Robust Methods of Testing Long Range Dependence In Time Series

by

Li Wang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Statistics

Waterloo, Ontario, Canada, 2007

©Li Wang, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Li Wang

Abstract

This thesis develops a novel robust periodogram method for detecting long memory. Though many test for long memory are based on the idea of linear regression, there exists no results in statistical literature on utilizing the robust regression methodology for detection of long memory. The advantage of the robust regression is a substantially less sensitivity to atypical observations or outliers, compared to the classical regression that is based on the least squares method.

The thesis suggests two versions of the robust periodogram methods based on the least quantile and the least trimmed methods. The new robust periodogram methods are shown to provide smaller bias in long memory estimation when compared with the classical periodogram method. However, variability of estimation is increased. Therefore, we develop the bootstrapped modification of the new robust periodogram methods to reduce variability of estimation. The new bootstrapped modifications of the robust periodogram tests substantially reduce variance of estimation and provides a competitively low bias. All proposed robust methods are illustrated by simulations and the case studies on currency exchange rates, and comparative analysis with other existing tests for long memory is carried out.

Acknowledgements

A very appreciative thank you to my supervisor, Doctor Yulia Gel. Her guidance and patience have been invaluable to the progress I've made over this year: for this thesis and for many of the other steps I've taken in my life and study. Thanks to Doctor Johnny Li and Shoja'eddin Chenouri for taking time to read my thesis and provide constructive advice.

I would like to thank the University of Waterloo, specifically the Department of Statistics and Actuarial Science, for giving me this wonderful opportunity. I'm also very thankful for the funding I have been afforded from the department, and Yulia too.

I would also like to thank Mary Lou Dufton for all of the useful advice she has given me over the year.

Thanks to Charlie Chen for Latex help. A big thanks to my colleague Bei Chen and my friend Dan Shen for the encouragement and support along the way.

Finally, my parents. Thank you for everything!!

Contents

1 Introduction **1**

2 Linear Time Series Models **4**
2.1 Fundamental Concepts 4
2.1.1 Stochastic Processes 4
2.1.2 Stationary Stochastic Processes 4
2.2 Autocovariance, Autocorrelation and Partial Autocorrelation 5
2.3 Introduction to Spectral Analysis 6
2.3.1 Spectral Density 7
2.3.2 The Periodogram 8
2.4 Models for Linear Time Series with Short Memory 8
2.4.1 Autoregressive (AR) models 8
2.4.2 Moving Average (MA) models 9
2.4.3 Autoregressive Moving Average (ARMA) Models 11
2.4.4 Autoregressive Integrated Moving Average (ARIMA) model 13
2.4.5 Model Estimation and Diagnostic Checking 13

3 Long Range Dependent Processes **17**
3.1 History 17
3.2 Autoregressive Fractional Integrated Moving Average (ARFIMA) model 17
3.2.1 Long Memory process 18
3.2.2 Parameter Estimation of ARFIMA 19
3.2.3 Tests for Long Range Dependence 20

4 Robust Methods for Detecting Long Range Dependence **22**
4.1 Overview . . . 22
4.2 Two Robust Linear Regression Procedures . . . 23
4.3 Robust Periodogram Method . . . 23
4.4 Simulation Study . . . 24
4.4.1 Simulation Description . . . 24
4.4.2 Results of Simulation Study . . . 25
4.5 Variance Reduction Methods . . . 28
4.6 Bootstrap Method . . . 28

5 Testing Long Range Dependence in Foreign Exchange Rates **31**
5.1 Data Description . . . 31
5.2 Preliminary Analysis . . . 31
5.3 Analysis of Results . . . 32
5.4 Summary . . . 33

6 Conclusions and Future Work **37**

Appendix **39**

References **45**

List of Tables

4.1 Long memory parameter d Estimation in ARFIMA$(0, d, 0)$ processes 27
4.2 Parameter Estimation of d in ARFIMA$(1, d, 0)$ and $(1, d, 1)$ processes 27
4.3 Bootstrap Estimation of d in ARFIMA$(0, d, 0)$ processes 29
4.4 Bootstrap Estimation for d in ARFIMA$(1, d, 0)$ and $(1, d, 1)$ processes 30

5.1 Parameter Estimation for d in four major exchange-rate change series 36
5.2 Standard deviation for d in four major exchange-rate change series 36

List of Figures

2.1 The ACF and PACF plots of a simulated AR(2) process 10
2.2 The ACF and PACF plots of a simulated MA(2) process 11
2.3 The ACF and PACF plots of a simulated ARMA(1,1) process 12
2.4 Time Series plots of a simulated ARIMA(1,1,1) and its first differenced series . 14

3.1 The ACF and PACF plots of a simulated ARFIMA(1,1,1) process 19

5.1 Time Series Plots of Four Major Currency Exchange Rates. 34
5.2 Four Major Currency Exchange-Rate Change Series. 35

Chapter 1

Introduction

Any set of observations that are obtained sequentially at equal or non-equal intervals, can be viewed as a time series. Although typically observations are naturally ordered by time, i.e. hours, days, months etc, there exists other types of ordering, for example, directions in space. Accurate and reliable modeling and forecasting of time series is one of the most important aspects in various fields of science and applications. Every day, inventory, production, scheduling of personnel and machinery, financial and marketing decisions are made which depend on projected short, medium and long-term forecasts of various time series such daily stock prices, quarterly interest rates, daily weather derivatives, yearly demographic rates etc.

The objectives for studying time series can be divided into three main components:

- understanding and description of the generating mechanism of a given time series, which includes exploratory data analysis and modeling,
- forecasting future values of a time series,
- and finally an optimal control of a dynamic system.

The intrinsic nature of a time series is such that the observed values are typically dependent over time or serially correlated, and the objective is to identify and model the postulated temporal dependence structure. If more recent observations depend on the past observed values linearly, such a process is called a linear time series. Clearly such a linear dependence is likely to vanish as the distance between observations increases. Based on how fast the dependence tends to vanish, linear time series can be divided into main categories, i.e. short and long memory processes. As the names imply, short memory linear time series has the property that correlation among

observations decays very fast, i.e. exponentially, as the distance between observation increases, and becomes negligible for measurements that are relatively far apart. In contrast, correlation among observations of a long memory process decays relatively slowly, i.e. hyperbolically, with the increase of a distance between observations and can not be ignored even for quite distant measurements.

Short memory processes are widely used and well investigated in various fields of science, for example, for review in statistics see Brockwell and Davis (2002), in econometrics see Tsay (2005) and in electrical engineering see Ljung (1998) and references therein. In contrast, being discovered only in mid-60s, long memory processes are still a relatively new branch in time series analysis that play an increasingly important role in many modern applications. The long range dependence (LRD) patterns are detected in a variety of fields such as finance, hydrology and computer science. Probably the most well known example of a time series that exhibits long memory is the Nile River minimum water levels data set that was originally published by Troussoun (1925) and then analyzed by Hurst in his seminal paper on long memory processes in 1951. Nowadays the Nile River data are widely used as a benchmark data set for detecting and modeling LRD (see Whitcher et al. (2002) and references therein on the most recent analysis of this data set).

The main focus of this thesis is on detecting long memory using methods of robust statistics. Prior to performing any modeling or forecasting of an LRD process, one needs to detect whether the given time series in fact exhibits long memory. There exists a variety of tests on LRD in both spectral and time domain, e.g. the most popular tests are the periodogram method in spectral domain and the aggregated and differenced variances methods in time domain. Though dealing with different statistics, all those methods are essentially based on the same idea of a simple linear regression where the LRD parameter is the least squares (LS) estimate of a linear regression slope. Though the LS estimates are known to be very sensitive to outliers, surprisingly there exists no literature on utilizing the methods of robust regression in estimating LRD. Thus, the main contributions of the thesis are the following:

1. the novel idea on employing robust methods for detecting the LRD parameter is proposed, which is not appeared in any prior statistical research;

2. the new robust periodogram method is developed; various types of the robust regression are employed and tested, i.e. the least quantile and least trimmed regression; all robust methods are illustrated by simulations and compared with the existing classical

approaches;

3. the new robust periodogram is shown to provide smaller bias of the LRD estimation when compared with the classical periodogram method; however, variability is increased;

4. therefore, the new bootstrapped modification of the robust periodogram method is proposed, which noticeably reduce spread of the LRD estimation and provides a competitively low bias;

5. the case study on detecting LRD in currency exchange rates and comparative analysis of various tests for LRD is carried out.

The thesis is organized as follows. Chapter 2 starts from introduction of some fundamental concepts and definitions in time series analysis such as weak and strict stationarity, autocorrelation and partial autocorrelation functions etc. Since modeling techniques for the LRD processes can be viewed as an extension of the available methodology for short memory time series, Chapter 2 discusses also the Box-Jenkins approach for short memory processes, e.g. Autoregressive (AR), Moving Average (MA) models, Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) models.

Long range dependent processes are introduced in Chapter 3 and followed by discussion of an Autoregressive Fractional Integrated Moving Average (ARFIMA) model that is the main modeling tool of LRD in discrete time. The chapter 3 concludes with the three most widely used procedures for detecting LRD: the periodogram, the aggregated and differenced variance methods.

Chapter 4 starts from discussion of the robust regression methods, i.e. least quantile and trimmed regression techniques. Then we propose the robust modification of the periodogram procedure that utilizes the least quantile and trimmed regression. All methods are illustrated by simulations.

Case studies on detection LRD in financial time series are presented in Chapter 5. Both classical and robust methods are applied to estimate LRD and comparative analysis is carried out. Chapter 6 summarizes the main results and concludes the thesis with the outline for the future work on improving the accuracy of the robust LRD detection.

Chapter 2

Linear Time Series Models

2.1 Fundamental Concepts

2.1.1 Stochastic Processes

To formalize the discussion of main concepts, we start by defining a time series. The definition is based on an abstract probability space $\{\Omega, \mathcal{F}, \mathbb{P}\}$, where Ω is the sample space of elementary events, $\mathcal{F}$ is a sigma-algebra defined on the sample space and $\mathbb{P}$ is a probability measure on Ω. We also introduce an index T that has an ordering relation defined on it. For example, for $t_1, t_2 \in T, t_1 < t_2$. Typically, $T = \mathbb{R}, T = \mathbb{Z}^+$ or $T = \mathbb{Z}$.

A stochastic process is a family of random variables $\{X(t,\omega), \omega \in \Omega, t \in T\}$ defined on a probability space $(\Omega, \mathbb{P})$. In particular, for a given ω, $X(t,\omega)$ is a function from T into $\mathbb{R}$, which is called a realization of the stochastic process. The population that consists of all possible realizations is called the ensemble in stochastic processes and time series analysis.

In reality, we have just one realization from a certain stochastic process, i.e. for a fixed ω, and we typically observe only finite number of points, i.e., T is finite. Therefore, we usually suppress the variable ω and simply write $X(t,\omega)$ as X_t. $\{X_t\}$ is called a time series with a fixed ω.

2.1.2 Stationary Stochastic Processes

Suppose that we have a stochastic process $\{X_t, t = 0, \pm 1, \pm 2, \ldots\}$. There are two widely used definitions of stationarity: strict stationarity and weak stationarity.

Definition 2.1: $\{X_t\}$ is said to be strictly or strongly stationary if, for any positive integer k, and k time points t_1, ..., t_k and any integer lag h, we have that the vectors $\{X_{t_1}, \ldots, X_{t_k}\}$ and $\{X_{t_{1+h}}, \ldots, X_{t_{k+h}}\}$ have the same joint distribution.

Definition 2.2: $\{X_t\}$ is said to be weakly stationary if all the variances of the process are finite and the following conditions are hold:

$$E(X_t) = \mu < \infty;$$

$$Cov(X_t, X_{t+h}) = \gamma_h < \infty;$$

where μ and γ_h do not depend on t.

Definition 2.3: White noise is a special case of weakly stationary process. $\{\epsilon_t\}$ is a white noise if $E(\epsilon_t) = 0$ and $\mathrm{Cov}(\epsilon_\mathrm{i}, \epsilon_\mathrm{j}) = \sigma^2\delta_\mathrm{ij}$, where δ_{ij} is the Kronecker symbol that is equal to 1 if $i = j$ and 0 otherwise. It can be written as $\epsilon_t \sim WN(0, \sigma^2)$.

Provided that the variances are finite, it is easily seen that a strictly stationary process is weakly stationary. However, the converse is not necessarily true. The reason is that weakly stationary processes necessarily satisfy the requirements on the first and second order moment but do not follow any restriction on higher moments.

The concept of weak stationarity is commonly used in time series analysis because it introduces an assumption that well describes real data and makes it possible to conduct modeling. Unless mentioned otherwise, processes discussed in this paper are weakly stationary processes. A weakly stationary time series can be described by its mean, variance and autocorrelation function in the time domain, or equivalently by its mean and spectral density function in the frequency domain. In next section we consider the properties of these functions in both the time and frequency domains.

2.2 Autocovariance, Autocorrelation and Partial Autocorrelation

A weakly stationary process $\{X_t\}$ has mean μ and variance σ^2, which are both constant and finite. The covariances between X_t and X_{t+k} defined as

$$\gamma_k = Cov(X_t, X_{t+k}),$$

and the correlation between X_t and X_{t+k} as

$$\rho_k = \frac{Cov(X_t, X_{t+k})}{\gamma_0},$$

where $\gamma_0 = Var(X_t) = Var(X_{t+k})$. As functions of k, γ_k is called the autocovariance function and ρ_k is called the autocorrelation function at lag k in time series analysis. This is because those functions represent the covariance and correlation between X_t and X_{t+k} from the same stochastic process, separated by k time lags.

For a weakly stationary process $\{X_t\}$, the autocovariance function γ_k and autocorrelation function ρ_k has the following properties:

- $\gamma_0 = \text{Var}(X_t); \rho_0 = 1.$
- $|\gamma_k| \leq \gamma_0; |\rho_k| < 1.$
- $\rho_k = \rho_{-k}; \gamma_k = \gamma_{-k}.$

The partial autocorrelation is based on the least-squares regression of X_t on $X_{t-k}, \ldots, X_{t-1}$. Mathematically, it can be calculated based on the following model:

$$Y_t = \sum_{j=1}^{k} a_{j,k} Y_{t-j} + \epsilon_t, t > k, \tag{2.1}$$

where ϵ_t is independent of $Y_1, \ldots, Y_{t-1}$. Here $a_{j,k}$'s are the coefficients of the AR(k) model.

Least squares estimates of $a_{j,k},\ j = 1, \ldots, k$, can be obtained by minimizing the mean sum of squares

$$\sigma^2 = \frac{1}{T} \sum_{t=k+1}^{T} (Y_t - \sum_{j=1}^{k} a_{j,k} Y_{t-j})^2.$$

The k-th order sample partial autocorrelation coefficient, or PACF, measures the improvement in mean squared residual errors from $(k-1)$-th order regression to k-th order regression. This is determined by $a_{k,k}$ in the equation 2.1. Thus, $a_{k,k}$ is called the k-th order sample PACF.

2.3 Introduction to Spectral Analysis

The spectral representation of a stationary time series $\{X_t\}$ essentially decomposes $\{X_t\}$ into a sum of sinusoidal components with uncorrelated random coefficients. The analysis of stationary

processes by means of their spectral representation is often referred to as the "frequency domain analysis" of time series or "spectral analysis". It is equivalent to "time domain" analysis based on the autocovariance function, but provides an alternate way of viewing the process, which for some applications may be easier to apply. A famous example for spectral analysis is the cyclical nature of sun spot activity (e.g., see Bloomfield (1976), or Shumway (1988)). It turns out that sun spot activity varies over 11 year cycles. Other examples including weather patterns, fluctuations in commodity prices, economic activity, etc. are also often used in the literature to demonstrate this technique. In the next two subsections we introduce the spectral density of a stationary process $\{X_t\}$ and the periodogram, a sample estimate of the spectral density.

2.3.1 Spectral Density

For any sequence of autocovariances $\{\gamma_k\}$ generated by a weakly stationary process, there exists a function F such that

$$\gamma_k = \int_{-\pi}^{\pi} e^{ik\lambda}\, dF(\lambda), \tag{2.2}$$

where F is the unique function on $[-\pi, \pi]$ satisfying

- $F(-\pi) = 0$,
- F is non-decreasing and right-continuous,
- F has increments symmetric about 0, meaning that for any $0 \leq a < b \leq \pi$, we have

$$F(b) - F(a) = F(-a) - F(-b).$$

Then F is called the spectral distribution function, because it has many of the properties of a probability distribution function except that $F(\pi) = \gamma_0 = Var(X_t)$, which is not necessarily 1.

Definition If F is everywhere continuous and differentiable, with $f(\lambda) = \frac{d}{d\lambda}(F(\lambda))$, then f is called the spectral density function. Therefore, equation 5.2 can be simplified to

$$\gamma_k = \int_{-\pi}^{\pi} e^{ik\lambda} f(\lambda)\, d\lambda.$$

If $\sum_{k=-\infty}^{\infty} \gamma_k < \infty$, then it can be shown that f always exists and is given by

$$f(\lambda) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \gamma_k e^{-ik\lambda}$$

2.3.2 The Periodogram

If $\{X_t\}$ is a stationary process with autocovariance function $\gamma(\cdot)$ and spectral density $f(\cdot)$, then as the sample $\hat{\gamma}(\cdot)$ of the observations $\{x_1, \ldots, x_n\}$ can be regarded as an estimate of $\gamma(\cdot)$, so also can the periodogram $I_T(\cdot)$ of the observations be regarded as a sample estimate of $f(\cdot)$.

Definition For spectral densities, the simplest estimate is given by the periodogram

$$I_T(\lambda) = \frac{1}{2\pi T}|\sum_{t=1}^{T} X_t e^{-i\lambda t}|^2.$$

The proof can be found in Brockwell and Davis (2002). However, the sample periodogram is too rough to be a good estimator of the spectral density for most practical applications. Various operations on the periodogram, e.g., smoothing and tapering, are introduced to improve on the accuracy of the raw periodogram as a spectral density estimator.

2.4 Models for Linear Time Series with Short Memory: the Box-Jenkins Methodology

Before start discussing modeling of short memory time series, we introduce a backshift operator B to simplify our notations. We define a backshift operator B as

$$BX_t = X_{t-1}, B^2X_t = B(BX_t) = BX_{t-1} = X_{t-2}, \ldots, B^kX_t = X_{t-k}$$

and the identity operator I as $IX_t = X_t$. Introducing the backshift operator B and identity operator I enable us to write a time series model in a compact form that is easy to manipulate and deal with.

2.4.1 Autoregressive (AR) models

Suppose that the current observed value is Y_t and the p past values are available, $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$. An AR model with order p can be expressed as follows:

$$Y_t = a_1Y_{t-1} + a_2Y_{t-2} + \ldots + a_pY_{t-p} + \epsilon_t, \tag{2.3}$$

where Y_t is weakly stationary, $a_1, a_2, \ldots, a_p$ are constants and $a_p \neq 0$. Unless otherwise stated, we assume that $\{\epsilon_t\}$ is a white noise series with mean zero and constant variance.

Using the backshift operator B, equation (2.3) can be rewritten as $(I - \sum_{r=1}^{p}(a_r B^r)Y_t = \epsilon_t$, or in more compact notation as $a(B)Y = \epsilon$, where

$$a(B) = 1 - a_1 B - a_2 B^2 - \ldots - a_p B^p.$$

The process Y_t is weakly stationary if the polynomial $a(B)$ has roots outside the unit circle (Box-Jenkins (1976)).

The ACF plot of an AR(p) process has p initial spikes and then damps out as a mixed exponential decay of order p (never 0). The PACF plot has p initial spikes and then cuts off. It makes perfect sense that in terms of ACF and PACF plots, correlations between two events become smaller and smaller as the time interval becomes larger and larger. Figure 2.1 shows an example of the ACF and PACF plots of a simulated AR(2) process with $a_1 = 0.7$ and $a_2 = 0.2$ based on 1000 observations.

2.4.2 Moving Average (MA) models

Independent from the autoregressive process, what happens now can also be affected by past errors which cannot be accounted for by the autoregressive component. A zero mean stationary process $\{Y_t\}$ is called a moving average process of order, q, if Y_t satisfies

$$Y_t = \epsilon_t + b_1\epsilon_{t-1} + b_2\epsilon_{t-2} + \ldots + b_q\epsilon_{t-q},$$

where ϵ_t is white noise. This equation can be rewritten as

$$Y_t = (I + b_1 B + b_2 B^2 + \ldots + b_q B^q)\epsilon_t,$$

or in the more compact form as $Y = b(B)\epsilon$, where $b(B)$ is the polynomial in backshift operator B of degree q.

There is a duality between the moving average process and the autoregressive process (e.g., see Box-Jenkins (1976), or Montgomery, et al., (1990)), that is, the moving average equation above can be inverted into an autoregressive form of infinite order. However, analogous to the stationarity condition described above, this can only be done if the moving average parameters follow certain conditions, i.e. the model is invertible.

Due to the duality between MA and AR models, we can easily find that for a MA(q) model, the PACF plot has q initial spikes and then damps out as a mixed exponential decay of order

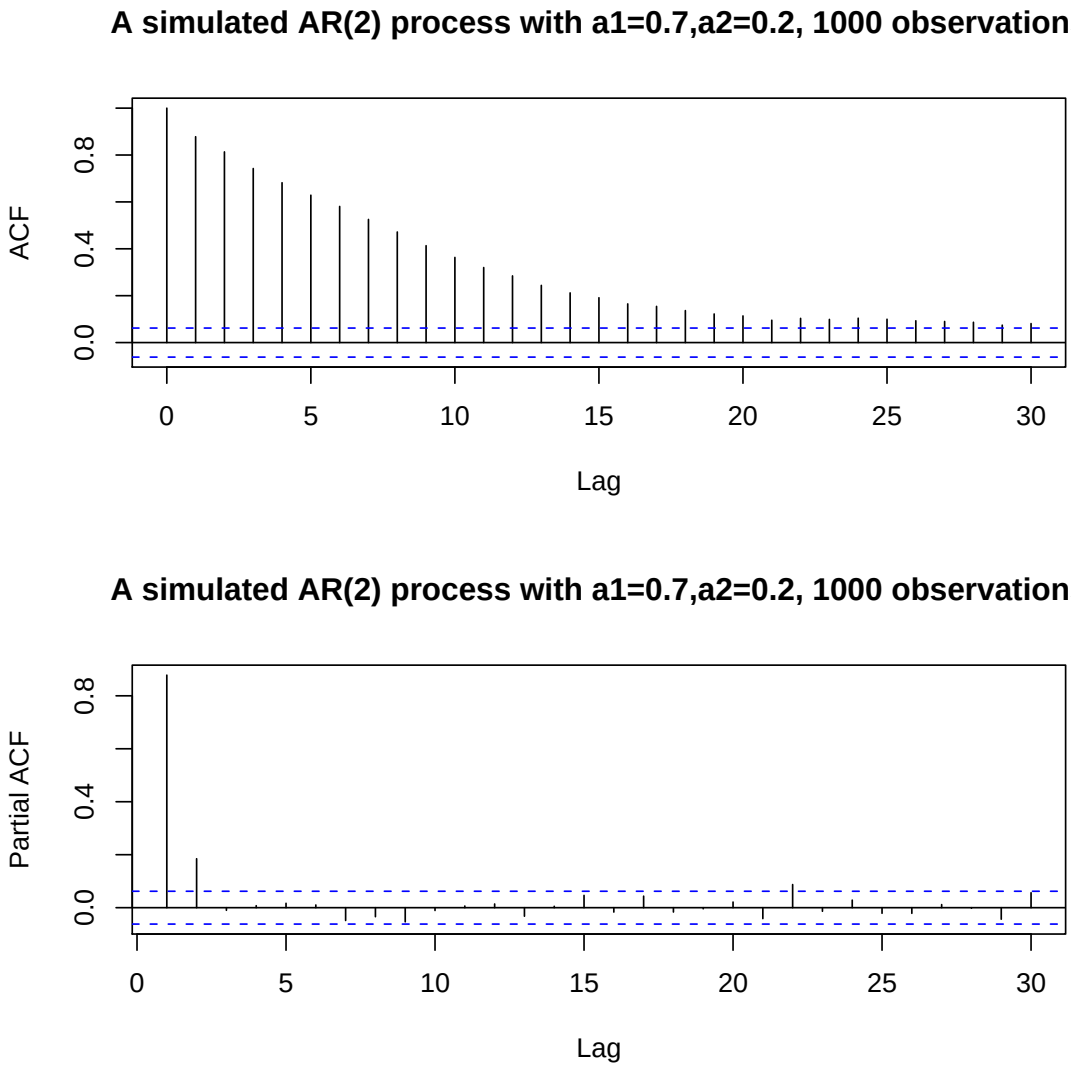

Figure 2.1: The ACF and PACF plots of a simulated AR(2) process

q (never 0). The ACF plot has q initial spikes and then cuts off. The sample ACF and PACF plots of a simulated MA(2) process with $b_1 = 0.7$ and $b_2 = 0.2$ based on 1000 observations are shown in Figure 2.2.

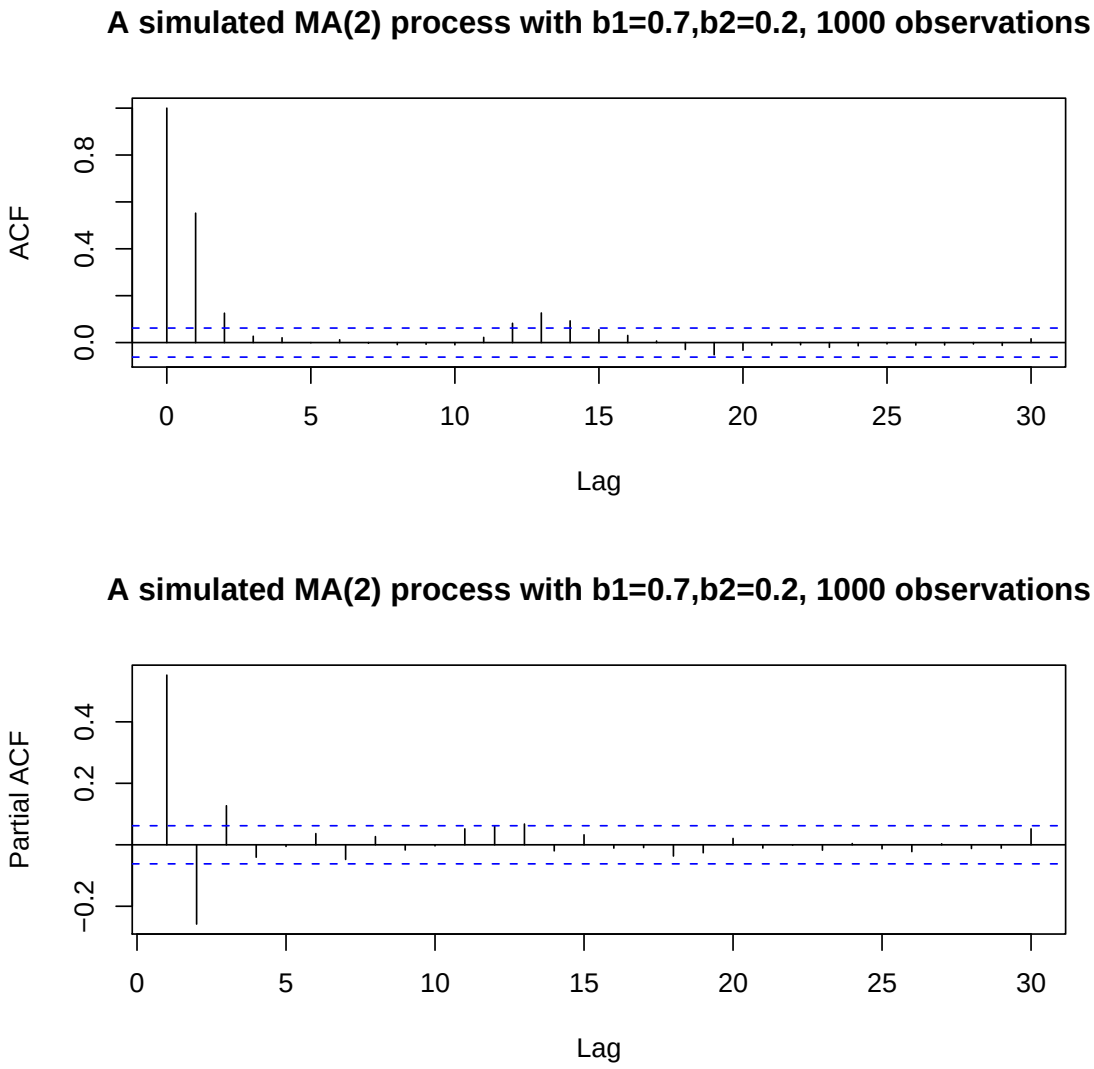

Figure 2.2: The ACF and PACF plots of a simulated MA(2) process

2.4.3 Autoregressive Moving Average (ARMA) Models

An ARMA(p, q) process is a mixture of p autoregressive components and q moving average components. In short, it can be expressed as $a(B)Y_t = b(B)\epsilon_t$, where ϵ_t is white noise, and $a(B)$ and $b(B)$ are the polynomials of degree p and q respectively.

In a mixed model ARMA(p, q) process, neither the ACF nor the PACF cut off at a certain lag. Both the ACF and PACF exhibit mixed exponential decay. This happens because AR(p) component brings mixed exponential decay into the ACF, while the MA component brings mixed exponential decay into the PACF. Figure 2.3 is an example of ACF and PACF plots of a simulated ARMA(1,1) process with $a_1 = 0.5$ and $b_1 = 0.2$ based on 1000 observations.

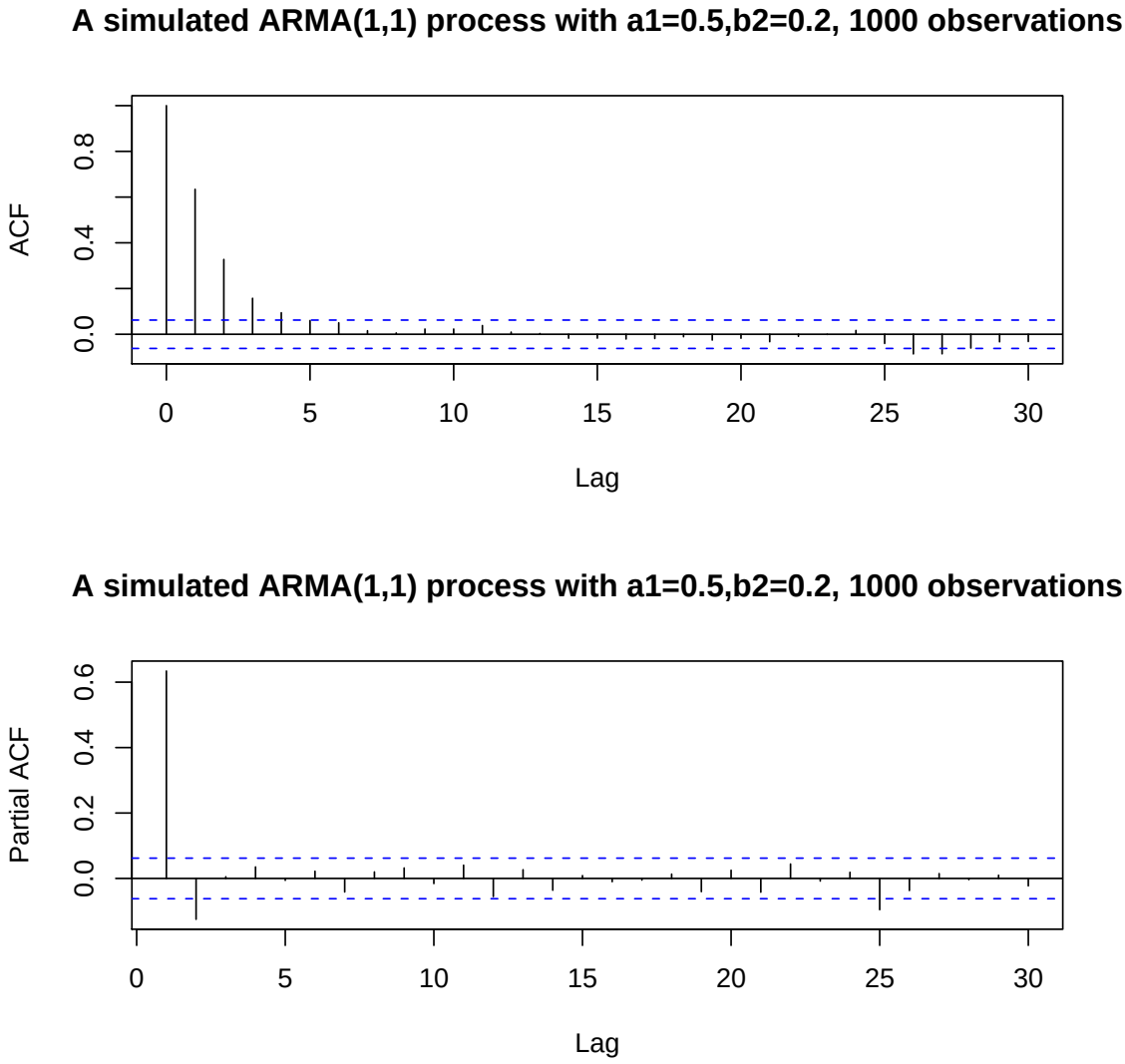

Figure 2.3: The ACF and PACF plots of a simulated ARMA(1,1) process

2.4.4 Autoregressive Integrated Moving Average (ARIMA) model

In the above sections we discussed ARMA models that require weak stationarity of time series data. However, weak stationarity is not always achieved in real life series, and non-stationarity can be caused by unit roots existing in the either the AR component of an ARMA model. Such phenomena can be dealt with ARMA models including an extra integrating (or differencing) parameter d, i.e., ARIMA models.

We define $\{Y_t\}$ to be an ARIMA(p, d, q) process if $a(B)Y_t = b(B)\epsilon_t$, where:

- p is the number of autoregressive terms,
- d is the number of differences and d takes positive integer values,
- q is the number of moving-average terms.

To identify the appropriate ARIMA model for a time series, we start by identifying the order of differencing needed to stabilize the series and remove the seasonality, perhaps in conjunction with a variance-stabilizing transformation such as logging or deflating. After stabilizing the data, we can start looking for an appropriate ARMA model to fit the data. Therefore, fitting an ARIMA model is basically a combination of differencing data and fitting an ARMA model. Figure 2.4 shows time series plots of a simulated ARIMA$(1, 1, 1)$ process with $a_1 = 0.7$ and $b_1 = 0.2$ based on 1000 observations and its single differenced transformation. The time series plot of raw data does not exhibit stationarity. It decreases first until the lag 390 and then increases thereafter. However, after differencing once, the process seems more stationary with mean zero and constant variance. Hence, an appropriate ARMA model can be fitted to the differenced data.

2.4.5 Model Estimation and Diagnostic Checking

After initial processing of the data, the time series now is stationary and model orders p and q are identified. This section reviews three common methods to estimate parameters $\{a_1, \ldots, a_p\}$ and $\{b_1, \ldots, b_q\}$, i.e. the maximum likelihood, the least squares and the Yule-Walker methods. Thereafter, we run diagnostic checks to verify that the model constructed fits the data well.

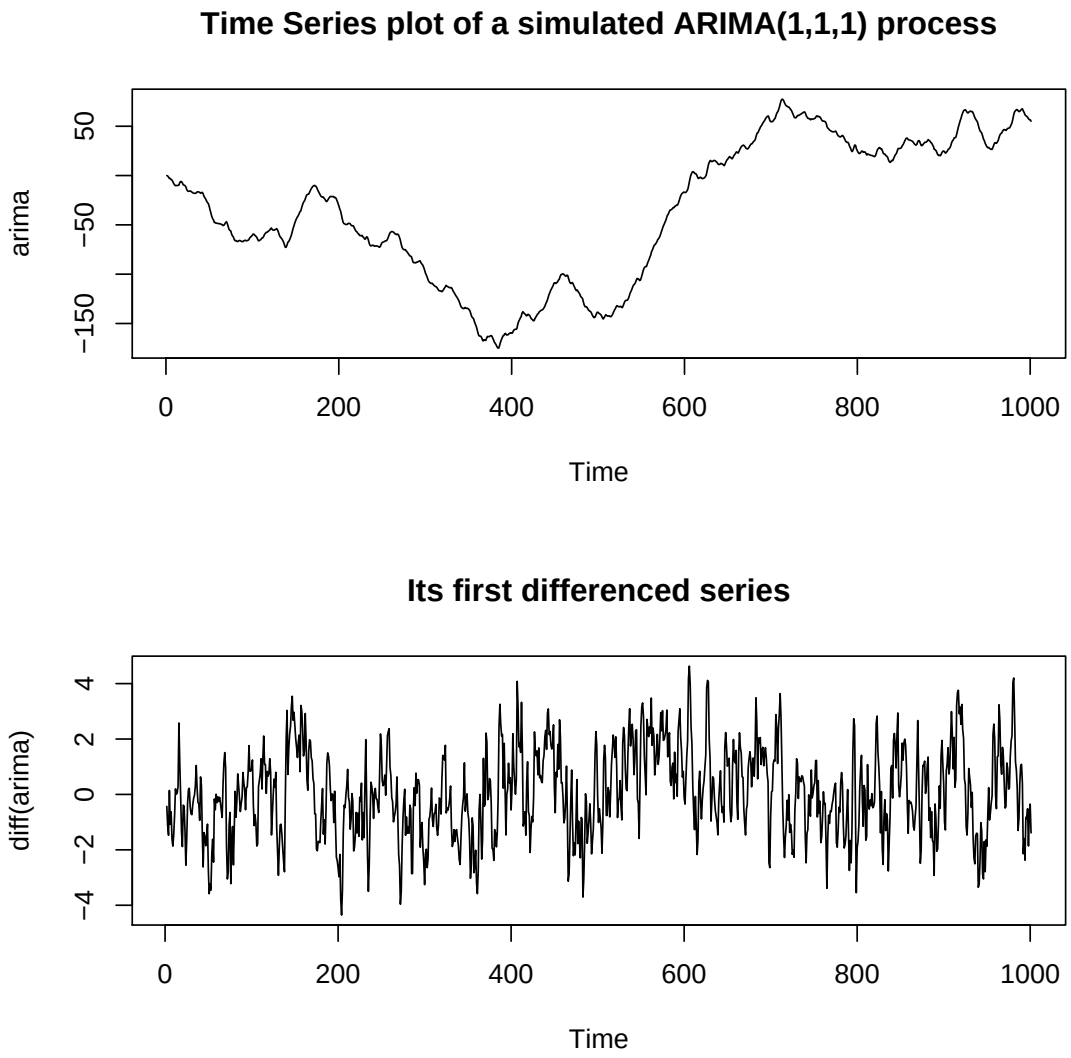

Figure 2.4: Time Series plots of a simulated ARIMA(1,1,1) and its first differenced series

Maximum Likelihood Estimation (MLE)

The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability (likelihood) function of the sample data. Consider an ARMA(p, q)

model as an example. The likelihood function is defined as follows

$$f(Y_1, \ldots, Y_T; \varphi),$$

where $\varphi = (a_1, \ldots, a_p, d, b_1, \ldots, b_q)$ and $f(\cdot|\varphi)$ is the joint distribution of $\{Y_1, \ldots, Y_T\}$.

The maximum likelihood estimate, φ, is the parameter value that maximizes the likelihood function. The method of maximum likelihood provides the smallest error variance among three methods and yields asymptotically consistent and normally distributed estimators that asymptotically reach the Cramer-Rao bound. (Brockwell and Davis (2002)). However, the MLE method is very computationally expensive, as Baillie (1996) notes.

Yule-Walker Estimation

The Yule-Walker method can be used to estimate the parameters in autoregressive time series models. Let us consider an autoregressive model of order p (AR(p). Parameter estimation is carried out by solving a linear system of the Yule-Walker equations that are derived as follows.

In first, we multiply each side of equation (2.3) by X_{t-k}, $k = 0, 1, \ldots, p$. By taking expectations on both sides, we get a set of linear equations called the Yule-Walker equations

$$\gamma(k) = a_1\gamma(k-1) + \ldots + a_p\gamma(k-p)$$

for $k = 1, 2, \ldots, p$, and

$$\gamma(0) = a_1\gamma(1) + \ldots + a_p\gamma(p) = \sigma^2.$$

In practice, we can substitute the sample estimates of the autocovariances $\gamma(1), \ldots, \gamma(p)$ to obtain the sample parameter estimates of $a_1, \ldots, a_p$.

The Yule-Walker estimation is a simple computational procedure, i.e. the model parameters are obtained by solving the p linear equations. However, in the YW method we replace theoretical estimates of the autocovariances with the sample ones, which would cause much higher error variances compared with the MLE or LS estimates.

Least Square Method

The LS method can be viewed as a stationary Kalman filter (see Abraham and Ledolter (2005)). The LS estimates satisfy the recursive equations that enable to use LS for the on-line parameter estimation and forecasting, i.e. when the new observations arrive at each time interval and should be immediately taken into account for modeling and prediction.

For large samples the LS estimates of the coefficients $\{a_1, \ldots, a_p, b_1, \ldots, b_q\}$ of an ARMA(p, q) model have approximately normal distribution similarly to the ML estimators. The LS method offers a higher error variance than MLE but provides the same asymptotic properties. Being a recursive procedure, LS is very computationally efficient.

Diagnostic Checking

When the model is identified and the parameters are estimated, diagnostic checks are then applied to the fitted model. There are two basic techniques:

- Overfitting: Having identified what is believed to be a correct model, we actually fit a more general one that includes additional parameters. Then we use likelihood ratio or t-tests to check if additional parameters are significant. If they are not, then we conclude the model fits the data well, and vice versa.

- Residual Analysis: This method assumes that if the fitted model is appropriate, then the residuals should behave in a manner that consistent with the model. For example, in ARMA/ARIMA models, the residuals should be white noise. Therefore, we can calculate the residuals from the fitted model and construct time series and ACF plots of residuals to check if they are consistent with white noise. In addition, normality assumption needs to be satisfied for residuals in order to construct confidence intervals and prediction intervals.

Chapter 3

Long Range Dependent Processes

3.1 History

Hurst (1951) detected the persistence of wet and drought periods from Nile River flow data, known as the Hurst phenomenon. In order to explain the Hurst phenomenon, Mandelbrot and Van Ness (1968) showed that persistence is compatible with stationarity, by constructing fractional Brownian motion (fBm), a weakly stationary stochastic process with a hyperbolically decaying autocorrelation function in continuous time and hence a long memory. After this germinating paper, long range dependence started to attract significant attention. Mandelbrot and Wallis (1969) showed that fractional Gaussian noise, which is a discrete time analogue of fBm, exhibits the Hurst phenomenon. Also, Mandelbrot and Wallis found that the Hurst exponent is equal to $1-d$, where d is a fractional differencing parameter. Currently, there are two major schools for modeling long memory processes: continuous time models, such as fractional Gaussian noise, and discrete time models, such as ARFIMA (introduced independently by Granger and Joyeux (1980) and Hosking (1981)). Since most real life time series are discrete, e.g., on daily or yearly basis, our main focus is on ARFIMA models in this thesis.

3.2 Autoregressive Fractional Integrated Moving Average (ARFIMA) model

Nowadays many practical time series exhibit the property that dependence between distant observations is small but not negligible. This property is referred as long-range dependence

or long memory. Probably the most well known example of a time series that exhibits long memory property is the Nile river minimum water levels data set that is currently widely used as a benchmark data. Before discussing the ARFIMA model, we first introduce the definition of long memory in both the time and frequency domains.

3.2.1 Long Memory process

Let $\{X_t\}$ be a stationary process with autocorrelation function $\rho(\cdot)$ and spectral density $f(\cdot)$.

Definition (time domain): $\{X_t\}$ is called a stationary process with long memory property if there exist a real number $H \in (0.5, 1)$ and a constant $c_t > 0$ such that

$$\lim_{k\to\infty} \frac{\rho(k)}{c_t k^{2(H-1)}} = 1,$$

where H is called the Hurst parameter and $d = H - 0.5$ is called the long memory parameter or fractional differencing parameter in ARFIMA(p, d, q) processes.

Definition (Frequency domain): $\{X_t\}$ is called a stationary process with long memory property if there exits a constant $c_f > 0$ such that

$$\lim_{\nu\to 0} \frac{f(\nu)}{c_f|\nu|^{1-2H}} = \lim_{\nu\to 0} \frac{f(\nu)}{c_f|\nu|^{-2d}} = 1. \tag{3.1}$$

Long memory processes are detected in many applications and play an increasingly important role in time series analysis. Autoregressive Fractional Integrated Moving Average (ARFIMA) models were introduced independently by Granger and Joyeux (1980) and Hosking (1981) to deal with long memory series in discrete time.

An ARFIMA(p, d, q) process is defined by

$$a(B)(1 - B)^d Y_t = b(B)\epsilon_t,$$

where $\{Y_t\}$ is the process of interest and $\epsilon_t \sim WN(0, \sigma^2)$; B is the backward shift operator; $a(B) = 1 - a_1 B - \cdots - a_p B^p, b(B) = 1 + b_1 B + \cdots + b_q B^q$. The operator $(1-B)^d$ is the fractional differencing operator defined by $(1 - B)^d = \sum_{k=0}^{\infty} \Gamma(k - d)B^k/\Gamma(k + 1)\Gamma(-d)$ with $\Gamma(\cdot)$ being the gamma function. ARFIMA(p, d, q) is stationary and invertible if $d \leq |0.5|$ and the roots of the $a(B)$ and $b(B)$ lie outside the unit circle. Note that the ARMA and ARIMA models can be thought of as particular cases of ARFIMA models having $d = 0$ and $d = 1, 2\cdots$ respectively.

ACF and PACF of an ARFIMA(p, d, q) process never cut off. Moreover, because of the long memory property, it has slower decay of autocorrelation than those of short memory models, e.g ARMA. This is shown by comparison of Figure 3.1 and Figure 2.3.

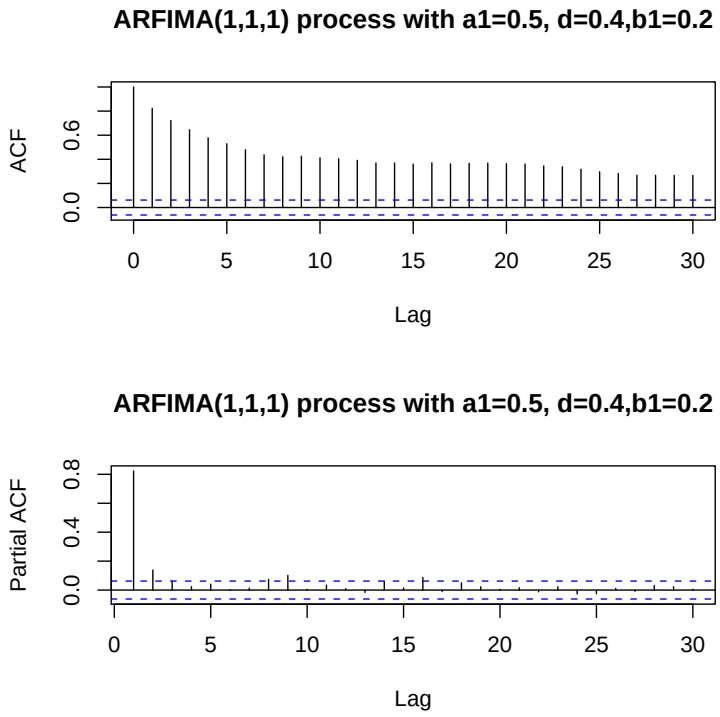

Figure 3.1: The ACF and PACF plots of a simulated ARFIMA(1,1,1) process

3.2.2 Parameter Estimation of ARFIMA

The two classes of ARFIMA estimation methods are developed, i.e. two-step and one-step procedures. The majority of applications utilize the two-step approach. In the first step, an estimate of the long memory parameter d is obtained (usually in the frequency domain). In second, the data is fractionally differenced using the estimated d and a standard ARMA estimation procedure is applied to the adjusted data.

The alternative one-step method is to simultaneously estimate the long memory parameter d using the maximum likelihood procedure in either time or frequency domain. The main drawbacks of the one-step approach are the need for accurate initial guess parameter values, the potential existence of local maxima in the likelihood function and very high computational complexity.

In the next section, we discuss the three most common methods for estimation of d, i.e. the

periodogram method, the aggregated- and differenced-variance methods.

3.2.3 Tests for Long Range Dependence

Periodogram method

The periodogram method is based on the equation 3.1. In particular, the power spectral density of a long memory process obeys a power law near the origin, i.e. $f(\nu) \sim c_f|\nu|^{-2d}$, as $\nu \to 0$. Thus, by taking logarithm on both sides, we get $\log f(\nu) \sim -2d\log(|\nu|)$, as $\nu \to 0$.

Since the spectral density $f(\nu)$ is the Fourier transform of the autocorrelation function, an estimate of the spectral density can be obtained by taking the inverse Fourier transform of the estimate of the autocorrelation function. This estimator is referred to as a periodogram $I(\nu)$.

Therefore, the long memory parameter d can be estimated from the least squares regression

$$\log(I(\nu_j)) = c - 2d\log(\nu_j) + \eta_j,\ j = 1, 2, \ldots, n$$

where $\nu_j = 2\pi j/T$, $j = 1, \ldots, T-1$, $n = g(T) \ll T$, and T is the sample size. $I(\nu_j)$ is the periodogram of the series at frequency ν_j defined by

$$I(\nu) = \frac{1}{2\pi T}|\sum_{t=1}^{T}(X_k - \bar{X})e^{ik\nu}|^2.$$

The periodogram plot is the graph of $\{\log(\nu_j), \log I(\nu_j)\}$, $j = 1, 2, \ldots, n$. The typical threshold value utilized in detection of d is $n = T^{0.5}$. Theoretically the log-log plot should provide a straight line with a slope of $-2d$.

Aggregated-variance methods

A characteristic trait of long-memory processes is that the variance of an N-member sample mean decreases more slowly than N^{-1} (Beran (1989)). In the same paper, Beran showed that given N data points X_i, $i = 1, \ldots, N$

$$Var(\frac{1}{N}\sum_{i=1}^{N} X_i) = N^{2d-1},\ \text{as } N \to \infty.$$

This suggests the following method for estimating d. Divide the series into $k = N/m$ blocks of size m and compute the mean for each block

$$x_k(m) = \frac{1}{m}\sum_{(k-1)m+1}^{km} X_i,\ \text{where } k = 1, \ldots, N/m.$$

Variance of the block means

$$s^2(m) = \frac{1}{N/(m-1)} \sum_{k=1}^{N/m} (x_k(m) - \bar{x})^2, \text{ where } \bar{x} \text{ is the overall mean.}$$

Now a log-log plot of $s^2(m)$ against m should yield a straight line with a slope of $2d - 1$. This is known as aggregated variance method.

A drawback of this method is that inhomogeneity in the data can produce a positive value of d even in the absence of long memory. A modification of the above method is called the differenced variance method, which avoids this problem.

Differenced-variance method

The main idea of the differenced variance method is to study the first-order difference of the above variances

$$\nabla s^2(m) = s^2(m+1) - s^2(m).$$

Teverovsky and Taqqu (1997) show that a log-log plot of this quantity against m will again asymptotically produce a straight line with slop $2d-1$ and the value of d is not affected by the inhomogeneity of the data.

Chapter 4

Robust Methods for Detecting Long Range Dependence

4.1 Overview

It is well known in statistical literature that even a few atypical observation, or outliers, can severely affect the sample estimates of moments (i.e. mean, variance etc) and consequently also the accuracy of estimates of a linear model that are based on the least squares (LS) method (Ord (1987), Weisberg (1985)). The similar problem occurs in estimation of parameters of a linear time series model (see Abraham and Ledolter (2005)).

Thus, a more robust to outliers procedure is to be used instead of the classical LS method that is based on minimizing the least square deviation of errors from the sample mean. There exists a variety of robust regression methods that utilize robust measures of location and scale, for example, a sample median or a trimmed mean rather than a sample mean and/or a least absolute deviation or median absolute deviation instead of the classical estimate of a standard deviation. In some cases it is difficult to judge whether the found atypical observations are in fact outliers or the underlying distribution of observed data is heavier than the reference distribution, i.e. typically the normal distribution. In those cases, the robust regression procedures are also found to be useful in providing more resistant to heavier tails estimates and thus are sometimes called to be "distributionally robust estimators" (Olive (2007)).

In the following section we discuss two robust statistics: least quantile of squares (LQS) estimator and least trimmed sum of squares (LTS) estimator.

4.2 Two Robust Linear Regression Procedures

Consider a multiple linear regression model

$$Y = X\beta + \epsilon,$$

where Y is a vector of response variables and X a vector of explanatory variables. Let β be a vector of unknown parameters and ϵ be random errors.

For a robust regression estimator $\hat{\beta}$ of β, suppose that $|r(\hat{\beta})|_i$ denote the i-th ordered absolute residual from the estimate $\hat{\beta}$ sorted from smallest to largest. Let $|r_i^2(\hat{\beta})|$ denote the i-th ordered squared residual.

Definition: The least quantile of squares (LQS(c_n)) estimator minimizes the criterion

$$Q_{LQS}(\hat{\beta}) = r_{c_n}^2(\hat{\beta}).$$

When $c_n/n \to 1/2$, the LQS(c_n) estimator is also known as the least median of squares (LMS) estimator (Hampel (1975)).

Definition: The least trimmed sum of squares (LTS(c_n)) estimator (Rousseeuw (1984)) minimizes the criterion

$$Q_{LTS}(\hat{\beta}) = \sum r_i^2(\hat{\beta}).$$

The LQS and LTS methods are less affected by outliers than the classical LS method since those procedures are based on order statistics of the residuals. In particular, LQS minimizes a certain quantile of residuals (e.g. %50-th percentile of residuals, or median) while LTS minimizes the sum of trimmed residuals. From another perspective, if outliers or measurement errors are not present, those robust methods may become less accurate than the classical estimators. When the sample size of observations increases, the effect of outliers is minimized and all estimators asymptotically yield the same result (under certain conditions on the underlying distribution). For finite samples, however, one typically needs to apply classical and robust procedures simultaneously and compare the obtained estimates in order to conclude on the effect of atypical observations for different scenarios. This tradeoff is typically worthwhile in many applications since outliers or measurement errors frequently occur in real data.

4.3 Robust Periodogram Method

As discussed in section (3.2.2), the periodogram method is based on the linear regression of $\log(f(\nu))$ vs. the regressor $\log(\nu)$, where $f(\nu)$ is the spectral density function of the long

memory process and ν is the frequency. The estimate of d is the slope of the linear regression minus 0.5.

Our approach consists of applying the robust regression procedures instead of the classical LS method. Two robust estimators, the least quantile of squares (LQS) and the least trimmed sum of squares (LTS), are employed for estimating the LRD parameter d in the periodogram method. In particular, the robust estimator of d is a slope estimate in the LQS and LTS robust linear regressions of the sample estimates of $\{log(f(\nu))$ vs. $\log(\nu)\}$.

The simulation study below provides comparison between the classical and robust periodogram methods.

4.4 Simulation Study

The advantage of working with simulated datasets in the context of long memory processes, i.e. ARFIMA(p, d, q), is that we know the true long memory parameters. Thus, for each simulated series, classical and robust periodogram methods are applied to detect the LRD pattern, which enables to carry out the detailed comparative study and evaluate the performance of each method.

In our case, several parameters may influence the estimation of parameter d in ARFIMA (p, d, q) models. For examples, the order of autoregressive part, i.e. p, and moving-average part, i.e. q. In particular, the aim of this simulation is to provide the insights regarding the parameter estimation of d from classical and robust periodogram methods with various p and q.

4.4.1 Simulation Description

The performance of each method is measured with respect to two criterions, bias and variance, that are calculated on Monte Carlo (MC) simulations of a long memory process from a given ARFIMA(p, d, q) model. Bias is the average absolute difference between the MC estimates and the true d, i.e. mean$|\hat{\mathrm{d}} - \mathrm{d}|$. Variance reflects the spread of the MC estimates from different simulated time series around the true d, i.e. var$(\hat{\mathrm{d}} - \mathrm{d})$. The method that provides the smallest variance and bias is considered to be the most preferable procedure.

We simulate time series processes from a selection of ARFIMA$(0, d, 0)$, $(1, d, 0)$ and $(1, d, 1)$ models, and compare the results by variance and bias provided by each method. For each sim-

ulated data set, we detect LRD by seven methods, i.e. classical periodogram method, denoted by PER in Table 4.1 and six robust periodogram methods. Regarding robust periodogram methods, we discuss two sets of robust estimators. The first set is based on the least quantile of squares estimation, i.e. the 25-th, 50-th and 75-th percentiles of model residuals that are denoted as $LQS_{0.25}$, $LQS_{0.5}$ and $LQS_{0.75}$ respectively in Table 4.1. The second set is based on minimizing the least trimmed sum of squares with different levels of trimming, i.e. c_n is equal to the first, second and third quartiles. We denote this set of robust estimators as $LTS_{0.25}$, $LTS_{0.5}$ and $LTS_{0.75}$ in Table 4.1 respectively.

4.4.2 Results of Simulation Study

We generate 1000 Monte Carlo simulations of an ARFIMA(p, d, q) process with a sample size of 1000 observations and summarize our findings below for each considered ARFIMA model by comparing two key statistics, bias and variance of the estimated d.

ARFIMA$(0, d, 0)$

We begin by comparing the LRD estimators for the purely fractional integrated ARFIMA $(0, d, 0)$ models that are well examined in the literature (Abraham and Ledolter (2005)). From Table 4.1, we can easily see that the bias and variance of the estimates are a trade off of each other, and no estimator dominates.

In the case of $d = 0.1$, $LQS_{0.75}$ has the lowest bias 0.0001, and its variance is the third lowest 0.015. $LQS_{0.25}$ has the second lowest bias 0.003 and the third highest variance 0.0539. $LTS_{0.75}$ has the third lowest bias 0.0039 and the second lowest variance. The PER provides the least variance and fourth lowest bias. In this case, $LTS_{0.5}$ performs the worst providing largest bias 0.0275 and largest variance 0.1.

At $d = 0.2$, $LQS_{0.75}$ has the second smallest bias 0.0067, and its variance is third smallest. $LQS_{0.25}$ has the smallest bias 0.0002, and third largest variance. $LQS_{0.5}$ has the third smallest bias 0.0094, and forth largest variance. PER has the second largest bias and smallest variance.In this case, $LTS_{0.25}$ and $LTS_{0.5}$ perform the worst. The earlier one provides the second largest bias and the second largest variance, while the latter one provides the largest bias and variance.

When $d = 0.4$, the biases of all methods increased substantially, approximately 10 times more than biases when d =0.1 or 0.2. It looks like when d approaches 0.5, the long range dependence becomes hard to estimate for all methods. In this case, PER has the smallest

variance, but largest variance among all methods. $LQS_{0.75}$ and $LTS_{0.75}$ are relatively good in this case providing the relatively low bias and variance.

Overall, classical periodogram estimator (PER) provides the smallest variance, but its bias is high. On the other hand, $LQS_{0.75}$ is a relatively good robust estimator providing smaller biases, but higher variance than PER in ARFIMA $(0, d, 0)$ processes.

ARFIMA$(1, d, 0)$ **and ARFIMA**$(1, d, 1)$

In this subsection we consider the multiple-parameter cases of ARFIMA $(1, d, 0)$ and ARFIMA $(1, d, 1)$. Those models allow to combine the long and short memory components. Here one element of the estimation error arises through the difficulty in discerning between p and d, since the AR part and long memory parameter d can imply similar patterns of autocorrelation for the first few lags. This agrees with our simulation results presented in table 4.2. Our findings reveals that the biases of all estimators are higher than that in ARFIMA $(0, d, 0)$.

None of estimators dominate in all cases. PER still provides the smallest variance among all estimators, but its bias is highest in all cases except ARFIMA $(1, d, 1)$ with $a_1 = 0.2, d = 0.4, b_1 = 0.4$. All robust methods provide relatively smaller biases, but higher variances compared to PER. In particular, $LQS_{0.25}$ and $LTS_{0.25}$ have the smallest biases, but highest variances. $LQS_{0.75}$ and $LTS_{0.75}$ provides relatively smaller biases and variances.

d	$Bias(d_{PER})$	$Bias(d_{LQS_{0.25}})$	$Bias(d_{LQS_{0.5}})$	$Bias(d_{LQS_{0.75}})$	$Bias(d_{LTS_{0.25}})$	$Bias(d_{LTS_{0.5}})$	$Bias(d_{LTS_{0.75}})$
0.1	0.0066	0.0227	0.0098	0.0001	0.0030	0.0275	0.0039
0.2	0.0145	0.0002	0.0094	0.0067	0.0177	0.0276	0.0099
0.4	0.1368	0.0952	0.1114	0.1256	0.1015	0.1300	0.1287
d	$Var(d_{PER})$	$Var(d_{LQS_{0.25}})$	$Var(d_{LQS_{0.5}})$	$Var(d_{LQS_{0.75}})$	$Var(d_{LTS_{0.25}})$	$Var(d_{LTS_{0.5}})$	$Var(d_{LTS_{0.75}})$
0.1	0.0062	0.0539	0.0252	0.0150	0.0703	0.1000	0.0117
0.2	0.0057	0.0521	0.0239	0.0146	0.0646	0.1971	0.010
0.4	0.0057	0.0621	0.0266	0.0150	0.0646	0.4144	0.012

Table 4.1: Comparative analysis of the classical periodogram and robust periodogram methods for estimation of the long memory parameter d in ARFIMA$(0, d, 0)$ processes with 1000 observations and number of Monte Carlo simulations is 1000.

(a_1, d, b_1)	$Bias(d_{PER})$	$Bias(d_{LQS_{0.25}})$	$Bias(d_{LQS_{0.5}})$	$Bias(d_{LQS_{0.75}})$	$Bias(d_{LTS_{0.25}})$	$Bias(d_{LTS_{0.5}})$	$Bias(d_{LTS_{0.75}})$
$(0.2, 0.2, 0)$	0.0317	0.0084	0.0200	0.0270	0.0004	0.0124	0.0200
$(0.2, 0.4, 0)$	0.0530	0.0267	0.0338	0.0429	0.0044	0.0298	0.0395
$(0.2, 0.2, 0.2)$	0.0170	0.0084	0.0049	0.0072	0.0185	0.0008	0.0069
$(0.2, 0.4, 0.4)$	0.0021	0.0476	0.0361	0.0161	0.0338	0.0159	0.0084
(a_1, d, b_1)	$Var(d_{PER})$	$Var(d_{LQS_{0.25}})$	$Var(d_{LQS_{0.5}})$	$Var(d_{LQS_{0.75}})$	$Var(d_{LTS_{0.25}})$	$Var(d_{LTS_{0.5}})$	$Var(d_{LTS_{0.75}})$
$(0.2, 0.2)$	0.0056	0.0536	0.0249	0.0136	0.060	0.0273	0.0111
$(0.2, 0.4)$	0.0061	0.0522	0.0232	0.014	0.0524	0.0280	0.0118
$(0.2, 0.2, 0.2)$	0.0052	0.0588	0.0246	0.0139	0.0607	0.0284	0.0114
$(0.2, 0.4, 0.4)$	0.0064	0.0633	0.0284	0.0149	0.0672	0.0317	0.0126

Table 4.2: Comparative analysis of the classical periodogram and robust periodogram methods for estimation of the long memory parameter d in ARFIMA$(1, d, 0)$ and $(1, d, 1)$ processes of 1000 observations and number of Monte Carlo simulations is 1000.

Comparing classical and robust periodogram methods for different ARFIMA(p, d, q) processes, we found that none of methods dominate in all cases. One method may be better than the others in terms of bias or variance, but none is better in terms of both bias and variance. Overall, the classical periodogram provides the smallest variance, but higher bias, while robust methods provide relatively smaller biases, but higher variances. In the next subsection, we discuss variance reduction methods that aim to minimize estimation variability of the robust periodogram procedures.

4.5 Variance Reduction Methods

In order to make a simulation statistically efficient, i.e., to obtain less variable estimates for long memory parameter d in ARFIMA (p, d, q), variance reduction techniques can be utilized. The main variance reduction procedures are bootstrap, common random numbers, antithetic variates, control variates, importance sampling and stratified sampling (Li and Winker (2003)). In the next subsection we apply the bootstrap procedure to the robust periodogram methods to reduce the spread of the LRD estimation. The other variance reduction techniques are referred as future work and not examined here.

4.6 Bootstrap Method

Bootstrapping was first introduced by B. Efron (1979) as a computer intensive, yet conceptually simple technique that leverages the collected data to obtain inference about statistic. An important advantage of the bootstrap is that it provides a common paradigm for inference that can be applied both to standard and non-standard statistical analysis. Bootstrap is particularly efficient to analyze data sets of medium size. We apply the bootstrap procedure to the robust periodogram estimators. Based on our simulation results, $LQS_{0.75}$ and $LTS_{0.75}$ are considered to be the best robust periodogram estimators in terms of both biases and variances. Therefore, in this work we apply the bootstrap procedure only to $LQS_{0.75}$ and $LTS_{0.75}$ and omit other robust procedures from the current analysis.

Suppose we obtain $\{\log(f(\nu))$ and $\log(\nu)\}$ from a simulated time series. We assume that $\{\log(f(\nu))$ and $\log(\nu)\}$ are independently and identically distributed (see Brockwell and Davis, 2003). As explained in section (5.1), robust procedures are applied to $\{\log(f(\nu))$ and $\log(\nu)\}$ in

order to estimate the long memory parameter d. For each simulated series, the following steps provide a bootstrap robust periodogram estimate for d.

1. Take a random sample with replacement from $\{log(f(\nu))$ and $log(\nu)\}$.
2. Apply a robust regression method to produce an estimate for d.
3. Resample 100 times and compute d for each new sample.
4. Take the average of all $\hat{d}$.

Tables 4.3 and 4.4 present the results on estimation of d for ARFIMA(p, d, q) processes using the bootstrapped robust regressions. The bootstrap modification of the periodogram method provides competitive results, i.e. estimation variability of $LQS_{0.75}$ and $LTS_{0.75}$ is reduced substantially compared to Tables 4.1 and 4.3. Although the estimation spread of the bootstrapped robust procedures is still somewhat higher than that of the classical periodogram, the robust bootstrap modifications yield noticeably smaller biases for all cases, except of $LTS_{0.75}$ for ARFIMA$(0, 0.1, 0)$. Overall, we can conclude that all three methods are efficient for detection of a long memory parameter d. However, the robust bootstrap modifications of the periodogram method yield a noticeably smaller bias and a comparative variability of estimation.

Bias	d_{PER}	$d_{LQS_{0.75}}$	$d_{LTS_{0.75}}$
$d = 0.1$	0.0074	0.0048	0.0049
$d = 0.2$	0.0152	0.0007	0.0006
$d = 0.4$	0.0388	0.0229	0.0173
Variance	d_{PER}	$d_{LQS_{0.75}}$	$d_{LTS_{0.75}}$
$d = 0.1$	0.0060	0.0074	0.0083
$d = 0.2$	0.0057	0.0066	0.0079
$d = 0.4$	0.0059	0.0067	0.0087

Table 4.3: Comparative analysis of the classical periodogram and bootstraped robust periodogram methods for estimation of the long memory parameter d in ARFIMA$(0, d, 0)$ processes of 1000 observations: bootstrap sample size is 100 and number of Monte Carlo simulations is 1000.

Bias	d_{PER}	$d_{LQS_{0.75}}$	$d_{LTS_{0.75}}$
$a_1 = 0.2, d = 0.2$	0.0351	0.0223	0.0196
$a_1 = 0.2, d = 0.4$	0.0532	0.0429	0.0383
$a_1 = 0.2, d = 0.2, b_1 = 0.2$	0.0152	0.0040	0.0012
$a_1 = 0.2, d = 0.4, b_1 = 0.4$	0.0053	0.0190	0.0206
Variance	d_{PER}	$d_{LQS_{0.75}}$	$d_{LTS_{0.75}}$
$a_1 = 0.2, d = 0.2$	0.0057	0.0066	0.0077
$a_1 = 0.2, d = 0.4$	0.0057	0.0074	0.0088
$a_1 = 0.2, d = 0.2, b_1 = 0.2$	0.0062	0.0072	0.0086
$a_1 = 0.2, d = 0.4, b_1 = 0.4$	0.0061	0.0078	0.0097

Table 4.4: Comparative analysis of the classical periodogram and bootstraped robust periodogram methods for estimation of the long memory parameter d in ARFIMA$(1, d, 1)$ and $(1, d, 1)$ processes of 1000 observations: bootstrap sample size=100 and number of Monte Carlo simulations=1000.

Chapter 5

Testing Long Range Dependence in Foreign Exchange Rates

5.1 Data Description

We examine the long range dependence of four nominal dollar spot rates per US Dollar. We consider British pound (BP), Euro, Swiss frank (SF), and Japanese yen (JY). Daily exchange rates from January 1974 to December 1987 are obtained from the FXHistory website at: http://www.oanda.com/convert/fxhistory. The Euro data are only available from December 15, 1998 to December 30, 2004. Time series plots of the data are presented in Fig. 5.1.

5.2 Preliminary Analysis

We start from the preliminary transformations of the raw data in order to make the time series being weakly stationary. The logarithmic transformation and classical differencing techniques are applied to all data sets. Graphs of the first-differenced log data are presented in Figure 5.2. Exchange-rate changes appear to have mean 0 and constant variance over the sample period, which implies that the filtered (transformed) data are relatively weakly stationary. Indeed, the augmented Dickey-Fuller test for a unit root on each individual exchange-rate change series yields p-values of less than 0.1, which along with the empirical assessment of Figure 5.2 indicates the likely weak stationarity. One should be careful though in relying solely on the Dickey-Fuller test, since as noted by Diebold and Rudebusch (1991a) and Sowell (1990b), this standard unit-

root test has low power for long memory processes.

5.3 Analysis of Results

The following methods are applied to estimate the long memory parameter d: Aggregated Variance method (AV), Differenced Variance Method (DV), classical periodogram method (PER) and bootstrap robust periodogram methods, i.e., bootstrap $LQS_{0.25}$, $LQS_{0.5}$, $LQS_{0.75}$, $LTS_{0.25}$, $LTS_{0.5}$ and $LTS_{0.75}$. All estimates of d are based on $n = T^{0.5}$ where $n = g(T) \ll T$ is the number of harmonic ordinates included in the spectral regression and T is the total sample size (see section (3.2.3)). It is commonly suggested to set the threshold of $n = T^{0.5}$ in detecting of long memory (Cheung (1993)).

We can say there exists an evidence of long memory if the estimate of d is significantly larger than 0. The results from various methods are compared, and our conclusion is that we can not solely rely on a single method. The results supported by most methods can be considered more reliable. Table 5.1 presents the estimates of d from various methods for four major foreign exchange-rate change series, while Table 5.2 presents the standard deviations of $\hat{d}$ in Table 5.1.

For Euro, $LQS_{0.50}$, $LQS_{0.75}$, $LTS_{0.5}$ and $LTS_{0.75}$ suggest that $\hat{d}$ is substantially greater than 0 (i.e., $\hat{d} > 0.28$), whereas the rest five methods suggest that $\hat{d}$ is only slightly greater than 0. AV, $LQS_{0.25}$ and $LTS_{0.25}$ provide similar results (i.e., $\hat{d} \approx 0.05$), but $LQS_{0.25}$ and $LTS_{0.25}$ have the highest variance among all methods, which make their estimates somewhat questionable. Hence, we conclude that there is no strong evidence of long memory in Euro exchange-rate change series, and we need more data to conduct further analysis.

For BP, there is evidence of long memory since the majority methods agree that the estimate of $\hat{d}$ is substantially greater than 0 except for AV, $LQS_{0.25}$ and $LTS_{0.25}$. Hence, it is reasonable to conclude that BP exchange-rate change series has long memory property with $\hat{d} \approx 0.2$.

For JY, AV, DV, $LQS_{0.75}$, $LTS_{0.25}$ and $LTS_{0.75}$ suggest $\hat{d} \approx 0.1$ indicating evidence of long memory. PER, $LQS_{0.5}$ and $LTS_{0.5}$ suggest $\hat{d} \approx 0.05$, while $LQS_{0.25}$ and $LTS_{0.25}$ suggest $\hat{d} < 0$. Hence, long memory property is likely to present in Japanese yen exchange-rate change series as most methods agree.

For SF, there is no strong evidence of long memory. AV, $LQS_{0.5}$, $LQS_{0.75}$, $LTS_{0.5}$ and $LTS_{0.75}$ provide $\hat{d} < 0.05$ while $LQS_{0.25}$ and $LTS_{0.25}$ have $\hat{d} < 0$. Only DV and PER suggest long memory. In this case, we need more data to make a clear conclusion of the existence of long memory property in Swiss frank exchange-rate change series.

5.4 Summary

We examine the long memory property of four major nominal exchange-rate series using various long range dependence (LRD) detection techniques. We show that in many cases tests for LRD do not agree on their results and typically more than one tests are needed to draw a reliable conclusion. By comparing estimates of $\hat{d}$ from various methods, we find evidence of long memory in British pound, and Japan Yen exchange-rate change data, whereas Euro and Swiss frank are difficult to conclude as we need more data to conduct further analysis. Recent studies on the dynamic properties of macroeconomic time series indicate that the long memory property in exchange-rate data may not be an independent incident. The LRD behavior of exchange rates can be related to the dynamic properties of other economic variables, e.g. the movement of relative national prices and consumer price indices, as mentioned by Aduaf and Jorion (1990).

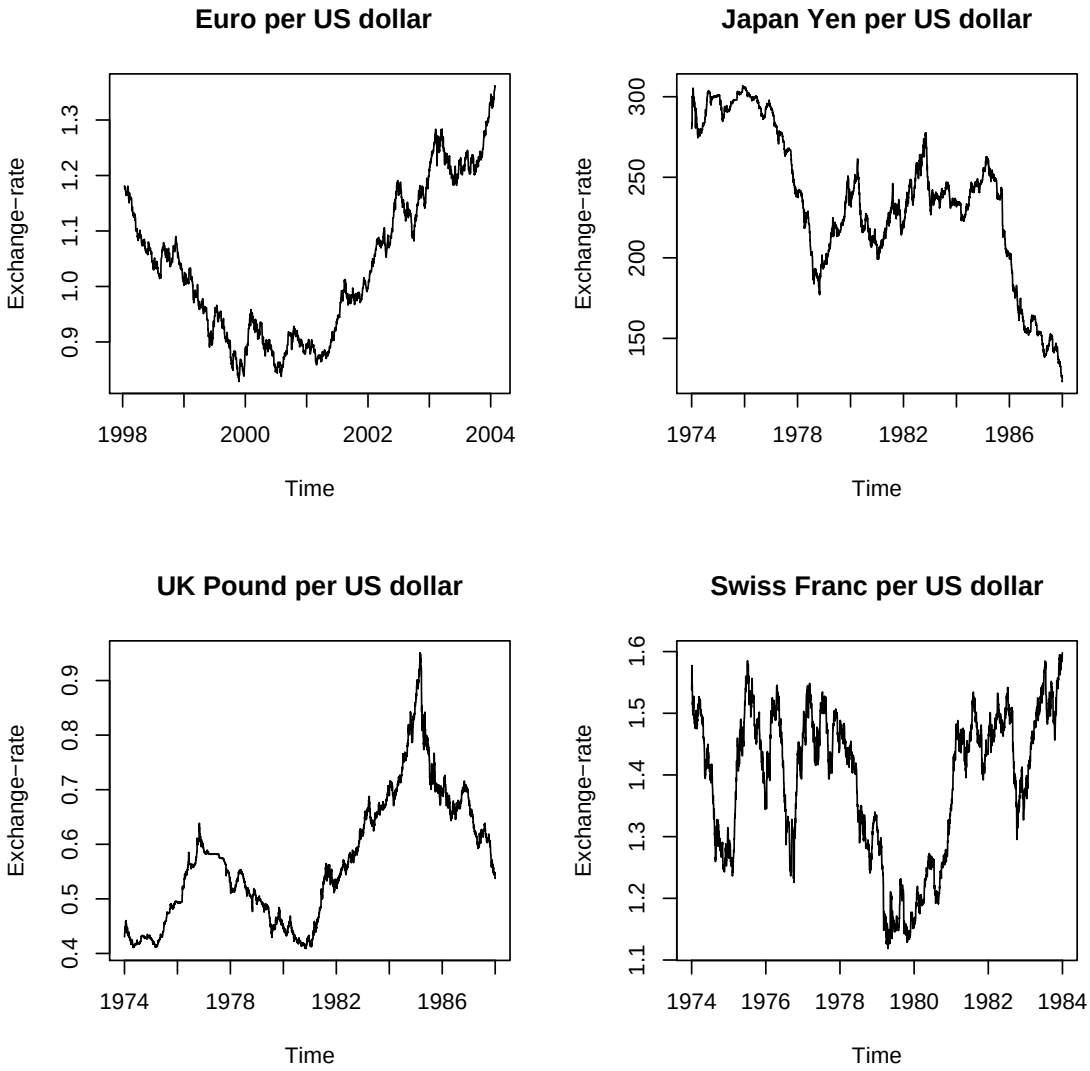

Figure 5.1: Time Series Plots of Four Major Currency Exchange Rates.

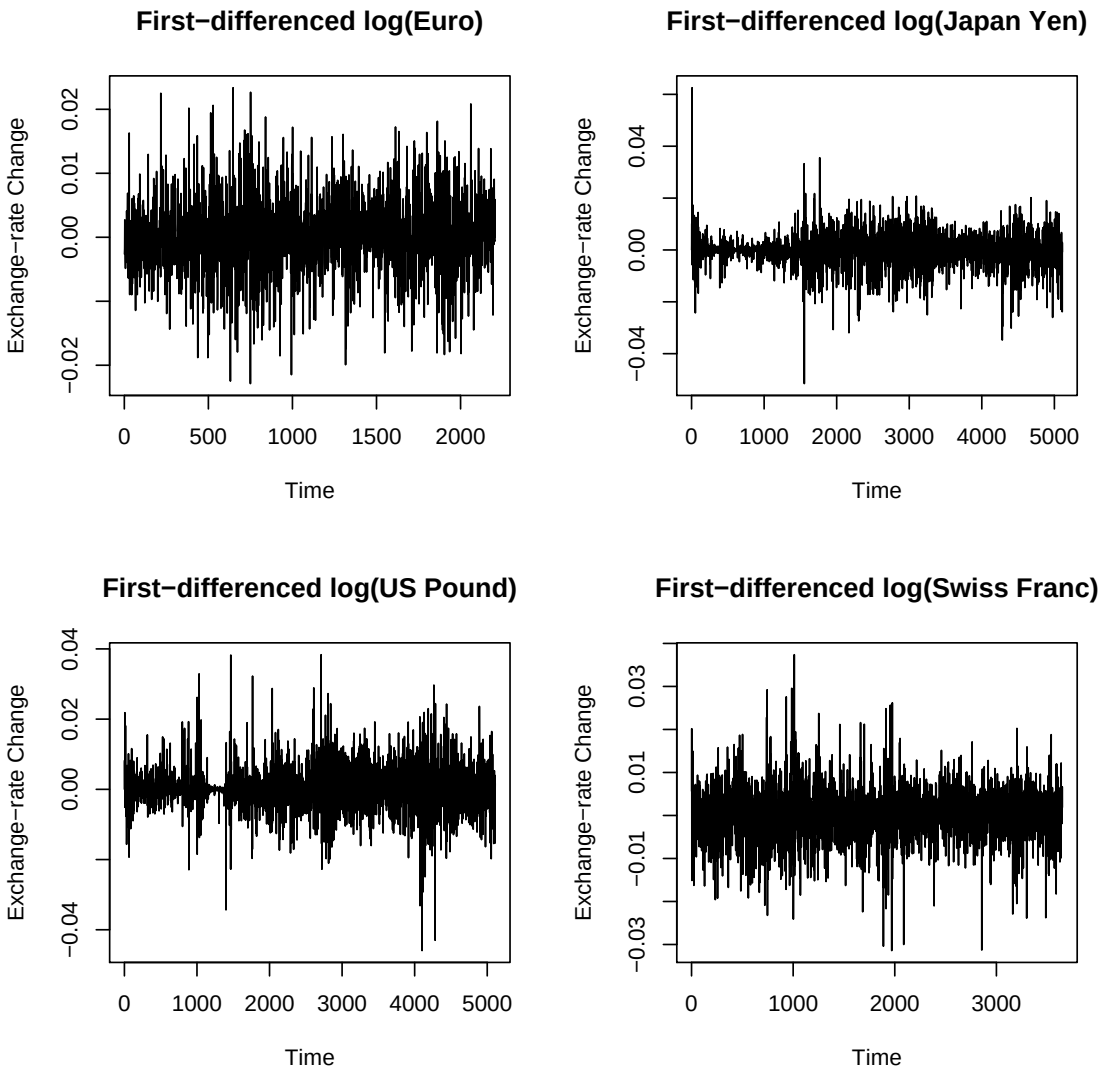

Figure 5.2: Four Major Currency Exchange-Rate Change Series.

	$\hat{d}_{AV}$	$\hat{d}_{DV}$	$\hat{d}_{PER}$	$\hat{d}_{LQS_{0.25}}$	$\hat{d}_{LQS_{0.5}}$	$\hat{d}_{LQS_{0.75}}$	$\hat{d}_{LTS_{0.25}}$	$\hat{d}_{LTS_{0.5}}$	$\hat{d}_{LTS_{0.75}}$
$Euro$	0.0503	0.0575	0.0250	0.0525	0.2894	0.3292	0.0556	0.2877	0.2996
BP	0.0633	0.1824	0.1656	0.0798	0.1606	0.2403	0.0651	0.1907	0.2176
JY	0.0933	0.2619	0.0631	-0.0538	0.0471	0.1303	-0.1124	0.0515	0.1178
SF	0.0191	0.1602	0.1306	-0.0497	0.0174	0.0658	-0.0684	0.0239	0.0107

Table 5.1: Estimations for $\hat{d}$ from various methods for four major exchange-rate change series

	$Sd(\hat{d}_{AV})$	$Sd(\hat{d}_{DV})$	$Sd(\hat{d}_{PER})$	$Sd(\hat{d}_{LQS_{0.25}})$	$Sd(\hat{d}_{LQS_{0.5}})$	$Sd(\hat{d}_{LQS_{0.75}})$	$Sd(\hat{d}_{LTS_{0.25}})$	$Sd(\hat{d}_{LTS_{0.5}})$	$Sd(\hat{d}_{LTS_{0.75}})$
$Euro$	0.0503	0.0575	0.0250	0.0525	0.2894	0.3292	0.0556	0.2877	0.2996
BP	0.0633	0.1824	0.1656	0.0798	0.1606	0.2403	0.0651	0.1907	0.2176
JY	0.0933	0.2619	0.0631	-0.0538	0.0471	0.1303	-0.1124	0.0515	0.1178
SF	0.0191	0.1602	0.1306	-0.0497	0.0174	0.0658	-0.0684	0.0239	0.0107

Table 5.2: Standard deviation for $\hat{d}$ from various methods for four major exchange-rate change series

Chapter 6

Conclusions and Future Work

We now summarize our findings on robust methods of estimating long memory parameter d in ARFIMA (p, d, q). In this thesis a new robust periodogram method for detecting long range dependence is proposed and illustrated. Moreover, the bootstrapped modification of the new robust periodogram test is developed in order to reduce variability of estimation. Currently two robust linear regressions are employed, i.e. the least quantile and the least trimmed methods. In particular, the long memory parameter d is a slope estimate of the robust linear regression of the logarithm of spectral density (i.e., $\log f(\nu)$) of long memory process vs. logarithm of frequency (i.e., $\log(\nu)$). The Monte Carlo simulation show that the robust periodogram estimators $LQS_{0.75}$ and $LTS_{0.75}$ provide the best combination of bias and variance among all robust methods. Compared to the classical periodogram methods, $LQS_{0.25}$ and $LTS_{0.75}$ provide lower biases, but higher variances. In order to reduce the variance, a bootstrap technique is employed for the $LQS_{0.25}$ and $LTS_{0.75}$ methods. The simulation study suggests that the bootstrapped modifications of the $LQS_{0.75}$ and $LTS_{0.75}$ periodogram methods yield the relatively low bias and reduce variability by 10 times compared to the simple robust $LQS_{0.75}$ and $LTS_{0.75}$ periodogram approaches.

In terms of accuracy and efficiency, bootstrap robust estimators, $LQS_{0.75}$ and $LTS_{0.75}$, and the classical periodogram estimator have their own advantages, i.e. the robust techniques provide smaller bias but somewhat higher variance than the classical estimator. In addition, the robust periodogram methods potentially have an advantage of dealing with heavy tailed model residuals. On the other hand, higher computational cost may be an issue for bootstrap versions of the $LQS_{0.75}$ and $LTS_{0.75}$ periodogram estimators. Overall, one can apply apply both the classical and robust bootstrap periodogram techniques in order to ensure the most

accurate detection of the long range parameter in time series.

In the study of time series of four foreign exchange rates, nine estimation methods are applied to the data, i.e. aggregated variance, differenced variance, periodogram and bootstrap robust periodogram methods: $LQS_{0.25}$, $LQS_{0.5}$, $LQS_{0.75}$, $LTS_{0.25}$, $LTS_{0.5}$ and $LTS_{0.75}$. Instead of relying on a single method, we compare results from various methods, and agree with the results supported by majority methods. Therefore, we conclude that based on the date available, long memory property is present in British pound and Japanese yen exchange rate, while it is not obvious in Euro and Swiss frank exchange-rate series. Hence, in order to detect long range dependence, large dataset is required.

The future work includes:

1. To derive asymptotic properties of the robust periodogram estimators and its bootstrapped modifications.

2. To apply the idea of robust regression to other tests for LRD that are based on a linear regression, for example, the aggregated and differenced variance methods, the Geweke and Proter-Hudak, the Higuchi and the aggregated absolute value/moment methods.

3. To investigate methodology that can lead to reduction of spread in robust LRD detection, e.g., control variants and stratified sampling that are currently not used widely in time series analysis.

4. To utilize other methods of robust regression, for example, based on minimizing the median absolute deviation (MAD) from the sample median of residuals.

5. To increase the accuracy of the periodogram estimator. Currently we use the raw periodogram as an estimator of the spectral density. However, it is too rough to be a good estimator in practice. We can try various techniques such as smoothing and tapering.

6. In terms of forecasting long memory time series, more sophisticated approaches can be conducted. For example, we can look into sources causing long memory in the exchange rates and incorporate this information into the estimated ARFIMA model to make more accurate prediction.

Appendix

R code

```
**********************************************************************

#####Importing statistical library in R##########
library(fracdiff)
library(fSeries)
library(MASS)
library(lmtest)


########Simulate an AR(2) process with sample size=1000, a1=0.7,
a2=0.2 and plot its ACF and PACF graphs#######
ar2=arima.sim(list(order = c(2,0,0), ar = c(0.7,0.2)), n = 1000)
acf(ar2,main="A simulated AR(2) process with a1=0.7,a2=0.2, 1000
observations") pacf(ar2,main="A simulated AR(2) process with
a1=0.7,a2=0.2, 1000 observations")

########Simulate an MA(2) process with sample size=1000, b1=0.7,
b2=0.2 and plot its ACF and PACF graphs#######
ma2=arima.sim(list(order = c(0,0,2), ma = c(0.7,0.2)), n = 1000)
acf(ma2,main="A simulated MA(2) process with b1=0.7,b2=0.2, 1000
observations")
pacf(ma2,main="A simulated MA(2) process with
b1=0.7,b2=0.2, 1000 observations")

########Simulate an ARMA(1,1) process with sample size=1000, a1=0.5,
a2=0.2 and plot its ACF and PACF graphs#######
arma11=arima.sim(list(order = c(1,0,1), ar = 0.5, ma=0.2), n = 1000)
```

```
acf(arma11,main="A simulated ARMA(1,1) process with a1=0.5,b2=0.2,
1000 observations")
pacf(arma11,main="A simulated ARMA(1,1) process
with a1=0.5,b2=0.2, 1000 observations")

########Simulate an ARIMA(1,1,1) process with sample size=1000,
a1=0.7, d=1, b1=0.2 and plot time series graphs of the raw data and
first-differenced data graghs#######
arima=arima.sim(list(order =
c(1,1,1), ar = 0.7, ma=0.2), n = 1000)
ts.plot(arima, main="Time
Series plot of a simulated ARIMA(1,1,1) process")
ts.plot(diff(arima),main="Its first differenced series")

########Simulate an ARFIMA(1,1,1) process with sample size=1000,
a1=0.5, d=0.4, b1=0.2 and plot its ACF and PACF graghs#######
arfima=farimaSim(n=1000,
model=list(ar=0.5,d=0.4,ma=0.2),method=c("time")) acf(arfima,main="A
simulated ARFIMA(1,1,1) process with a1=0.7,d=0.4,b2=0.2, 1000
observations") pacf(arfima,main="A simulated ARFIMA(1,1,1) process
with a1=0.7,d=0.4,b2=0.2,1000 observations")

##Run aggregated variance method to estimate d##
aggvarFit(arfima)
##Run differenced variance method to estimate d##
diffvarFit(arfima)
##Run periodogram method to estimate d##
perFit(arfima)


#####Monte Carlo simulation with 1000 iterations################
**********Simulate ARFIMA(0,1,0) processes with
$d=0.1$*****************************
N=1000
per=double(N)
lqs25=double(N)
lqs50=double(N)
lqs75=double(N)
lts25=double(N)
lts50=double(N)
```

```
lts75=double(N)

for (i in 1:N) {

    cut.off=0.1
    x1=farimaSim(n=1000, model=list(ar=0,d=0.1,ma=0),method=c("time"))
    data = list(x = x1)
    x = as.vector(x1)
    n = length(x)
    FFT = Mod(fft(x))^2/(2 * pi * n)
    pgram = FFT[1:(n%/%2 + 1)]
    N = length(pgram)

    X = (pi/n) * c(2:((n * cut.off)))
    Y = pgram[2:((n * cut.off))]
    fitH = lsfit(log10(X), log10(X/Y)/2)
    fit1 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=n/4)
    fit2 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=n/2)
    fit3 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=98/4*3)
    fit4 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=n/4)
    fit5 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=n/2)
    fit6 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=98/4*3)
    per[i]=fitH$coef[[2]]-0.5       ####a vector containing 1000 $\hat{d}_{PER}$#############
    lts25[i]=fit1$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.25}}$#####
    lts50[i]=fit2$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.5}}$######
    lts75[i]=fit3$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.75}}$#####

    lqs25[i]=fit4$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.25}}$#####
    lqs50[i]=fit5$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.5}}$#####
    lqs75[i]=fit6$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.75}}$#####
}


****************Simulate ARFIMA(1,1,0) processes with
$a_{1}=0.2,d=0.2$*****************************

N=1000
per=double(N)
lqs25=double(N)
lqs50=double(N)
```

```
lqs75=double(N)
lts25=double(N)
lts50=double(N)
lts75=double(N)

for (i in 1:N) {

    cut.off=0.1
    x1=farimaSim(n=1000, model=list(ar=0.2,d=0.2,ma=0),method=c("time"))
    data = list(x = x1)
    x = as.vector(x1)
    n = length(x)
    FFT = Mod(fft(x))^2/(2 * pi * n)
    pgram = FFT[1:(n%/%2 + 1)]
    N = length(pgram)

    X = (pi/n) * c(2:((n * cut.off)))
    Y = pgram[2:((n * cut.off))]
    fitH = lsfit(log10(X), log10(X/Y)/2)
    fit1 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=n/4)
    fit2 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=n/2)
    fit3 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=98/4*3)
    fit4 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=n/4)
    fit5 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=n/2)
    fit6 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=98/4*3)
    per[i]=fitH$coef[[2]]-0.5       ####a vector containing 1000 $\hat{d}_{PER}$#############
    lts25[i]=fit1$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.25}}$#####
    lts50[i]=fit2$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.5}}$######
    lts75[i]=fit3$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.75}}$#####

    lqs25[i]=fit4$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.25}}$#####
    lqs50[i]=fit5$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.5}}$#####
    lqs75[i]=fit6$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.75}}$#####
}

****************Simulate ARFIMA(1,1,1) processes with
$a_{1}=0.2,d=0.4,b_{1}=0.4$*****************************

N=1000
per=double(N)
```

```
lqs25=double(N)
lqs50=double(N)
lqs75=double(N)
lts25=double(N)
lts50=double(N)
lts75=double(N)

for (i in 1:N) {

    cut.off=0.1
    x1=farimaSim(n=1000, model=list(ar=0.2,d=0.4,ma=0.4),method=c("time"))
    data = list(x = x1)
    x = as.vector(x1)
    n = length(x)
    FFT = Mod(fft(x))^2/(2 * pi * n)
    pgram = FFT[1:(n%/%2 + 1)]
    N = length(pgram)

    X = (pi/n) * c(2:((n * cut.off)))
    Y = pgram[2:((n * cut.off))]
    fitH = lsfit(log10(X), log10(X/Y)/2)
    fit1 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=n/4)
    fit2 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=n/2)
    fit3 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lts"),quantile=98/4*3)
    fit4 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=n/4)
    fit5 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=n/2)
    fit6 = lqs(log10(X), log10(X/Y)/2,intercept = TRUE, method = c("lqs"),quantile=98/4*3)
    per[i]=fitH$coef[[2]]-0.5       ####a vector containing 1000 $\hat{d}_{PER}$#############
    lts25[i]=fit1$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.25}}$#####
    lts50[i]=fit2$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.5}}$######
    lts75[i]=fit3$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LTS_{0.75}}$#####

    lqs25[i]=fit4$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.25}}$#####
    lqs50[i]=fit5$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.5}}$#####
    lqs75[i]=fit6$coef[[2]]-0.5     ####a vector containing 1000 $\hat{d}_{LQS_{0.75}}$#####
}

#########################################################################################################################

#####Applying bootstrap robust methods on a series with bootstrap
```

```
sample=100##### #####Return a vector of $\hat{d}$ with length
100##### robustd=function(series){
    data = list(x = series)
    x = as.vector(series)
    n = length(x)
    FFT = Mod(fft(x))^2/(2 * pi * n)
    pgram = FFT[1:(n%/%2 + 1)]
    N = length(pgram)

    X = (pi/n) * c(2:(n^0.5))
    Y = pgram[2:((n^0.5))]
    data=cbind(log10(X),log10(X/Y)/2)
    n=length(log10(X))
    per=double(100)

lqs25=double(100) lqs50=double(100) lqs75=double(100)
lts25=double(100) lts50=double(100) lts75=double(100)

for(b in 1:100) {
    data.bs <- data[sample(1:n,n,replace=T),]
    fit1 = lsfit(data.bs[,1], data.bs[,2])
     fit2 = lqs(data.bs[,1], data.bs[,2],intercept = TRUE, method = c("lqs"),quantile=n*1/4)
    fit3 = lqs(data.bs[,1], data.bs[,2],intercept = TRUE, method = c("lqs"),quantile=n*2/4)
    fit4 = lqs(data.bs[,1], data.bs[,2],intercept = TRUE, method = c("lqs"),quantile=n*3/4)

    fit5 = lqs(data.bs[,1], data.bs[,2],intercept = TRUE, method = c("lts"),quantile=n*1/4)
    fit6 = lqs(data.bs[,1], data.bs[,2],intercept = TRUE, method = c("lts"),quantile=n*2/4)
    fit7 = lqs(data.bs[,1], data.bs[,2],intercept = TRUE, method = c("lts"),quantile=n*3/4)
    per[b]=fit1$coef[[2]]-0.5       ##a vector containing 100 bootstrap $\hat{d}_{PER}$##
    lqs25[b]=fit2$coef[[2]]-0.5     ##a vector containing 100 bootstrap $\hat{d}_{LQS_{0.25}}$##
    lqs50[b]=fit3$coef[[2]]-0.5     ##a vector containing 100 bootstrap $\hat{d}_{LQS_{0.5}}$##
    lqs75[b]=fit4$coef[[2]]-0.5     ##a vector containing 100 bootstrap $\hat{d}_{LQS_{0.75}}$##
    lts25[b]=fit5$coef[[2]]-0.5     ##a vector containing 100 bootstrap $\hat{d}_{LTS_{0.25}}$##
    lts50[b]=fit6$coef[[2]]-0.5     ##a vector containing 100 bootstrap $\hat{d}_{LTS_{0.5}}$##
    lts75[b]=fit7$coef[[2]]-0.5     ##a vector containing 100 bootstrap $\hat{d}_{LTS_{0.75}}$##
  }

}
```

References

Abraham, B. and Ledolter, J. (2005) *Statistical methods for forecasting*, Wiley, New York.

Abuaf, N. and Jorion, P. (1990), "Purchasing power parity in the long run," *Journal of Finance*, **45**, 157-174.

Barkoulas, J. and Baum, F. B. (1996), "Long term dependence in stock returns," *Economics Letters*, **53**, 253-259.

Beran, J. (1989), "A test of location for data with slowly decaying serial correlations," *Biometrika*, **76**, 261-269.

BloomFileld, P. (1976), *Fourier Analysis of Time Series: An Introduction.* Wiley, New York.

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1994), *Time Series Analysis: Forecasting and Control.* Prentice-Hall, Inc., New Jersey.

Brockwell, P. J. and Davis, R. A. (2002), *Introduction to Time Series and Forecasting.* Springer, New York.

Cheung, Y. W. (1993), "Long memory in foreign-exchange rates," *Journal of Business and Economic Statistics,* **11**, 93-101.

Diebold, F. X. and Rudebusch, G. D. (1991), "On the power of dickey-fuller tests against fractional alternatives," *Economics Letters*, **35**, 155-160.

Efron, B. (1979), "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics,* **7**, 1-26.

Granger, C. W. J. and Joyeux R. (1980), "An introduction to long-memory time series models and fractional differencing," *Time Series Analysis*, **1**, 15-29.

Hosking, J. R. M. (1981), "Fractional differencing," *Biometrika*, **68**, 165-176.

Hurst, H. E. (1951), "Long term storage capacity of reservoirs," *American Society of Civil Engineers,* **116**, 770-808.

Li X. and Winker, P. (2003), "Time Series Simulation with Quasi Monte Carlo Methods," *Computational Economics*, **21**, 23-43

Ljung, L. (1998), *System Identification: Theory for the User*, 2nd Edition, Prentice Hall, New Jersey.

Mandelbrot and Wallis, J. R. (1968), "Noah, joseph and operational hydrology," *Water Resources Research,* **4**, 909-918.

Montgomery, D. C. and Johnson, L. A. and Gardiner, J. (1988), *Forecasting and time series analysis.* McGraw-Hill, New York.

Shumway, R. H. (1988), *Applied Statistical Time Series Analysis.* Englewood Cliffs: Prentice Hall, New Jersey.

Sowell, F. (1990), "Fractional unit root distribution," *Econometrica*, **50**, 495-505.

Stuart, A. and Ord K. (1987), *Kendall's Advanced Theory of Statistics*, 5th edition, Oxford University Press, New York.

Teverovsky, V. and Taqqu, M. (1997), "Testing for long-range dependence in the presence of shifting means or a slowly declining trend, using a variance-type estimator," *Journal of Time*

Series Analysis, **18**, 279-304.

Toussoun, O. (1925), "Memoire sur l'histoire du nil," *Memoires a l'Institut d'Egypte*, **18**, 366-404.

Tsay, R. S. (2005), *Analysis of Financial Time Series*, 2nd edition, Wiley, New York.

Whitcher, B. and Byers, S. and Guttorp, P. and Percival, D. (2002), "Testing for homogeneity of variance in time series: Long memory, wavelets and the nile river," *Water Resources Research*, **38**, no. 5, pp. 10.1029/2001WR000509.

Weisberg, S. (1985), *Applied Linear Regression*, 2nd edition, Wiley, New York.

Wilcox, R. R. (1997), *Introduction To Robust Estimation and Hypothesis Testing,* Academic Press, California.