Time Series Data Analysis

Analysing:

- **Reliance Communication**
- **Indian Foreign Trade (Export)**
- **S&P BSE Sensex**
- **Power Consumption**
- **Foreign Exchange Reserves weekly for Gold**

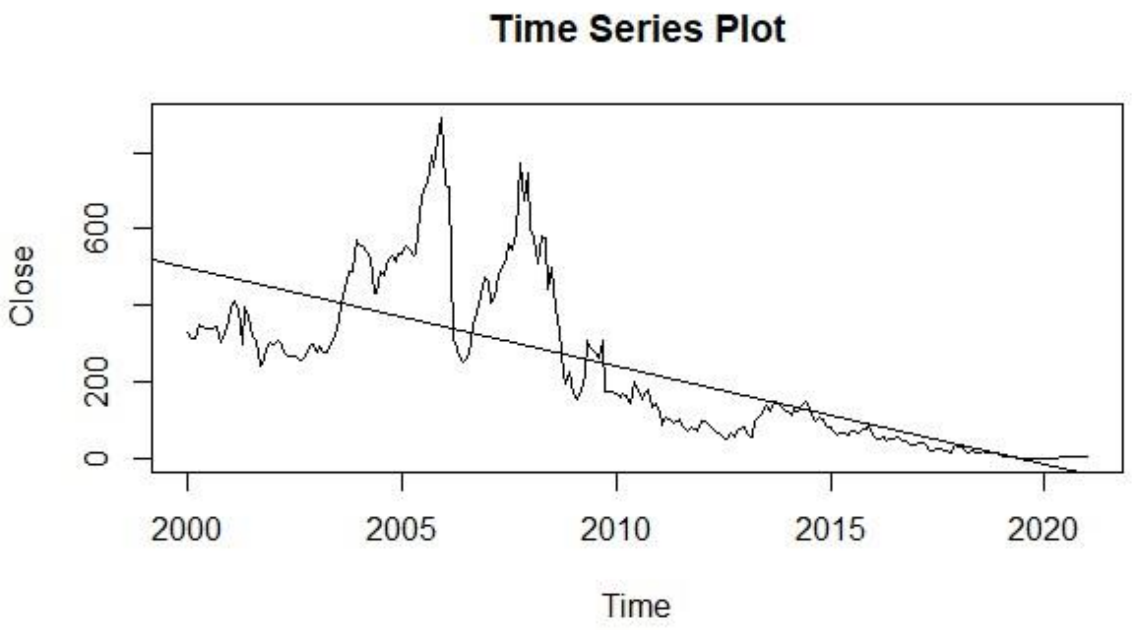
Reliance Communication:

The data consists of closing prices of stock per month starting from 1st January 2000 to 18th December 2020. The data was collected from the website of yahoo finance. The link is given below. The data consists of 254 data points which also has columns of opening stock prices for the month, high stock price for the month, low stock price for the month, closing and adjusted stock prices for the month along with volume. Since univariate time series is taken into picture consideration of a single variable and working on it is essential with respect to the course curriculum. Hence closing prices of the stocks of reliance communication is taken into account. We proceed further with the analysis of the closing prices for the month.

Link:
https://in.finance.yahoo.com/quote/RCOM.BO/history?period1=946857600&period2=1608336000&interval=1mo&filter=history&frequency=1mo&includeAdjustedClose=true

```
> df=read.table("E:/Xavier/M.Sc. Part2 semester 3/Time Series/Final Assignment/Reliance communications/RCOM.BO.csv", header=T, sep="\t")
> head(df)
        Date.Open.High.Low.Close.Adj.Close.Volume
1       01-01-2000,240,339.5,240,329,321.313599,206245880
2       01-02-
        2000,328.649994,380,311.450012,311.450012,304.173706,125524617
3       01-03-2000,305,322.5,199,314.5,307.152344,131331592
4       01-04-
        2000,320.75,376.899994,279.100006,346.899994,338.79541,17481448
        4 5    01-05-
        2000,346.899994,357.399994,304,343.700012,335.670166,94802462
6   01-06-2000,342,353.799988,323.100006,340.899994,332.935577,109300215
> str(df)
'data.frame':	253 obs. of  1 variable:
 $ Date.Open.High.Low.Close.Adj.Close.Volume: chr  "01-01-2000,240,339.5,240,329,321.313599,206245880" "01-02-2000,328.649994,380,311.450012,311.450012,304.173706,125524617" "01-03-2000,305,322.5,199,314.5,307.152344,131331592" "01-04-2000,320.75,376.899994,279.100006,346.899994,338.79541,174814484" ...
```

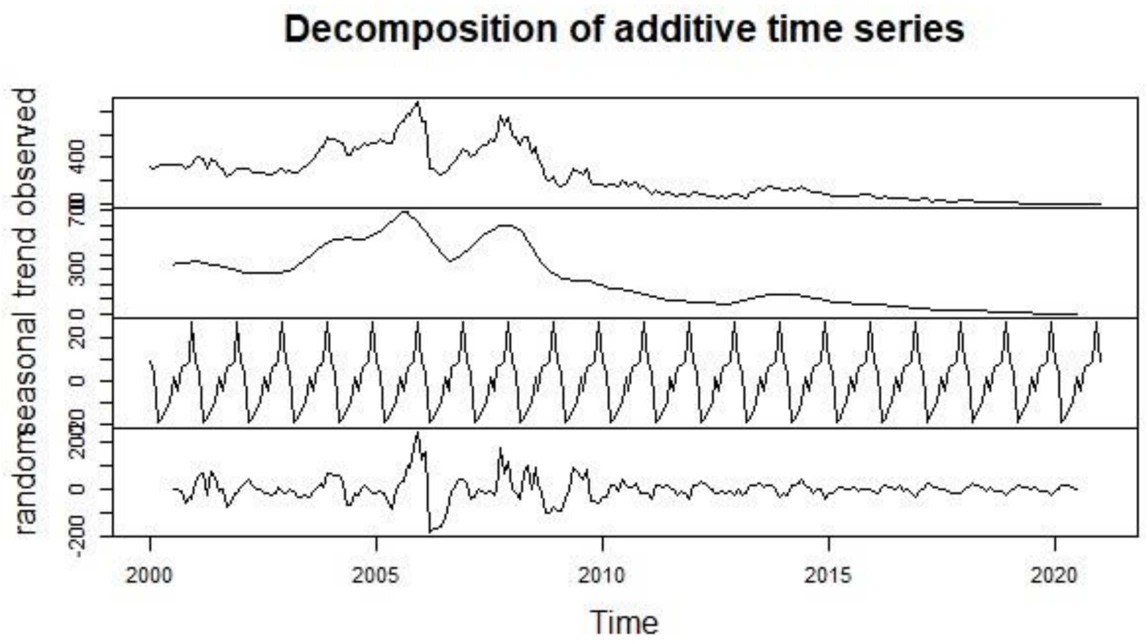
```
R=read.table("clipboard",sep="\t",header = T)
> View(R)
> #Explonatory analysis
> rel=ts(R, frequency = 12, start=c(2000,1))
> plot.ts(rel, plot.type = "single", main="Time Series Plot")
>  abline(reg=lm(rel~time(rel)))
```



The above plot is the time series plot for the closing stock prices of reliance communications. It shows a downwards decreasing trend. This leads us to saying that that the stock prices are decreasing over the years.

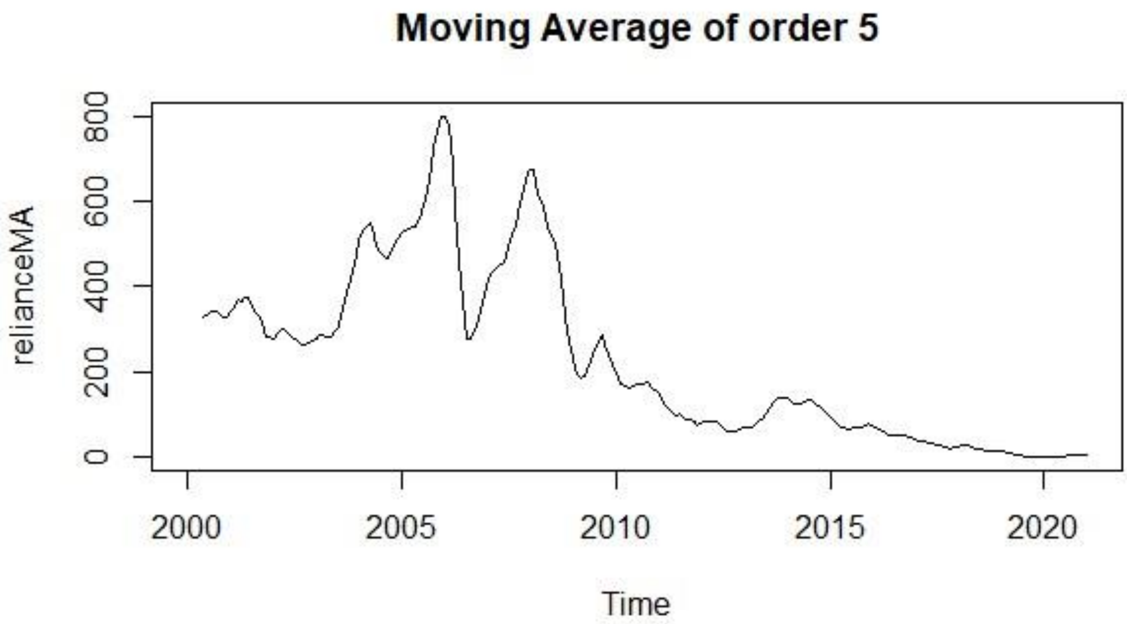
```
> reliance=decompose(rel)
> plot(reliance)
```

After decomposing the series we observe that the series has trend and seasonality. Due to unavailability of the appropriate amount of the data its difficult to obtain cyclic effects in the time series. The trend is decreasing in nature, whereas seasonality is expressed in terms of highs and lows. Random component seems to be volatile between year 2005 to 2010.



library("TTR")

```
> #Moving Average of order 5
> relianceMA=SMA(rel, n=5)
> plot.ts(relianceMA, main="Moving Average of order 5")
```

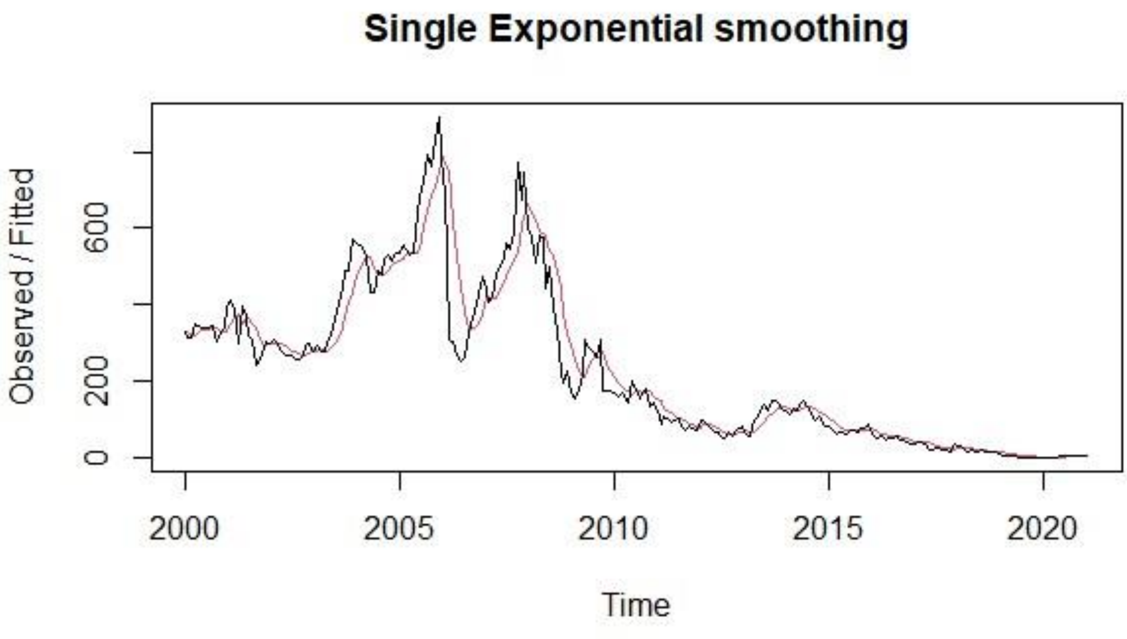


Moving average graph of order 5 shows the similar plot of the time series plot, but in a smoothed manner.

#Holt Winter

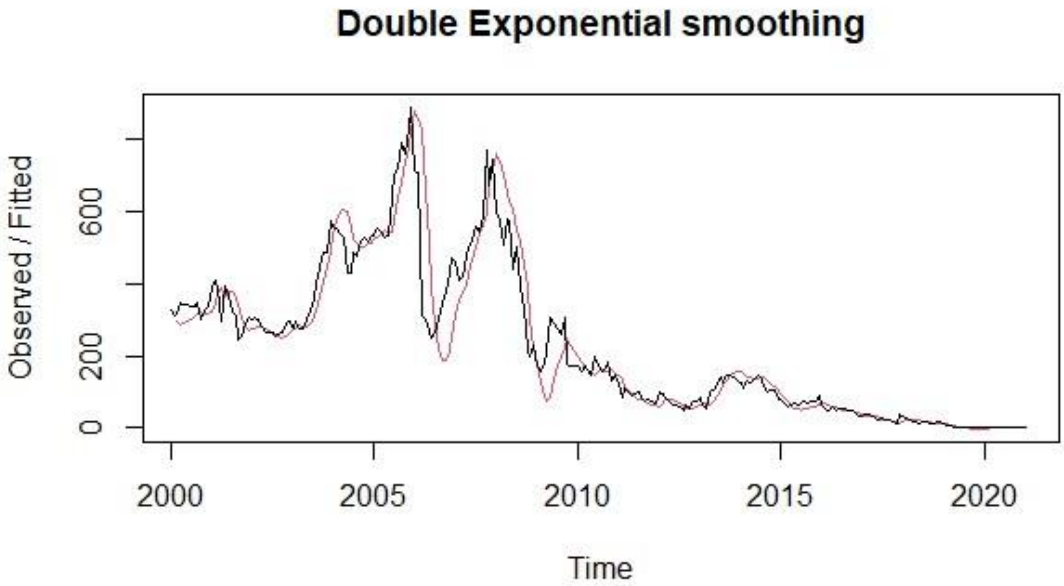
> relianceHW=HoltWinters(rel, alpha=0.3, beta=FALSE, gamma=FALSE)
> plot(relianceHW, main="Single Exponential smoothing")

Alpha =0.3



Single exponential smoothing doesn’t leads us to any conclusion. Although it isn’t a good fit to the data. It’s somewhat represents the given time series.

relianceHW1=HoltWinters(rel, alpha=0.3, beta=0.2, gamma=FALSE)
> plot(relianceHW1, main="Double Exponential smoothing") Alpha =0.3 Beta=0.2

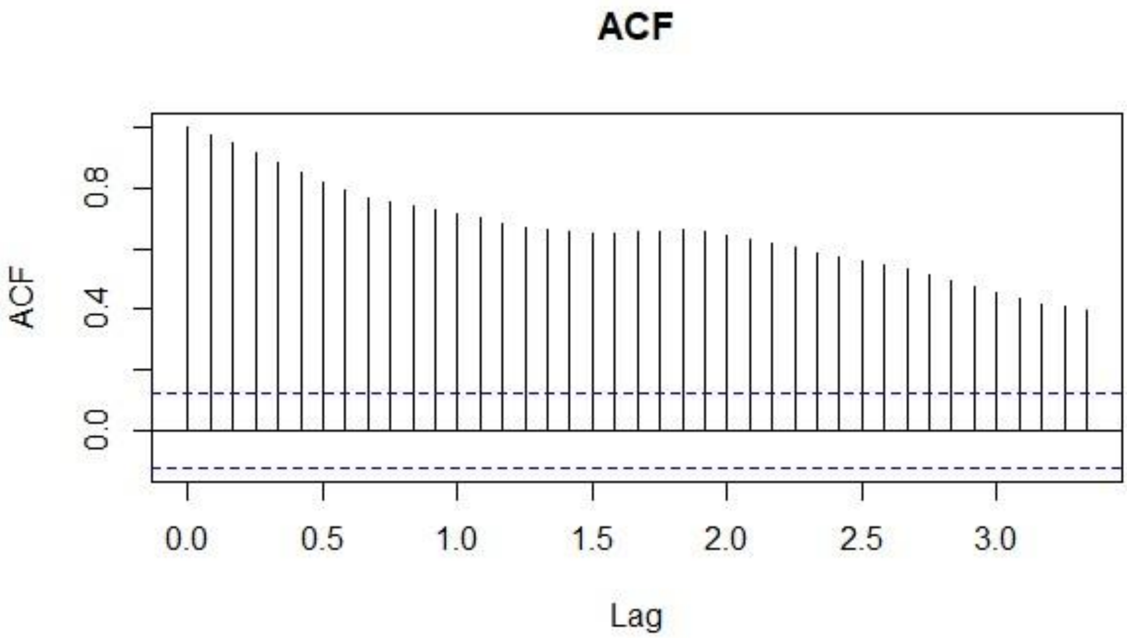


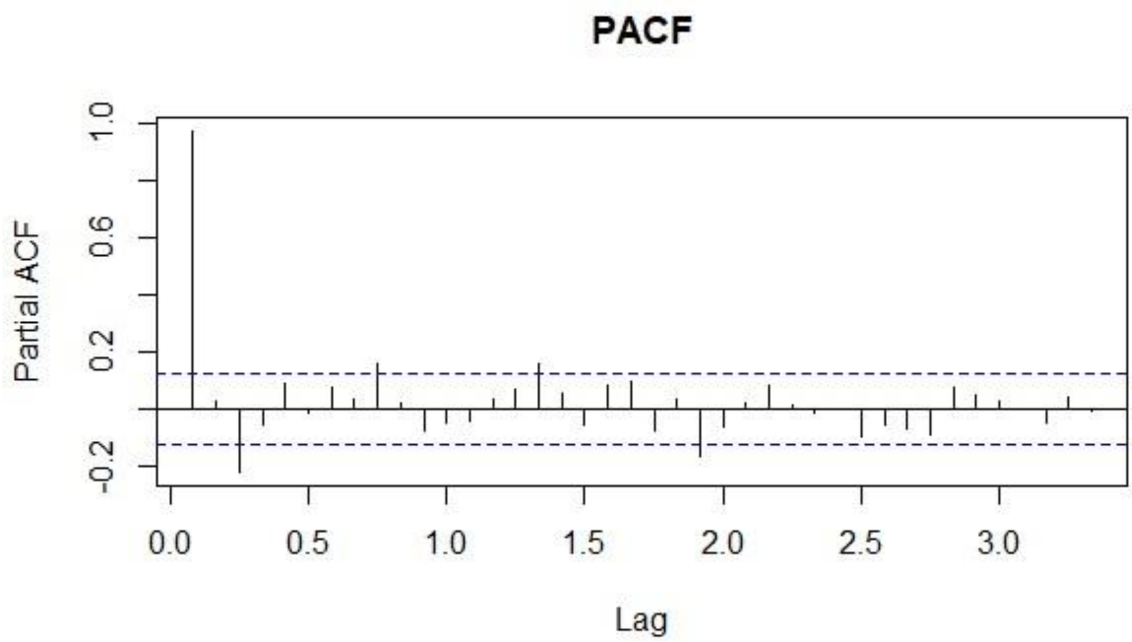
reldiff=diff(rel, differences = 1)

> plot.ts(reldiff)

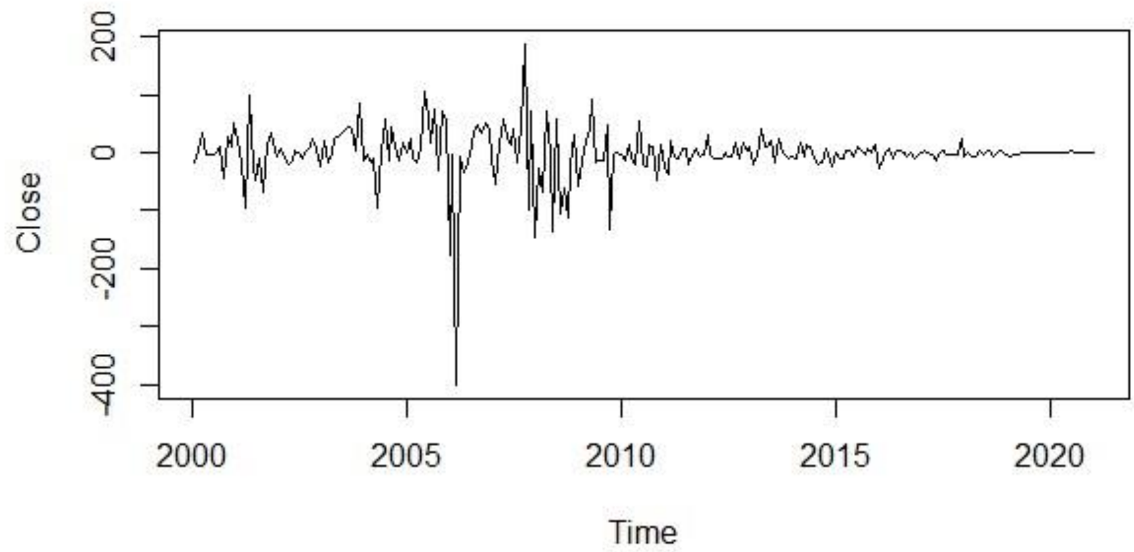
From the above graph it is clear that single and double exponential smoothing is not a very appropriate models to fit the given time series data. We might need to carry out further analysis to fit the given time series data.

We plot ACF and PACF to check whether We can get some information to proceed.



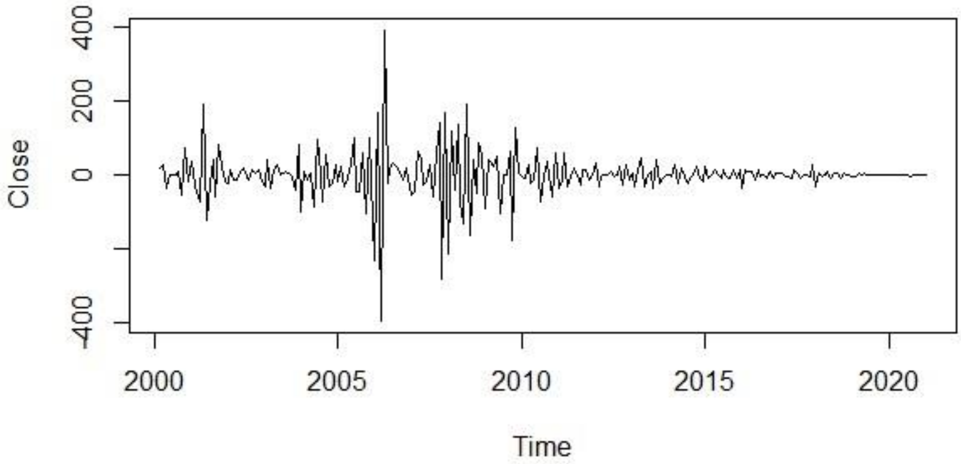


Since ACF and PACF couldn't lead to the conclusion of the selection of the model we might need to carry out further analysis for the selection of the model. To carry out further analysis we proceed by differencing the given time series data and check whether it is stationary after differencing.



reldiff1=diff(rel, differences = 2)

> plot.ts(reldiff1)



#Univariate analysis
> acf(rel, lag=40, main="ACF")
> pacf(rel, lag=40, main="PACF")

Here we can see that the series has appeared to be stationary after the first and second differencing of the series.

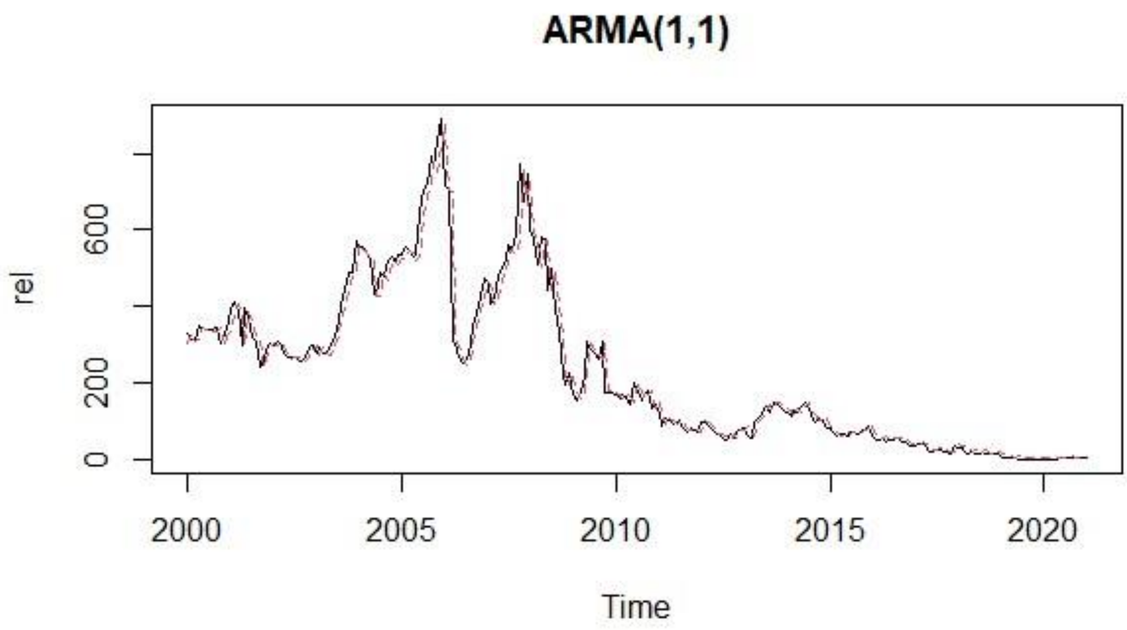
Fitting of ARMA(1,1):

#Fitting ARMA(1,1) model
> relAR1=arima(rel, order=c(1,0,1))
> print(relAR1)

```
Call:
arima(x = rel, order = c(1, 0, 1))

Coefficients:
         ar1      ma1  intercept
      0.9772  -0.0204   210.3588
s.e.  0.0124   0.0525   103.6978

sigma^2 estimated as 1956:  log likelihood = -1319.2,  aic = 2646.41
```

> ts.plot(rel, main="ARMA(1,1)")
> relARfit1=rel-residuals(relAR1)

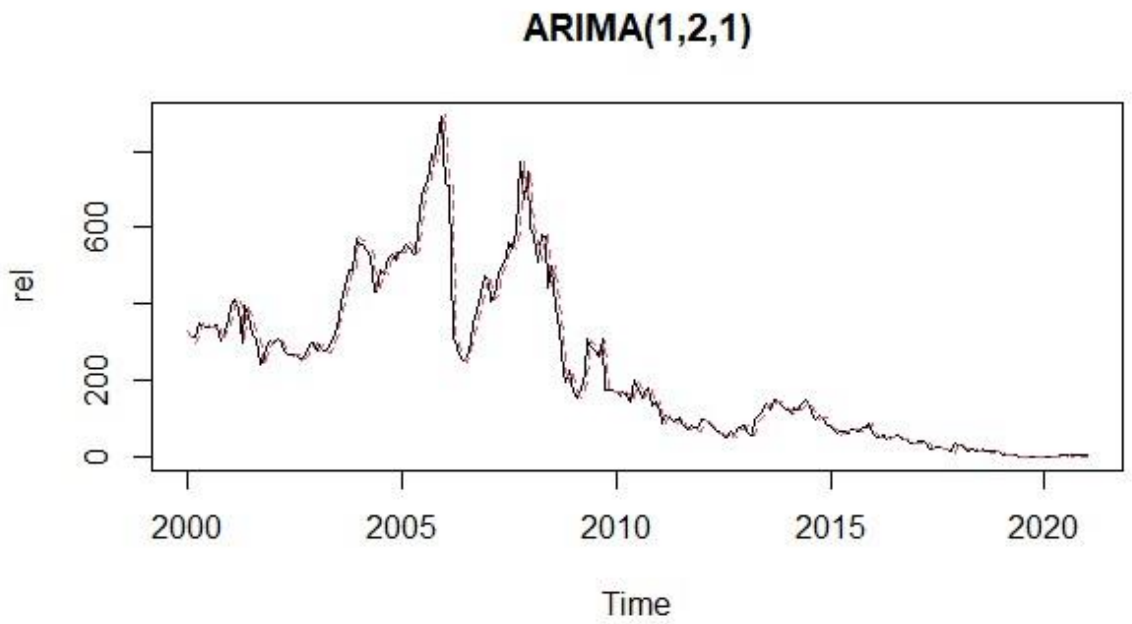
> points(relARfit1, type="l", col=2, lty=2)



ARMA(1,1) is not a good fit for the data since it shows irregularities. But AIC and BIC would be the determining factors during the selection.

Fitting of ARIMA(1,2,1) :

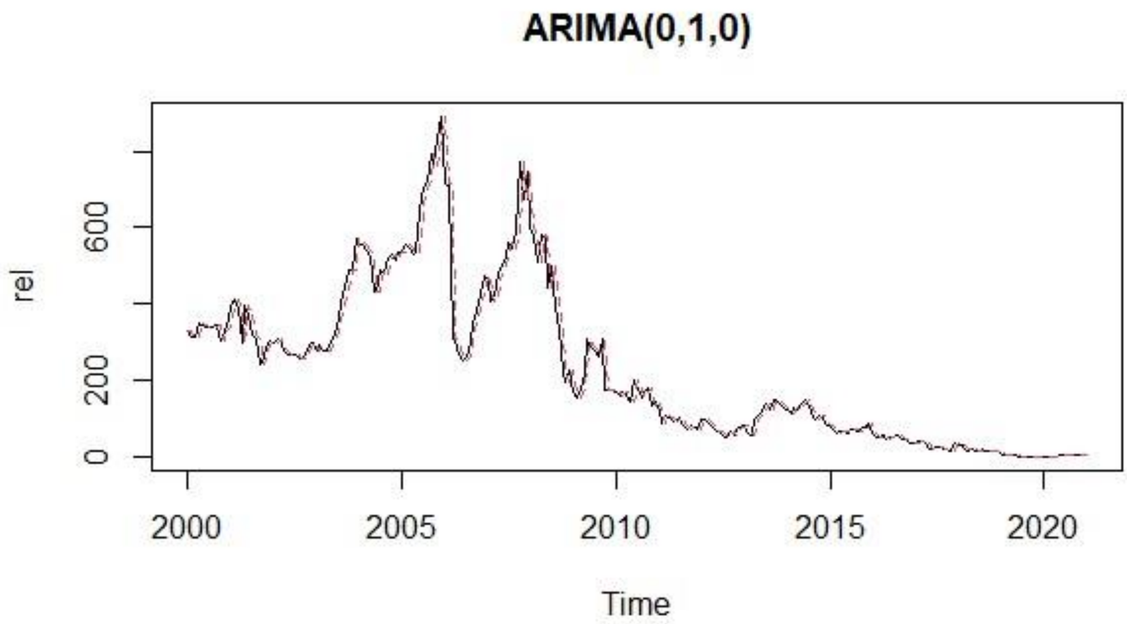
```
#Fitting ARIMA(1,2,1) model
> relAR2=arima(rel, order=c(1,2,1))
> print(relAR2,)

Call:
arima(x = rel, order = c(1, 2, 1))

Coefficients:
          ar1      ma1
      -0.0376  -1.0000
s.e.   0.0631   0.0121

sigma^2 estimated as 1985:  log likelihood = -1311.9,  aic = 2629.79
> ts.plot(rel, main="ARIMA(1,2,1)")
> relARfit2=rel-residuals(relAR2)
> points(relARfit2, type="l", col=2, lty=2)
```



We can see from the above graph the ARIMA(1,2,1) fitting for the given time series data.

Fitting of ARIMA(0,1,0):

```
#Fitting ARIMA(0,1,0) model
> relAR3=arima(rel, order=c(0,1,0))
> print(relAR3)

Call:
arima(x = rel, order = c(0, 1, 0))


sigma^2 estimated as 1982:  log likelihood = -1314.13,  aic = 2630.26
> ts.plot(rel, main="ARIMA(0,1,0)")
> relARfit3=rel-residuals(relAR3)
> points(relARfit3, type="l", col=2, lty=2)
```

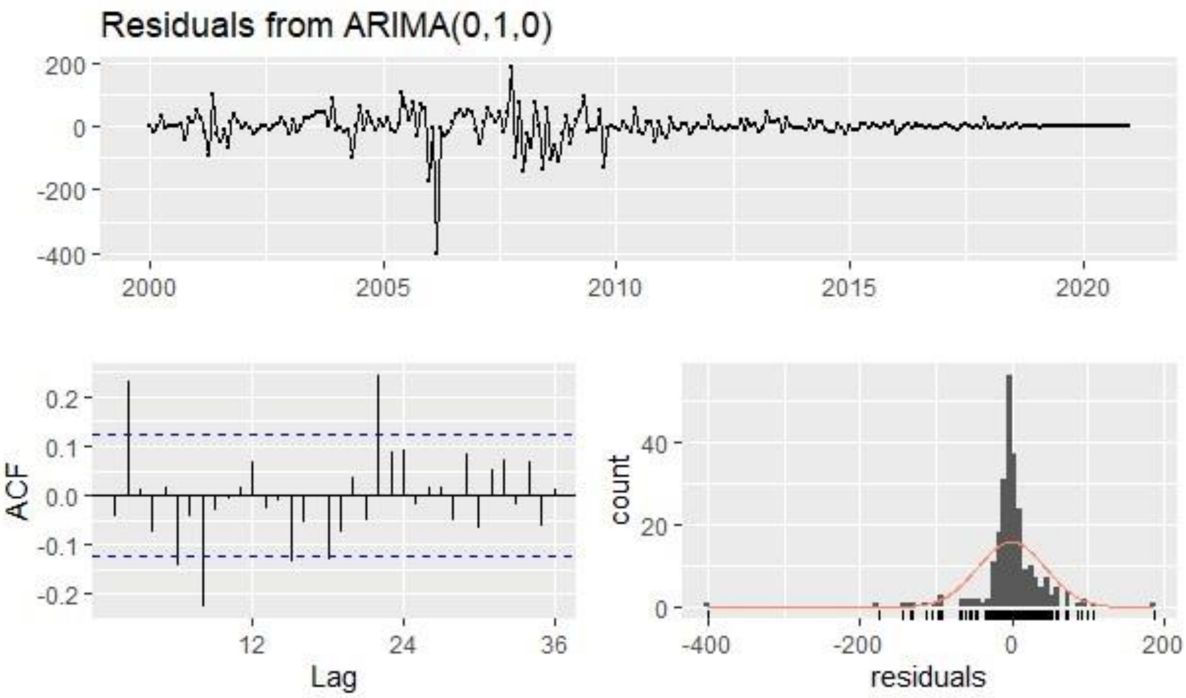



```
> AIC(relAR1)  #AIC of ARMA(1,1) model
[1] 2646.406
> AIC(relAR2)  #AIC of ARIMA(1,1,1) model
[1] 2631.128
> AIC(relAR3)  #AIC of ARIMA(0,1,0) model
[1] 2630.263
> #BIC
> BIC(relAR1)  #BIC of ARMA(1,1) model
[1] 2660.539
> BIC(relAR2)  #BIC of ARIMA(1,1,1) model
[1] 2641.716
> BIC(relAR3)  #BIC of ARIMA(0,1,0) model
[1] 2633.792
```

Models	AIC	BIC
ARMA(1,1)	2646.406	2660.539
ARIMA(1,1,1)	2631.128	2641.716
ARIMA(0,1,0)	2630.263	2633.792

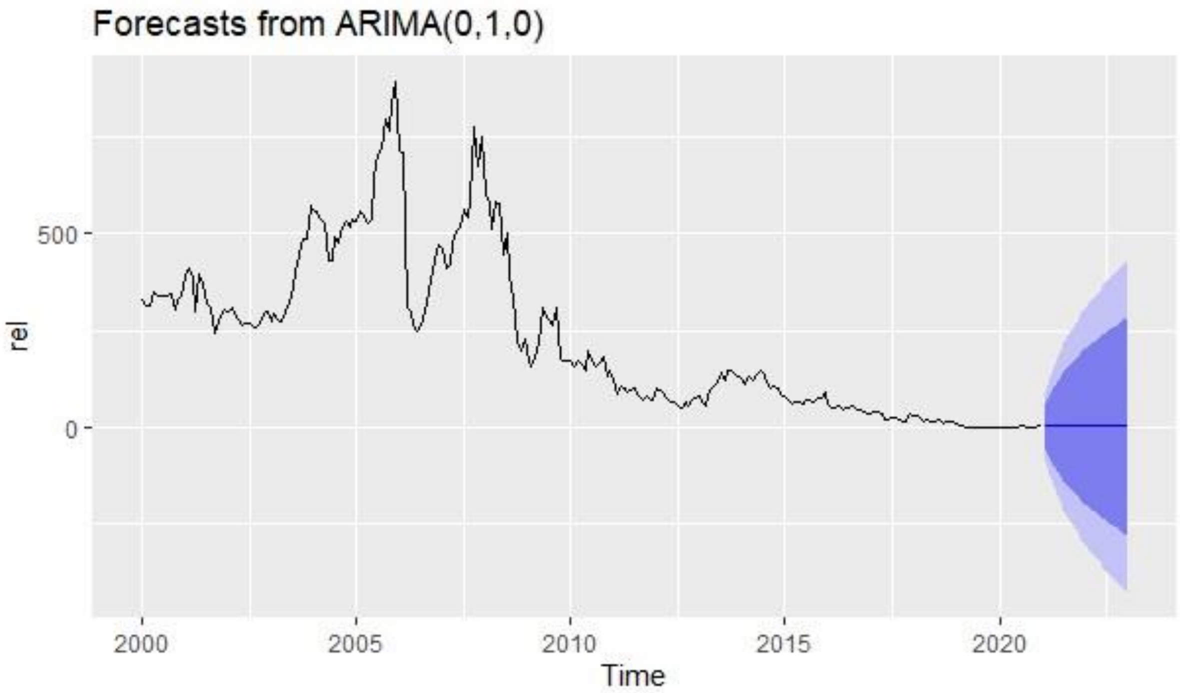
From the above table we can see that AIC and BIC is minimum for ARIMA(0,1,0). There we proceed with ARIMA(0,1,0). Therefore the considered series can be represented by a random walk.

```
> checkresiduals(relAR3)

        Ljung-Box test

data:  Residuals from ARIMA(0,1,0) Q* =
70.829, df = 24, p-value = 1.637e-06

Model df: 0.   Total lags used: 24
```



From the above plot we can say that the residuals are stationary in nature. Ljung-Box test also signifies that the residuals are stationary in nature. Thus the best fit for the model ARIMA(0,1,0) which is a random walk.

```
> autoplot(forecast(relAR3))
```



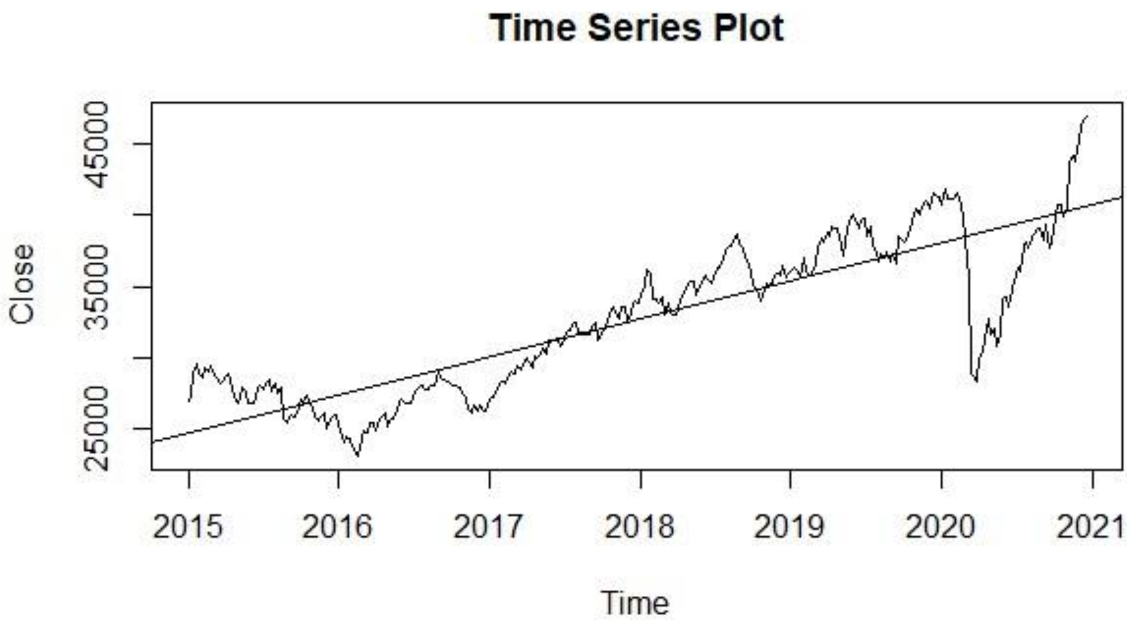
The above graph shows the forecast of the time series data. The forecasting is carried out after the last value.

S&P BSE SENSEX:

The data used for the modelling is S&P index of BSE SENSEX from 1st January 2015 to 17th December 2020. The data is weekly. The data consists of 312 data points. Variables in the data are such as opening and closing sensex for the week, high and low stock sensex for the week and volume. The data was collected from yahoo finance website. The link for the data is given below.

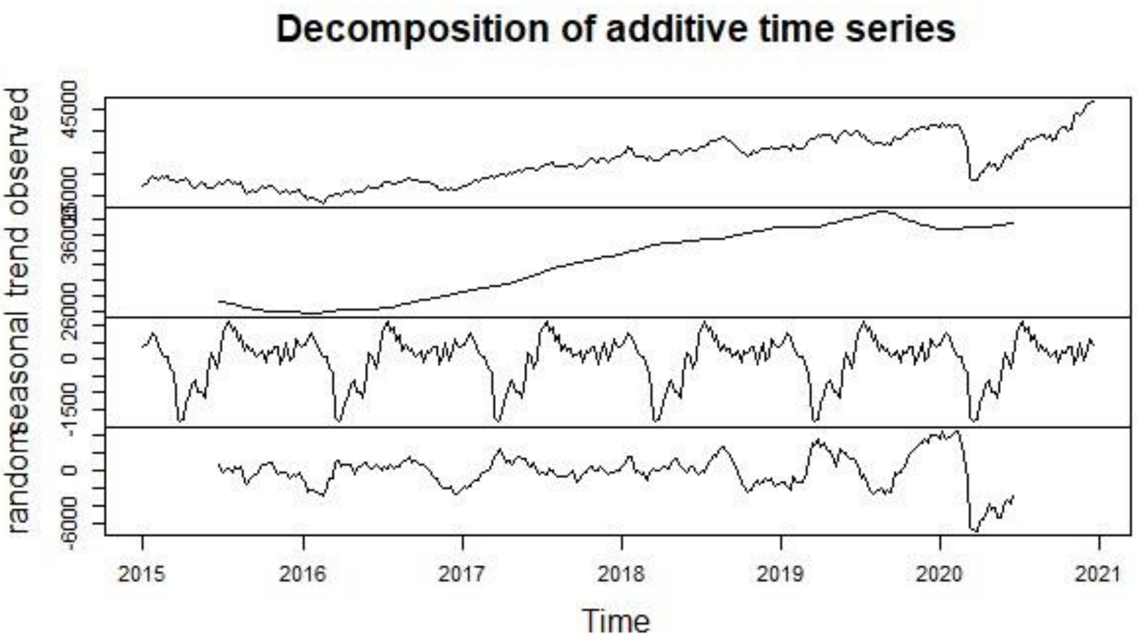
Link: https://in.finance.yahoo.com/quote/%5EBSESN/history?p=%5EBSESN Weekly data from January 1 2015 to December 17 2020.

```
> #Explonatory analysis
> bse=ts(R, frequency = 365.25/7, start=c(2015,1))
> plot.ts(bse, plot.type = "single", main="Time Series Plot")
> abline(reg=lm(bse~time(bse)))
```



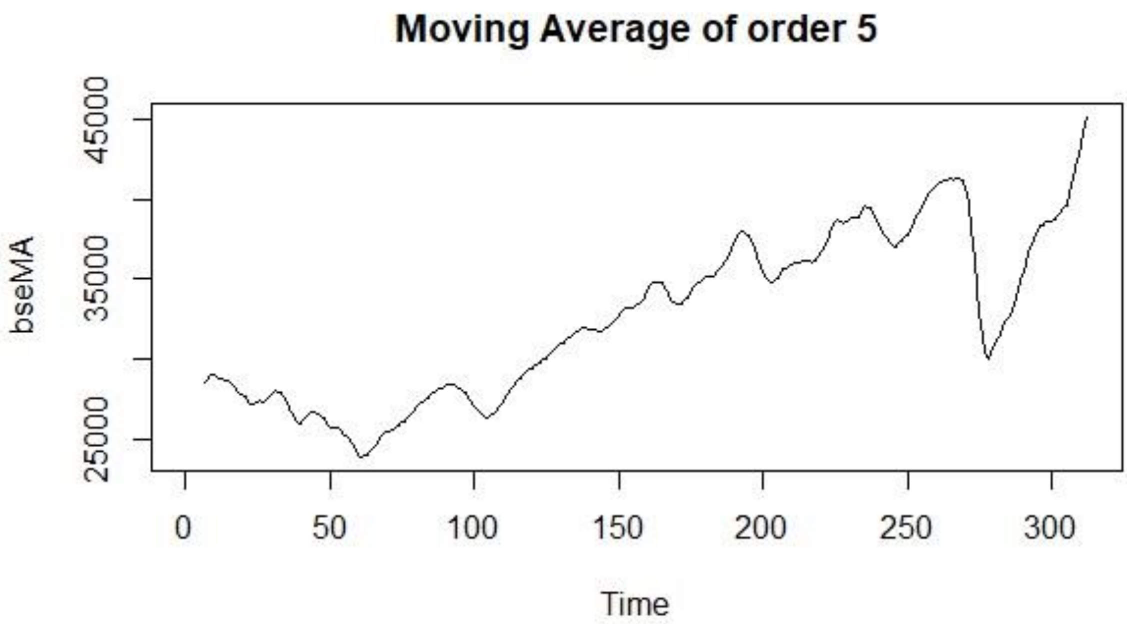
The time series plot shows an upward increasing trend of the sensex data. The fall in 2020 in sensex is due to imposition of lockdown in march 2020 and covid-19 situation.

```
> bse1=decompose(bse)
> plot(bse1)
```



The decomposition of the data shows the trend and seasonality in the data. The trend is increasing in nature where as seasonality is regular.

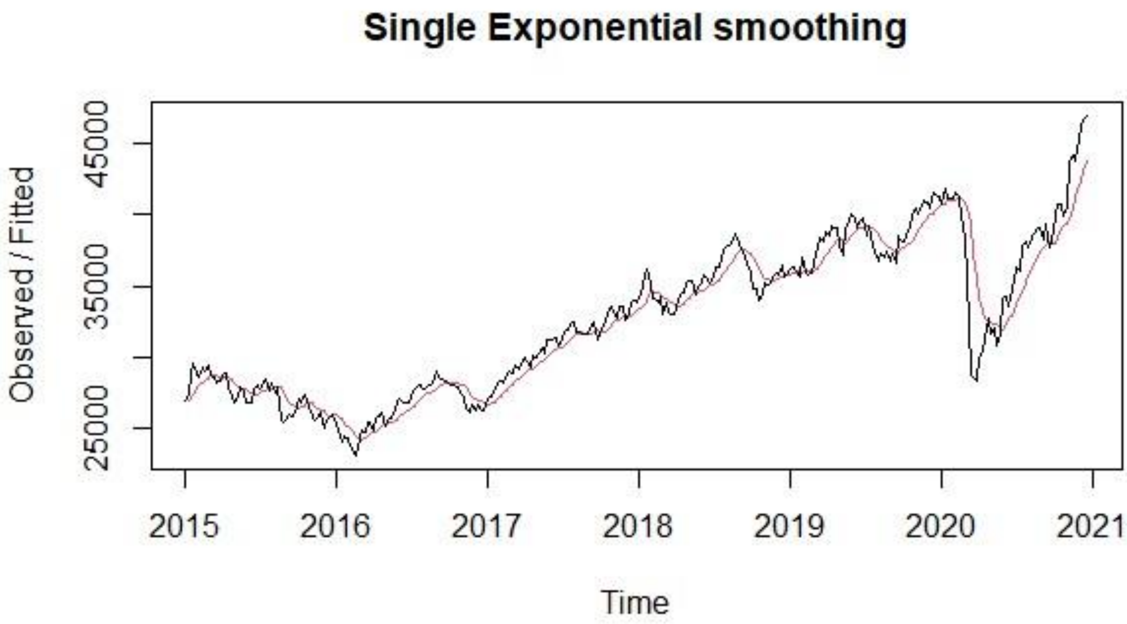
```
> library("TTR")
> #Moving Average of order 7
> bseMA=SMA(bse, n=7)
> plot.ts(bseMA, main="Moving Average of order 5")
```



Moving average of order 5 shows the smoothened time series data of sensex.

```
> #Holt Winter
```

```
> bseHW=HoltWinters(bse, alpha=0.2, beta=FALSE, gamma=FALSE)
> plot(bseHW, main="Single Exponential smoothing")
```
Alpha=0.2



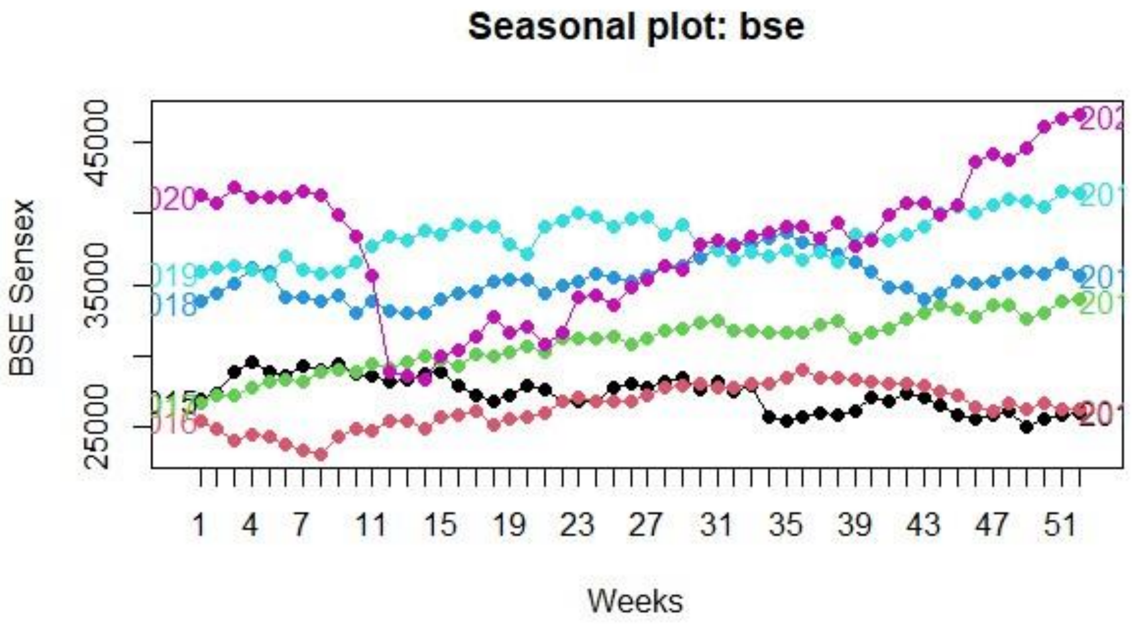
Single Exponential smoothing is not an appropriate fit for the data since there are some areas untouched by the smoothing curve. Therefore we carry out further analysis for the same.

```
> bseHW1=HoltWinters(bse, alpha=0.2, beta=0.3, gamma=FALSE)
> plot(bseHW1, main="Double Exponential smoothing")
```

Alpha=0.3 Beta=0.2

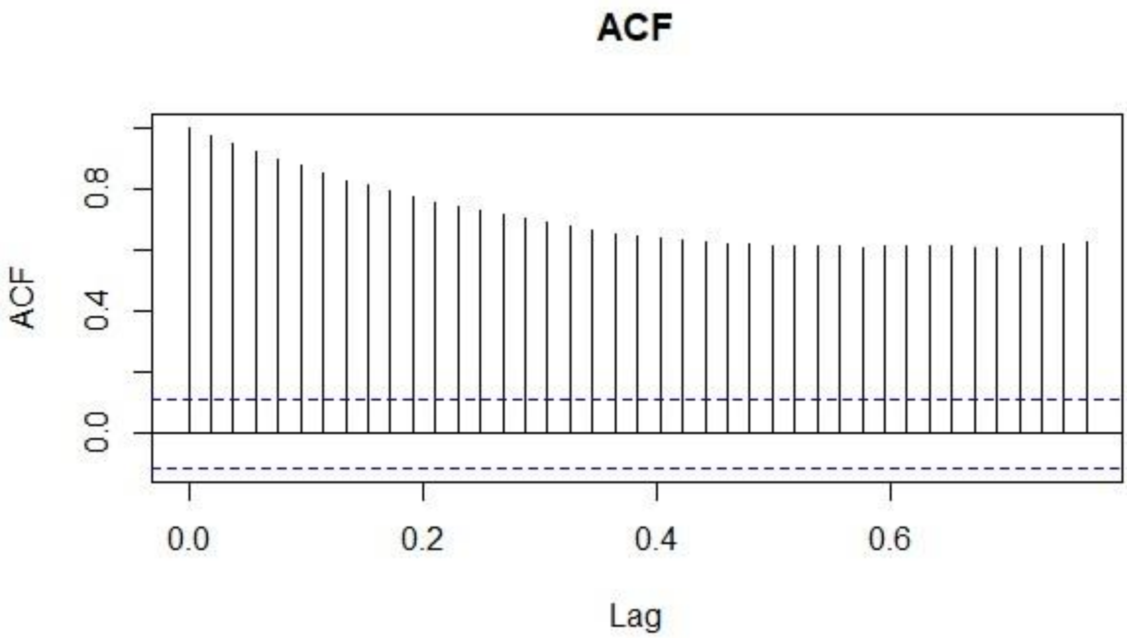
Double exponential smoothing is not so conclusive either.

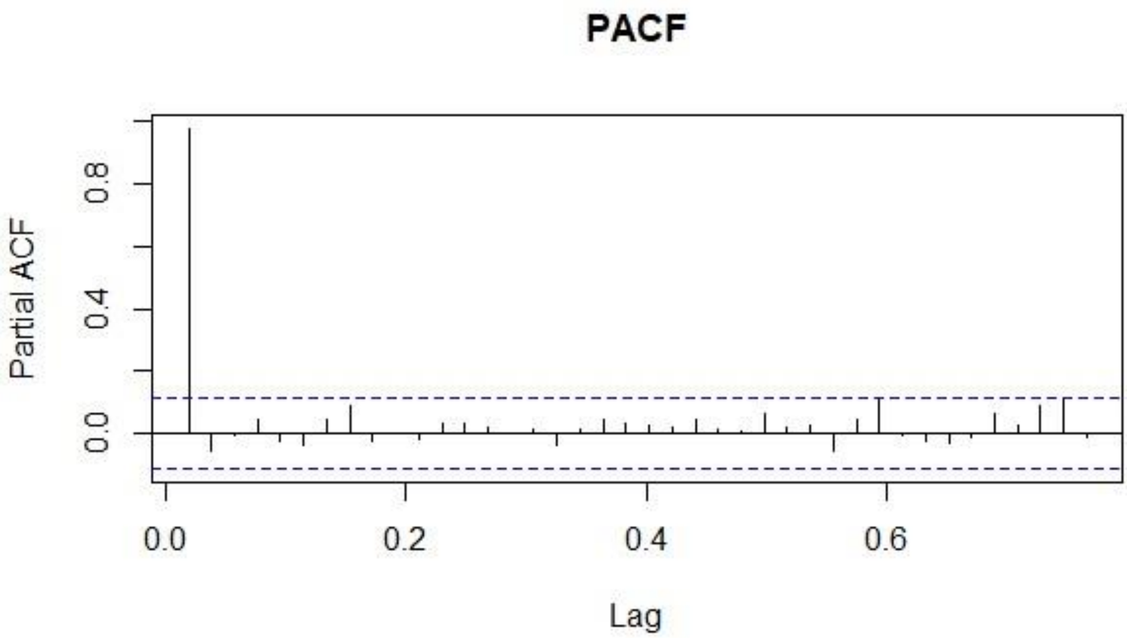
```
> seasonplot(bse, year.labels.left = TRUE, year.labels = TRUE, col = 1:20, pch=19, yl
ab = "BSE Sensex ", xlab="Weeks")
```




```
> #Univariate analysis
> acf(bse, lag=40, main="ACF")
```

> pacf(bse, lag=40, main="PACF")

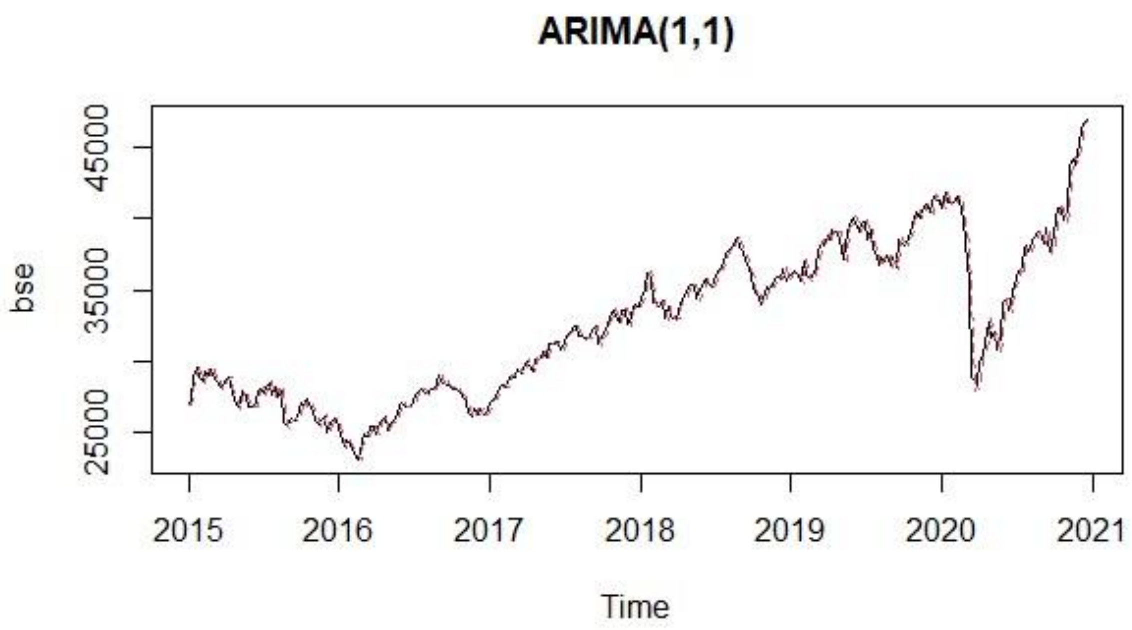




ACF and PACF aren’t conclusive since pattern cannot be observed. It’s quite difficult to draw conclusions based on ACF and PACF. The major determining factor for the data would be season since we observe wide variety of seasonal effects in seasonal plot. Therefore there are chances that the appropriate fit for the data would be the one in which seasons are involved. We carry out further analysis to check the same.

Fitting of ARIMA(1,1,1):

```
> #Fitting ARIMA(1,1,1) model
> bseARIMA=arima(bse, order=c(1,1,1))
> print(bseARIMA)

Call: arima(x = bse, order = c(1,
1, 1))

Coefficients:
         ar1      ma1
      0.5001  -0.3751
s.e.  0.3160   0.3385

sigma^2 estimated as 620157:  log likelihood = -2515.32,  aic = 5036.64
> ts.plot(bse, main="ARIMA(1,1)")
> bseARIMAfit=bse-residuals(bseARIMA)
> points(bseARIMAfit, type="l", col=2, lty=2)
```



Although fitted ARIMA(1,1,1) model shows quite similarities with the actual data, AIC and BIC would be the determining factor for the selection of the model.

Fitting SARIMA(0,1,1)(0,0,1):

```
> #fitting SARIMA(0,1,1)(0,0,1)[52]
> library("sarima")
> bseSARIMA=arima(bse, order=c(0,1,1), seasonal=c(0,0,1))
> print(bseSARIMA)
```

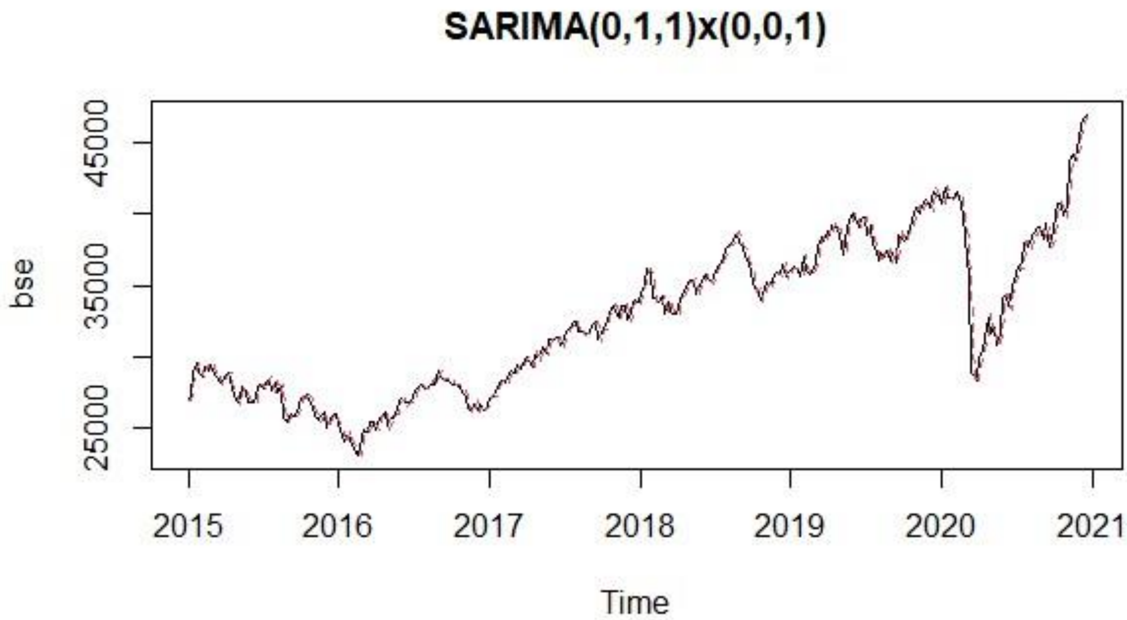
Call: arima(x = bse, order = c(0, 1, 1), seasonal = c(0,
0, 1))

Coefficients:

	ma1	sma1
	0.1159	-0.1597
s.e.	0.0527	0.0693

sigma^2 estimated as 610338: log likelihood = -2513.5, aic = 5033.01

```
> ts.plot(bse, main="SARIMA(0,1,1)x(0,0,1)")
> bseSARIMAfit=bse-residuals(bseSARIMA)
> points(bseSARIMAfit, type="l", col=2, lty=2)
```



Although the seasonal factor does its part in estimating the model but still it isn't the better fit for the data. Hence we move forward and check whether the difference factor for the SARIMA model with difference as 52

Fitting of SARIMA(0,1,1)*(0,0,1)[52]:

```
> bseSARIMA1=sarima(log(bse)~0| sar(52,1,c(-0.1)) + i(1) +s(365.25/7), ss.method =
"base")
> print(bseSARIMA1)
```

Sarima model

Call: sarima(model = log(bse) ~ 0 | sar(52, 1, c(-0.1)) + i(1) + s(365.25/7), ss.method = "bas e")

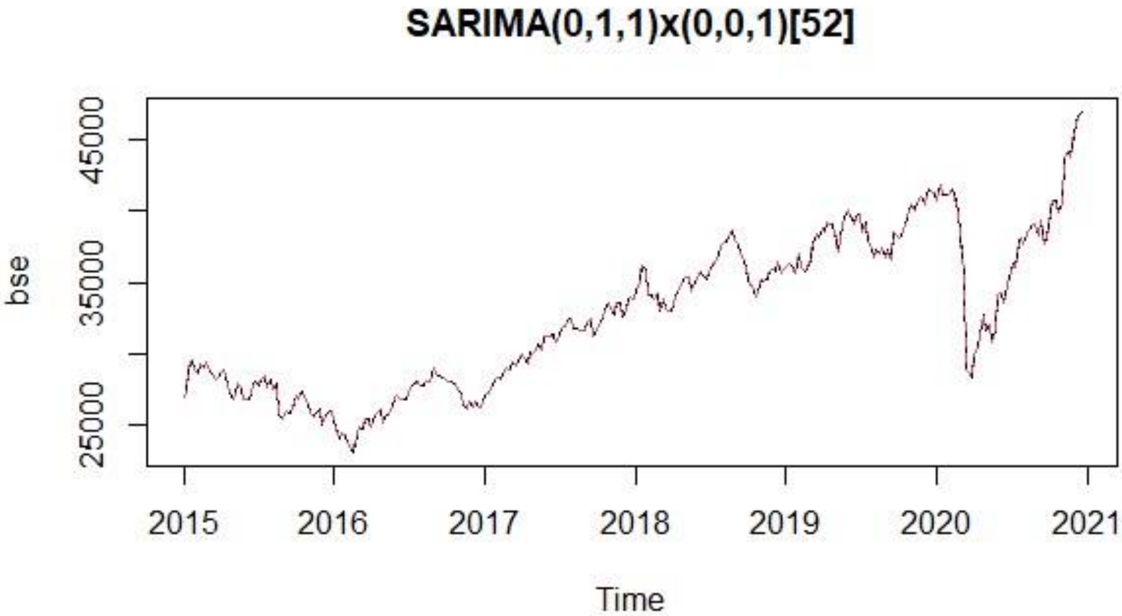
Unit root terms:
(1 - B)(1 + B + ... + B^51)

Coefficients:

```
        sar1
      0.2415
s.e.  0.0692
```

sigma^2 estimated as 0.0164: log likelihood = -111.64, aic = 227.28

```
> ts.plot(bse, main="SARIMA(0,1,1)x(0,0,1)[52]")
> bseSARIMAfit1=bse-residuals(bseSARIMA1)
> points(bseSARIMAfit1, type="l", col=2, lty=2)
```



We further check the AIC and BIC of the models and hence which is the appropriate fit for the sensex model.

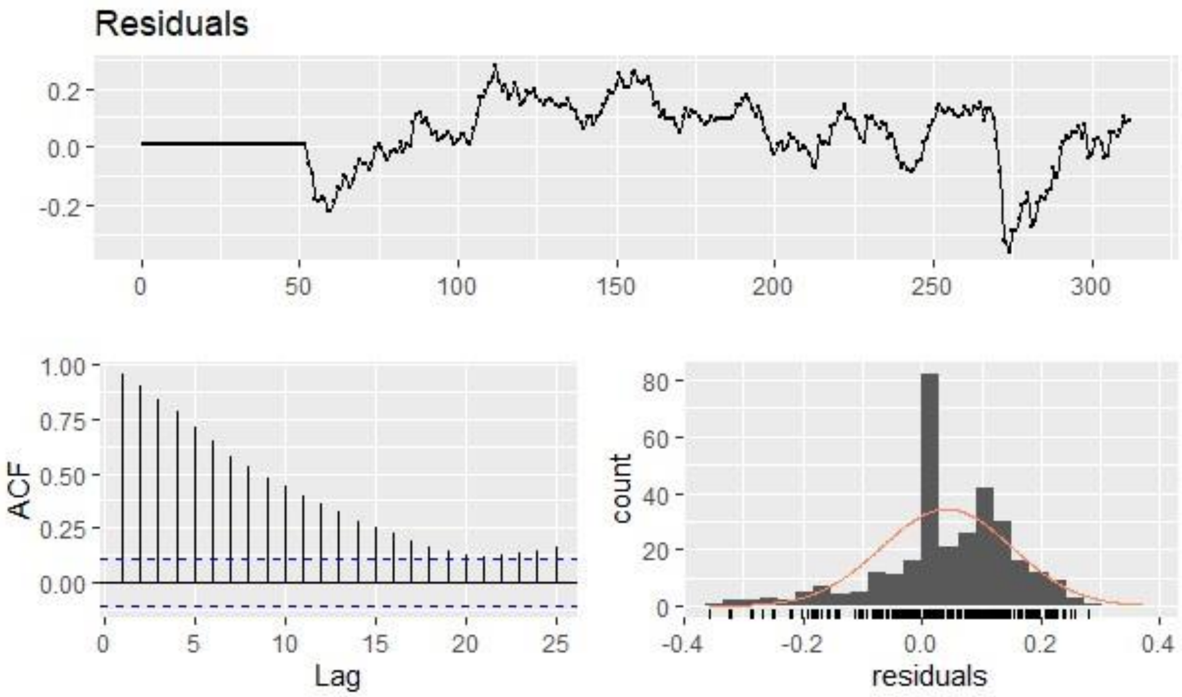
```
> #To check which model fits the best
> #AIC
> AIC(bseARIMA)  #AIC of ARIMA(1,1,1) model
[1] 5036.637
> AIC(bseSARIMA)  #AIC of SARIMA(0,1,1)*(0,0,1) model [1]
5033.007
> AIC(bseSARIMA1)  #AIC of ARIMA(0,1,1)*(0,0,1)[52] model
```

[1] 227.2842
> #bIC
> BIC(bseARIMA) #AIC of ARIMA(1,1,1) model
[1] 5047.856
> BIC(bseSARIMA) #AIC of SARIMA(0,1,1)*(0,0,1) model [1]
5044.226
> BIC(bseSARIMA1) #AIC of ARIMA(0,1,1)*(0,0,1)[52] model
[1] 234.4056

Model	AIC	BIC
ARIMA(1,1,1)	5036.637	5047.856
SARIMA(0,1,1)*(0,0,1)	5033.007	5044.226
SARIMA(0,1,1)*(0,0,1)[52]	227.2842	234.4056

The data on AIC and BIC of each of the models leads us to the conclusion that SARIMA(0,1,1)*(0,0,1)[52] is the appropriate fit for the model with seasonal factor and difference factor as 52. We further check the residual of the fitted model are stationary in nature.

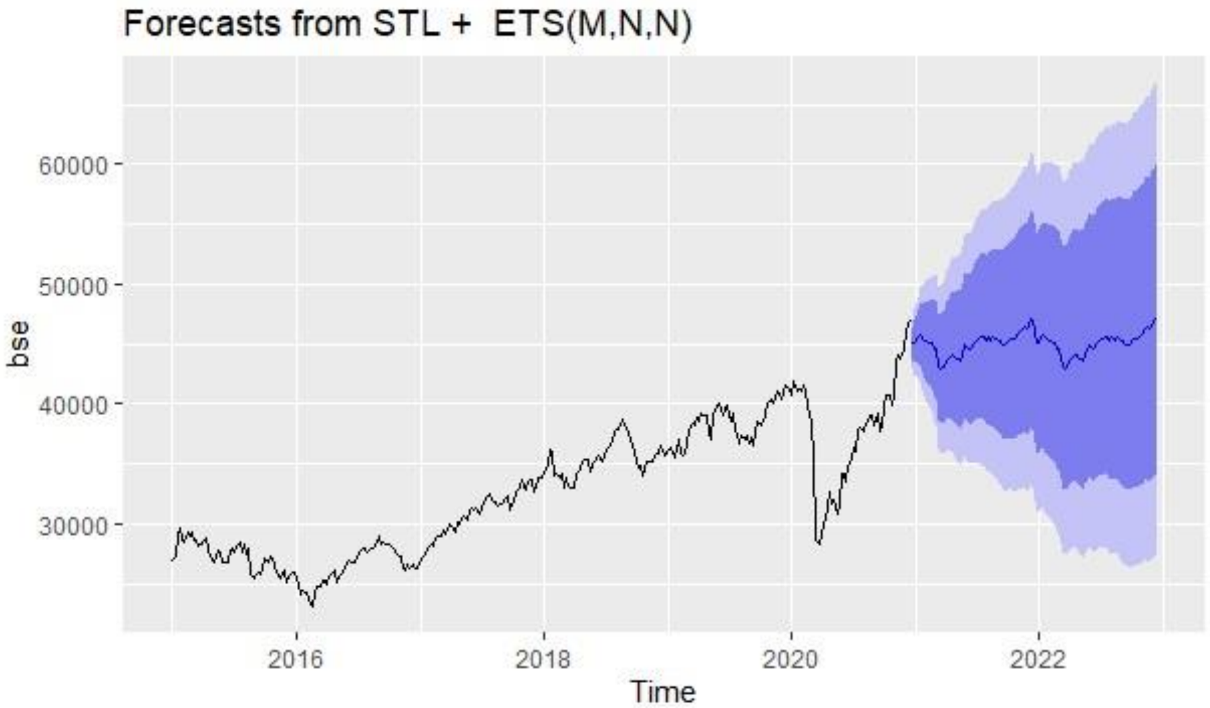
```
> checkresiduals(bseSARIMA1)

	Ljung-Box test

data:  Residuals
Q* = 1594.3, df = 9, p-value < 2.2e-16

Model df: 1.   Total lags used: 10
```



Ljung-Box test and the above graphs leads us to the conclusion that the residuals are stationary in nature. Hence we carry out further forecast of the time series.

> autoplot(forecast(bse))

The below given graph is the graph of forecast of the data for future observations. Hen ce the conclusion.



GARCH Fitting (Volatility Forecasting):

```
> #GArch fitting
> library(rmgarch)
> library(rugarch)
> ug_spec=ugarchspec()
> ug_spec
```

- The below given specifications are default specifications for the GARCH model .

```
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model            : sGARCH(1,1)
Variance Targeting     : FALSE

Conditional Mean Dynamics
-----------------------------------
Mean Model             : ARFIMA(1,0,1)
Include Mean           : TRUE
GARCH-in-Mean                    : FALSE

Conditional Distribution
-----------------------------------
Distribution      :  norm
Includes Skew :  FALSE
Includes Shape :  FALSE
Includes Lambda          :  FALSE
```

- We modify the specifications to ARMA(1,0) that is AR(1).

```
> ug_spec=ugarchspec(mean.model = list(armaOrder=c(1,0)))
```

```
> ug_spec

*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
------------------------------------
GARCH Model           : sGARCH(1,1)
Variance Targeting    : FALSE

Conditional Mean Dynamics
------------------------------------
Mean Model            : ARFIMA(1,0,0)
Include Mean          : TRUE
GARCH-in-Mean         : FALSE

Conditional Distribution
------------------------------------
Distribution       :  norm
Includes Skew :  FALSE
Includes Shape :  FALSE
Includes Lambda       :  FALSE
```

We can see that the conditional distribution for residuals is the normal distribution. The mean model changes from ARMA(1,1) to ARMA(1,0) or AR(1).

Model Estimation:

```
> #model estimation
> ugfit=ugarchfit(spec = ug_spec, data=bse)
> ugfit

*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model : sGARCH(1,1)
Mean Model      : ARFIMA(1,0,0)
```

```
Distribution      : norm

Optimal Parameters
------------------------------------
       Estimate    Std.  Error    t  value  Pr(>|t|)  mu
2.6912e+04   1.1988e+03   22.4487  0.000000  ar1
1.0000e+00  6.1450e-03 162.7365 0.000000 omega
2.8673e+04  2.1280e+04   1.3474 0.177848 alpha1
2.0060e-01   5.7040e-02    3.5169  0.000437  beta1
7.7089e-01  7.5106e-02  10.2640 0.000000

Robust Standard Errors:
       Estimate    Std.  Error      t  value  Pr(>|t|)  mu
2.6912e+04   6.7092e+02   40.11296  0.000000  ar1
1.0000e+00  8.6260e-03 115.92342 0.000000 omega
2.8673e+04  3.4198e+04   0.83844 0.401783 alpha1
2.0060e-01   1.4507e-01    1.38276  0.166740  beta1
7.7089e-01  1.5917e-01   4.84334 0.000001

LogLikelihood : -2484.401

Information Criteria
------------------------------------

Akaike       15.958
Bayes        16.018
Shibata      15.957
Hannan-Quinn 15.982

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
statistic p-value
Lag[1]                  0.3771  0.5391
Lag[2*(p+q)+(p+q)-1][2]    0.4842  0.9679
Lag[4*(p+q)+(p+q)-1][5]    1.5214  0.8414 d.o.f=1
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
statistic p-value
Lag[1]                  0.6933  0.4050
```

Lag[2*(p+q)+(p+q)-1][5]   1.3875 0.7675
Lag[4*(p+q)+(p+q)-1][9]   2.7821 0.7943 d.o.f=2

Weighted ARCH LM Tests

	Statistic	Shape	Scale	P-Value
ARCH Lag[3]	0.07441	0.500	2.000	0.7850
ARCH Lag[5]	1.03667	1.440	1.667	0.7221
ARCH Lag[7]	1.79369	2.315	1.543	0.7609

Nyblom stability test
------------------------------------ Joint
Statistic: 1.0341
Individual Statistics:
mu   0.1516 ar1
0.2097 omega
0.4336 alpha1
0.3948
beta1 0.3277

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic: 1.28 1.47 1.88
Individual Statistic: 0.35 0.47 0.75

Sign Bias Test
----
t-value   prob sig
Sign Bias   0.4158 0.67787
Negative Sign Bias 2.2162 0.02741 **
Positive Sign Bias 1.1448 0.25319
Joint Effect   8.9387 0.03012 **

Adjusted Pearson Goodness-of-Fit Test:
----
group statistic p-value(g-1) 1
20   30.18   0.04956
2   30   40.12   0.08207
3   40   42.10   0.33816
4   50   72.62   0.01584

Elapsed time : 0.3155301

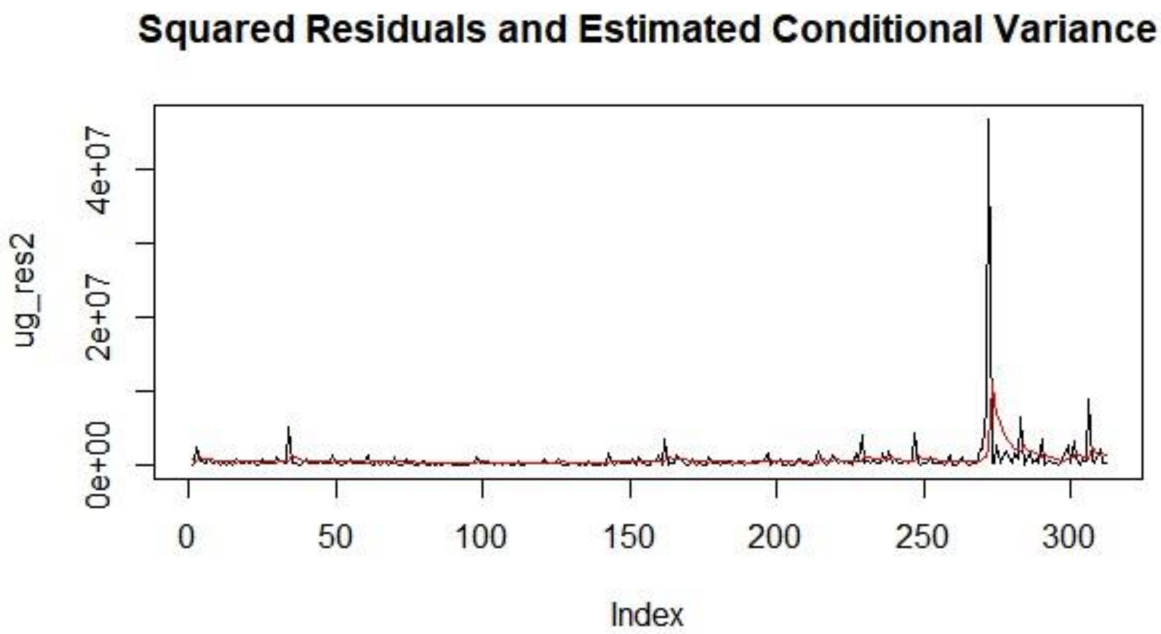
```
> names(ugfit@model)
 [1] "modelinc"   "modeldesc"  "modeldata"  "pars"       "start.pars" "fixed.pars"
 [7] "maxOrder"   "pos.matrix" "fmodel"     "pidx"       "n.start"
> names(ugfit@fit)
 [1] "hessian"         "cvar"            "var"             "sigma"
 [5] "condH"           "z"               "LLH"             "log.likelihoods"
 [9] "residuals"       "coef"            "robust.cvar"     "A"
[13] "B"               "scores"          "se.coef"         "tval"
[17] "matcoef"         "robust.se.coef"  "robust.tval"     "robust.matcoef"
[21] "fitted.values"   "convergence"     "kappa"           "persistence"
[25] "timer"           "ipars"           "solver"
> ugfit@fit$coef     #estimated coefficients
```

Estimated Coefficients:

mu	ar1	omega	alpha1	beta1
2.691246e+04	**1.000000e+00**	**2.867310e+04**	**2.006008e-01**	**7.708902e-01**

Here we can see the **estimated coefficients** for mu, ar, omega, alpha and beta.

```
> ug_var=ugfit@fit$var  #variances
> ug_res2=(ugfit@fit$residuals)^2   #squared residuals
plot(ug_res2, type="l", main="Squared Residuals and Estimated Conditional Variance
")
> lines(ug_var, col="red")
```



The black line is the line of squared residuals and the red line is the conditional varian ces. We do observe a typical GARCH pattern where the conditional variances increase s as the squared residual increases. This is due to the volatility. This can be observed b etween 250 and 300.

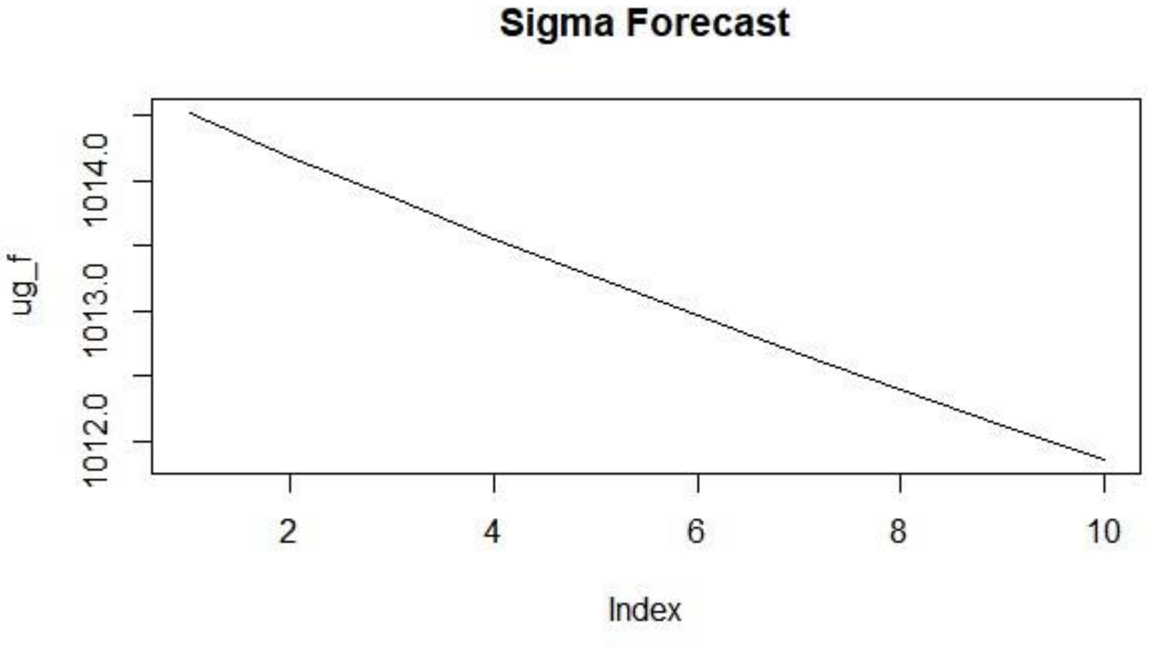
Model Forecasting:

```
> #model forecasting
> ugfor=ugarchforecast(ugfit, n.ahead = 10)
> ugfor

*------------------------------------*
*       GARCH Model Forecast         *
*------------------------------------*
Model: sGARCH
Horizon: 10
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=1970-11-09 05:30:00]:
     Series Sigma
T+1   46961  1015
T+2   46961  1014
T+3   46961  1014
T+4   46961  1014
```

T+5   46961 1013
T+6   46961 1013
T+7   46961 1013
T+8   46961 1012
T+9   46961 1012
T+10 46961 1012

Above we can see the forecasted values of the volatility from GARCH model from 10 timepoints ahead. Apart from mean variance turns out to be a major factor in the forecast. Therefore we check the nature of the variance by plotting the sigma forecasted values.

```
> ug_f=ugfor@forecast$sigmaFor
> plot(ug_f, type="l")
```



We do observe that for the next 10 time points ahead we are forecasting decreasing volatility.

Indian Foreign Trade (Export):

The dataset consists of India's International Trade Monthly data (from April 1990 to July 2012) consisting of 364 observations. The headers in the excel file are as follows:

1) Year
2) Month
3) Exports
4) Imports
5) Trade Balance

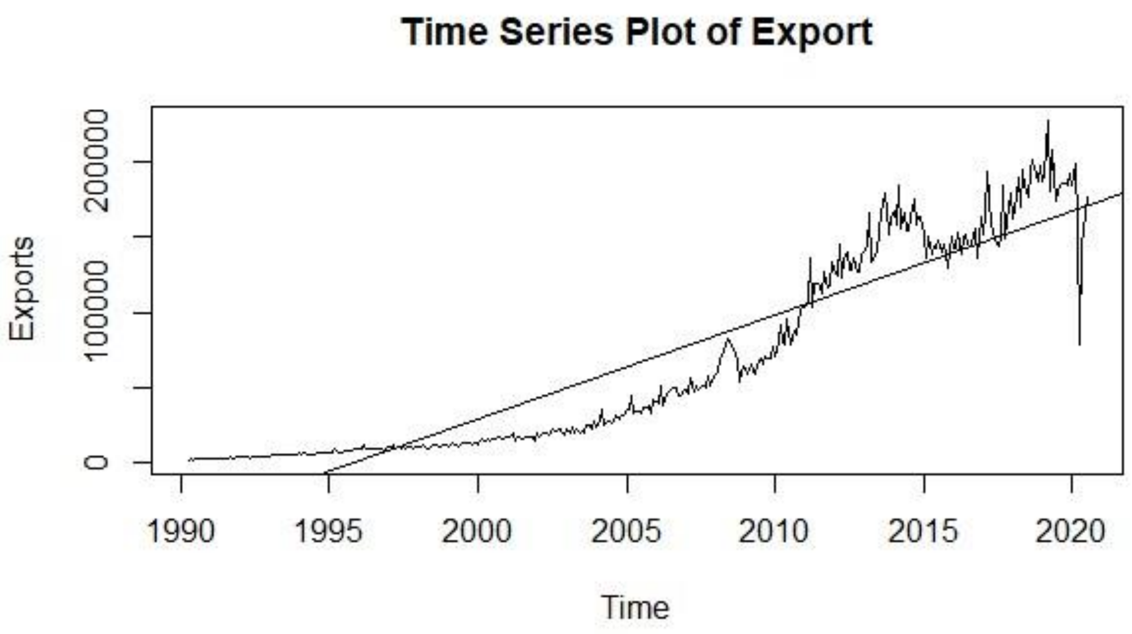
NOTE : For the purpose of this project we have considered 1) Year 2) Month 3) Exports.

DATASET LINK : https://dbie.rbi.org.in/DBIE/dbie.rbi?site=statistics

The data is from April 1990 to July 2020. The variable taken into consideration is export. The data is monthly with April to March as one cycle.

We carry out analysis considering the variable Exports.

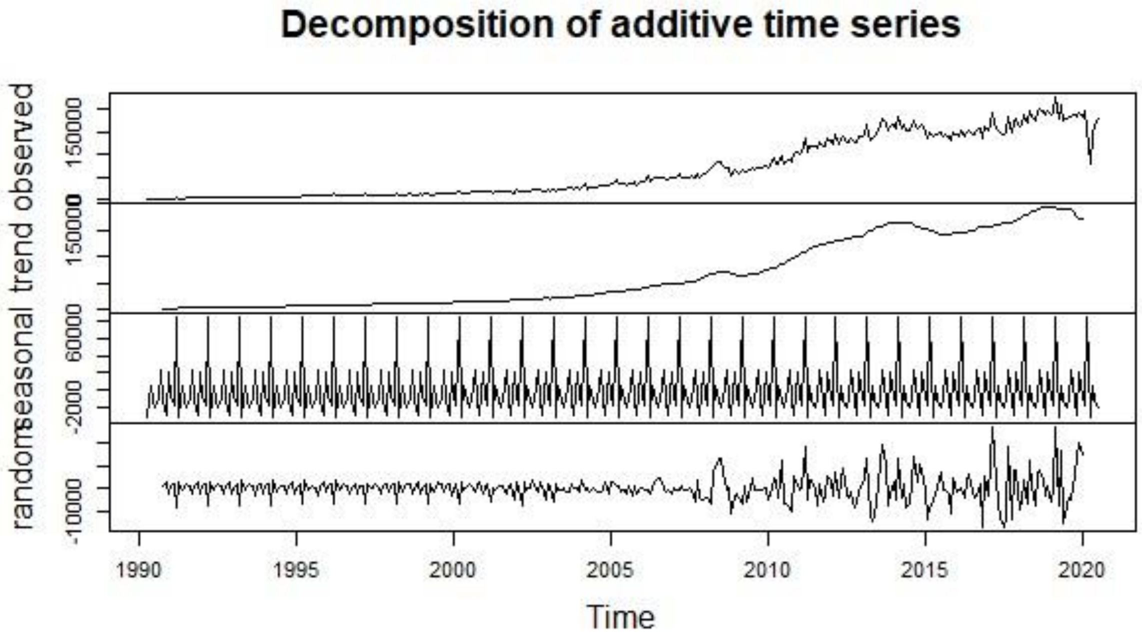
```
> # Explonatory Analysis
> export=ts(R, frequency = 12, start=c(1990,4))
> plot.ts(export, plot.type = "single", main="Time Series Plot of Export")
> abline(reg=lm(export~time(export)))
```



The time series plot along with regression line shows an upward trend in the data. The dramatic fall after 2020 is in march of 2020 because of covid-19 situation and imposition of lockdown across the world along with India. It affected the export trade of India.

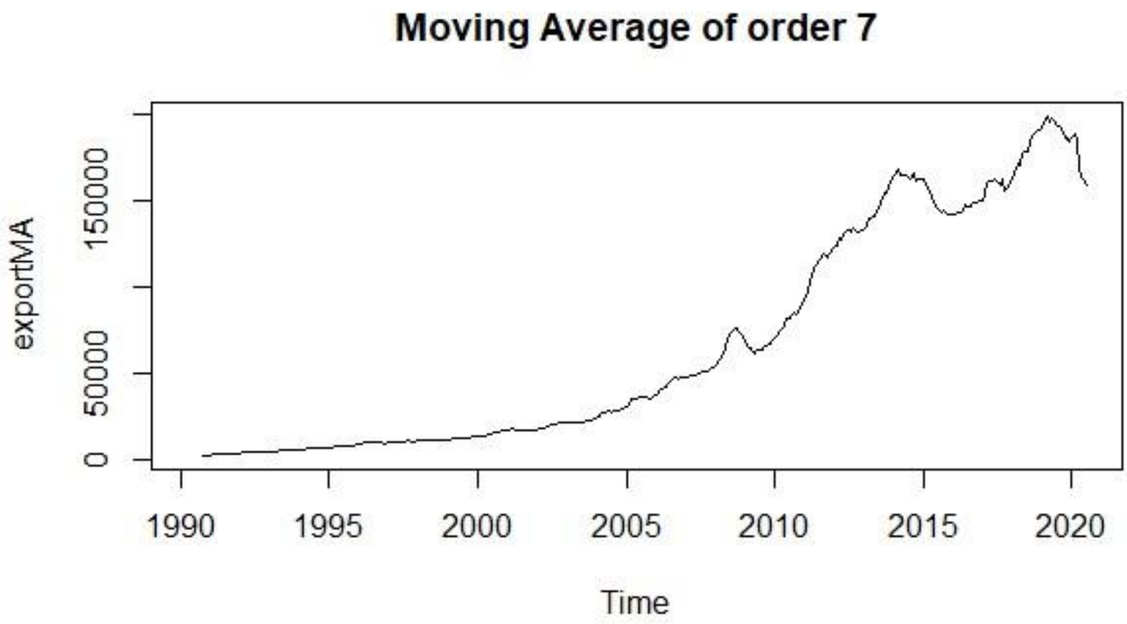
```
> export1=decompose(export)
> plot(export1)
```

Time series plot shows there is an increasing trend in time series data. We carry out fu rther analysis to find out the best fit for the given model.



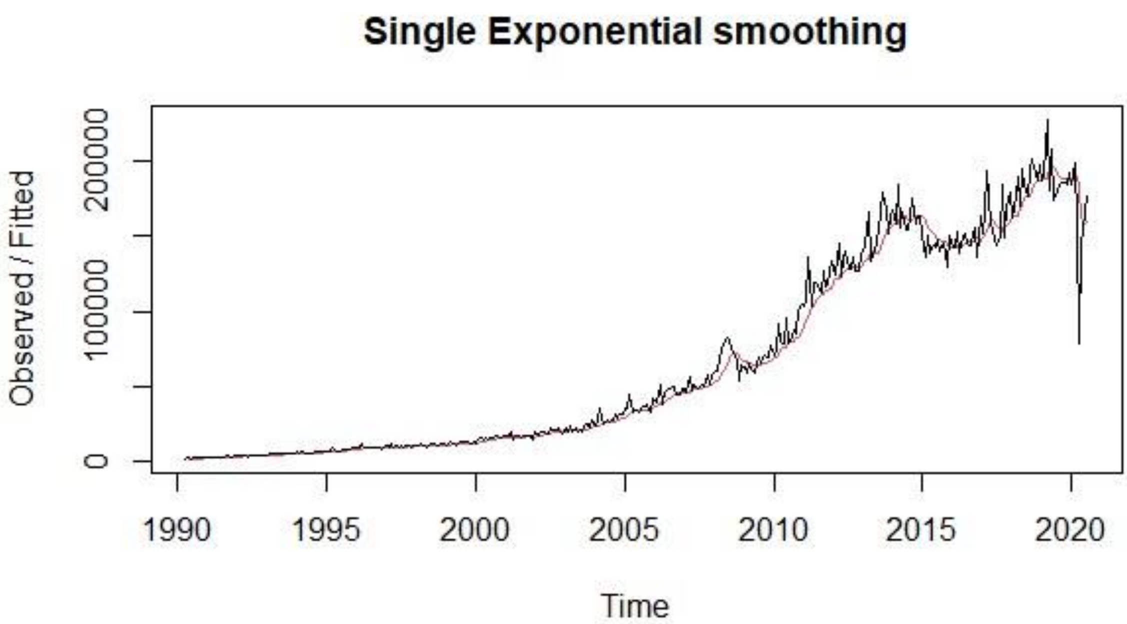
Here we can see that there exists trend and seasonality in the data of Exports.

```
> #Moving Average of order 7
> exportMA=SMA(export, n=7)
> plot.ts(exportMA, main="Moving Average of order 7")
```



The moving average plot of order 7 is smoothened time series plot of the export data.

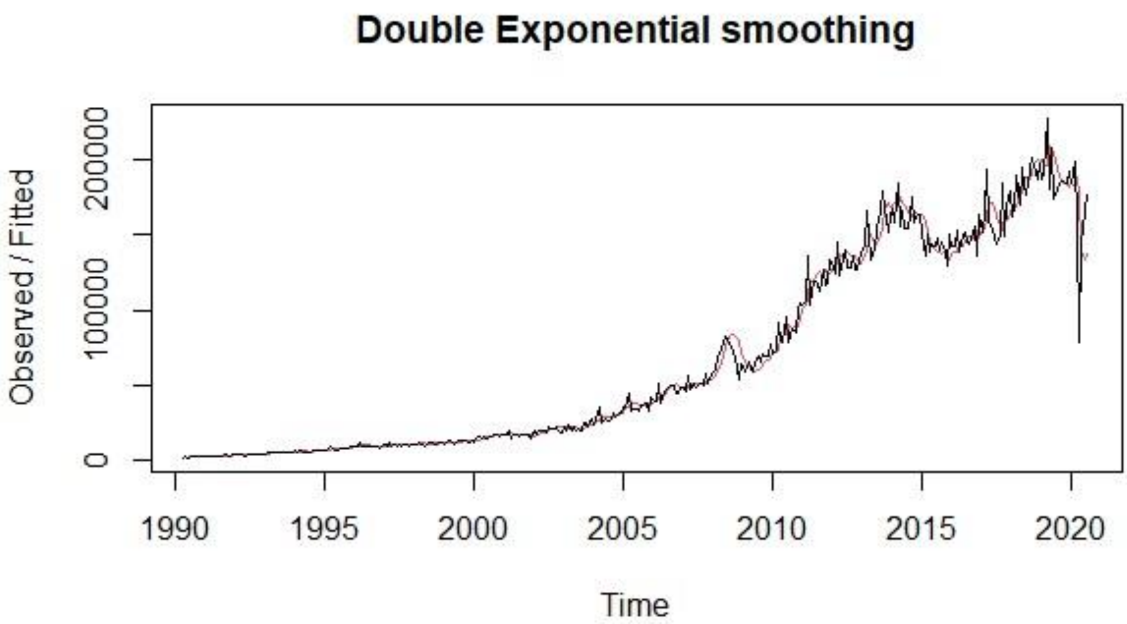
```
> #Holt Winter
> exportHW=HoltWinters(export, alpha=0.2, beta=FALSE, gamma=FALSE)
> plot(exportHW, main="Single Exponential smoothing")
```



Single exponential smoothing is not the appropriate fit for the data. Although it estimated the data upto some extent. > #Double Exponential smoothing

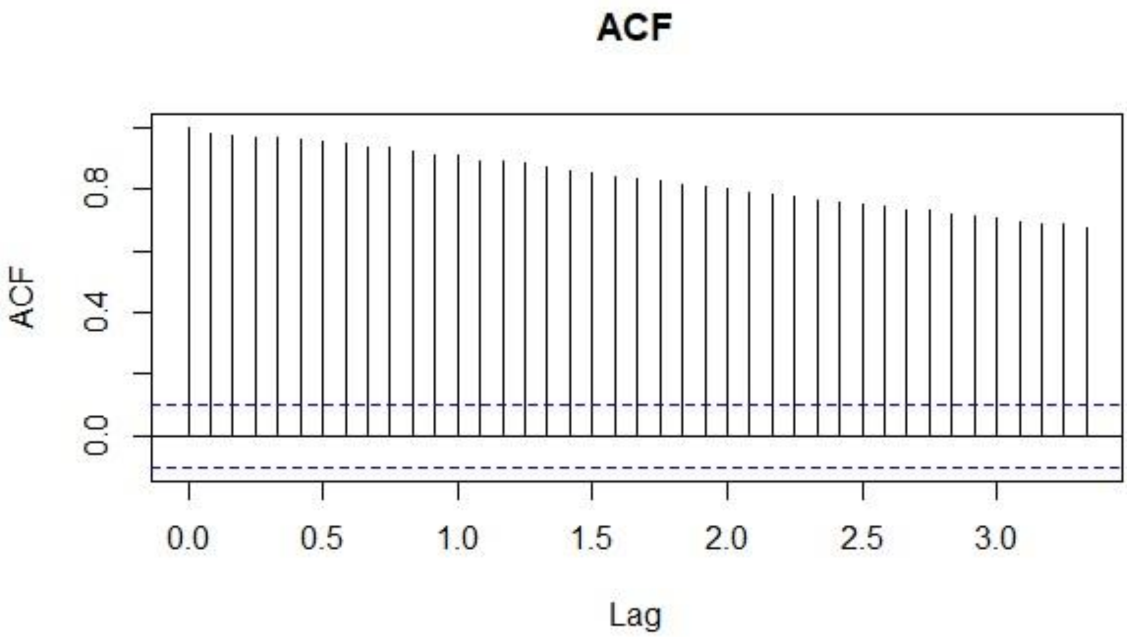
```
> exportHW1=HoltWinters(export, alpha=0.3, beta=0.2, gamma=FALSE)
```

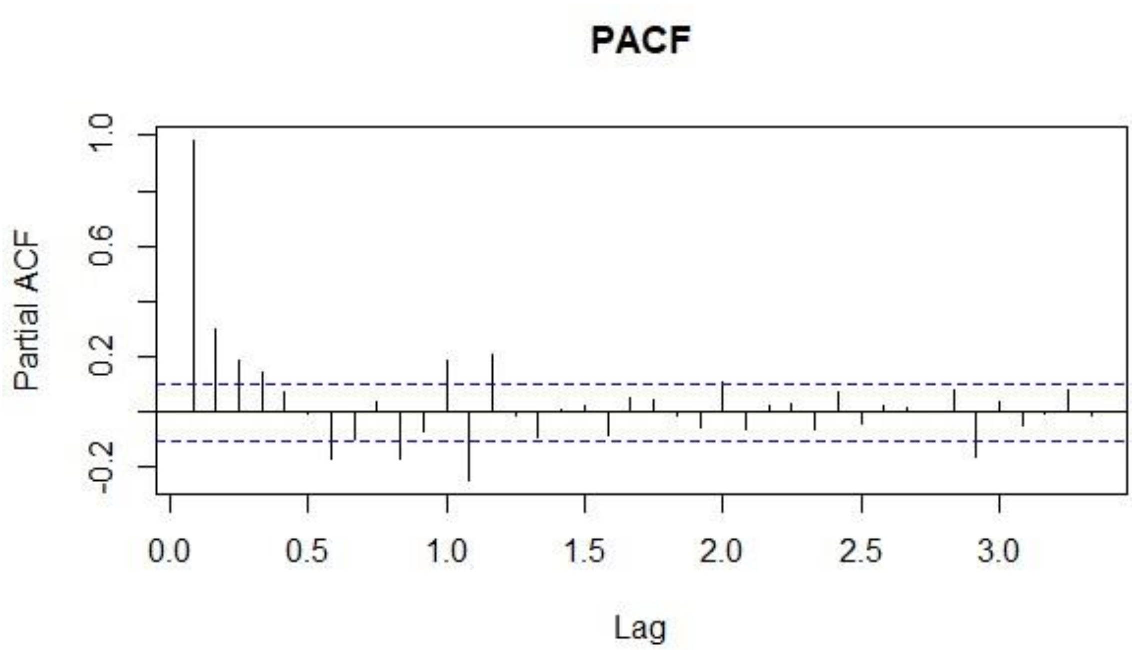
```
> plot(exportHW1, main="Double Exponential smoothing")
```



Neither is double exponential smoothing appropriate fit for the data. We carry out further analysis to check the appropriate fit for the data.

```
> #Univariate analysis
> acf(export, lag=40, main="ACF")
> pacf(export, lag=40, main="PACF")
```

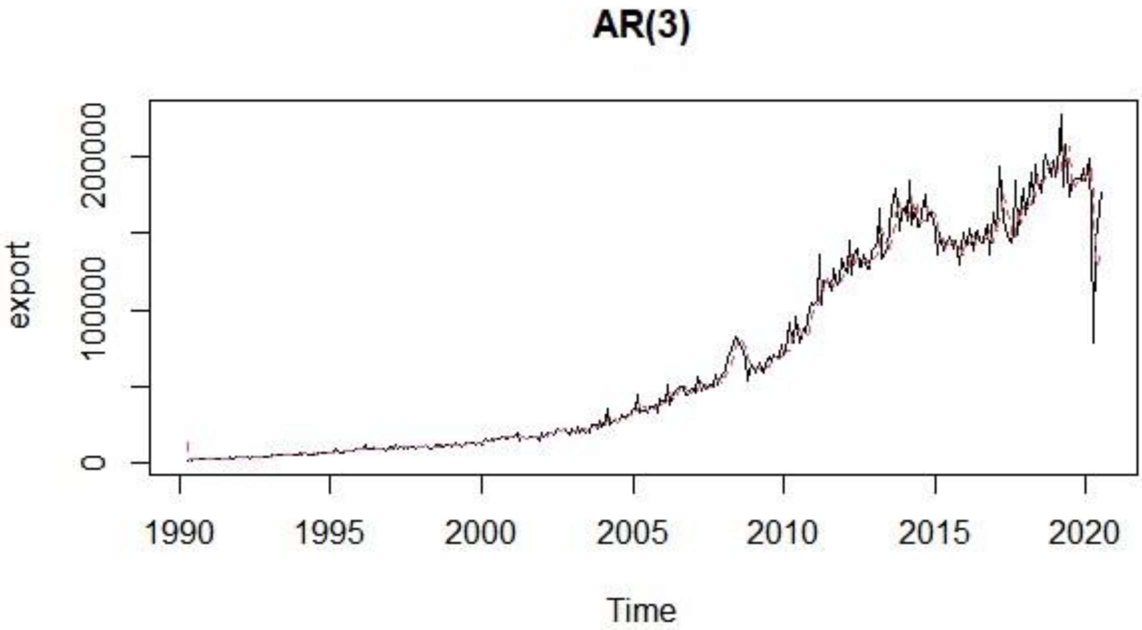




Although ACF might not be that conclusive but PACF gives an idea that there are chances that the good fit for the model would be AR(3). Therefore we carry out further analysis to find out the best fit for the model.

Fitting AR(3) model.:

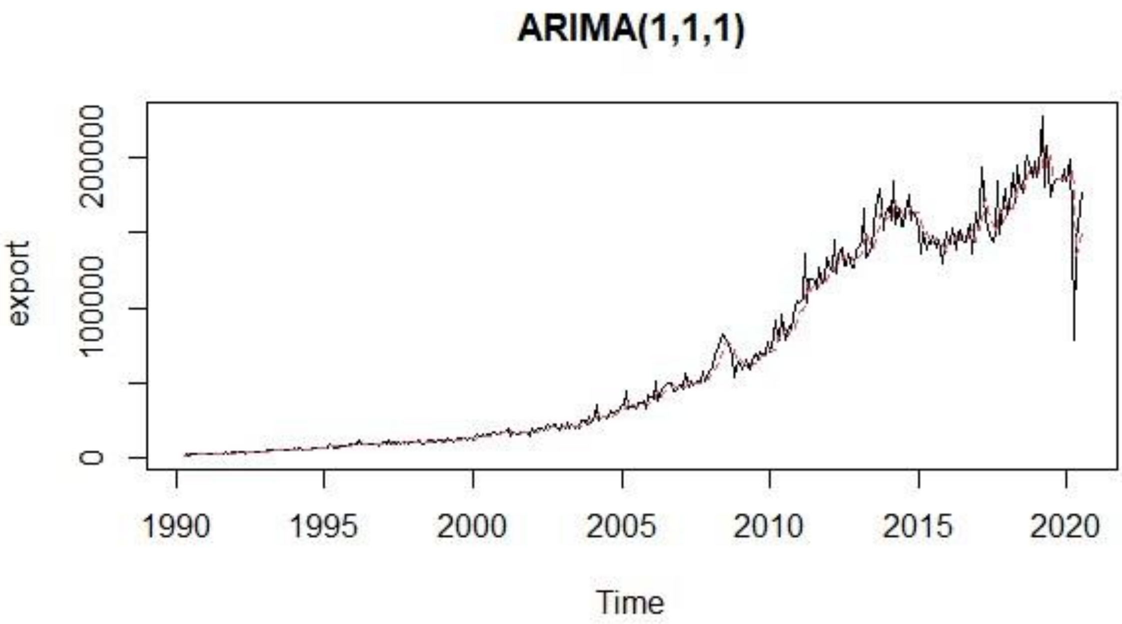
```
> #Fitting AR(3) model
> expAR=arima(export, order=c(3,0,0))
> print(expAR)

Call:
arima(x = export, order = c(3, 0, 0))

Coefficients:
         ar1     ar2     ar3  intercept
      0.5005  0.2390  0.2559   92440.29
s.e.  0.0509  0.0557  0.0511   66783.10

sigma^2 estimated as 94447501:  log likelihood = -3860.87,  aic = 7731.75
> ts.plot(export, main="AR(3)")
> expARfit=export-residuals(expAR)
> points(expARfit, type="l", col=2, lty=2)
```



Although PACF is quite conclusive in the selection of the model but in actual context AIC and BIC would help us to decide which model is the best fit to the data.

Fitting of ARIMA(1,1,1) model:

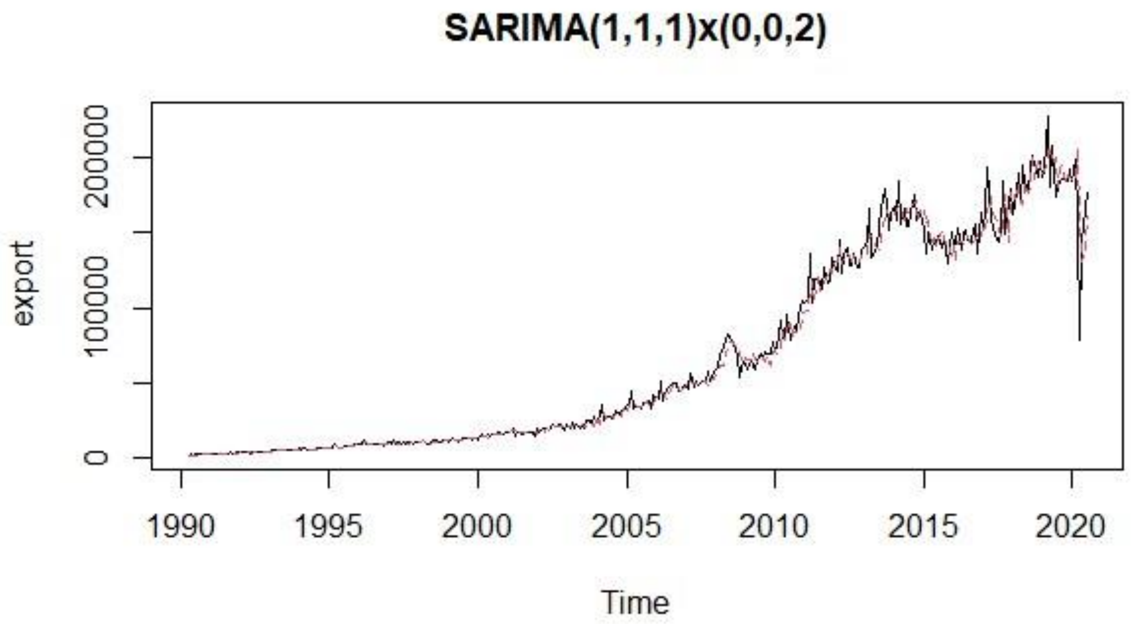
```
> #Fitting ARIMA(1,1,1) model
> expARIMA=arima(export, order=c(1,1,1))
> print(expARIMA)

Call:
arima(x = export, order = c(1, 1, 1))

Coefficients:
         ar1      ma1
      0.0920  -0.6468
s.e.  0.0809   0.0577

sigma^2 estimated as 91767901:  log likelihood = -3843.05,  aic = 7692.1
> ts.plot(export, main="ARIMA(1,1,1)")
> expARIMAfit=export-residuals(expARIMA)
> points(expARIMAfit, type="l", col=2, lty=2)
```



The same condition prevails in ARIMA(1,1,1) also. Although It fits better than AR(3) but it still isn't a good fit to the data. This leads us to the assumption that maybe the data contains seasonal components which act as determining factor in the fitting of the model. Therefore we move forward with seasonal ARIMA model.

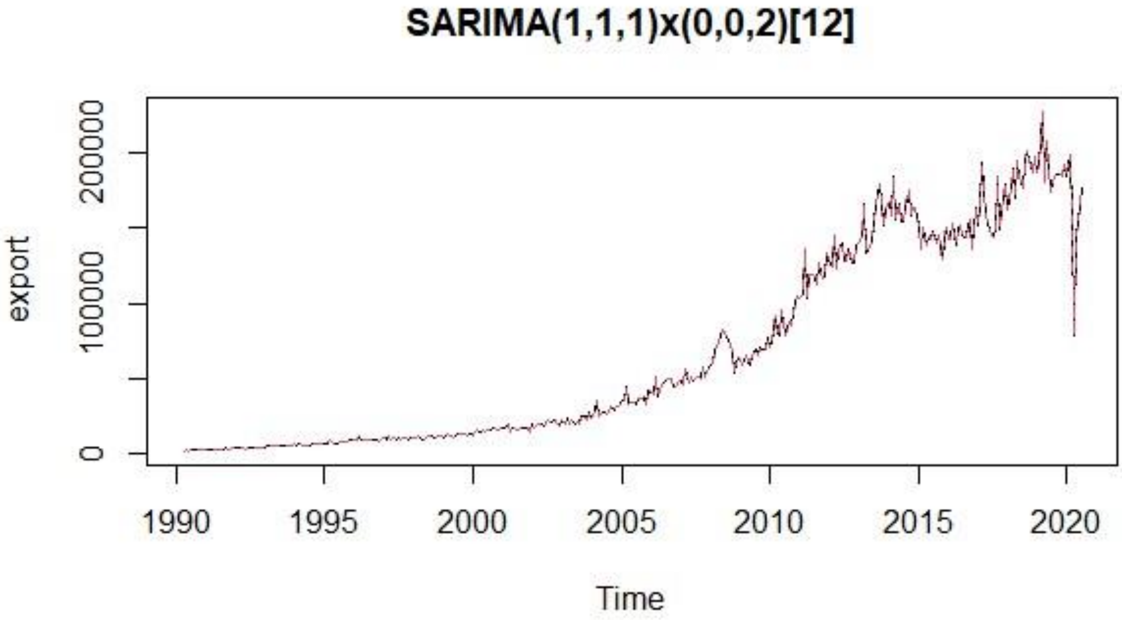
Fitting of SARIMA(1,1,1)*(0,0,2):

```
> #fitting SARIMA(1,1,1)(0,0,2)[12]
> library("sarima")
> expSARIMA=arima(export, order=c(1,1,1), seasonal=c(0,0,2))
> print(expSARIMA)

Call:
arima(x = export, order = c(1, 1, 1), seasonal = c(0, 0, 2))

Coefficients:
        ar1      ma1    sma1    sma2
0.3493  -0.7432  0.4113  0.2793
s.e.  0.0885   0.0585  0.0640  0.0610

sigma^2 estimated as 76877566:  log likelihood = -3812.49,  aic = 7634.97
> ts.plot(export, main="SARIMA(1,1,1)x(0,0,2)")
> expSARIMAfit=export-residuals(expSARIMA)
> points(expSARIMAfit, type="l", col=2, lty=2)
```



There exists a seasonal factor that creates such kind of disturbance in the series. Therefore we carry out further analysis to check whether it majorly affects the given time series model.

```
> expSARIMA1=sarima(log(export)~0|ma(1,0.3) + sma(12,1,c(-0.1)) + i(1) +s(12), ss.
method = "base") > print(expSARIMA1)
*Sarima model*
Call: sarima(model = log(export) ~ 0 | ma(1, 0.3) + sma(12, 1, c(-0.1)) + i(1) +
s(12),      ss.method = "base")

Unit root terms:
    (1 - B)(1 + B + ... + B^11)

Coefficients:
        ma1    sma1
0.6114  0.1259
s.e.  0.0591  0.0476

sigma^2 estimated as 0.02219:  log likelihood = 96.15,  aic = -186.3
> ts.plot(export, main="SARIMA(1,1,1)x(0,0,2)[12]")
> expSARIMAfit1=export-residuals(expSARIMA1)
> points(expSARIMAfit1, type="l", col=2, lty=2)
```



From the above graph it is clear that SARIMA(1,1,1)*(0,0,2)[12] is the best fit for the model. We check the best fit by comparing AIC and BIC of all the models.

```
> AIC(expAR)
[1] 8254.322
> AIC(expARIMA)
[1] 7692.1
> AIC(expSARIMA)
[1] 7634.971
> AIC(expSARIMA1)
[1] -186.295
> BIC(expAR)
[1] 8273.807
> BIC(expARIMA)
[1] 7703.783
> BIC(expSARIMA)
[1] 7654.443
> BIC(expSARIMA1)
[1] -174.7041
```

Models	AIC	BIC
AR(3)	8254.322	8273.807
ARIMA(1,1,1)	7692.1	7703.783
SARIMA(1,1,1)*(0,0,2)	7634.971	7654.443

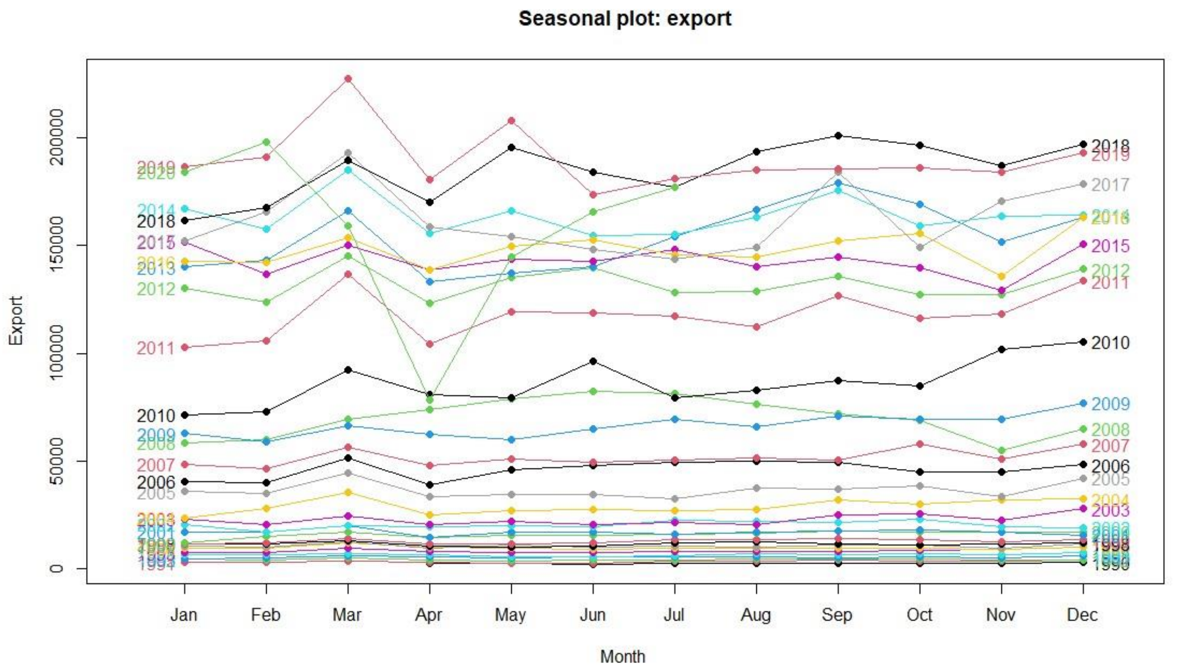
SARIMA(1,1,1)*(0,0,2)[12]	-186.295	-174.7041

From the above table we can conclude that AIC and BIC both are minimum for SARIMA(1,1,1)*(0,0,2)[12]. Therefore SARIMA(1,1,1)*(0,0,2)[12] is the best fit for the data. Since there exists seasonal component and it majorly affects the series hence it is the best fit.

```
> Box.test(export, lag=12, type='Ljung')

        Box-Ljung test

data:  export
X-squared = 4018.8, df = 12, p-value < 2.2e-16
```

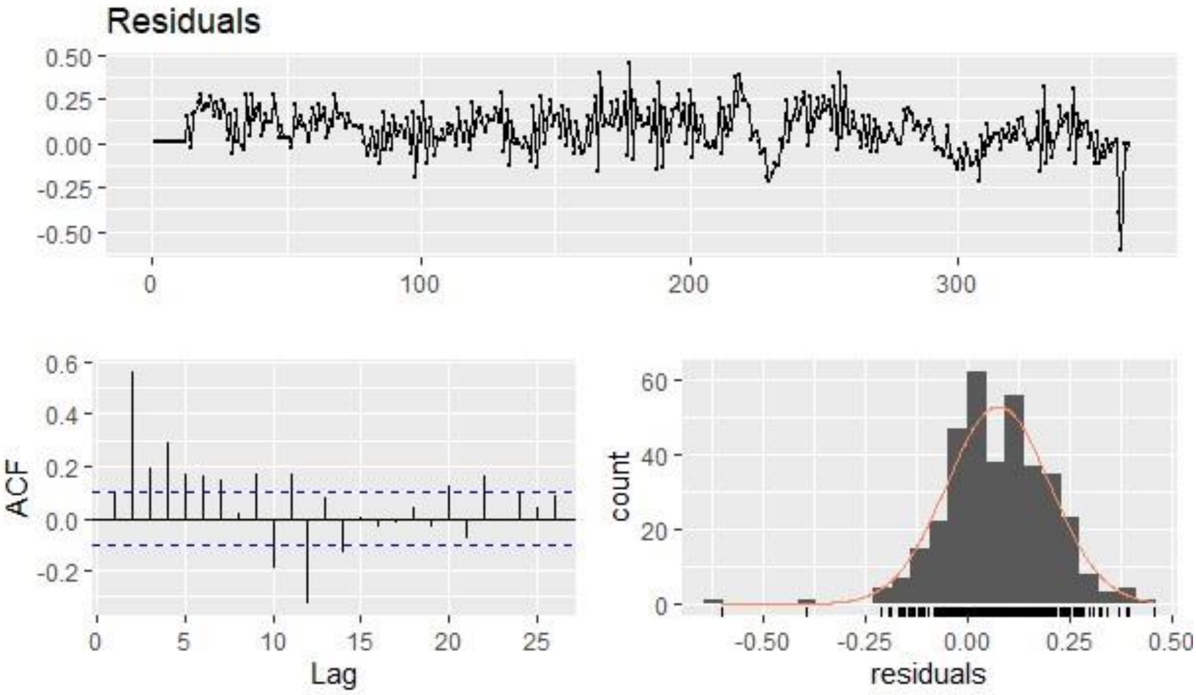
```
> seasonplot(export, year.labels.left = TRUE, year.labels = TRUE, col = 1:20, pch=19,
ylab = "Export", xlab="Month ")
```

Box-Ljung test leads us to the conclusion that the residuals are stationary in nature.



The seasonal plot signifies that there exists a seasonal component that affects the model. The plot for the year 2020 shows a strong decrease in exports from February to June. This is obvious as restrictions were imposed by most of the countries due to the pandemic COVID 19.

```
> checkresiduals(expSARIMA1)

	Ljung-Box test

data:  Residuals
Q* = 217.48, df = 8, p-value < 2.2e-16

Model df: 2.   Total lags used: 10
```



We can conclude from the graph that the residuals are stationary.

Power Consumption:

The dataset consists of electricity consumption data (Recorded at every hour for 10 days from 28/06/2017 to 11/07/2017) from a high-rise residential building inside the IIT Bombay campus. All the timestamps mentioned in the dataset is of Indian Standard Time (GMT+5.30), and India doesn't follow daylight saving time. The headers in the CSV files are as follows:

1) TS – Unix Time stamp (epochs)
2) V1 – Voltage of phase 1 (V)
3) V2 – Voltage of phase 2 (V)
4) V3 – Voltage of phase 3 (V)
5) W1 – Electricity consumption of phase 1 (Wh)
6) W2 – Electricity consumption of phase 2 (Wh)
7) W3 – Electricity consumption of phase 3 (Wh)
8) Virtual Apartment- VirtualApartment ID
9) date_time – Date and Time DD-MM-YYYY HH-MM
10) Day – day of record
11) Date – Date in YYYY-MM-DD
12) Time – Time in HH:MM:SS
13) Power - Sum of W1 + W2 + W3 (Wh)

DATASET LINK : http://seil.cse.iitb.ac.in/residential-dataset/ NOTE

: For the purpose of this project we are only going to consider

columns 9) date_time & 13) Power

```
> getwd()
[1] "C:/Users/Ankit Javeri/Downloads/TIME SERIES PROJECT/
Finalized Datasets"
> data = read.csv("power.csv")
```

We have a look at first 6 lines from the dataset

```
> head(data)
date_time    power
1 28/06/2017 0:30 898.5755
2 28/06/2017 1:30 273.3225
3 28/06/2017 2:30 244.4858
4 28/06/2017 3:30 612.9034
5 28/06/2017 4:30 330.3106
6 28/06/2017 5:30 614.8134
```

We will consider the column of power consumption for further analysis.

```
> str(data)
'data.frame':	240 obs. of  2 variables:
 $ date_time: chr  "28/06/2017 0:30" "28/06/2017 1:30"  "28/06/2017
2:30" "28/06/2017 3:30" ...
 $ power    : num  899 273 244 613 330 ...
```

We can see the class of the data.

```
> summary(data)
  date_time            power
Length:240         Min.   :  82.83
 Class :character   1st Qu.: 354.20
 Mode  :character   Median : 471.00
                   Mean   : 526.29
3rd Qu.: 651.00
                   Max.   :1383.65
> pow = data[,2]
> head(pow)
[1] 898.5755 273.3225 244.4858 612.9034 330.3106 614.8134
```

Exploratory analysis

```
> pow = ts(pow, frequency = 24, start = 1)
> plot.ts(pow, plot.type = "single", main="Time Series Plot")
> abline(reg = lm(pow~time(pow)))
> power=decompose(pow)
> plot(power)
```

Moving Average of order 5

```
> powerMA=SMA(pow, n=5)
> plot.ts(powerMA, main="Moving Average of order 5")
```

Holt Winter

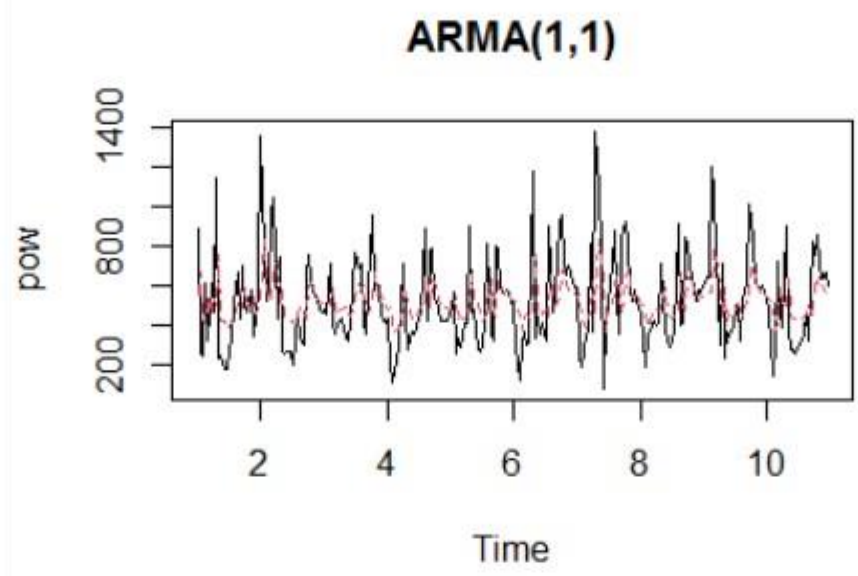
```
> powerHW=HoltWinters(pow, alpha=0.3, beta=FALSE, gamma=FALSE)
> plot(powerHW, main="Single Exponential smoothing")
> powerHW1=HoltWinters(pow, alpha=0.3, beta=0.2, gamma=FALSE)
> plot(powerHW1, main="Double Exponential smoothing")
> powdiff=diff(pow, differences = 1)
> plot.ts(powdiff)
```

Univariate analysis

```
> acf(pow, lag=40, main="ACF")
> pacf(pow, lag=40, main="PACF")
```

Fitting ARMA(1,1) model

```
> powAR1=arima(pow, order=c(1,0,1))
> print(powAR1)

Call: arima(x = pow, order = c(1,
0, 1))

Coefficients:
        ar1      ma1  intercept
     0.4223  -0.0536   527.4896 s.e.
0.1427   0.1556    22.7581

sigma^2 estimated as 46565:  log likelihood = -1630.45,  aic = 3268.91
> ts.plot(pow, main="ARMA(1,1)")
```




```
> powARfit1=pow-residuals(powAR1)
> points(powARfit1, type="l", col=2, lty=2)
```

Fitting ARIMA(1,1,1) model

```
> powAR2=arima(pow, order=c(1,1,1))
> print(powAR2)
```

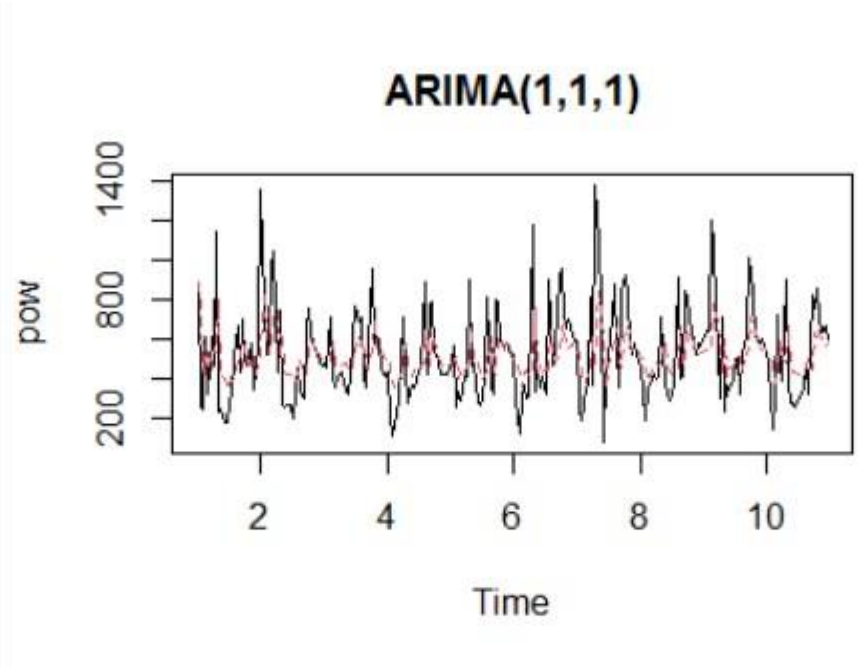
Call: arima(x = pow, order = c(1,
1, 1))

Coefficients:

	ar1	ma1
	0.3825	-1.00
s.e.	0.0604	0.03

sigma^2 estimated as 46784: log likelihood = -1626.48, aic = 3258.97

```
> ts.plot(pow, main="ARIMA(1,1,1)")
```




```
> powARfit2=pow-residuals(powAR2)
> points(powARfit2, type="l", col=2, lty=2)
```

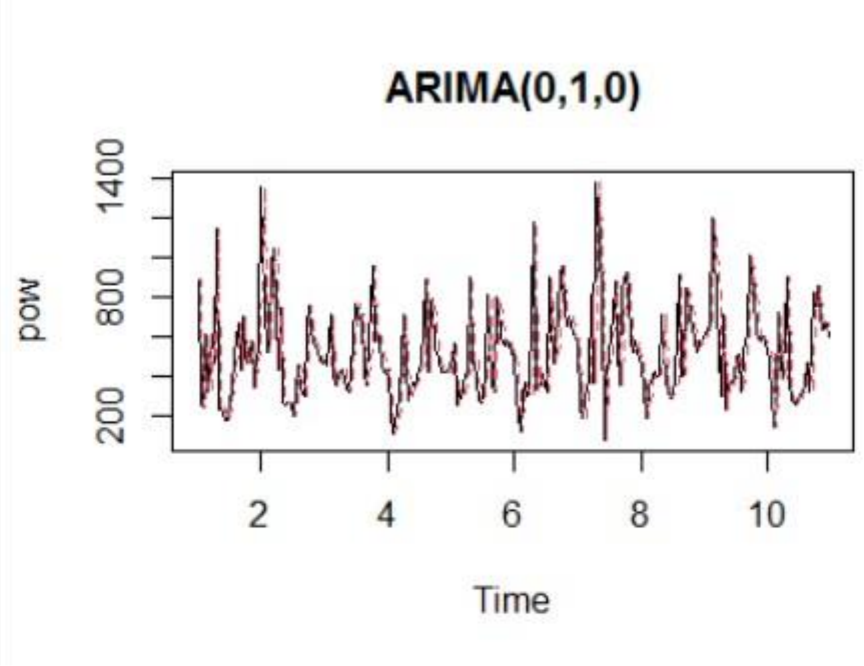
Fitting ARIMA(0,1,0) model

```
> powAR3=arima(pow, order=c(0,1,0))
> print(powAR3)
```

Call: arima(x = pow, order = c(0,
1, 0))

sigma^2 estimated as 67614: log likelihood = -1668.15, aic = 3338.31

```
> ts.plot(pow, main="ARIMA(0,1,0)")
```



> powARfit3=pow-residuals(powAR3)
> points(powARfit3, type="l", col=2, lty=2)

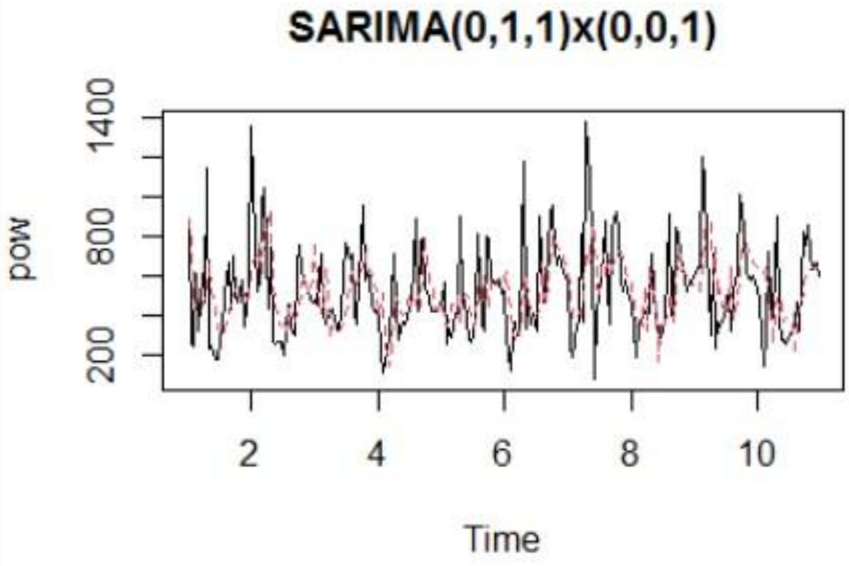
Fitting SARIMA(1,1,1)(0,0,2)[12]

> library("sarima")
> powSARIMA=arima(pow, order=c(0,1,1), seasonal=c(0,0,1))
> print(powSARIMA)

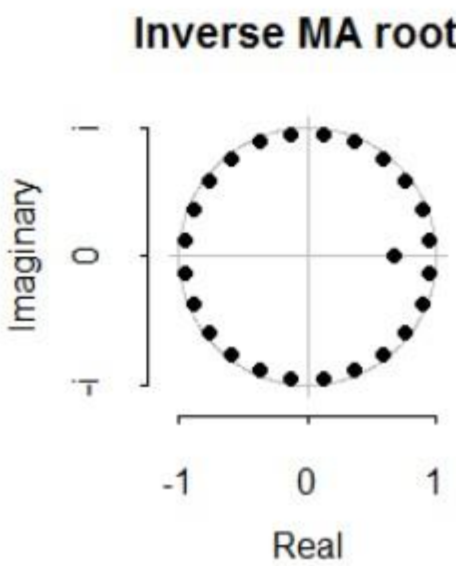
Call: arima(x = pow, order = c(0, 1, 1), seasonal = c(0,
0, 1))

Coefficients:
ma1 sma1 -
0.6683 0.3877
s.e. 0.0891 0.0605

sigma^2 estimated as 46590: log likelihood = -1625.9, aic = 3257.8
> ts.plot(pow, main="SARIMA(0,1,1)x(0,0,1)")



> powSARIMAfit=pow-residuals(powSARIMA)
> points(powSARIMAfit, type="l", col=2, lty=2)
> plot(powSARIMA)



This leads us to the conclusion that SARIMA(0,1,1)*(0,0,1) is an appropriate fit to the model

Foreign Exchange Reserves weekly for Gold:

DATA DESCRIPTION

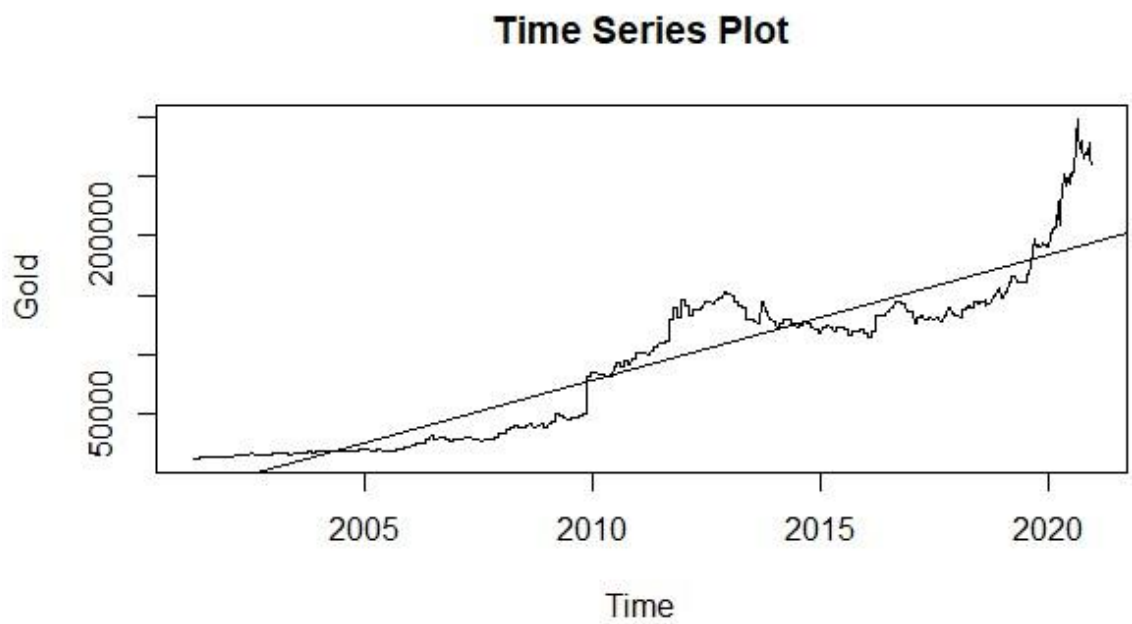
The dataset consists of weekly foreign exchange reserves in rupees and US dollars (from 06 April 2001 to 27 November 2020) comprising of 1029 observations. The headers in the excel file are as follows:

1) Year/Weekend Date
2) Foreign Currency Assets – in Rupees & US Dollars
3) Gold – in Rupees & US Dollars
4) SDRs – in Rupees & US Dollars
5) Total – in Rupees & US Dollars

NOTE : For the purpose of this project we are considering the column 3) Gold - in rupees only.

LINK : https://dbie.rbi.org.in/DBIE/dbie.rbi?site=statistics

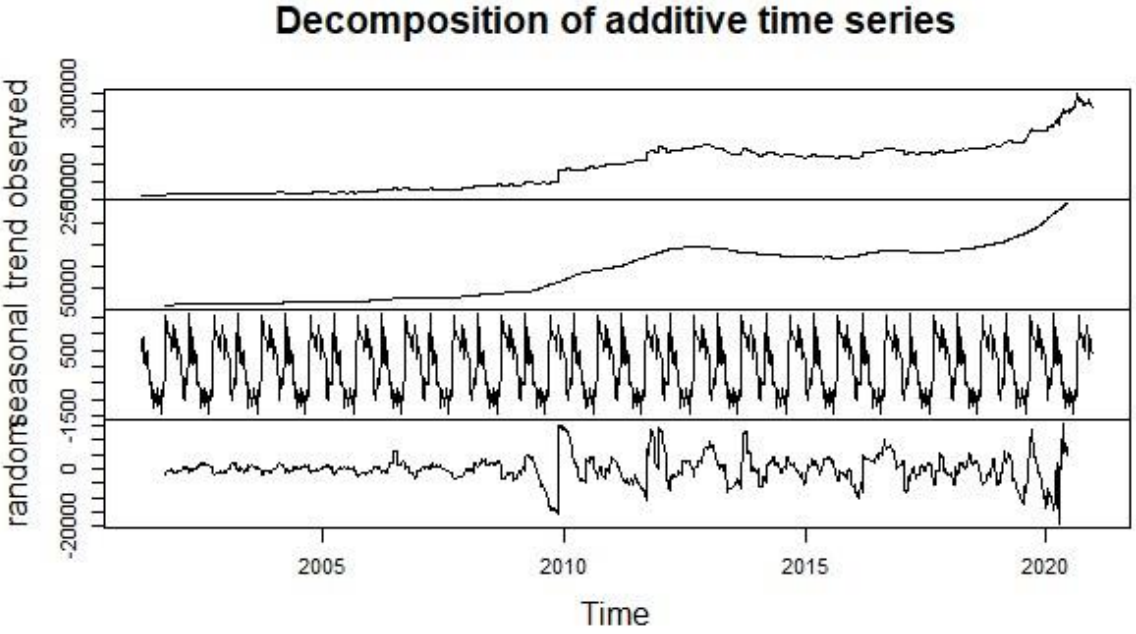
```
> #Explonatory analysis
> ferw=ts(R, frequency =365.25/7, start = c(2001,14))
> plot.ts(ferw, plot.type = "single", main="Time Series Plot")
> abline(reg=lm(ferw~time(ferw)))
```



The time series plot results in upward increasing trend. That means the gold price is increasing over the years stating from April 2001.

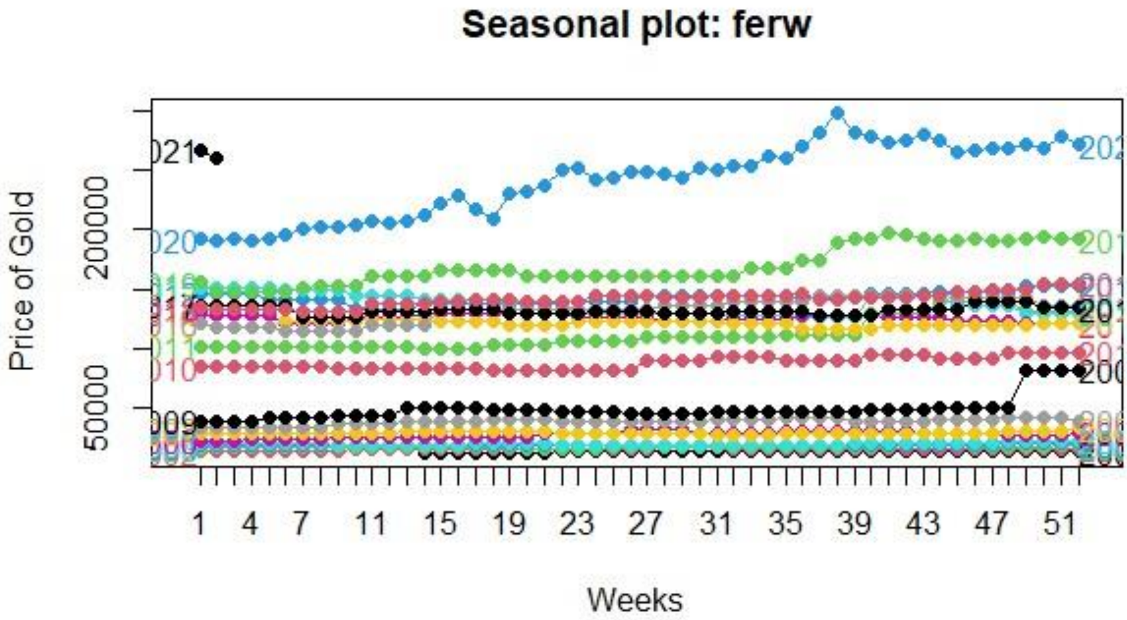
```
> ferw1=decompose(ferw)
```

```
> plot(ferw1)
```

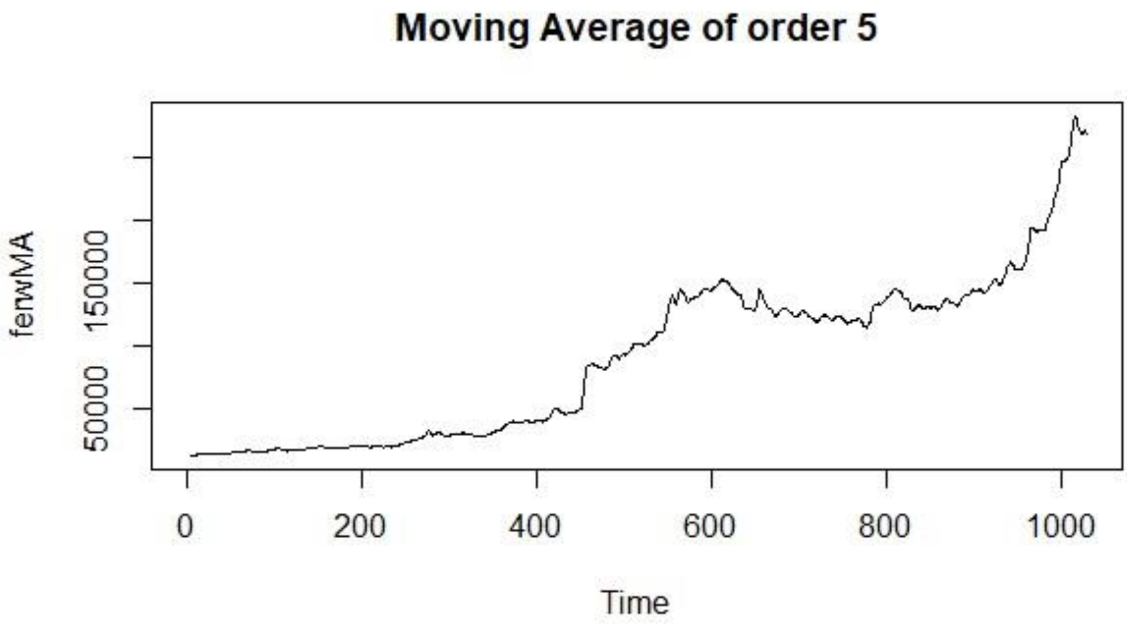


There exists trend and seasonality in the data of gold reserves. The seasonality in the data is mostly due to wedding seasons which results in increase in purchase of gold.

```
> seasonplot(ferw, year.labels.left = TRUE, year.labels = TRUE, col = 1:20, pch=19,
ylab = "Price of Gold", xlab="Weeks")
```

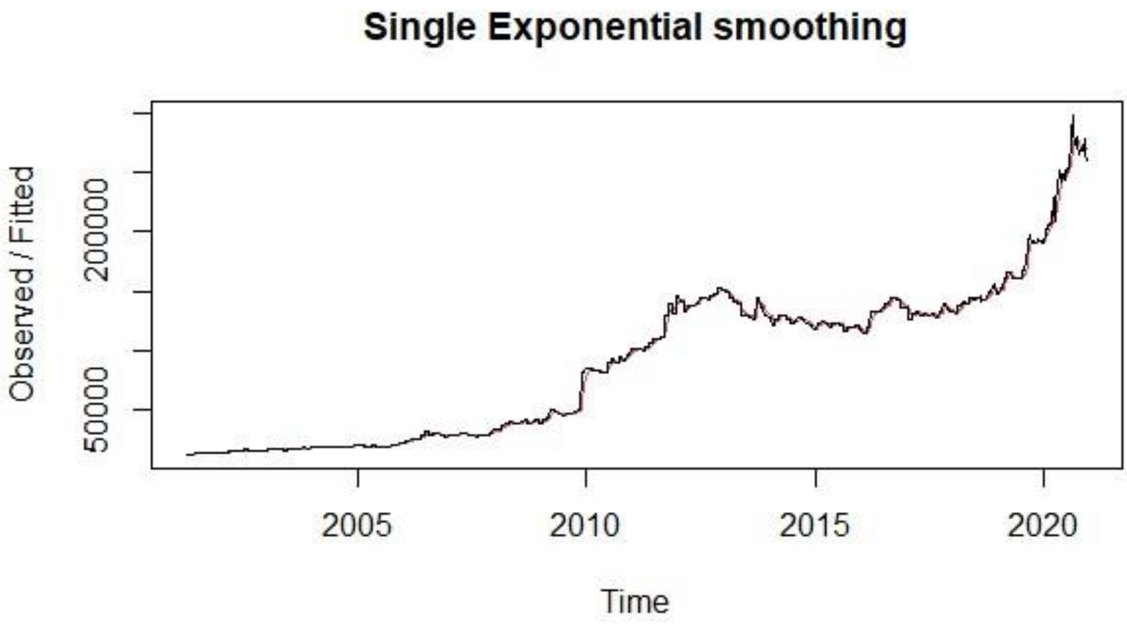



```
> #Moving Average of order 5
> ferwMA=SMA(ferw, n=5)
> plot.ts(ferwMA, main="Moving Average of order 5")
```



Moving average of order 5 results in smoothened graph on foreign exchange reserves of gold.

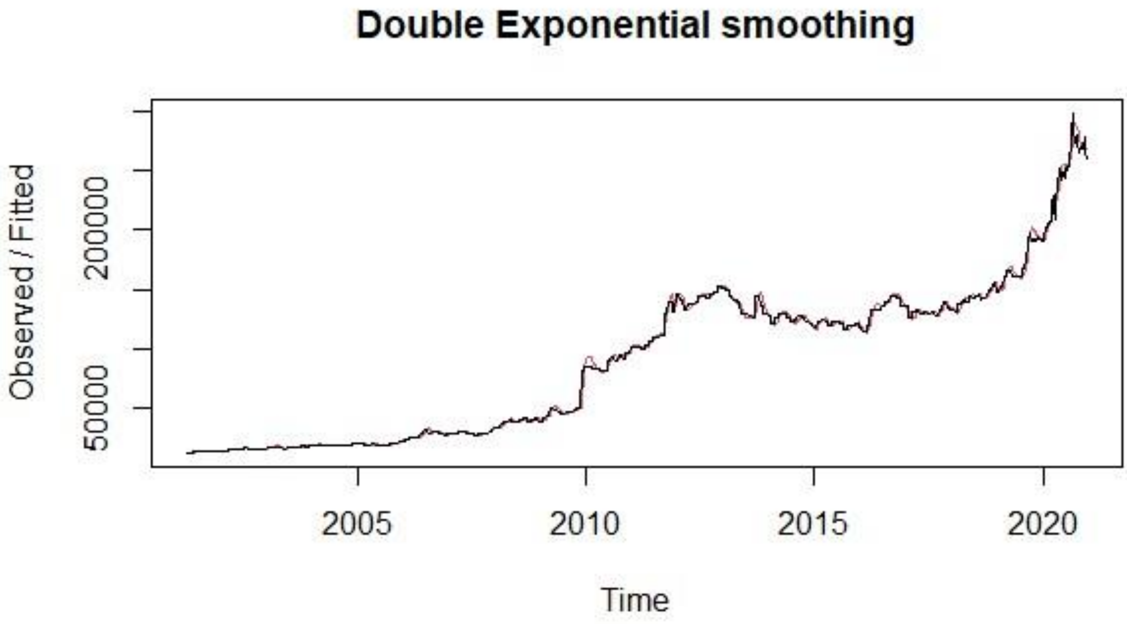
```
> #Holt Winter
> ferwHW=HoltWinters(ferw, alpha=0.3, beta=FALSE, gamma=FALSE)
> plot(ferwHW, main="Single Exponential smoothing")
```



Although single exponential smoothing fits quite well to the data with alpha value of 0.3 it isn't the good fit for the data since there are some irregularities in the data.

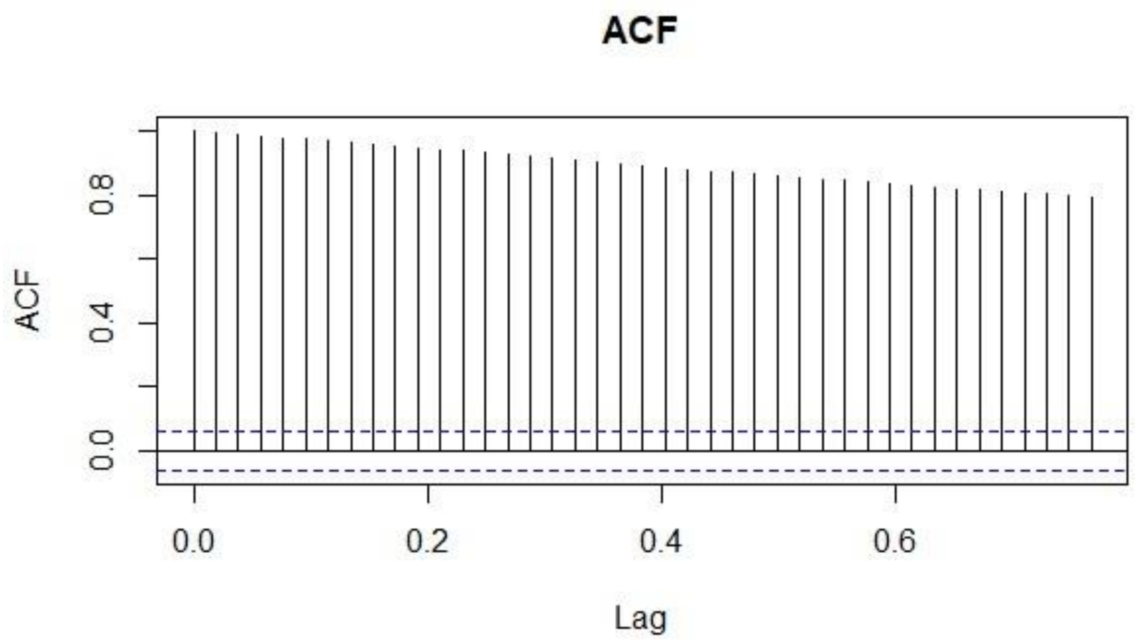
```
> ferwHW1=HoltWinters(ferw, alpha=0.3, beta=0.2, gamma=FALSE)
```

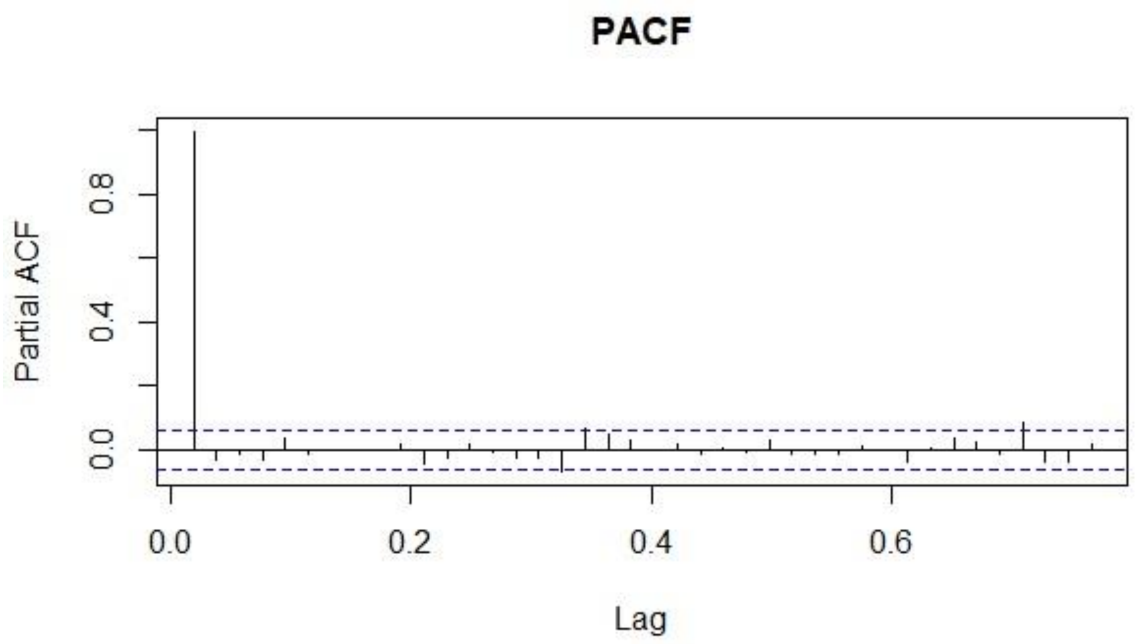
```
> plot(ferwHW1, main="Double Exponential smoothing")
```



The double exponential smoothing graph with alpha=0.3 and beta=0.2 isn't quite conclusive either.

```
> #Univariate analysis
> acf(ferw, lag=40, main="ACF")
> pacf(ferw, lag=40, main="PACF")
```





The ACF and PACF of the data is not conclusive either because it doesn't result in a final conclusion determining the appropriate for the data. Hence we move forward and carry out the analysis.

Fitting of ARIMA(1,1,1):

```
> #Fitting ARIMA(1,1,1) model
> ferwARIMA=arima(ferw, order=c(1,1,1))
> print(ferwARIMA)
```

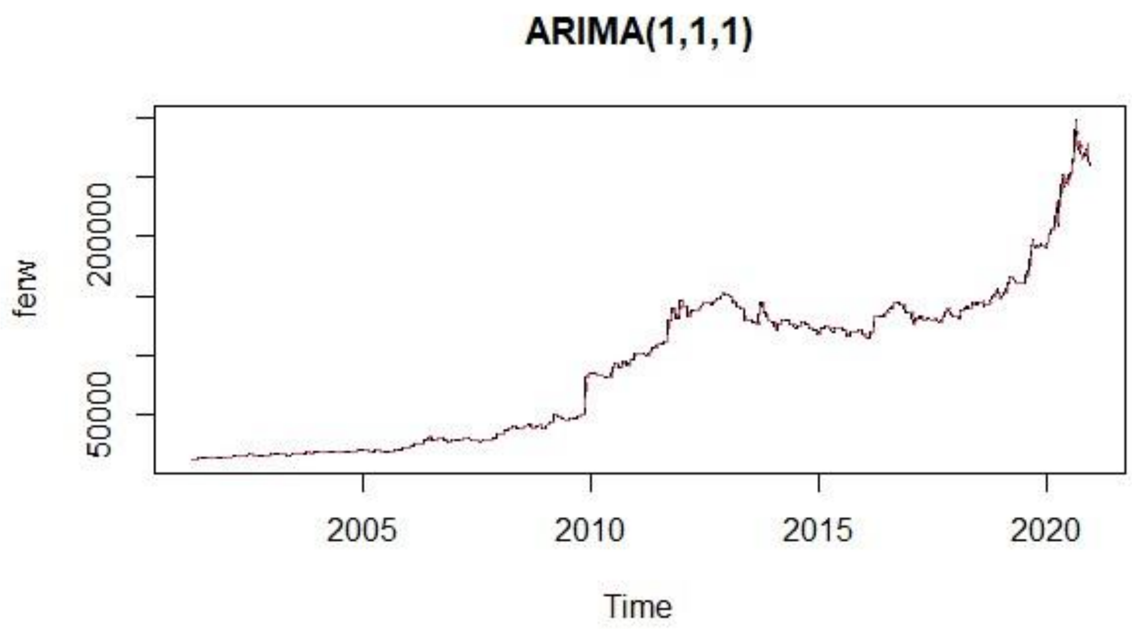
Call: arima(x = ferw, order = c(1,
1, 1))

Coefficients:

```
        ar1     ma1
      0.0095  0.0114
s.e.     NaN     NaN
```

sigma^2 estimated as 6738927: log likelihood = -9540.5, aic = 19087

```
> ts.plot(ferw, main="ARIMA(1,1,1)")
> ferwARIMAfit=ferw-residuals(ferwARIMA)
>  points(ferwARIMAfit, type="l", col=2, lty=2)
```



Although ARIMA(1,1,1) fits good for the given data, we would reach the conclusion for appropriate selection of the model after comparing AIC and BIC with other models.

Fitting SARIMA(0,1,0)*(0,0,1):

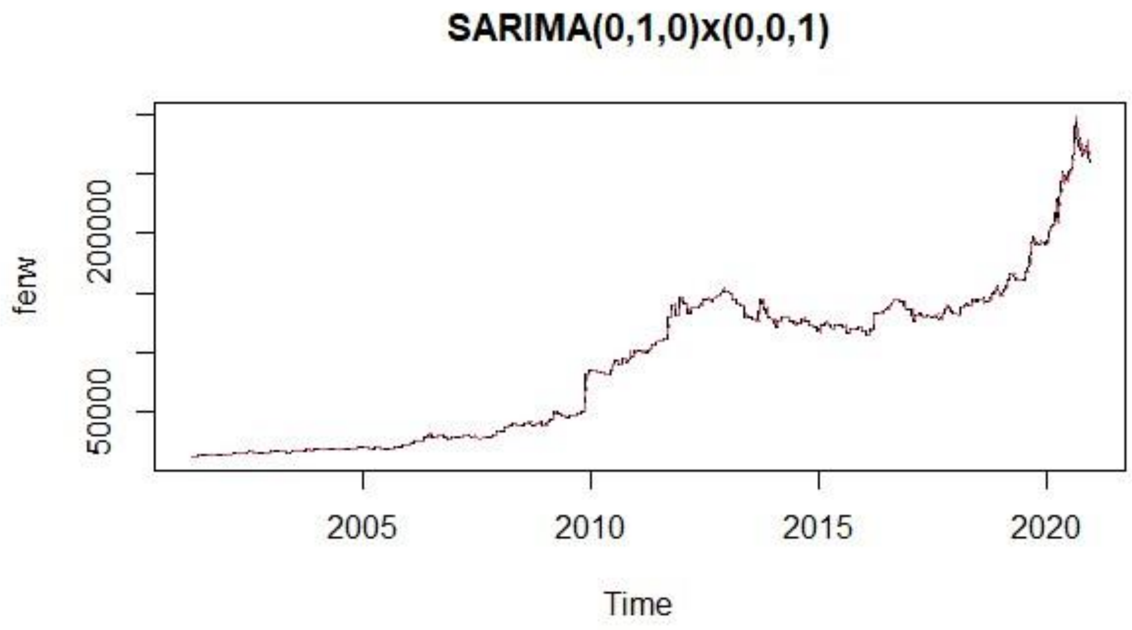
```
> #fitting SARIMA(0,1,0)(0,0,1)[52]
> library("sarima")
> ferwSARIMA=arima(ferw, order=c(0,1,0), seasonal=c(0,0,1))
> print(ferwSARIMA)
```

Call: arima(x = ferw, order = c(0, 1, 0), seasonal = c(0,
0, 1)) Coefficients: sma1   0.1536

s.e. 0.0360

sigma^2 estimated as 6618918: log likelihood = -9531.89, aic = 19067.77

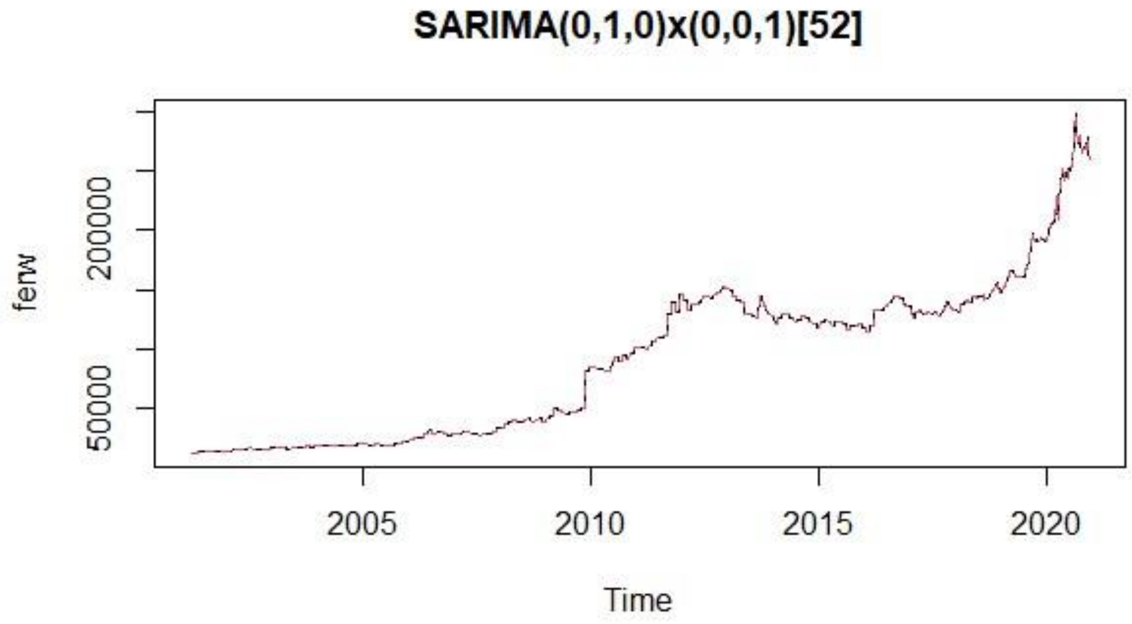
```
> ts.plot(ferw, main="SARIMA(0,1,0)x(0,0,1)")
> ferwSARIMAfit=ferw-residuals(ferwSARIMA)
> points(ferwSARIMAfit, type="l", col=2, lty=2)
```



Although the fit is good the model isn't an appropriate fit to the data. This might be due to the seasonal factors present in the data.

```
> ferwSARIMA1=sarima(log(ferw)~0| sar(52,1,c(-0.1)) + i(1) +s(365.25/7), ss.method = "bas
e")
> print(ferwSARIMA1)
*Sarima model*
Call:
sarima(model = log(ferw) ~ 0 | sar(52, 1, c(-0.1)) + i(1) + s(365.25/7), ss.method = "base")

Unit root terms:
    (1 - B)(1 + B + ... + B^51)

Coefficients:
sar1
      0.5081
s.e.  0.0387

sigma^2 estimated as 0.04307:  log likelihood = -173.16,  aic = 350.32
```

```
> ts.plot(ferw, main="SARIMA(0,1,0)x(0,0,1)[52]")
> ferwSARIMAfit1=ferw-residuals(ferwSARIMA1)
> points(ferwSARIMAfit1, type="l", col=2, lty=2)
```



SARIMA(0,1,0)*(0,0,1)[52] is an appropriate fit for the data. We compare the best fit by comparing its AIC and BIC with other models.

```
> AIC(ferwARIMA)
[1] 19087
> AIC(ferwSARIMA)
[1] 19067.77
> AIC(ferwSARIMA1) [1]
350.3151
> BIC(ferwARIMA)
[1] 19101.81
> BIC(ferwSARIMA)
[1] 19077.64
> BIC(ferwSARIMA1)
[1] 360.0841
```

Model	AIC	BIC
ARIMA(1,1,1)	19087	19101.81
SARIMA(0,1,0)*(0,0,1)	19067.77	19077.64
SARIMA(0,1,0)*(0,0,1)[52]	350.3151	360.0841

AIC and BIC is minimum for SARIMA(0,1,0)*(0,0,1)[52]. Therefore this is the best fit for the data.

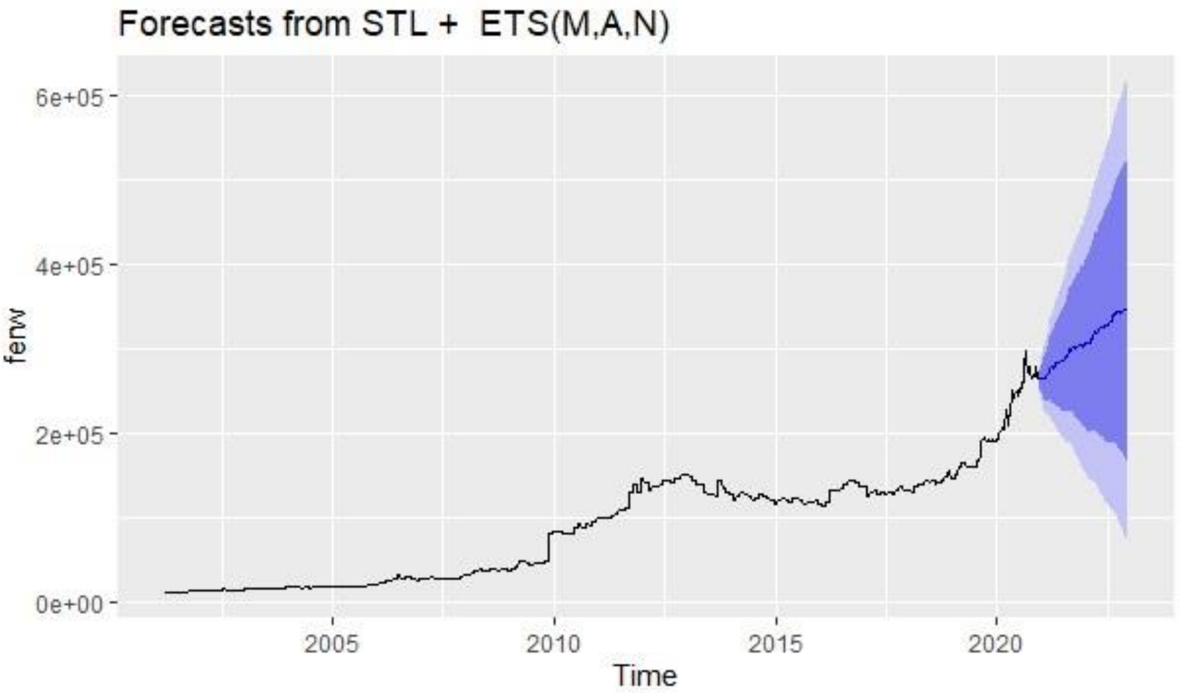
Hence We carry out forecasting based on this model. Hence the SARIMA model with 52 as the difference parameter is the best fit for the data. Further we carry out prediction based on the SARIMA(0,1,0)*(0,0,1)[52] model.

```
> a=stlf(ferw)
> predict(a,10)
         Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
2020.970       264486.4 255040.5 273932.3 250040.13 278932.7
2020.989       264700.0 251245.5 278154.5 244123.11 285276.8
2021.008       265239.3 248642.6 281836.0 239856.87 290621.8
2021.027       265474.2 246172.8 284775.7 235955.18 294993.2
2021.047       265446.5 243712.9 287180.1 232207.77 298685.1
2021.066       266127.7 242150.6 290104.8 229457.89 302797.5
2021.085       266668.8 240587.4 292750.2 226780.74 306556.9
2021.104       266926.8 238848.2 295005.4 223984.30 309869.3
2021.123       271048.4 241057.8 301039.1 225181.63 316915.3
2021.142       271311.9 239478.2 303145.7 222626.43 319997.4
2021.162       272146.7 238527.0 305766.3 220729.88 323563.5
2021.181       273690.9 238333.2 309048.6 219615.93 327765.8
2021.200       277829.4 240774.2 314884.5 221158.36 334500.4
2021.219       279357.1 240639.2 318075.1 220143.12 338571.1
2021.238       278436.5 238085.7 318787.4 216725.23 340147.8
2021.257       277815.7 235857.9 319773.6 213646.72 341984.7
2021.277       280246.0 236703.7 323788.3 213653.78 346838.3
2021.296       279895.9 234788.8 325003.0 210910.49 348881.3
2021.315       281130.8 234476.1 327785.4 209778.59 352482.9
2021.334       283270.7 235083.6 331457.8 209574.87 356966.5
2021.353       283512.0 233805.8 333218.3 207492.85 359531.2
2021.372       283566.0 232352.3 334779.8 205241.38 361890.7
2021.392       284216.9 231505.9 336927.9 203602.44 364831.4
2021.411       285273.5 231074.3 339472.7 202382.93 368164.0
2021.430       286066.5 230386.9 341746.1 200911.90 371221.1
2021.449       285816.4 228663.3 342969.6 198408.29 373224.6
2021.468       285875.7 227255.0 344496.4 196223.07 375528.3
2021.487       287270.3 227187.2 347353.5 195381.09 379159.6
2021.507       287622.0 226080.8 349163.2 193502.85 381741.2
2021.526       289000.2 226004.6 351995.8 192656.80 385343.6
2021.545       290402.9 225956.0 354849.7 191839.96 388965.8
```

2021.564	291815.2	225919.7	357710.8	191036.70	392593.8
2021.583	292140.2	224798.0	359482.5	189149.18	395131.3
2021.602	294415.9	225628.5	363203.3	189214.70	399617.2
2021.621	295938.6	225707.1	366170.0	188528.85	403348.3
2021.641	300610.1	228935.3	372284.9	190992.98	410227.2
2021.660	299253.6	226135.8	372371.4	187429.60	411077.6
2021.679	301188.5	226627.7	375749.3	187157.53	415219.4
2021.698	302229.5	226225.3	378233.7	185991.09	418467.9
2021.717	302684.9	225236.7	380133.2	184238.06	421131.8
2021.736	303355.0	224461.7	382248.2	182698.20	424011.7
2021.756	303209.6	222870.2	383549.0	180341.13	426078.1
2021.775	301347.5	219560.5	383134.5	176265.10	426429.9
2021.794	303210.4	219974.2	386446.7	175911.58	430509.3
2021.813	303396.8	218709.4	388084.2	173878.63	432915.0
2021.832	304083.4	217942.8	390224.1	172342.66	435824.2
2021.851	304195.5	216599.3	391791.7	170228.69	438162.3
2021.871	304117.8	215063.6	393172.0	167921.11	440314.4
2021.890	305490.2	214975.3	396005.1	167059.67	443920.8
2021.909	304826.9	212848.6	396805.3	164158.21	445495.6
2021.928	305032.4	211587.7	398477.2	162121.02	447943.9
2021.947	302254.4	207340.1	397168.7	157095.56	447413.3
2021.966	306089.5	209702.4	402476.5	158678.17	453500.8
2021.986	306303.0	208439.8	404166.3	156634.14	455971.9
2022.005	306842.4	207499.5	406185.3	154910.54	458774.3
2022.024	307077.3	206251.1	407903.5	152876.91	461277.6
2022.043	307049.5	204736.3	409362.8	150574.94	463524.1
2022.062	307730.8	203926.6	411534.9	148976.06	466485.5
2022.081	308271.9	202972.9	413570.8	147231.02	469312.8
2022.101	308529.9	201732.0	415327.7	145196.63	471863.1
2022.120	312651.5	204350.6	420952.4	147019.57	478283.5
2022.139	312915.0	203106.8	422723.2	144977.84	480852.1
2022.158	313749.8	202429.9	425069.6	143500.78	483998.7
2022.177	315293.9	202458.1	428129.8	142726.36	487861.5
2022.196	319432.4	205076.0	433788.8	144539.39	494325.5
2022.216	320960.2	205078.7	436841.7	143734.69	498185.7
2022.235	320039.6	202628.3	437450.9	140474.52	499604.7
2022.254	319418.8	200473.0	438364.6	137506.92	501330.7
2022.273	321849.1	201364.0	442334.2	137583.08	506115.1
2022.292	321499.0	199469.7	443528.2	134871.37	508126.5
2022.311	322733.8	199155.5	446312.1	133737.13	511730.5

```
2022.331       324873.8 199741.4 450006.1 133500.32 516247.2
2022.350       325115.1 198423.6 451806.6 131357.17 518873.0
2022.369       325169.1 196913.4 453424.8 129018.90 521319.3
2022.388       325820.0 195994.9 455645.1 127269.58 524370.4 2022.407
326876.5 195476.8 458276.3 125917.96 527835.1
2022.426       327669.6 194689.9 460649.2 124294.72 531044.4
2022.446       327419.5 192854.6 461984.4 121620.26 533218.8
2022.465       327478.8 191323.2 463634.3 119246.87 535710.7
2022.484       328873.4 191121.8 466625.0 118200.56 539546.3
2022.503       329225.1 189872.0 468578.2 116102.91 542347.3
2022.522       330603.3 189643.1 471563.4 115023.31 546183.3
2022.541       332005.9 189433.1 474578.7 113959.65 550052.2
2022.561       333418.3 189227.2 477609.4 112897.11 553939.5
2022.580       333743.3 187928.3 479558.3 110738.57 556748.1
2022.599       336019.0 188574.4 483463.6 110521.97 561516.0
2022.618       337541.6 188461.7 486621.6 109543.52 565539.7
2022.637       342213.2 191492.1 492934.3 111705.14 572721.2
2022.656       340856.7 188488.6 493224.8 107829.82 573883.6
2022.676       342791.6 188770.6 496812.5 107236.89 578346.2
2022.695       343832.6 188152.9 499512.2 105741.08 581924.1
2022.714       344288.0 186943.7 501632.4 103650.59 584925.4
2022.733       344958.0 185943.0 503973.0 101765.55 588150.5
2022.752       344812.7 184121.0 505504.4  99055.93 590569.4
2022.771       342950.6 180576.1 505325.0  94620.31 591280.8
2022.791       344813.5 180750.3 508876.7  93900.44 595726.5
2022.810       344999.9 179241.7 510758.1  91494.67 598505.1
2022.829       345686.5 178227.3 513145.7  89579.70 601793.3
2022.848       345798.6 176632.1 514965.1  87080.75 604516.4
2022.867       345720.8 174840.9 516600.8  84382.48 607059.2
2022.886       347093.3 174493.6 519693.0  83124.83 611061.7
2022.906       346430.0 172104.3 520755.7  79821.84 613038.2
2022.925       346635.5 170577.5 522693.5  77378.01 615893.0
2022.944       343857.5 166060.8 521654.2  71940.95 615774.1
```

```
> autoplot(forecast(ferw))
```



The above plot shows the forecasting for the data based on SARIMA model. Prices tend to increase because of the increasing trend. Hence gold prices tend to increase over the years.