Modeling Foreign Exchange Rate INR/USD of India

Assignment 2 : Time Series

Anurima Dey, Roll : DST-19/20-005

Foreign Exchange Rate: Time Series Analysis.

What is Exchange Rate?

In *Finance*, exchange rate is the rate at which one currency is exchanged with another, which is equivalently saying that the value of one currency in comparison with other currency. Exchange rate is an important variable as it's value influences the decision of the traders, investors, importers, exporters, bankers, financiers, policy makers of a nation.

Forecasting the Exchange rate is very important both at a individual level for the practitioners and Researchers and at ministry level for policy making. It is the avenue to understand the economic conditions of a country and taking measures to improve different factors to produce the maximum financial returns and profit. For all certain reasons, forecasting the *exchange rates* is also quiet a challenging task across the globe.

In this report I shall be utilizing the time series concepts to do an analysis and predict the daily exchange rates of the *Indian Rupee (INR) against the United States Dollar (USD).* I shall do the analysis comparatively on recent data i.e. *Daily exchange rates from January 2010 to December 2019.*

A little history of *Foreign Exchange Rate*:
The foreign exchange market in India is believed to have begun in 1978 when the government allowed banks to trade foreign exchange with each other. Today, it is almost unnecessary to reiterate the observation that globalization and liberalization have significantly enhanced the scope for the foreign exchange market in India. The Indian exchange rate is regime, as noted by Goyal (2018) is a managed float, where the central bank allows markets to discover the equilibrium level but only intervenes to prevent excessive volatility.

Research Objectives: To Find out the perfect model for the Foreign Exchange Rate. I shall do so using the conventional method followed by a ACF, PACF analysis.

Information on dataset.

Frequency : Monthly
Unit : INR/USD
Source : Organisation for Economic Co-operation and Development @CEIC.
Series ID : 279672802
SR Code : SR5084711.
First Obs. Date : 01-1957
Last Obs. Date : 05-2020

Step 1 : Loading the Data, Plotting the data, and Conclusion

```
FER.data=readxl::read_xlsx("INForeignExchangeRateMonthlyAverageINR_USD.xlsx")
head(FER.data)
```

```
## # A tibble: 6 x 2
##   Month               FER_INR_USD
##   <dttm>                    <dbl>
## 1 1957-01-01 00:00:00        4.78
## 2 1957-02-01 00:00:00        4.78
## 3 1957-03-01 00:00:00        4.79
## 4 1957-04-01 00:00:00        4.80
## 5 1957-05-01 00:00:00        4.80
## 6 1957-06-01 00:00:00        4.80
```

```
tail(FER.data)
```

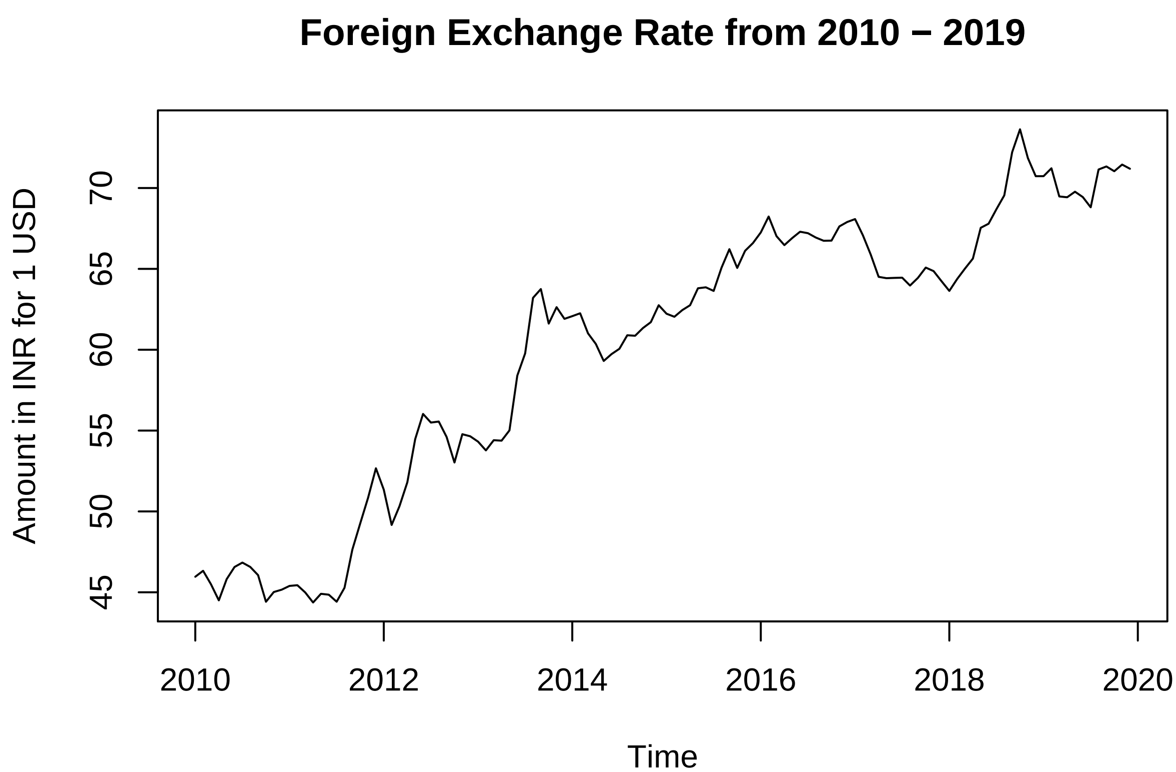
```
## # A tibble: 6 x 2
##   Month               FER_INR_USD
##   <dttm>                    <dbl>
## 1 2019-12-01 00:00:00        71.2
## 2 2020-01-01 00:00:00        71.3
## 3 2020-02-01 00:00:00        71.5
## 4 2020-03-01 00:00:00        74.4
## 5 2020-04-01 00:00:00        76.2
## 6 2020-05-01 00:00:00        75.7
```

```
# Monthly Seasonality (frequency set to 12 for monthly data)
FER <- ts(FER.data$FER_INR_USD, start = c(1957, 01),
          end = c(2019,12),frequency=12)
FER.window <- window(FER, start= c(2010))
FER.window
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2010 45.95980 46.32790 45.49650 44.49950 45.81150 46.56455 46.83640 46.56710
## 2011 45.39450 45.43680 44.98950 44.36810 44.90480 44.85180 44.41590 45.27850
## 2012 51.34610 49.16350 50.32300 51.80290 54.47350 56.03020 55.49480 55.55980
## 2013 54.31680 53.77370 54.40460 54.37570 55.01080 58.39730 59.77540 63.20884
## 2014 62.07600 62.25400 61.01400 60.35662 59.30500 59.73070 60.05860 60.89520
## 2015 62.22593 62.03761 62.44984 62.75316 63.80033 63.86074 63.63498 65.07233
## 2016 67.25233 68.23767 67.02185 66.46953 66.90673 67.29686 67.20762 66.93964
## 2017 68.08037 67.07545 65.87666 64.50709 64.42484 64.44299 64.45588 63.96838
## 2018 63.63690 64.37384 65.02133 65.63635 67.53936 67.79307 68.69336 69.54651
## 2019 70.73287 71.22177 69.47865 69.42741 69.77312 69.43886 68.80830 71.14570
##           Sep      Oct      Nov      Dec
## 2010 46.05900 44.41190 45.01650 45.15910
## 2011 47.63800 49.25690 50.84330 52.66750
## 2012 54.60550 53.02390 54.77580 54.64780
## 2013 63.75210 61.61560 62.63300 61.91030
## 2014 60.86490 61.34200 61.70420 62.75295
## 2015 66.21781 65.05800 66.11709 66.59551
```

```
## 2016 66.73773 66.74769 67.62566 67.90043
## 2017 64.44095 65.08128 64.86261 64.24232
## 2018 72.21528 73.63230 71.85424 70.73107
## 2019 71.33366 71.03945 71.45170 71.19260
```

```r
plot(FER.window, main = "Foreign Exchange Rate from 2010 - 2019", ylab = "Amount in INR for 1 USD")
```



Conclusion : The graph clearly shows a *upward increasing trend* but it does not show any systematic seasonal peaks. Still to check if there is seasonality we shall apply a *seasonal Dummy LM test. (Method 1)*

Step 2 : Checking for seasonality

a) Here we shall fit a dummy to find out if there is seasonality
b) If seasonality exists we will use Ratio to MA to remove the seasonality
c) then we shall check for non stochastic trend, and remove it finally.

a) Checking the existence of seasonality using dummy indices

```
# all libraries required, some dependencies are there.
library(caret)
library(lattice)
library(ggplot2)
library(mltools)
library(data.table)
```

```
library(zoo)

# Saving the months in  a vector
p <- month.abb[cycle(FER.window)]

# Changing the months into factors
p <- as.factor(p)

# fitting dummy using mltools package
# the dummy variable is a list

dmy <- one_hot(as.data.table(p))

# taking trend
t <- c(1: length(FER.window))

# Setting the dummies and creating a dataframe
m <- as.matrix(FER.window)
X <- cbind(as.data.frame(m),t, as.data.frame(dmy))
names(X)[1] <- "Y"
head(X, n= 12)
```

```
##           Y  t p_Apr p_Aug p_Dec p_Feb p_Jan p_Jul p_Jun p_Mar p_May p_Nov
## 1  45.95980  1     0     0     0     0     1     0     0     0     0     0
## 2  46.32790  2     0     0     0     1     0     0     0     0     0     0
## 3  45.49650  3     0     0     0     0     0     0     0     1     0     0
## 4  44.49950  4     1     0     0     0     0     0     0     0     0     0
## 5  45.81150  5     0     0     0     0     0     0     0     0     1     0
## 6  46.56455  6     0     0     0     0     0     0     1     0     0     0
## 7  46.83640  7     0     0     0     0     0     1     0     0     0     0
## 8  46.56710  8     0     1     0     0     0     0     0     0     0     0
## 9  46.05900  9     0     0     0     0     0     0     0     0     0     0
## 10 44.41190 10     0     0     0     0     0     0     0     0     0     0
## 11 45.01650 11     0     0     0     0     0     0     0     0     0     1
## 12 45.15910 12     0     0     1     0     0     0     0     0     0     0
##    p_Oct p_Sep
## 1      0     0
## 2      0     0
## 3      0     0
```

```
## 4       0      0
## 5       0      0
## 6       0      0
## 7       0      0
## 8       0      0
## 9       0      1
## 10      1      0
## 11      0      0
## 12      0      0
```

Here we shall check if the coefficients of the seasonal dummies are significant. If any one of the coefficient is significant then we say that seasonality exists.

Note: To remove multicolinearity we omit one of the dummies, thankfully R automatically does the trick for us.

Here

$$y_t = \beta_0 + \beta_1 t + D_{1t} + \beta_2 D_{2t} + ... + \beta_{12} D_{12t}$$

where D_i= Monthly dummies.
Here for each coefficient β_i the null and alternate hypothesis are as follows:
$H_0 i : \beta_i = 0$
$H_a i : \beta_i \neq 0$

```
summary(lm(Y~.,data = X))
```

```
##
## Call:
## lm(formula = Y ~ ., data = X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7478 -2.4131 -0.0396  2.3897  6.5872
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.698576   1.104161  42.293   <2e-16 ***
## t            0.233142   0.008179  28.505   <2e-16 ***
## p_Apr       -1.801150   1.381665  -1.304    0.195
## p_Aug       -0.335151   1.381084  -0.243    0.809
## p_Dec       -0.305959   1.381278  -0.222    0.825
## p_Feb       -0.764279   1.382246  -0.553    0.581
## p_Jan       -0.419202   1.382609  -0.303    0.762
## p_Jul       -0.982086   1.381157  -0.711    0.479
## p_Jun       -0.846362   1.381278  -0.613    0.541
## p_Mar       -1.380052   1.381931  -0.999    0.320
## p_May       -1.258929   1.381447  -0.911    0.364
## p_Nov       -0.164366   1.381157  -0.119    0.905
## p_Oct       -0.498732   1.381084  -0.361    0.719
## p_Sep              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.088 on 107 degrees of freedom
## Multiple R-squared:  0.8858, Adjusted R-squared:  0.873
## F-statistic: 69.17 on 12 and 107 DF,  p-value: < 2.2e-16
```

The lm summary shows that none of the *seasonal coefficients are significant*, so we do not have enough evidence to reject the null hypothesis. So we can consider that all the monthly coefficients are 0 showing that it has *got no seasonality.*

2.c : Here we shall apply the **Augmented Dickey Fuller(ADF)** test to check for any *Stochastic trend.*

The equation is :

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + ...$$

Null Hypothesis: $H_0 : \gamma = 0$
Alternate Hypothesis : $H_a : \gamma < 0$
Reject Null hypothesis at 5% level of significance if *test statistic value* $<$ *tau*1 value at *5%* significance level i.e. if the test statistic lies on the left side of the critical value at 5% level of significance.

Note: Here the function will automatically choose the lag which takes the *lowest AIC*, and calculates accordingly.

```
library(urca)

# considering there is still a trend in the data, we take type as "trend"
# we provide a maximum lag and the function automatically the optimum lag according to AIC criterion
summary(ur.df(FER.window, type = "trend", lags = 15, selectlags = "AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3295 -0.6869  0.0267  0.5092  3.0874
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.304367   1.635101   3.244  0.00160 **
## z.lag.1     -0.102398   0.034183  -2.996  0.00345 **
## tt           0.018345   0.008152   2.250  0.02661 *
## z.diff.lag   0.271168   0.093629   2.896  0.00464 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.016 on 100 degrees of freedom
## Multiple R-squared:  0.1482, Adjusted R-squared:  0.1226
## F-statistic: 5.798 on 3 and 100 DF,  p-value: 0.001075
##
##
## Value of test-statistic is: -2.9955 4.6702 5.1773
##
## Critical values for test statistics:
```

```
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

This shows that the test statistic is $-2.996 > -3.43$, so the test statistic lies on the right side of the critical value, so we can say that our data has got a unit root, i.e. there is a stochastic trend which needs to be removed first.

For that we take the first lag difference.

```
FER.window.diff1 <- diff(FER.window)
```

Now again we shall check if there is still a stochastic trend. We will confirm the same using the *Augmented Dickey Fuller test*

```
library(urca)
summary(ur.df(FER.window.diff1, type = "trend", lags = 15, selectlags = "AIC"))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22363 -0.73406  0.03552  0.61094  3.09417
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.568140   0.269795   2.106   0.0378 *
## z.lag.1     -1.011365   0.169140  -5.979 3.73e-08 ***
## tt          -0.004674   0.003542  -1.320   0.1900
## z.diff.lag1  0.272309   0.147643   1.844   0.0682 .
## z.diff.lag2  0.143507   0.122377   1.173   0.2438
## z.diff.lag3  0.203568   0.098890   2.059   0.0422 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.048 on 97 degrees of freedom
## Multiple R-squared:  0.415,  Adjusted R-squared:  0.3849
## F-statistic: 13.76 on 5 and 97 DF,  p-value: 3.743e-10
##
##
## Value of test-statistic is: -5.9795 11.9275 17.891
##
## Critical values for test statistics:
```

```
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

This shows that the *test statistic* is $-5.9795 < -3.43$, Since it lies in the left side of the critical value, so we can *reject the null hypothesis*. Thus, we can say that our data has no unit root, i.e. stochastic trend is not there any more.

Now we fit a linear trend to the data to check if there is any deterministic trend.

```
model_select <- function(y, x ,poly.deg)
{
  model <- lm(y~poly(x,poly.deg))
  return(summary(model))

}
t=seq.int(1,length(FER.window.diff1),1)
y <- FER.window.diff1
```

Assuming a linear trend we fit the following model:

$$y_t = a + bt + u_t, \quad where\ u_t \sim N(0,1)$$

Now to determine if a linear trend is present, we test the following hypothesis.
$H_0 : b = 0$
$H_a : b \neq 0$

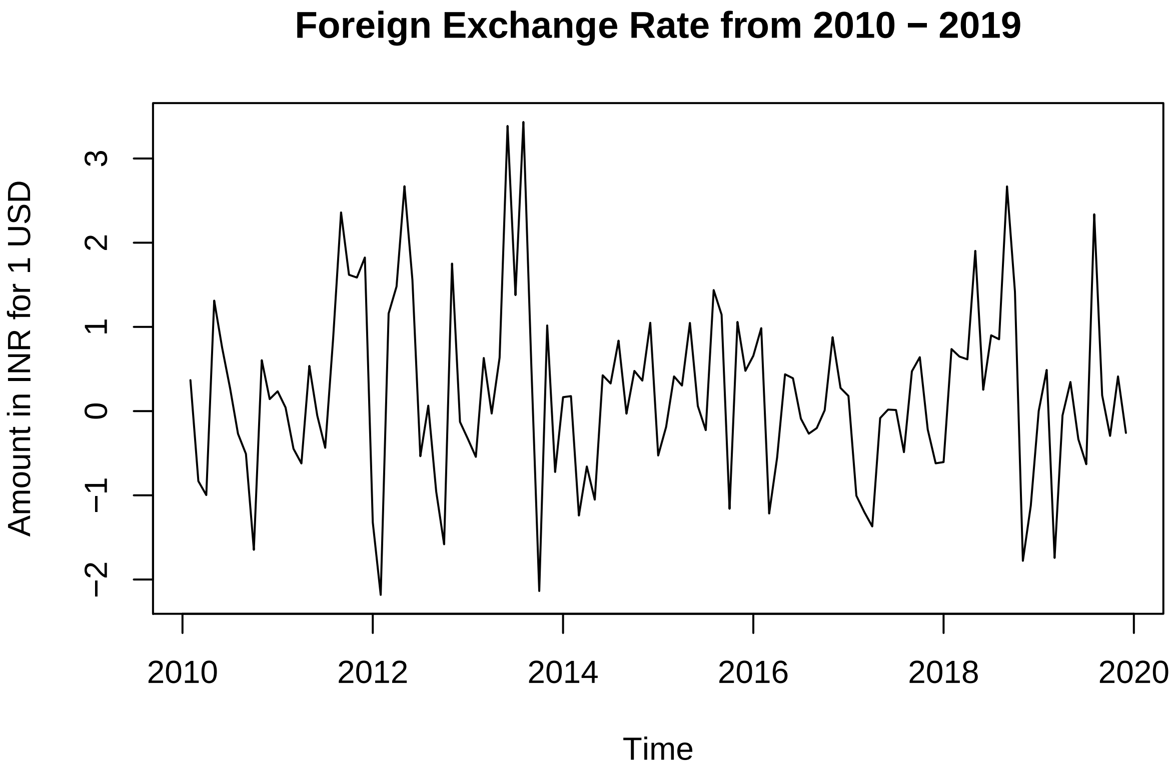
```
model_select(y,t,1)
```

```
##
## Call:
## lm(formula = y ~ poly(x, poly.deg))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4394 -0.7100 -0.0381  0.5282  3.1996
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.21204    0.09695   2.187   0.0307 *
## poly(x, poly.deg) -0.47963    1.05758  -0.454   0.6510
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.058 on 117 degrees of freedom
## Multiple R-squared:  0.001755,   Adjusted R-squared:  -0.006777
## F-statistic: 0.2057 on 1 and 117 DF,  p-value: 0.651
```

Well this shows that *we fail to reject the null hypothesis* that is the coefficient $b = 0$, so here apparently there does not exist a deterministic trend as such in the data.

We plot the difference data once more to see how it looks now.

```r
plot(FER.window.diff1, main = "Foreign Exchange Rate from 2010 - 2019",
     ylab = "Amount in INR for 1 USD")
```



Now that the process is stationary from a **Trend Stationary process**, we proceed forward to building the model.

Step 3 : Sample Dividing Procedure.

Now we shall divide the data into *In sample* and *Out Sample*, we shall keep 9.5 years for the training (in sample) and rest 6 months for forecasting (out sample).

In sample: Jan 2010 - Jun 2019

```
FER.train <-  window(FER.window, end = c(2019,06))
FER.train
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2010 45.95980 46.32790 45.49650 44.49950 45.81150 46.56455 46.83640 46.56710
## 2011 45.39450 45.43680 44.98950 44.36810 44.90480 44.85180 44.41590 45.27850
## 2012 51.34610 49.16350 50.32300 51.80290 54.47350 56.03020 55.49480 55.55980
## 2013 54.31680 53.77370 54.40460 54.37570 55.01080 58.39730 59.77540 63.20884
## 2014 62.07600 62.25400 61.01400 60.35662 59.30500 59.73070 60.05860 60.89520
## 2015 62.22593 62.03761 62.44984 62.75316 63.80033 63.86074 63.63498 65.07233
## 2016 67.25233 68.23767 67.02185 66.46953 66.90673 67.29686 67.20762 66.93964
## 2017 68.08037 67.07545 65.87666 64.50709 64.42484 64.44299 64.45588 63.96838
## 2018 63.63690 64.37384 65.02133 65.63635 67.53936 67.79307 68.69336 69.54651
## 2019 70.73287 71.22177 69.47865 69.42741 69.77312 69.43886
##           Sep      Oct      Nov      Dec
## 2010 46.05900 44.41190 45.01650 45.15910
## 2011 47.63800 49.25690 50.84330 52.66750
## 2012 54.60550 53.02390 54.77580 54.64780
## 2013 63.75210 61.61560 62.63300 61.91030
## 2014 60.86490 61.34200 61.70420 62.75295
## 2015 66.21781 65.05800 66.11709 66.59551
## 2016 66.73773 66.74769 67.62566 67.90043
## 2017 64.44095 65.08128 64.86261 64.24232
## 2018 72.21528 73.63230 71.85424 70.73107
## 2019
```

Out sample: July 2019- Dec 2019

```
FER.test <- window(FER.window, start= c(2019,07), end= c(2019,12))
FER.test
```

```
##           Jul      Aug      Sep      Oct      Nov      Dec
## 2019 68.80830 71.14570 71.33366 71.03945 71.45170 71.19260
```

```
FER.train.d1 <- window(FER.window.diff1, end = c(2019,06))
```

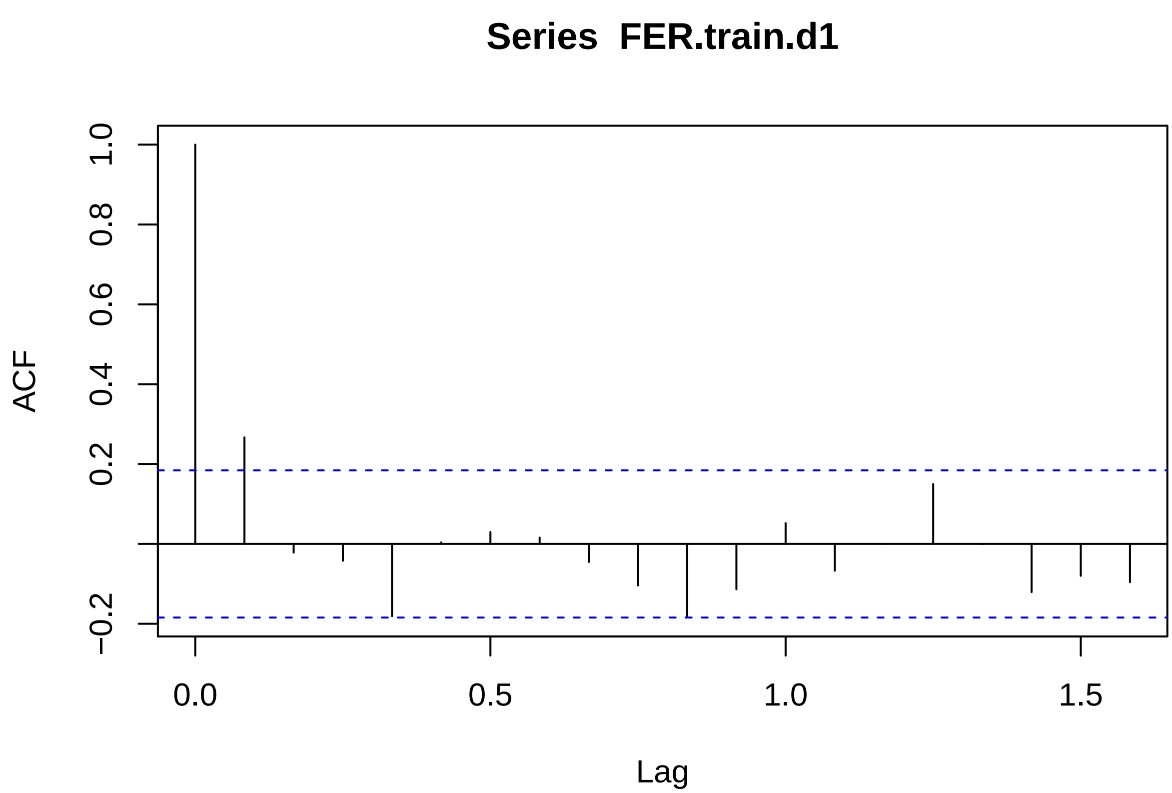
Step 4 : The ACF analysis.

Here we shall make a table calculating all the acf, its test statistic and draw conclusion accordingly. We will be using a maximum lag of 19 for all the below calculations.

```
max_lag <- 19
max_lag
```

```
## [1] 19
```

```
acf(FER.train.d1, lag.max = max_lag)
```




```
rho <- acf(FER.train.d1, lag.max = max_lag, plot = F)
length(rho$acf)
```

```
## [1] 20
```

Ljung-Box test:

Null Hypothesis: H_0: $\rho_1 = ... = \rho_k = 0$, at a given lag k.
Alternative Hypo: H_1: $\rho_i \neq 0 \; \forall \; i = 1, 2, ...k$

Test Statistic:

$$Q(k) = n(n+2)(\sum_{j=1}^{k} \frac{\hat{e}_j^2}{n-j}) \sim \chi_k^2$$

Note : In statistical hypothesis testing, the **p-value** or probability value is the probability of obtaining test results at least as extreme as the results actually observed, assuming that the null hypothesis is correct. So here, the P-value of the test is the probability that a **chi-square test statistic** having the respective degrees of freedom is more extreme than the obtained test statistic.

Interpret results: If the P-value is less than the significance level (0.05), we reject the null hypothesis.

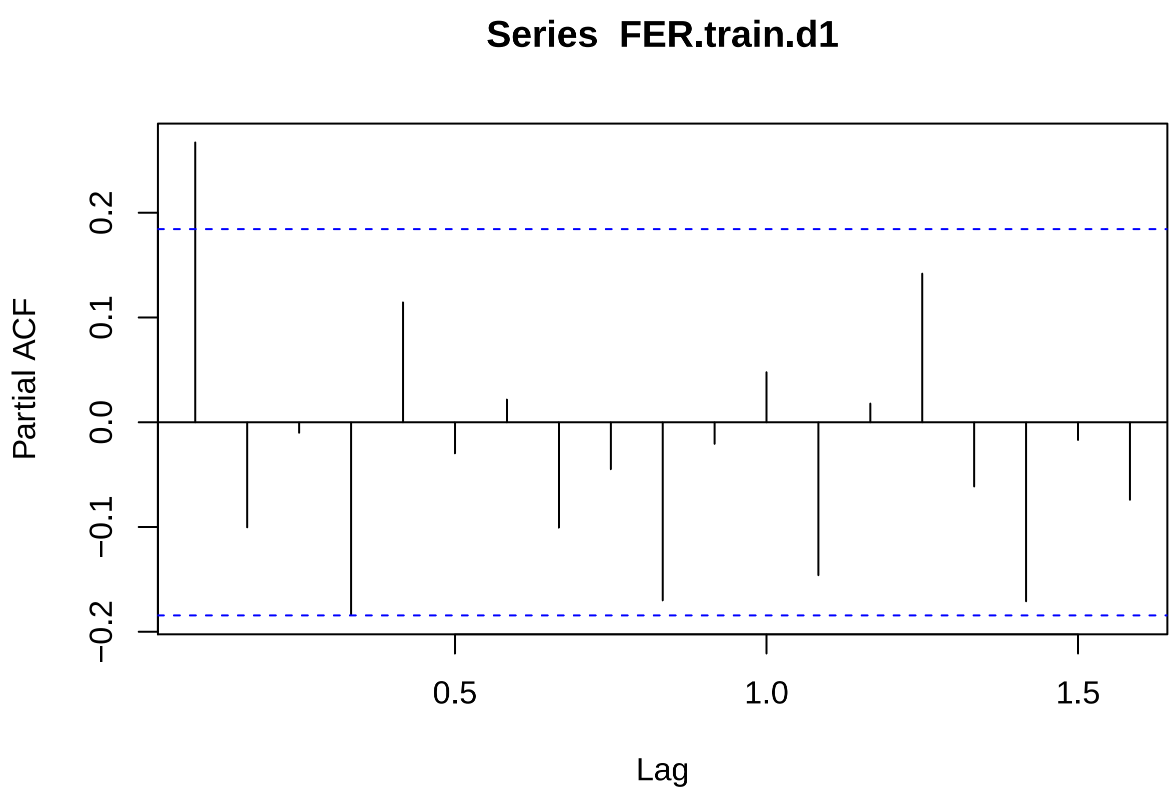
```
k <- max_lag+1
Qt <- t(sapply(1:k,function(i) Box.test(FER.train.d1, lag = i, type = "Ljung-Box")))
Qt <- as.data.frame(Qt)

result <- NULL

for(i in 1:k){
  if(Qt$p.value[[i]] < 0.05){
  result[i] <- "reject"

} else {result[i] <- "accept"}
}
acf_table <- data.frame(K= 1:k,
                        SAMPLE_ACF = rho$acf,
                        P_VALUES=as.numeric(Qt$p.value),
                        TEST_STATISTIC = as.numeric(Qt$statistic),
                        RESULT = as.vector(result))
acf_table
```

```
##     K    SAMPLE_ACF    P_VALUES TEST_STATISTIC RESULT
## 1   1  1.0000000000 0.004020672       8.274455 reject
## 2   2  0.2670487686 0.015528406       8.330169 reject
## 3   3 -0.0218148561 0.036041174       8.542095 reject
## 4   4 -0.0423545917 0.014267101      12.455446 reject
## 5   5 -0.1811755242 0.029031674      12.457286 reject
## 6   6  0.0039105730 0.050475023      12.565706 accept
## 7   7  0.0298784610 0.082557834      12.596998 accept
## 8   8  0.0159763809 0.117062895      12.851480 accept
## 9   9 -0.0453456619 0.115280121      14.203149 accept
## 10 10 -0.1040072421 0.048174300      18.426978 reject
## 11 11 -0.1829715892 0.044313757      20.075663 reject
## 12 12 -0.1137576820 0.059447429      20.425830 accept
## 13 13  0.0521687134 0.072696056      21.011775 accept
## 14 14 -0.0671491335 0.101331016      21.011780 accept
## 15 15 -0.0001863997 0.065125775      23.998099 accept
## 16 16  0.1500697026 0.089546022      23.998099 accept
## 17 17  0.0000281360 0.075020495      25.969364 accept
## 18 18 -0.1206757426 0.081970938      26.843864 accept
## 19 19 -0.0799564557 0.081220442      28.116055 accept
## 20 20 -0.0959294463 0.100331763      28.396902 accept
```

We see that the acfs are partly diminishing and partly significant but cuts off entirely after lag 11. So we can try and fit a *MA(11)* model. After looking at the PACF and deciding the same.

Step 5: Analysing PACF

```
pacf(FER.train.d1, lag.max = max_lag)
```



PACF

Null & Alternate Hypothesis:

H_0: $\phi_1 = \phi_2 = ... = \phi_k = 0$, at a given lag k.
H_1: $\phi_i \neq 0 \; \forall \; i = 1, 2, ...k$

Test Statistic:

$$Q(k) = \sqrt{n}\hat{\phi}_k \sim N(0, 1)$$

Interpret results: The null Hypothesis follows a standard normal distribution, so at *5%* level of significant test statistic value is *1.96* $\implies$ if $|Q(k)| < 1.96$ we reject the null hypothesis else we shall accept.

```
n <- length(FER.train)
phi <- pacf(FER.train.d1, lag.max = max_lag, plot = F)
Qt <- as.numeric(phi$acf*sqrt(n))
Qt <- abs(Qt)
result <- NULL

for(i in 1:max_lag){
  if(abs(Qt[i]) > 1.96){
  result[i] <- "reject"
```

```r
  } else {result[i] <- "accept"}
}
pacf_table <- data.frame(K= 1:max_lag,
                         SAMPLE_PACF = phi$acf,
                         TEST_STATISTIC = Qt,
                         RESULT = as.vector(result))
pacf_table
```

```
##     K  SAMPLE_PACF TEST_STATISTIC RESULT
## 1   1  0.267048769      2.8513006 reject
## 2   2 -0.100281479      1.0707132 accept
## 3   3 -0.009968734      0.1064369 accept
## 4   4 -0.183011860      1.9540320 accept
## 5   5  0.114332277      1.2207347 accept
## 6   6 -0.029601284      0.3160552 accept
## 7   7  0.021611046      0.2307428 accept
## 8   8 -0.100555949      1.0736437 accept
## 9   9 -0.044779572      0.4781150 accept
## 10 10 -0.170021465      1.8153325 accept
## 11 11 -0.020593376      0.2198771 accept
## 12 12  0.047755754      0.5098919 accept
## 13 13 -0.145984585      1.5586888 accept
## 14 14  0.017810319      0.1901622 accept
## 15 15  0.141827403      1.5143023 accept
## 16 16 -0.061289975      0.6543979 accept
## 17 17 -0.170879068      1.8244892 accept
## 18 18 -0.016887391      0.1803080 accept
## 19 19 -0.073970733      0.7897913 accept
```

This shows that the pacf cuts off entirely after lag 1. Suggesting a *AR(1)* model. Since both the models seems feasible, so we shall select the model depending on the min **AIC** and **BIC**.

Step 6 : Model fitting and Residual Analysis.

Model AR(1) | Model MA(11)

```
FER.model1 <- arima(FER.train, order = c(1,1,0))
FER.model2 <- arima(FER.train, order = c(0,1,11))
```

AIC and BIC of AR(1) and MA(11)

```
bic1 <- BIC(FER.model1)
bic2 <- BIC(FER.model2)
```

```
cat("AR(1) BIC : ", bic1, "\n", "MA(11) BIC : ", bic2, sep = "")
```

```
## AR(1) BIC : 335.9623
## MA(11) BIC : 373.2594
```

```
cat("\n","AR(1) AIC : ", FER.model1$aic, "\n", "MA(11) AIC : ", FER.model2$aic, sep = "")
```

```
##
## AR(1) AIC : 330.5075
## MA(11) AIC : 340.5307
```

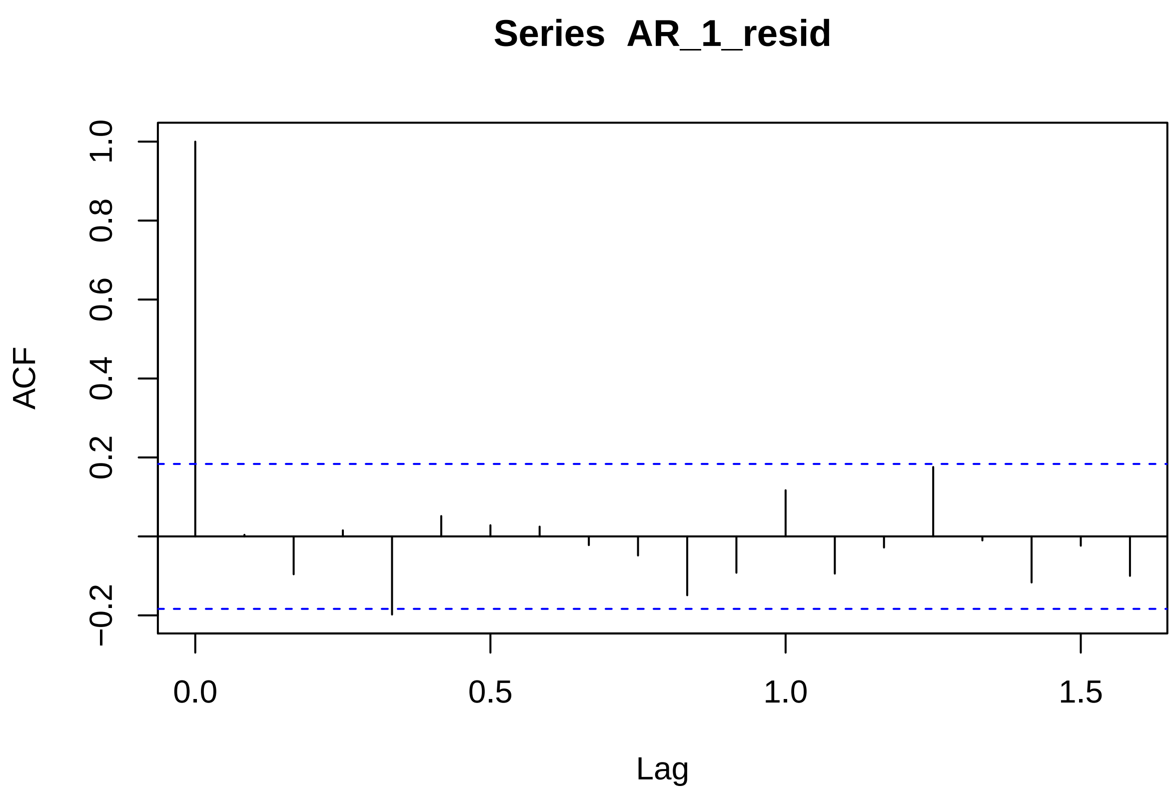
As both AIC and BIC of model AR(1) is less than MA(11), so **we select AR(1) model as the final model**.

```
FER.model1
```

```
##
## Call:
## arima(x = FER.train, order = c(1, 1, 0))
##
## Coefficients:
##          ar1
##       0.2928
## s.e.  0.0895
##
## sigma^2 estimated as 1.052:  log likelihood = -163.25,  aic = 330.51
```

Now we shall test the residuals to justify our model specification.

```
AR_1_resid <- na.omit(FER.model1$residuals)
acf(AR_1_resid, lag.max = max_lag, plot = T)
```



Again on the **residuals** we apply **Ljung Box Test**.

Remember: Ljung-Box test:

Null Hypothesis: H_0: $\rho_1 = ... = \rho_k = 0$, at a given lag k.
Alternative Hypo: H_1: $\rho_i \neq 0 \; \forall \; i = 1, 2, ...k$

Test Statistic:

$$Q(k) = n(n+2)(\sum_{j=1}^{k} \frac{\hat{e}_j^2}{n-j}) \sim \chi^2_{k-p-q}$$

where p, q are the parameters of the model.

Note : In statistical hypothesis testing, the **p-value** or probability value is the probability of obtaining test results at least as extreme as the results actually observed, assuming that the null hypothesis is correct. So here, the P-value of the test is the probability that a **chi-square test statistic** having the respective degrees of freedom is more extreme than the obtained test statistic. Also note that the degrees of freedom here is k-p-q, so the first two values comes out to be null. In the given code we have removed the first two values.

Interpret results: If the P-value is less than the significance level (0.05), we reject the null hypothesis.

```r
parameter <- sum(eval((FER.model1$call$order)))
resid_acf <- acf(AR_1_resid, lag.max = max_lag, plot = F)
Rt <- t(sapply(1:k,function(i) Box.test(AR_1_resid, lag = i, type = "Ljung-Box", fitdf = parameter)))
Rt <- as.data.frame(Rt)
result <- NULL

for(i in (parameter+1):k)
  {
  if(Rt$p.value[[i]] < 0.05){
  result[i] <- "reject"

} else {result[i] <- "accept"}
}
residual_table <- cbind(K= 1:(k-parameter),na.omit(data.frame(
                           RESIDUAL_ACF = resid_acf$acf,
                            P_VALUES=as.numeric(Rt$p.value),
                            TEST_STATISTIC = as.numeric(Rt$statistic),
                            RESULT = as.vector(result)))
                           )
residual_table
```

```
##      K RESIDUAL_ACF   P_VALUES TEST_STATISTIC RESULT
## 3    1  -0.09595970 0.29048530       1.117371 accept
## 4    2   0.01539112 0.05433082       5.825327 accept
## 5    3  -0.19789333 0.10463115       6.147922 accept
## 6    4   0.05156562 0.18151178       6.246042 accept
## 7    5   0.02830798 0.27612796       6.322085 accept
## 8    6   0.02480504 0.38167669       6.383037 accept
## 9    7  -0.02210376 0.46318245       6.677777 accept
## 10   8  -0.04837633 0.30165379       9.503001 accept
## 11   9  -0.14906019 0.30476967      10.591251 accept
## 12  10  -0.09206651 0.26141221      12.364340 accept
## 13  11   0.11694561 0.26027279      13.527110 accept
## 14  12  -0.09423799 0.32481684      13.632085 accept
## 15  13  -0.02817490 0.16684542      17.760029 accept
## 16  14   0.17579359 0.21726456      17.774087 accept
## 17  15  -0.01020708 0.18651450      19.631856 accept
## 18  16  -0.11673479 0.23364933      19.708250 accept
## 19  17  -0.02354962 0.22187334      21.099452 accept
## 20  18  -0.09997132 0.26599947      21.270230 accept
```

This shows that our model fit is *Good enough* as we fail to reject the null hypothesis showing that the residue is plain white noise present. This means that our model fitting is Alright.

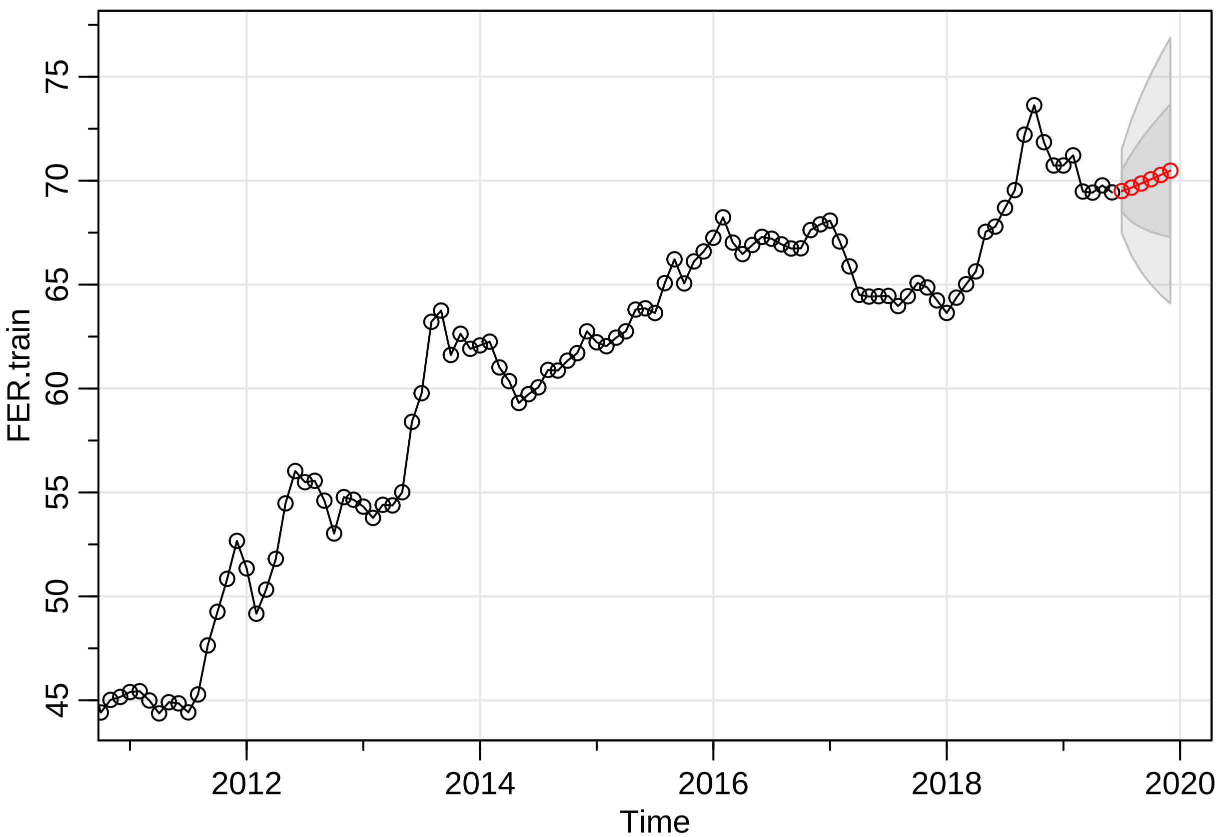
Therefore, the correctly specified model is a AR(1) model given by:

$$y_t = 0.2928 + 0.0895 y_{t-1} + e_t \quad where \quad e_t \sim WN$$

Step 7 : Forecasting on the Out Sample.

Now we shall forecast 6 months using our correctly specified model and find the **RMSE** and **MAPE**.

```
library(astsa)
FER.pred <- sarima.for(FER.train, n.ahead = 6 , p = 1, d = 1, q= 0, P=0, D= 0, Q= 0, 12)
```



The plot shows the **training(in sample data)** in **black** and the **predicted(out sample data)** in **red**.

Here is a table which shows the **actual** and **predicted values**.
We see that the values are close enough.

```
PRED_ACTtable <- data.frame(Predict = FER.pred$pred, Actual = FER.test)
PRED_ACTtable
```

```
##    Predict   Actual
## 1 69.50190 68.80830
## 2 69.67038 71.14570
## 3 69.86684 71.33366
## 4 70.07072 71.03945
## 5 70.27658 71.45170
## 6 70.48296 71.19260
```

```
rmse_DL <- sqrt(mean((PRED_ACTtable$Predict- PRED_ACTtable$Actual)^2))
mape_DL <- mean(abs(PRED_ACTtable$Predict- PRED_ACTtable$Actual)/abs(PRED_ACTtable$Actual))*100
```

We display the **RMSE** and **MAPE** values of the forecast are as follows:

```
cat("RMSE : ", rmse_DL,"\n", "MAPE : ", mape_DL, sep = "")
```

```
## RMSE : 1.127838
## MAPE : 1.523837
```

This is the entire analysis of the *Foreign Exchange Rate.*