Data Analytics

Time Series Analysis – Assignment - 3

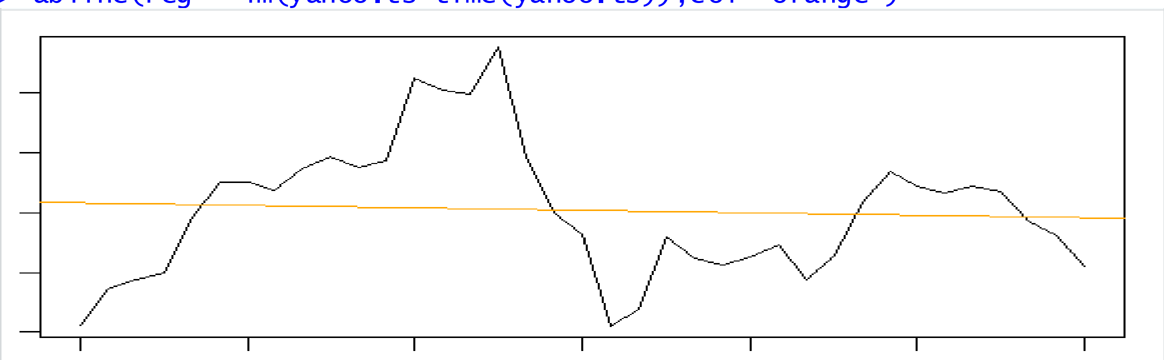
Name: - Ashish
Roll no.: - 17BCS006

===

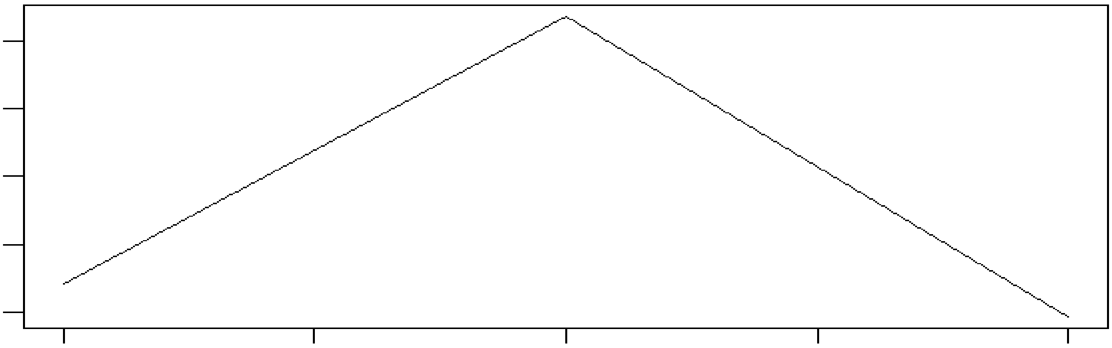
```
> # Loading the Required Libraries
> library(readr)
> library(ggfortify)
> library(tseries)
> library(Metrics)
> library(ggplot2)
> library(forecast)
> library(TTR)
> library(dplyr)
> require(graphics)
>
> yahoo=read.csv("C:/Users/Ashish/Downloads/yahoo.csv",header = T,
stringsAsFactors = F)
> class(yahoo)
[1] "data.frame"
> colnames(yahoo)
[1] "Date"      "Open"   "High"   "Low"   "Close"   "Adj.Close" "Volume"
> head(yahoo)
        Date  Open  High   Low Close Adj.Close Volume
1 2019-05-03 17.97 18.72 17.84 18.30  18.05834 178500
2 2019-05-06 17.96 18.74 17.96 18.61  18.36425 263100
3 2019-05-07 18.40 18.75 18.35 18.69  18.44319 179400
4 2019-05-08 18.63 19.06 18.54 18.74  18.49253  82700
5 2019-05-09 18.58 19.26 18.50 19.22  18.96619  89700
6 2019-05-10 19.05 19.50 19.05 19.43  19.24853  72800
> class(yahoo$Date)
[1] "character"
> class(yahoo$Adj.Close)
[1] "numeric"
> which(yahoo$Date == "2019-05-03")
[1] 1
> which(yahoo$Date == "2020-05-01")
[1] 252
> yahoo[1:252,]
          Date  Open  High   Low Close Adj.Close Volume
1   2019-05-03 17.97 18.72 17.84 18.30  18.05834 178500
2   2019-05-06 17.96 18.74 17.96 18.61  18.36425 263100
3   2019-05-07 18.40 18.75 18.35 18.69  18.44319 179400
4   2019-05-08 18.63 19.06 18.54 18.74  18.49253  82700
5   2019-05-09 18.58 19.26 18.50 19.22  18.96619  89700
6   2019-05-10 19.05 19.50 19.05 19.43  19.24853  72800
7   2019-05-13 19.14 19.59 19.02 19.43  19.24853 115600
8   2019-05-14 19.45 19.73 19.29 19.37  19.18909 102700
9   2019-05-15 19.18 19.76 19.02 19.55  19.36741  81500
10  2019-05-16 19.60 19.88 19.59 19.65  19.46648  78900
11  2019-05-17 19.47 19.96 19.47 19.56  19.37732  91600
12  2019-05-20 19.45 19.88 19.42 19.61  19.42685  92100
13  2019-05-21 19.68 20.33 19.68 20.31  20.12031  96500
.
.
.

141 2019-11-20 24.06 24.33 24.00 24.20  24.12607 125200
142 2019-11-21 24.31 24.35 23.95 23.99  23.91671  36000
 [ reached 'max' / getOption("max.print") -- omitted 110 rows ]
```

```
> # Q.2. Build a timeSeries object with the data.
> yahoo.ts = ts(data = yahoo$Adj.Close, frequency = 12,
start = c(2019-05-03), end=c(2020-05-01))
> class(yahoo.ts)
[1] "ts"
> str(yahoo.ts)
 Time-Series [1:37] from 2011 to 2014: 18.1 18.4 18.4 18.5 19 ...
> yahoo.ts
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2011 18.05834 18.36425 18.44319 18.49253 18.96619 19.24853 19.24853 19.18909 19.36741 19.46648 19.37732 19.42685
2012 20.12031 20.02125 19.99153 20.37788 19.46648 19.00087 18.82255 18.04983 18.19843 18.79283 18.61451 18.56498
2013 18.62442 18.72348 18.43619 18.64423 19.09003 19.33769 19.21881 19.15937 19.21881 19.16928 18.93152 18.80273
2014 18.55507
> start(yahoo.ts)
[1] 2011    1
> end(yahoo.ts)
[1] 2014    1
> #The time() function extracts the time index as a ts object
> time(yahoo.ts)
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2011 2011.000 2011.083 2011.167 2011.250 2011.333 2011.417 2011.500 2011.583 2011.667 2011.750 2011.833 2011.917
2012 2012.000 2012.083 2012.167 2012.250 2012.333 2012.417 2012.500 2012.583 2012.667 2012.750 2012.833 2012.917
2013 2013.000 2013.083 2013.167 2013.250 2013.333 2013.417 2013.500 2013.583 2013.667 2013.750 2013.833 2013.917
2014 2014.000
> #The frequency per period and time interval between observations of a ts object
> frequency(yahoo.ts)
[1] 12
> deltat(yahoo.ts)
[1] 0.08333333
> #This will plot a time series of the data.
> par(mar=c(1,1,1,1))
> plot(yahoo.ts, col="blue", lwd=2, ylab="Adjusted close",
main="Monthly closing price of SBUX")
> #plot a subset of the data use the window() function inside of plot()
> par(mar=c(1,1,1,1))
> plot(window(yahoo.ts, start = c(2019-05-03), end=c(2020-05-01),
 ylab="Adjusted close",col="blue", lwd=2, main="Monthly closing price of SBUX"))
```

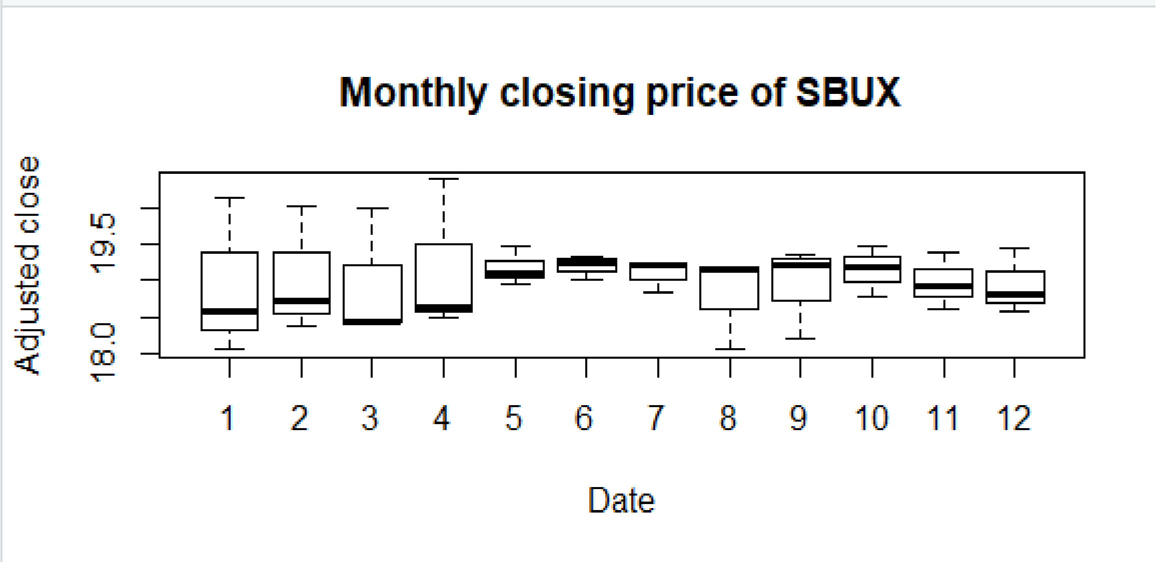

```
> #This will fit a line (also called trend line) shown in below figure.
> par(mar=c(1,1,1,1))
> abline(reg = lm(yahoo.ts~time(yahoo.ts)),col="orange")
```



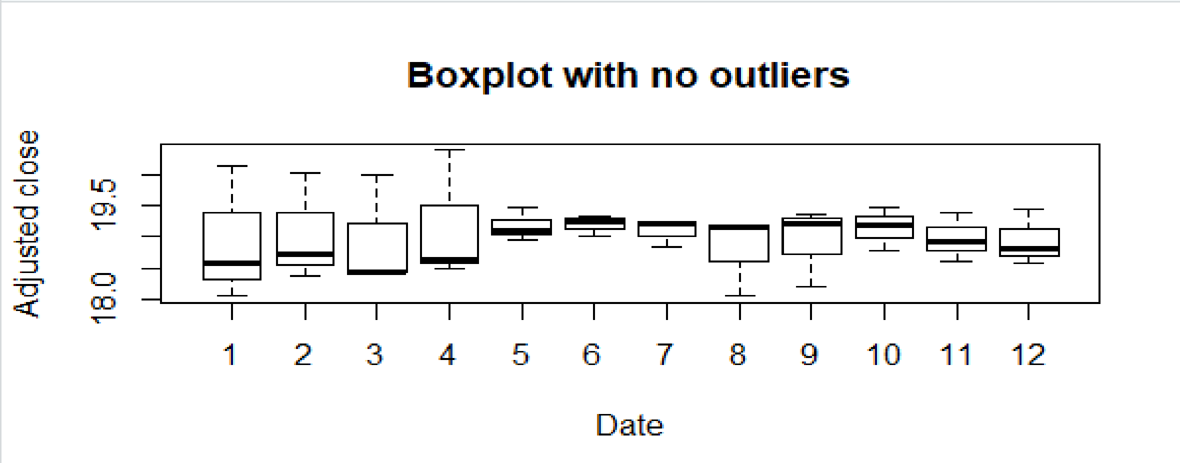
```
>
> #----------------------------------------
> # Q.3. It will plot the yearly mean values
> par(mar=c(1,1,1,1))
> plot(aggregate(yahoo.ts,FUN = mean))
```



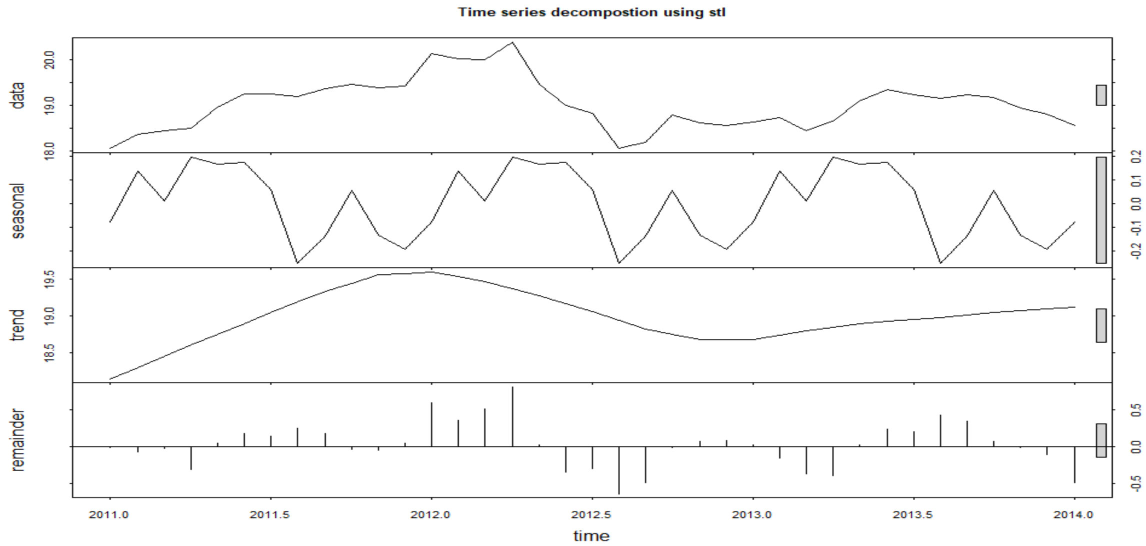
```
>
> #----------------------------------------
> #Ques 4. Boxplot across Quarters of month
> boxplot(yahoo.ts~cycle(yahoo.ts),xlab="Date", ylab = "Adjusted close",
 main ="Monthly closing price of SBUX")
```



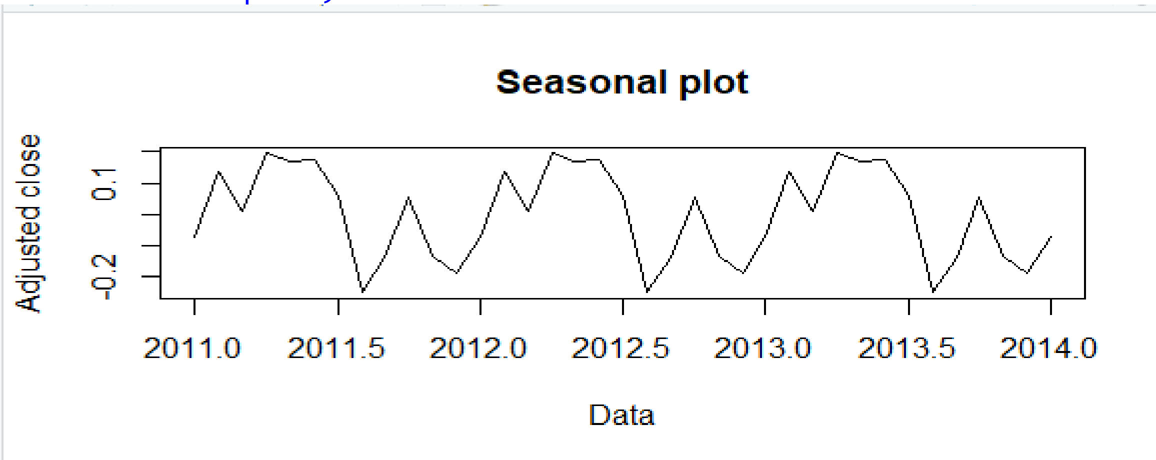

```
> #Data Cleaning : Since in the above box plot there was an outlier so I use
tsclean() to remove the outlier.
> yahoo1=tsclean(yahoo.ts)
> boxplot(yahoo1~cycle(yahoo1), xlab="Date", ylab = "Adjusted close",
main ="Boxplot with no outliers")
```



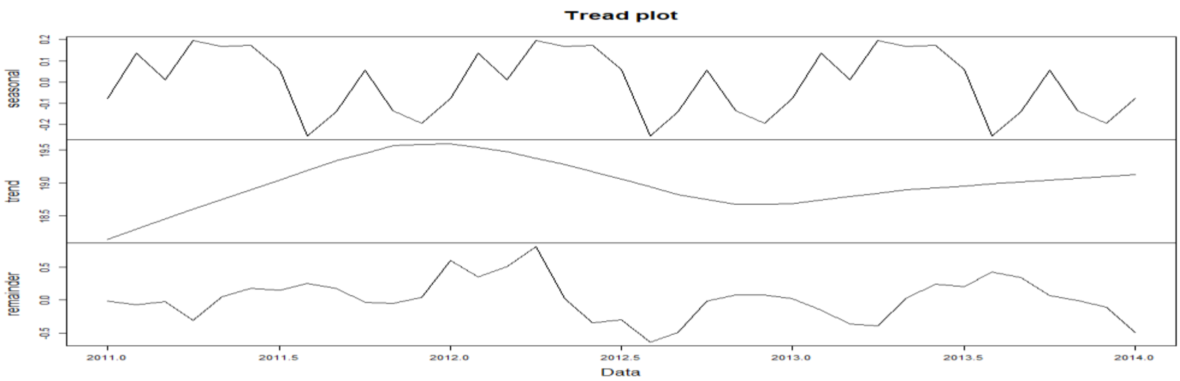
```
#------------------------------------------
> #Ques 5. Decomposition of Cleaned data (yahoo1) using stl function
> yahoo.ts_stl <- stl(yahoo1, s.window = "periodic")
> #par(mar=c(1,1,1,1))
> plot(yahoo.ts_stl, main = "Time series decompostion using stl")
```



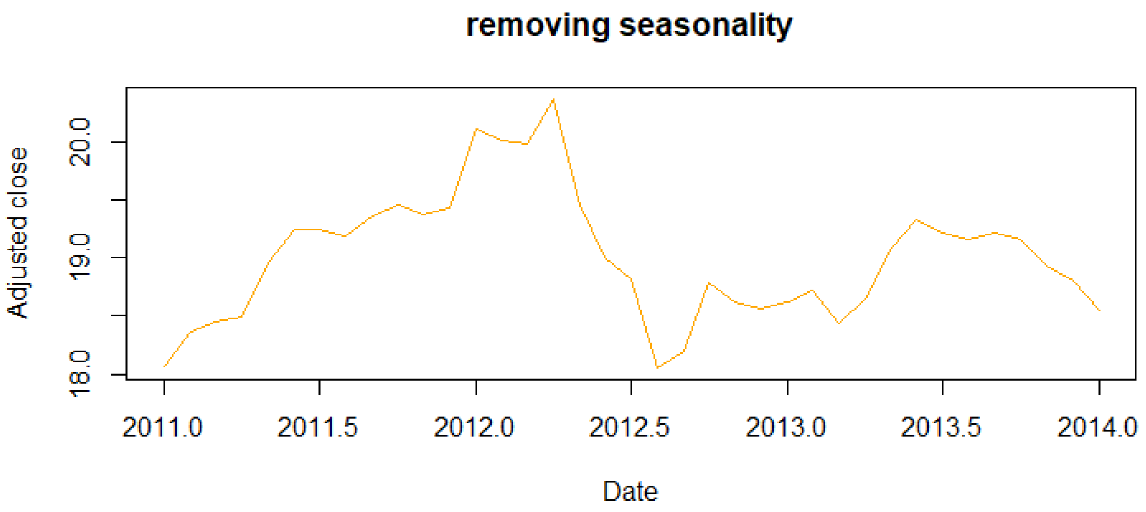

```
>
> #------------------------------------------
> # Ques 6. Type of Seasonality : yearly since the period is 12.
> yahoo.ts_stl_seasonal <- yahoo.ts_stl$time.series[,1]#seasonal
> plot((yahoo.ts_stl_seasonal),xlab="Data",ylab = "Adjusted close",
main="Seasonal plot")
```



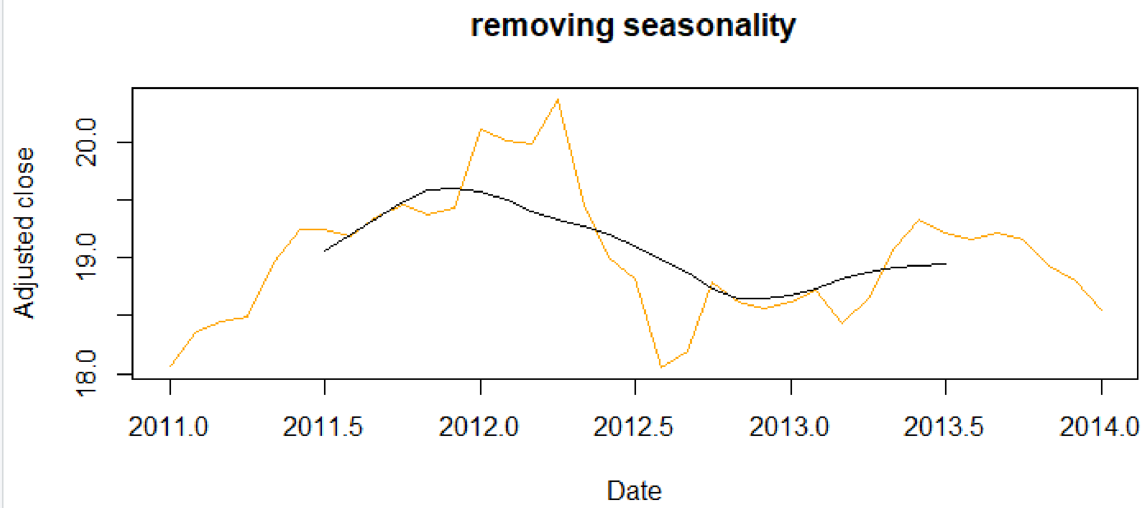

```
> yahoo.ts_stl_trend <- yahoo.ts_stl$time.series #tread
> plot((yahoo.ts_stl_trend),xlab="Data",ylab = "Adjusted close",main="Tread plot")
> yahoo.ts_stl_random <- yahoo.ts_stl$time.series #random
> plot((yahoo.ts_stl_random),xlab="Data",ylab = "Adjusted close",main="Tread plot")
```



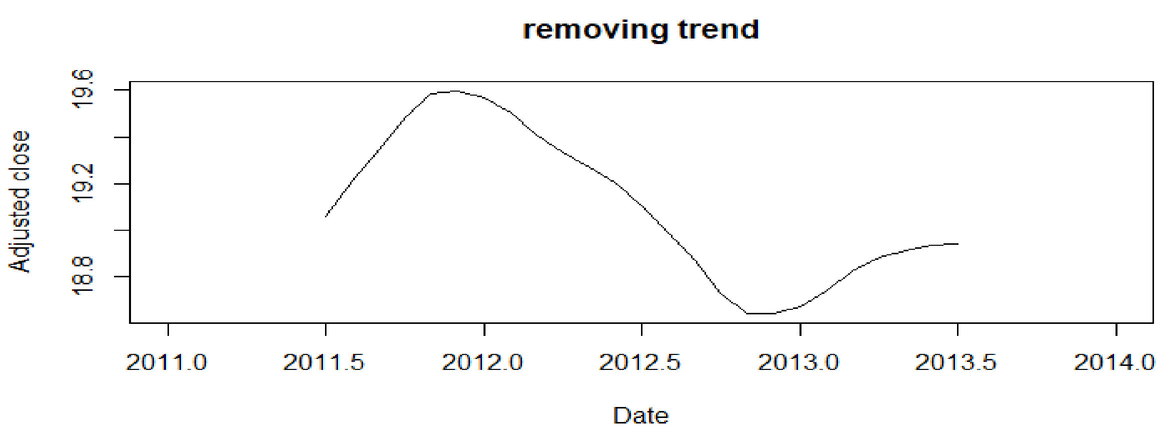
```
> #-------------------------------------------
> # Ques 7. Residue after removing seasonality and trend
> trend_yahoo.ts = ma(yahoo.ts,order = 12, centre = T)
> plot(as.ts(yahoo.ts),col = "orange",xlab="Date", ylab = "Adjusted close",
main ="removing seasonality")
```



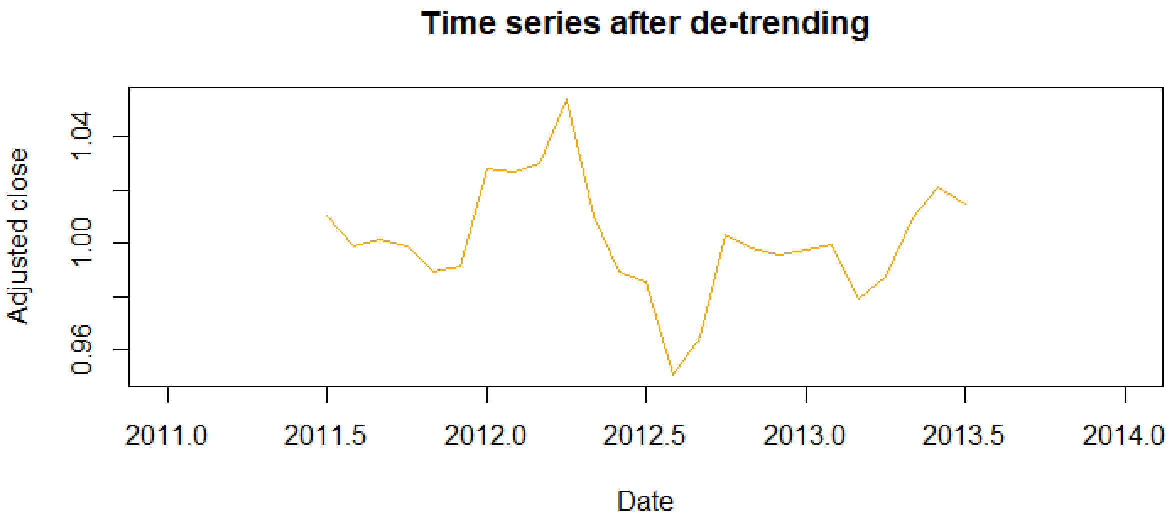

```
> lines(trend_yahoo.ts)
```



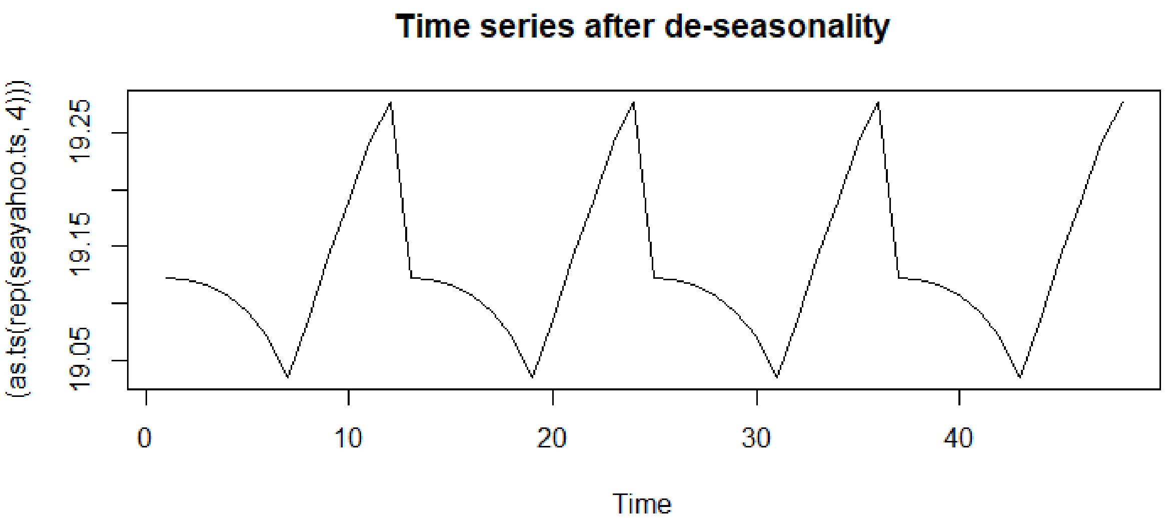

```
> plot(as.ts(trend_yahoo.ts),col = "black",xlab="Date", ylab = "Adjusted close",
main ="removing trend")
```



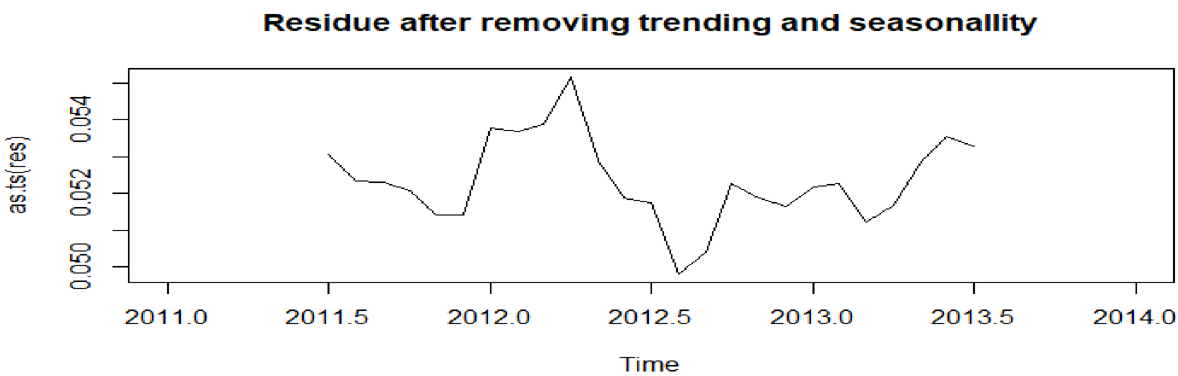
```
> # Removing trend from time series:
> Dtr = yahoo.ts/trend_yahoo.ts
> plot(as.ts(Dtr),main="Time series after de-trending",col = "orange",xlab="Date",
 ylab = "Adjusted close")
```



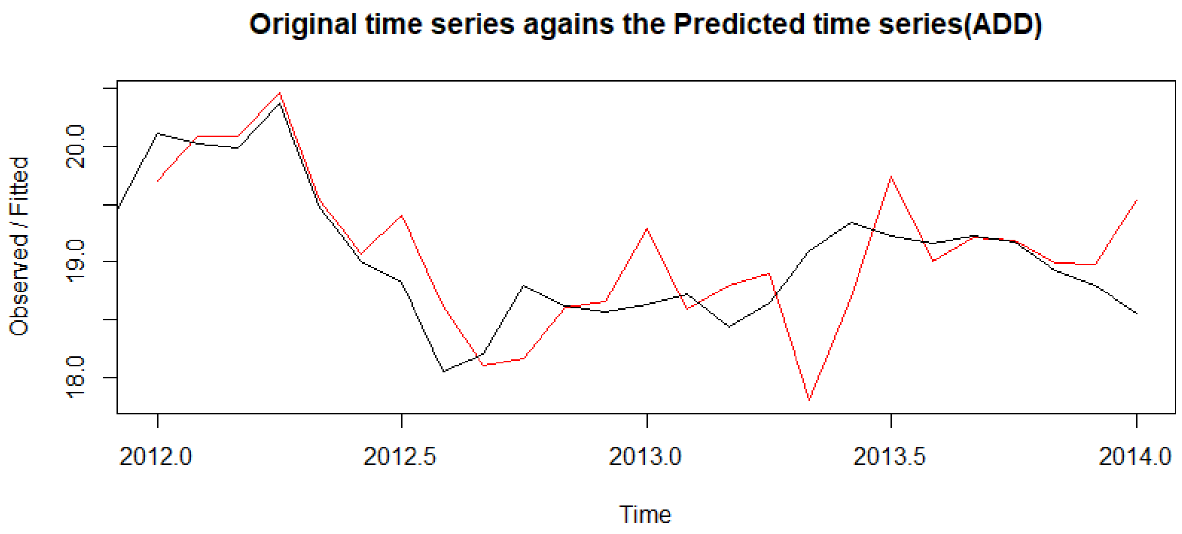

```
> #Removing seasonality
> m = t(matrix(data = trend_yahoo.ts, nrow = 12))
> seayahoo.ts = colMeans(m,na.rm = T)
> plot((as.ts(rep(seayahoo.ts,4))), main ="Time series after de-seasonality")
```




```
> res = yahoo.ts/ (trend_yahoo.ts * seayahoo.ts)
> plot(as.ts(res),main="Residue after removing trending and seasonallity")
```

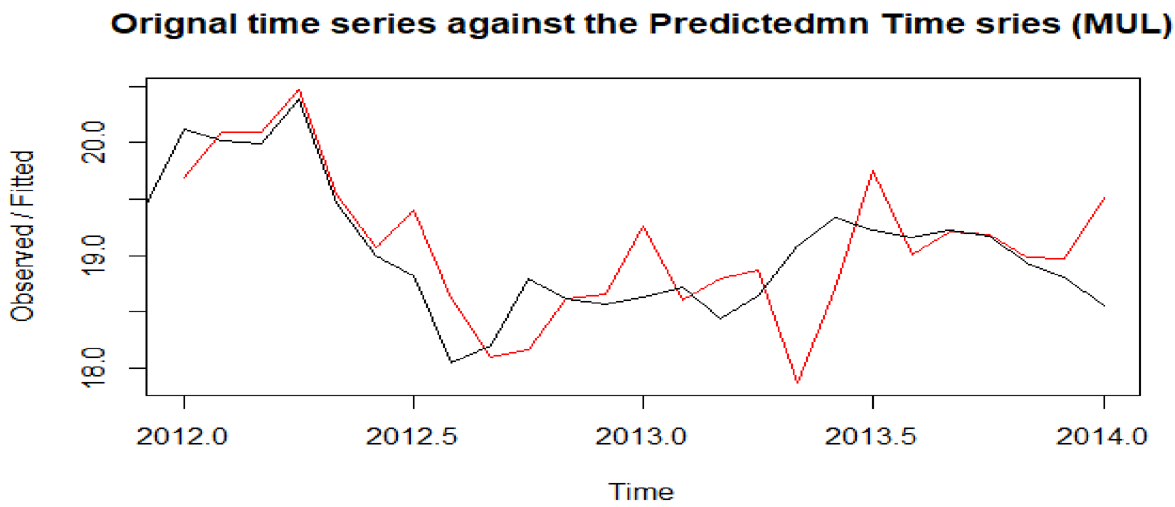


```
>
> #-------------------------------------------
> #Ques 8. Building Holt Winters Model and
> #Ques 12.Improving model by changing alpha,beta & gamma values
> #Trainig Dataset
> yahoo.ts_train <- ts(yahoo1, frequency = 12, start = c(2019-05-03),
end=c(2020-05-01))
> yahoo.ts_train
> yahoo.ts_train
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2011 18.05834 18.36425 18.44319 18.49253 18.96619 19.24853 19.24853 19.18909 19.36741 19.46648 19.37732 19.42685
2012 20.12031 20.02125 19.99153 20.37788 19.46648 19.00087 18.82255 18.04983 18.19843 18.79283 18.61451 18.56498
2013 18.62442 18.72348 18.43619 18.64423 19.09003 19.33769 19.21881 19.15937 19.21881 19.16928 18.93152 18.80273
2014 18.55507
> #Model 1: Seasonal HoltWinters Model
> HWyahoo.ts <- HoltWinters(yahoo.ts_train)
> HWyahoo.ts
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = yahoo.ts_train)

Smoothing parameters:
 alpha: 1
 beta : 0
 gamma: 0

Coefficients:
            [,1]
a   18.211058941
b    0.004903408
s1   0.310162601
s2   0.376621267
s3   0.839753267
s4  -0.011802566
s5  -0.409715524
s6  -0.014329399
s7  -0.228724399
s8  -0.183961941
s9  -0.227967191
s10 -0.416528649
s11 -0.377517524
s12  0.344010059
> plot(HWyahoo.ts,main="Original time series agains the Predicted time series(ADD)")
```

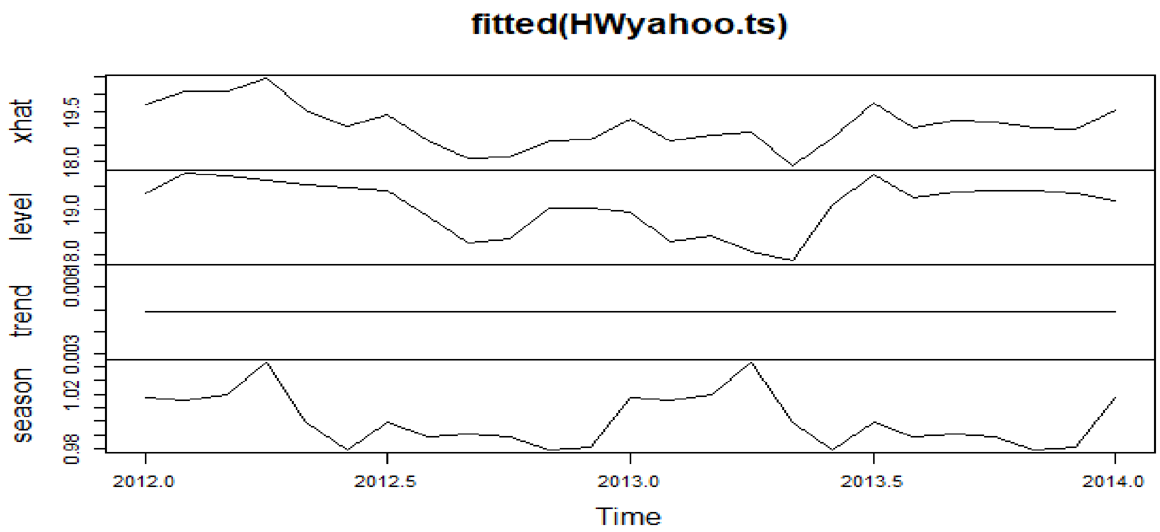



```
> HWyahoo.ts <- HoltWinters(yahoo.ts_train, seasonal ="multiplicative")
> HWyahoo.ts
```

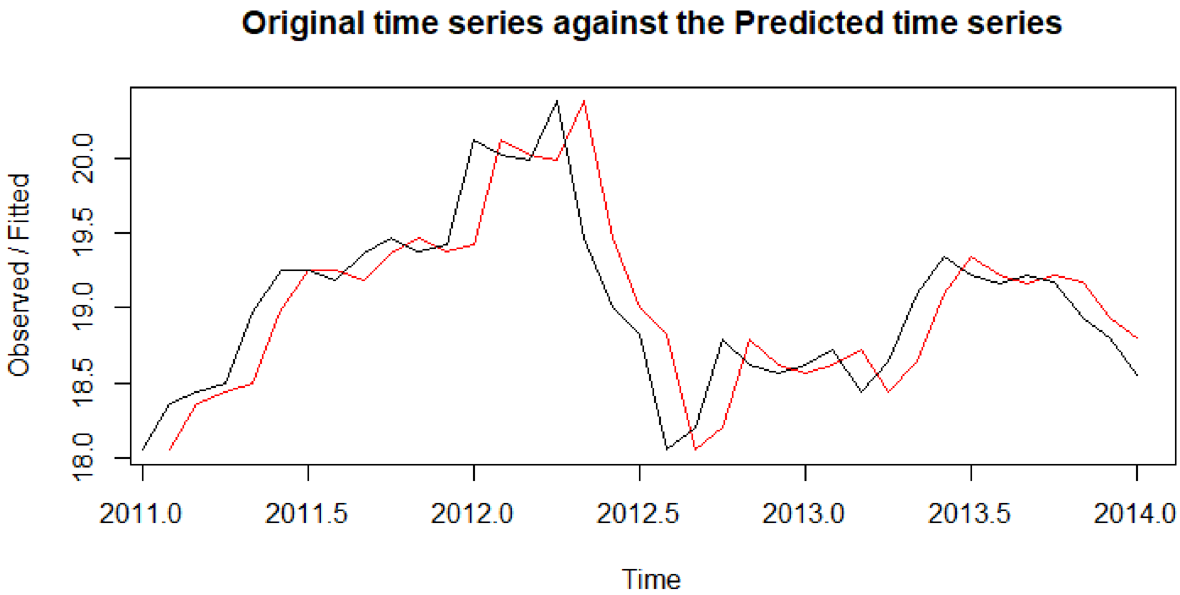
```
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
Call:
HoltWinters(x = yahoo.ts_train, seasonal = "multiplicative")
Smoothing parameters:
 alpha: 1
 beta : 0
 gamma: 0
Coefficients:
           [,1]
a    18.239660883
b     0.004903408
s1    1.015668389
s2    1.019186321
s3    1.043008989
s4    0.999454387
s5    0.978987776
s6    0.999436022
s7    0.988313476
s8    0.990611005
s9    0.988368376
s10   0.978848762
s11   0.980824064
s12   1.017292433

> plot(HWyahoo.ts, main = "Orignal time series against the Predictedmn Time
sries (MUL)")
```




```
> plot(fitted(HWyahoo.ts))
```

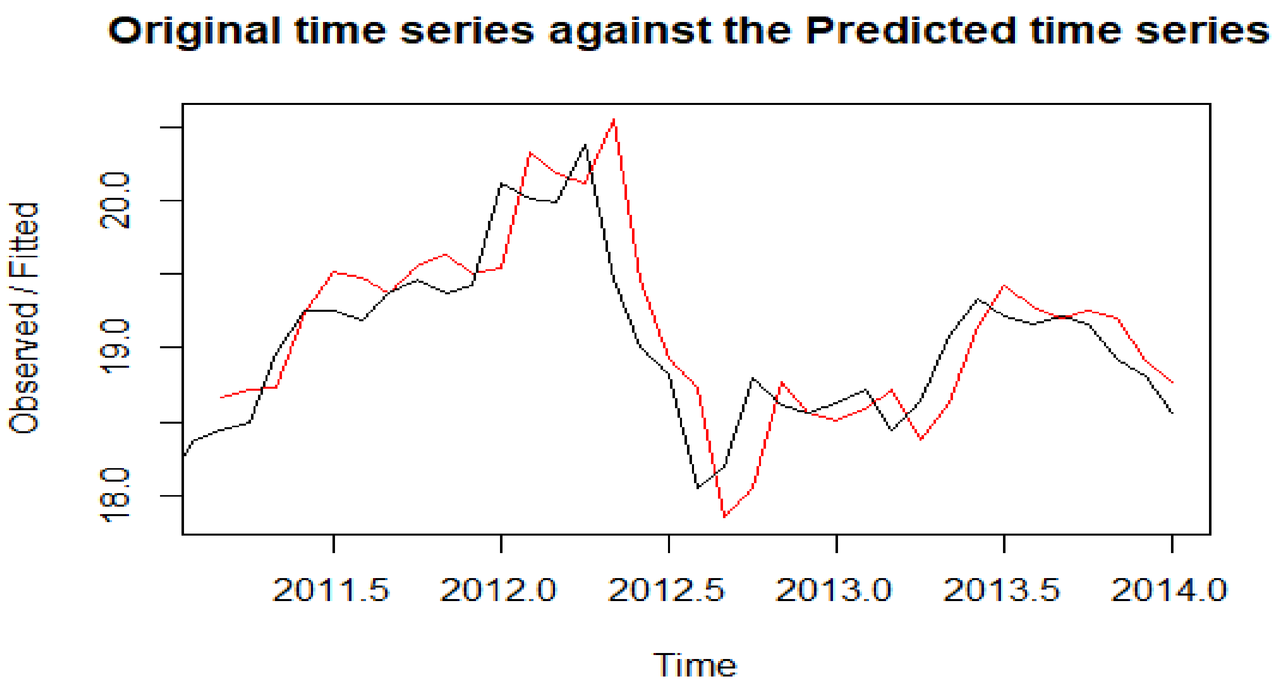


```
> #Model 2 : Simple Exponential Smoothing Model
> #This model has no trend and zero seasonal components
> Hotyahoo.ts <- HoltWinters(yahoo.ts_train, beta = F, gamma = F)
> Hotyahoo.ts
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = yahoo.ts_train, beta = F, gamma = F)

Smoothing parameters:
 alpha: 0.9999411
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 18.55508
> Hotyahoo.ts$fitted #Store the forecasts made by HW
              xhat    level
 Feb 2011 18.05834 18.05834
 Mar 2011 18.36423 18.36423
 Apr 2011 18.44319 18.44319
 May 2011 18.49253 18.49253
 Jun 2011 18.96616 18.96616
 Jul 2011 19.24851 19.24851
 Aug 2011 19.24853 19.24853
 Sep 2011 19.18910 19.18910
 Oct 2011 19.36740 19.36740
 Nov 2011 19.46647 19.46647
 Dec 2011 19.37732 19.37732
 Jan 2012 19.42685 19.42685
 Feb 2012 20.12027 20.12027
 Mar 2012 20.02125 20.02125
 Apr 2012 19.99153 19.99153
 May 2012 20.37786 20.37786
 Jun 2012 19.46653 19.46653
 Jul 2012 19.00090 19.00090
 Aug 2012 18.82256 18.82256
 Sep 2012 18.04988 18.04988
 Oct 2012 18.19842 18.19842
 Nov 2012 18.79280 18.79280
 Dec 2012 18.61452 18.61452
 Jan 2013 18.56498 18.56498
 Feb 2013 18.62441 18.62441
 Mar 2013 18.72348 18.72348
 Apr 2013 18.43621 18.43621
 May 2013 18.64422 18.64422
 Jun 2013 19.09000 19.09000
 Jul 2013 19.33768 19.33768
 Aug 2013 19.21882 19.21882
 Sep 2013 19.15938 19.15938
 Oct 2013 19.21881 19.21881
 Nov 2013 19.16928 19.16928
 Dec 2013 18.93154 18.93154
 Jan 2014 18.80274 18.80274
> plot(Hotyahoo.ts, main = "Original time series against the Predicted time series")
```

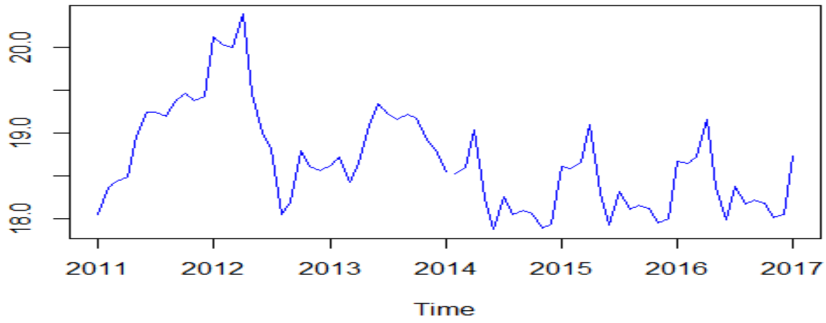



```
> #Model 3 : Non-seasonal Holt Winters
> #It is better Prediction model than hotyahoo.ts
```

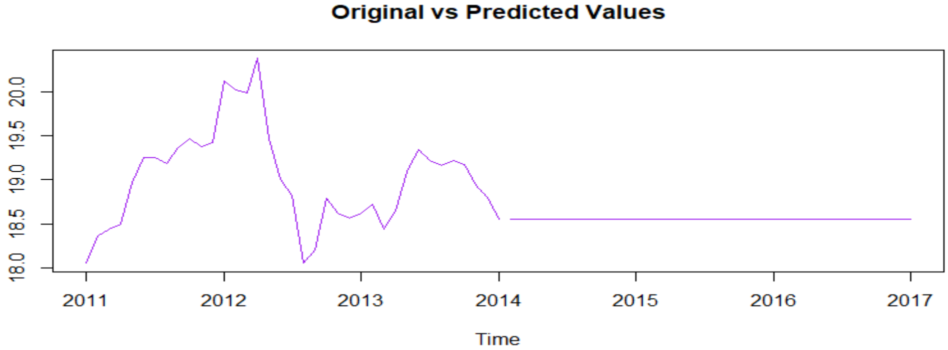
```
> Hotyahoo.ts1 <- HoltWinters(yahoo.ts_train, gamma =F)
> Hotyahoo.ts1
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = yahoo.ts_train, gamma = F)

Smoothing parameters:
 alpha: 1
 beta : 0.1607397
 gamma: FALSE

Coefficients:
         [,1]
a 18.55506900
b -0.06829953
> Hotyahoo.ts1$fitted
             xhat    level        trend
Mar 2011 18.67015 18.36425  0.305906000
Apr 2011 18.71262 18.44319  0.269424346
May 2011 18.72658 18.49253  0.234048043
Jun 2011 19.23876 18.96619  0.272563048
Jul 2011 19.52267 19.24853  0.274134432
Aug 2011 19.47860 19.24853  0.230070132
Sep 2011 19.37263 19.18909  0.183534668
Oct 2011 19.55011 19.36741  0.182696142
Nov 2011 19.63573 19.46648  0.169253293
Dec 2011 19.50503 19.37732  0.127716005
Jan 2012 19.54200 19.42685  0.115149531
Feb 2012 20.32842 20.12031  0.208106850
Mar 2012 20.17998 20.02125  0.158731963
Apr 2012 20.11997 19.99153  0.128440563
May 2012 20.54778 20.37788  0.169897826
Jun 2012 19.46256 19.46648 -0.003911002
Jul 2012 18.92274 19.00087 -0.078124062
Aug 2012 18.72832 18.82255 -0.094229532
Sep 2012 17.84654 18.04983 -0.203289437
Oct 2012 18.05171 18.19843 -0.146726496
Nov 2012 18.76523 18.79283 -0.027598491
Dec 2012 18.56268 18.61451 -0.051825428
Jan 2013 18.51352 18.56498 -0.051456783
Feb 2013 18.59078 18.62442 -0.033631584
Mar 2013 18.71118 18.72348 -0.012301807
Apr 2013 18.37969 18.43619 -0.056503662
May 2013 18.63025 18.64423 -0.013981301
Jun 2013 19.14995 19.09003  0.059923348
Jul 2013 19.42779 19.33769  0.090101215
Aug 2013 19.27532 19.21881  0.056509466
Sep 2013 19.19725 19.15937  0.037872260
Oct 2013 19.26015 19.21881  0.041338571
Nov 2013 19.19601 19.16928  0.026732380
Dec 2013 18.91574 18.93152 -0.015781898
Jan 2014 18.76879 18.80273 -0.033946471
> plot(Hotyahoo.ts1,main="Original time series against the Predicted time series")
```



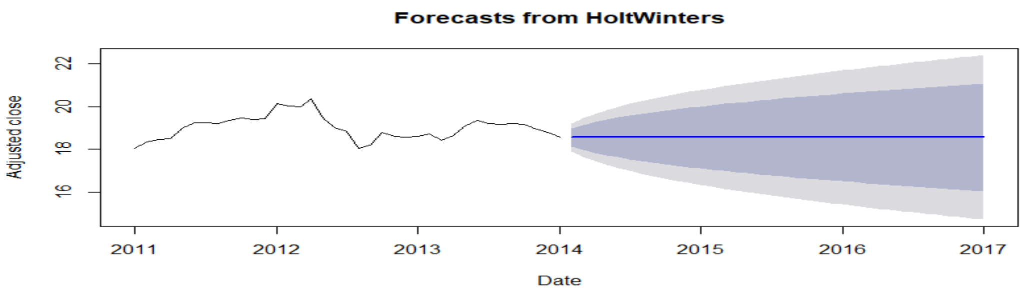
```
> #-----------------------------------------
> #Ques 9. Predict the values for the next 25% of the time
> #Ques 10. Plot the predicted values along with the actual values to compare them.
> HW_pred = predict(HWyahoo.ts, n.ahead = 3*12)
> round(HW_pred) # predicted values for next  years
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2014      19  19  19  18  18  18  18  18  18  18  18
2015  19  19  19  19  18  18  18  18  18  18  18  18
2016  19  19  19  19  18  18  18  18  18  18  18  18
2017  19
> ts.plot(HW_pred, yahoo1, col = "blue")
```



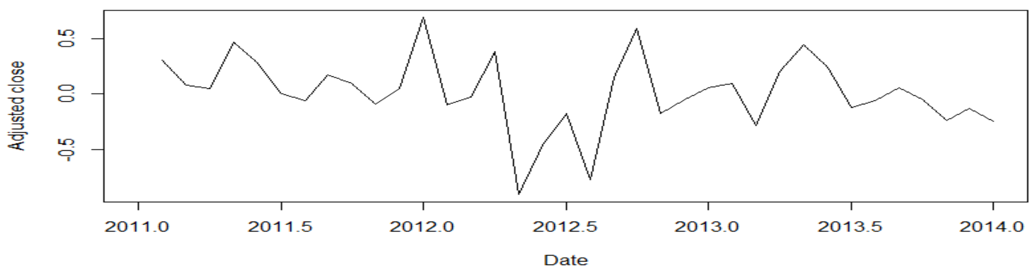

```
> H_pred = predict(Hotyahoo.ts,n.ahead = 3*12)
> ts.plot(yahoo1,H_pred,col = "purple", main = "Original vs Predicted Values")
```



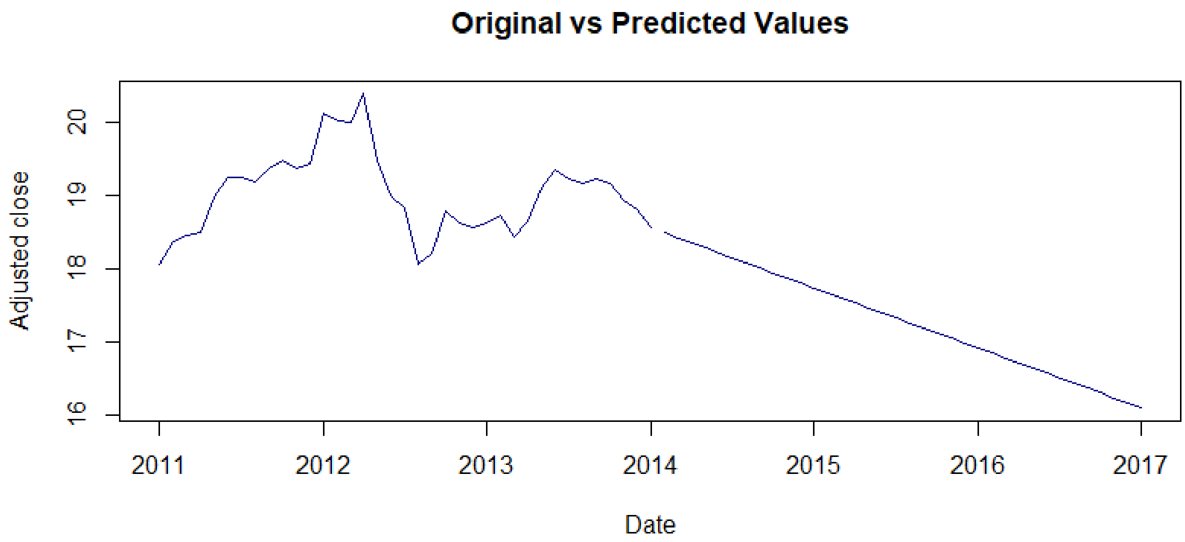

```
> # method 2 : Predict values using forecast()
> J_Pred = forecast(Hotyahoo.ts,h=36)
>
> plot(J_Pred)
```



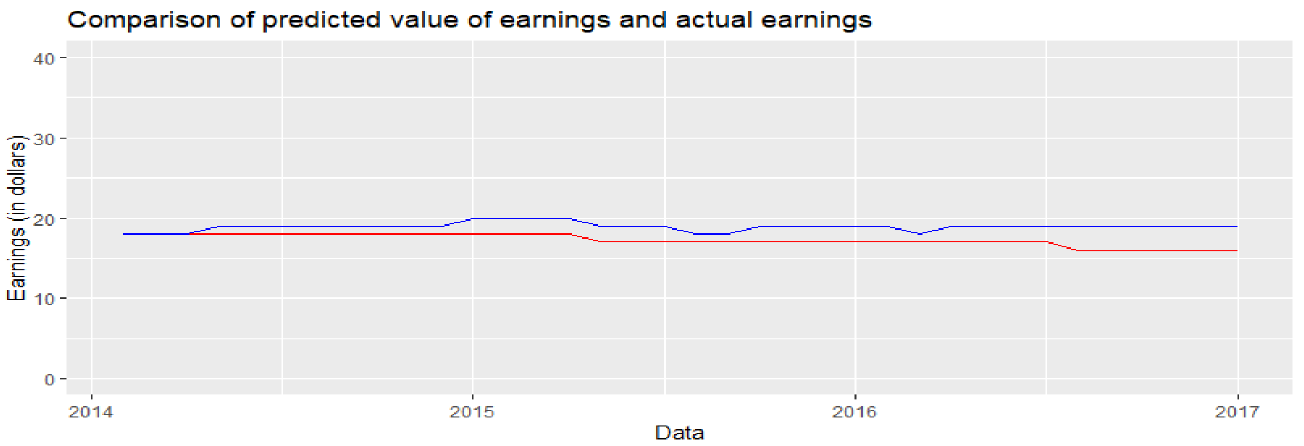

```
> plot(J_Pred$residuals)
```



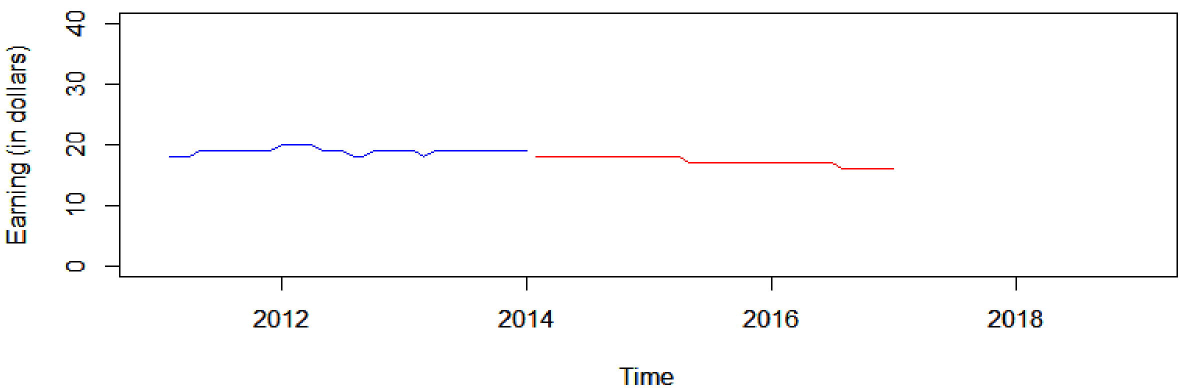
```
> summary(Hotyahoo.ts$fitted)
      xhat            level      
 Min.   :18.05   Min.   :18.05  
 1st Qu.:18.62   1st Qu.:18.62  
 Median :19.05   Median :19.05  
 Mean   :19.03   Mean   :19.03  
 3rd Qu.:19.35   3rd Qu.:19.35  
 Max.   :20.38   Max.   :20.38  

> # Model 3
> H_pred1 = predict(Hotyahoo.ts1,n.ahead = 3*12)
> data3 = round(H_pred1) # predicted values for next 3 years
> data4 = round(tail(yahoo1,36),0) # original values
> 
> ts.plot(yahoo1,H_pred1,col = "navyblue", main = "Original vs Predicted Values") # d
```




```
> 
> 
> # Plot of Predicted value vs Actual Quarterly Earnings for 3 years
> X = time(data3)
> Y1 = data.frame(data3)
> Y2 = data.frame(data4)
> df = tbl_df(data.frame(X,Y1,Y2))
> df
# A tibble: 36 x 3
   X            fit data4
   <ts>       <dbl> <ts> 
 1 2014.083     18  18   
 2 2014.167     18  18   
 3 2014.250     18  18   
 4 2014.333     18  19   
 5 2014.417     18  19   
 6 2014.500     18  19   
 7 2014.583     18  19   
 8 2014.667     18  19   
 9 2014.750     18  19   
10 2014.833     18  19   
# ... with 26 more rows
> 
> #par(mar=c(1,1,1,1))
> ggplot(df,aes(X))+
+   geom_line(aes(y=data3),colour='red')+
+   geom_line(aes(y=data4),colour='blue')+
+   labs(y = "Earnings (in dollars)", x = "Data")+
+   ggtitle("Comparison of predicted value of earnings and actual earnings")+
+   scale_y_continuous(limits=c(0,40))
```

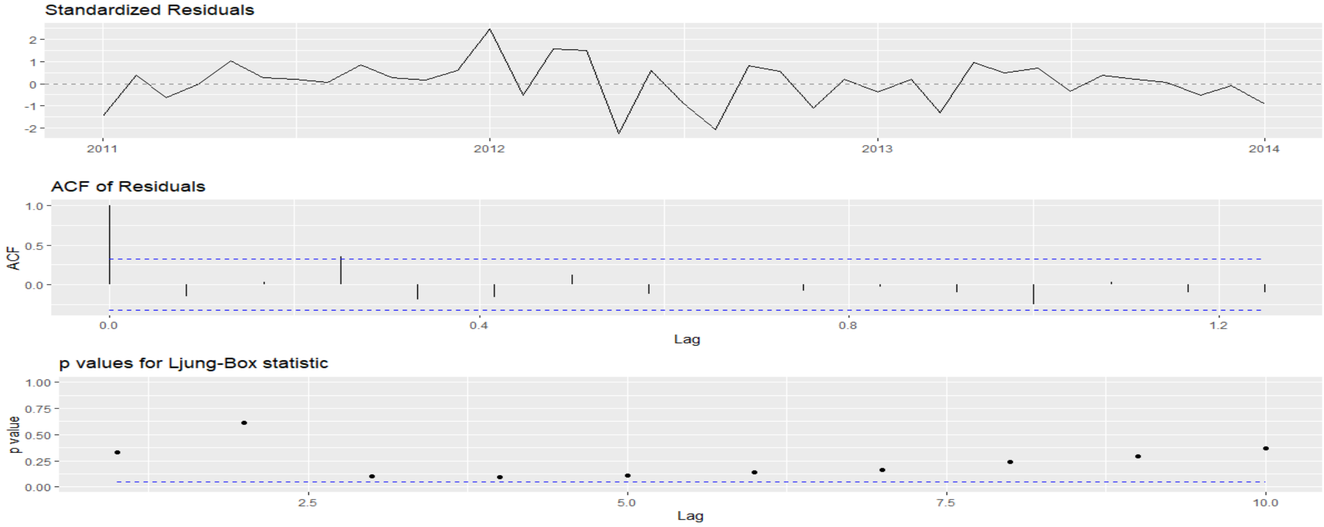



```
>
> # Visualization
> #par(mar=c(1,1,1,1))
> plot.ts(data3, col="red", type ="l",xlim=c(2011,2019),ylim=c(0,40),xlab = "DAta",
 ylab = "Earning (in dollars)")
> par(new=TRUE)
> plot.ts(data4, col="blue",type="l",xlim=c(2011,2019),ylim=c(0,40),xlab = "DAta",
ylab = "Earning (in dollars)")
```

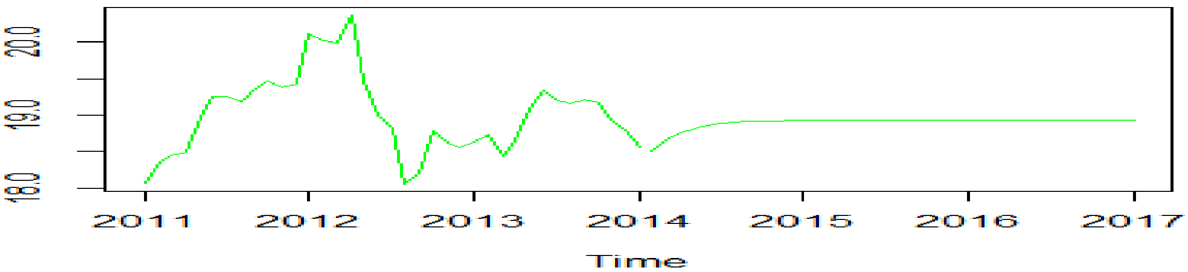



```
> #------------------------------------------
> # 11. RMSE (Root Mean Squared Error)
> # rmse(predicted,actual) Note: Here, data4 contains the actual values of 25% data
>
> # Model 1 :
> HWyahoo.ts$SSE
[1] 5.022137
> H_rmse = rmse(HW_pred,data4)
> H_rmse
[1] 1.579646
>
> # Model 2 :
> Hotyahoo.ts$SSE
[1] 3.746099
> HW_rmse = rmse(H_pred,data4)
> HW_rmse
[1] 4.792509
>
> # Model 3 :
> Hotyahoo.ts1$SSE
[1] 4.092141
> HW1_rmse = rmse(H_pred1,data4)
> HW1_rmse
[1] 1.20867
```

```
#-----------------------------------------
> # 13. Arima Model trained using 75% of the given data (i.e., yahoo.ts_train)
> arimayahoo.ts <- auto.arima(yahoo.ts_train)
> arimayahoo.ts
Series: yahoo.ts_train
ARIMA(1,0,1) with non-zero mean

Coefficients:
         ar1     ma1     mean
      0.6334  0.6615  18.9280
s.e.  0.1500  0.1674   0.2098

sigma^2 estimated as 0.0928:  log likelihood=-7.85
AIC=23.7   AICc=24.95   BIC=30.14
> # Since (p,d,q) = (0,1,1) we know that d = 0 => stationary data,
hence using auot.arima(), we get best arima model
> # Since d = 1, it will automatically differentiate the data
i.e., diff(yahoo1) once to get stationary data
> ggtsdiag(arimayahoo.ts)
```

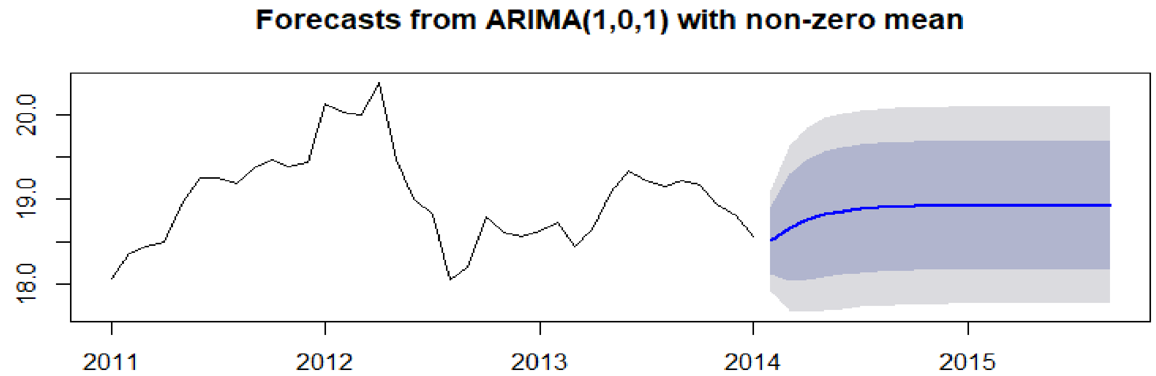



```
> #-----------------------------------------
> # 14. Predicted Values for next 25% of time
> # Using predict()
> predicted = predict(arimayahoo.ts,n.ahead = 3*12)
> predicted # predicted values for next 3 years
$pred
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2014          18.50760 18.66173 18.75935 18.82118 18.86034 18.88514 18.90085 18.91081 18.91711 18.92110 18.92363
2015 18.92523 18.92624 18.92689 18.92729 18.92755 18.92771 18.92782 18.92788 18.92792 18.92795 18.92797 18.92798
2016 18.92798 18.92799 18.92799 18.92799 18.92799 18.92799 18.92799 18.92799 18.92799 18.92799 18.92799 18.92799
2017 18.92799

$se
           Jan       Feb       Mar       Apr       May       Jun       Jul       Aug       Sep       Oct       Nov       Dec
2014           0.3046327 0.4983895 0.5575024 0.5795239 0.5881264 0.5915422 0.5929069 0.5934535 0.5936727 0.5937606 0.5937958
2015 0.5938099 0.5938156 0.5938179 0.5938188 0.5938192 0.5938193 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194
2016 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194 0.5938194
2017 0.5938194

> ts.plot(yahoo1,predicted$pred,col = "green") # dotted predicted lines
```




```
>
```

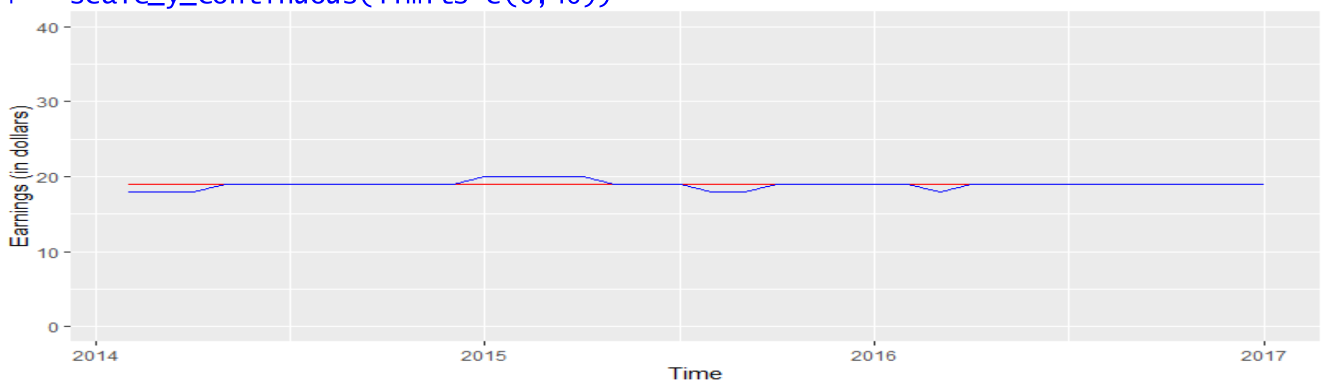
```
> # Using forecast()
> Ar = forecast(arimayahoo.ts,h = 20)
> Ar # 2nd column shows the predicted value and rest 4 columns are predicted value ur
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Feb 2014       18.50760 18.11720 18.89800 17.91053 19.10467
Mar 2014       18.66173 18.02302 19.30044 17.68490 19.63855
Apr 2014       18.75935 18.04488 19.47382 17.66666 19.85203
May 2014       18.82118 18.07849 19.56387 17.68533 19.95703
Jun 2014       18.86034 18.10663 19.61406 17.70763 20.01305
Jul 2014       18.88514 18.12705 19.64324 17.72574 20.04455
Aug 2014       18.90085 18.14101 19.66070 17.73878 20.06293
Sep 2014       18.91081 18.15026 19.67135 17.74766 20.07395
Oct 2014       18.91711 18.15629 19.67793 17.75353 20.08068
Nov 2014       18.92110 18.16016 19.68203 17.75735 20.08485
Dec 2014       18.92363 18.16265 19.68461 17.75981 20.08745
Jan 2015       18.92523 18.16423 19.68623 17.76138 20.08907
Feb 2015       18.92624 18.16524 19.68725 17.76239 20.09010
Mar 2015       18.92689 18.16588 19.68789 17.76302 20.09075
Apr 2015       18.92729 18.16628 19.68830 17.76343 20.09116
May 2015       18.92755 18.16654 19.68856 17.76369 20.09141
Jun 2015       18.92771 18.16670 19.68872 17.76385 20.09158
Jul 2015       18.92782 18.16681 19.68883 17.76395 20.09168
Aug 2015       18.92788 18.16687 19.68889 17.76402 20.09175
Sep 2015       18.92792 18.16691 19.68893 17.76406 20.09179
> plot(Ar)
```



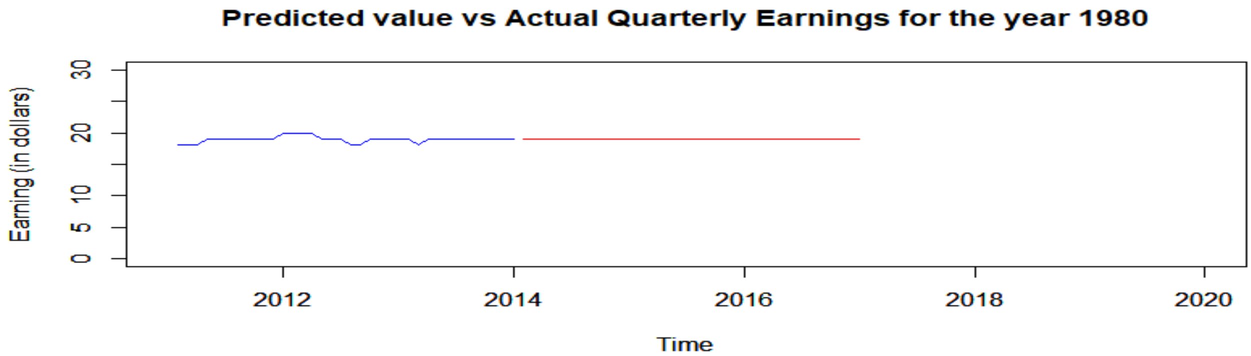

```
> JArPredict = forecast(arimayahoo.ts,h = 20)
> JArPredict
> #plot(JArPredict$residuals)
> #qqnorm(JArPredict$residuals)
>
> # Testing ARIMA model
> data1 = round(tail(predicted$pred,36),0) # predicted values
> data2 = round(tail(yahoo1,36),0) # original values
> data1
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2014      19  19  19  19  19  19  19  19  19  19  19
2015  19  19  19  19  19  19  19  19  19  19  19  19
2016  19  19  19  19  19  19  19  19  19  19  19  19
2017  19
> data2
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2011      18  18  18  19  19  19  19  19  19  19  19
2012  20  20  20  20  19  19  19  18  18  19  19  19
2013  19  19  18  19  19  19  19  19  19  19  19  19
2014  19
```

```
> #---------------------------------------------------
> # 15.Visualization:Plot predicted values with actual values for better comparison
> x = time(data1)
> y1 = data.frame(data1)
> y2 = data.frame(data2)
> df = tbl_df(data.frame(x,y1,y2))
> df
# A tibble: 36 x 3
   x        data1 data2
   <ts>     <ts>  <ts>
 1 2014.083 19    18
```

```
 2 2014.167 19     18
 3 2014.250 19     18
 4 2014.333 19     19
 5 2014.417 19     19
 6 2014.500 19     19
 7 2014.583 19     19
 8 2014.667 19     19
 9 2014.750 19     19
10 2014.833 19     19
# ... with 26 more rows
> # Plot of Predicted value vs Actual Quarterly Earnings for 3 years i.e., 2011-14
> ggplot(df,aes(x,hp))+
+   geom_line(aes(y=data1),colour='red')+
+   geom_line(aes(y=data2),colour='blue')+
+   labs(y = "Earnings (in dollars)", x = "Time")+
+   scale_y_continuous(limits=c(0,40))
```

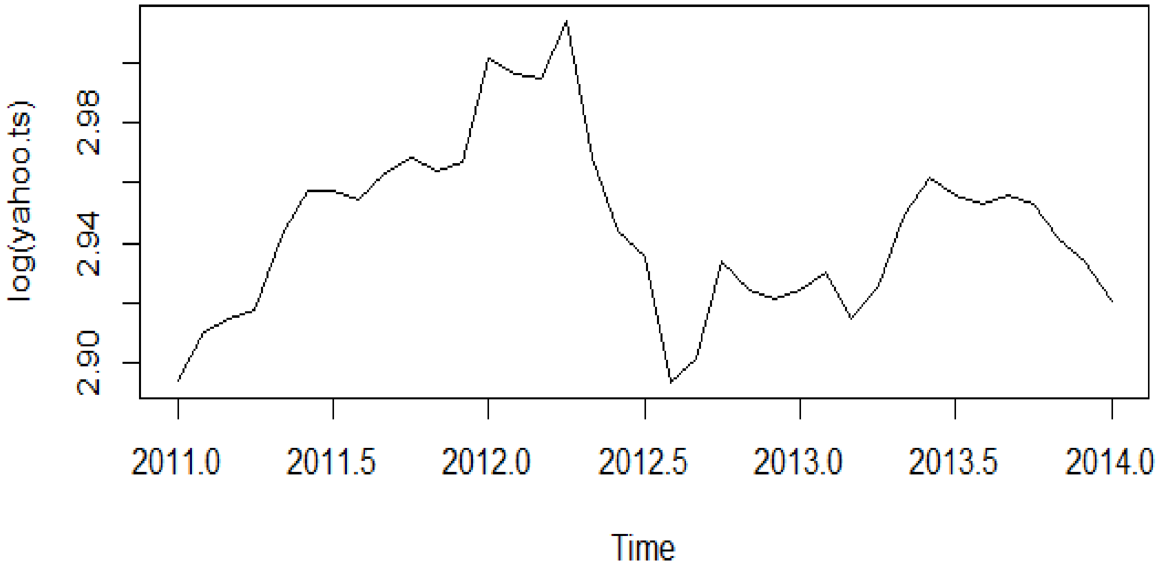



```
>      geom_smooth(method="lm")
geom_smooth: na.rm = FALSE, orientation = NA, se = TRUE
stat_smooth: na.rm = FALSE, orientation = NA, se = TRUE, method = lm
position_identity
> plot.ts(data1, col="red", type = "l",xlim=c(2011,2014),ylim=c(0,20),
 ylab = "Earning (in dollars)")
> par(new=TRUE)
> plot(data2, col="blue",type="l",xlim=c(2011,12014),ylim=c(0,20),
ylab = "Earning(in dollars)"
,main = "Predicted value vs Actual Quarterly Earnings for the year 1980")
```

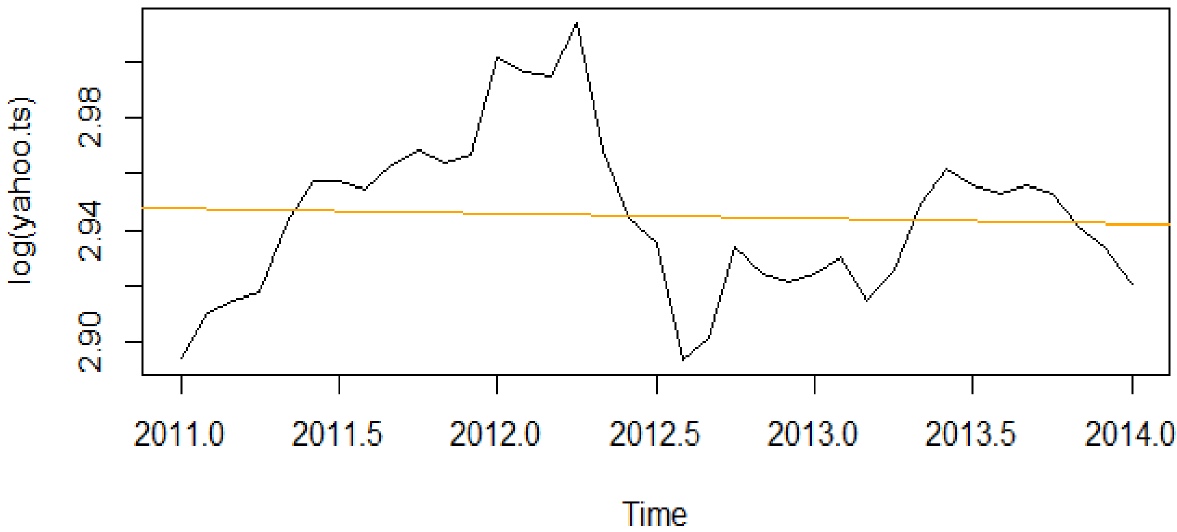



```
> #---------------------------------------------
> # 16. Accuracy of ARIMA Model (using auto.arima())
> summary(arimayahoo.ts)
Series: yahoo.ts_train
ARIMA(1,0,1) with non-zero mean

Coefficients:
         ar1     ma1     mean
      0.6334  0.6615  18.9280
s.e.  0.1500  0.1674   0.2098
```

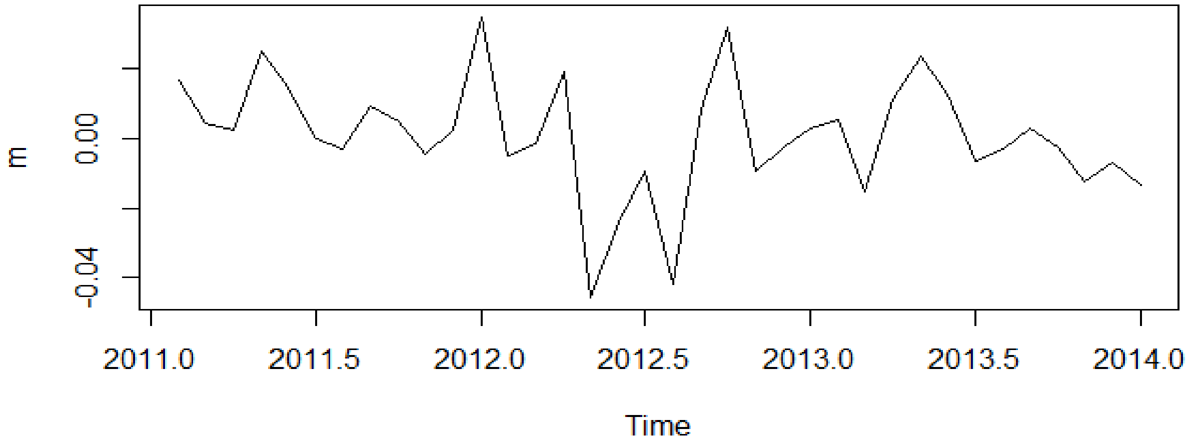
```
sigma^2 estimated as 0.0928:  log likelihood=-7.85
AIC=23.7   AICc=24.95   BIC=30.14

Training set error measures:
                    ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set 0.0169883 0.2920217 0.2231868 0.0666217 1.171267 0.2399256 -0.1523175
> accuracy(arimayahoo.ts) #: this is a function for arima model so to run this we need to unload "Metrics" library
                    ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set 0.0169883 0.2920217 0.2231868 0.0666217 1.171267 0.2399256 -0.1523175
> rmse(data1,data2)
[1] NaN

> #-----------------------------------------------
> # Ques17. Tuning the model by manually giving (p,d,q) values
> # p : AR (Auto-regressive model), d : I (Difference/ Integration part),
  q : MA (Moving average time)
>
> # First make the Data Stationary
> plot(log(yahoo.ts)) # homongenizing variance
```



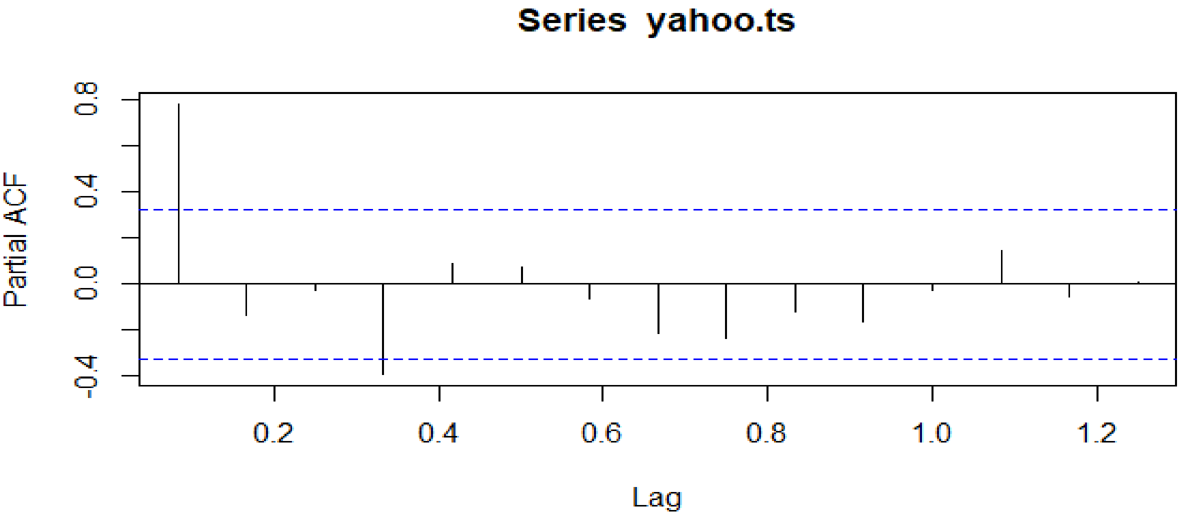

```
> abline(reg = lm(log(yahoo.ts)~time(yahoo.ts)),col="orange")
```



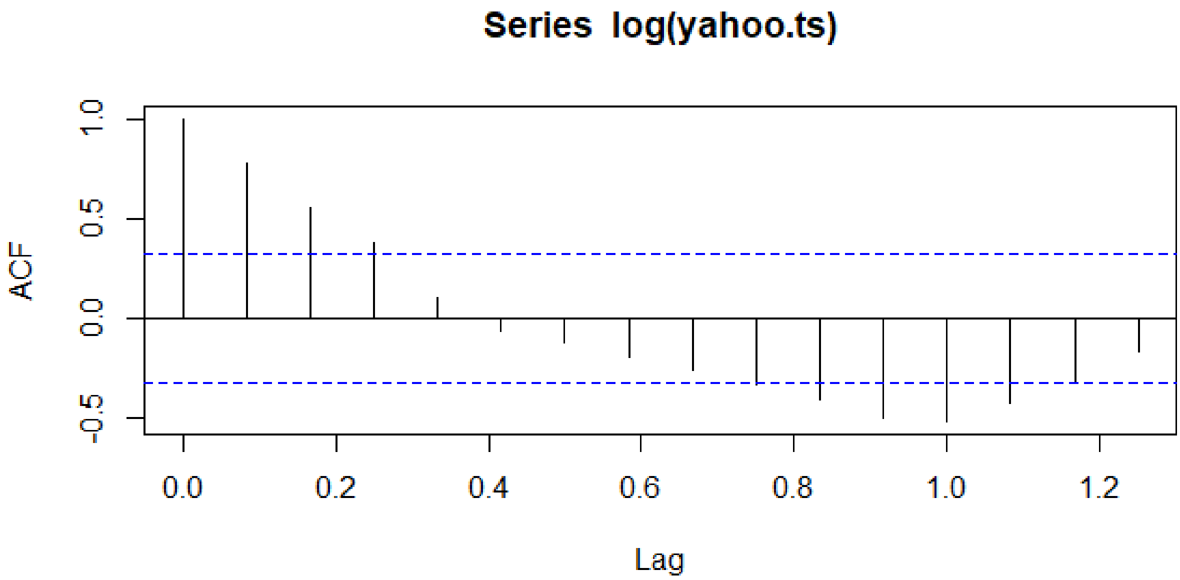
```
> m = diff(log(yahoo.ts))
> plot(m) # homogenizing mean
```



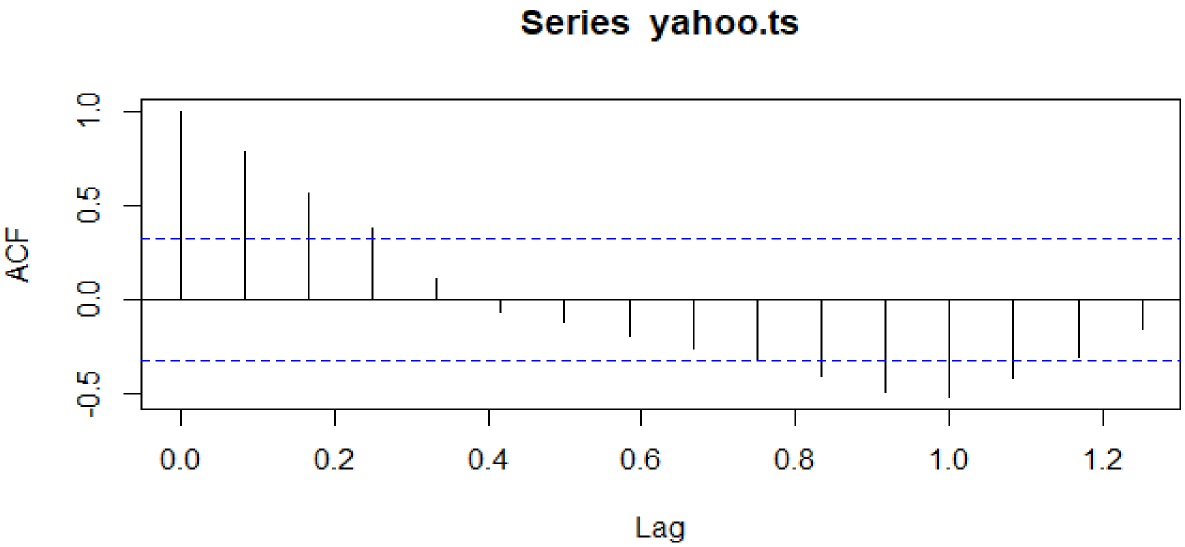

```
> # Preprocessing
> pacf(yahoo.ts) # q = 1
```



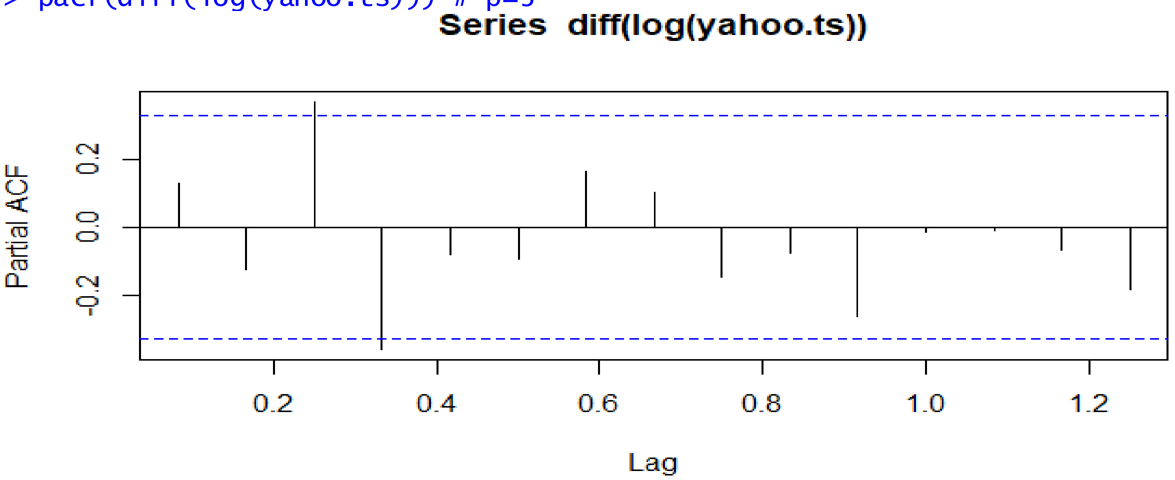

```
> pacf = acf(log(yahoo.ts))
```



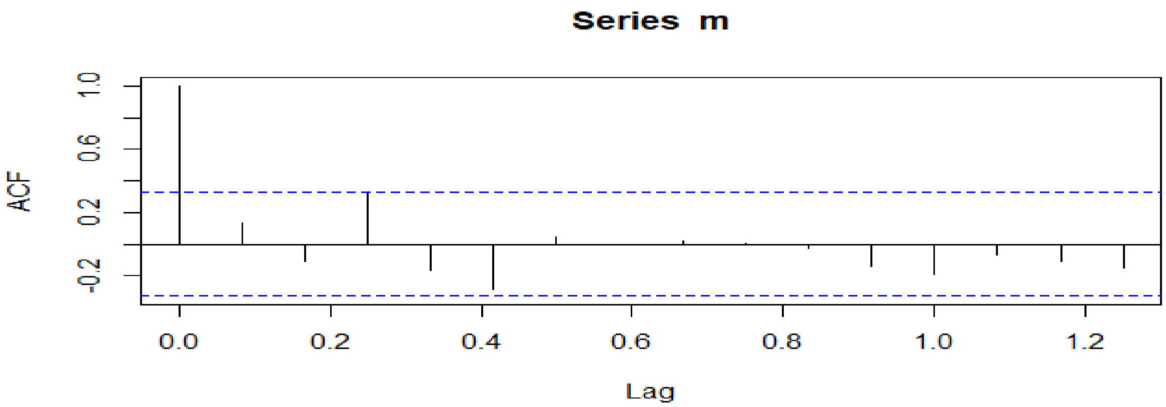
```
> acf(yahoo.ts,plot =TRUE )
```




```
> # Each observation is positively associated with its recent past at
 least through 4 lags.
> pacf(diff(log(yahoo.ts))) # p=3
```

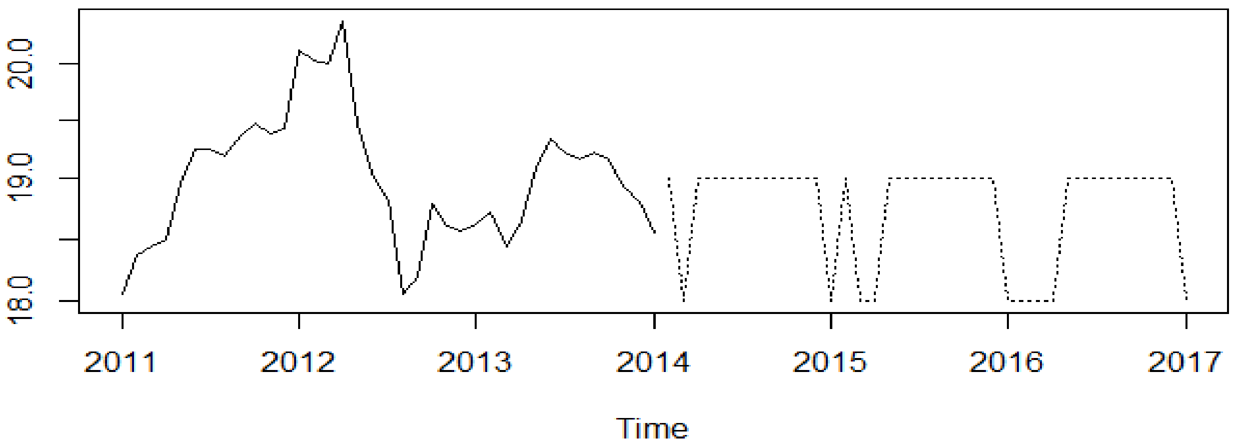



```
> acf(m) # q =1
```



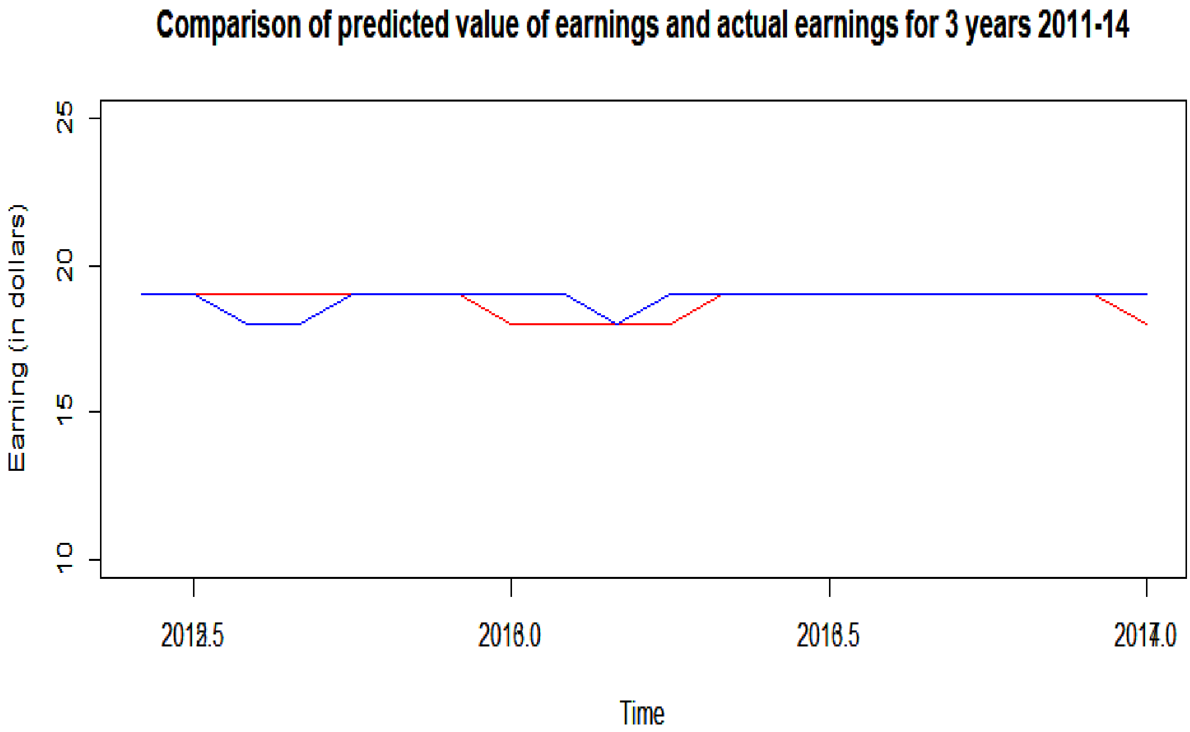

```
>
> # First Try :
> fit=arima(log(yahoo.ts_train),c(3,1,1),seasonal = list(order=c(3,1,1),period=4))
> fit

Call:
arima(x = log(yahoo.ts_train), order = c(2, 1, 1), seasonal = list(order = c(0,
```

```
    1, 0), period = 12))

Coefficients:
          ar1      ar2     ma1
      -0.4076  -0.0917  0.7673
s.e.   0.2576   0.2190  0.1744

sigma^2 estimated as 0.0007023:  log likelihood = 52.85,  aic = -97.7

> pred = predict(fit,n.ahead = 3*12)
> pred # next 3 ten years pred values(in log)
$pred
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
2014          2.920978 2.914396 2.899990 2.902381 2.906865 2.892601 2.887789 2.890516 2.892816 2.884466 2.879260
2015 2.882832 2.886499 2.877073 2.872695 2.874501 2.878346 2.870279 2.864585 2.867764 2.871930 2.862661 2.857919
2016 2.860587 2.864462 2.855997 2.850889 2.853952 2.858024 2.849164 2.844351 2.847152 2.851163 2.842602 2.837588
2017 2.840587

$se
            Jan        Feb        Mar        Apr        May        Jun        Jul        Aug        Sep        Oct        Nov        Dec
2014            0.01408325 0.02320874 0.02929884 0.03872597 0.04663397 0.05273069 0.05995021 0.06688416 0.07448822 0.08271928 0.09055339
2015 0.09838232 0.10619351 0.11364449 0.12092136 0.12809350 0.13605119 0.14402794 0.15171930 0.15964122 0.16779334 0.17571235 0.18357559
2016 0.19146876 0.19981853 0.20816914 0.21637213 0.22469087 0.23330475 0.24180831 0.25022377 0.25870156 0.26755893 0.27638263 0.28509685
2017 0.29391263

> pred1 = round(2.718^pred$pred,0)
> pred1
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2014      19  18  19  19  19  19  19  19  19  19  19
2015  18  19  18  18  19  19  19  19  19  19  19  19
2016  18  18  18  18  19  19  19  19  19  19  19  19
2017  18
> ts.plot(yahoo.ts,pred1,log="y",lty=c(1,3))#dotted predicted lines
```

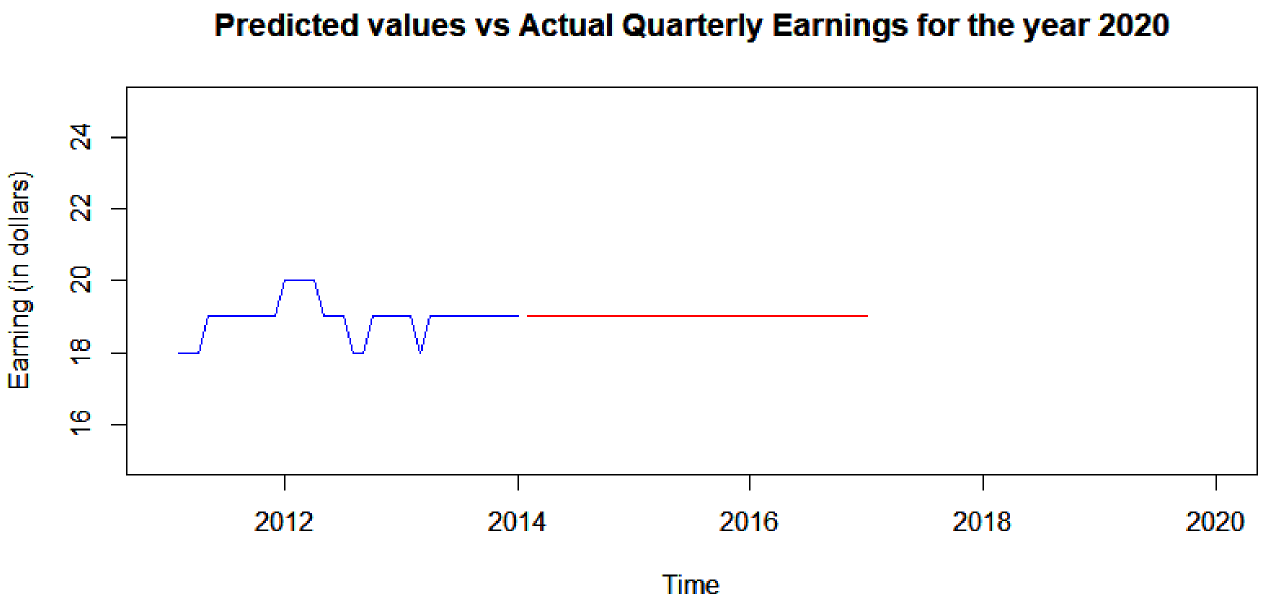



```
>
> # Testing the model
> data11 = round(tail(pred1,20),0)#predicted values
> data22 = round(tail(yahoo.ts,20),0)#original values
> data11
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2015                      19  19  19  19  19  19  19
2016  18  18  18  18  19  19  19  19  19  19  19  19
2017  18

> data22
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2012                      19  19  18  18  19  19  19
2013  19  19  18  19  19  19  19  19  19  19  19  19
2014  19

> par(mfrow = c(2,1))
> # Visualization
> par(mar=c(1,1,1,1))
```

```
> plot.ts(data11, col="red", type = "l",ylim=c(0,20), ylab = "Earning (in dollars)"
rnings for 3 years 2011-14")
> par(new=TRUE)
> plot.ts(data22, col="blue",type="l",ylim=c(0,20), ylab = "Earning (in dollars)" ,
ings for 3 years 2011-14")
```

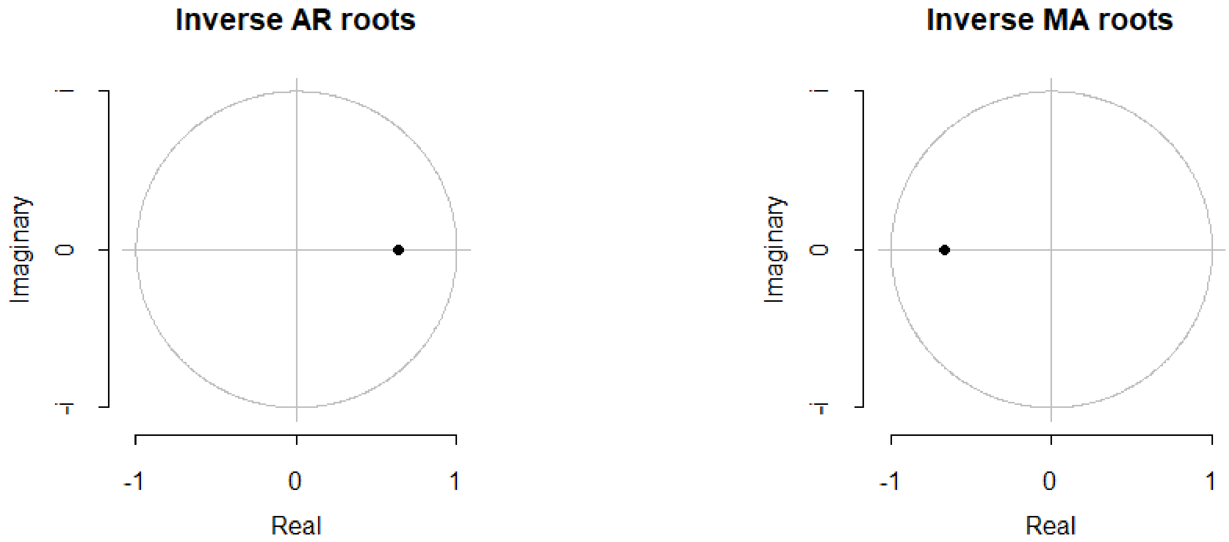



```
>
> plot.ts(data1, col="red", type = "l",xlim=c(2014,2015),ylim=c(0,20),
 ylab = "Earning (in dollars)", main = "Predicted values vs Actual Quarterly
Earnings for the year 2014")
> par(new=TRUE)
> plot(data2, col="blue",type="l",xlim=c(2014,2015),ylim=c(0,20),
ylab = "Earning (in dollars)", main = "Predicted values vs Actual Quarterly
Earnings for the year 2014")
```

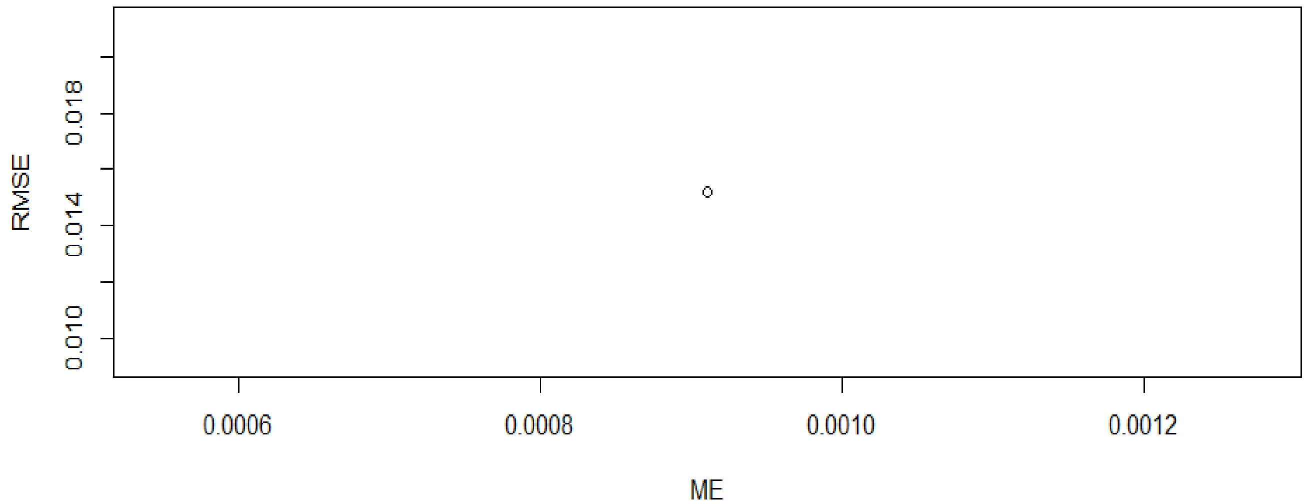



```
>
```

```
> # Accuracy of model
> # accuracy(fit)
> rmse(pred1,data2)
[1] 0.5
> accuracy(pred1,data2)
           ME RMSE  MAE      MPE     MAPE     ACF1 Theil's U
Test set 0.25  0.5 0.25 1.315789 1.315789 0.287037       Inf
>
> # Second Try :
> yahoo.ts_fit <- auto.arima(log(yahoo.ts_train))
> plot(yahoo.ts_fit)
```



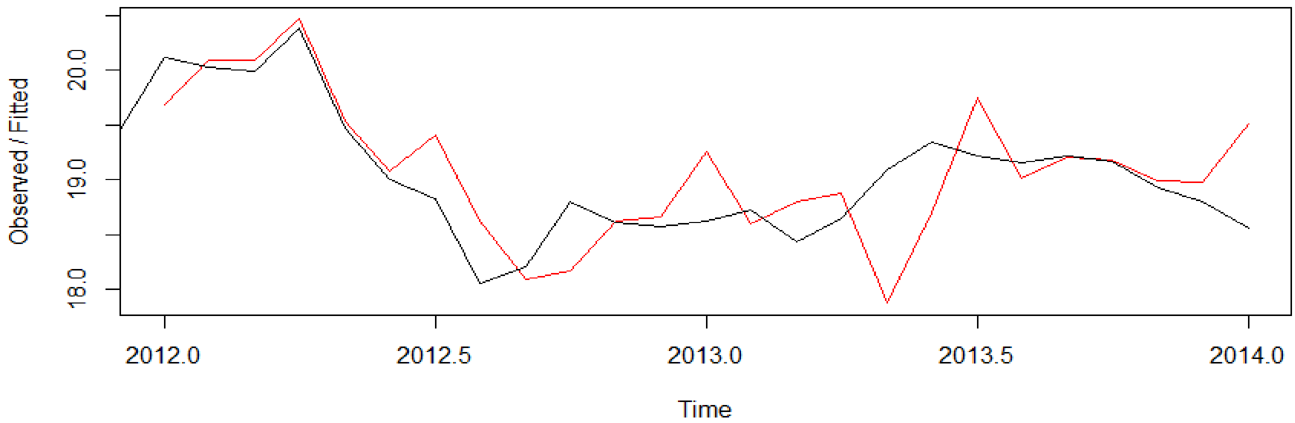

```
> accuracy(yahoo.ts_fit)
                       ME       RMSE        MAE        MPE     MAPE      MASE
Training set 0.0009111758 0.01519777 0.01167771 0.02836161 0.396319 0.2398298
                   ACF1
Training set -0.1537559
> plot(accuracy(yahoo.ts_fit))
```




```
> # 19. Raw data Vs Cleaned data
> par(mfrow = c(2,1))
> HW <- HoltWinters(yahoo.ts,seasonal = "multiplicative") # This will give same fitt
> HW
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
```

```
Call:
HoltWinters(x = yahoo.ts, seasonal = "multiplicative")

Smoothing parameters:
 alpha: 1
 beta : 0
 gamma: 0

Coefficients:
            [,1]
a    18.239660883
b     0.004903408
s1    1.015668389
s2    1.019186321
s3    1.043008989
s4    0.999454387
s5    0.978987776
s6    0.999436022
s7    0.988313476
s8    0.990611005
s9    0.988368376
s10   0.978848762
s11   0.980824064
s12   1.017292433
> plot(HW, main = "Original time series against the Fitted time series : Raw Data")
```

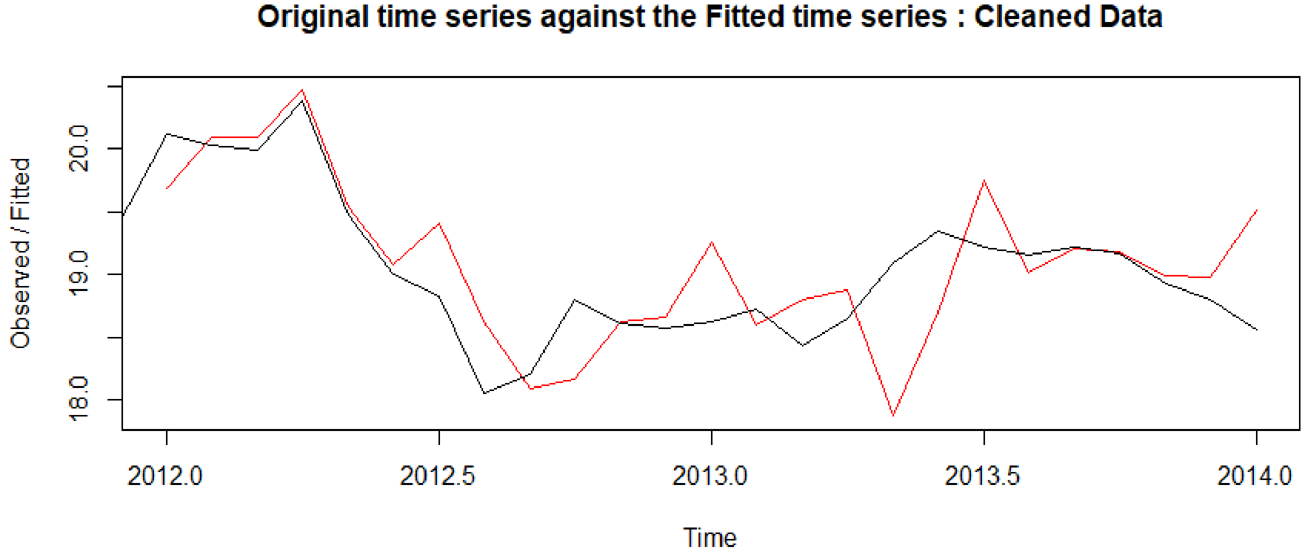
Original time series against the Fitted time series : Raw Data




```
> HWC <- HoltWinters(yahoo1,seasonal = "multiplicative") # This will give same fitti
> HWC
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
Call:
HoltWinters(x = yahoo1, seasonal = "multiplicative")

Smoothing parameters:
 alpha: 1
 beta : 0
 gamma: 0

Coefficients:
            [,1]
a    18.239660883
b     0.004903408
s1    1.015668389
s2    1.019186321
s3    1.043008989
s4    0.999454387
s5    0.978987776
```

```
s6   0.999436022
s7   0.988313476
s8   0.990611005
s9   0.988368376
s10  0.978848762
s11  0.980824064
s12  1.017292433
```

```
> plot(HWC,main="Original time series against the Fitted time series:Cleaned Data")
```




```
> HW$SSE
[1] 5.022137
> HWC$SSE
[1] 5.022137
```