Modelling volatility in a Time Series Data

Project Report

SYDE 631: Time Series Modelling

Author:

Harshwin Venugopal

Master of Engineering

Submitted to:

Professor Keith Hipel

Department of Systems Design Engineering

University of Waterloo

Waterloo, ON N2L 3G1, Canada

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to Professor Keith W. Hipel for his exemplary guidance and encouragement throughout the course. His excellent teaching skills and insights about the subject, helped me gain underlying theory quickly and strengthen the mathematical background through his well thought assignments. The help and guidance given by him time to time shall carry me a long way in my ability to develop time series applications, which I might embark upon in future.

Abstract

In this report, we analyze the usability of ARCH and GARCH model to portray the volatility of the data. I real life, many data contain volatility. Especially in financial data. This report clearly mentions the steps to analyse the data and check for the presence of volatility and finally apply ARCH and GARCH models, using a time series data of currency exchange rate of Brazillion Real to US Dollars.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Exchange rate is the factor of conversion, from one countries currency to another. Exchange rates of most countries fluctuates in small instances of time. This is set by foreign exchange market using the supply and demand for that particular currency by comparing to the other currency. They trade 24 hours a day, seven days a week. This explains the fluctuations in the rate. Exchange rates tell how much the currency is worth with in a foreign currency. Similar to price being charged to purchase that currency.

The countries currency depends on the country's proceedings. ***Interest Rate*** paid by country's central bank is a huge factor. The higher is the interest rate, more valuable is that currency. ***Money supply*** created by that country's central bank, is another factor. If the government prints too much currency, then there is too much of it chasing very few goods. If people have more currency, they would bid up the prices of goods and services. That would create *inflation*. Some people would invest overseas, where there is no inflation. But there might be not much demand for their currency since there is so much of it. That is one of the reason's inflation will reduce the value of the currency. Also, Country's financial stability and economic growth impacts its exchange rates. Investors will buy its goods and services if the country is growing strong economically. From theoretical economics point of view, exchange rates between two countries offset the differences in inflation rate between them. In other words, in an efficient international economy, exchange rate would give each currency same purchasing power in its economy (Principal Power Parity).

Predicting and analyzing a country's exchange rate not only helps in investing, for travel purposes, it also helps in predicting how well the country is doing with respect to others. Time series is the historical representation of data points which are collected at periodic intervals over time. We try to analyze the internal structure of this time series data, to gain some insights about the fluctuations and model it using some Time series models. So that, it we get better insights of the data and hence some very useful intuition about the country. The time series models, help us obtain an understanding of the underlying forces and structure that produced the observed data and fit a model and perhaps to forecast, monitor or even give a feedback or control. Here in this report, I will be using US dollars(USD) to Brazilian real(BRL) as the time series data. The data is obtained by scrapping the online currency exchange websites. Here I have used from this source[4]. The data contains exchange rates of USD to BRL, from February 2000 to August 2017, taking only the average of exchange rate for a day. The data is saved in *'.csv'* format(comma separated values) to load it to R environment easily. R is an open source software and programming language used for statistical computing. Brazil is a developing country and hence there may be some fluctuations in exchange rate due to political instabilities. Investments in a country are highly dependent on volatility of the exchange rate with the currency of that country. The plot of USD/BRL data shows a high volatility and hence higher risks of investment.

1.1 Key objectives

The key objectives of the present study can be divided broadly in three categories:

1. To provide a graphical descriptive analysis of the observed daily exchange rates (exploratory data analysis).

2. To fit appropriate time series models and to see why some models fail (confirmatory data analysis).

3. To forecast using the best fitted time series model(s).

1.2 Organization of the report

This report is organized as follows. In Chapter 2, an overview of the analysis methodology is presented. Chapter 3 and 4 contain complete and detailed numerical study on exploratory and confirmatory data analysis respectively. Chapter 5 provides conclusion, future applications and insights that can be drawn from the analysis.

Chapter 2

Background

2.1 Introduction

In this study, we try to model ARCH and GARCH models, which have become very important tools in the analysis of financial time series, due to the reasons, portrayed in the next sections. These models are useful when the goal of the study is to analyze and forecast volatility. A great workhorse of econometrics is the least the least squares model[5]. Basic foundation of least squares model is that expected value of all error terms when squared is the same at any given instance of time. This assumption is called *Homoskedasticity.* This is the building block of ARCH/GARCH models. On the contrary, the data in which the variances of the error terms are not equal, in which the error terms may reasonably be expected to be larger for some points than others are said to be *Heteroskedasticity.* Instead of considering this as a problem, the ARCH and GARCH models treat Heteroskedasticity as a variance to be modeled. Hence, not only the deficiencies of the least squares corrected, but variance of each error terms are predicted. This is particularly common in finance.

To understand Heteroskedasticity better, we need to know the difference between conditional and unconditional variance. The conditional variance of a time series is just its variance, given the past values. Commonly referred to as Volatility. On the other hand, the unconditional variance of the time series data is its variance without considering the past data. The class of Heteroskedastic models is a generalization of the models by relaxing the assumption of constant conditional variance of the given time series. Implying to model a non linear time series. ARMA (Auto-Regressive-Moving-Average), a classic time series model can not capture this behaviour because, its variance is considered as constant. To predict the exchange rate, we need to consider more general models in which both the conditional expectation and conditional variance depend on the past data. The reason is, if we use ARMA model, the residual process which is assumed to be white in ARMA, will not be white in this case, due to time varying components of variance. A straight forward approach to estimate the conditional variance is using non-parametric regression. Modelling Heteroskedasticity is very important in the analysis of financial and economic applications, especially when the goal is to analyze and forecast volatility.

2.1.1 ARCH Model:

One of the simplest model to model conditional variance is ARCH model. Suppose we fit an ARMA model, where the residual process is Z_t which is assumed to white noise with constant variance given by σ^2. But in our case, the data exhibits, volatile clustering, where it shows periods of high volatility and periods of low volatility, indicating non-constant variance. In that case, error Z_t is rewritten as the product of deterministic but time varying component σ_t and the random process R_t with mean of 0 and variance 1. We try to model the volatility of σ_t. Autoregressive conditional heteroskedasticity (ARCH) was introduced by Robert Engle in 1982. For this contribution, he was honored with Nobel prize for economics in 2003. Arch model assume the variance of the current error or innovation to be function of previous time periods error terms. The variance is related to the square of the previous innovations. ARCH family models assume that the volatility or σ_t^2 is a linear function with lagged values of the mean equation errors Z_t. Thus on the whole, time-series dynamic of volatility is like AR process.

Time series analysis of datasets involves two major steps: exploratory data analysis and confirmatory data analysis. The confirmatory data analysis consists of the three stages of model construction: identification of parameters, estimation of parameters and diagnostic check of residuals. After time series modelling, the most important applications are forecasting and simulation. The complete methodology of the time series analysis procedure can be found in reference Hipel and McLeod (1994). Description about various statistical procedures or tests can be found in reference [2]. The following sections, give a detailed picture of method used to model this data.

2.1.2 GARCH Model:

As the name suggests, Generalized ARCH Model (GARCH) is the extension of ARCH model. Instead of the model assuming positive and negative shocks have the same effect on volatility as it takes into consideration only square of the previous shocks. As positive and negative shocks may behave differently in nature. Also, the ARCH model might overpredict the volatility as they respond slowly to large isolated shocks because lagged variance effects disappear quickly. Lastly, ARCH model does not give new insight for understanding the source of variations of the series. The intuition behind GARCH model is that it is a weighted average of past residual, but it has a decaying asymptotic weights, which never goes to zero. Hence the model captures the impact of lagged variance over a long time. This model was introduced by Bollerslev in 1986.

2.2 Exploratory data analysis: methodology

The exploratory data analysis consists of several steps. The complete explanation about the exploratory data analysis can be found in Hipel and McLeod (1994). The first step is to plot the original observations against time. By visual inspection of the time series data, we can easily detect the seasonality, nonstationarity, changing variance, extreme values, correlation and long term cycles. Seasonality in a data can be easily modelled by using seasonal ARMA or ARIMA(Auto-Regressive-Integrated-Moving-Average) models. The nonstationarity present in the data can often be removed using seasonal and/or nonseasonal differencing. The seasonal and nonseasonal correlation in the time series can be easily modelled by appropriately deciding upon which AR (Autoregressive) and MA (moving average) parameters should be included in the time series model. If the variance of the series changes over time (heteroscedastic), we can solve this problem by invoking Box-Cox transformation. Also by using ARCH and GARCH models.

A wide range of statistical tests are available for checking stationarity/nonstationarity in a data set. In this study, we have used three methods for checking nonstationarity: Augmented Dickey Fuller (ADF) test, Phillips-Perron (PP) test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. A short description of these tests are provided in the subsequent sections.

The components of the ARIMA model can be designed by visual inspection of autocorrelation function (ACF), partial autocorrelation function (PACF), inverse autocorrelation function (IACF) and inverse partial autocorrelation function (IPACF). In this project we have used only ACF and PACF for determining the components of the time series models.

2.2.1 Tests for checking Stationarity/Nonstationarity

Augmented Dickey Fuller (ADF) test

A full description about the ADF test can be found in reference [2]. In statistics, an augmented Dickey-Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. The alternative hypothesis is stationarity or trend-stationarity. It is an augmented version of the Dickey-Fuller test for a larger and more complicated set of time series models. The augmented Dickey-Fuller (ADF) statistic, used in the test, is a negative number. The more negative it is, the stronger the rejection of the hypothesis that there is a unit root at some level of confidence. The testing procedure for the ADF test is the same as for the DickeyFuller test but it is applied to the model as before.

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t \tag{2.1}$$

where α is a constant, β is the coefficient on a time trend and p is the lag order of the AR process. By including lags of the order p the ADF formulation allows for higher-order autoregressive processes. This means that the lag length p has to be determined when applying the test. The unit root test is then carried out under the null hypothesis $\gamma = 0$ against the alternative hypothesis of $\gamma \neq 0$. Once a value for the test statistic

$$DF_\tau = \frac{\hat{\gamma}}{SE(\hat{\gamma})} \tag{2.2}$$

is computed, it can be compared to the relevant critical value for the Dickey-Fuller test. If the test statistic is less than the critical value, then the null hypothesis of $\gamma = 0$ is rejected and no unit root is present. The intuition behind the test is that if the series is integrated then the lagged level of the series (y_{t-1}) will provide no relevant information in predicting the change in y_t besides the one obtained in the lagged changes (Δy_{t-k}). In this case the $\gamma = 0$ and null hypothesis is not rejected.

Phillips-Perron (PP) test

The full description about the PP test can be found in reference [2]. However, a compact description about the method is presented here. In statistics, the Phillips-Perron test is a unit root test. That is, it is used in time series analysis to test the null hypothesis that a time series is integrated of order 1. It builds on the Dickey-Fuller test of the null hypothesis $\rho = 1$ in $\Delta y_t = \rho y_{t-1} + u_t$, where Δ is the first difference operator. Like the augmented Dickey-Fuller test, the Phillips-Perron test addresses the issue that the process generating data for y_t might have a higher order of autocorrelation than is admitted in the test equation, making y_{t-1} endogenous and thus invalidating the Dickey-Fuller t-test.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test

In statistics, Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests [2] are used for testing a null hypothesis that an observable time series is stationary around a deterministic trend (i.e. trend-stationary) against the alternative of a unit root. Contrary to most unit root tests, the presence of a unit root is not the null hypothesis but the alternative. Additionally, in the KPSS test, the absence of a unit root is not a proof of stationarity but, by design, of trend-stationarity. This is an important distinction since it is possible for a time series to be non-stationary, have no unit root yet be trend-stationary. In both unit root and trend-stationary processes, the mean can be growing or decreasing over time; however, in the presence of a shock, trend-stationary processes are mean-reverting (i.e. transitory, the time series will converge again towards the growing mean, which was not affected by the shock) while unit-root processes have a permanent impact on the mean (i.e. no convergence over time). The series is expressed as the sum of deterministic trend, random walk, and stationary error, and the test is the Lagrange multiplier test of the hypothesis that the random walk has zero variance.

2.2.2 Autocorrelation function (ACF)

The sample ACF measures the amount of linear dependence between observations in a time series that are separated by some time lags. The expression for sample autocovariance

(c_k) of the time series $\{z_t\}$ at lag k can be given as

$$c_k = \frac{1}{N}\sum_{t=1}^{N-k}(z_t - \bar{z})(z_{t+k} - \bar{z}) \tag{2.3}$$

where, N is the total number of data points and $\bar{z}$ is the mean of the time series $\{z_t\}$. Now, the expression of the sample ACF can be written as

$$r_k = \frac{c_k}{c_0} \tag{2.4}$$

Because the ACF is symmetric about lag zero, it is only required to plot the sample ACF for positive lags except for lag zero, to a maximum lag of about $N/4$.

2.2.3 Partial autocorrelation function (PACF)

In time series analysis, the partial autocorrelation function (PACF) gives the partial correlation of a time series with its own lagged values, controlling for the values of the time series at all shorter lags. It contrasts with the autocorrelation function, which does not control for other lags. This function plays an important role in data analyses aimed at identifying the extent of the lag in an autoregressive model. The use of this function was introduced as part of the Box - Jenkins approach to time series modelling, where by plotting the partial autocorrelation functions one could determine the appropriate lags p in an AR (p) model. PACF can be estimated using Yule-Walker equations as

$$[P]\{\phi\} = \{\rho\} \tag{2.5}$$

where $[P]$ is the autocorrelation matrix, $\{\phi\}$ is the vector of AR parameters and $\{\rho\}$ is the autocorrelation column vector.

How to identify the orders of AR(p) and MA(q) processes?

Identification and estimation of AR and MA parameters are essentially important for reasonable model simulation and future predictions. In the process of calibration, the identification of the number of AR and MA parameters along with their estimation is performed. The number of AR parameters and AR process behaviour can be characterized by sample ACF and sample PACF if they exhibit the following patterns:

1. Exponential or dampened sinusoidal decaying of sample ACF.
2. The number of significant PACFs indicates the number of AR parameters to be included in the model.
3. For a pure AR model, the sample PACF truncates and is not significantly different from zero after lag p.

Similarly, the number of MA parameters and its process can be characterized by following attributes:

1. Exponential or dampened sinusoidal decaying of sample PACF.
2. The number of significant ACFs indicates the number of MA parameters to be included in the model.
3. For a pure MA model, the sample ACF truncates and is not significantly different from zero after lag q.

2.2.4 Estimation of AR and MA parameters

There are different techniques available for parameter estimation. In this study, maximum likelihood estimation (MLE) method is used for estimating the parameters of AR and

MA processes. In the following section, a general description of the maximum likelihood method is presented.

Method of Maximum Likelihood (ML)

The method of maximum likelihood is a procedure for directly deriving the point estimator(s) of the parameter(s). The objective is to find the distribution parameter values that maximize the likelihood of obtaining the set of observations. Consider a random variable X with density function $f_X(x;\xi)$, where ξ is the parameter to be estimated. Assuming random sampling, the likelihood of obtaining n independent observations is given by the likelihood function

$$L(x_1, x_2, \cdots, x_n; \xi) = f_X(x_1;\xi) f_X(x_2;\xi) \cdots f_X(x_n;\xi) \tag{2.6}$$

The likelihood is maximized by taking the derivative with respect to ξ and setting it equal to zero as

$$\frac{\delta L(x_1, x_2, \cdots, x_n; \xi)}{\delta \xi} = 0 \tag{2.7}$$

Because of the multiplication in the likelihood function, it is sometimes more convenient to maximize the log-likelihood function as

$$\frac{\delta \ln L(x_1, x_2, \cdots, x_n; \xi)}{\delta \xi} = 0 \tag{2.8}$$

For probability density functions with two or more parameters, the ξ parameter becomes a vector of collection of parameters and the likelihood is maximized by taking the derivative with respect to all the parameters separately and equating them to zero.

The solution from the maximum likelihood method is unique. Maximum likelihood estimates usually satisfy all the properties of the point estimators: unbiasedness, consistency, efficiency (smallest variance) and sufficiency (ML method uses all the sample data). For the estimation of AR and MA parameters, the ξ vector will contain the ϕ_i's, θ_i's and/or Φ_i's, Θ_i's and the maximum likelihood has to be maximized with respect to these parameters.

2.2.5 Model selection criteria

There are different measures available for selection of appropriate models among different candidate models based on the purpose of the model. In this study, Akaike Information Criterion (AIC) is used as a selection criteria for candidate models.

Akaike Information Criterion (AIC)

The expression for the AIC can be written as

$$AIC = -2 \ln ML + 2k \tag{2.9}$$

where ML denotes maximum likelihood, ($\ln ML$) is the value of the maximized log likelihood function for a model fitted to a given data set, and k is the number of independently adjusted parameters within the model. When there are several models available for modelling a given time series, the model that possesses the minimum value of the AIC should be selected. This procedure is referred to as minimum AIC estimation (MAICE).

Table 2.1: Violations of the residual assumptions and the corrective actions

Violations of residual assumptions	Corrective measures
Dependence and non-whiteness	Consider other models
Heteroscedasticity (variance change)	Box-Cox data transformation
Non-normality	Box-Cox data transformation

2.3 Diagnostic checking

The most useful and informative diagnostic checks deal with determining whether or not the assumptions underlying the innovation series are satisfied by the residuals of the calibrated time series model. When fitting a model to a time series the estimated innovations or residuals are assumed to be independent, homoscedastic (constant variance) and normally distributed. Among these three assumptions, independence and whiteness is by far the most important.

If there are violations of the residual assumptions, we have to take corrective measures against those problems. Firstly, if the residuals are dependent and is not a white noise, it is not possible to correct this problem by a data transformation. Rather, the identification and estimation stages must be repeated in order to determine a suitable model. If the less important assumptions of homoscedasticity and normality are violated, they can often be corrected by a Box-Cox transformation of the data. Table 2.1 lists the main problems regarding violations of the residual assumptions and their corrective measures. Various statistical tests are available for diagnostic checking and some of them, which are used in this study, are listed below.

2.3.1 Whiteness test

A number of statistical tests are available for determining whether or not the residuals are uncorrelated or white. In this study, the graph of Residual Autocorrelation Function (RACF) is employed to check for whiteness of the innovations or residuals.

Residual Autocorrelation Function (RACF)

The RACF at lag k is computed as

$$r_k(\hat{a}) = \sum_{t=k+1}^{n} \frac{\hat{a}_t \hat{a}_{t-k}}{\sum_{i=1}^{n} \hat{a}_i^2} \tag{2.10}$$

The value of RACF can range between -1 to +1. Also, RACF is symmetric about lag zero, hence, one can plot the RACF against lags for positive lags from lag one to about lag $n/4$. RACF is normally distributed with zero mean and variance of $1/n$. Therefore, we can plot $+1.96/\sqrt{n}$ and $-1.96/\sqrt{n}$ above and below the lag axis to draw the 95% confidence interval. A more accurate derivation for the RACF is also available in (Hipel and McLeod, 1994), but this methodology is not used in this study.

The most important values of the RACF to examine are those located at the first few lags for nonseasonal data. If one or more of the values of the RACF fall outside the 95% confidence interval, this means that the current model is inadequate.

2.3.2 Constant variance test

Changing variance or heteroscedasticity may occur in different ways. The variance of the residuals may increase or decrease over time or the variance may be a function of the magnitude of the series. Though, there are several statistical tests available for checking variance changes in residuals, in this study, only the visual inspection of the plot of the residuals is used for checking heteroscedasticity.

2.3.3 Normality tests

A wide range of statistical tests are available for checking whether or not the residuals follow normal distribution. The normality tests, which are used in this study are described below.

Normal probability paper plot

Probability paper plots refer to graph paper for plotting observed data from a chosen distribution. In our case, the chosen distribution is the normal distribution. The scale of the graph paper should be such that a linear relationship is produced for the plotted data. The objective is to write the cumulative distribution fucntion (cdf) in linear form. The lack of linearity is used as a basis for determining whether the sample data could have come from the particular distribution. In order to plot the sample data against a particular pdf, we need to assign a probability associated with each data point. We need to find the empirical cdf, i.e., rank probability as

$$P_i = \frac{i}{n+1} \tag{2.11}$$

where i is the rank and n is the total number of data points (in our case, residuals). We then equate the empirical distribution with the chosen cdf (in our case, it is normal cdf).

$$P_i = \Phi\left(\frac{x_i - \mu}{\sigma}\right) \tag{2.12}$$

where Φ represents the standard normal distribution, x_i's are the data points, μ and σ are the mean and standard deviation of the data points respectively. Here data points are nothing but the residuals. After re-arranging this expression, we will get

$$x_i = \sigma\Phi^{-1}(P_i) + \mu \tag{2.13}$$

This is an equation of straight line with slope σ and intercept μ. The normal probability paper is therefore constructed by plotting the data x_i against the corresponding percentile value from the standard normal distribution $[\Phi^{-1}(P_i)]$. A straight line can be fitted to the data in normal probability paper using linear regression analysis. The coefficient of determination (R^2) gives information about the goodness of fit of the model. The higher R^2 value indicates better fit than the lower value.

Anderson-Darling test

The AndersonDarling test [2] assesses whether a sample comes from a specified distribution. It makes use of the fact that, when given a hypothesized underlying distribution and assuming the data does arise from this distribution, the cumulative distribution function (cdf) of the data can be assumed to follow a Uniform distribution. The data can be then tested for uniformity with a distance test. The formula for the test statistic A to assess if data (note that the data must be put in order) comes from a cdf is

$$A^2 = -n - S \tag{2.14}$$

where n is the total number of data points and the expression for S can be given as

$$S = \sum_{i=1}^{n} \frac{2i-1}{n}[\ln(F(y_i)) + \ln(1 - F(y_{n+1-i}))] \tag{2.15}$$

where n is the total number of data points, i is the rank of the data, $F(\cdot)$ is the normal cdf (for our case) and y_i's are the data points. The test statistic can then be compared against the critical values of the theoretical distribution.

Kolmogorov-Smirnov test

In statistics, the Kolmogorov-Smirnov test (K-S test or KS test) [2] is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution. The Kolmogorov-Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution. The null distribution of this statistic is calculated under the null hypothesis that the sample is drawn from the reference distribution. The empirical distribution function F_n for n iid observations X_i is defined as

$$F_n(X) = \frac{1}{n}\sum_{i=1}^{n} I_{[-\infty,x]}(X_i) \tag{2.16}$$

where $I_{[-\infty,x]}$ is the indicator function, equal to 1 if $X_i \leq x$ and equal to zero otherwise. The K-S statistic for a given cumulative distribution function $F(x)$ is

$$D_n = \sup_x |F_n(x) - F(x)| \tag{2.17}$$

where D_n is the supremum of the set of distances. In practice, the statistic requires a relatively large number of data points to properly reject the null hypothesis.

1. $E_t[z_{t-j}] = z_{t-j}\ , \quad j = 0, 1, 2, \cdots$
2. $E_t[z_{t+j}] = \hat{z}_t(j)\ , \quad j = 1, 2, 3, \cdots$
3. $E_t[a_{t-j}] = a_{t-j}\ , \quad j = 0, 1, 2, \cdots$
4. $E_t[a_{t+j}] = 0\ , \quad j = 1, 2, 3, \cdots$

A complete description about the MMSE forecasting procedure can be found in

Hipel and McLeod (1994). If, the data used for forecasting is a Box-Cox transformed data, then we have to perform an inverse Box-Cox transformation of the MMSE forecasts.

Chapter 3

Exploratory Data Analysis

The exploratory data analysis helps us to find out what kind of models will be appropriate to model the process. The time series data in this study. We find that ARCH and GARCH models are appropriate for this kind of data. All the necessary steps of exploratory data analysis are provided in the subsequent sections.

3.1 Analysis of the Exchange rate USD/BRL data

STEP 1: Seasonality, stationarity and constant variance check

Fig. 3.1 is the time series plot of US dollar and Brazillian Real daily exchange rate from February,2000 to August,2017. It can be seen that Exchange rate from BRL to USD was increasing from the year 2000 to end of 2002. Then the the rate decreased till 2012 , with ups and downs along the way. This shows that economic policies of Brazil where in good stance during this time and were developing positively. After that, the exchange rate plungers high again. We can see the ups and downs over large range, pointing to volatility. Perhaps may be non stationary with mean level increasing slightly. The nonstationarity of the raw data is also confirmed by the fact that the sample ACF, plotted in Fig. 3.2,

Table 3.1: Stationarity tests on the Exchange rate USD/BRL

Test	p-value	h-value	null hypothesis	Stationary/Nonstationary?
ADF	0.7467	1	reject	stationary
PP	0.8739	1	reject	stationary
KPSS	0.01	0	failed to reject	stationary

Table 3.2: Normality tests on the Data

Test	p-value	h-value	null hypothesis	Normal/Nonnormal?
AD	very low	1	rejected	Not Normal
KS	very low	0	rejected	Not Normal

which is almost constant for every lag. Table 3.1 shows the results of the stationarity tests. All the tests confirm that the given data is stationary. The variance appears to be constant over time. Differencing will take care of the stochastic trend, while the AR and MA parameters can describe the remaining dependence among the observations.

STEP 2: Normality check

From the graph of the exchange rate data, it is not clear whether a data transformation is required or not. When, it is possible to visually confirm that the variance of the data is not changing over time, we do not know whether the data follows normal distribution or not. To check the normality of the data, we have to perform some statistical tests. The normal probability paper plot (Normal PPP) (Fig.3.3) is showing a not so good fit with the data. The normality tests (Table 3.2) also confirmed that the data does not follow normal distribution.

STEP 3: Differencing (test for Heteroskedacity

To remove any trend differencing at lag 1 is conducted. The differenced data is plotted in Fig. 3.4, which might have removed any trend in the data. The differenced time series data reflect better, that volatility present in the data. The ACF and PACF plot for the

Figure 3.1: Daily exchange rate of USD/BRL

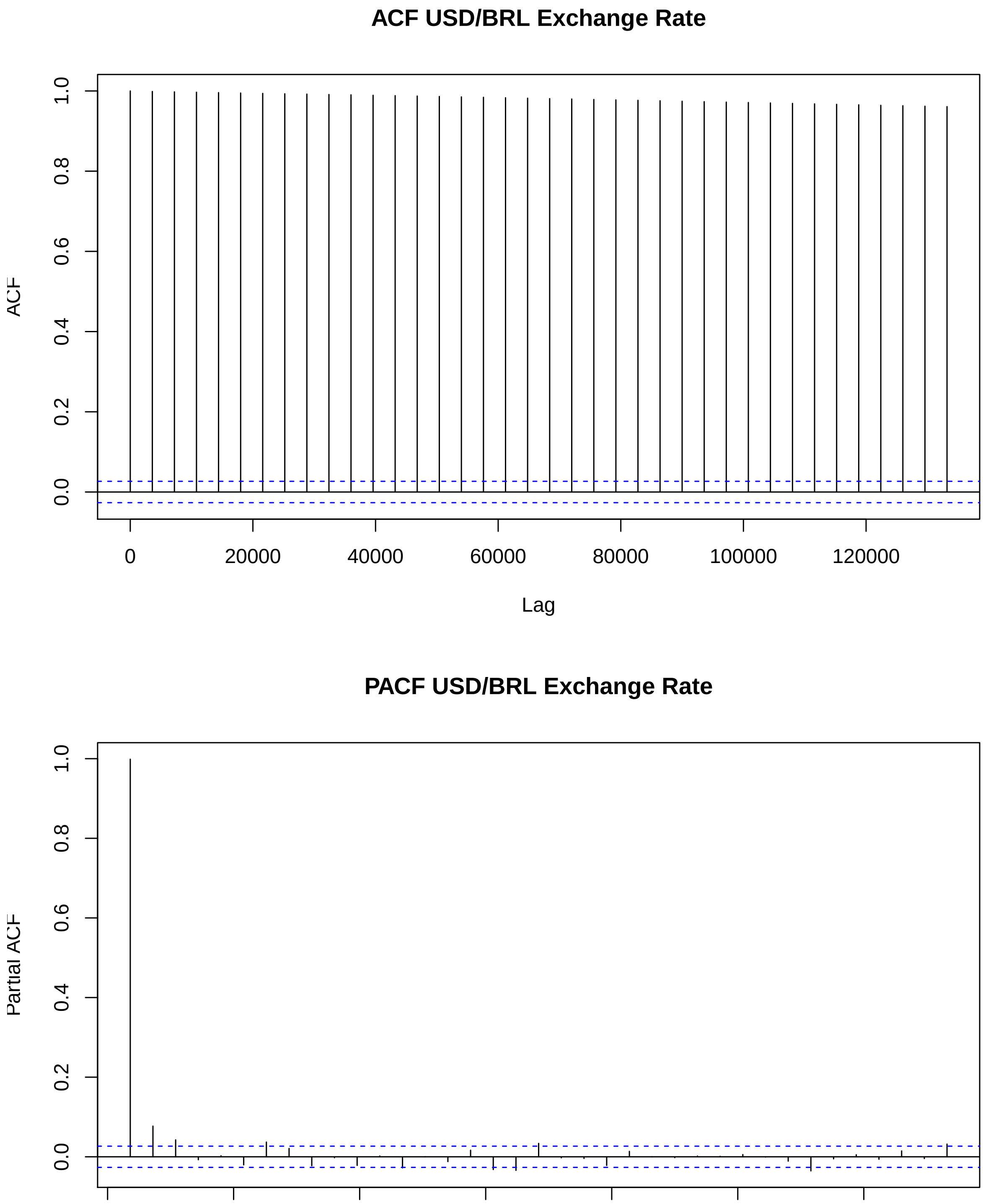


Figure 3.2: Sample ACF and PACF with 95% confidence limits for the USD/BRL raw data

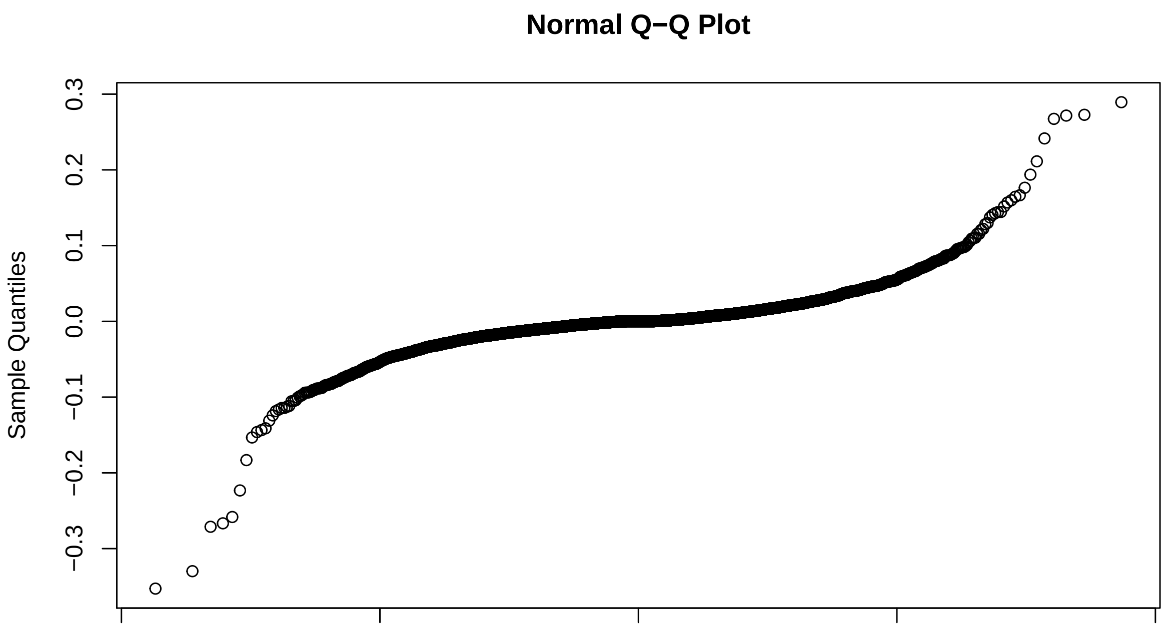


Figure 3.3: Normal probability plot for the daily estimate of USD/BRL exchange rate

Table 3.3: Correlation test for residuals

Test for	p-value	h-value	null hypothesis	Correlated or uncorrelated
Residual	0.0001466	1	rejected	Correlated
Squared Residual	2.2e-16	1	rejected	Correlated

differenced time series data is shown in Fig. 3.5. It can be seen that AR order of 9 and MA order of 8 will fit the data most appropriately. These orders are quite large, indicating that this differenced time series are long memory processes.

STEP 4:Squared residual analysis

The next step is to plot squared residuals. Fig. 3.6 shows the residual and squared residuals of the time series data. Large variability clusters over a long period of time can be witnessed in the above plots. The ACF plots of the residuals resemble the ACF of white noise, while the ACF plots of the square residuals do not as can be seen from Fig. 3.7, indicating squared residuals are correlated. Testing for uncorrelation is also confirmed using the Box Ljung test 3.3

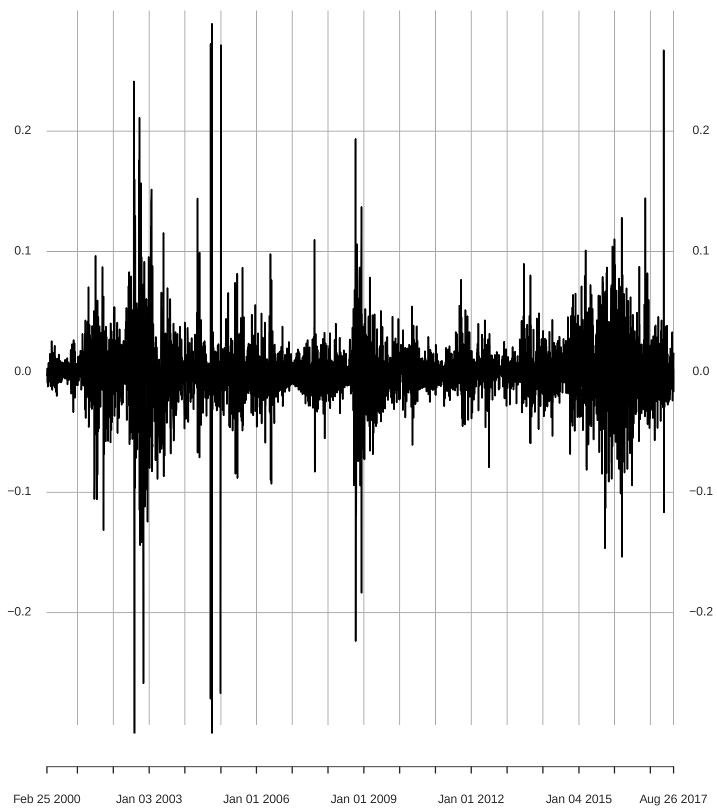
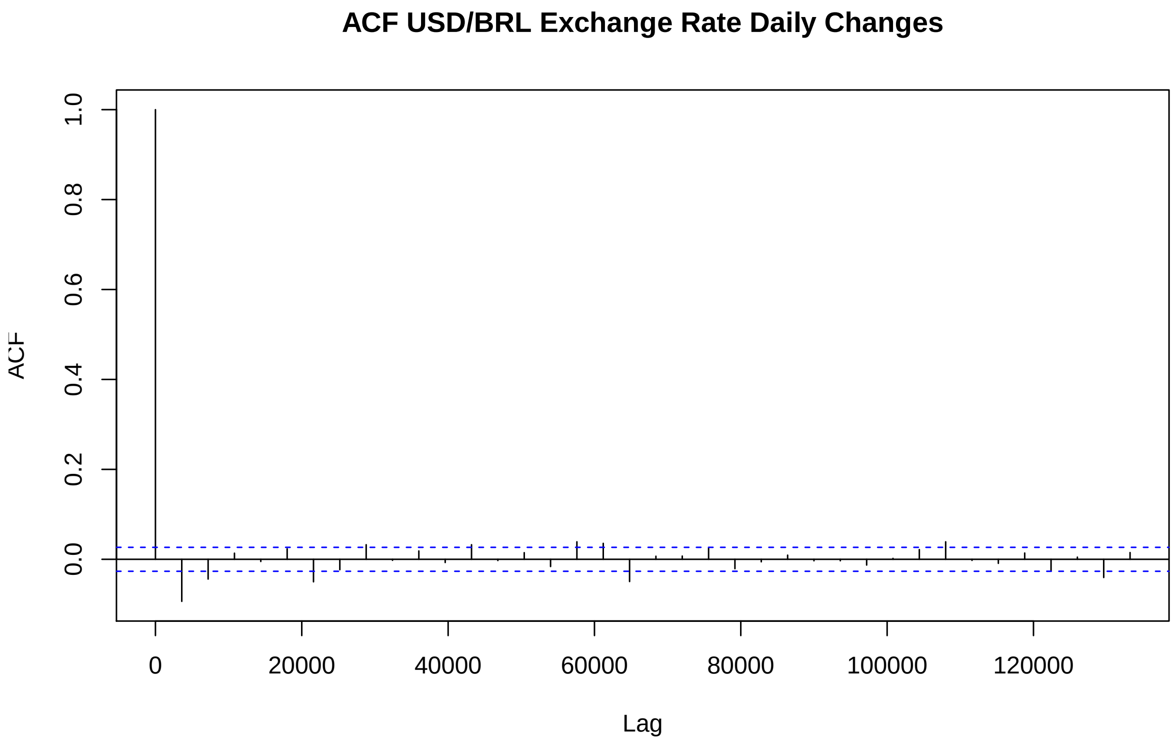


Figure 3.4: Differenced exchange rate of USD/BRL



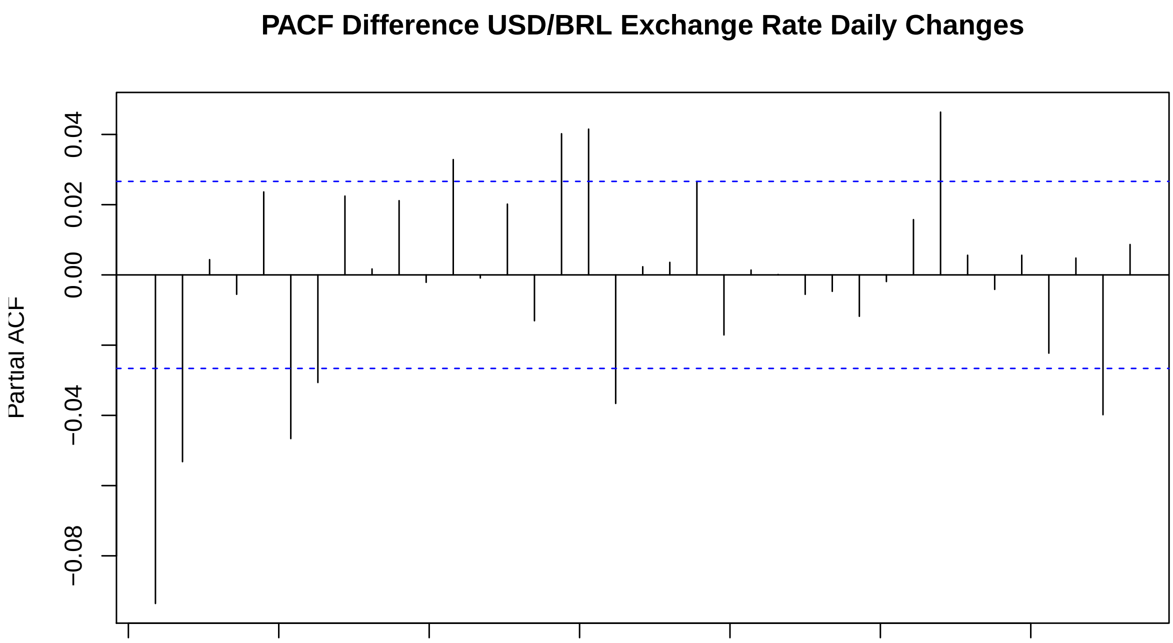


Figure 3.5: Sample ACF and PACF with 95% confidence limits for the differenced USD/BRL exchange rate

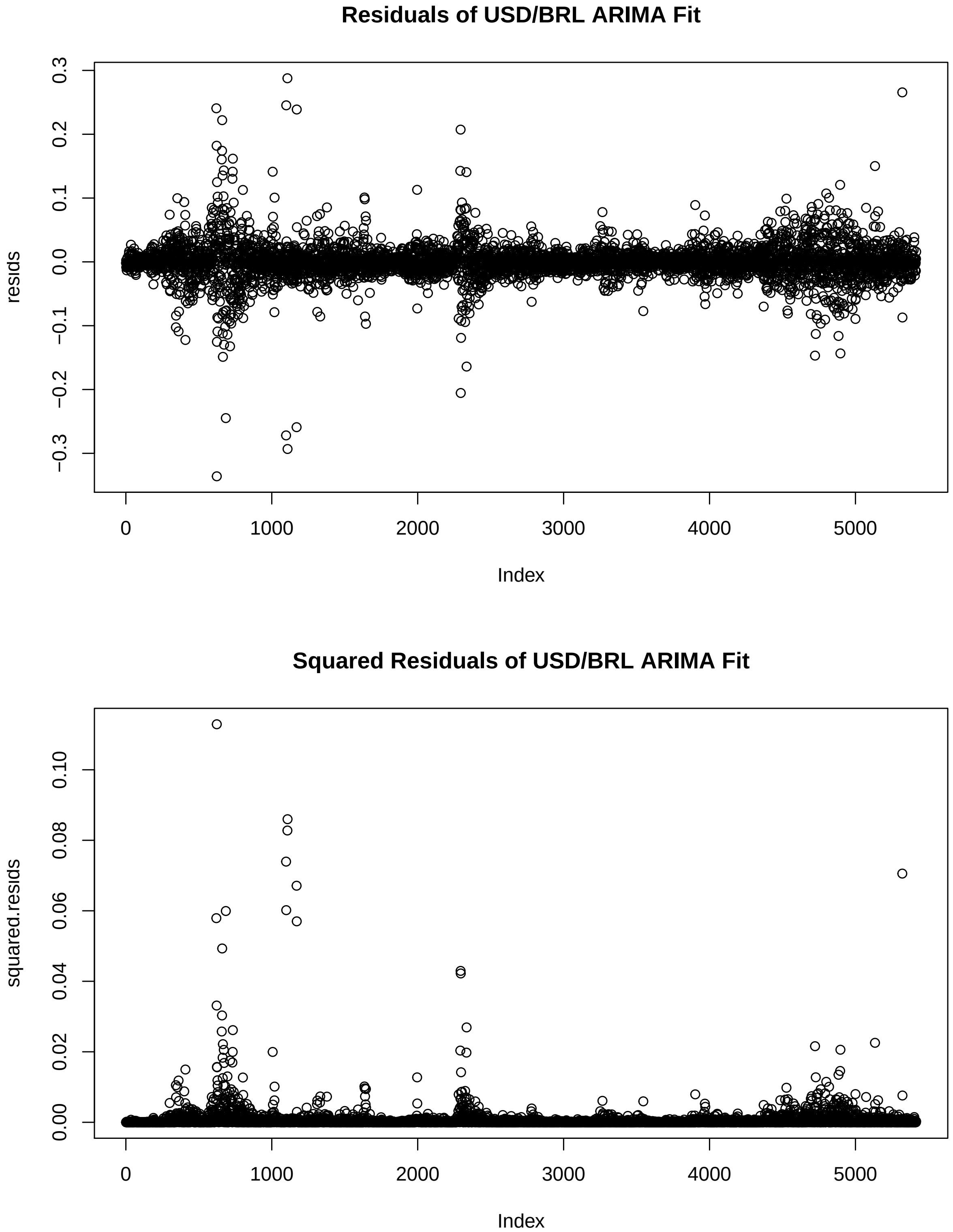
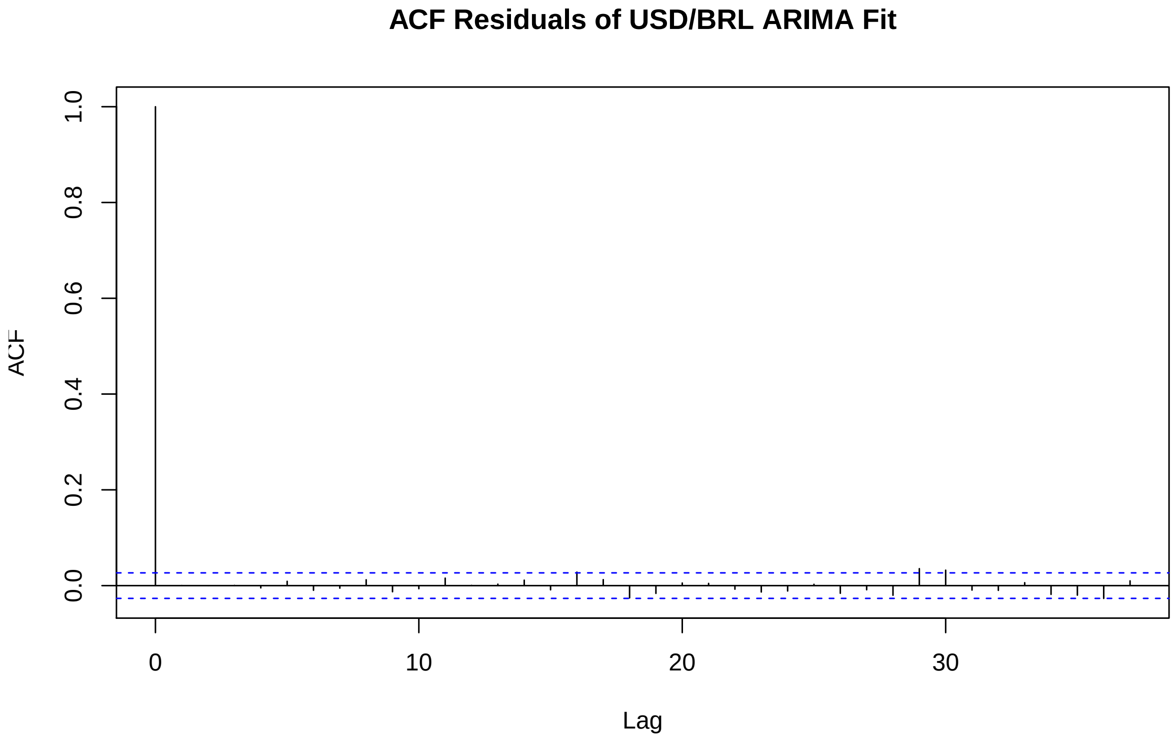


Figure 3.6: residual and squared residual plots



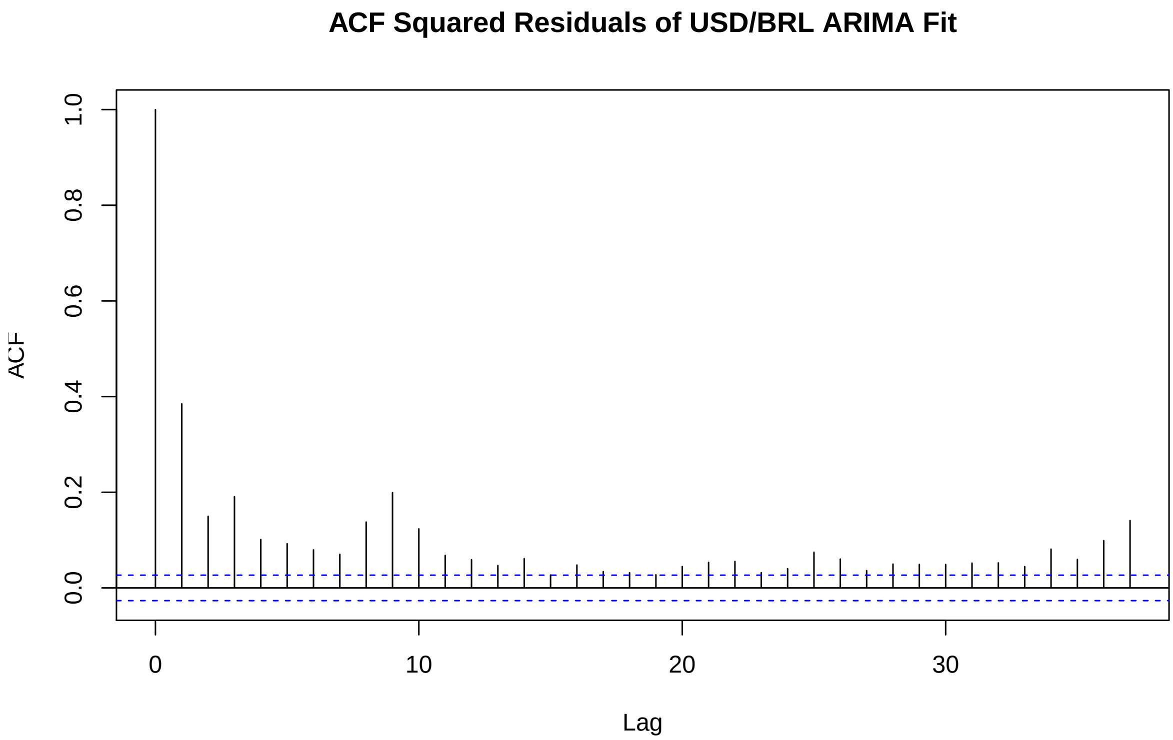


Figure 3.7: ACF and PACF of squared residuals

Chapter 4

Confirmatory Data Analysis

Confirmatory data analysis consists of fitting a range of models to the data sets and constructing the models using the three stages of model construction techniques. Modelling of exchange rate data is carried out and presented in the subsequent sections.

4.1 Analysis of the Model : ARCH and GARCH

In the previous chapter, it became clear that we need to use some models, which models the volatility present in the data. Hence we try to fit ARCH and GARCH model.

Identification of the order

ARCH model

In order to identify the number of AR and MA terms required in the model of the differenced data, sample ACF and PACF graphs are displayed in Fig. 4.1. We call the Arch order of 6, by doing the same residual analysis as mentioned before. The results of which are shown in table 4.1. The table contains 7 estimates including 0 which is the

Table 4.1: Candidate models, estimated parameter values and standard error of the data

Candidate model's order	standard error	t value	p value
O	2.776e-06	31.60	$< 2e - 16$
1	8.526e-03	38.25	$< 2e - 16$
2	7.764e-03	28.86	$< 2e - 16$
3	1.322e-02	12.85	$< 2e - 16$
4	1.086e-02	10.33	$< 2e - 16$
5	9.551e-03	8.43	$< 2e - 16$
6	6.295e-03	44.05	$< 2e - 16$

estimate of the mean. All coefficients are statistically significant, indicating the order 6 ARCH model is appropriate. Larger order may be explored.

GARCH model

For fitting GARCH model, we use a subset of the data, which is the whole data, except the last two months of the time series. Which is used for testing the data. To select the GARCH order, we loop through all combinations of m and n orders, taking values between values between 0 and 3 to fit the ARMA with orders 9 and 8 (as obtained in previous section) and GARCH model with m and n, for which we obtain the smallest BIC. We are using Bayesian information criterion to select the parsimonious GARCH model. The selected orders for (m,n) are (1,1).

Diagnostic checks

The table 4.2 shows the check for correlation in the residuals and the squared residuals. According to the test, when you check the hypothesis of uncorrelated residuals, but we did not reject the new hypothesis of uncorrelated squared residuals. This is the output of the summary fit for the ARCH model applied to the ARMA residuals for the Brazilian real. As the p value was high only for the residual squared term and hence we reject the null hypothesis and say that the residual square data is correlated. All coeeficients are

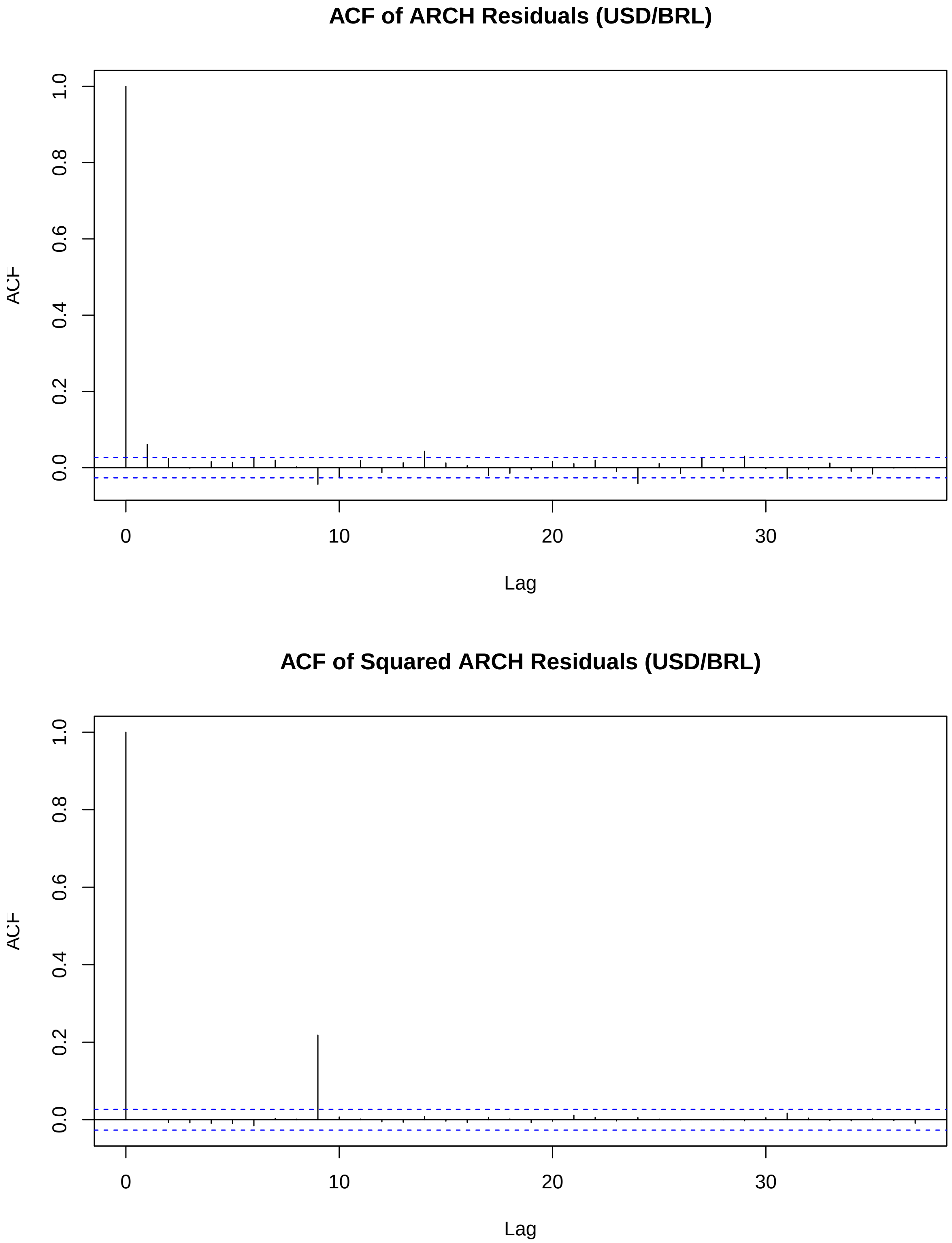


Figure 4.1: ACF and PACF of ARCH model

Table 4.2: Diagnostic check for ARCH model using Jarque Bere test

	X-squared	p value
Residual	125760	$< 2e - 16$
Residual squared	0.00015328	0.9901

Table 4.3: Accuracy measure of forecast data

Error type	value
Mean Squared Prediction Error	0.0001651447
Mean Absolute Prediction Error	0.009399743
Mean Absolute Percentage Error	26.73593
Precision Measure	1.039619

statistically significant as p value in table 4.1 is low for all the six values.

We continue to refine the ARMA-GARCH model by fine tuning the model parameters separately. As simultaneous estimation would be difficult.

4.1.1 Forecasting

The fitted models can be used for forecasting the exchange rate or even to perform Transfer Function Noise model analysis, as exchange rate is one of the key indicator of major policy changes in the country. Here we show the prediction of the currency in the last two months of the time period. The predictions are one lag ahead, by setting the current predicted value as the "observed data" for the next prediction. Prediction for mean and variance is obtained once the model is fit. Predictions are obtained at each time point within the time period of two months. the prediction accuracy measures are shown on the table 4.3 the smaller the precision error, more accurate is the prediction. From the plots 4.2 and 4.3 The variability prediction is in brown and the original data is in blue.

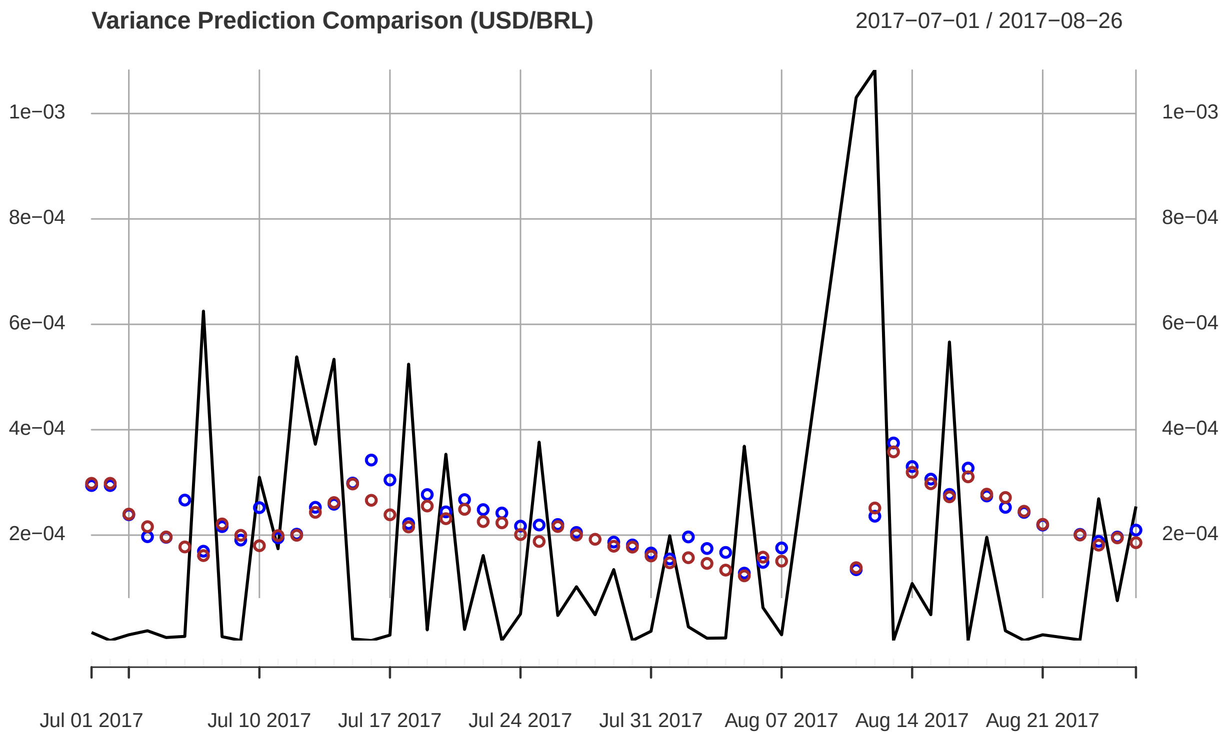


Figure 4.2: Variance prediction of ARMA-GARCH model

Jul 01 2017 Jul 10 2017 Jul 17 2017 Jul 24 2017 Jul 31 2017 Aug 07 2017 Aug 14 2017 Aug 21 2017

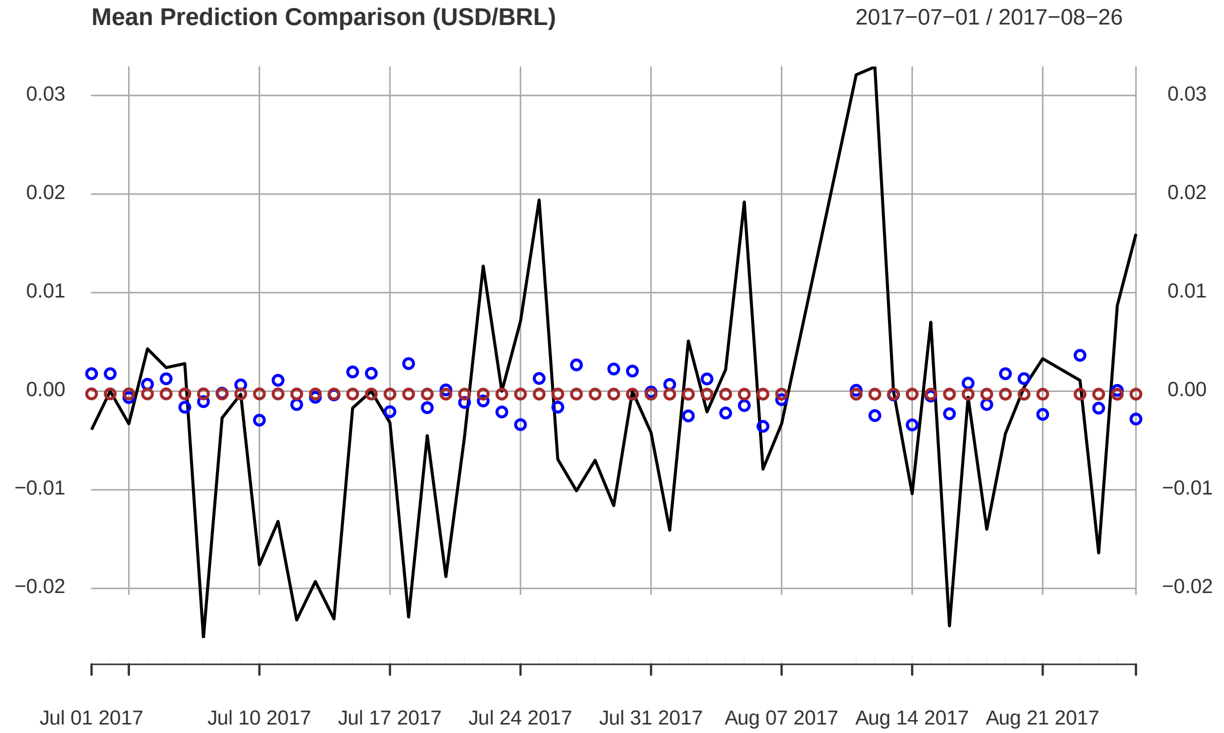


Figure 4.3: Mean prediction of ARMA-GARCH model

Chapter 5

Conclusions

It is observed that the ARMA-GARCH model prediction performs better when the conditional mean does not depend on the time. From the forecast plot, the predicted means are close to zero with some variations in the predictions performed with more complex ARMA-GARCH for Brazilian Real currency. The prediction of conditional variance or volatility is some what captured as seen in the graph. We have seen that ARMA-GARCH model, performs differently across different time series. This GARCH model provides basis for the development of many other applications in the series of Heteroskedasticity models, which are widely in econometric's.

Appendices

Appendix A

R script

The R code, used in this project, is presented below. The R script is well documented and hope it does not need any further explanation.

A.1 MainCode

```
library(data.table)
library(xts)
library(tseries)
library(lmtest)
library(nortest)


#Load data
    #USD/BRL data
        data=read.csv("exchange_rate_USD_BRL_daily.csv",header=TRUE)
        data$Date=as.POSIXct(data$Date,format='%m/%d/%Y')
#POSIXct -- calender dates and times
        data=xts(data[,2],data[,1]) #xts converts raw data to time series
        colnames(data)="rate"
```

```
##############################################################################################

#Exploratory analysis

    #checking for Stationarity
        # Augmented Dickey-Fuller(ADF) test
        adf.test(data)
        #Phillips—Perron Unit Root Test
        pp.test(data)
        #Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test
        kpss.test(data)

    #normality test
        # Anderson Darling (AD) test
        ad.test(data)
        #Kolmogorov-Smirnov Tests
        ks.test(data,"pnorm")

    #Plot original exchange rates
        plot(data$rate,type='l',main='USD/BRL Exchange Rate',ylab="Exchange rate")

    #Differencing the series
        diff.rate=diff(data$rate); diff.rate <- diff.rate[!is.na(diff.rate)]


    #Plot differenced series
        plot(diff.rate,type='l',main='USD/BRL Exchange Rate Daily Changes',ylab="Differe
        plot(diff.rate,type='l',main='USD/BRL Exchange Rate Daily Changes',ylab="Differe
        hist(diff.rate)

    #ACF & PACF plots on original series
        par(mfcol=c(2,1))
        acf(data$rate,main='ACF USD/BRL Exchange Rate')
```

```
            pacf(data$rate,main='PACF USD/BRL Exchange Rate')

        #ACF & PACF plots on differenced series
            par(mfcol=c(2,1))
            acf(diff.rate,main='ACF USD/BRL Exchange Rate Daily Changes')
            pacf(diff.rate,main='PACF Difference USD/BRL Exchange Rate Daily Changes')


#Fit ARIMA on differenced series
        final.aic=Inf
        final.order=c(0,0,0)
        for (p in 1:10) for (d in 0:1) for (q in 1:10)
        {
            current.aic=AIC(arima(diff.rate,order=c(p, d, q)))
            if(current.aic<final.aic)
            {
                final.aic=current.aic
                final.order=c(p,d,q)
                final.arima=arima(diff.rate, order=final.order)
            }
        }

        # What is the selected order?
        final.order
        #9 0 8


#Residual Analysis
        resids = resid(final.arima)[-1]
        squared.resids=resids^2

        par(mfcol=c(2,1))
        plot(resids,main='Residuals of USD/BRL ARIMA Fit')
```

```
    plot(squared.resids,main='Squared Residuals of USD/BRL ARIMA Fit')

    par(mfcol=c(2,1))
    acf(resids,main='ACF Residuals of USD/BRL ARIMA Fit')
    acf(squared.resids,main='ACF Squared Residuals of USD/BRL ARIMA Fit')



    #test for serial correlation
        Box.test(resids,lag=18,type='Ljung',fitdf=17)
    #test for arch effect
        Box.test((resids)^2,lag=18,type='Ljung',fitdf=17)


#ARCH Fit
    library(tseries)
    garch.fit = garch(resids, order = c(0,6),trace=F)
    # What order?
    pacf(resids^2, ,main="Squared Residuals")
    summary(garch.fit)

    #Evaluate goodness of fit
        resids.fgarch = residuals(garch.fit)[-c(1:7)]
        resids.fgarch=resids.fgarch[!is.na(resids.fgarch)]
        par(mfcol=c(2,1))
        acf(resids.fgarch,main="ACF of ARCH Residuals (USD/BRL)")
        acf(resids.fgarch^2,main="ACF of Squared ARCH Residuals (USD/BRL)")
        Box.test(resids.fgarch,lag=10,type='Ljung')
        Box.test(resids.fgarch^2,lag=10,type='Ljung')


#GARCH Model
    #Divide data into training and testing
    #Predict July and August
```

```
        data.test=diff.rate[5365:nrow(diff.rate),]
        data.train=diff.rate[-c(5365:nrow(diff.rate)),]

    #GARCH Order Selection
        library(rugarch)
        #Select model with smallest BIC (if prediction is the objective)
            final.bic = Inf
            final.order = c(0,0)
            for (p in 0:3) for (q in 0:3)
            {
                spec = ugarchspec(variance.model=list(garchOrder=c(p,q)),
                mean.model=list(armaOrder=c(9, 8), include.mean=T),
                distribution.model="std")
                fit = ugarchfit(spec, data.train, solver = 'hybrid')
                current.bic = infocriteria(fit)[2]
                if (current.bic < final.bic)
                {
                    final.bic = current.bic
                    final.order = c(p, q)
                }
            }
        final.order
        #[1] 1 1



    #Refine the GARCH order
        final.bic = Inf
        final.order.garch = c(0,0)
        for (p in 0:3) for (q in 0:3)
        {
            spec = ugarchspec(variance.model=list(garchOrder=c(p,q)),
            mean.model=list(armaOrder=c(final.order.arma[1], final.order.arma[2]),
```

```
          include.mean=T), distribution.model="std")
          fit = ugarchfit(spec, data.train, solver = 'hybrid')
          current.bic = infocriteria(fit)[2]
          if (current.bic < final.bic)
        {
          final.bic = current.bic
          final.order.garch = c(p, q)
          }
        }
  final.order.garch
  #[1] 1 1


#Goodness of Fit
  spec.1 = ugarchspec(variance.model=list(garchOrder=c(1,1)),
      mean.model=list(armaOrder=c(9,8),
      include.mean=T), distribution.model="std")
  final.model.1 = ugarchfit(spec.1, data.train, solver = 'hybrid')

  spec.2 = ugarchspec(variance.model=list(garchOrder=c(1,1)),
      mean.model=list(armaOrder=c(0,0),
      include.mean=T), distribution.model="std")
  final.model.2 = ugarchfit(spec.2, data.train, solver = 'hybrid')

  #Compare Information Criteria
  infocriteria(final.model.1)
  infocriteria(final.model.2)

#Residual Analysis
  resids.final.model = residuals(final.model.2)
  par(mfcol=c(2,1))
  acf(resids.final.model,main="ACF of GARCH Residuals (USD/BRL)")
  acf(resids.final.model^2,main="ACF of Squared GARCH Residuals (USD/BRL)")
```

```
        Box.test(resids.final.model,lag=10,type='Ljung')
        Box.test(resids.final.model^2,lag=10,type='Ljung')
        qqnorm(resids.final.model)

    #Prediction of the return time series and the volatility sigma
        nfore = length(data.test)
        fore.series.1 = NULL
        fore.sigma.1 = NULL
        fore.series.2 = NULL
        fore.sigma.2 = NULL
        for(f in 1: nfore)
        {
            #Fit models
            data = data.train
            if(f>2)
            data = c(data.train,data.test[1:(f-1)])
            final.model.1 = ugarchfit(spec.1, data, solver = 'hybrid')
            final.model.2 = ugarchfit(spec.2, data, solver = 'hybrid')
            #Forecast
            fore = ugarchforecast(final.model.1, n.ahead=1)
            fore.series.1 = c(fore.series.1, fore@forecast$seriesFor)
            fore.sigma.1 = c(fore.sigma.1, fore@forecast$sigmaFor)
            fore = ugarchforecast(final.model.2, n.ahead=1)
            fore.series.2 = c(fore.series.2, fore@forecast$seriesFor)
            fore.sigma.2 = c(fore.sigma.2, fore@forecast$sigmaFor)
        }


    #Compute Accuracy Measures
        #Mean Squared Prediction Error (MSPE)
            mean(na.omit((fore.series.1 - data.test)^2))
            mean(na.omit((fore.series.2 - data.test)^2))
        #Mean Absolute Prediction Error (MAE)
```

```
        mean(na.omit(abs(fore.series.1 - data.test)))
        mean(na.omit(abs(fore.series.2 - data.test)))
    #Mean Absolute Percentage Error (MAPE)
        mean(na.omit(abs(fore.series.1 - data.test)/(data.test+0.000001)))
        mean(na.omit(abs(fore.series.2 - data.test)/(data.test+0.000001)))
    #Precision Measure (PM)
        sum(na.omit((fore.series.1 - data.test)^2)/sum((data.test-mean(data.test))^2
        sum(na.omit((fore.series.2 - data.test)^2)/sum((data.test-mean(data.test))^2



#Mean Prediction Comparison Plot
    n=length(data)
    ymin = min(na.omit(c(as.vector(data.test),fore.series.1,fore.series.2)))
    ymax = max(na.omit(c(as.vector(data.test),fore.series.1,fore.series.2)))
    data.plot = data.test
    names(data.plot)="Fore"
    plot(data.test,type="l", ylim=c(ymin,ymax), xlab=" ",
    ylab="USD/BRL Exchange Rate",main="Mean Prediction Comparison (USD/BRL)")
    data.plot$Fore=fore.series.1
    points(data.plot,lwd= 2, col="blue")
    data.plot$Fore=fore.series.2
    points(data.plot,lwd= 2, col="brown")


#Compare squared observed time series with variance forecasts
    ymin = min(na.omit(c(as.vector(data.test^2),fore.sigma.1^2,fore.sigma.2^2)))
    ymax = max(na.omit(c(as.vector(data.test^2),fore.sigma.1^2,fore.sigma.2^2)))
    plot(data.test^2,type="l", ylim=c(ymin,ymax), xlab=" ", ylab="USD/BRL Exchange
        main="Variance Prediction Comparison (USD/BRL)")
    data.plot$Fore=fore.sigma.1^2
    points(data.plot,lwd= 2, col="blue")
    data.plot$Fore=fore.sigma.2^2
```

```
        points(data.plot,lwd= 2, col="brown")
```

References

[1] Hipel, K. W., & McLeod, A. I. (1994). Time series modelling of water resources and environmental systems (Vol. 45). Elsevier.

[2] https://en.wikipedia.org

[3] https://www.thebalance.com/how-do-exchange-rates-work-3306084

[4] https://www.ofx.com

[5] http://www.stern.nyu.edu/rengle/GARCH101.PDF