Data Science Foundations
In-Course Assessment

TIME SERIES ANALYSIS AND FORECAST

Exchange Rate Forecast With ARIMA

12/01/2021

STUDENT

Mohana Kamanooru

A0223038@tees.ac.uk

MODULE LEADER

Dr. Ala Al Kafri

A.AlKafri@tees.ac.uk

Abstract

Exchange rate analysis and the forecast is widely studied and researched in the current rapidly emerging era. These results facilitate better decision-making in many sectors where money is involved. This thesis follows the complete cycle of analyzing, forecasting, and visualizing the data using the Data Science Process. The exchange rate from GBP to EUR from the historical data available on ofx.com. Data from 2010 to 2020 is analyzed and forecasted from 2020 to 2022. Data prediction used various forecasting methods, both statistical and machine learning methods(Holt-Winters and ARIMA). This study also compares all the mentioned forecasting methods' accuracy and determines the most appropriate method. From the results, this thesis concludes that there cannot be any right or wrong method to forecast data, but it all depends on the type of forecast requirement and type of dataset considered. In this research, the ARIMA forecast method is more appropriate with an accuracy of 96.73%. Accuracy may further be improved with other hybrid models using Artificial Neural Networks etc.

Acknowledgments

Thanks to my Professor Dr. Ala Al Kafri, for the continual guidance and advice throughout the research process. I want to thank my parents Mr. Ramanachari Kamanooru, Mrs. Vijayalakshmi Kamanooru, and my husband, Mr. Santhosh Kanakam, for their continued encouragement and support in my career, and I am very grateful for everything I could learn from all the support received.

Table of Contents

Table of Figures/Tables

Exchange Rate Analysis and Forecast

1. Introduction

The economic growth of the country is a very critical and essential aspect. Economic growth plays a significant role in making decisions towards the financial markets, trade, money flows, interest rates, inflation, and currency exchange rates. Exchange rates facilitate international and intercontinental trading, and this globalization encourages trading giving abundant opportunities to keep interest rates and inflation at low values. Technological growth, human capital, and physical capital goods define the country's economic growth. It is a measure of goods and services provided by the government. Value of goods and services differ for every type/variety location of produce too. Hence the economic growth is not measured in terms of goods or services but by US Dollars. Then and there comes the currency exchange rates into undeniable considerations.

The currency exchange rates have a predominant contribution in international trading markets. A country with a higher and stable economy has higher export rates and cheaper import rates. Thereby it is quite important to find the right gap and time to invest and generate higher profits. Many factors like the country's inflation, interest rates, public debt, and robust economic functioning of the country influence exchange rates. In concise, it is clear of possibilities for an investor from weaker economic country to invest in overseas equities and enhance their returns. Analyzing exchange rates and understanding the trends, seasonality, and patterns in the exchange currency rate will enable financial decision-makers to invest wisely or make a wise decision.

1.1. Overview of Study Problem

According to the specification document's study requirements, an enormous amount of currency exchange data is available from the ofx.com website. However, it is tough to know the trend of the exchange rate from the webpages directly. From this big data, it is hard to predict future exchange rates or to make any conclusion. Hence, the objective of this analysis is to be able to visualize the trend of exchange rates over different periods and to predict the exchange rates for the future too. This research will be helpful to visualize the changes in the exchange rate. This information can be used in the real-time corporate world by various stock-market investors, business investors, and many.

Analyzing and understanding the fluctuations and changes in exchange rates could be a real challenging task, as other important factors affect the rates. Even unpredicted natural disasters, pandemics, political stability, inflation, gross domestic product, interest rates, and monetary policies can directly and indirectly influence. Furthermore, it is not a simple task to determine how long the effects continue to influence. Understanding and forecasting the rates with available tools and techniques in the popular and emerging field of Data Science following every step in its process will surely be worthy of the effort.

1.2.Issues in exchange rates domain

International trading itself is quite complicated and always includes indigestible and unpredictable factors influencing future settlement rates. In such scenarios, investors have a broad scope of making a profit, but only when they could effectively and accurately forecast the future exchange rate.

Investors and firms should always consider the amount of risk involved in deciding the best suitable prediction. Furthermore, be sure that past data is clear, relevant, and understood. As suggested in the business article [1], a combination of technical and fundamental analysis comes in handy in arriving at an informed decision on future exchange rate prediction. Technical analysis involves statistical methods like pattern recognition and mathematical models, whereas fundamental analysis considers the impact of many other political, economic, supply, and demand factors on price trends.

Data scientists and financial firms use concepts of data science and statistical methods, and evolving machine learning techniques, and artificial neural networks [2] to forecast the exchange rates more accurately. Even after having all these models and methods, it is still not easy to conclude one specific method or algorithm works best because of one other crucial factor 'Time'. i.e., the seasonality, trend, and cyclicity factors. It is essential to understand the repetitive, consistent, and random information existing in the data. The period of forecast matters too. Is it a short-term forecast for 6 to 12 months? Is it a long-term forecast of 10 to 20 years? Both scenarios have different weightage for different factors. Therefore, it is undoubtfully a risky task to decide on investments based on forecasted exchange rates.

2. Literature Review

This research is focused on the data sciences processes to analyze the data and forecast future exchange rates. The massive amount of data available on the exchange website is ambiguous. The recent literature reviews and published journals are reliable to analyze and forecast. This research is entirely based on past data to predict the future and does not consider other relative factors and their influence on exchange rate fluctuations. Because the dataset is a univariate time series, traditional analysis is more suitable than functional analysis. As mentioned in the business article [1] this study also focuses on the pattern recognitional, mathematical models, and other analytical theories to forecast the rates.

There are various studies on both statistical methods and machine learning methods. The statistical analysis in [3] Analysis of foreign exchange using descriptive statistics suggests that the future price fluctuations are entirely dependent on past rates and market situations. On the other hand, machine learning methods in [2] Exchange rate forecast using ANN[4] show how to achieve more accurate forecast results using neural networks.

Many available sources also provide information on different forecasting methods on time series[5] analysis like Holt-Winters, ARIMA[6], ARMA, and other hybrid techniques available with recommendations[7]. From the consensus of these articles, the current research is on time series analysis and forecasting methods[8].

3. Technical Implementation

3.1. Data Science Process

Data science provides efficient steps to follow so that the data can be easily analyzed and visualized. The process focuses on five major steps outlined below. This research follows the mentioned steps to prepare the data and identify an appropriate model and visualize the predicted exchange rates.

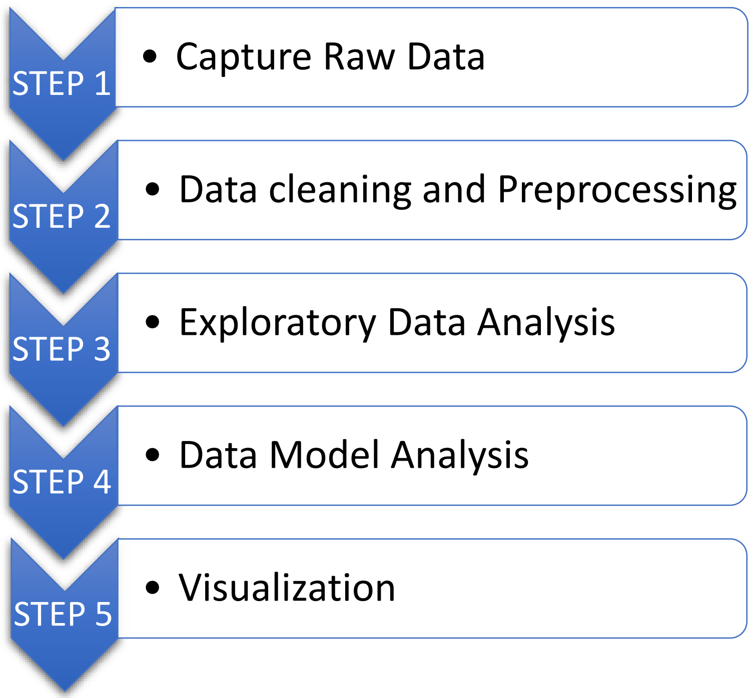


Figure 1 Data Science Process

3.2. Raw Data Capture

Historical exchange rate data between many currencies are available on ofx website. This thesis follows the requirements mentioned in the specification document. Hence, the data is collected from Jan 2010 to Nov 2020 from GBP to EUR using web scraping methods.

The web data consists of exchange rate information from one currency to another currency and different time frequencies. These could be selected using parameters available on the webpage. After providing the input parameters, the data will be as an HTML table on the webpage. The data retrieved consist of two columns date field and the exchange rate field.

The screenshot of the website , with input parameters, is shown below.

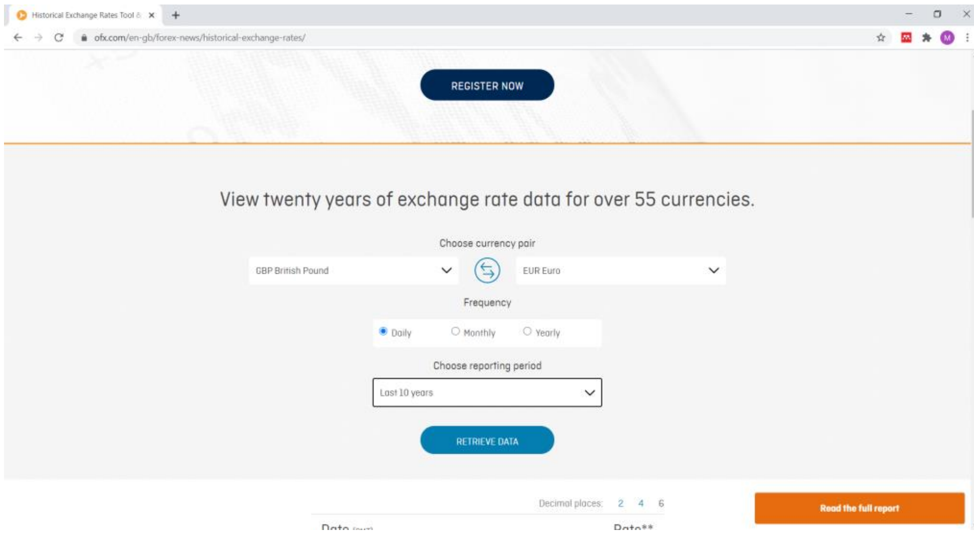


Figure 2 Webpage Parameters

Data	Value	Variable_name
From Currency	GBP Pounds	from_currency_element
To Currency	EUR Euro	to_currency_element
Frequency	Daily	freq_time_id
Period	Last 10 years	time_period

Figure 3 Parameter set values

The data available in this HTML page cannot be saved because it is embedded in the code and not available in the page source. Web scraping and R programming are used to get the data in a CSV file. List of packages and imported libraries are listed in the R markdown file. The captured data has two columns and it doesn't contain any duplicate data or noise or null values. It contains the exchange rates from GBP to EUR from 1990 to 2020.

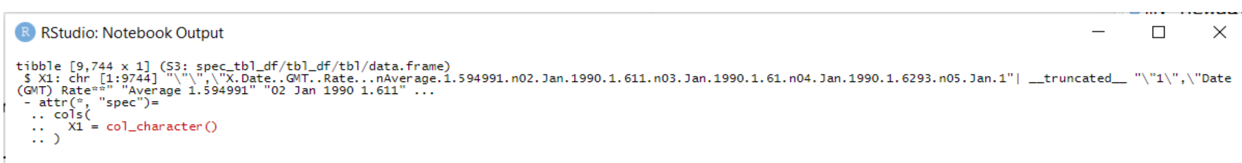


Figure 4 Structure of raw dataset

RStudio: Notebook Output

X1 <chr>
"","X.Date..GMT..Rate...nAverage.1.594991.n02.Jan.1990.1.611.n03.Jan.1990.1.61.n04.Jan.1990.1.6293.n05.Jan.1990.1.637.n08.Jan.1990.1.653.n09.Jan.1990.1.6537.n10.Jan.1990.1.662.n11.Ja...
"1","Date (GMT) Rate**
Average 1.594991
02 Jan 1990 1.611
03 Jan 1990 1.61
04 Jan 1990 1.6293

6 rows

Figure 5 Raw dataset content

RStudio: Notebook Output

```
      X1
Length:9744
Class :character
Mode  :character
```

Figure 6 Summary of raw dataset

3.3. Data Cleaning and Pre-Processing

The raw data contains length column names and data types are incorrect. Firstly, the data from the CSV file is read into a data frame object and during the data cleaning process the columns are renamed, and appropriate data type is assigned. The dataset is also filtered for the data between Jan 2010 and Nov 2020. Since imported data had no blank values, outliers, or noise further processing is not necessary. The dataset is now ready for analysis. The data is shaped as below after pre-processing,

RStudio: Notebook Output

```
'data.frame':	4 obs. of  2 variables:
 $ Date: Date, format: "2020-07-31" "2020-08-31" "2020-09-30" "2020-10-31"
 $ Rate: num  1.27 1.31 1.3 1.3
```

Figure 7 Structure of processed dataset

RStudio: Notebook Output

	Date <date>	Rate <dbl>
1	2020-07-31	1.266913
2	2020-08-31	1.313673
3	2020-09-30	1.295021
4	2020-10-31	1.298405

4 rows

Figure 8 Processed dataset Content

RStudio: Notebook Output

```
      Date                 Rate
 Min.   :2020-07-31   Min.   :1.267
 1st Qu.:2020-08-23   1st Qu.:1.288
 Median :2020-09-15   Median :1.297
 Mean   :2020-09-15   Mean   :1.294
 3rd Qu.:2020-10-07   3rd Qu.:1.302
 Max.   :2020-10-31   Max.   :1.314
```

Figure 9 Summary of processed dataset

3.4. Exploratory Data Analysis

3.4.1. Understanding Data

What is Timeseries Data?

The data collected sequentially with equally spaced time intervals is a time series. If the data is only dependent on time, then it is called a univariate time series dataset.[9]. Time-series is analyzed for forecasting business, understanding past behavior, planning future decisions, and evaluating current accomplishments.

The processed dataset is rectangular. It has “**date**” as one column and exchange “**rate**” in decimal values in another column. The exchange rate is linked only to the date information, and there are no other factors involved in the dataset. Therefore, we can confirm this is a univariate time series dataset.

3.4.2. Analyzing Timeseries Data

The main objective of any time-series data is to predict future values based on historical data. To forecast with higher accuracy, a good understanding of stationarity, seasonality, and auto-correlation of the target variable are essential. To further analyze the exchange rate dataset, create and plot the time series object with the processed exchange rate dataset.

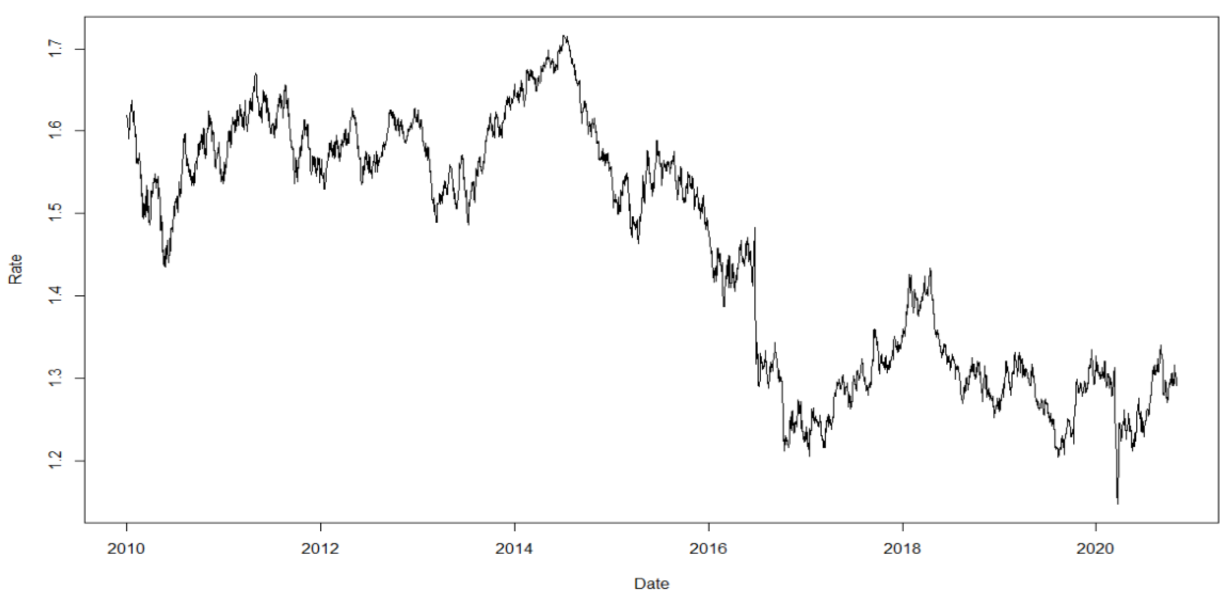


Figure 10 Plot of Processed Dataset

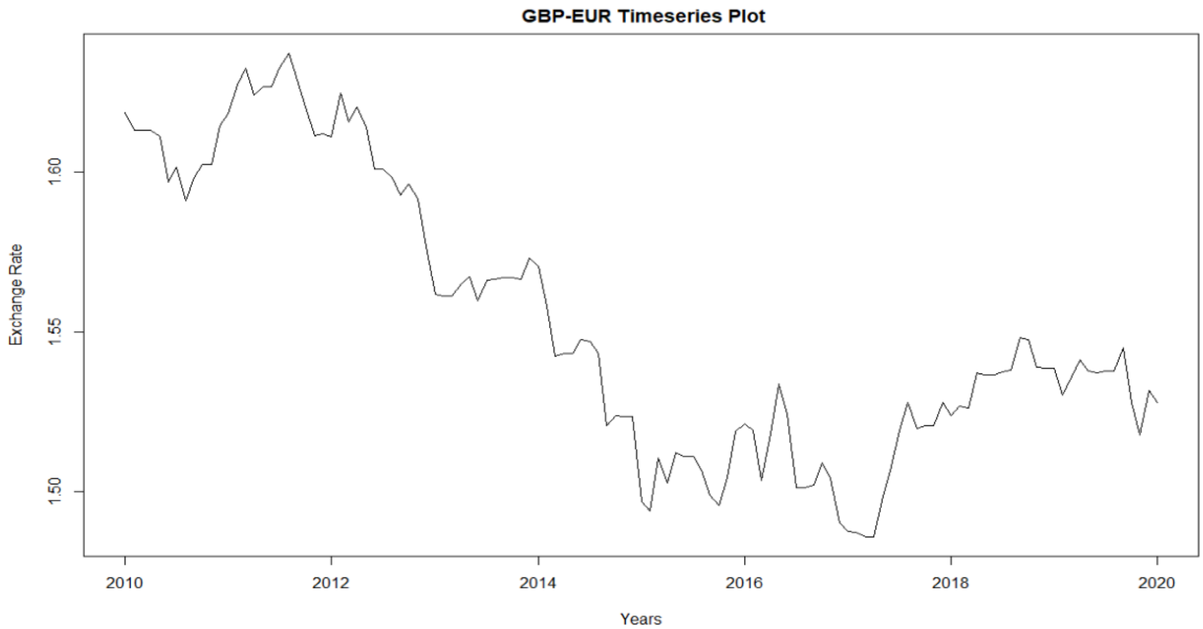


Figure 11 Timeseries Plot

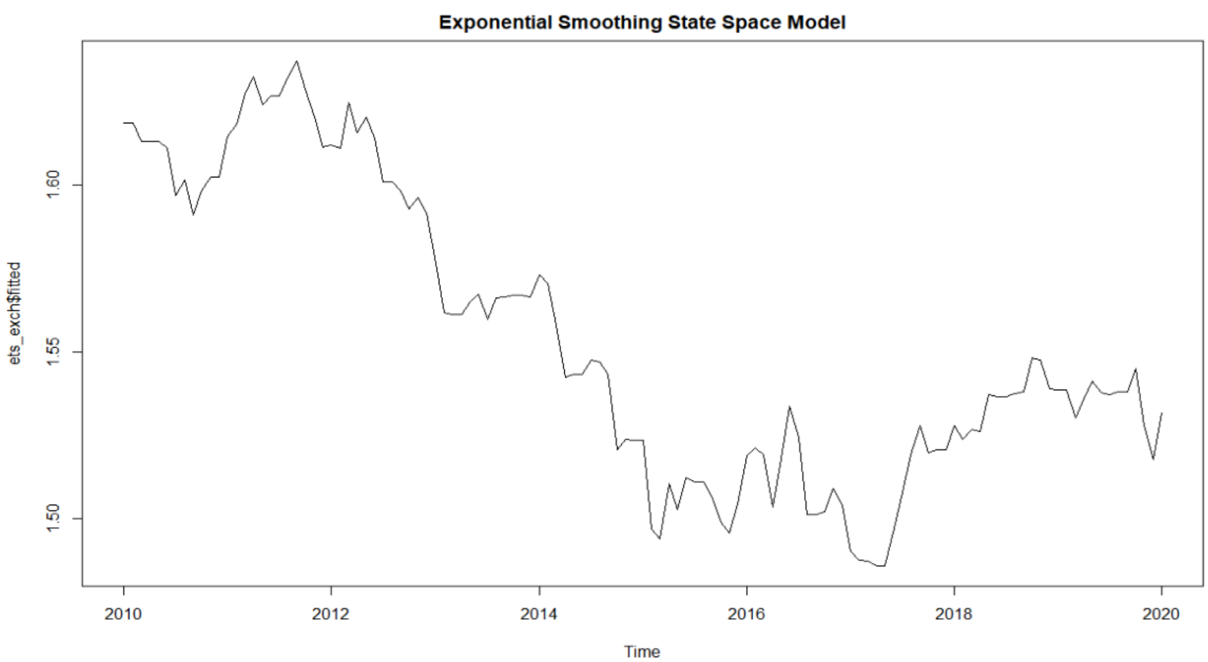


Figure 12 Timeseries ETS Function

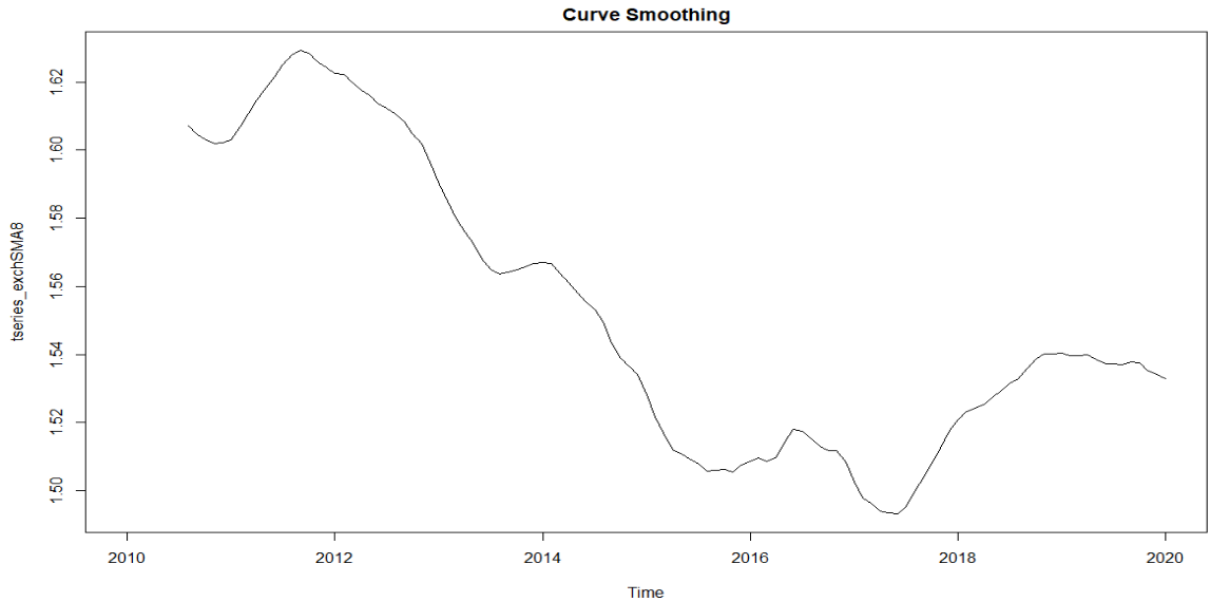


Figure 13 Timeseries curve smoothing

From the figure, the exchange rate is fluctuating and follows the trend but the stationary, seasonality and correlation cannot be determined.

3.4.3. Stationarity, Seasonality, and Trend

Time series decomposition provides a better understanding of the time series components. Decomposing divides the series into four parts - trend, seasonality, irregularity, and cyclicity. The plot below shows these four components in the exchange rate dataset.[9]

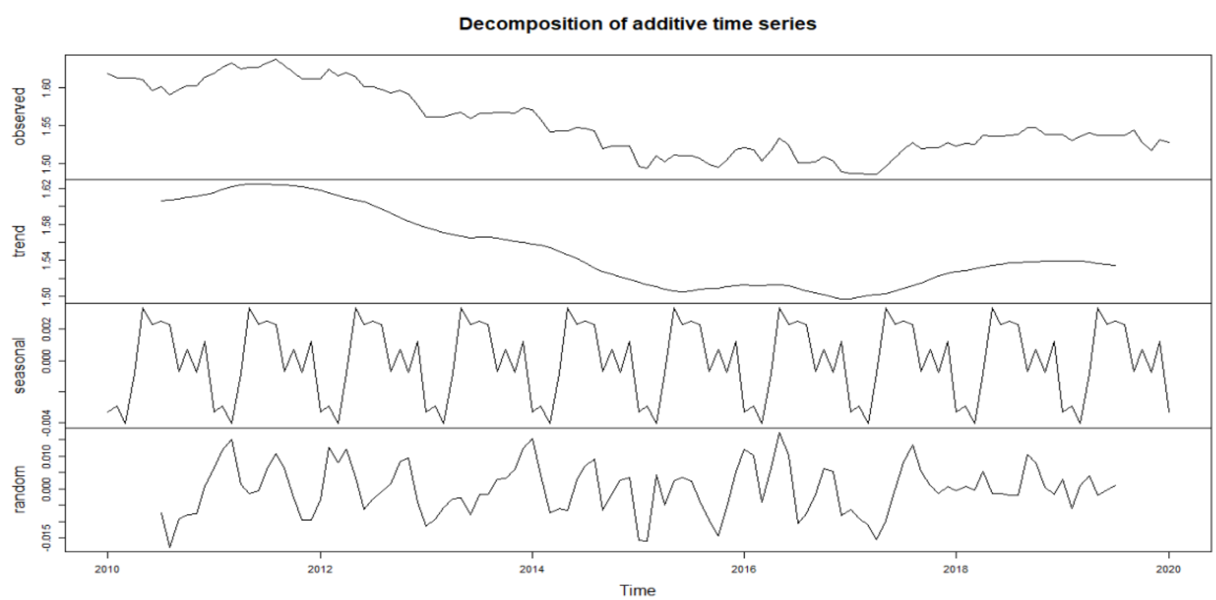


Figure 14 Timeseries Decomposition

Trend Cycle

Figure 14 shows the trend cycle of the time series, not considering the fluctuations and seasonality. Based on the trend, the nature of the time-series is defined, if it is following a downward trend or upward trend, or a horizontal trend. From the blow figure, it can be assumed that the exchange rate might follow a slightly downward trend.

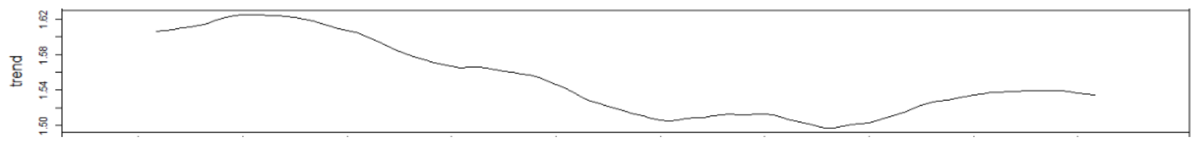


Figure 15 Trend

Irregularity

Irregularity is the random factor of the time-series; the reasons behind the fluctuations are not available in the dataset and not considered while forecasting exchange rate. This irregularity is said to be noise.

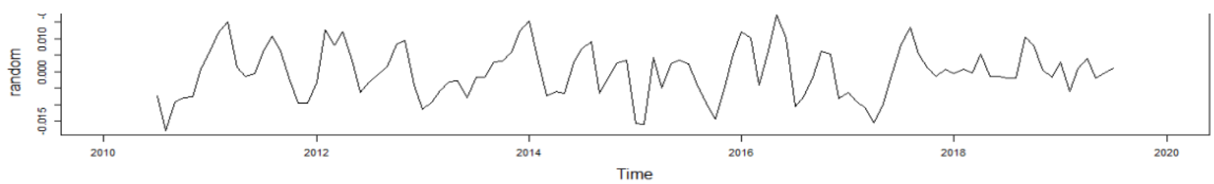


Figure 16 Irregularity

Cyclicity

Cyclicity can happen on infrequent occasions, such as global crises or phenomena that occur at unexpected times and exist for a few years or longer. There is no evidence of recurrence and does not follow any pattern. The exchange rate dataset does not have any cyclicity observed in the plots.

Seasonality

Seasonality describes the fluctuations happening over regular intervals and repetitive, the rate of change between consecutive seasons might be low but high compared between seasons with longer intervals.

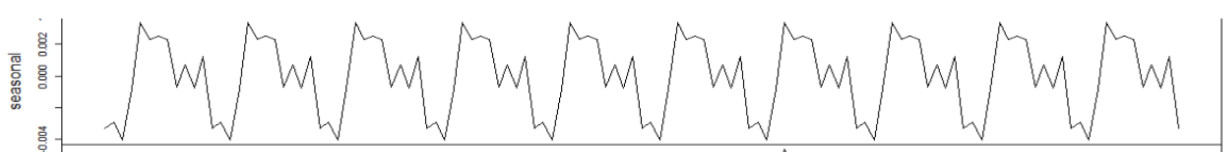


Figure 17 Seasonality

Stationarity

The time series curve is stationary if its mean and variance are constant, and the covariance is not a time function. To determine the exchange rate dataset's stationarity, we perform the Dickey Fuller Test on the series. The value of p determines the stationarity through this statistical test, If p>0 indicates non-stationarity [10]and if p=0, then the series is considered stationary.[11] From the test result p = 0.9, hence the exchange rate data set is non-stationary.

```
        Augmented Dickey-Fuller Test

data:  tseries_exch
Dickey-Fuller = -1.1493, Lag order = 4, p-value = 0.9113
alternative hypothesis: stationary
```

Figure 18 The Dickey-Fuller Test

3.4.4. Data Analysis Conclusions

From this explorative analysis of the data, the time series data of exchange rate is seasonal, follows the trend, has random fluctuations, and from the dickey-fuller test result, it is also evident that the exchange rate data set is non-stationary time series. For a time-series to be forecasted, it should be stationary. Therefore, the forecasting models Holt winters and ARIMA might be appropriate which differentiate the time series and make it stationary to analyze and forecast.

3.5. Data Model Analysis

There is an infinite number of forecasting methods available in the market. Also, there are businesses which provide forecasting service like Currencies Direct Financial Markets [1]. However, in real-time scenarios, there is no perfect forecasting tool that proves its efficiency over other forecasting techniques. Some methods are simple to apply, and others may use high-end technologies with AI, ANN, and additional hybrid models which might become complicated and costly.

3.5.1. Types and process of Forecast

Time-series forecast falls into three categories based on time horizons. Shot-ranged forecast(less than three months, but rarely up to a year) for sales and customer demand, etc., Medium ranged forecast (from 3 months to 3 years), and Long-range forecasts (any range above three years) for making investment decisions.

3.5.2. Forecasting Process

Every forecast follows a seven-step process to be successful. The whole process is about starting the forecast project, designing, and implement it. The steps are outlined below.

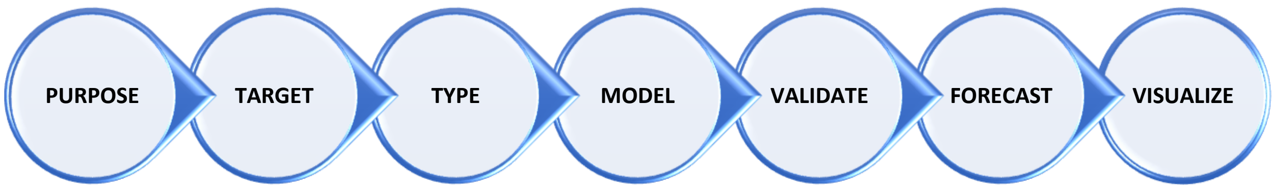


Figure 19 Forecasting Process

1. **Purpose**: Identify the purpose of the forecast
2. **Target**: Determine the target element to be forecasted
3. **Type**: Choose the time horizon = short medium or long-range forecast
4. **Model**: Analyze appropriate forecast model for the chosen data
5. **Validate**: Train and test the model for available data for accuracy
6. **Forecast**: Forecast the target with an accurate model
7. **Visualize**: forecast the result and analyze the results

3.6. Forecasting Models

In this research, statistical methods, and machine learning models (Holt-Winters and ARIMA Model) will be used to forecast the exchange rate. Machine learning techniques are very handy to identify a better suitable model for the dataset. First, split the dataset into a training sample and testing sample in 80:20 ratios. Since the data is gathered from 2010 Jan to Nov 2020, it is sampled as below. The accuracy is measured by MAPE /Mean Absolute Percentage Error, and RMSE/Root Mean Square Error.

Training Data – Date Vs Rate from Jan 2010 to Dec 2018

Testing Data – Date Vs Rate from Jan 2019 to Nov 2020

Accuracy – measured with mean and mean square error.

3.7. Naïve, Mean and Drift Approach

Naïve, mean and drift methods are simple statistical methods that depend only on the past value of the data and do not require any other factors or information to evaluate.

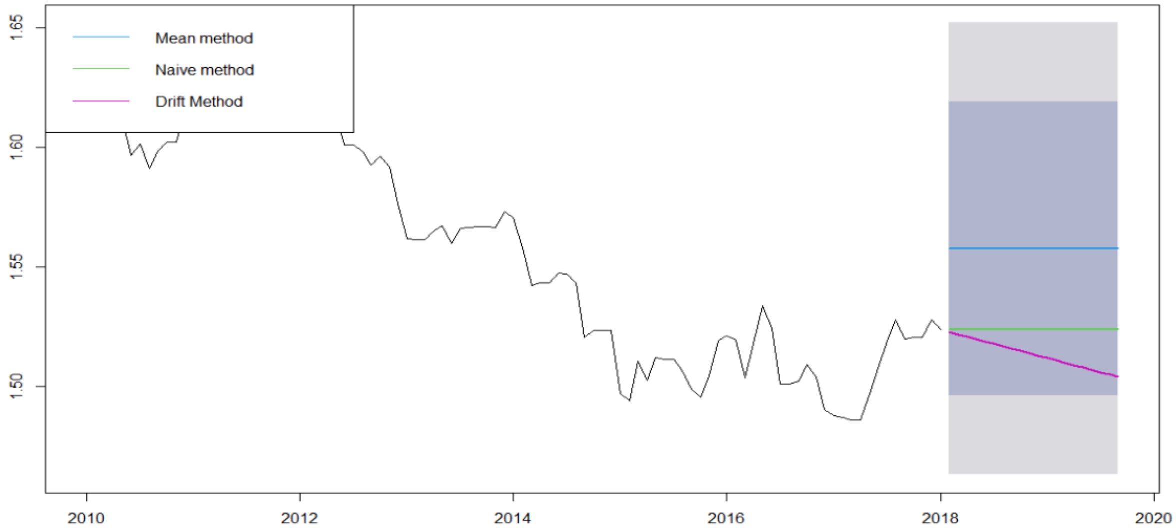


Figure 20 Test Results - Mean, Naive, Drift

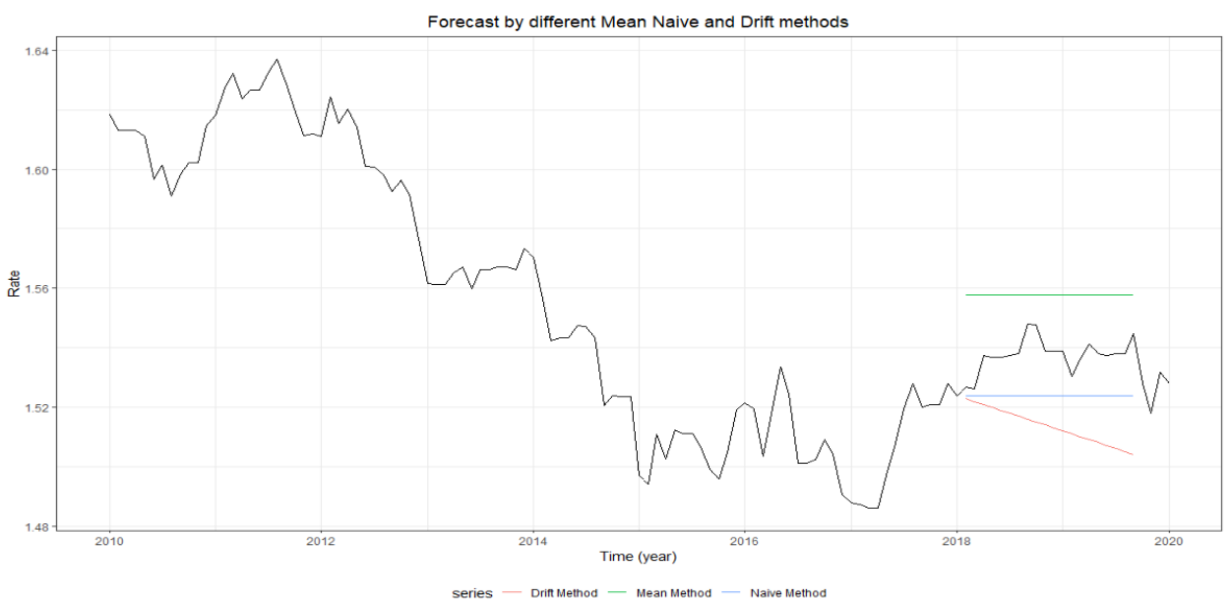


Figure 21 Forecast - Mean, Naive, and Drift

```
accuracy(mean_ex, total_exch)
```

```
                        ME       RMSE        MAE         MPE     MAPE      MASE      ACF1 Theil's U
Training set -6.409926e-17 0.04702457 0.04214583 -0.09095092 2.704796 1.6455403 0.9726000        NA
Test set     -1.988083e-02 0.02060675 0.01988083 -1.29412946 1.294129 0.7762266 0.4318227  3.899103
```

Figure 22 Mean Accuracy

```
accuracy(naive_ex, total_exch)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.0009858333	0.008484374	0.00617800	-0.06428024	0.3986106	0.2412136	0.1021370	NA
Test set	0.0139196500	0.014938117	0.01391965	0.90397507	0.9039751	0.5434783	0.4318227	2.991489

Figure 23 Naive Accuracy

```
accuracy(drift_ex, total_exch)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-9.020562e-17	0.008426906	0.006302281	-0.0009059021	0.4064263	0.2460660	0.1021370	NA
Test set	2.427090e-02	0.025965238	0.024270900	1.5766371211	1.5766371	0.9476322	0.6365786	5.202887

Figure 24 Drift Accuracy

3.8. Holt-winters

Forecasting the univariate time-series using the Holt-Winters object using R programming results as below with 80% training and 20% testing.

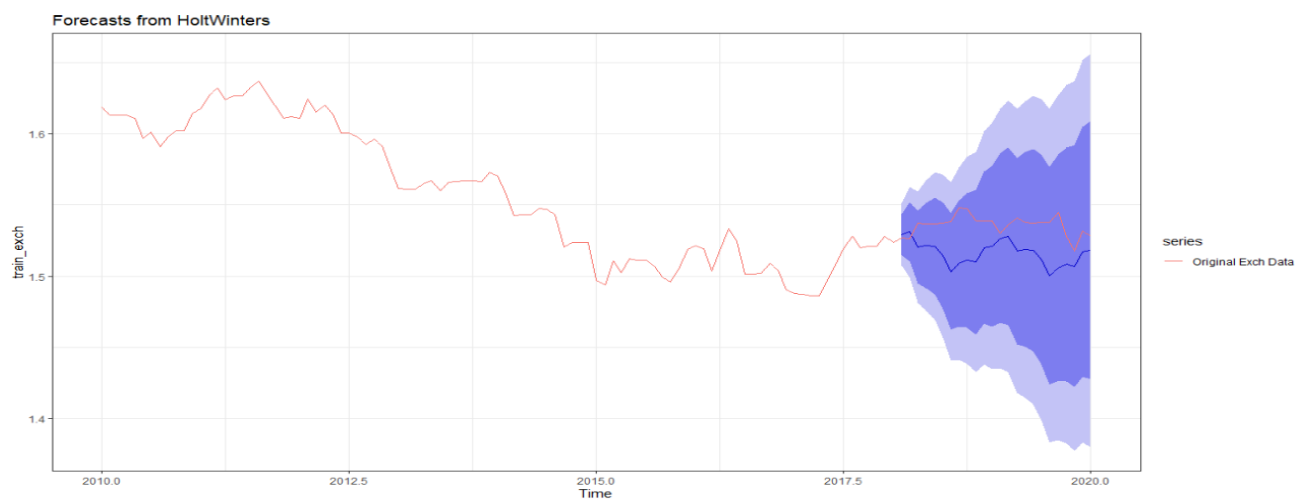


Figure 25 Test Results - Holt-Winters

```
accuracy(forecast(train_holt, h=24), test_exch)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-0.001086076	0.01123869	0.008930055	-0.06947983	0.5778938	0.3486648	0.1468536	NA
Test set	0.098891311	0.09986852	0.098891311	6.11770905	6.1177090	3.8611089	0.8241218	14.94505

Figure 26 Accuracy - Holt Winters

Seasonal Method

The current research uses the additive Holt-winters method to forecast because the additive method is preferred if seasonal variations are nearly constant and multiplicative when seasonal variations are proportional to the changes in series.[12]

Figure 27 Test Results - Holt-Winters (Seasonal)

```
accuracy(forecast(fit1 , h=24), total_exch)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.0006623137	0.01393349	0.01152412	0.04480054	0.7411147	0.4499474	0.7957722	NA
Test set	0.0035689661	0.01309089	0.01100577	0.23003495	0.7170534	0.4297089	0.7687161	1.910831

Figure 28 Accuracy - Holt Winters (Additive)

```
accuracy(forecast(fit2 , h=24), total_exch)
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	0.000254011	0.01356191	0.01086125	0.01264748	0.6983993	0.4240665	0.7478286	NA
Test set	0.033932728	0.03512946	0.03393273	2.20801951	2.2080195	1.3248682	0.5169253	5.181197

Figure 29 Accuracy- Holt-Winters (Multiplicative)

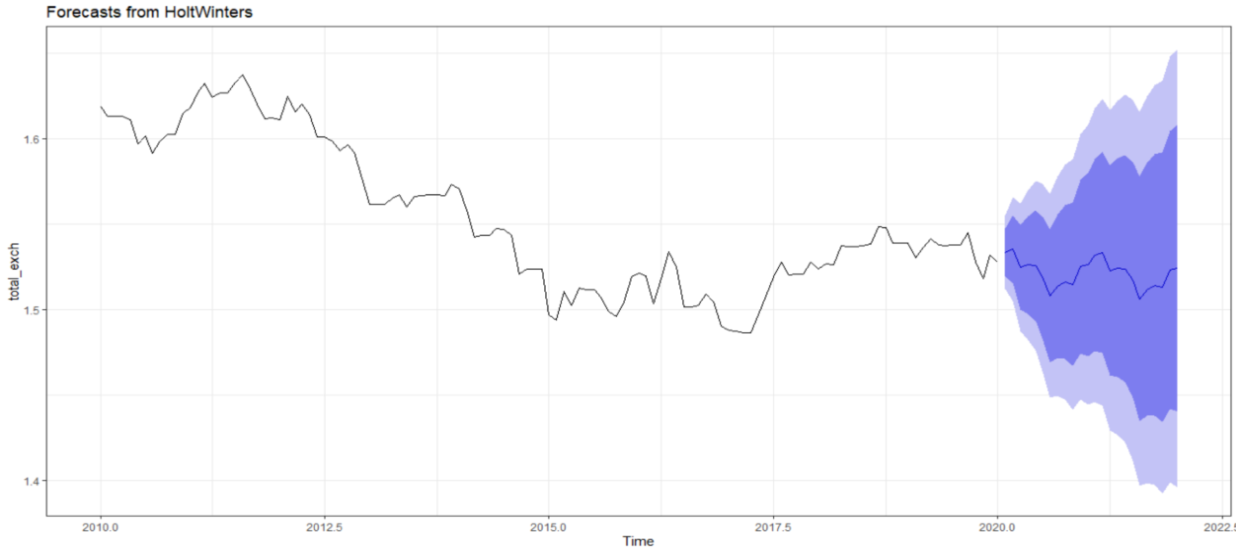


Figure 30 Forecast - Holt -Winters

3.9. ARIMA

3.9.1. Understanding ARIMA

Auto-Regressive Integrated Moving Average (ARIMA) method is widely used for time series forecast analysis. ARIMA focuses on autocorrelation, while other approaches focus on seasonality and the data trend. ARIMA model is applied when the time series is stationary, but from the Augmented Dickey-Fuller test result, the exchange rate dataset is non-stationary. Series can be made stationary by differentiating. To understand this model, better, explicit knowledge on ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) [13] is essential. The forecast process is explained in the next section, and this thesis exactly follows the below process to forecast the exchange rate from 2020 to 2022.

3.9.2. Forecast Process

The flow chart proposed in [14] is shown below and these guidelines are followed in this current thesis to forecast exchange rats from GBP to EUR for 24 months.

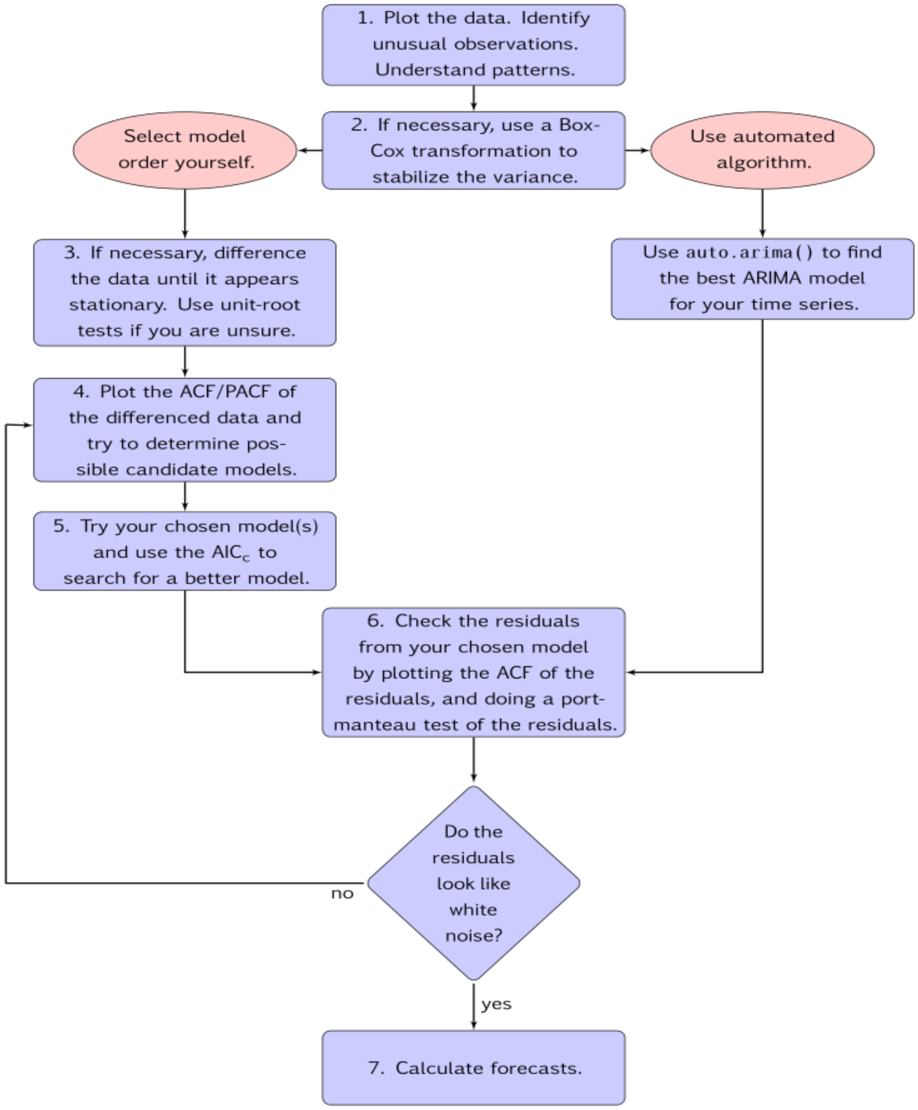


Figure 31 ARIMA Process

(Source: https://otexts.com/fpp2/arima-r.html)

3.9.3. The Dickey-Fuller Test - Stationarity

The exchange rate time series is non-stationary, the series is differentiated, and the stationarity is checked using the Dickey-Fuller Test. For a stationary series, the p-value = 0. Higher-order series is always stationary, but it results in overfitting to the current data and may not yield better forecast values. From the plot, ACF has a lag at 4.

```
##  Augmented Dickey-Fuller Test for zero order
## data:  total_exch
## Dickey-Fuller = -1.1493, Lag order = 4, p-value = 0.9113
## alternative hypothesis: stationary
```

```
##  Augmented Dickey-Fuller Test for first order
## data:  diff(total_exch, differences = 1)
## Dickey-Fuller = -4.428, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
##  Augmented Dickey-Fuller Test
## data:  diff(total_exch, differences = 2)
## Dickey-Fuller = -8.7937, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

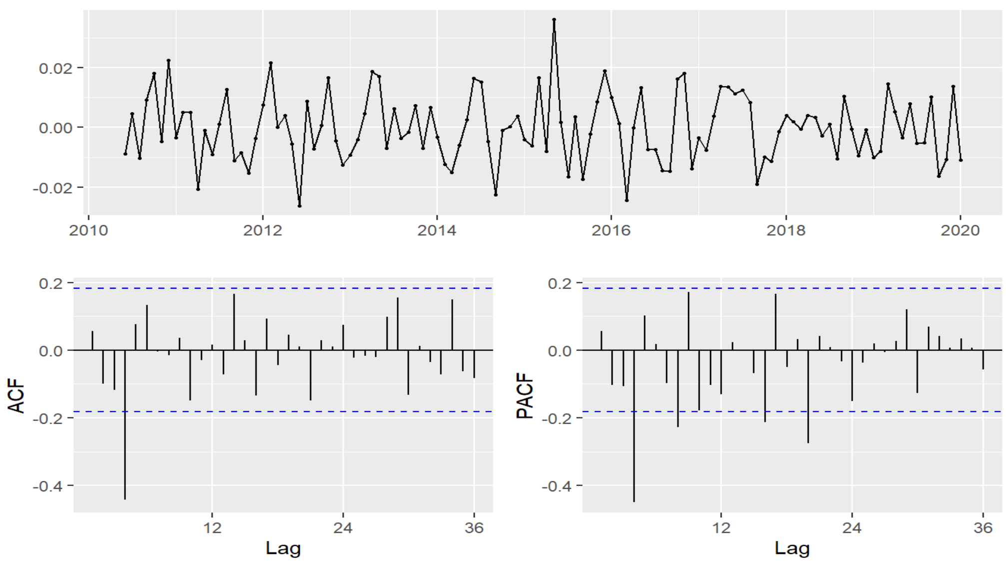


Figure 32 ACF, PACF plot order 2

3.9.4. Seasonality (P, D, Q)

ARIMA calculation requires three parameters p, d, and q. ARIMA *(p, d, q),* p is the autocorrelation order, d order of differentiation, and q is the moving average order. In this thesis, the values of p, d, and q are taken with basic assumptions from ACF, PACF, and L Jung-Box test. As the series also has a seasonal component the values P, D and Q are predicted first based on the AIC value. The lower the AIC value more appropriate the P, D, and Q are.

```
## Series: total_exch
## ARIMA(0,1,1)(0,1,0)[12]
## Coefficients:
##          ma1
##       0.1745
## s.e.  0.1061
##
## sigma^2 estimated as 0.0001483:  log likelihood=323.43
## AIC=-642.85   AICc=-642.74   BIC=-637.49
```

```
## Series: total_exch
## ARIMA(0,1,1)(1,1,0)[12]
##
## Coefficients:
##          ma1     sar1
##       0.1706  -0.5921
## s.e.  0.1160   0.0731
##
## sigma^2 estimated as 9.311e-05:  log likelihood=346.56
## AIC=-687.12   AICc=-686.89   BIC=-679.07
```

```
## Series: total_exch
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##          ma1     sma1
##       0.1591  -0.9997
## s.e.  0.1106   0.1904
##
## sigma^2 estimated as 6.764e-05:  log likelihood=352.67
## AIC=-699.35   AICc=-699.12   BIC=-691.3
```

Table 1 (Seasonal P, D, Q) – Comparing AICc

SEASONAL POINTS P, D, AND Q	AICc
ARIMA(0,1,1)(0, 1, 0)	-642.74
ARIMA(0,1,1)(1, 1, 0)	-686.89
ARIMA(0,1,1)(0, 1, 1)	-699.12

Since AICc of (P, D, Q) at the point (0, 1, 0) is -642.74 and the lowest, It is considered for further forecast and study.

3.9.5. ARIMA (p, d, q)(P, D, Q)

Differentiating the series by order 1 (it is ideal to keep to d = 0 since the series is non-stationary d cannot be 0), and the order of mean is assumed to be 1. After the L Jung-Box test and plots, the value of p is analyzed and further assumptions of p, d, and q are made.

The seasonal points are chosen at (0,1,0) and series is stationary at order 1, as the value of p= 0.008535 (is < 0.5) . This implies that the null hypothesis can be rejected.

```
p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  diff(tseries_exch, differences = 1)
Dickey-Fuller = -4.428, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Figure 33 Dickey Fuller Test order 1

Therefore, ARIMA forecast will be done at different points, with d = 1 and seasonal point (0,1,0) to analyze the AIC value. For all points of the ARIMA forecast, the most appropriate and accurate point can be determined by AIC (Akaike Information Criteria) value. Forecast with the lowest AC value is considered the most accurate model.

```
# Series: total_exch
## ARIMA(0,1,0)(0,1,0)[12]
##
## sigma^2 estimated as 0.0001502:  log likelihood=322.27
## AIC=-642.55   AICc=-642.51   BIC=-639.87
```

```
## Series: total_exch
## ARIMA(1,1,0)(0,1,0)[12]
##
## Coefficients:
##          ar1
##       0.1140
## s.e.  0.0958
##
## sigma^2 estimated as 0.0001496:  log likelihood=322.98
## AIC=-641.96   AICc=-641.84   BIC=-636.59
```

```
## Series: total_exch
## ARIMA(0,1,1)(0,1,0)[12]
##
## Coefficients:
##          ma1
##       0.1745
## s.e.  0.1061
##
## sigma^2 estimated as 0.0001483:  log likelihood=323.43
## AIC=-642.85   AICc=-642.74   BIC=-637.49
```

```
## Series: total_exch
## ARIMA(1,1,1)(0,1,0)[12]
##
## Coefficients:
##           ar1     ma1
##       -0.2132  0.3740
## s.e.   0.2950  0.2693
##
```

```
## sigma^2 estimated as 0.0001489:  log likelihood=323.71
## AIC=-641.42   AICc=-641.19   BIC=-633.38
```

also forecasting the ARIMA point using the auto. arima function with the seasonal parameter,

```
## Series: total_exch
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.4039  -0.9022  -0.3288  0.7997
## s.e.  0.1152   0.0770   0.1684  0.1107
##
## sigma^2 estimated as 6.401e-05:  log likelihood=410.99
## AIC=-811.99   AICc=-811.46   BIC=-798.05
```

Table 2 ARIMA(p,d,q) - Comparing AIC

ARIMA (p , d, q)	AIC
ARIMA(0,1,0)(0, 1, 0)	-642.55
ARIMA(0,1,1)(0, 1, 0)	-642.85
ARIMA(1,1,0)(0, 1, 0)	-641.96
ARIMA(1,1,1)(0, 1, 0)	-641.42
AUTO.ARIMA AT (2,1,2)(0,1,0)	-811.99

AIC is lowest at the ARIMA (1,1,1)(0,1,0)[12] , and using auto ARIMA function, the predicted values of p,d and q are (2, 1, 2) and AIC value is higher -811.99. However automation to identify the points cannot be trusted all the time. Moreover, on occasions, higher AIC values have lower MAE and lower RMSE values. To eliminate this confusion, the Accuracy test is run on both points to check the magnitude of the error.

Accuracy ARIMA (1, 1, 0) (0, 1, 0) [12]

```
accuracy(forecast(fittrain110, h=24), test_exch)
##                        ME       RMSE         MAE        MPE     MAPE      MASE
## Training set 0.0002809638 0.01210965 0.009142173 0.01828603 0.592574 0.3569467
## Test set     0.0511089296 0.05583798 0.051108930 3.16295712 3.162957 1.9954953
##                    ACF1 Theil's U
## Training set 0.02707039        NA
## Test set     0.79649453  8.021124
```

Accuracy ARIMA (1, 1, 1) (0, 1, 0) [12]

```
accuracy(forecast(fittrain111, h=24), test_exch)
##                        ME       RMSE         MAE        MPE      MAPE      MAS
E
## Training set 0.0002708413 0.01200594 0.009099092 0.01766165 0.5897584 0.355264
6
## Test set     0.0527554841 0.05727679 0.052755484 3.26471685 3.2647169 2.059783
3
##                    ACF1 Theil's U
## Training set -0.0269132        NA
## Test set      0.7945842  8.237161
```

Accuracy ARIMA (2 , 1 ,2) (0 ,1 ,0) [12]
auto.arima forecast point

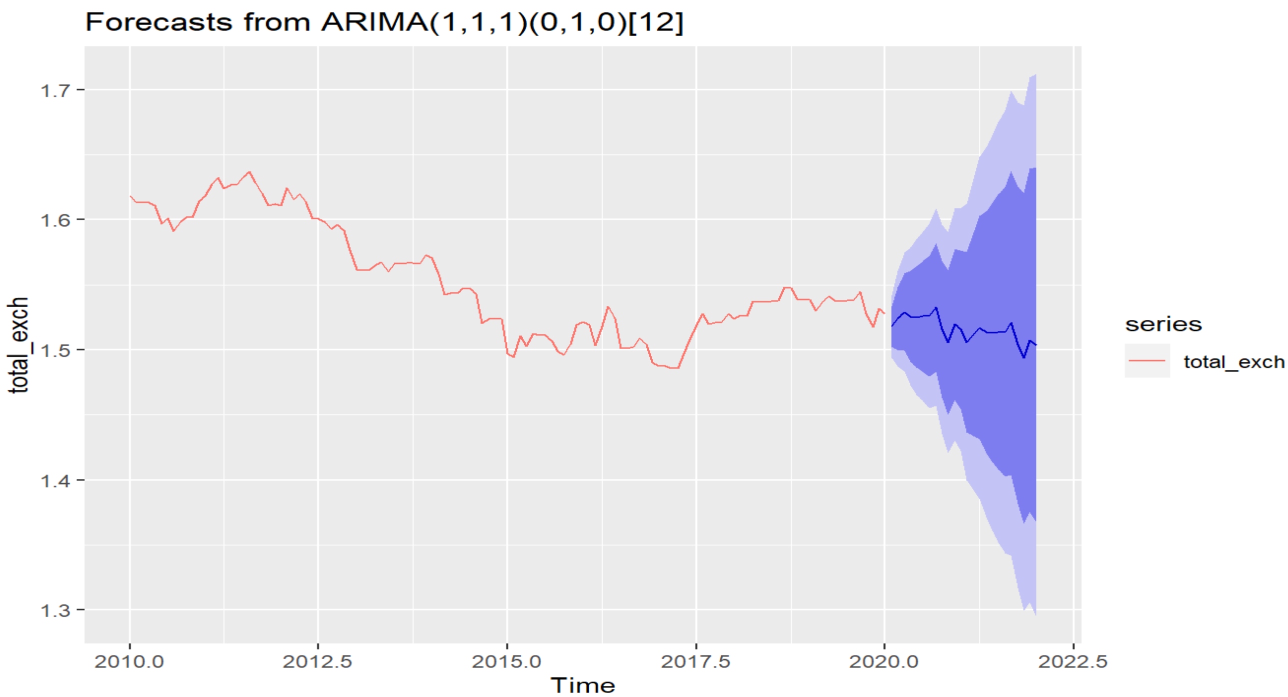
```
accuracy(forecast(fittrain212 ,h=24), test_exch)
##                        ME       RMSE         MAE       MPE      MAPE      MASE
## Training set 0.0002458815 0.01066285 0.008168308 0.0163111 0.5282355 0.3189231
## Test set     0.0629301656 0.06676816 0.062930166 3.8935118 3.8935118 2.4570432
##                     ACF1 Theil's U
## Training set -0.01608847        NA
## Test set      0.77445045  9.624916
```

3.9.6. ARIMA Forecast and Conclusion

Comparing the Root Mean Square and Mean Absolute Percentage Errors for all the three points, ARIMA (1, 1, 1) (0, 1, 0)[12] has the least error values hence accurate compared to the other points.

```{r}
accuracy(forecast(fittrain110, h=24), test_exch)
accuracy(forecast(fittrain111, h=24), test_exch)
accuracy(forecast(fittrain212 ,h=24), test_exch)
```

```
                       ME       RMSE         MAE        MPE      MAPE      MASE       ACF1 Theil's U
Training set 0.0002809638 0.01210965 0.009142173 0.01828603 0.592574 0.3569467 0.02707039        NA
Test set     0.0511089296 0.05583798 0.051108930 3.16295712 3.162957 1.9954953 0.79649453  8.021124
                       ME       RMSE         MAE        MPE      MAPE      MASE       ACF1 Theil's U
Training set 0.0002708413 0.01200594 0.009099092 0.01766165 0.5897584 0.3552646 -0.0269132        NA
Test set     0.0527554841 0.05727679 0.052755484 3.26471685 3.2647169 2.0597833  0.7945842  8.237161
                       ME       RMSE         MAE       MPE      MAPE      MASE        ACF1 Theil's U
Training set 0.0002458815 0.01066285 0.008168308 0.0163111 0.5282355 0.3189231 -0.01608847        NA
Test set     0.0629301656 0.06676816 0.062930166 3.8935118 3.8935118 2.4570432  0.77445045  9.624916
```



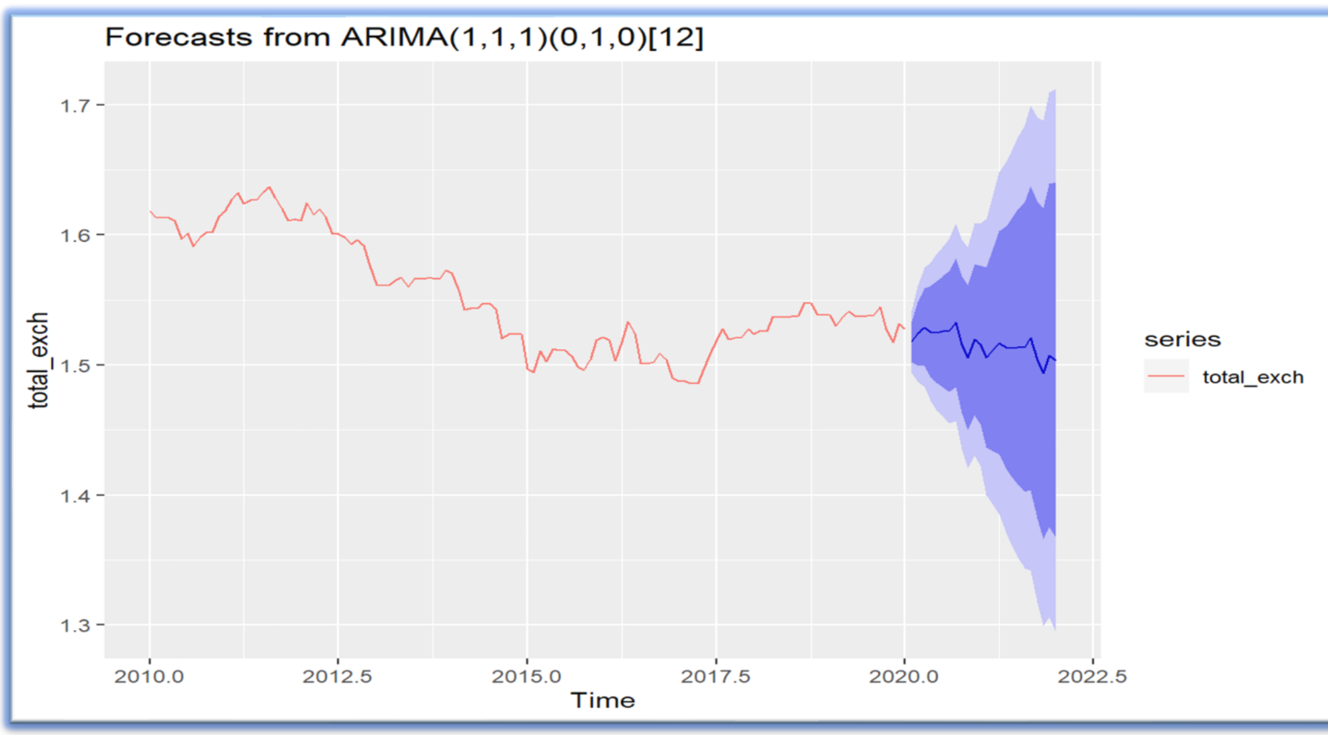
4. Forecast Results and Conclusions

Forecast Model After forecasting the exchange rate data from GBP to EUR from 2018 to 2020, the accuracies of statistical methods and machine learning methods are as below.

Table 3 Accuracy Matrix

METHOD	ME	RMSE	MAE	MAPE	Accuracy
Mean	0.05786772	0.05936853	0.05786772	3.575571	96.42
Naïve	0.0916682000	0.092622941	0.0916682	5.6680018	94.33
Drift	0.1020194	0.103566970	0.102019450	6.3066997	93.69
Holt-Winters	0.098891311	0.09986852	0.098891311	6.1177090	93.88
Holt-Winters (additive)	0.0829129661	0.08377256	0.08291297	5.1299051	94.87
Holt-Winters(multiplicative)	0.113276728	0.11468113	0.11327673	7.0058935	92.99
ARIMA(1,1,1)(0,1,0)[12]	0.0527554841	0.05727679	0.052755484	3.2647169	96.74
auto.arima() ARIMA(2,1,2)(0,1,0)[12]	0.0629301656	0.06676816	0.062930166	3.8935118	96.11

From all these observations, ARIMA model at (1, 1,1) (0,1,0) has the least values for all the errors MAPE 3.26 and highest accuracy of 96.73% . Forecasting the exchange rate from GBP to EUR from 2020 to 2022 (24 months), the ARIMA (1, 1, 1) (0, 1, 0) plot is shown below.



5. References

[1] “Forecasting exchange rates and countering currency risk | Currencies Direct,” *Currencies Direct*, 2020.

[2] J. Zheng, X. Fu, and G. Zhang, “Research on exchange rate forecasting based on deep belief network,” *Neural Computing and Applications*, vol. 31, 2019, doi: 10.1007/s00521-017-3039-z.

[3] “Analysis of foreign exchange rates using descriptive statistics.” https://www.ukessays.com/essays/economics/analysis-of-foreign-exchange-rates-using-descriptive-statistics-economics-essay.php (accessed Dec. 25, 2020).

[4] W. HUANG, K. K. LAI, Y. NAKAMORI, and S. WANG, “FORECASTING FOREIGN EXCHANGE RATES WITH ARTIFICIAL NEURAL NETWORKS: A REVIEW,” *International Journal of Information Technology & Decision Making*, vol. 03, no. 01, pp. 145–165, Mar. 2004, doi: 10.1142/S0219622004000969.

[5] “Time series analysis and forecasting explained.” https://blog.johngalt.com/time-series-analysis-forecasting-explained (accessed Dec. 26, 2020).

[6] “ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+.” https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/ (accessed Dec. 26, 2020).

[7] “Exchange Rate Prediction (Part 1): EDA & Data Visualisation | Analytics Vidhya.” https://medium.com/analytics-vidhya/exchange-rate-prediction-part-1-276b6cd5338 (accessed Dec. 26, 2020).

[8] “The Complete Guide to Time Series Analysis and Forecasting | by Marco Peixeiro | Towards Data Science.” https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775 (accessed Dec. 26, 2020).

[9] “Chapter 6 Time series decomposition | Forecasting: Principles and Practice.” https://otexts.com/fpp2/decomposition.html (accessed Dec. 27, 2020).

[10] “The Complete Guide to Time Series Analysis and Forecasting | by Marco Peixeiro | Towards Data Science.” https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775 (accessed Dec. 27, 2020).

[11] “5.6 Summary: stationarity testing | Applied Time Series Analysis for Fisheries and Environmental Sciences.” https://nwfsc-timeseries.github.io/atsa-labs/sec-boxjenkins-stationarity-summary.html (accessed Dec. 28, 2020).

[12] “7.3 Holt-Winters’ seasonal method | Forecasting: Principles and Practice.” https://otexts.com/fpp2/holt-winters.html (accessed Dec. 28, 2020).

[13] “Identifying the orders of AR and MA terms in an ARIMA model.” https://people.duke.edu/~rnau/411arim3.htm (accessed Dec. 28, 2020).

[14] "8.7 ARIMA modelling in R | Forecasting: Principles and Practice." https://otexts.com/fpp2/arima-r.html (accessed Dec. 28, 2020).

6. Appendices

6.1. Source Code

1. captureData.Rmd (Source code) , (R Notebook html) , (Output csv file)

captureData.Rmd captureData.nb.html raw_dataset.csv

captureData.Rmd

MohanaKamanooru

26/11/2020

Hide

```
#Defining constants
FILE_PATH          <- "raw_dataset.csv"
EURO_URL           <- 'https://www.ofx.com/en-gb/forex-news/historical-exchange-rates/'
FROM_CURRENCY      <- 'GBP British Pound'
TO_CURRENCY        <- 'EUR Euro'
TIME_FREQ          <- 'daily'
TIME_PERIOD        <- 'Last 10 years'
FROM_CURRENCY_ID   <- "select2-csid-container"
TO_CURRENCY_ID     <- "select2-20lw-container"
TIME_PERIOD_ID     <- "select2-xgvk-container"
TIME_FREQ_ID       <- "historicalrates-frequency-daily"
BUTTON_PATH        <- '/html/body/div[1]/main/div[2]/div[1]/div/div/div/div[5]/div/div/button/'
```

```
# Defining the function to create Selenium web Servers -------------------

euro_remote_server_driver <- function(local_port , local_browser, local_version,
local_verbose, local_check){

  euro_rsd <- rsDriver( port = local_port ,

            browser = local_browser ,

            #version = local_version,

            verbose = local_verbose ,

            check = local_check)

  euro_rcd <- euro_rsd$client

  return(euro_rcd)


  }


euro_close_connection <- function(remDriver){

  remDriver$close()

  return()

}


# Element accessing function ---------------------------------------------


set_euro_element <- function(set_value, euro_element_id , client_driver , euro_at
tr){

  euro_element <- client_driver$findElement('id', euro_element_id)

  euro_element$setElementAttribute(euro_attr,set_value)

}


get_euro_element <- function(euro_element_id , client_driver , euro_attr){

  euro_element <- client_driver$findElement(euro_attr, euro_element_id)

  return (euro_element)

}



## creating web drivers

# euro_client_driver <- ecd ----------------------------------------------
```

```
ecd <- euro_remote_server_driver(
  45226L,'firefox',
  'latest',FALSE,FALSE)

ecd$navigate(EURO_URL)
ecd$maxWindowSize()
sourcetext <- ecd$getPageSource()
view(sourcetext)
sourcetext

RESULT_TABLE_ID    <- 'historical-rates--table'
##setting the input parameters

from_currency_element <- set_euro_element( FROM_CURRENCY, FROM_CURRENCY_ID , ecd,
'title')
to_currency_element   <- set_euro_element( TO_CURRENCY, TO_CURRENCY_ID, ecd,'titl
e')
time_freq_element     <- set_euro_element( TIME_FREQ , TIME_FREQ_ID, ecd,'value')
time_period_element   <- set_euro_element( TIME_PERIOD, TIME_PERIOD_ID, ecd,'titl
e')
button_element <- ecd$findElement("xpath" , BUTTON_PATH)
button_element$clickElement()

result_table_element <- ecd$findElement('class', RESULT_TABLE_ID)
raw_table_data <- result_table_element$getElementText()

# close connection --------------------------------------------------------
euro_close_connection()

# Writing captured data into csv ------------------------------------------
write.csv(raw_table_data,file = FILE_PATH )
```

2. exchangeForecast.Rmd

exchangeForecast.Rmd exchangeForecast.html

exchangeForecast

MohanaKamanooru

28/12/2020

Pre-Processing the captured raw data set

```
#reading the dataset and creating a data frame
exch_raw_dataset <- read_delim(FILE_PATH,"\t", escape_double = FALSE, trim_ws = T
RUE, col_names = FALSE)
```

```
##
## -- Column specification -------------------------------------------------------
--
## cols(
##   X1 = col_character()
## )
```

```
exch_raw_dataset <- exch_raw_dataset[-c(1,2,3),]

col_name <- colnames(exch_raw_dataset)

exch_raw_dataset <- separate(exch_raw_dataset,
                             col_name,into = c("date","month","year","Rate"),
                             convert = TRUE,
                             sep = " " ,
                             )
head(exch_raw_dataset)
```

```
## # A tibble: 6 x 4
##    date month  year Rate
##   <int> <chr> <int> <chr>
## 1     2 Jan    1990 1.611
```

```
## 2      3 Jan    1990 1.61
## 3      4 Jan    1990 1.6293
## 4      5 Jan    1990 1.637
## 5      8 Jan    1990 1.653
## 6      9 Jan    1990 1.6537
```

```
summary(exch_raw_dataset)
```

```
##        date           month                 year           Rate
##   Min.   : 1.00   Length:9741        Min.   :1990   Length:9741
##   1st Qu.: 8.00   Class :character   1st Qu.:1999   Class :character
##   Median :16.00   Mode  :character   Median :2007   Mode  :character
##   Mean   :15.76                      Mean   :2006
##   3rd Qu.:23.00                      3rd Qu.:2014
##   Max.   :31.00                      Max.   :2020
```

Date Formatting

```
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
## Warning in data.frame(exch_date_format, as.numeric(exch_raw_dataset$Rate)): NA
s
## introduced by coercion
##          Date   Rate
## 1 1990-01-02 1.6110
## 2 1990-01-03 1.6100
## 3 1990-01-04 1.6293
## 4 1990-01-05 1.6370
## 5 1990-01-08 1.6530
## 6 1990-01-09 1.6537
```

Filtering the processed dataset

```
## [1] "Date" "Rate"
## 'data.frame':    3924 obs. of  2 variables:
##  $ Date: Date, format: "2010-01-01" "2010-01-02" ...
##  $ Rate: num  1.62 1.61 1.61 1.61 1.61 ...
##         Date     Rate
## 1 2010-01-01 1.618450
## 2 2010-01-02 1.613007
## 3 2010-01-03 1.613046
## 4 2010-01-04 1.613046
## 5 2010-01-05 1.610954
## 6 2010-01-06 1.596704
##       Date                 Rate
##  Min.   :2010-01-01   Min.   :1.148
##  1st Qu.:2012-09-15   1st Qu.:1.305
##  Median :2015-05-31   Median :1.512
##  Mean   :2015-05-29   Mean   :1.458
##  3rd Qu.:2018-02-11   3rd Qu.:1.584
##  Max.   :2020-10-30   Max.   :1.716
```

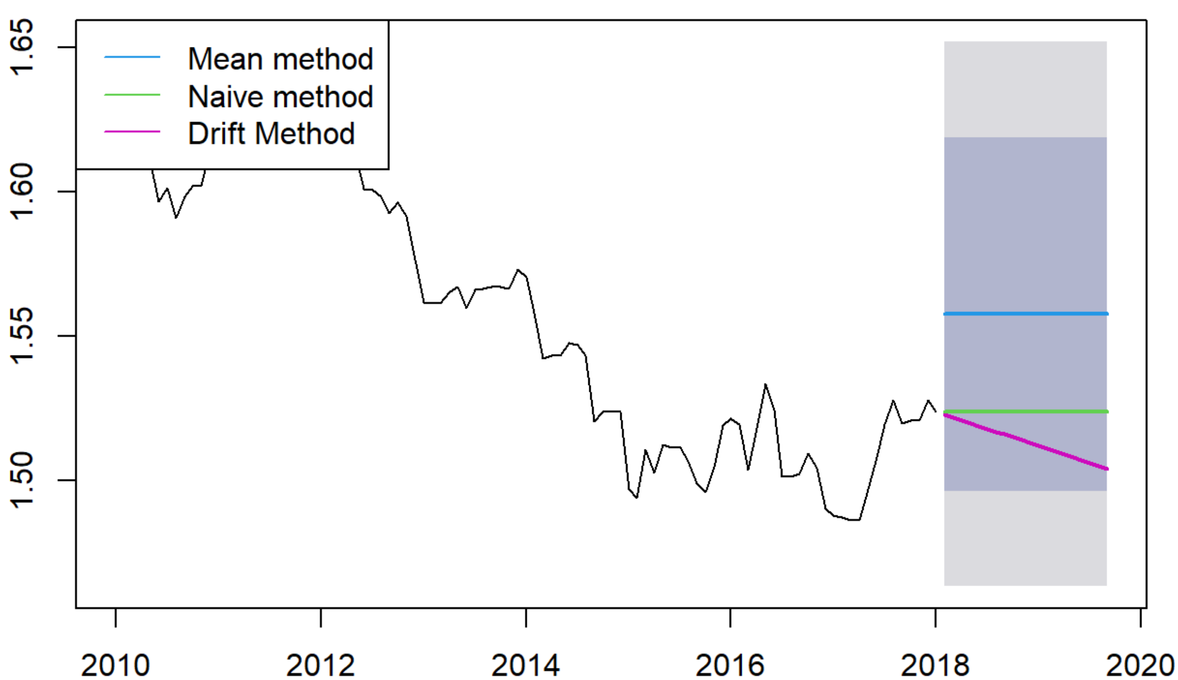
#Test and Train Data Sampling

```
# Sampling data 80% for training and 20% testing


total_exch <- ts(exch_processed_df$Rate, start=c(2010), frequency = 12 , end = (2
020))
train_exch<- ts(exch_processed_df$Rate, start=c(2010), frequency = 12 , end = (20
18))
test_exch<- ts(exch_processed_df$Rate, start=c(2018), frequency = 12 , end = (202
0))
```

#Train and Test <— Naive, Mean and Drift

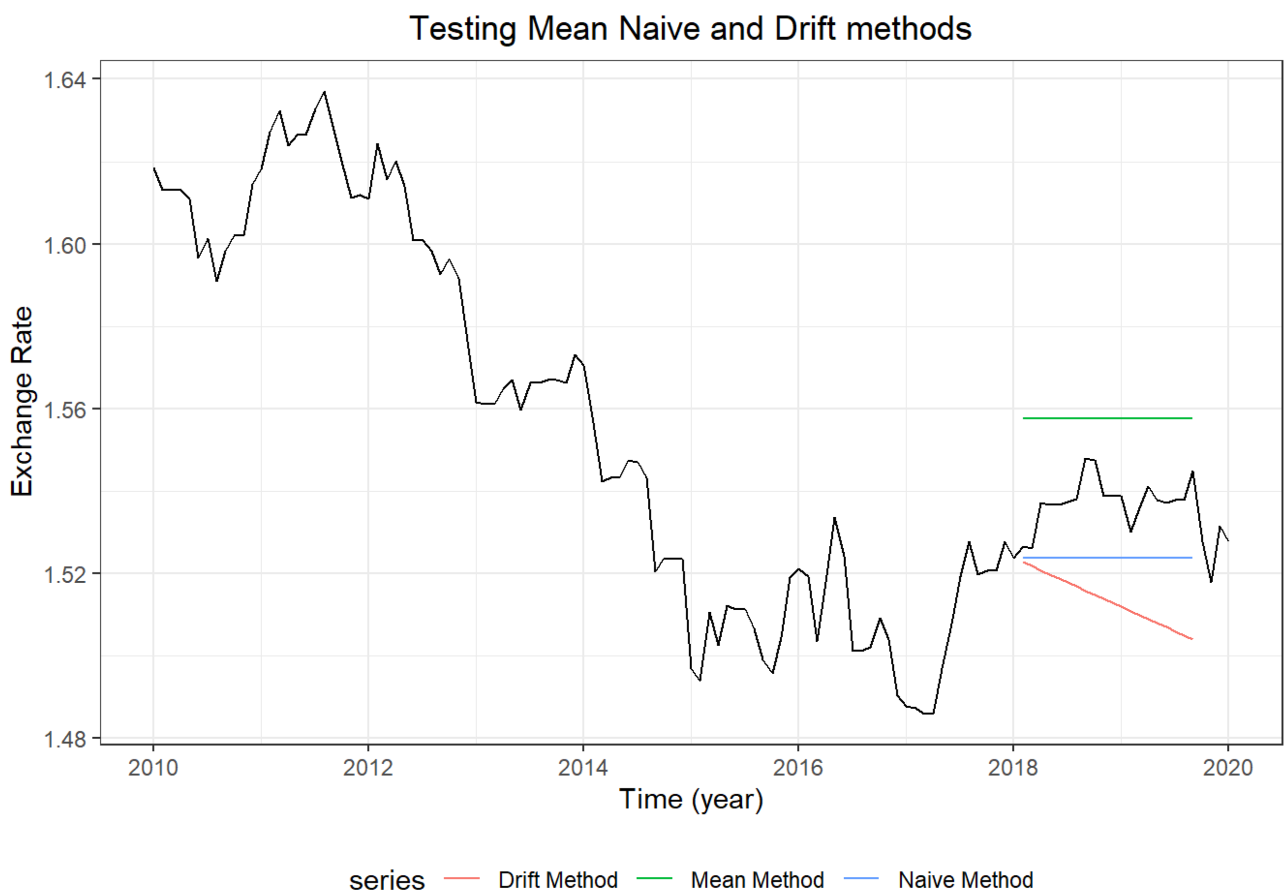
```
#forecast by taking the mean of the values
#?meanf
mean_ex <- meanf(train_exch, h=20)
```

```
#forecast by taking the last observation forward
#?naive
naive_ex<- naive(train_exch, h=20)

#forecast by drift model- equivalent to an ARIMA(0,1,0)
drift_ex <- rwf(train_exch, h=20, drift = T)

plot(mean_ex,main = "")
lines(naive_ex$mean, col=123, lwd = 2)
lines(drift_ex$mean, col=22, lwd = 2)
legend("topleft",lty=1,col=c(4,123,22),
       legend=c("Mean method","Naive method","Drift Method"))
```




```
#forecast and compare against testing data
```

```
autoplot(total_exch) +
  autolayer(mean_ex$mean, series = "Mean Method") +
  autolayer(naive_ex$mean, series = "Naive Method") +
  autolayer(drift_ex$mean, series = "Drift Method") +
  ggtitle("Testing Mean Naive and Drift methods") +
  xlab("Time (year)") + ylab("Exchange Rate") +
  theme_bw() + theme(plot.title = element_text(hjust = 0.5),legend.position = "bo
ttom")
```



Accuracy <— Naive Mean and Drift

```
# Checking the accuracy
accuracy(mean_ex, test_exch)
```

```
##                         ME       RMSE        MAE         MPE     MAPE     MASE
## Training set -6.409926e-17 0.04702457 0.04214583 -0.09095092 2.704796 1.645540
```

```
## Test set      5.786772e-02 0.05936853 0.05786772  3.57557085 3.575571 2.259385
##                     ACF1 Theil's U
## Training set 0.9726000        NA
## Test set     0.8490381  8.706142
```

```
accuracy(naive_ex, test_exch)
```

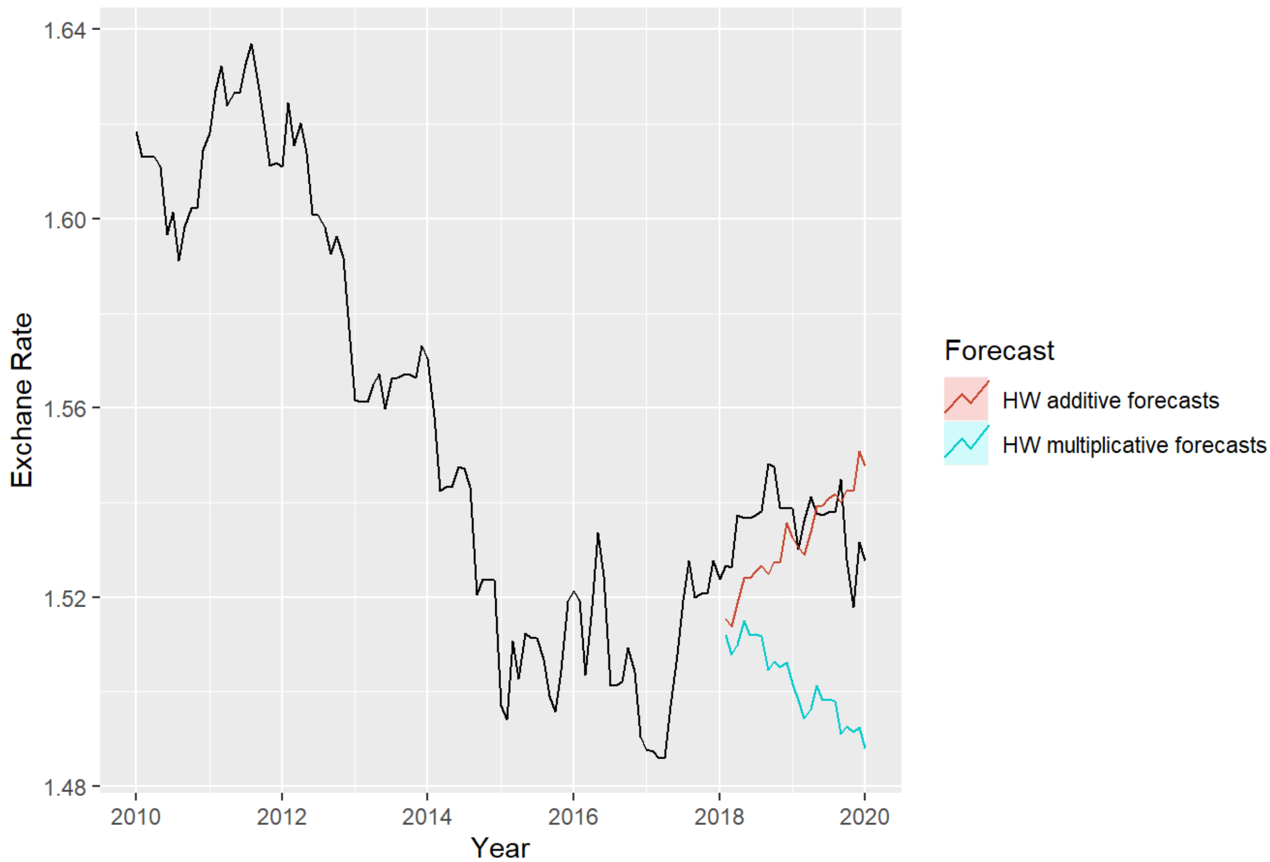
```
##                         ME        RMSE       MAE         MPE      MAPE
## Training set -0.0009858333 0.008484374 0.0061780 -0.06428024 0.3986106
## Test set      0.0916682000 0.092622941 0.0916682  5.66800182 5.6680018
##                     MASE      ACF1 Theil's U
## Training set 0.2412136 0.1021370        NA
## Test set     3.5790900 0.8490381  13.57754
```

```
accuracy(drift_ex, test_exch)
```

```
##                         ME        RMSE         MAE           MPE      MAPE
## Training set -9.020562e-17 0.008426906 0.006302281 -0.0009059021 0.4064263
## Test set      1.020194e-01 0.103566970 0.102019450  6.3066996780 6.3066997
##                    MASE      ACF1 Theil's U
## Training set 0.246066 0.1021370        NA
## Test set     3.983244 0.8757613  15.24293
```

Train and Test <— Holt-Winters Seasonal (additive , multiplicative)

```
fit1 <- hw(train_exch,seasonal="additive")
fit2 <- hw(train_exch,seasonal="multiplicative")

autoplot(total_exch) +
  autolayer(fit1, series="HW additive forecasts", PI=FALSE) +
  autolayer(fit2, series="HW multiplicative forecasts",
    PI=FALSE) +
  xlab("Year") +
  ylab("Exchane Rate") +
  guides(colour=guide_legend(title="Forecast"))
```



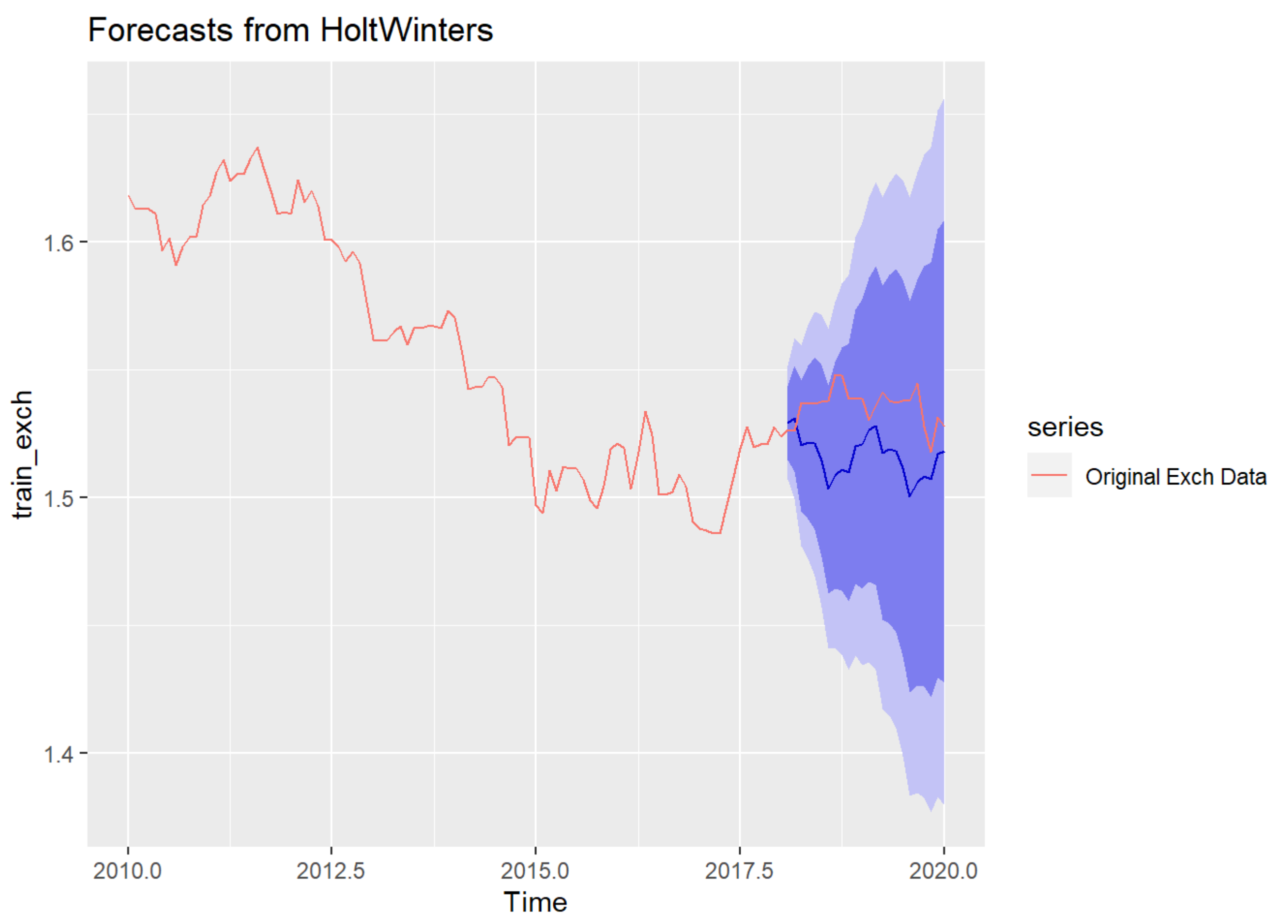
Accuracy <– Holt-Winters Seasonal (additive , multiplicative)

```
# Accuracy
accuracy(forecast(fit1 , h=24), test_exch)
##                        ME       RMSE        MAE        MPE      MAPE      MASE
## Training set 0.0006623137 0.01393349 0.01152412 0.04480054 0.7411147 0.4499474
## Test set     0.0829129661 0.08377256 0.08291297 5.12990508 5.1299051 3.2372509
##                   ACF1 Theil's U
## Training set 0.7957722        NA
## Test set     0.7275861  12.36527
accuracy(forecast(fit2 , h=24), test_exch)
##                       ME       RMSE        MAE         MPE      MAPE      MASE
```

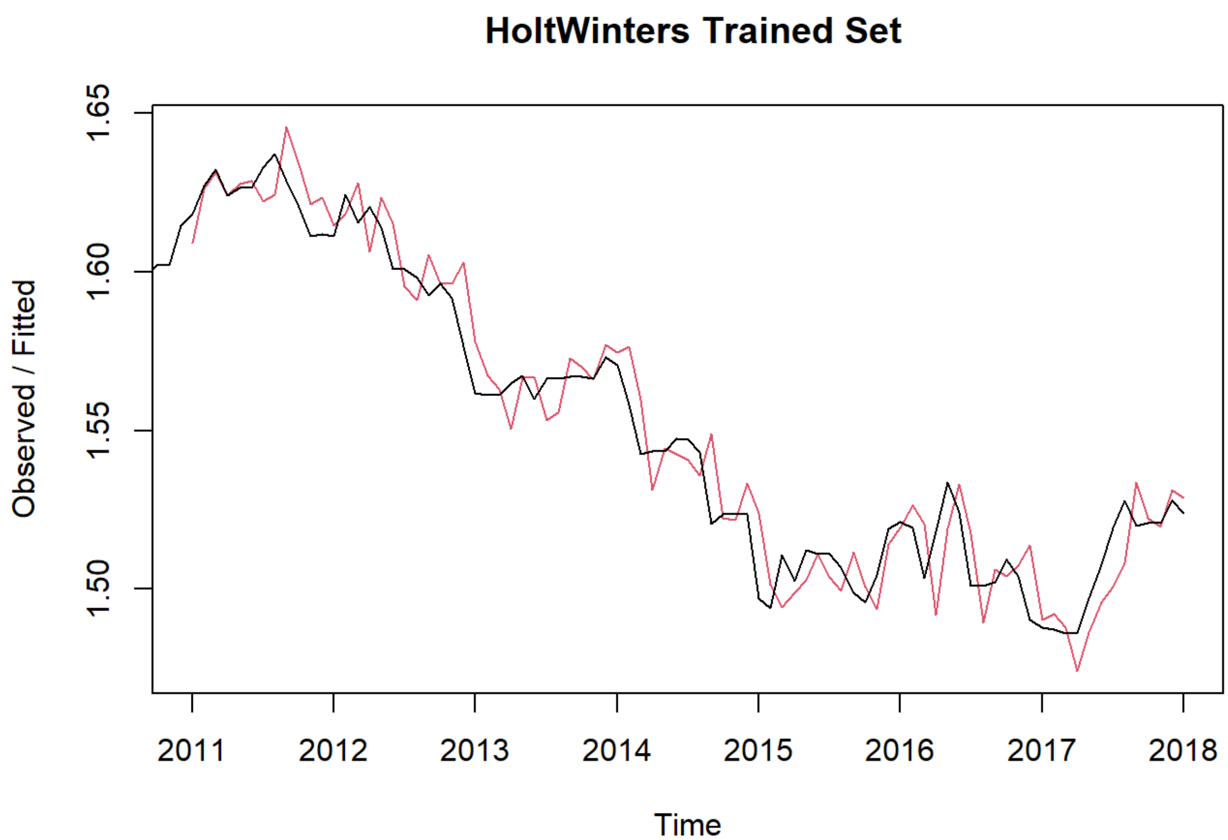
```
## Training set 0.000254011 0.01356191 0.01086125 0.01264748 0.6983993 0.4240665
## Test set     0.113276728 0.11468113 0.11327673 7.00589346 7.0058935 4.4227726
##                      ACF1 Theil's U
## Training set 0.7478286        NA
## Test set     0.8982091  17.12899
```

Train and Test <— Holt-Winters

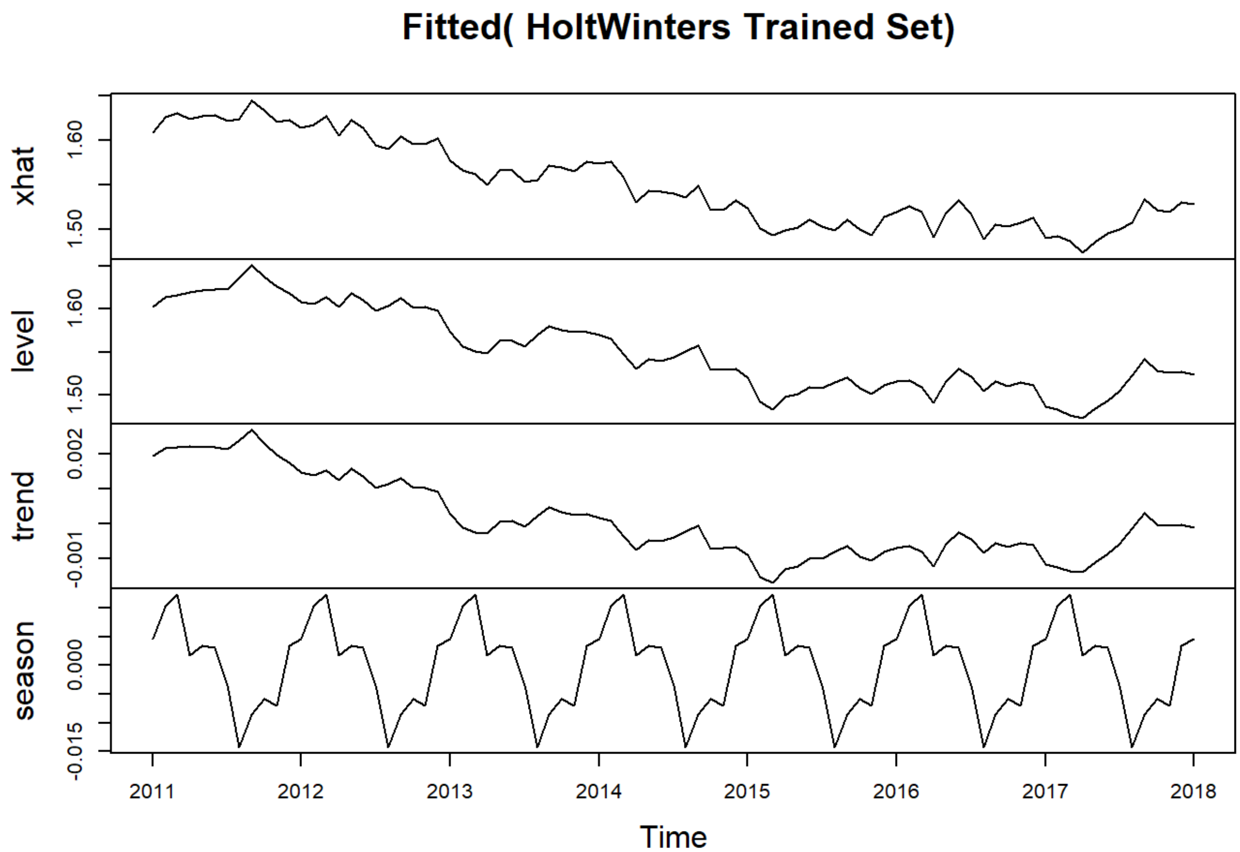
```
train_holt <-HoltWinters(train_exch , gamma = 0)


train_holt %>%
  forecast(h=24) %>%
  autoplot(series = "Forecast Data") + autolayer(total_exch, series = "Original E
xch Data")
```



```
plot(train_holt, main = " HoltWinters Trained Set")
```




```
plot(fitted(train_holt) ,main = " Fitted( HoltWinters Trained Set)")
```



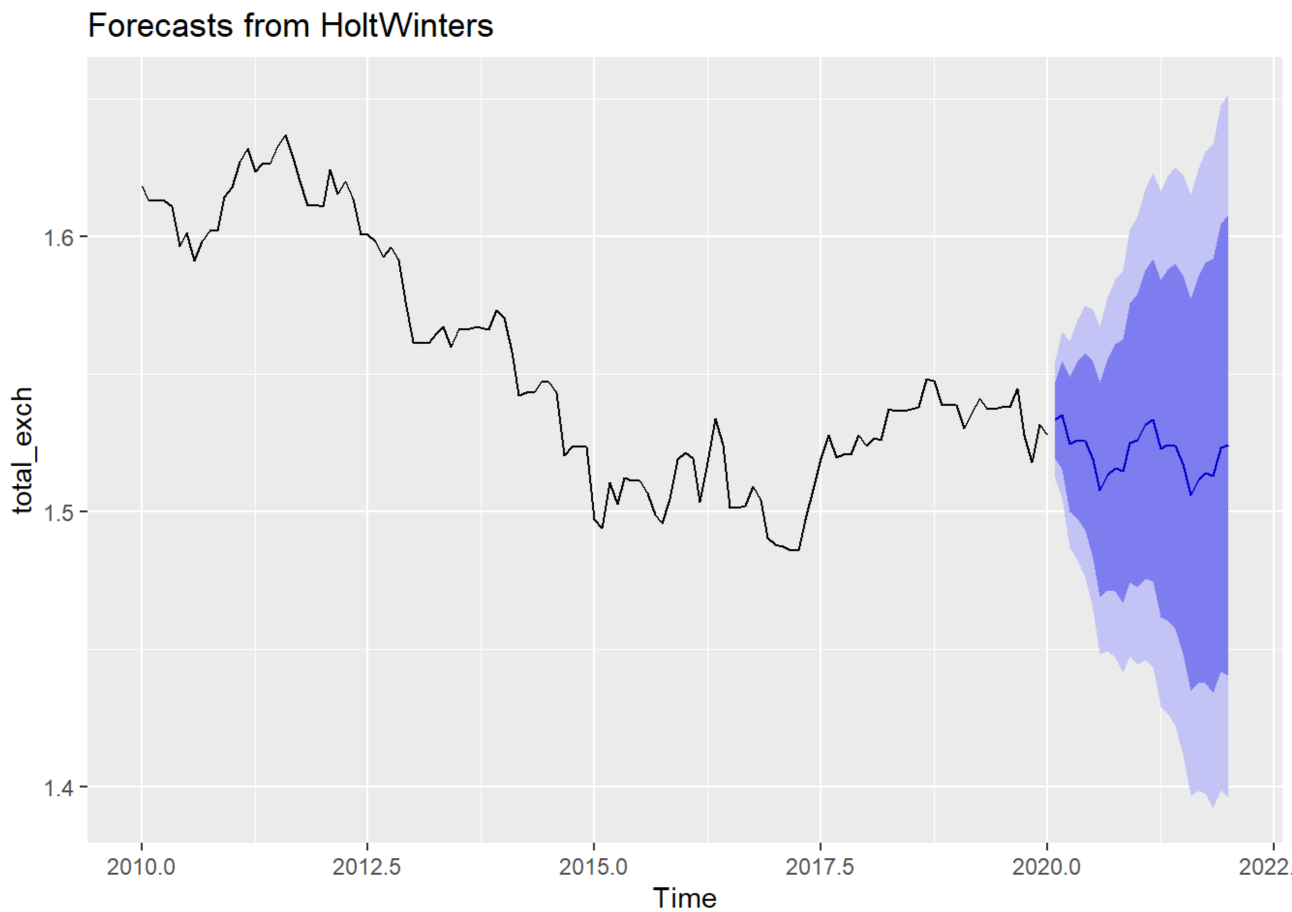
Accuracy <– Holt-Winters

```
accuracy(forecast(train_holt, h=24), test_exch)
##                        ME       RMSE         MAE         MPE      MAPE
## Training set -0.001086076 0.01123869 0.008930055 -0.06947983 0.5778938
## Test set      0.098891311 0.09986852 0.098891311  6.11770905 6.1177090
##                   MASE      ACF1 Theil's U
## Training set 0.3486648 0.1468536        NA
## Test set     3.8611089 0.8241218  14.94505
```

Forecast for 24months using HoltWinters

```
total_holt <- HoltWinters(total_exch,gamma=0)
total_holt %>%
```

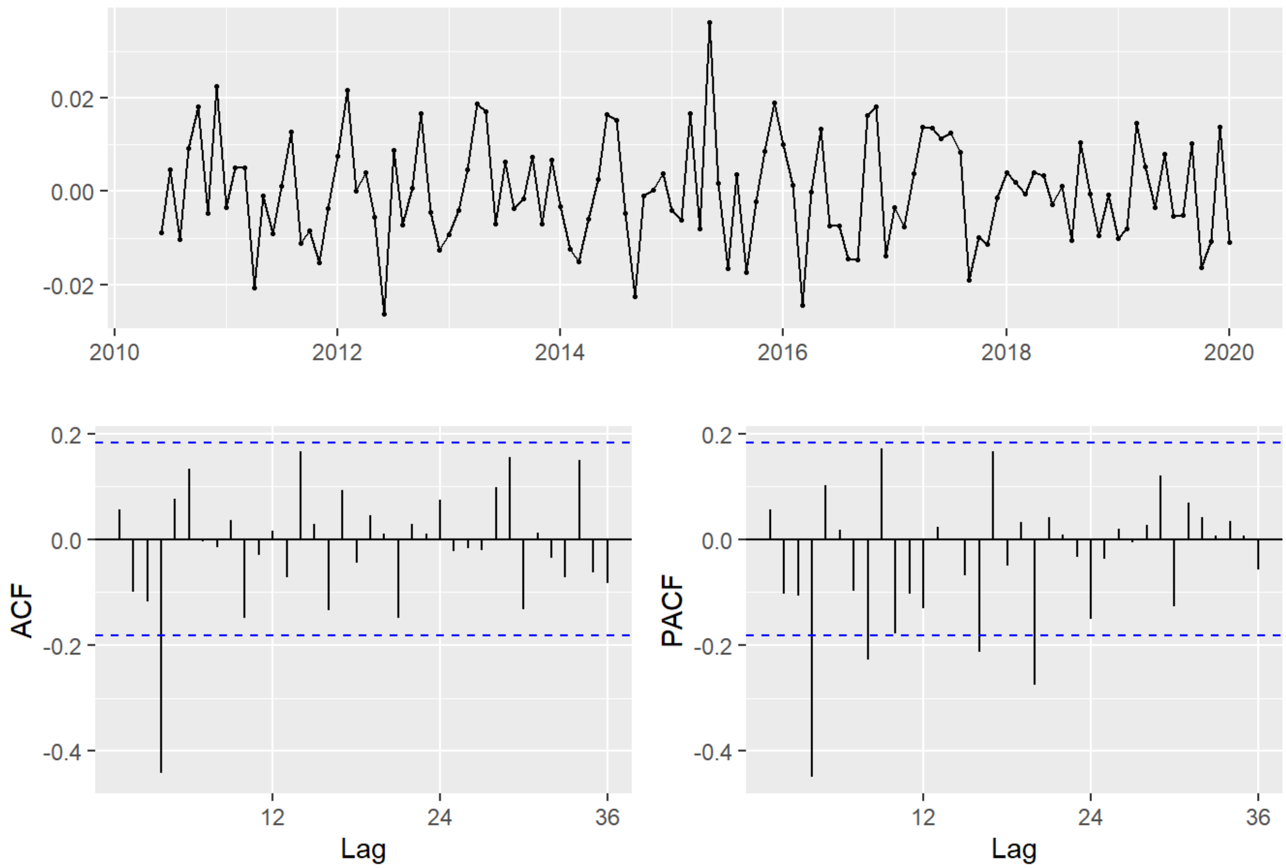
```
  forecast(h=24) %>%
  autoplot(series = "Forecast Data")
```



#ARIMA

Diiferencing the series to make it stationary

```
total_exch %>% diff(lag=4) %>%diff()%>% ggtsdisplay()
```




```
#to check how stationary the time series is.
#?adf.test4
# Augmented Dickey-Fuller Test - Computes the Augmented Dickey-Fuller test for th
e null that x has a unit root.
adf.test(total_exch)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  total_exch
## Dickey-Fuller = -1.1493, Lag order = 4, p-value = 0.9113
## alternative hypothesis: stationary
```

```
adf.test(diff(total_exch, differences = 1))
```

```
## Warning in adf.test(diff(total_exch, differences = 1)): p-value smaller than
## printed p-value
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  diff(total_exch, differences = 1)
## Dickey-Fuller = -4.428, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
adf.test(diff(total_exch, differences = 2))
## Warning in adf.test(diff(total_exch, differences = 2)): p-value smaller than
## printed p-value
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(total_exch, differences = 2)
## Dickey-Fuller = -8.7937, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
# differentiating higher order -- over fitting the series( can get high accuracy
for the test data but may not forecast better values for future)
```

Predicting the P, D and Q values for seasonal component

```
Arima(total_exch, order=c(0,1,1), seasonal=c(0,1,0)) #AIC=-642.85   AICc=-642.74
BIC=-637.49
## Series: total_exch
## ARIMA(0,1,1)(0,1,0)[12]
##
## Coefficients:
##          ma1
##       0.1745
## s.e.  0.1061
##
## sigma^2 estimated as 0.0001483:  log likelihood=323.43
## AIC=-642.85   AICc=-642.74   BIC=-637.49
Arima(total_exch, order=c(0,1,1), seasonal=c(1,1,0)) #AIC=-687.12   AICc=-686.89
BIC=-679.07
```

```
## Series: total_exch
## ARIMA(0,1,1)(1,1,0)[12]
##
## Coefficients:
##          ma1     sar1
##       0.1706  -0.5921
## s.e.  0.1160   0.0731
##
## sigma^2 estimated as 9.311e-05:  log likelihood=346.56
## AIC=-687.12   AICc=-686.89   BIC=-679.07
```

```
Arima(total_exch, order=c(0,1,1), seasonal=c(0,1,1)) #AIC=-699.35   AICc=-699.12   BIC=-691.3
```

```
## Series: total_exch
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##          ma1     sma1
##       0.1591  -0.9997
## s.e.  0.1106   0.1904
##
## sigma^2 estimated as 6.764e-05:  log likelihood=352.67
## AIC=-699.35   AICc=-699.12   BIC=-691.3
```

Predicting the p, d and q for ARIMA point

```
Arima(total_exch, order=c(0,1,0), seasonal=c(0,1,0)) #AIC=-642.55   AICc=-642.51   BIC=-639.87
```

```
## Series: total_exch
## ARIMA(0,1,0)(0,1,0)[12]
##
## sigma^2 estimated as 0.0001502:  log likelihood=322.27
## AIC=-642.55   AICc=-642.51   BIC=-639.87
```

```
Arima(total_exch, order=c(1,1,0), seasonal=c(0,1,0)) #AIC=-641.96   AICc=-641.84   BIC=-636.59
```

```
## Series: total_exch
```

```
## ARIMA(1,1,0)(0,1,0)[12]
##
## Coefficients:
##          ar1
##       0.1140
## s.e.  0.0958
##
## sigma^2 estimated as 0.0001496:  log likelihood=322.98
## AIC=-641.96   AICc=-641.84   BIC=-636.59
```

```
Arima(total_exch, order=c(0,1,1), seasonal=c(0,1,0)) #AIC=-642.85   AICc=-642.74   BIC=-637.49
```

```
## Series: total_exch
## ARIMA(0,1,1)(0,1,0)[12]
##
## Coefficients:
##          ma1
##       0.1745
## s.e.  0.1061
##
## sigma^2 estimated as 0.0001483:  log likelihood=323.43
## AIC=-642.85   AICc=-642.74   BIC=-637.49
```

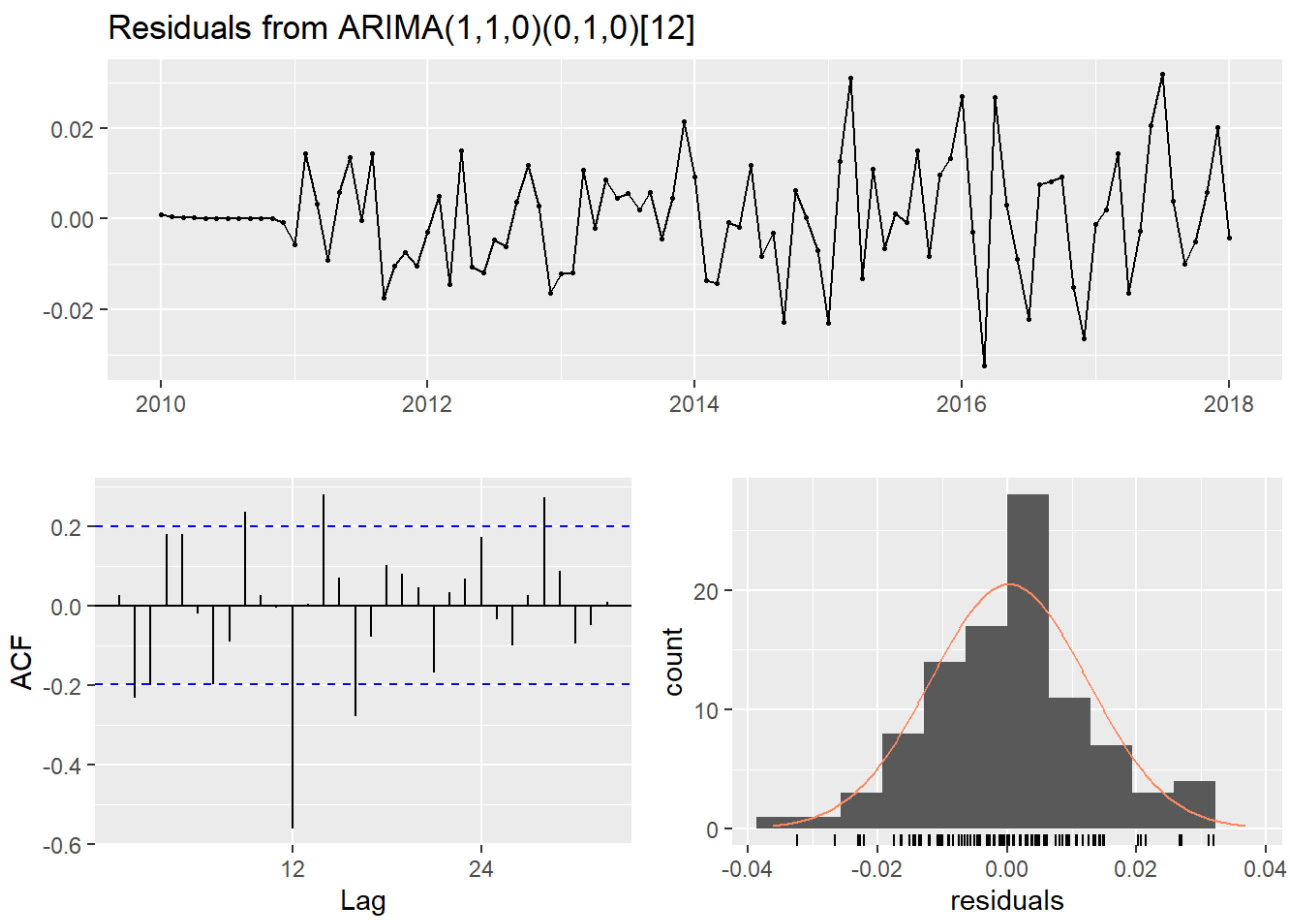
```
Arima(total_exch, order=c(1,1,1), seasonal=c(0,1,0)) #AIC=-641.42   AICc=-641.19   BIC=-633.38
```

```
## Series: total_exch
## ARIMA(1,1,1)(0,1,0)[12]
##
## Coefficients:
##           ar1     ma1
##       -0.2132  0.3740
## s.e.   0.2950  0.2693
##
## sigma^2 estimated as 0.0001489:  log likelihood=323.71
## AIC=-641.42   AICc=-641.19   BIC=-633.38
```

```
# AICc is low at seasonal c PDQ 111
```

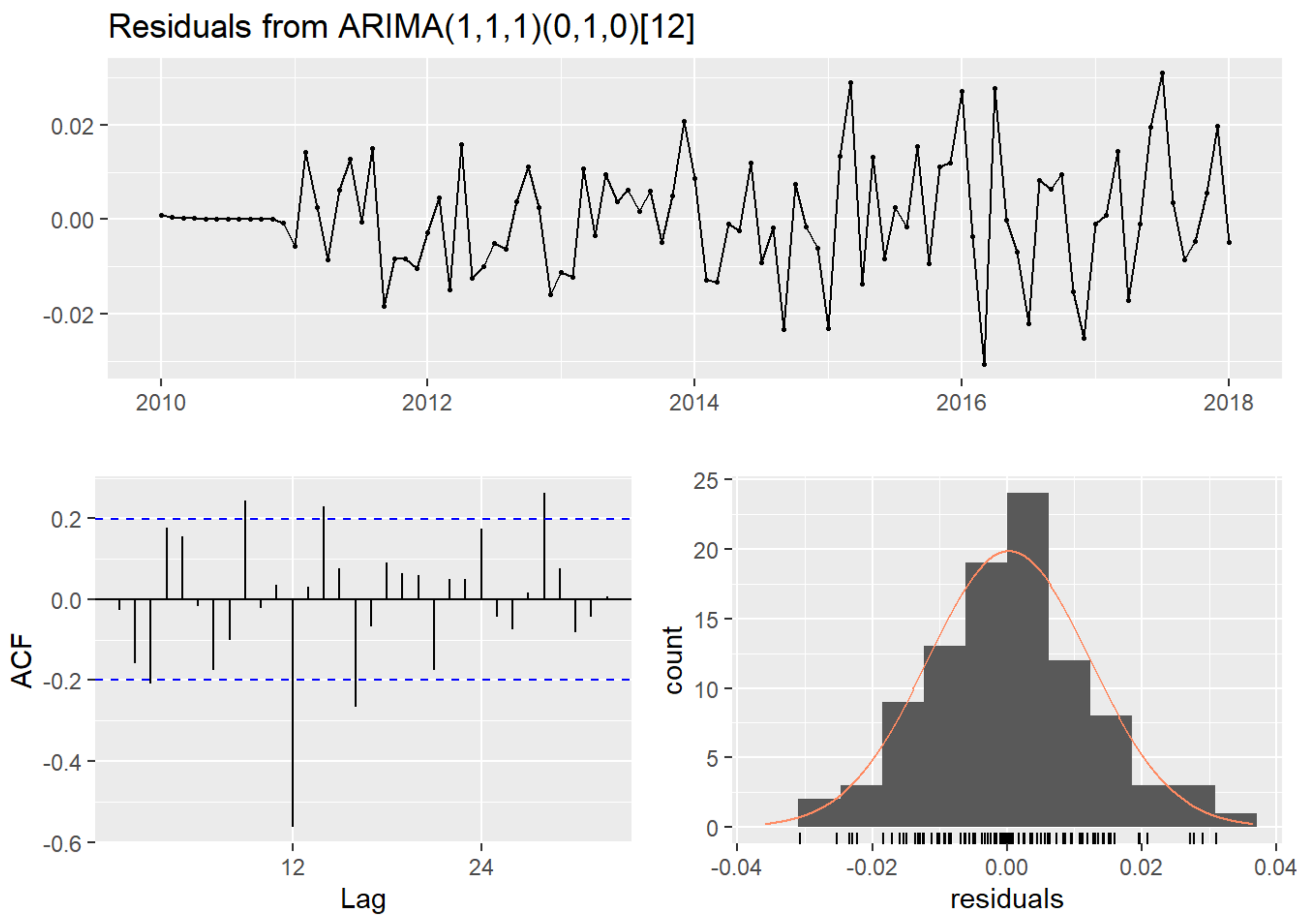
Training <— ARIMA (1,1,0) and ARIMA(1,1,1)

```
# Analysinsg Residulas and p Value

fit110 <- Arima(train_exch, order=c(1,1,0), seasonal=c(0,1,0)) # p-value = 1.28e-10

checkresiduals(fit110)
```



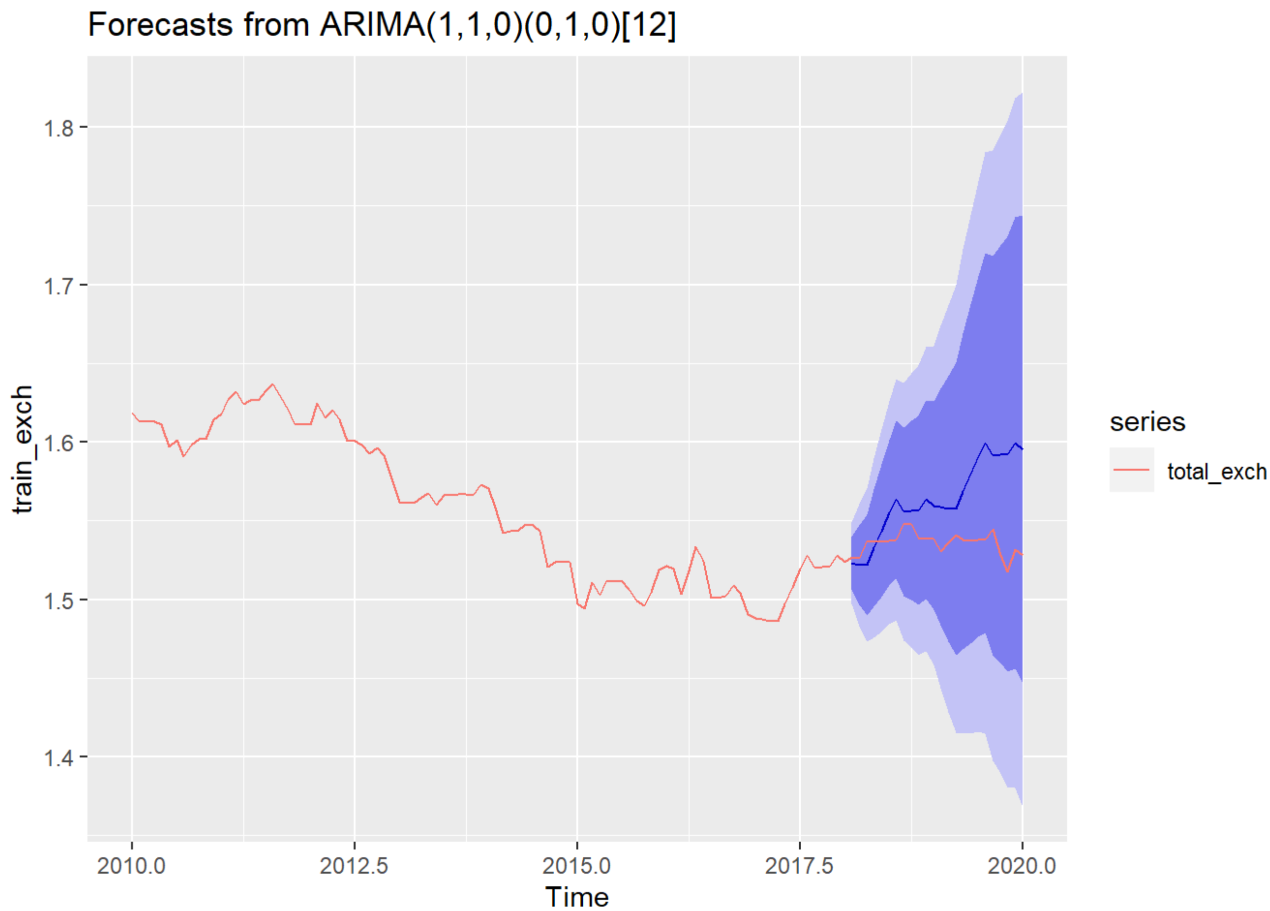

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(0,1,0)[12]
## Q* = 84.687, df = 18, p-value = 1.28e-10
```

```
##
## Model df: 1.   Total lags used: 19
fit111 <- Arima(train_exch, order=c(1,1,1), seasonal=c(0,1,0)) # p-value = 1.297e
-09
checkresiduals(fit111)
```

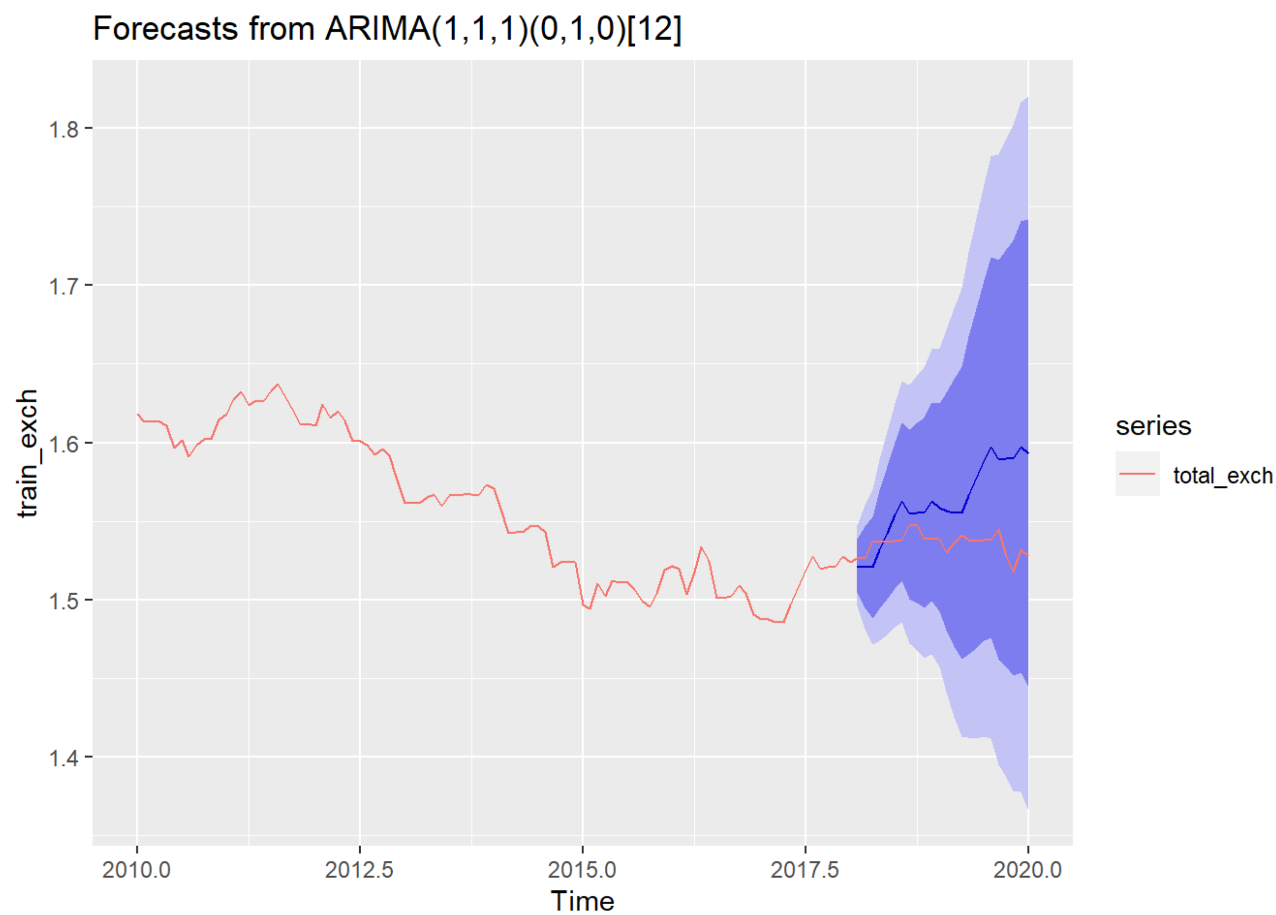



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,1,0)[12]
## Q* = 77.008, df = 17, p-value = 1.297e-09
##
## Model df: 2.   Total lags used: 19
# training

fittrain110 <- Arima(train_exch, order=c(1,1,0), seasonal=c(0,1,0))
```

```
fittrain110 %>% forecast(h=24) %>% autoplot()+ autolayer(total_exch)
```




```
fittrain111 <- Arima(train_exch, order=c(1,1,1), seasonal=c(0,1,0))
fittrain111 %>% forecast(h=24) %>% autoplot()+ autolayer(total_exch)
```



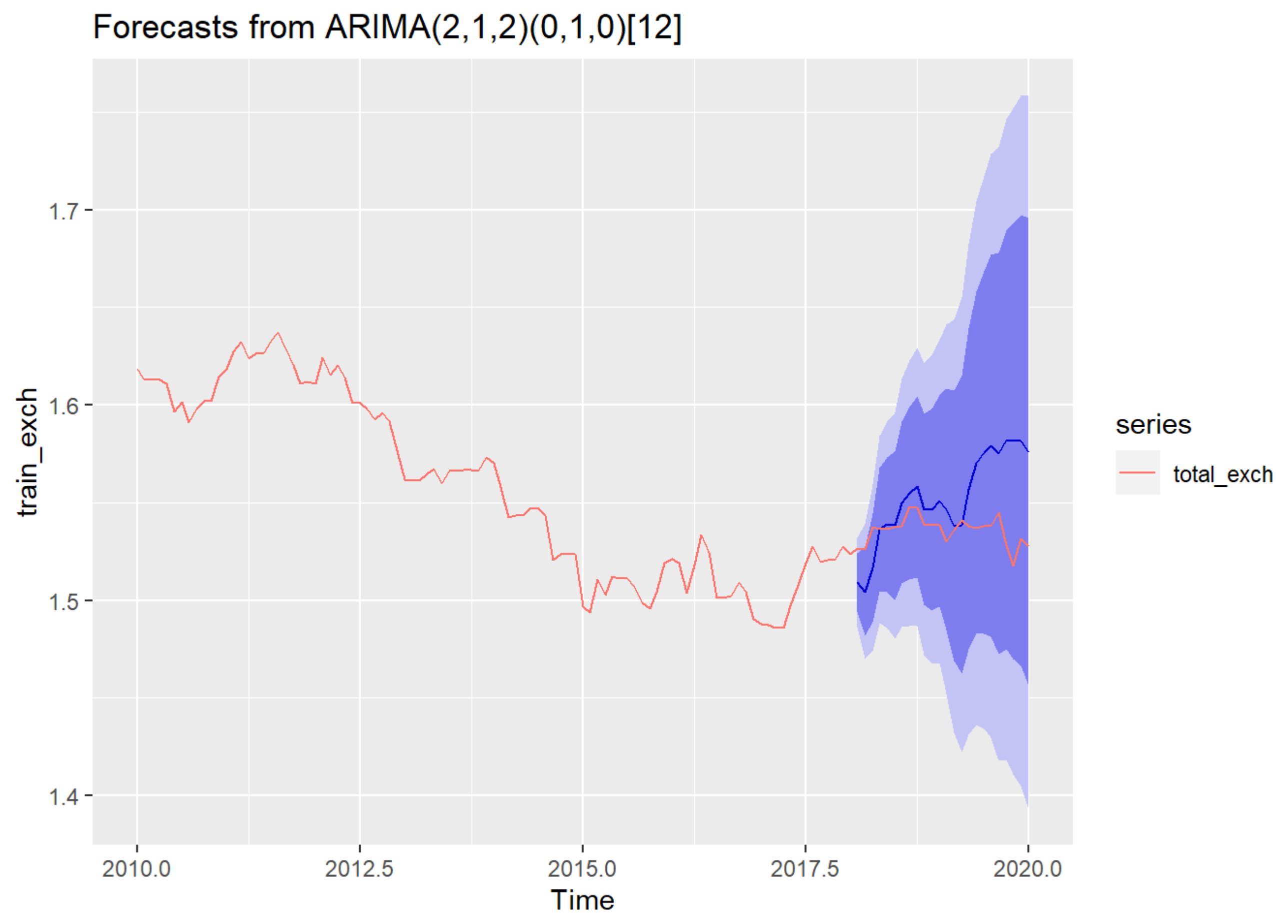
#Predicting - autoARIMA

```
auto.arima(total_exch, seasonal = TRUE , approximation = FALSE, stepwise = FALSE)
# ARIMA(2,1,2)
```

```
## Series: total_exch
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.4039  -0.9022  -0.3288  0.7997
## s.e.  0.1152   0.0770   0.1684  0.1107
##
## sigma^2 estimated as 6.401e-05:  log likelihood=410.99
## AIC=-811.99   AICc=-811.46   BIC=-798.05
```

Training and Testing <— Auto Arima(2,1,2)

```
fittrain212 <- Arima(train_exch, order=c(2,1,2), seasonal=c(0,1,0))
fittrain212 %>% forecast(h=24) %>% autoplot()+ autolayer(total_exch)
```



##Accuracy <— ARIMA (1,1,0) ARIMA(1,1,1) and ARIMA (2,1,2)

```
accuracy(forecast(fittrain110, h=24), test_exch)
##                        ME       RMSE         MAE        MPE     MAPE      MASE
## Training set 0.0002809638 0.01210965 0.009142173 0.01828603 0.592574 0.3569467
## Test set     0.0511089296 0.05583798 0.051108930 3.16295712 3.162957 1.9954953
##                    ACF1 Theil's U
## Training set 0.02707039        NA
## Test set     0.79649453  8.021124
```

```
accuracy(forecast(fittrain111, h=24), test_exch)
```

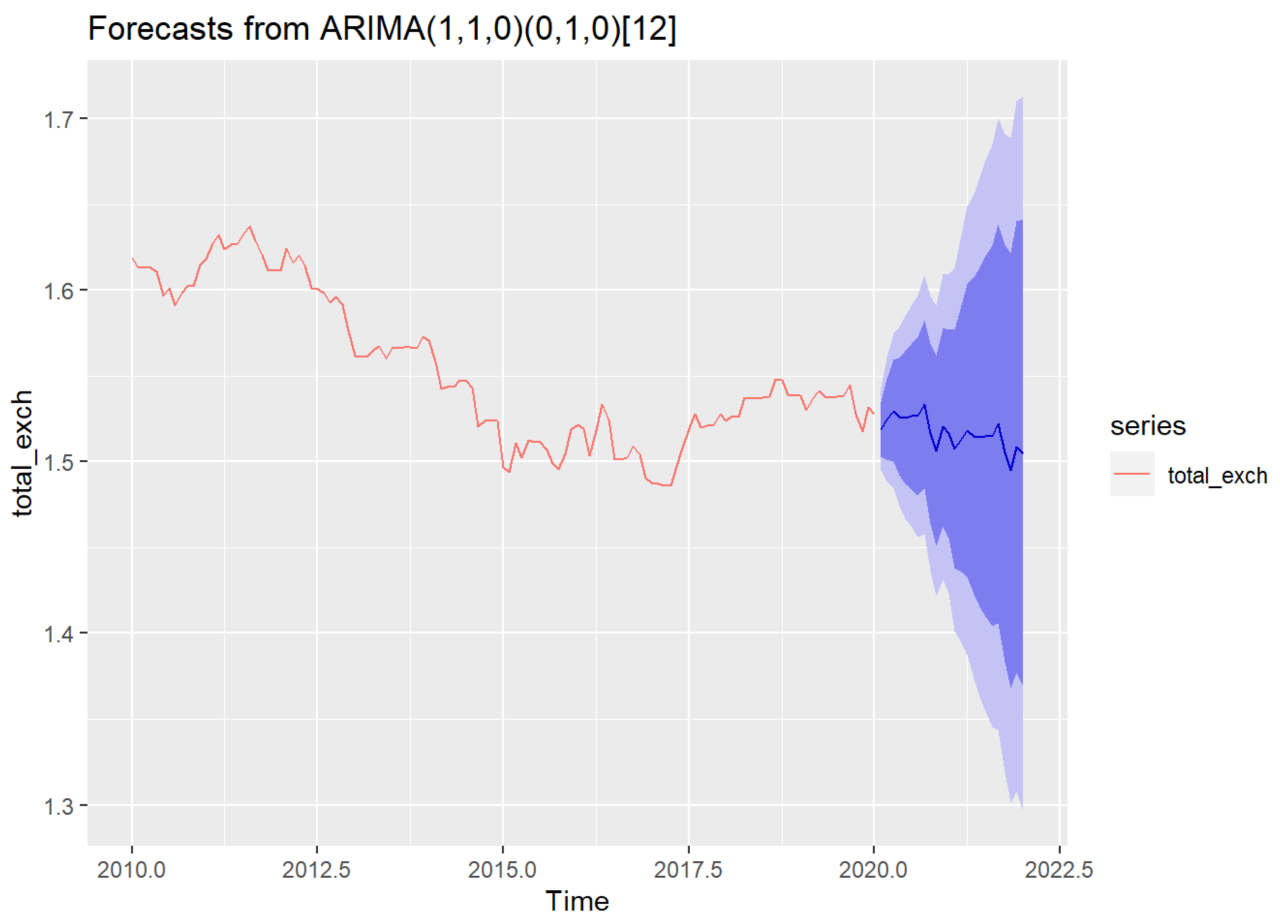
```
##                          ME       RMSE         MAE        MPE      MAPE      MAS
E
## Training set 0.0002708413 0.01200594 0.009099092 0.01766165 0.5897584 0.355264
6
## Test set     0.0527554841 0.05727679 0.052755484 3.26471685 3.2647169 2.059783
3
##                    ACF1 Theil's U
## Training set -0.0269132        NA
## Test set      0.7945842  8.237161
```

```
accuracy(forecast(fittrain212 ,h=24), test_exch)
```

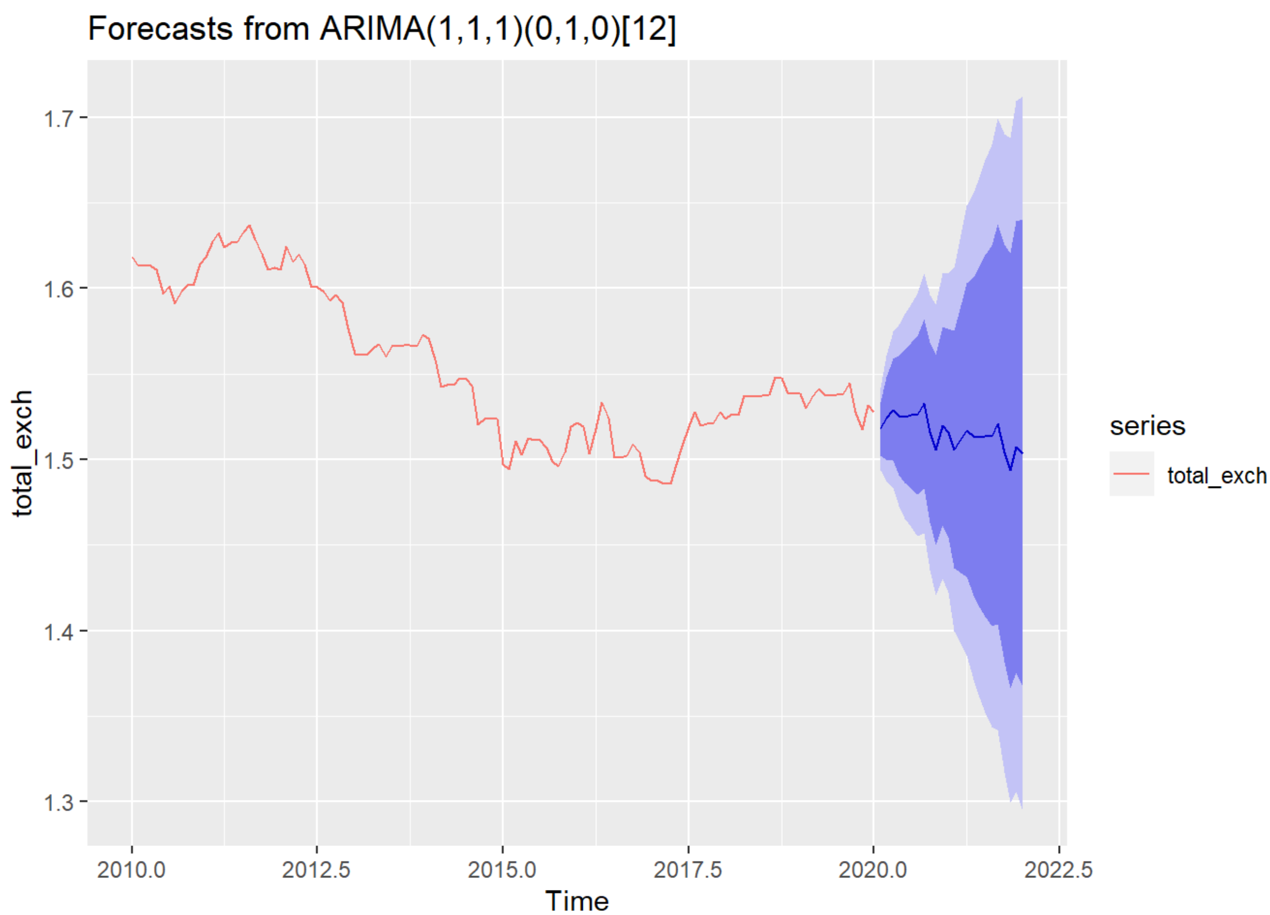
```
##                          ME       RMSE         MAE       MPE      MAPE      MASE
## Training set 0.0002458815 0.01066285 0.008168308 0.0163111 0.5282355 0.3189231
## Test set     0.0629301656 0.06676816 0.062930166 3.8935118 3.8935118 2.4570432
##                     ACF1 Theil's U
## Training set -0.01608847        NA
## Test set      0.77445045  9.624916
```

#Forecast - ARIMA (1,1,0) ,ARIMA(1,1,1), ARIMA (2,1,2)

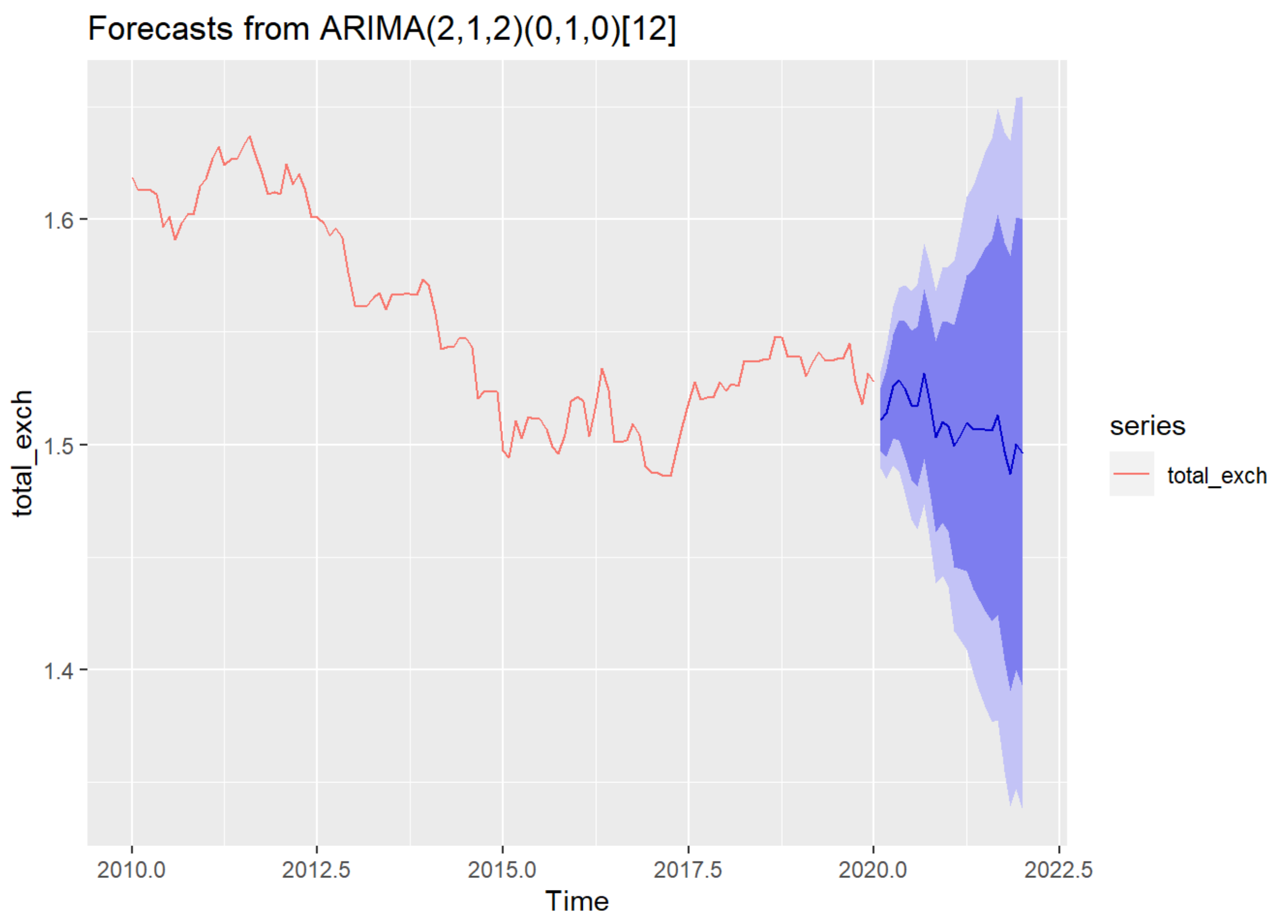
```
fittrain110 <- Arima(total_exch, order=c(1,1,0), seasonal=c(0,1,0))
fittrain110 %>% forecast(h=24) %>% autoplot()+ autolayer(total_exch)
```




```
fittrain111 <- Arima(total_exch, order=c(1,1,1), seasonal=c(0,1,0))
fittrain111 %>% forecast(h=24) %>% autoplot()+ autolayer(total_exch)
```




```
fittrain212 <- Arima(total_exch, order=c(2,1,2), seasonal=c(0,1,0))
fittrain212 %>% forecast(h=24) %>% autoplot()+ autolayer(total_exch)
```



6.2. Analyzing ACF, PACF order 1 to 5

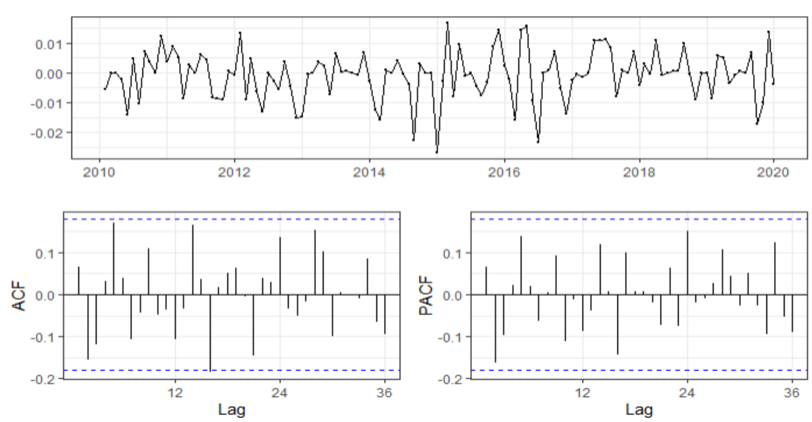


Figure 34 ACF and PACF order 1

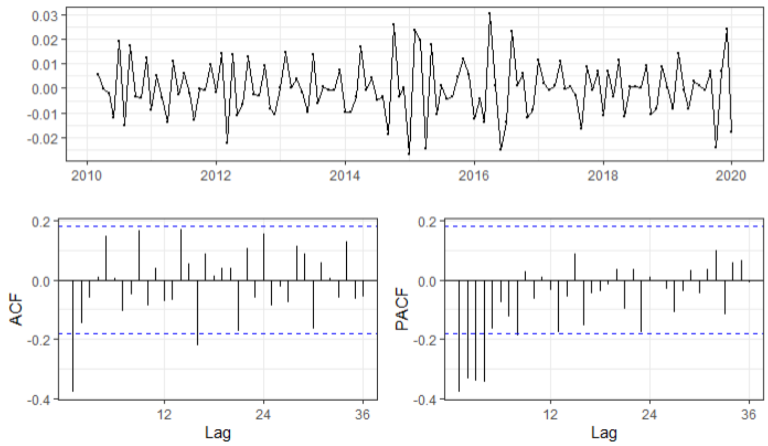
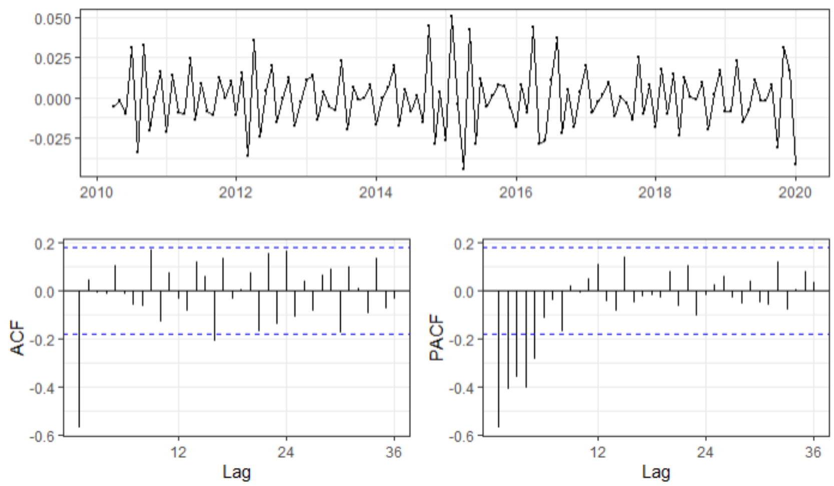


Figure 35 ACF and PACF order 2



ote

Figure 36 ACF and PACF order 3

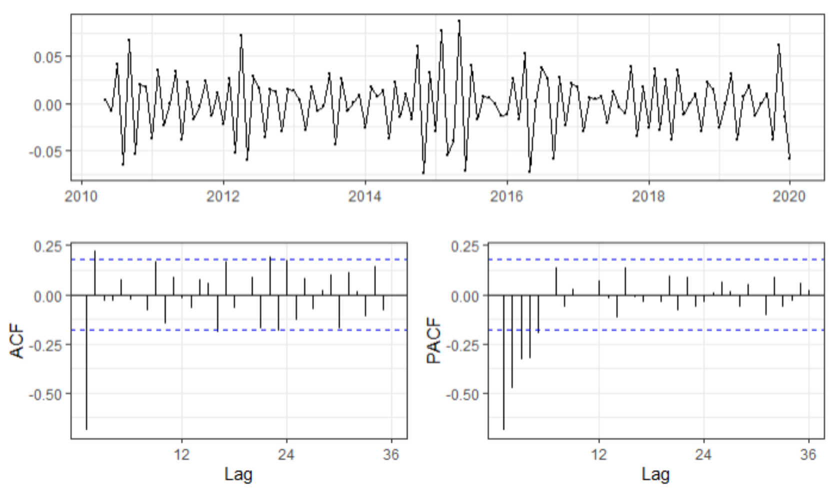


Figure 37 ACF and PACF order 4

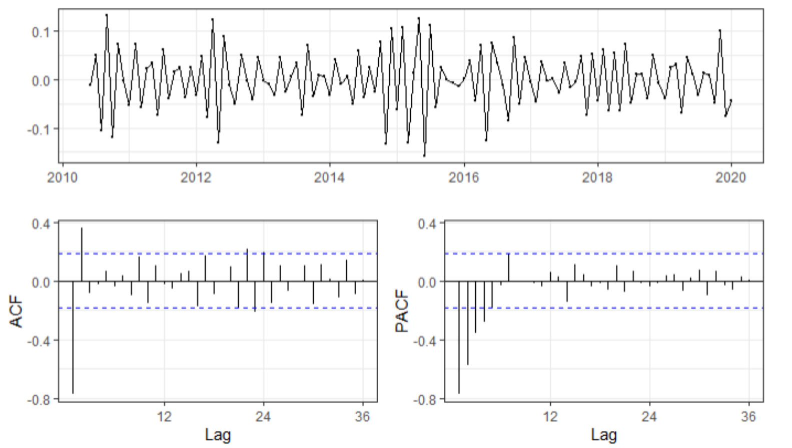
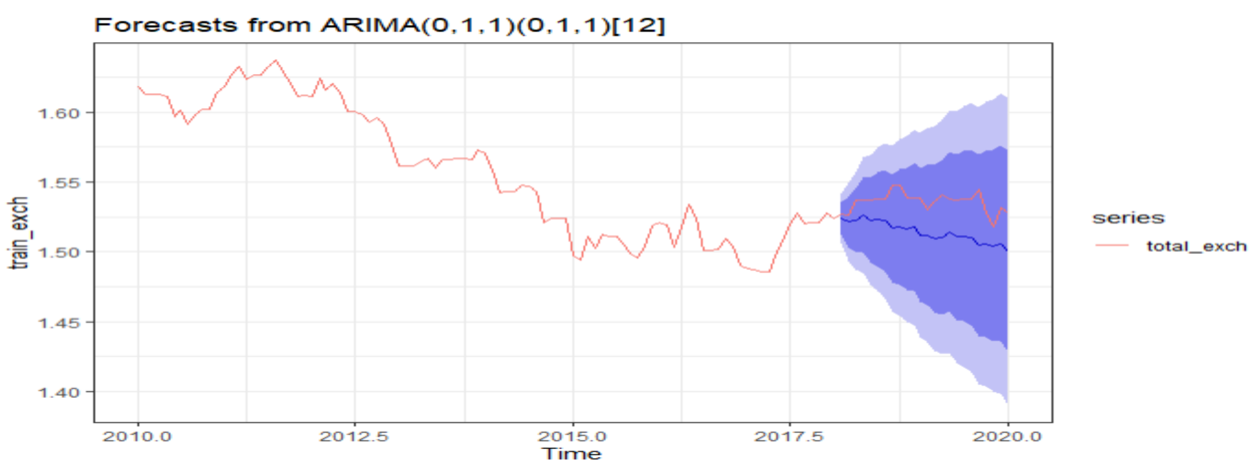
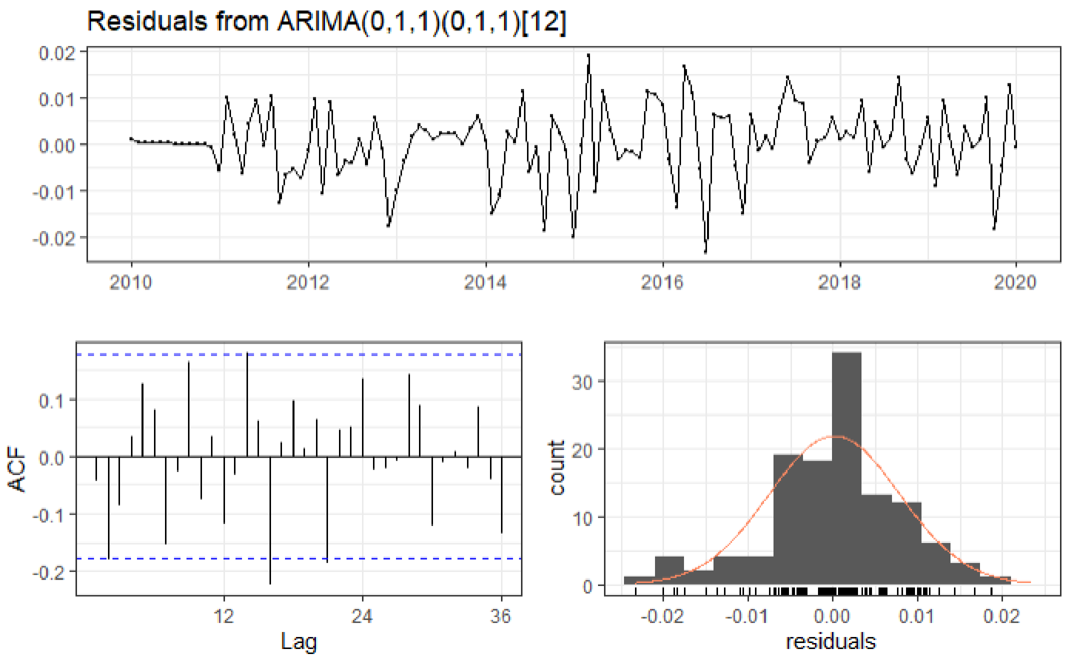


Figure 38 ACF and PACF order 5

6.3. ARIMA Forecast at different points

ARIMA (0,1,1)

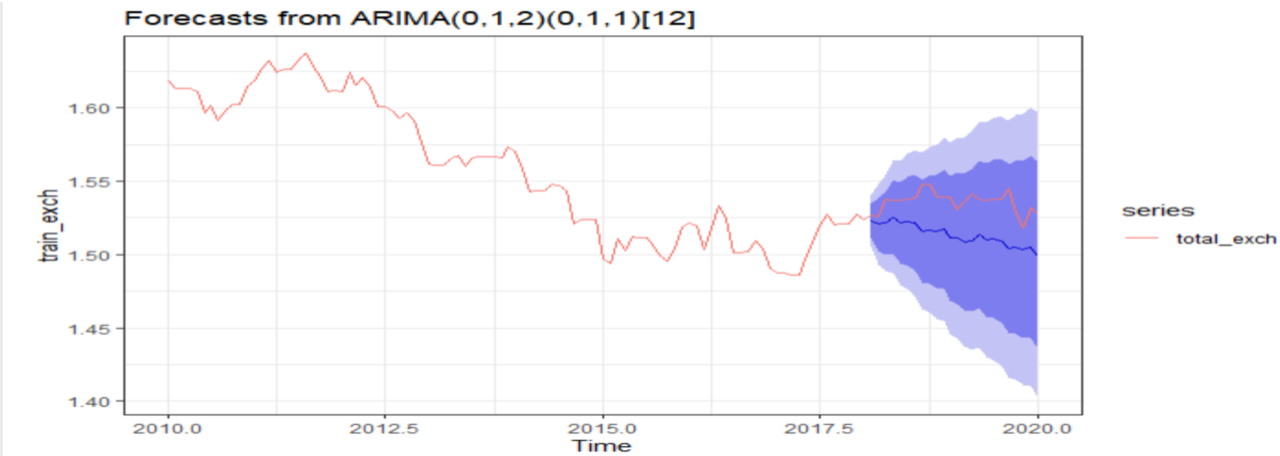


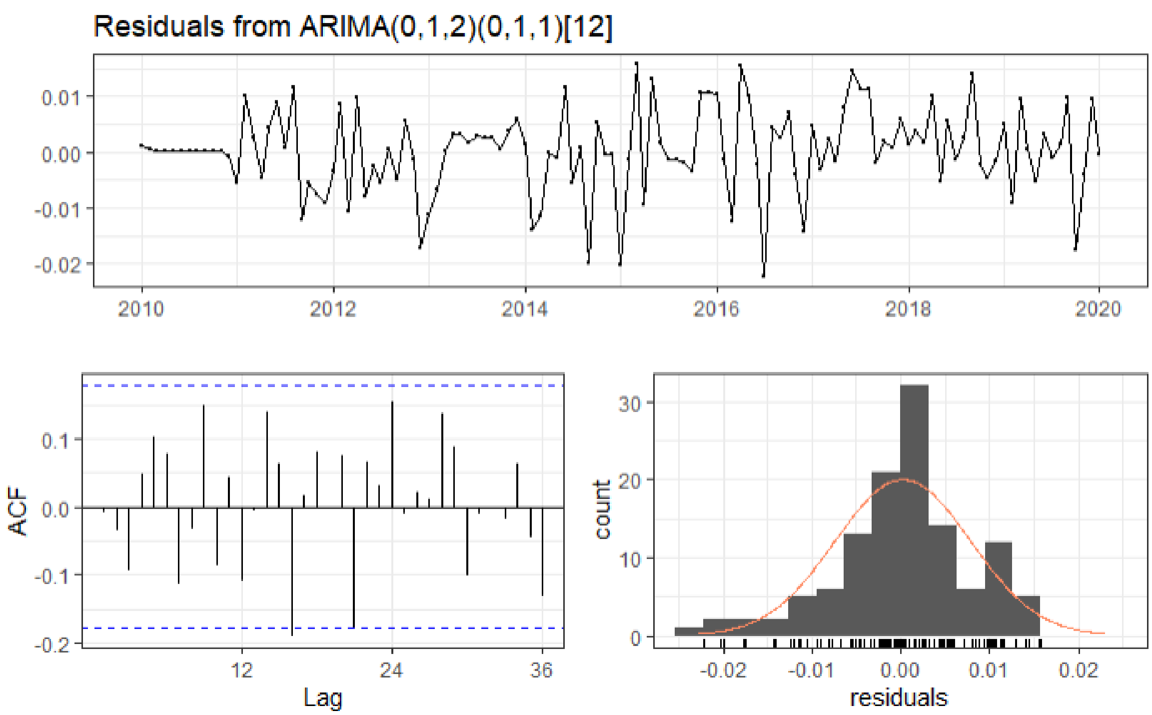


	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	3.726958e-05	0.007768719	0.005657429	0.002747153	0.3660141	0.2208885	-0.04018906	NA
Test set	2.158532e-02	0.023266372	0.021585325	1.403965562	1.4039656	0.8427767	0.50378536	3.442178

Figure 39 Accuracy ARIMA (0, 1, 1)

ARIMA (0,1,2)

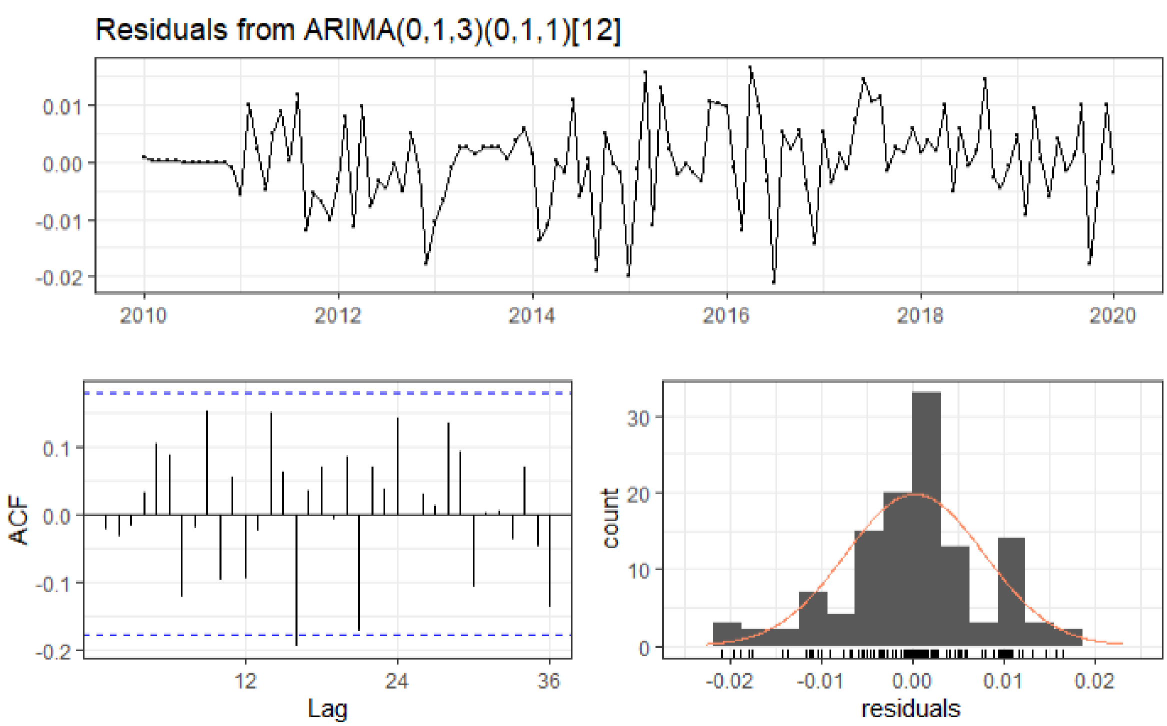




	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	2.370662e-05	0.007735696	0.005621277	0.001885501	0.3634349	0.2194769	-0.007225799	NA
Test set	2.228794e-02	0.023921961	0.022287936	1.449711829	1.4497118	0.8702094	0.504233165	3.538762

Figure 40 Accuracy ARIMA (0, 1, 2)

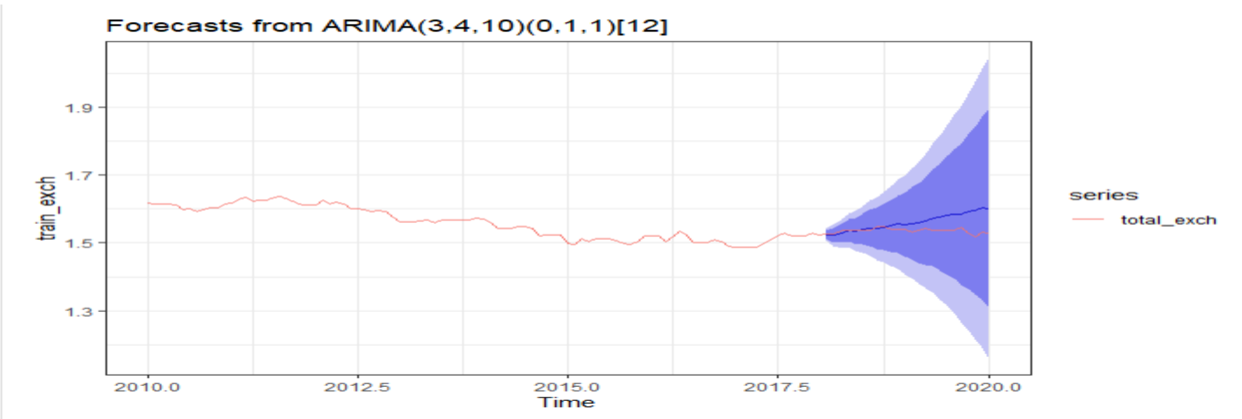
ARIMA (0,1,3)

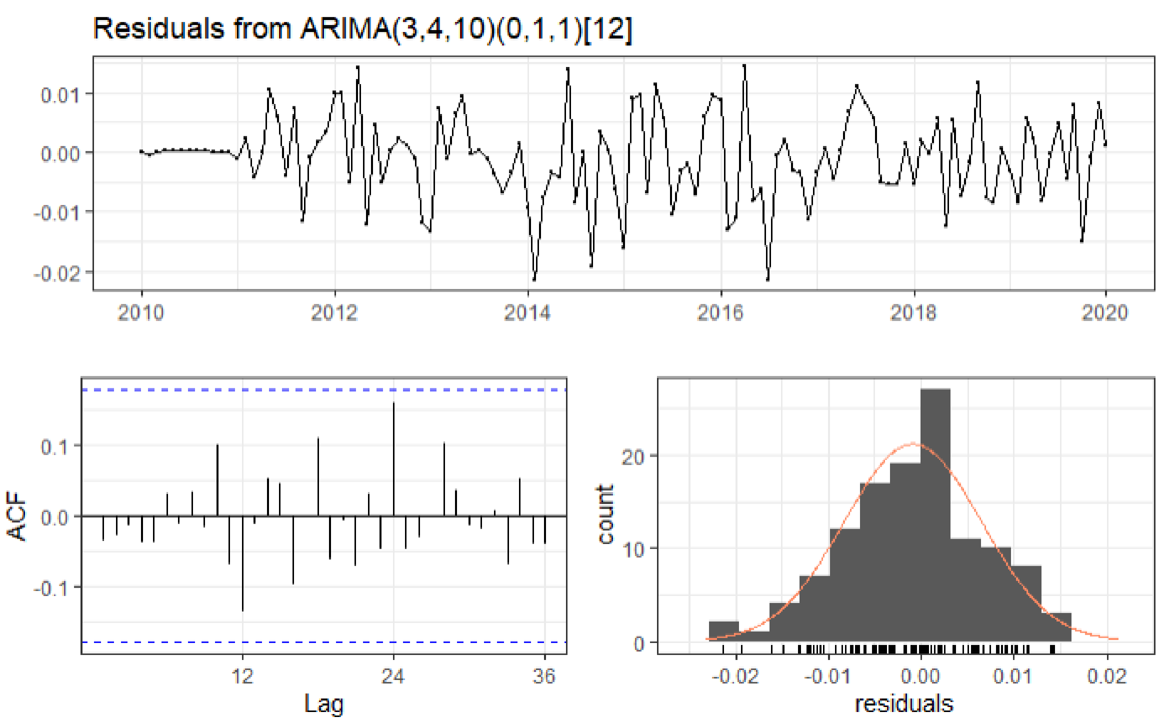



```
                       ME        RMSE         MAE         MPE      MAPE      MASE         ACF1 Theil's U
Training set 2.370662e-05 0.007735696 0.005621277 0.001885501 0.3634349 0.2194769 -0.007225799        NA
Test set     2.228794e-02 0.023921961 0.022287936 1.449711829 1.4497118 0.8702094  0.504233165  3.538762
```

Figure 41 Accuracy ARIMA (0, 1, 3)

ARIMA (3,4,10)

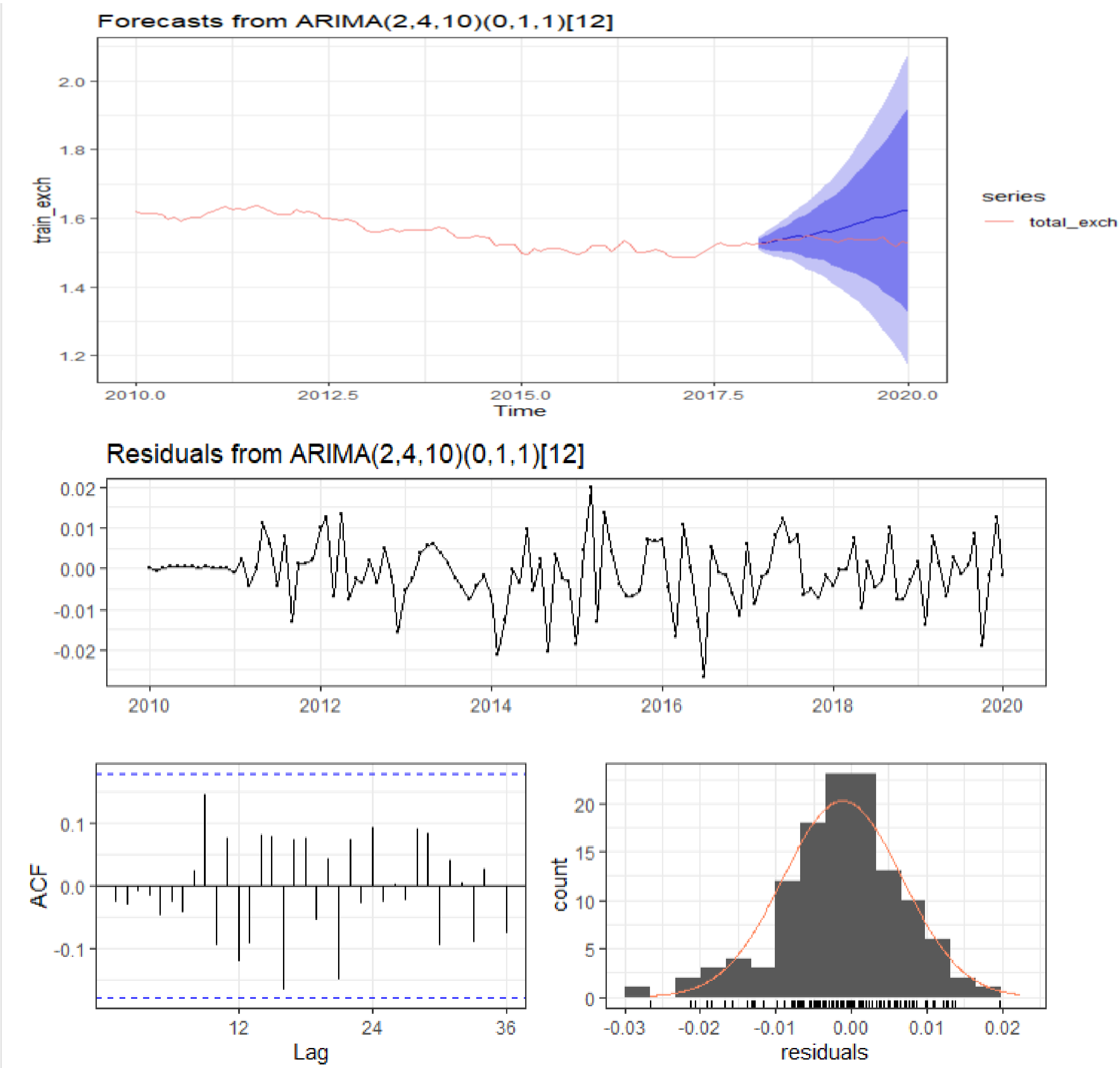




	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	2.370662e-05	0.007735696	0.005621277	0.001885501	0.3634349	0.2194769	-0.007225799	NA
Test set	2.228794e-02	0.023921961	0.022287936	1.449711829	1.4497118	0.8702094	0.504233165	3.538762

Figure 42 Accuracy ARIMA (3, 4, 10)

ARIMA (2,4,10)




```
                   ME        RMSE         MAE         MPE      MAPE      MASE         ACF1 Theil's U
Training set 2.370662e-05 0.007735696 0.005621277 0.001885501 0.3634349 0.2194769 -0.007225799        NA
Test set     2.228794e-02 0.023921961 0.022287936 1.449711829 1.4497118 0.8702094  0.504233165  3.538762
```

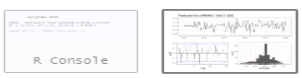
Figure 43Accuracy ARIMA (2, 4, 10)

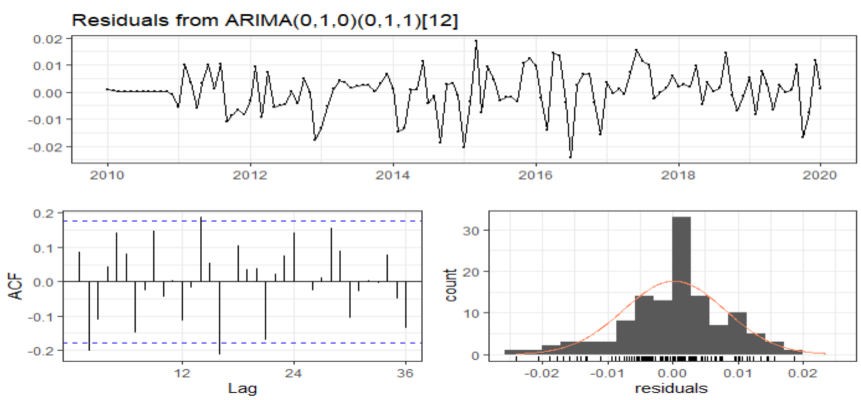
auto Arima : (0,1,0)

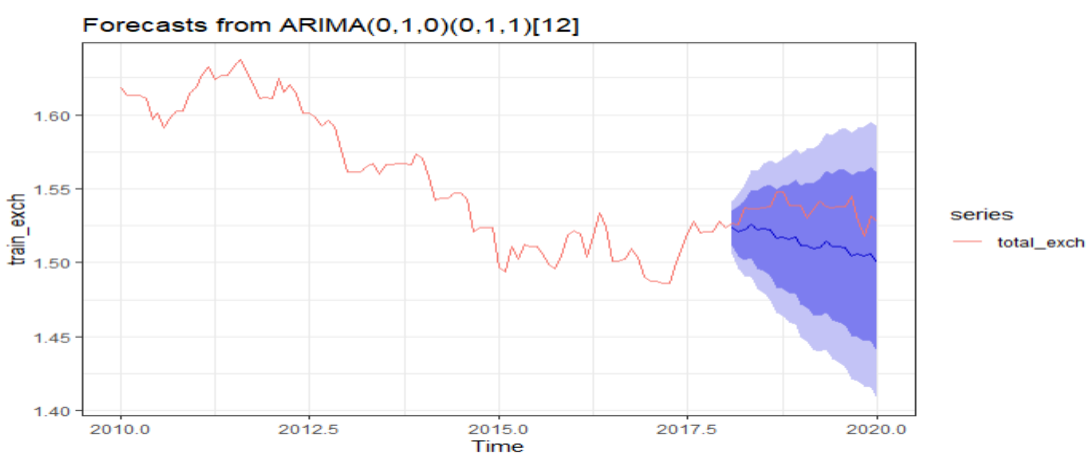
```
auto.arima(total_exch)
```

```
Series: total_exch
ARIMA(0,1,0)

sigma^2 estimated as 6.678e-05:  log likelihood=406.59
AIC=-811.19   AICc=-811.16   BIC=-808.4
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,0)(0,1,1)[12]
Q* = 41.374, df = 23, p-value = 0.01073

Model df: 1.   Total lags used: 24
```







```
                       ME        RMSE         MAE       MPE      MAPE      MASE      ACF1 Theil's U
Training set 2.114538e-05 0.007926751 0.005738952 0.0016676 0.3710922 0.2240714 0.1456806        NA
Test set     2.168609e-02 0.023345239 0.021686089 1.4105201 1.4105201 0.8467109 0.5015164  3.453689
```

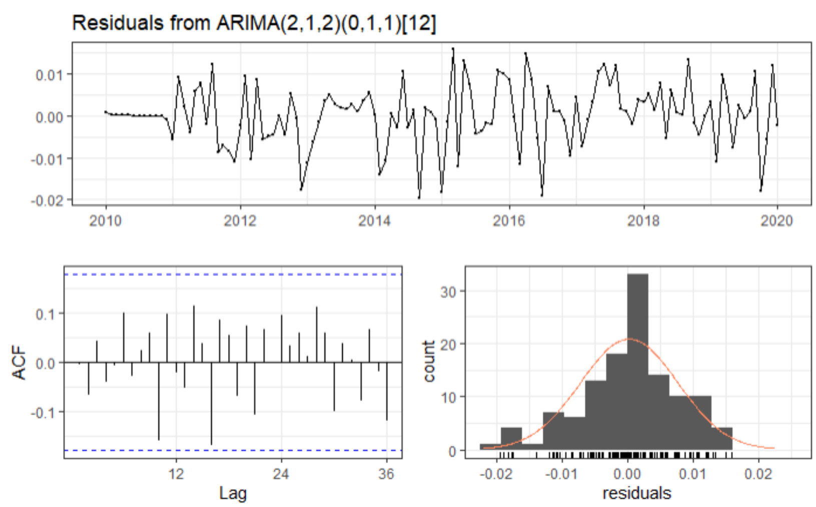
Figure 44 Accuracy auto ARIMA (0, 1, 0)

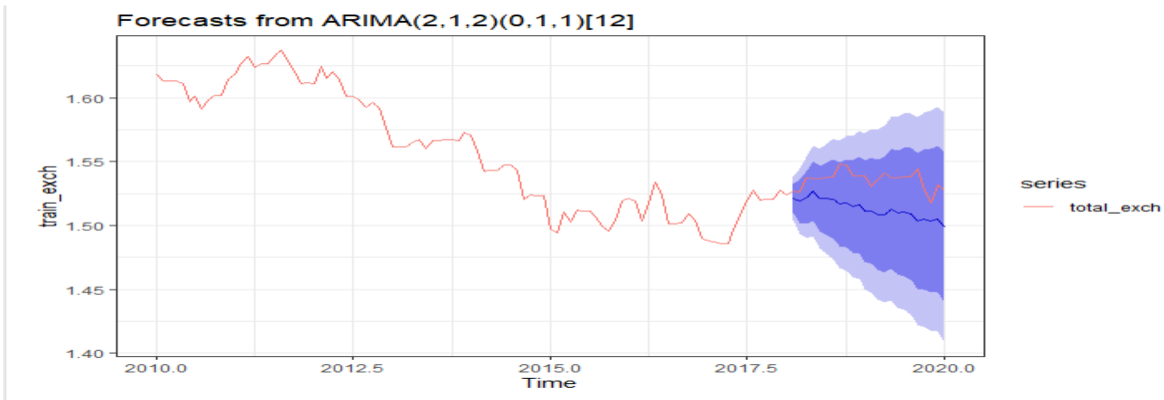
auto Arima : (2,1,2)

```{r}
auto.arima(total_exch , approximation = FALSE , stepwise = FALSE, seasonal = FALSE)
```

```
Series: total_exch
ARIMA(2,1,2)

Coefficients:
         ar1      ar2      ma1     ma2
      0.4039  -0.9022  -0.3288  0.7997
s.e.  0.1152   0.0770   0.1684  0.1107

sigma^2 estimated as 6.401e-05:  log likelihood=410.99
AIC=-811.99   AICc=-811.46   BIC=-798.05
```

```
        Ljung-Box test

data:  Residuals from ARIMA(2,1,2)(0,1,1)[12]
Q* = 20.998, df = 19, p-value = 0.3369

Model df: 5.   Total lags used: 24
```





	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	1.366702e-05	0.007398481	0.005492934	0.00136165	0.3549758	0.2144659	0.01834155	NA
Test set	2.266519e-02	0.024223920	0.022665189	1.47431740	1.4743174	0.8849388	0.50477282	3.58144

Figure 45 Accuracy ARIMA (2, 1, 2)