Forecasting Australia's Exchange rate using Trade-weighted Index

Time series Analysis Report

July 21

Sridevi Pamarthi - S3778317
Divya Ulaganathan - S3759465
(Group 50 – Trendspotters)

Contents

Table of Figures

Table of Tables

1. Introduction

The aim of this research is to find, whether the Australia's Exchange rate can withstand the Coronavirus pandemic, by performing time series analysis on its respective trade weighted index.

The Trade Weighted Index (TWI) is measure used to find the effective value of an exchange rate against a basket of currencies. It provides a broader measure of general trends in a currency. TWI is often used as one of the indicators of Australia's international competitiveness. It can measure whether a currency is appreciating or depreciating on average, relative to its trading partners. It is often used to gauge the value of the Australian dollar when bilateral exchange rates exhibit diverging trends.

Therefore, we use the TWI for forecasting Australia's exchange rate in our upcoming analysis and forecast the monthly TWI for next 10 months.

2. Methodology

To aid our analysis, we have extracted the monthly Trade Weighted Index (TWI) of Australian dollar data from Reserve Bank of Australia starting with January 2010 till April 2020.

In upcoming section, we will be applying a set of possible models for this time series data such as deterministic and stochastic trend models. We then choose the best model among them by applying model-building strategies. The possible values for the next 10 months will be forecasted based on it.

The analysis work has been carried out using the R studio with R Markdown template. The packages such as TSA, tseries, fUnitRoots, lmtest, FitAR, dplyr, readxl and forecast were used to produce all the steps and results shown in this report.

Initially, the given data frame is converted to time series object for further analysis. The following are time series and scatter plot of data to understand its distribution.

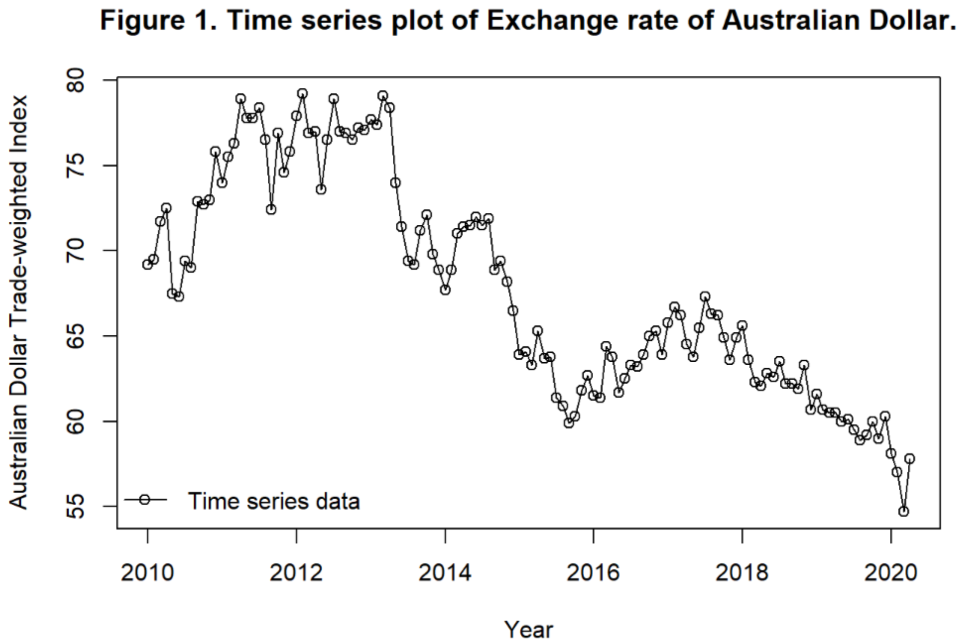

Figure 1. Time series plot of exchange rate of Australian dollar

As per the time series plot from Figure 1, we could observe the following characteristics:

- Trend: The trend line indicates downward trend, implying that Exchange rate of Australian Dollars has decreased over the years.
- Seasonality: There is no clear indication of seasonality. As there is no repeating pattern seen.
- Intervention: There seems to be an intervention around 2013 after which the trend decreased massively. Likewise, there is a slight intervention around year 2015 and 2018 during which the Exchange rate of Australian Dollars were high and decreased the following year.
- Changing variance: There is implications of changing variance throughout the series.
- There are few hanging observations which indicates high correlation.
- The series is non-stationary, so AR or MA behavior could not be commented clearly. Although we could identify autoregressive behavior of the series as the recovery after decreased TWI on year 2015 was gradual in 2016.

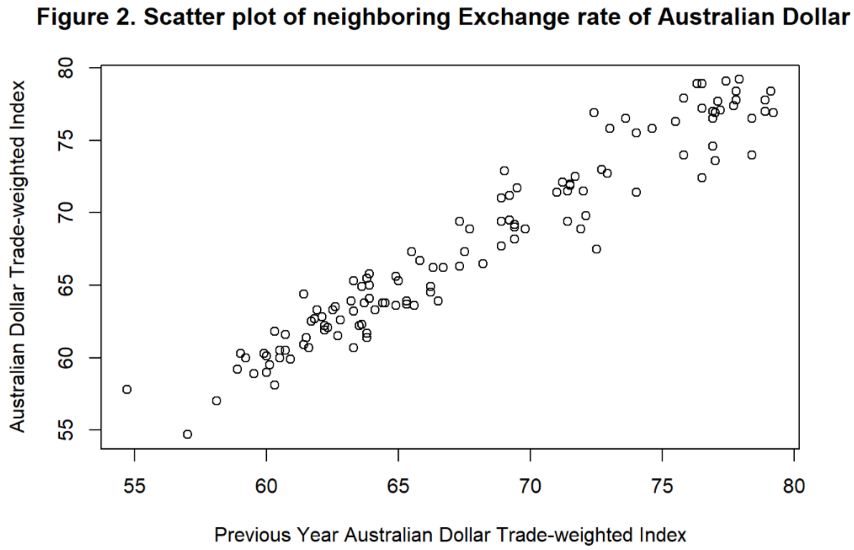

Figure 2. Scatter plot of neighboring Exchange rate of Australian dollar

The scatter plot from Figure 2, shows an upward trend and high correlation between neighboring Exchange rate of Australian Dollar values. The correlation calculated between neighboring Exchange rate of Australian Dollars is 0.9658673 which indicates high correlation.
This point out that forecasting the next month's Exchange rate of Australian Dollar value based on the previous month could provide a value close to actual value.

The given time series dataset has a small sample size with 124 elements. Due to the small sample size the deterministic trend analysis might not give proper result compared to stochastic trend analysis. Hence, we shall consider “exchange_rate_ts” time series as stochastic trend. In the upcoming section we apply the Cosine model and Quadratic model to the time series data even though it is not necessary, just to examine the summary statistics to see their respective performance on a small sample sized series.

Later, we shall perform stochastic trend analysis to identify all possible models and do diagnostic tests on best fitting model before predicting next 10 TWI exchange rates.

2.1 Exploring deterministic trend models:

Cosine trend model:

Cosine trend model has been applied to the data and observed that the p value is greater than 0.05 which indicates that the model is insignificant. Also, the R-squared value is negative**.**

Quadratic model:

Quadratic model has been applied to data and is plotted as below

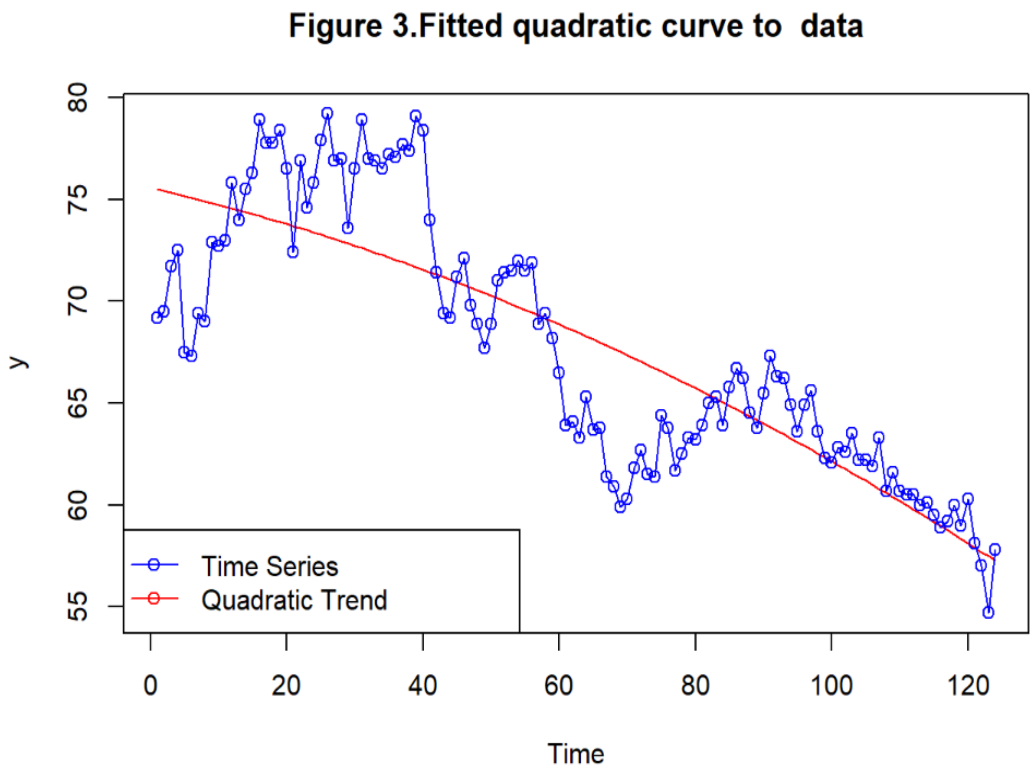

Figure 3. Fitted quadratic curve to data

From figure 3, even though quadratic model looks fits to data R-squared value is 0.70 which still has 30% variation in the time series.

2.2 Finding the best fitting ARIMA model

The following subsections consists of the steps to identify the best fitting ARIMA model for the given time series.

2.2.1 ACF and PACF Plots

Figure 4. ACF And PACf plots

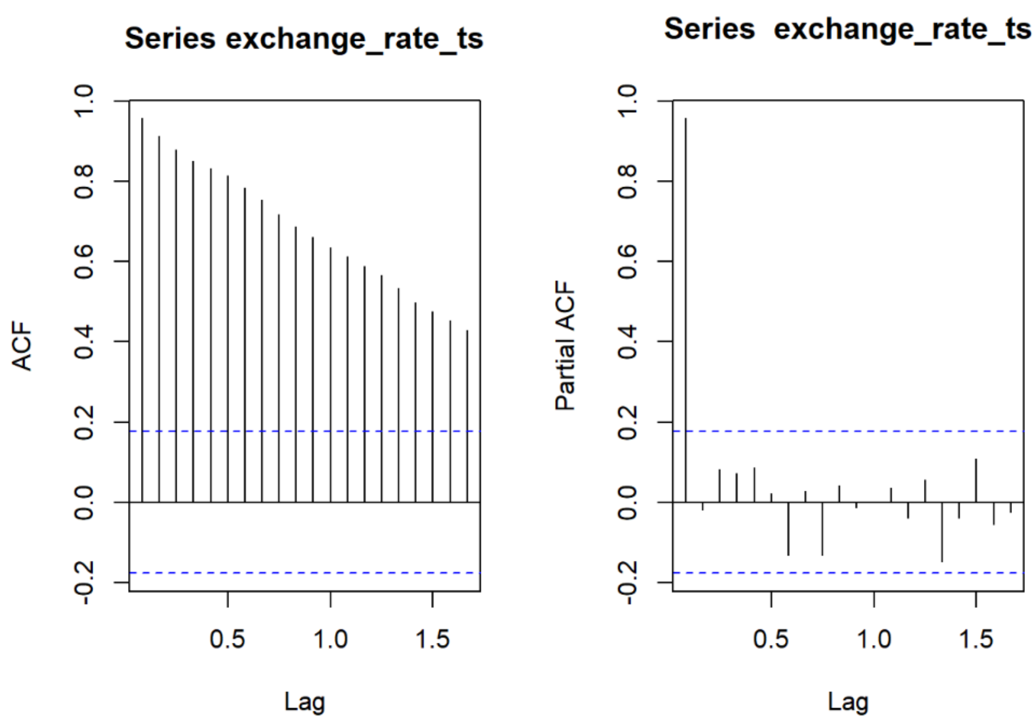

Figure 4. ACF and PACF plots

From the above **Figure 4**, we can observe the PACF has a cut off on after 1st lag which means this is mostly an AR(1) process. But we can also observe a slowly decaying pattern of the ACF and PACF have high correlation. This indicates that there is presence of trend and non-stationarity in the exchange_rate_ts series.

In order to confirm the existence of trend and non-stationarity in the series, we perform Augmented Dickey-Fuller Test.

2.1.1.1 Performing ADF test on original series

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 0
##   STATISTIC:
##     Dickey-Fuller: -0.7111
##   P VALUE:
##     0.387
##
## Description:
##  Sun Jun 07 01:29:16 2020 by user: Sridevi
```

The p-value generated from the ADF test is greater than 0.05. Hence, we fail to reject null hypothesis H0. Thus, the exchange_rate_ts series is non stationary and it has to be made stationary in order to identify models.

In order to, overcome the non-stationary natures of a series, various transformations like Box-Cox transformation and log transformation can be applied to the series. In some cases, the series can be converted to normal form just after transformation. In order to check that, we use Shapiro-Wilk normality test and Q-Q plot for normality. If the series tends to not be normal from test and q-q plot, we apply differentiation.

2.2.2 Box-Cox transformation

In order to calculate lambda for transforming the data, box-cox transformation was applied to "exchange_rate_ts". But resultant graph didn't have a smooth curve. Hence, we have applied Box-Cox transformation using yule-walker method.

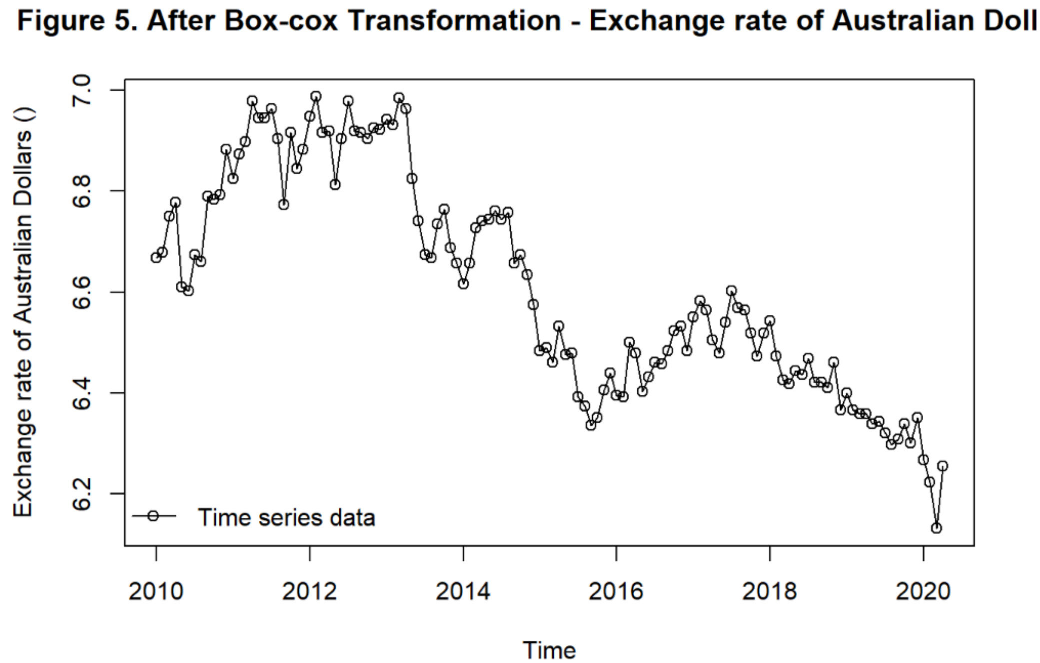

Figure 5. After Box-Cox transformation – exchange rate of Australian Dollar

The lamda value obtained from Box cox transformation, is 0.45. Hence, we apply the Box-cox transformation to "exchange_rate_ts" using lamda 0.45. From figure 5, Applying Box-Cox transformation doesn't detrend series much. However, the variance has decreased.

2.2.3 Q-Q plot:

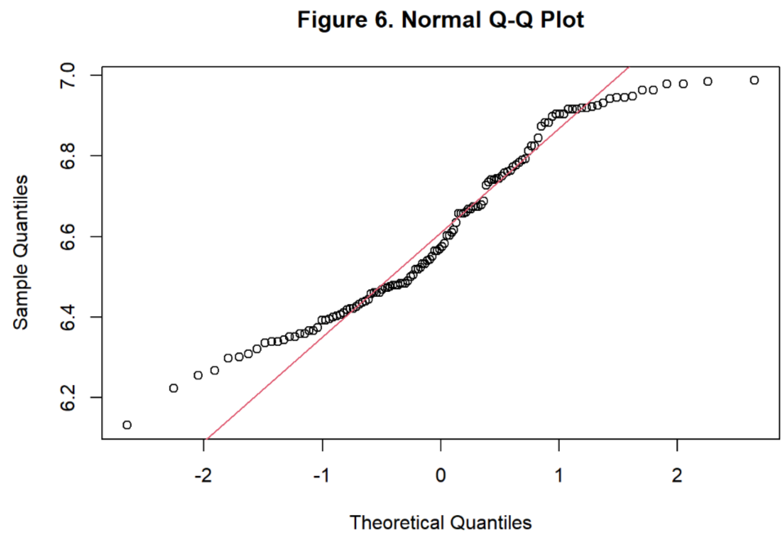

Figure 6. Normal Q-Q plot

From Figure 6, some points are quite far away from the normal line. The series does not seem to be normal. Let's verify with Shapiro-Wilk normality test.

2.2.4 Sharpino-wilk test:

```
## ## Shapiro-Wilk normality test

## ## data: exchange_rate_bc

## W = 0.9505, p-value = 0.0001815
```

Shapiro-Wilk normality test confirms that the series is not normal. As the series is not stationary, differentiation has to be performed.

2.2.5 Differencing

We apply first differentiation on box cox transformed data and then plotted the values after first differentiation to check for existence of trend and apply adfTest for non-stationarity check identified.

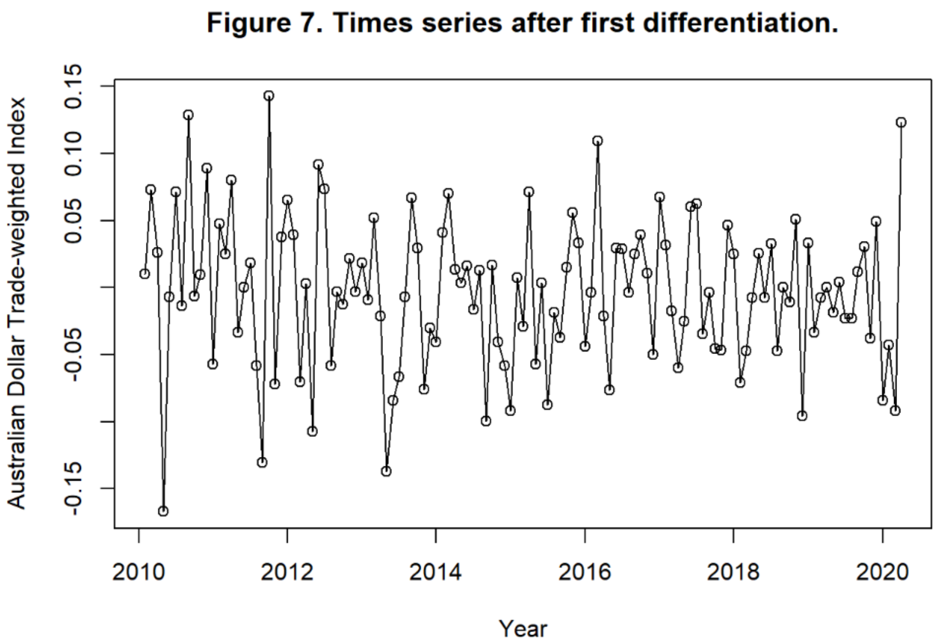

Figure 7. Time series after first differentiation

Figure 7 shows that first differencing detrended the series and. After performing adfTest, we can confirm that the series is stationary based on p -value which is less than 0.05.

Conducting ADF test on differenced series using **adfTest**() function to check if the differenced series is stationary.

```
##
## Title:
## Augmented Dickey-Fuller Test
##
## Test Results:
## PARAMETER:
## Lag Order: 5
## STATISTIC:
## Dickey-Fuller: -4.1147
## P VALUE:
## 0.01
##
## Description:
## Sun Jun 07 02:01:56 2020 by user: Sridevi
```

After performing adfTest, we confirm that the series is stationary based on p -value which is less than 0.05**.**

2.2.5.1 Normality check for Differenced series:

As First differentiated series satisfied stationarity, we check it for normality. As seen before the series is already normal. This test is performed, just to confirm the differentiated object is also normal. From figure 8. q-q plot the data points appears to be normal except for few points. Shapiro-Wilk normality test confirms the same.

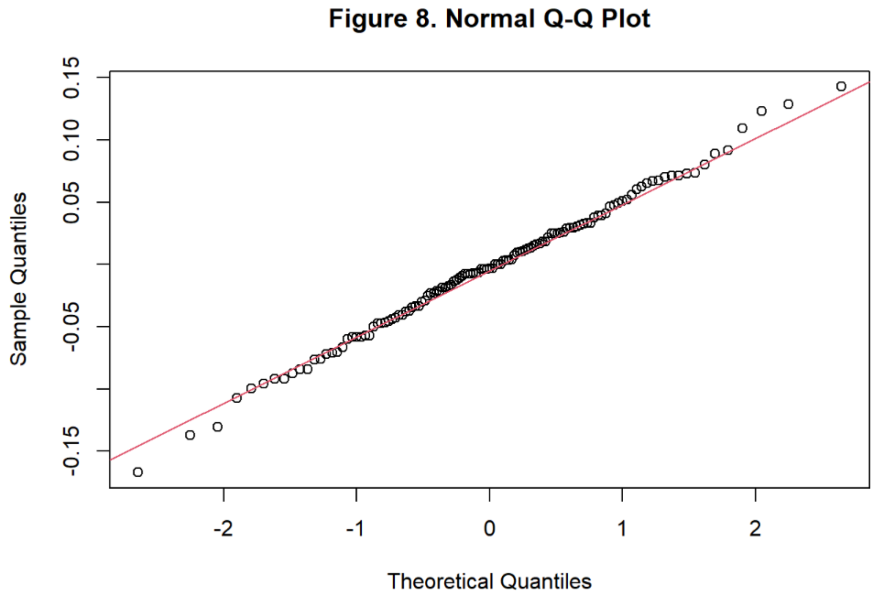

Figure 8. Normal Q-Q plot

2.1.5.2 Shapiro-Wilk normality test for Differenced series:

```
##
## Shapiro-Wilk normality test
##
## data: exchange_rate_diff
## W = 0.99595, p-value = 0.9806
```

From Figure 8, The series seems to be normal. Shapiro-Wilk normality test confirms the same.

2.2.6 ACF and PACF of Differenced series:

Figure 9.ACF and PACF plots of differenced series

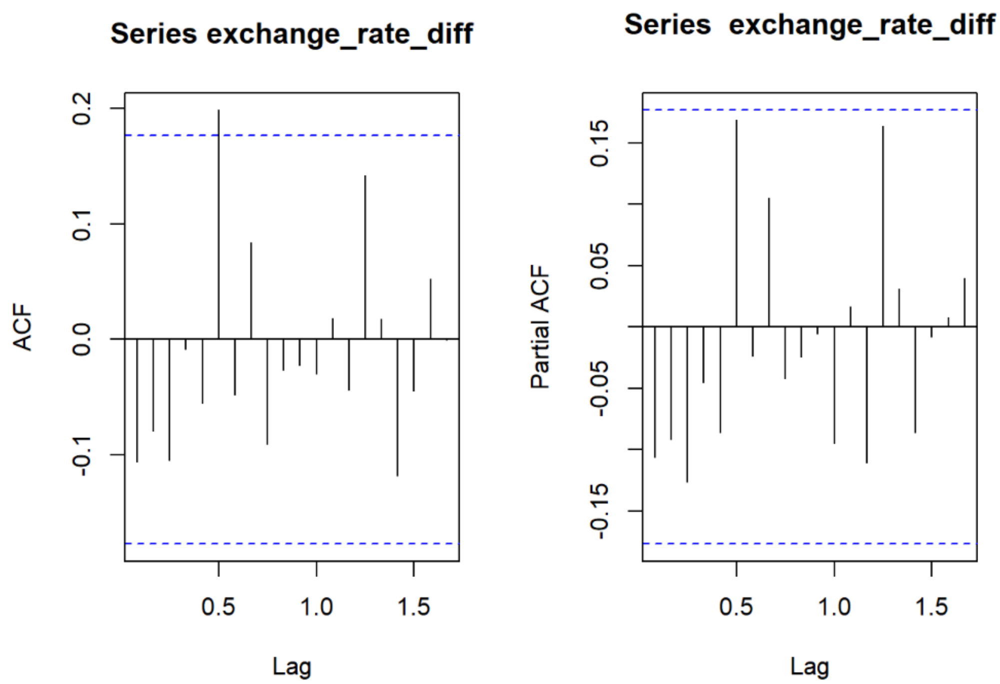

Figure 9. ACF and PACF plots of differenced series

From figure 9, we couldn't find any interesting trend. On ACF, lag 6 has maximum significance. Meanwhile, PACF doesn't have a maximum significant lag. Hence, ACF has 1 significant lag and PACF has 0 significant lags.

Therefore, the possible ARIMA(p, d, q) models from ACF and PACF are ARIMA(0,1,1) d = 1 due to first differentiation.

2.2.7 EACF (Extended ACF)

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o o x o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x o x o o o o o o o o  o  o  o
## 5 x x x x x o o o o o o  o  o  o
## 6 o x o x x o o o o o o  o  o  o
## 7 x x x x o x o o o o o  o  o  o
```

According to EACF, we set the vertex at (1,0). After choosing the vertex we choose the points by taking the value of vertex, one point to the right and one point below the vertex. The possible ARIMA(p, d, q) models are ARIMA(1,1,1), ARIMA(1,1,2) and ARIMA(2,1,2).

2.2.8 BIC (Bayesian Information Criterion)

From the below BIC table Figure 10, The test-lags are considered as AR and error-lags are considered as MA. Test-lag6 has the maximum value. Hence we can get AR(6). The Maximum value of error-lag is error-lag3. Hence we can get MA(3). Then the position where max text-lag meets the error-lag (top position) constitutes to ARIMA(6,1,3). The possible models are AR(6), MA(3) and ARIMA(6,1,3).

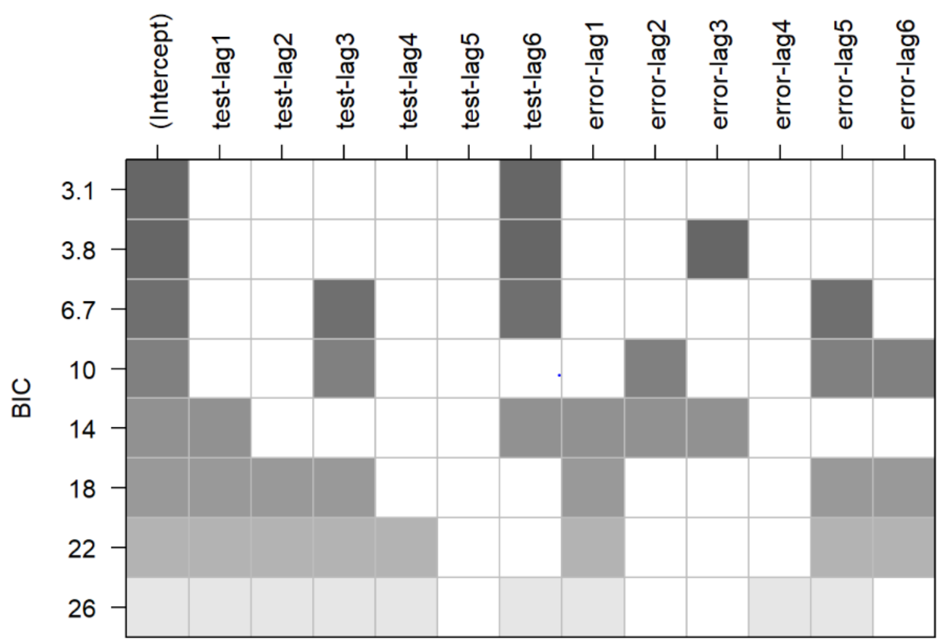

Figure 10. Bayesian Information Criterion

The set with possible candidate models is {ARIMA(0,1,1) ,ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(2,1,2),ARIMA(6,1,3)}

2.3 Model Selection

Following results have taken by fitting the candidate models to series using 'CSS' and 'ML' methods:

Table 1. ARIMA models results

Model	Method	Result
ARIMA(0,1,1)	CSS	MA(1) component is insignificant
ARIMA(0,1,1)	ML	MA(1) component is insignificant
ARIMA(1,1,1)	CSS	AR(1) component is insignificant. MA(1) component is significant
ARIMA(1,1,1)	ML	AR(1) component is insignificant. MA(1) component is significant
ARIMA(1,1,2)	CSS	AR(1),MA(2) components are insignificant
ARIMA(1,1,2)	ML	AR(1),MA(2) components are insignificant
ARIMA(2,1,2)	CSS	AR(2),MA(2) components are significant
ARIMA(2,1,2)	ML	AR(2),MA(2) components are significant
ARIMA(6,1,3)	CSS	AR(6),MA(3) components are insignificant
ARIMA(6,1,3)	ML	AR(6),MA(3) components are insignificant

From above output, we select ARIMA(2,1,2) model as a best fit model for this series as it has both AR and MA components as significant.

2.4 Overfitting models:

As we have chosen ARIMA(2,1,2) as our best model we further check for overfitting. The overfitting models derived from ARIMA(2,1,2) are ARIMA(3,1,2) and ARIMA(2,1,3). These models were verified using CSS and ML method and results are as follows.

ARIMA (3,1,2) results for both CSS and ML method:

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.3441798  0.0079596  43.2406 < 2.2e-16 ***
## ar2 -0.8207154  0.0234520 -34.9955 < 2.2e-16 ***
## ma1 -0.5358801  0.0861018  -6.2238 4.853e-10 ***
## ma2  1.1488294  0.0344175  33.3792 < 2.2e-16 ***
## ma3 -0.2034726  0.0870765  -2.3367   0.01945 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -1.605229   0.082672 -19.4168  < 2e-16 ***
## ar2 -0.749009   0.080035  -9.3585  < 2e-16 ***
## ma1  1.583047   0.130039  12.1736  < 2e-16 ***
## ma2  0.552442   0.221493   2.4942  0.01263 *
## ma3 -0.244848   0.119372  -2.0511  0.04025 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA (2,1,3) results for both CSS and ML method:

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.298663         NA      NA        NA
## ar2 -0.590851   0.104185 -5.6712 1.418e-08 ***
## ar3 -0.103820   0.089623 -1.1584    0.2467
## ma1  0.178644         NA      NA        NA
## ma2  0.559514         NA      NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -0.53101    0.37551 -1.4141   0.1573
## ar2  0.14054    0.37103  0.3788   0.7048
## ar3 -0.15297    0.10261 -1.4908   0.1360
## ma1  0.42096    0.37392  1.1258   0.2603
## ma2 -0.29518    0.31644 -0.9328   0.3509
```

2.5 Residual Analysis On ARIMA(2,1,2)

2.2.4.1 ACF of Residuals

The Residual analysis is performed on chosen model model_212_ml

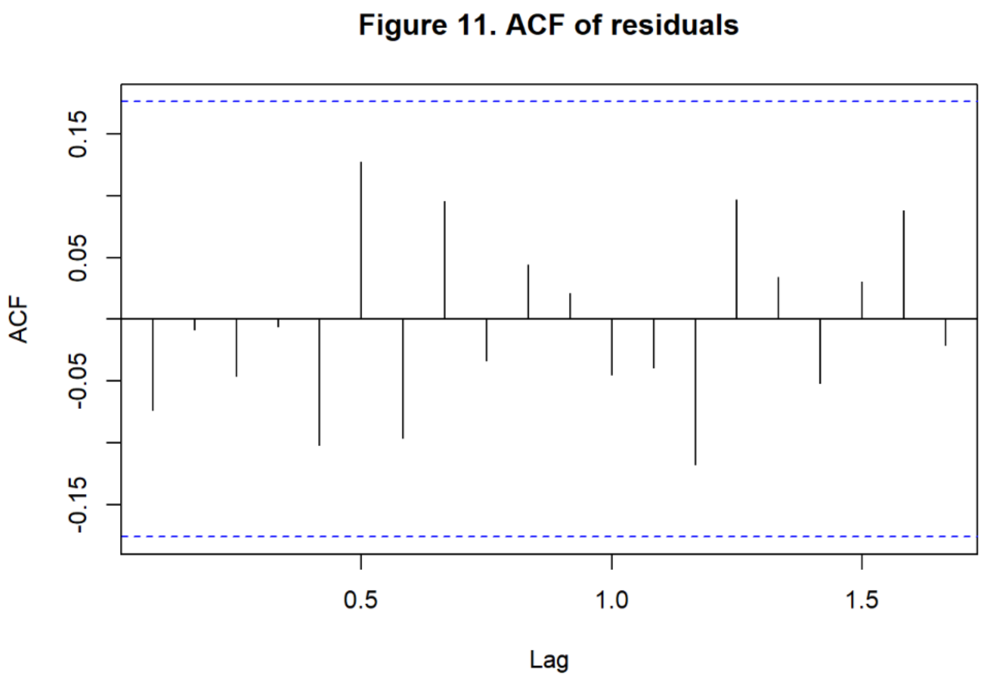

Figure 11. ACF of residuals

From figure 11, there are no significant lags in acf plot of residuals of model_212_ml.

2.5.1 EACF of Residuals

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 o x o x o o o o o o o  o  o  o
## 5 x o o x x o o o o o o  o  o  o
## 6 x x o o o x o o o o o  o  o  o
## 7 x o x o o o o o o o o  o  o  o
```

From EACF, we can still consider ARIMA(2,1,2) as a valid model from above EACF as the vertex can be set at (2,1).

2.5.2 McLeod-Li test on model Residuals

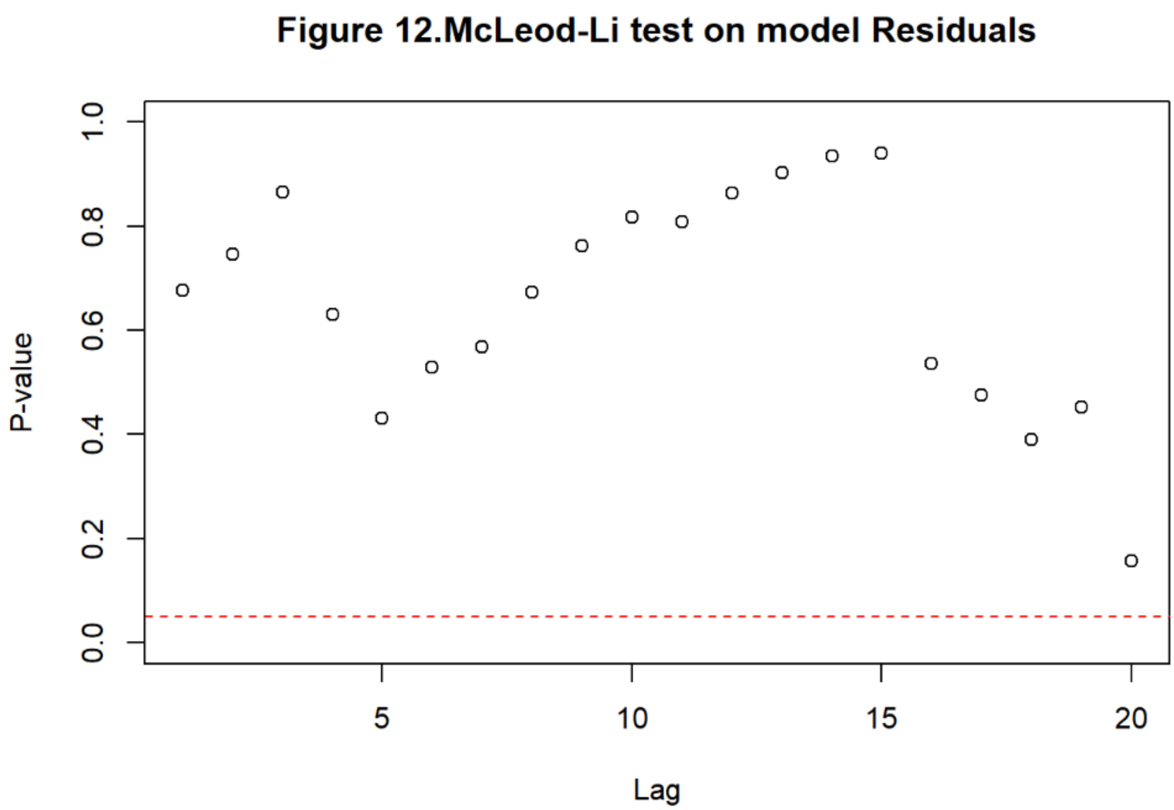

Figure 12. McLeod-Li test on model Residuals

The McLeod-Li test (shown in Figure 12) for standardized model residuals also confirm that the ARIMA(2,1,2) model fits the residual series well.

2.6 Diagnostic tests for ARIMA(2,1,2)

Following diagnostic tests have carried out for the best model ARIMA(2,1,2) to check whether the model fits the data.

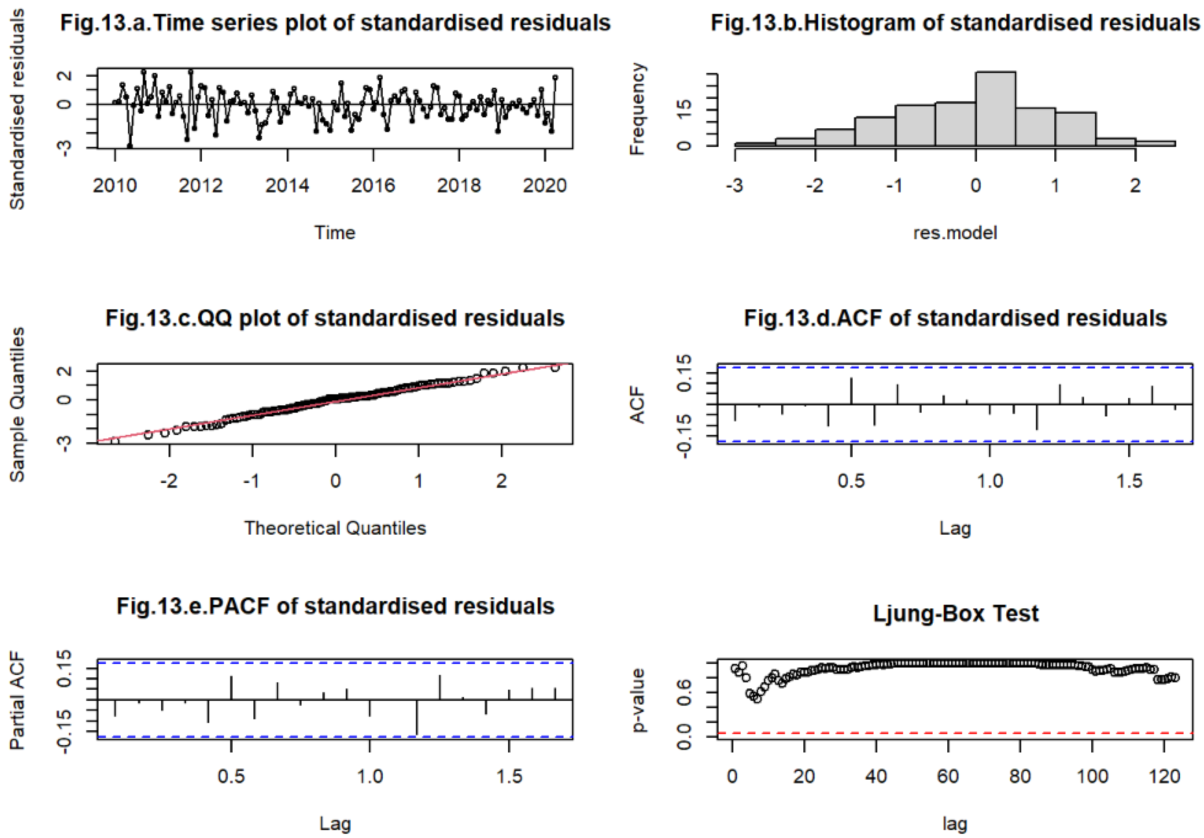

Figure 13. Diagnostic tests for ARIMA(2,1,2)

## ## Shapiro-Wilk normality test ## ## data: res.model ## W = 0.99077, p-value = 0.5809	## ## Box-Ljung test ## ## data: residuals(model_212_ml) ## X-squared = 4.5039, df = 6, p-value = 0.6088

The following observations have been identified from the above **Figure 13** and above tests:

- Time series (Fig.13.a) of data is symmetrically distributed around axis. The series is stationary.
- Histogram (Fig.13.b) is normally distributed.
- ACF(Fig.13.d) do not have any significant lag.
- PACF(Fig.13.e) do not have any significant lag.
- The patterns in ACF and PACF indicates existence of white noise behavior.
- The qq-plot(Fig.13.c) shows normal distribution. And Shapiro-Wilk test confirms it.
- All p-values are greater than 0.05 in the Ljung-Box test plot (Fig.13.f), so this confirms that ARIMA(2,1,2) model successfully deals with serial correlation.
- As the p-value from Shapiro-Wilk normality test for the ARIMA(2,1,2) model is more than 0.05 , hence the data is normally distributed.

Therefore, residual analysis also confirms ARIMA(2,1,2) model as a best fit model. Thus, we use this model for forecasting the Exchange rate of Australian Dollars for next 10 months

3. Forecasting:

Using forecast library, we use predict function to forecast next 10 TWI values. The forecasted values are as follows.

Table 2. Forecasting results for the next 10 months

	Point Forecast	Lo 95	Hi 95
May-20	57.77514	55.0526	60.60435
Jun-20	58.13657	54.34109	62.14122
Jul-20	58.22389	53.42227	63.36473
Aug-20	57.92902	52.28111	64.05483
Sep-20	57.80099	51.54233	64.65338
Oct-20	58.03022	51.24685	65.5136
Nov-20	58.18223	50.84901	66.33811
Dec-20	58.01261	50.14963	66.83233
Jan-21	57.85171	49.54737	67.23398
Feb-21	57.96769	49.26545	67.85838

Forecasted results with the ARIMA(2,12) are fitted in the plot below

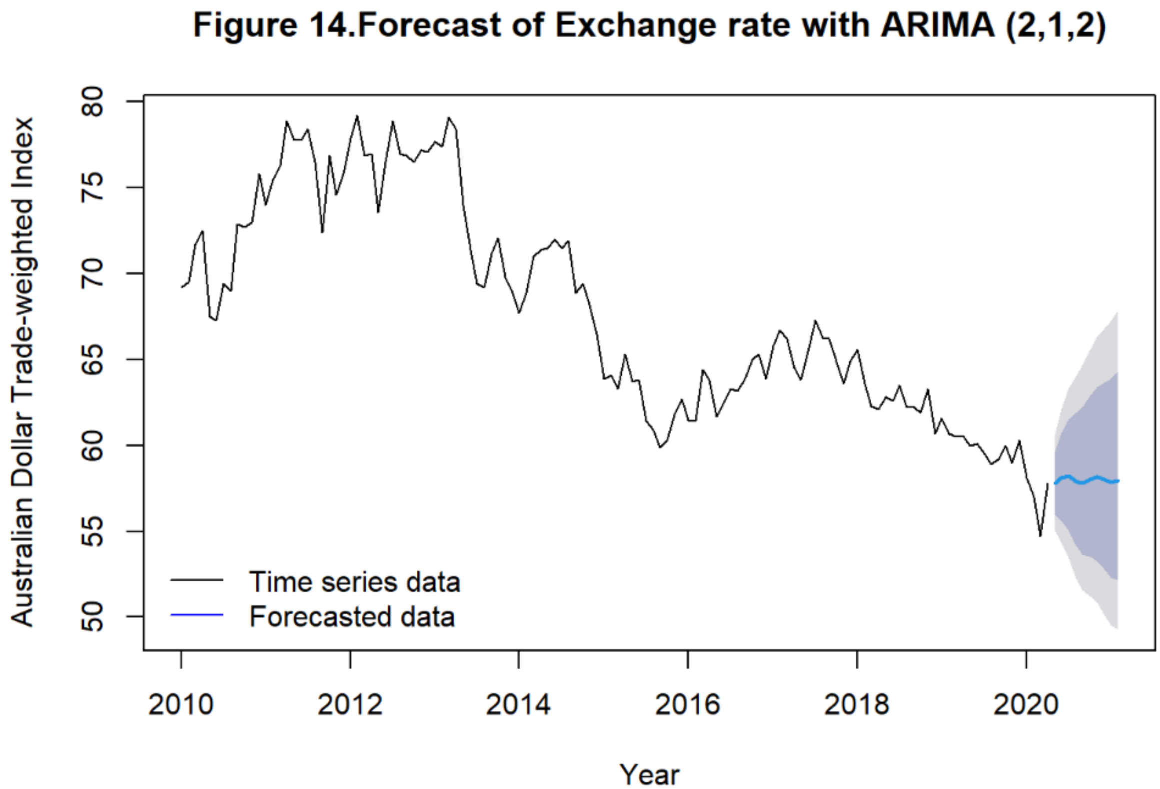

Figure 14. Forecast of Exchange rate with ARIMA(2,1,2)

4. Conclusion:

From the Table 2 in section 3 Forecasting, the fitted forecasted values for next 10 months are 57.759, 58.126, 58.226, 57.927, 57.788, 58.022, 58.182, 58.009, 57.840 and 57.960.

Although the trade weighted index dropped during the first quarter of 2020 due to the pandemic, the recent developments in April indicates a moderate forecast for next 10 months. Hence, we can conclude that the Australia's Exchange rate can withstand the Coronavirus pandemic if the existing trade measures were in practice.

5. Summary

This report is aimed at finding the best model and providing the forecasted results of next 10 months, for the Australian exchange rate. The time series of data (Figure 1), exhibits the stochastic trend and so the ARIMA models have applied to find the best fitting model and then carried out the diagnostic testing to check if the model fits the data appropriately.

From Table1 and sections 2.3 and 2.6, ARIMA(2,1,2) appears as the best fitting model to this time series, out of all the candidate models and the results of the next ten months are forecasted accordingly. From the forecast results (Figure 14), it appears that the Australian exchange rate will likely be same in next 10 months.

6. References

[1]. Data Source: Historical Data. (2020). Retrieved 8 June 2020, from https://www.rba.gov.au/statistics/historical-data.html

[2]. Exchange Rates – Parliament of Australia. (2020). Retrieved 8 June 2020, from https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/MSB/feature/Exchangerates

[3]. MATH1318 Time Series Analysis notes prepared by A/Prof Zeynep Kalaylioglu

7. Abbreviations

ARIMA	Autoregressive moving average model
ACF	Autocorrelation function
PACF	Partial autocorrelation function
BIC	Bayesian Information Criterion

8. Appendix:

```
#Loading the required packages
library(readxl)
# data manipulation
```

```
library(dplyr)
# time series package
library(TSA)
library(tseries)

library(fUnitRoots)
library(lmtest)
library(FitAR)
library(forecast)

# Data import

exchange_rate <- read_excel("Exchange_Rate.xlsx",
    col_types = c("date", "numeric"))
print("head values of exchange_rate:")
head(exchange_rate)

print("Initial class of exchange_rate:")
class(exchange_rate)

# Converting dataframe object to time series object
exchange_rate_ts <- ts(as.vector(exchange_rate['trade_weighted_index']), frequency=12, start=c(20
10,1))

print("Class of Dataframe after conversion:")
class(exchange_rate_ts)

print("head values of exchange_rate:")
head(exchange_rate_ts)

# Time series plot of data vs time
plot(exchange_rate_ts,type='o',xlab='Year',ylab='Australian Dollar Trade-weighted Index',main= "F
igure 1. Time series plot of Exchange rate of Australian Dollar.")
legend("bottomleft", lty=1,pch =1,bty ="n", col=c("black"), text.width =15,
       c("Time series data"))

# Scatter plot neighboring Exchange rate of Australian Dollars
```

```
plot(y=exchange_rate_ts,x=zlag(exchange_rate_ts), xlab='Previous Year Australian Dollar Trade-wei
ghted Index', ylab='Australian Dollar Trade-weighted Index', main = "Figure 2. Scatter plot of ne
ighboring Exchange rate of Australian Dollars")


# Calculating correlation between neighboring Exchange rate of Australian Dollars

y = exchange_rate_ts

x = zlag(exchange_rate_ts)

index = 2:length(x)

print("correlation between neighboring Exchange rate of Australian Dollars:")

print(cor(y[index],x[index]))


#Harmonic Model

har.=harmonic(exchange_rate_ts,1)


cosine_trend=lm(exchange_rate_ts~har.)

summary(cosine_trend)


#Quadratic trend model


t = time(exchange_rate_ts)

t2 = t^2

quadratic_model = lm(exchange_rate_ts~ t + t2)

summary(quadratic_model)


##Fitted quadratic curve to data

plot(ts(fitted(quadratic_model)),ylim=c(min(c(fitted(quadratic_model),

     as.vector(exchange_rate_ts))),max(c(fitted(quadratic_model),

                                         as.vector(exchange_rate_ts)))),

     ylab='y' ,col="red", main = "Figure 3.Fitted quadratic curve to  data")

     lines(as.vector(exchange_rate_ts),type="o",col="blue")

     legend ("bottomleft", lty = 1, pch = 1,col = c("blue","red"), text.width = 45,

     c("Time Series","Quadratic Trend "))


#Seasonal trend model

exchange_rate_seasonal <- ts(as.vector(exchange_rate['trade_weighted_index']), start=c(2010,1), e
nd= c(2020,4), frequency=12)

head(exchange_rate_seasonal)
```

```
plot(exchange_rate_seasonal, type='o', ylab= 'exchange_rate')

month.=season(exchange_rate_seasonal)
seasonal_model=lm(exchange_rate_seasonal~month.-1) # -1 removes the intercept term
summary(seasonal_model)

plot(ts(fitted(seasonal_model)),ylim=c(min(c(fitted(seasonal_model),
      as.vector(exchange_rate_seasonal))),
      max(c(fitted(seasonal_model),as.vector(exchange_rate_seasonal)))),
     ylab='exchange_rate' ,col="red",
     main = "Seasonal model ")
     lines(as.vector(exchange_rate_seasonal),type="o",col="blue")
     legend ("bottomleft", lty = 0.5, pch = 0.5,col = c("blue","red"), text.width =50,
     c("Time Series","seasonal model "))

# checking for stationarity

# Initial ACF plot
par(mfrow=c(1,2))
acf(exchange_rate_ts)
pacf(exchange_rate_ts)

# Initial PACF  plot
pacf(exchange_rate_ts, main="Figure.4.Series exchange_rate_ts ")
ar(diff(exchange_rate_ts))
adfTest(exchange_rate_ts, lags=0, title=NULL, description=NULL)

#Box cox transformation : lambda graph
#exchange_rate_transform = BoxCox.ar(exchange_rate_ts)
exchange_rate_transform = BoxCox.ar(exchange_rate_ts, method = "yule-walker")

#Range value from Box cox transformation
print("Range value from Box cox transformation")
exchange_rate_transform$ci

lambda = 0.2
# Box-cox transformation
```

```
exchange_rate_bc = ((exchange_rate_ts^lambda)-1)/lambda
plot(exchange_rate_bc, type = "o", ylab = "Exchange rate of Australian Dollars ()",
 main = "Figure 5. After Box-cox Transformation - Exchange rate of Australian Dollars")
legend("bottomleft", lty=1,pch =1,bty ="n", col=c("black"), text.width =15,
       c("Time series data"))

#Checking For normality
qqnorm(exchange_rate_bc,main= "Figure 6. Normal Q-Q Plot" )
qqline(exchange_rate_bc, col = 2)
shapiro.test(exchange_rate_bc)

#First differentiation
exchange_rate_diff = diff(exchange_rate_bc,differences=1)
par(mfrow=c(1,1))
#plotting exchange_rate_diff
plot(exchange_rate_diff,type='o',xlab='Year',ylab='Australian Dollar Trade-weighted Index',main=
"Figure 7. Times series after first differentiation.")

#ADF test
order = ar(diff(exchange_rate_diff))$order
adfTest(exchange_rate_diff, lags = order,  title = NULL,description = NULL)

#Checking For normality
qqnorm(exchange_rate_diff,main= "Figure 8. Normal Q-Q Plot" )
qqline(exchange_rate_diff, col = 2)
shapiro.test(exchange_rate_diff)

#plot acf and pacf of time series
#win.graph(width=4.875, height=2.5,pointsize=8)
par(mfrow=c(1,2))
AutoCorrelation <- acf(exchange_rate_diff, plot = FALSE)
PartialAutoCorrelation <- pacf(exchange_rate_diff, plot = FALSE)
#plot(AutoCorrelation, main = "Autocorrelation Plot\nof Changes in Exchange Rate")
#plot(PartialAutoCorrelation, main = "Partial Autocorrelation\nof Changes in Exchange Rate")

# ACF- PACF
#par(mfrow=c(1,2))
```

```
acf(exchange_rate_diff)
pacf(exchange_rate_diff)

#EACF
eacf(exchange_rate_diff)

bic_table = armasubsets(y=exchange_rate_diff, nar=6, nma=6,y.name='test',ar.method='ols')
plot(bic_table)



# Model fitting
model_011_css = arima(exchange_rate_bc,order=c(0,1,1),method='CSS')
coeftest(model_011_css)

# Model fitting
model_011_ml = arima(exchange_rate_bc,order=c(0,1,1),method='ML')
coeftest(model_011_ml)

# Model fitting
model_111_css = arima(exchange_rate_bc,order=c(1,1,1),method='CSS')
coeftest(model_111_css)

# Model fitting
model_111_ml = arima(exchange_rate_bc,order=c(1,1,1),method='ML')
coeftest(model_111_ml)

model_112_css = arima(exchange_rate_bc,order=c(1,1,2),method='CSS')
coeftest(model_112_css)

model_112_ml = arima(exchange_rate_bc,order=c(1,1,2),method='ML')
coeftest(model_112_ml)

model_212_css = arima(exchange_rate_bc,order=c(2,1,2),method='CSS')
coeftest(model_212_css)

model_212_ml = arima(exchange_rate_bc,order=c(2,1,2),method='ML')
coeftest(model_212_ml)
```

```
model_613_css = arima(exchange_rate_bc,order=c(6,1,3),method='CSS')
coeftest(model_613_css)

model_613_ml = arima(exchange_rate_bc,order=c(6,1,3),method='ML')
coeftest(model_613_ml)

sort.score <- function(x, score = c("bic", "aic")){
  if (score == "aic"){
    x[with(x, order(AIC)),]
  } else if (score == "bic") {
    x[with(x, order(BIC)),]
  } else {
    warning('score = "x" only accepts valid arguments ("aic","bic")')
  }
}

# AIC and BIC values
#{ARIMA(0,1,1) ,ARIMA(1,1,1), ARIMA(1,1,2), ARIMA(2,1,2),ARIMA(6,1,3)}
sort.score(AIC(model_011_ml,model_111_ml,model_112_ml,model_212_ml,model_613_ml),
           score ="aic")

#sort.score(BIC(model_212_ml), score = "bic")
sort.score(BIC(model_011_ml,model_111_ml,model_112_ml,model_212_ml,model_613_ml),
           score ="bic")

model_213_css = arima(exchange_rate_bc,order=c(2,1,3),method='CSS')
coeftest(model_213_css)

model_213_ml = arima(exchange_rate_bc,order=c(2,1,3),method='ML')
coeftest(model_213_ml)

model_312_css = arima(exchange_rate_bc,order=c(3,1,2),method='CSS')
coeftest(model_312_css)

model_312_ml = arima(exchange_rate_bc,order=c(3,1,2),method='ML')
coeftest(model_312_ml)
```

```
model_212_css = arima(exchange_rate_bc,order=c(2,1,2),method='CSS')
coeftest(model_212_css)

model_212_ml = arima(exchange_rate_bc,order=c(2,1,2),method='ML')
coeftest(model_212_ml)

sort.score(AIC(model_011_ml,model_111_ml,model_112_ml,model_212_ml,model_613_ml), score = "aic")
sort.score(BIC(model_011_ml,model_111_ml,model_112_ml,model_212_ml,model_613_ml), score = "bic" )
par(mfrow=c(1,1))
acf(model_212_ml$residuals,main="Figure 11. ACF of residuals")
eacf(model_212_ml$residuals)
McLeod.Li.test(y=model_212_ml$residuals, main="Figure 12.McLeod-Li test on model Residuals")

# The following function provides a handy way of displaying the diagnostic plots.
# You need to install FitAR package to run this function.
residual.analysis <- function(model, std = TRUE){
  library(TSA)
  library(FitAR)
  if (std == TRUE){
    res.model = rstandard(model)
  }else{
    res.model = residuals(model)
  }
  par(mfrow=c(3,2))
  plot(res.model,type='o',ylab='Standardised residuals', main="Fig.13.a.Time series plot of stand
ardised residuals")
  abline(h=0)
  hist(res.model,main="Fig.13.b.Histogram of standardised residuals")
  qqnorm(res.model,main="Fig.13.c.QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  acf(res.model,main="Fig.13.d.ACF of standardised residuals")
  pacf(res.model,main="Fig.13.e.PACF of standardised residuals")
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = length(model$residuals)-1 , StartLag = k + 1, k = 0, SquaredQ = FA
LSE)
  par(mfrow=c(1,1))
```

```
}
residual.analysis(model = model_212_ml)
Box.test(residuals(model_212_ml), lag=6, type="Ljung-Box")


fit = Arima(exchange_rate_ts,c(2,1,2), lambda = 0.2)


exchange_rate_forecast <- forecast(fit,h=10, level=95)
exchange_rate_forecast
plot(forecast(fit,h=10),xlab="Year",ylab="Australian Dollar Trade-weighted Index",main="Figure 14
.Forecast of Exchange rate with ARIMA (2,1,2)")
legend("bottomleft", lty=1,bty ="n", col=c("black","blue"), text.width =15,
       c("Time series data","Forecasted data"))


tab2 <- "Parameter Estimation
----------------------------------------------------------------------
| Model          | Method  | Result                                   |
|----------------|:-------:|-----------------------------------------:|
| ARIMA(0,1,1)   | CSS     | MA(1) component is insignificant         |
| ARIMA(0,1,1)   | ML      | MA(1) component is insignificant         |
| ARIMA(1,1,1)   | CSS     | MA(1) component is significant           |
| ARIMA(1,1,1)   | ML      | MA(1) component is significant           |
| ARIMA(1,1,2)   | CSS     | AR(1),MA(2) components are insignificant |
| ARIMA(1,1,2)   | ML      | AR(1),MA(2) components are insignificant |
| ARIMA(2,1,2)   | CSS     | AR(2),MA(2) components are significant   |
| ARIMA(2,1,2)   | ML      | AR(2),MA(2) components are significant   |
| ARIMA(6,1,3)   | CSS     | AR(6),MA(3) components are insignificant |
| ARIMA(6,1,3)   | ML      | AR(6),MA(3) components are insignificant |
"
```

```
cat(tab2)
```