[2]) for(q in 0:maxord[3])
  {
    fit <- arima(x.ts, order = c(p,d,q))
    fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
    if (fit.aic < best.aic)
    {
      best.aic <- fit.aic
      best.fit <- fit
      best.model <- c(p,d,q)
    }
    list(best.aic, best.fit, best.model)
  }
}
get.best.arima(close.data, maxord=c(2,2,2))
close.arima<-arima(close.data, ord=c(2,1,0),
                     xreg=1:length(close.data))
nobs=length(close.data)
close.pred <- predict(close.arima, n.ahead=100,
                      newxreg=(nobs+1):(nobs+100))
ts.plot(close.data,close.pred$pred, col=1:2,
         xlim=c(5000,5544),ylim=c(800,1400))
#forecasts
library("forecast")
par(mfrow=c(2,2))
close.data.arima <- arima(close.data,
                           order=c(0,1,2))
forecasts <- forecast.Arima(close.data.arima ,
                             h=500,level=c(99.5))
plot.forecast(forecasts)
diff.close.data.arima <- arima(diff(close.data),
                                order=c(0,0,2))
forecasts2 <- forecast.Arima(diff.close.data.arima ,
                             h=500,level=c(99.5))
plot.forecast(forecasts2)
close.data.arima2 <- arima(close.data,
                            order=c(1,1,1))
forecasts3 <- forecast.Arima(close.data.arima2 ,
                             h=500,level=c(99.5))
plot.forecast(forecasts3)
diff.close.data.arima2 <- arima(diff(close.data),
                                 order=c(1,0,1))
forecasts4 <- forecast.Arima(diff.close.data.arima2 ,
                             h=500,level=c(99.5))
plot.forecast(forecasts4)
dev.new()
par(mfrow=c(2,2))
acf(forecasts$residuals, lag.max=40)
acf(forecasts2$residuals, lag.max=40)

```

```

acf(forecasts3$residuals, lag.max=40)
acf(forecasts4$residuals, lag.max=40)
dev.new()
par(mfrow=c(2,2))
plot.ts(forecasts$residuals)
plot.ts(forecasts2$residuals)
plot.ts(forecasts3$residuals)
plot.ts(forecasts4$residuals)
dev.new()
par(mfrow=c(2,2))
plotForecastErrors <- function(forecasterrors)
{
  mybinsize <- IQR(forecasterrors)/4
  mysd    <- sd(forecasterrors)
  mymin   <- min(forecasterrors) - mysd*5
  mymax   <- max(forecasterrors) + mysd*3
  mynorm  <- rnorm(10000, mean=0, sd=mysd)
  mymin2 <- min(mynorm)
  mymax2 <- max(mynorm)
  if (mymin2 < mymin) { mymin <- mymin2 }
  if (mymax2 > mymax) { mymax <- mymax2 }
  mybins <- seq(mymin, mymax, mybinsize)
  hist(forecasterrors, col="red", freq=FALSE,
        breaks=mybins)
  myhist <- hist(mynorm, plot=FALSE,
                 breaks=mybins)
  points(myhist$mids, myhist$density, type="l",
         col="blue", lwd=2)
}
plotForecastErrors(forecasts$residuals)
plotForecastErrors(forecasts2$residuals)
plotForecastErrors(forecasts3$residuals)
plotForecastErrors(forecasts4$residuals)
#mean of forecast residuals
mean(forecasts$residuals)
mean(forecasts2$residuals)
mean(forecasts3$residuals)
mean(forecasts4$residuals)
#ARIMA with Drift
fit1 = arima(close.data, order=c(0,1,2))
nobs = length(close.data)
fit2 = arima(close.data, order=c(0,1,2),
             xreg=1:nobs)
fore1 = predict(fit1, 200)
fore2 = predict(fit2, 200, newxreg=(nobs+1):(nobs+200))
par(mfrow=c(2,1))
ts.plot(close.data,fore1$pred, col=1:2,
        main="Prediction Without Constant Term")
ts.plot(close.data,fore2$pred, col=1:2,
        main="Prediction With Constant Term")
par(mfrow=c(2,1))
ts.plot(close.data,fore1$pred, col=1:2,
        xlim=c(5000,5644),ylim=c(800,1400),

```

```

    main="Prediction Without Constant Term")
ts.plot(close.data,fore2$pred, col=1:2,
        xlim=c(5000,5644),ylim=c(800,1400),
        main="Prediction With Constant Term")
#predictions for sarima models with frequency 260
par(mfrow=c(2,2))
close.s1<-arima(close.data, order=c(2,1,2),
                  seas=list(order=c(0,1,1), 260))
predict<-predict(close.s1, n.ahead=300)
ts.plot(cbind(close.data, predict$pred), lty=1:2, col=1:2,
        main="ARIMA(2,1,2)x(0,1,1) with Frequency 260")
lines(predict$pred+2*predict$se, col="red", lty=3)
lines(predict$pred-2*predict$se ,col="red", lty=3)
close.s2<-arima(close.data, order=c(0,1,2),
                  seas=list(order=c(0,1,1), 260))
predict<-predict(close.s2, n.ahead=300)
ts.plot(cbind(close.data, predict$pred), lty=1:2, col=1:2,
        main="ARIMA(0,1,2)x(0,1,1) with Frequency 260")
lines(predict$pred+2*predict$se, col="red", lty=3)
lines(predict$pred-2*predict$se ,col="red", lty=3)
close.s3<-arima(close.data, order=c(2,1,0),
                  seas=list(order=c(0,1,1), 260))
predict<-predict(close.s3, n.ahead=300)
ts.plot(cbind(close.data, predict$pred), lty=1:2, col=1:2,
        main="ARIMA(2,1,0)x(0,1,1) with Frequency 260")
lines(predict$pred+2*predict$se, col="red", lty=3)
lines(predict$pred-2*predict$se ,col="red", lty=3)
close.s4<-arima(close.data, order=c(1,1,1),
                  seas=list(order=c(0,1,1), 260))
predict<-predict(close.s4, n.ahead=300)
ts.plot(cbind(close.data, predict$pred), lty=1:2, col=1:2,
        main="ARIMA(1,1,1)x(0,1,1) with Frequency 260")
lines(predict$pred+2*predict$se, col="red", lty=3)
lines(predict$pred-2*predict$se ,col="red", lty=3)
#Accuracy
library(forecast)
a<- window(close.data,1,5344)
fit0<-arima(a, order=c(0,1,1),
             seas=list(order=c(0,1,1), 260))
fore0<-forecast(fit0,h=101)
accuracy(fore0,window(close.data,5344,5444),d=0,D=0)
fit<-arima(a, order=c(1,1,0),
            seas=list(order=c(0,1,1), 260))
fore<-forecast(fit,h=101)
accuracy(fore,window(close.data,5344,5444),d=0,D=0)
fit1<-arima(a, order=c(1,1,1),
            seas=list(order=c(0,1,1), 260))
fore1<-forecast(fit1,h=101)
accuracy(fore1,window(close.data,5344,5444),d=0,D=0)
fit2<-arima(a, order=c(0,1,2),
            seas=list(order=c(0,1,1), 260))
fore2<-forecast(fit2,h=101)
accuracy(fore2,window(close.data,5344,5444),d=0,D=0)

```

```

fit3<-arima(a, order=c(2,1,2),
            seas=list(order=c(0,1,1), 260))
fore3<-forecast(fit3,h=101)
accuracy(fore3,window(close.data,5344,5444),d=0,D=0)
fit4<-arima(a, order=c(2,1,0),
            seas=list(order=c(0,1,1), 260))
fore4<-forecast(fit4,h=101)
accuracy(fore4,window(close.data,5344,5444),d=0,D=0)
fit5<-arima(a, order=c(2,1,1),
            seas=list(order=c(0,1,1), 260))
fore5<-forecast(fit5,h=101)
accuracy(fore5,window(close.data,5344,5444),d=0,D=0)
fit6<-arima(a, order=c(1,1,2),
            seas=list(order=c(0,1,1), 260))
fore6<-forecast(fit6,h=101)
accuracy(fore6,window(close.data,5344,5444),d=0,D=0)
fit7<-arima(a, order=c(1,1,0))
fore7<-forecast(fit7,h=101)
accuracy(fore7,window(close.data,5344,5444),d=0,D=0)
fit8<-arima(a, order=c(0,1,1))
fore8<-forecast(fit8,h=101)
accuracy(fore8,window(close.data,5344,5444),d=0,D=0)
fit9<-arima(a, order=c(0,1,2))
fore9<-forecast(fit9,h=101)
accuracy(fore9,window(close.data,5344,5444),d=0,D=0)
fit10<-arima(a, order=c(2,1,0))
fore10<-forecast(fit10,h=101)
accuracy(fore10,window(close.data,5344,5444),d=0,D=0)
fit11<-arima(a, order=c(1,1,1))
fore11<-forecast(fit11,h=101)
accuracy(fore11,window(close.data,5344,5444),d=0,D=0)
fit12<-arima(a, order=c(2,1,2))
fore12<-forecast(fit12,h=101)
accuracy(fore12,window(close.data,5344,5444),d=0,D=0)
fit13<-arima(a, order=c(1,1,2))
fore13<-forecast(fit13,h=101)
accuracy(fore13,window(close.data,5344,5444),d=0,D=0)
fit14<-arima(a, order=c(2,1,1))
fore14<-forecast(fit14,h=101)
accuracy(fore14,window(close.data,5344,5444),d=0,D=0)
par(mfrow=c(2,2))
plot(fore,main="ARIMA(1,1,0)x(0,1,1)
           with Frequency 260")
plot(fore2,main="ARIMA(0,1,2)x(0,1,1)
           with Frequency 260")
plot(fore4,main="ARIMA(2,1,0)x(0,1,1)
           with Frequency 260")
plot(fore3,main="ARIMA(2,1,2)x(0,1,1)
           with Frequency 260")
# Wavelet Transform Based ARIMA Fitting
close.dwt.la8.2<-dwt(close.data[1:4096],"la8",2)
V2<- close.dwt.la8.2$s2
W2<- close.dwt.la8.2$d2

```

```

W1<- close.dwt.la8.2$d1
arima.V2<-arima(V2,order=c(1,1,1))
arima.W2<-arima(W2,order=c(1,1,1))
arima.W1<-arima(W1,order=c(1,1,1))
fore.V2<-forecast(arima.V2,h=100)
fore.W2<-forecast(arima.W2,h=100)
fore.W1<-forecast(arima.W1,h=100)
sum(fore.V2$residuals+fore.W2$residuals
    +fore.W1$residuals)
close.modwt.la8.2<-modwt(close.data[1:5344],
                           "la8",2)
class(close.modwt.la8.2)
names(close.modwt.la8.2)
cV2<-close.modwt.la8.2$s2
cW2<-close.modwt.la8.2$d2
cW1<-close.modwt.la8.2$d1
sum(close.data[1:5344]^2)
sum(cW1^2)+sum(cW2^2)+sum(cV2^2)
sum(close.data[1:5344]^2)-(sum(cW1^2)
                           +sum(cW2^2)+sum(cV2^2))
auto.arima(cV2)
auto.arima(cW2)
auto.arima(cW1)
a<-arima(cV2,order=c(5,1,5))
b<-arima(cW2,order=c(2,0,4))
c<-arima(cW1,order=c(0,0,5))
fore0<-forecast(a,h=100)
fore1<-forecast(b,h=100)
fore2<-forecast(c,h=100)
fit.arima<-arima(close.data[1:5344],
                  order=c(1,1,1))
forecast.arima<-forecast(fit.arima,h=100)
sum(forecast.arima$residuals)
sum(fore0$residuals+fore1$residuals
    +fore2$residuals)
sum(fore.V2$residuals+fore.W2$residuals
    +fore.W1$residuals)
#GARCH Fittings
y=diff(log(close.data))*100
y=y-mean(y)
garchFit(~garch(1,0),data=y,
         include.mean=FALSE)
summary(garchFit(~garch(1,0),data=y,
                 include.mean=FALSE))
garchFit(~garch(1,0),data=y,
         include.mean=FALSE,cond.dist="sstd",trace=F)
summary(garchFit(~garch(1,0),data=y,
                 include.mean=FALSE,cond.dist="sstd",
                 trace=F))
garchFit(~garch(4,0),data=y,
         include.mean=FALSE,cond.dist="sstd",trace=F)
summary(garchFit(~garch(4,0),data=y,
                 include.mean=FALSE,cond.dist="sstd",

```

```

        trace=F))
garchFit(~garch(4,1),data=y,include.mean=FALSE,
         cond.dist="sstd",trace=F)
summary(garchFit(~garch(4,1),data=y,
                 include.mean=FALSE,cond.dist="sstd",
                 trace=F))
garchFit(~garch(4,1),data=y,include.mean=FALSE,
         cond.dist="sged",trace=F)
summary(garchFit(~garch(4,1),data=y,
                 include.mean=FALSE,cond.dist="sged",
                 trace=F))
garchFit(~garch(1,1),data=y,include.mean=FALSE,
         cond.dist="sstd",trace=F)
summary(garchFit(~garch(1,1),data=y,
                 include.mean=FALSE,cond.dist="sstd",
                 trace=F))
garchFit(~garch(1,1),data=y,
         include.mean=FALSE,cond.dist="sged",trace=F)
summary(garchFit(~garch(1,1),data=y,
                 include.mean=FALSE,cond.dist="sged",
                 trace=F))

#Other methods
#AAR
library(tsDyn)
a<- window(close.data,1,5344)
fit<-aar(a,m=3,d=1,steps=100)
pred<-predict(fit, n.ahead=100)
accuracy(pred,window(close.data,5344,5444),d=0,D=0)
fit<-aar(a,m=4,d=1)
pred<-predict(fit, n.ahead=100)
accuracy(pred,window(close.data,5344,5444),d=0,D=0)
fit<-aar(a,m=10,d=1)
pred<-predict(fit, n.ahead=100)
accuracy(pred,window(close.data,5344,5444),d=0,D=0)
#LSTAR(Logistic Smooth Transition AutoRegressive Model)
mod.lstar <- lstar(a,m=4, d=1,mTh=c(0,1,2,3),
                    control=list(maxit=3000))
pred<-predict(mod.lstar, n.ahead=101)
accuracy(pred,window(close.data,5344,5444),d=0,D=0)
#SETAR(Self Threshold Autoregressive model)
selectSETAR(log(close.data), m=4, mL=1:3, mH=1:3,
            thSteps = 5, thDelay=0:2)
mod <- list()
mod[["linear"]] <- linear(close.data, m=4)
mod[["setar"]] <- setar(close.data, m=4,d=1, thDelay=1)
mod[["lstar"]] <- lstar(close.data, m=4,d=1, thDelay=1)
mod[["nnetTs"]] <- nnetTs(close.data, m=4, size=5)
mod[["aar"]] <- aar(close.data, m=4)
mod[["sarima"]] <- arima(close.data, order=c(0,1,2),
                           seas=list(order=c(0,1,1), 260))
sapply(mod, AIC)

```