Boğaziçi University

Industrial Engineering Department

IE 360

Statistical Forecasting & Time Series

Project Report

Group Number:	15
Group Members:	Alperen Yıldız
	Gülce Karabacak
	İrem Yazgan
	Yavuzhan Yavuz
Instructor:	Refik Güllü
Due date:	May 28, 2019

Initialization of packages & forecast data:

```
library(forecast)

library(readxl)

library(ggplot2)

windows()

setwd("C:/Users/YAZGAN/Desktop/360 final proj")

data <- read_xls("C:/Users/hp/Desktop/Project/beer.xls")
```

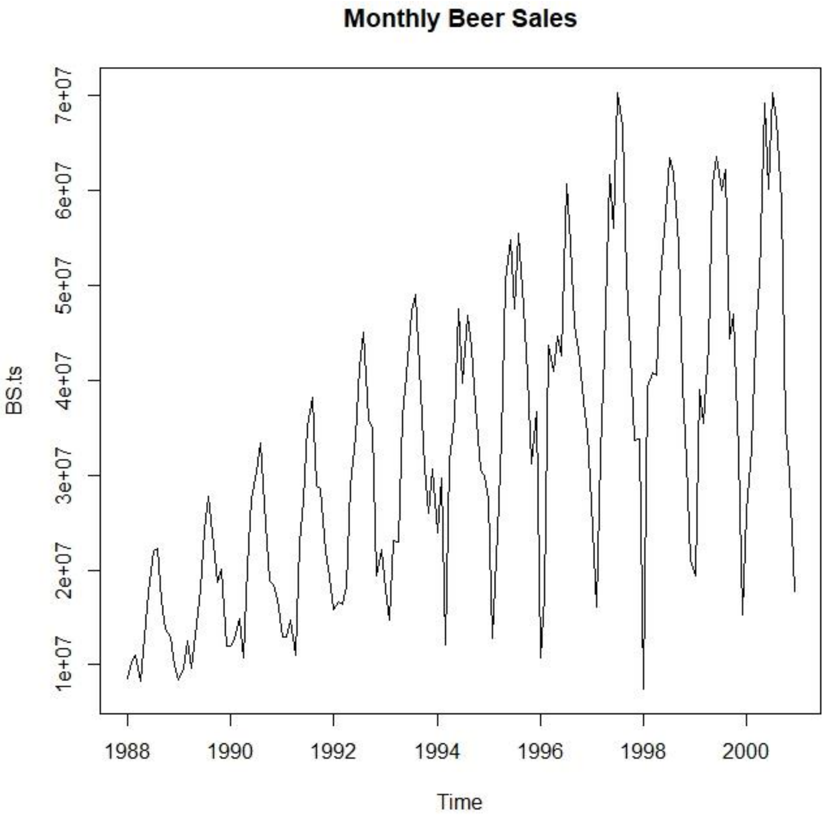
In the R code above, excel data is imported into R via "readxl" package and setup is completed. Windows function is being used for the sake of better graphics on plots.

1. Plot the time series of "Beer Sales". Comment on the shape of the time series. Specifically, do you think the time series is stationary with respect to its mean and variance?

```
BS<-data[,2]

BS.ts<-ts(BS,freq=12,start=c(1988,1))

plot(BS.ts)
```

Since given data is aggregate and contains extra information and related factors with sales data, firstly sales data is extracted and followingly transformed into time series data on monthly basis.

Plot of time series of Beer Sales:



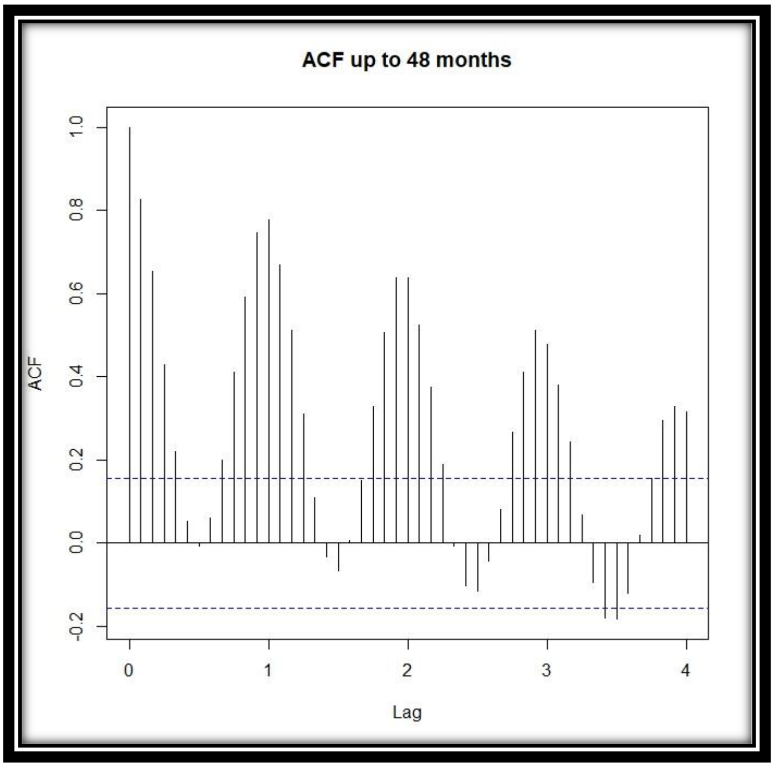
When the shape of Beer Sales plot is analyzed, one can see the obvious positive trend by year and yearly seasonality. The increase in variance of seasonal shifts should be noticed also.

For the stationary decision, we should check stationarity conditions, which are constant mean and constant variance. Violation of one of these conditions is enough to come up with the nonstationary decision. The plot above evidently reveals that the mean of the time series is increasing as the time and so does the variance. Therefore, it is concluded that the data is not stationary.

2. Plot the autocorrelation function of the time series (get autocorrelations for at least 24 lags). What do you think the autocorrelation values at different lags indicate?

The R code below is utilized for omitting NA's in the vectors and plotting auto correlation functions of the time series of beer sales.

```
BS.ts<-na.omit(BS.ts)

acf(BS.ts,lag.max = 48)

pacf(BS.ts,lag.max = 48)
```

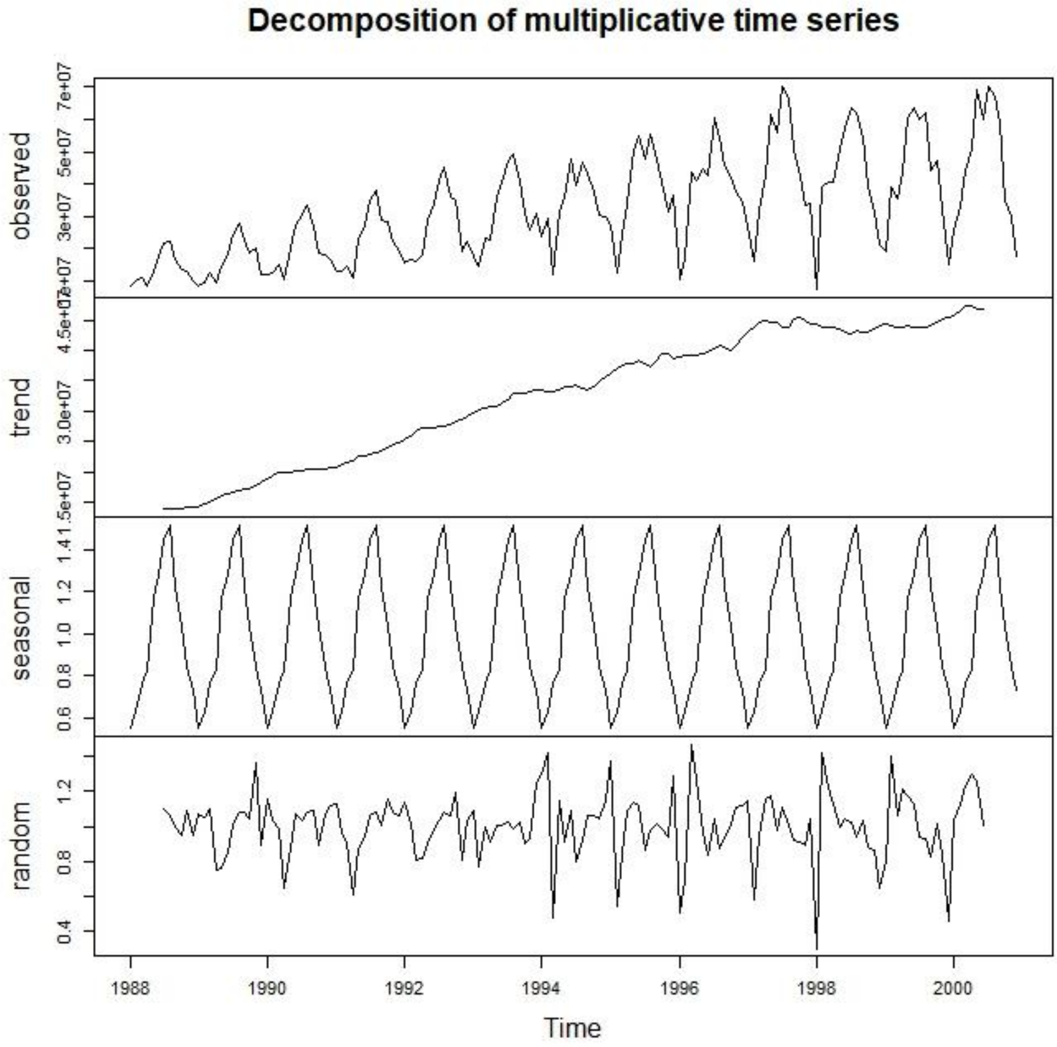


When the ACF plot is analyzed, significant spikes at 12,24,36... months can be seen and they are positive. This indicates positive seasonality on yearly basis with lag=12.

Additional analysis:

```
bs.dec <- decompose(BS.ts, type="multiplicative")

plot(bs.dec)
```

Since the amplitude of the seasonal effect seems to increase as t increases, assuming seasonal effect acts proportionally rather than constant difference per year might be more useful in correct analysis approach. That's why multiplicative model type is used in decomposing.



As decomposed model graphs show, obvious trend and significant seasonal effect are proven by decomposition model. Linear/time-varying increase in trend should be reduced with differencing method and the multiplicative progress can be eliminated with logarithmic transformation.

Method A: FORECASTING WITH REGRESSION

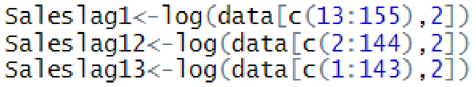
1. Preliminary Transformation

Since the model is multiplicative, logarithmic function is used to turn the model to an additive process.

2. Definition of new variables

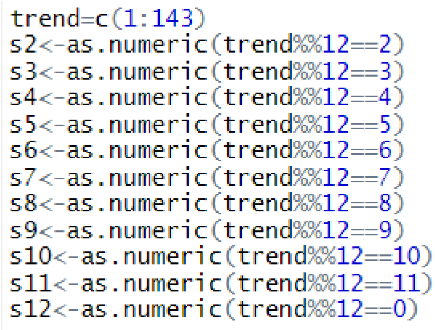
There are two way to include variables for seasonality and trend:

1. One way is using lagged variables for trend and seasonality. For trend Y_{t-1} variable and for seasonality Y_{t-12} and Y_{t-13} variables are created.



```
Saleslag1<-log(data[c(13:155),2])
Saleslag12<-log(data[c(2:144),2])
Saleslag13<-log(data[c(1:143),2])
```

2. Other way is creating different vectors for trend and seasonality:



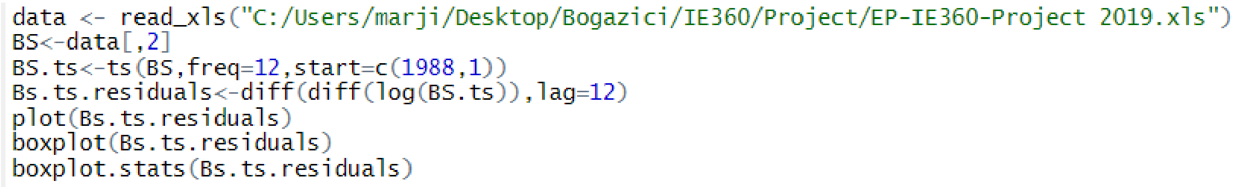
```
trend=c(1:143)
s2<-as.numeric(trend%%12==2)
s3<-as.numeric(trend%%12==3)
s4<-as.numeric(trend%%12==4)
s5<-as.numeric(trend%%12==5)
s6<-as.numeric(trend%%12==6)
s7<-as.numeric(trend%%12==7)
s8<-as.numeric(trend%%12==8)
s9<-as.numeric(trend%%12==9)
s10<-as.numeric(trend%%12==10)
s11<-as.numeric(trend%%12==11)
s12<-as.numeric(trend%%12==0)
```

The length of the vector is 143, because:

- The models will be compared and because of Y_{t-13} variable the regression starts from 14th variable.
- The data set contains 156 observations of sales value.

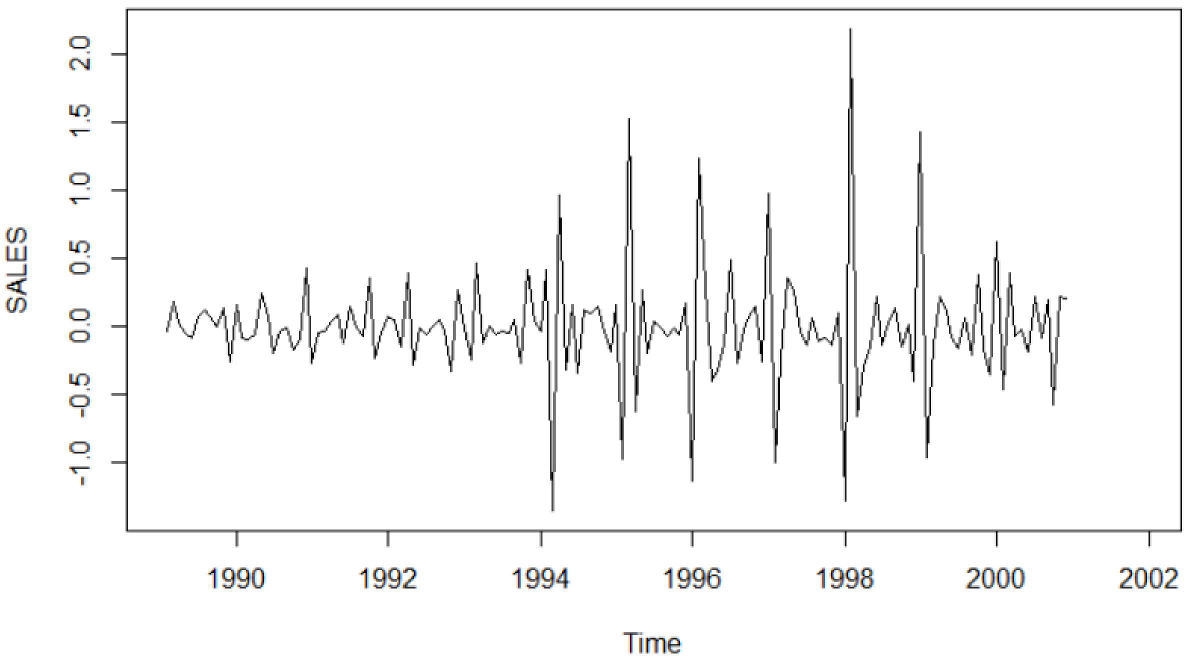
3. Detection of elements to be extracted

Since the data is multiplicative and consists trend and seasonality, there is need for logarithmic for turning it to a additive process and differentials for removing trend and seasonality. After these processes outlier analysis can be made.

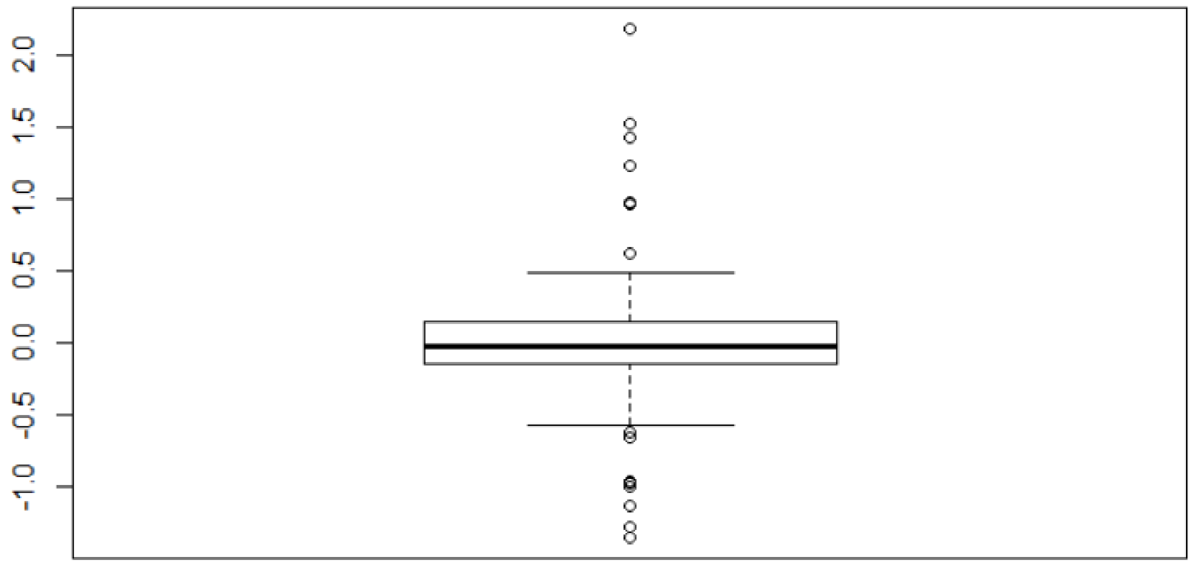


```
data <- read_xls("C:/Users/marji/Desktop/Bogazici/IE360/Project/EP-IE360-Project 2019.xls")
BS<-data[,2]
BS.ts<-ts(BS,freq=12,start=c(1988,1))
Bs.ts.residuals<-diff(diff(log(BS.ts)),lag=12)
plot(Bs.ts.residuals)
boxplot(Bs.ts.residuals)
boxplot.stats(Bs.ts.residuals)
```

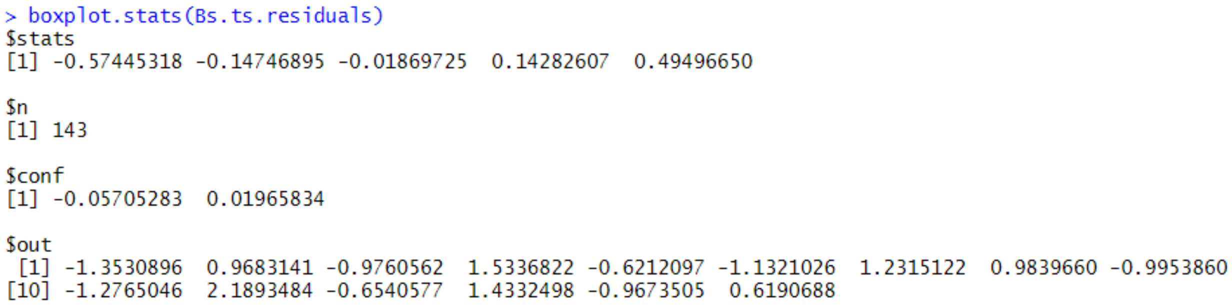
Additive time series with trend and seasonality removed: Some outliers seems to exist:



Box plot of the data:

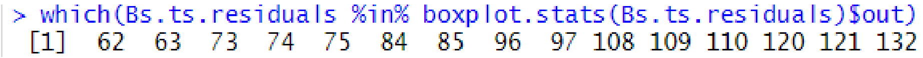


Statistical approach to outliers.



```
> boxplot.stats(Bs.ts.residuals)
$stats
[1] -0.57445318 -0.14746895 -0.01869725  0.14282607  0.49496650

$n
[1] 143

$conf
[1] -0.05705283  0.01965834

$out
 [1] -1.3530896  0.9683141 -0.9760562  1.5336822 -0.6212097 -1.1321026  1.2315122  0.9839660 -0.9953860
[10] -1.2765046  2.1893484 -0.6540577  1.4332498 -0.9673505  0.6190688
```

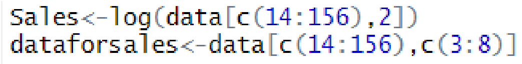
These outlier elements can be extracted from the data.



```
> which(Bs.ts.residuals %in% boxplot.stats(Bs.ts.residuals)$out)
 [1]  62  63  73  74  75  84  85  96  97 108 109 110 120 121 132
```

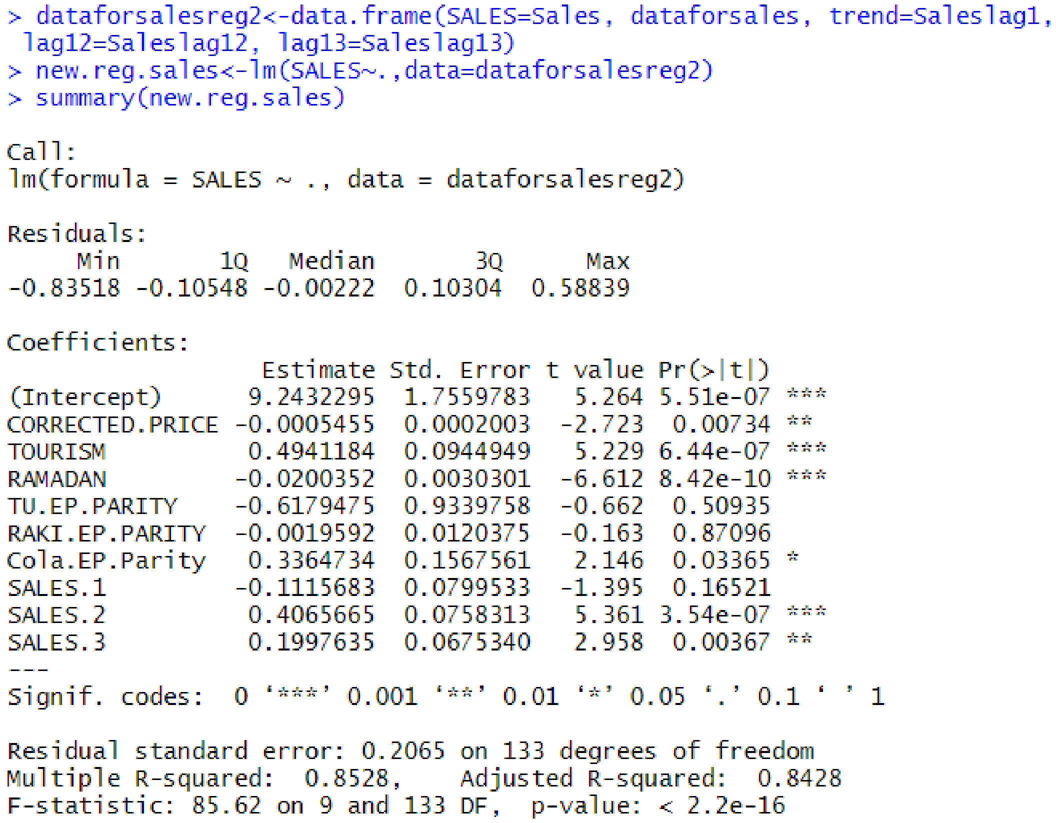
4. Model

Sales are fitted into model as logarithms and all the explanatory variables are used.



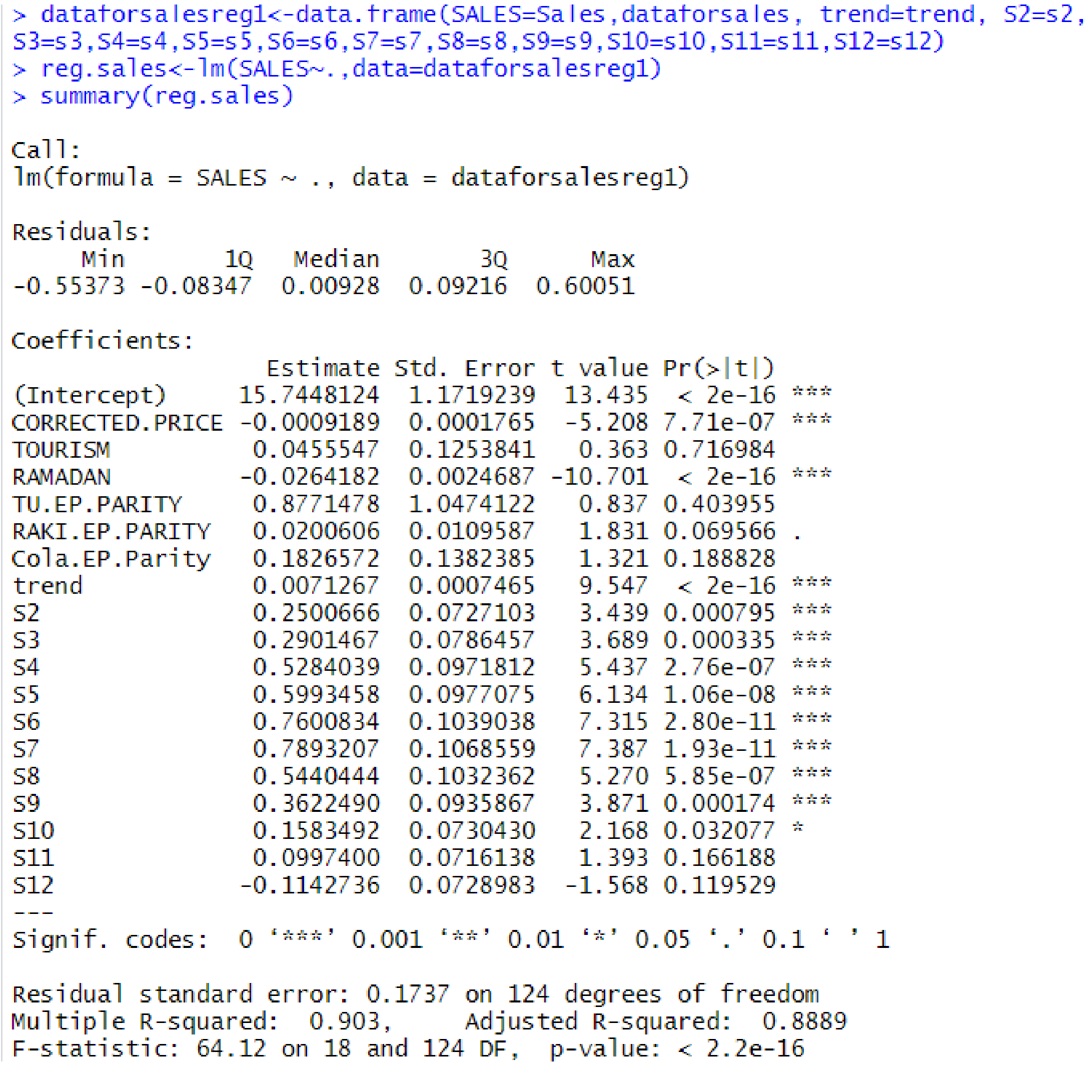
```
Sales<-log(data[c(14:156),2])
dataforsales<-data[c(14:156),c(3:8)]
```

Lagged variables for trend and seasonality:



```
> dataforsalesreg2<-data.frame(SALES=Sales, dataforsales, trend=Saleslag1,
  lag12=Saleslag12, lag13=Saleslag13)
> new.reg.sales<-lm(SALES~.,data=dataforsalesreg2)
> summary(new.reg.sales)

Call:
lm(formula = SALES ~ ., data = dataforsalesreg2)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.83518 -0.10548 -0.00222  0.10304  0.58839 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      9.2432295  1.7559783   5.264 5.51e-07 ***
CORRECTED.PRICE -0.0005455  0.0002003  -2.723  0.00734 ** 
TOURISM          0.4941184  0.0944949   5.229 6.44e-07 ***
RAMADAN         -0.0200352  0.0030301  -6.612 8.42e-10 ***
TU.EP.PARITY    -0.6179475  0.9339758  -0.662  0.50935    
RAKI.EP.PARITY  -0.0019592  0.0120375  -0.163  0.87096    
Cola.EP.Parity   0.3364734  0.1567561   2.146  0.03365 *  
SALES.1         -0.1115683  0.0799533  -1.395  0.16521    
SALES.2          0.4065665  0.0758313   5.361 3.54e-07 ***
SALES.3          0.1997635  0.0675340   2.958  0.00367 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2065 on 133 degrees of freedom
Multiple R-squared:  0.8528,	Adjusted R-squared:  0.8428 
F-statistic: 85.62 on 9 and 133 DF,  p-value: < 2.2e-16
```

Vectors for trend and seasonality:

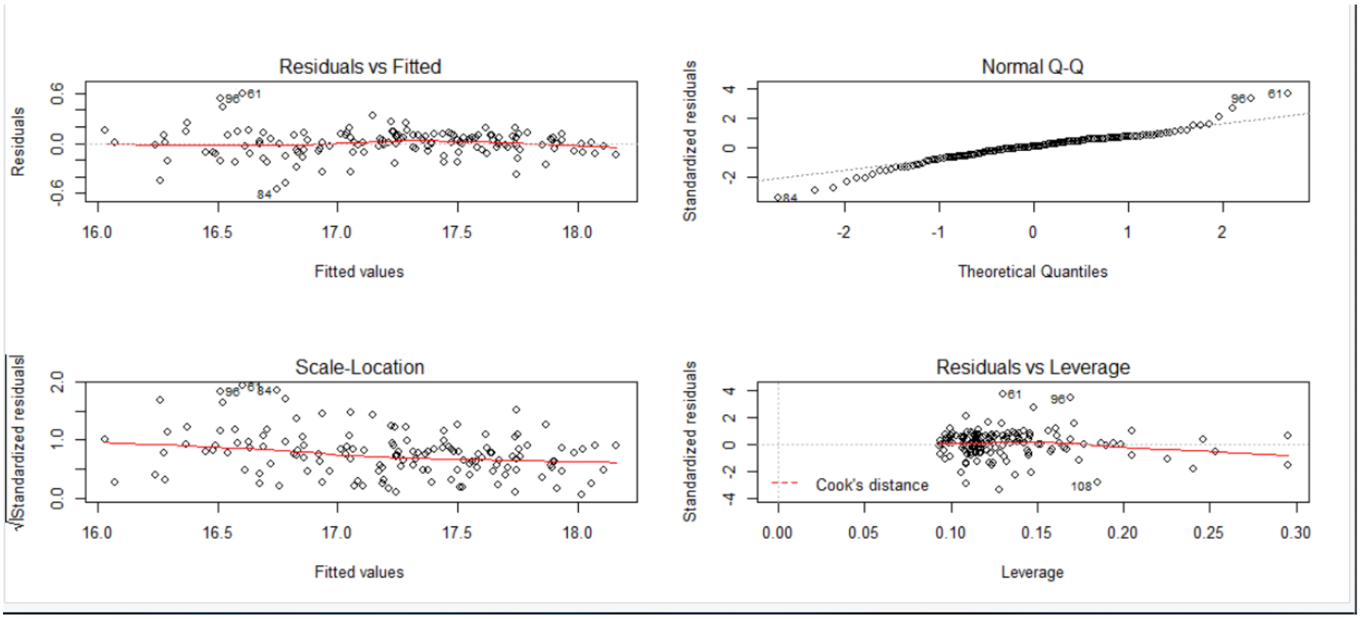


```
> dataforsalesreg1<-data.frame(SALES=Sales,dataforsales, trend=trend, S2=s2,
S3=s3,S4=s4,S5=s5,S6=s6,S7=s7,S8=s8,S9=s9,S10=s10,S11=s11,S12=s12)
> reg.sales<-lm(SALES~.,data=dataforsalesreg1)
> summary(reg.sales)

Call:
lm(formula = SALES ~ ., data = dataforsalesreg1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.55373 -0.08347  0.00928  0.09216  0.60051 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     15.7448124  1.1719239  13.435  < 2e-16 ***
CORRECTED.PRICE -0.0009189  0.0001765  -5.208 7.71e-07 ***
TOURISM          0.0455547  0.1253841   0.363 0.716984    
RAMADAN         -0.0264182  0.0024687 -10.701  < 2e-16 ***
TU.EP.PARITY     0.8771478  1.0474122   0.837 0.403955    
RAKI.EP.PARITY   0.0200606  0.0109587   1.831 0.069566 .  
Cola.EP.Parity   0.1826572  0.1382385   1.321 0.188828    
trend            0.0071267  0.0007465   9.547  < 2e-16 ***
S2               0.2500666  0.0727103   3.439 0.000795 ***
S3               0.2901467  0.0786457   3.689 0.000335 ***
S4               0.5284039  0.0971812   5.437 2.76e-07 ***
S5               0.5993458  0.0977075   6.134 1.06e-08 ***
S6               0.7600834  0.1039038   7.315 2.80e-11 ***
S7               0.7893207  0.1068559   7.387 1.93e-11 ***
S8               0.5440444  0.1032362   5.270 5.85e-07 ***
S9               0.3622490  0.0935867   3.871 0.000174 ***
S10              0.1583492  0.0730430   2.168 0.032077 *  
S11              0.0997400  0.0716138   1.393 0.166188    
S12             -0.1142736  0.0728983  -1.568 0.119529    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1737 on 124 degrees of freedom
Multiple R-squared:  0.903,	Adjusted R-squared:  0.8889 
F-statistic: 64.12 on 18 and 124 DF,  p-value: < 2.2e-16
```

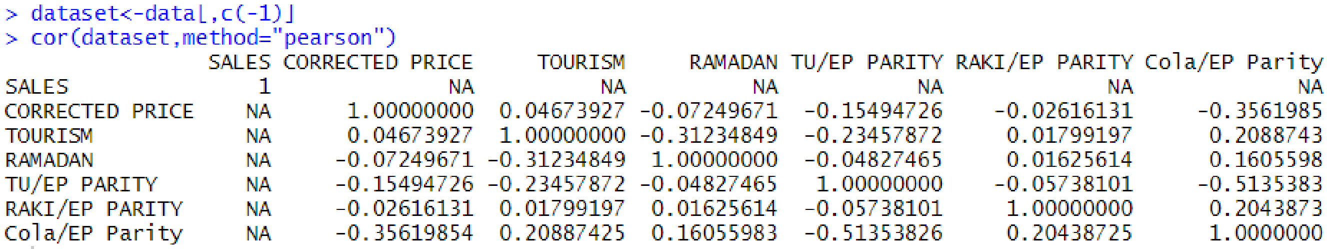
By looking at the model summary, the R-squared values are good. The F-value is significant that the variables explain the “SALES” variable. There is a trade-off between the F-statistic value and the R-squared value. Since the error explained by the model is higher with vectors used for trend and seasonality, second model will be used.

Investigating the regression coefficients, it can bee seen that “TOURISM”, “RAKI.EP.PARITY”, and “Cola.EP.Parity” are not significant to explain the “SALES” variable.

par(mfrow=c(2,2))
plot(reg.sales)

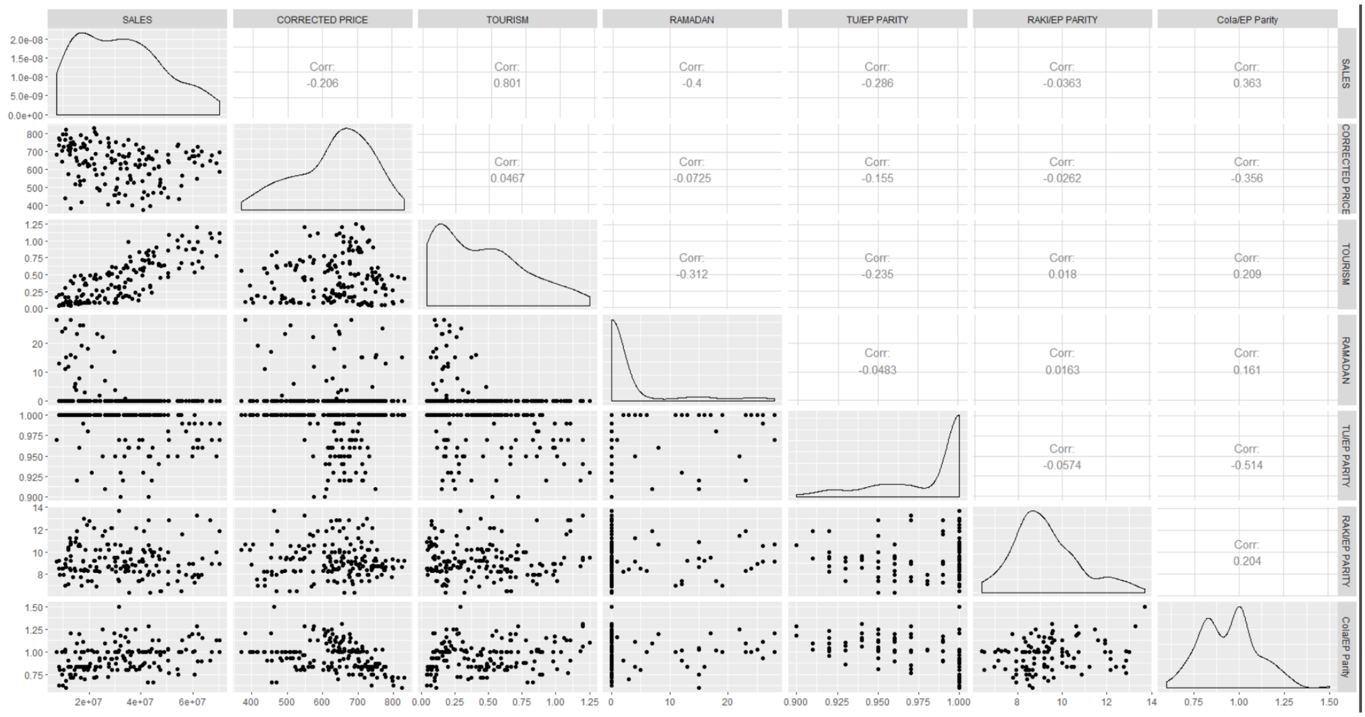


Diagnostic plot shows that the assumptions are satisfied. Correlations between the regression coefficients should be investigated.



```
> dataset<-data[,c(-1)]
> cor(dataset,method="pearson")
                SALES CORRECTED PRICE     TOURISM     RAMADAN TU/EP PARITY RAKI/EP PARITY Cola/EP Parity
SALES               1              NA          NA          NA           NA             NA             NA
CORRECTED PRICE    NA      1.00000000  0.04673927 -0.07249671  -0.15494726    -0.02616131     -0.3561985
TOURISM            NA      0.04673927  1.00000000 -0.31234849  -0.23457872     0.01799197      0.2088743
RAMADAN            NA     -0.07249671 -0.31234849  1.00000000  -0.04827465     0.01625614      0.1605598
TU/EP PARITY       NA     -0.15494726 -0.23457872 -0.04827465   1.00000000    -0.05738101     -0.5135383
RAKI/EP PARITY     NA     -0.02616131  0.01799197  0.01625614  -0.05738101     1.00000000      0.2043873
Cola/EP Parity     NA     -0.35619854  0.20887425  0.16055983  -0.51353826     0.20438725      1.0000000
```

```
library(GGally)
dataset<-data[,c(-1)]
ggpairs(dataset)
```

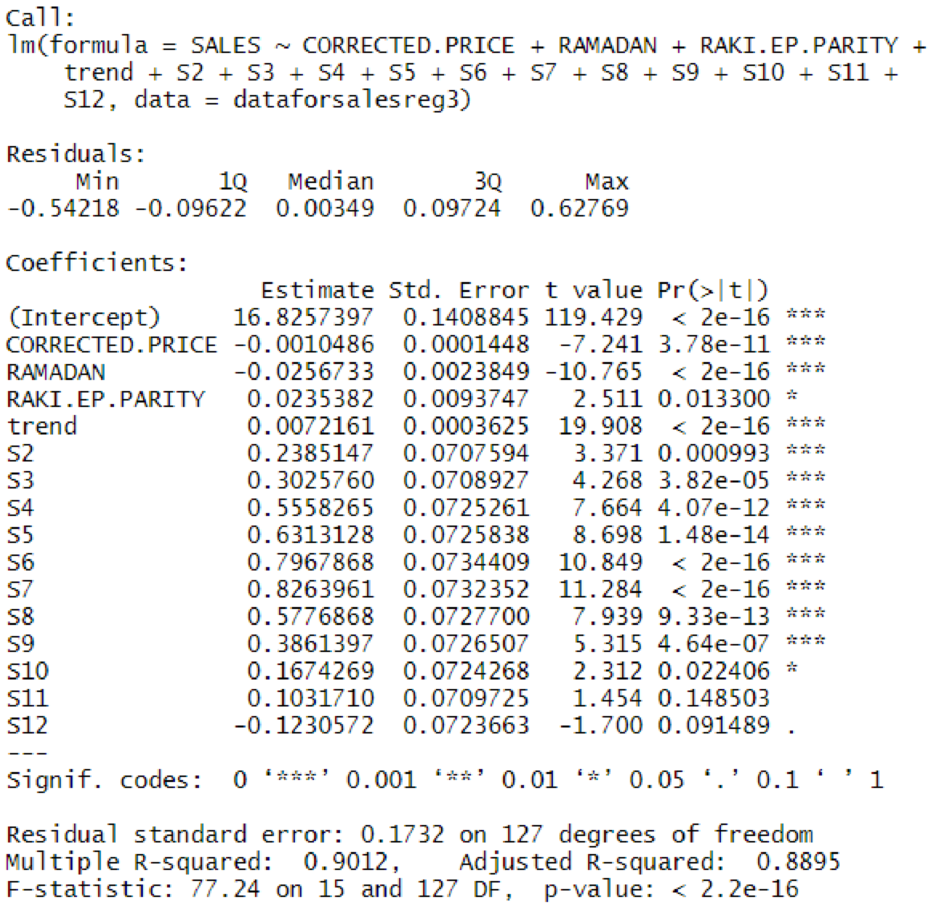


The correlation matrix shows that the correlation between the Cola/EP variable and other variables are too high. Therefore, this explanatory variable should be excluded from the model. This high correlation may cause misfunction of the model:

```
> dataforsales<-data[c(14:156),c(3:7)]
> dataforsalesreg3<-data.frame(SALES=Sales,dataforsales, trend=trend, S2=s2,
S3=s3,S4=s4,S5=s5,S6=s6,S7=s7,S8=s8,S9=s9,S10=s10,S11=s11,S12=s12)
> reg.sales.3<-lm(SALES~.,data=dataforsalesreg3)
> summary(reg.sales.3)

Call:
lm(formula = SALES ~ ., data = dataforsalesreg3)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.54572 -0.08215  0.00809  0.08624  0.62739 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     16.1014558  1.1438148  14.077  < 2e-16 ***
CORRECTED.PRICE -0.0009919  0.0001681  -5.902 3.16e-08 ***
TOURISM          0.0341241  0.1254580   0.272 0.786074    
RAMADAN         -0.0257334  0.0024209 -10.630  < 2e-16 ***
TU.EP.PARITY     0.6707265  1.0387816   0.646 0.519666    
RAKI.EP.PARITY   0.0249287  0.0103515   2.408 0.017491 *  
trend            0.0074264  0.0007133  10.411  < 2e-16 ***
S2               0.2404610  0.0725615   3.314 0.001204 ** 
S3               0.2943821  0.0788144   3.735 0.000284 ***
S4               0.5312395  0.0974470   5.452 2.56e-07 ***
S5               0.6054666  0.0978884   6.185 8.12e-09 ***
S6               0.7669661  0.1040823   7.369 2.05e-11 ***
S7               0.7947101  0.1070961   7.421 1.57e-11 ***
S8               0.5486204  0.1034855   5.301 5.03e-07 ***
S9               0.3644651  0.0938505   3.883 0.000166 ***
S10              0.1625987  0.0731896   2.222 0.028109 *  
S11              0.0992055  0.0718260   1.381 0.169684    
S12             -0.1228252  0.0728268  -1.687 0.094186 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1742 on 125 degrees of freedom
Multiple R-squared:  0.9016,	Adjusted R-squared:  0.8882 
F-statistic: 67.39 on 17 and 125 DF,  p-value: < 2.2e-16
```

F-statistic improved where R-squared is still 90%. In contrast there are still regression coefficients with large p values. Therefore another model is suggested:

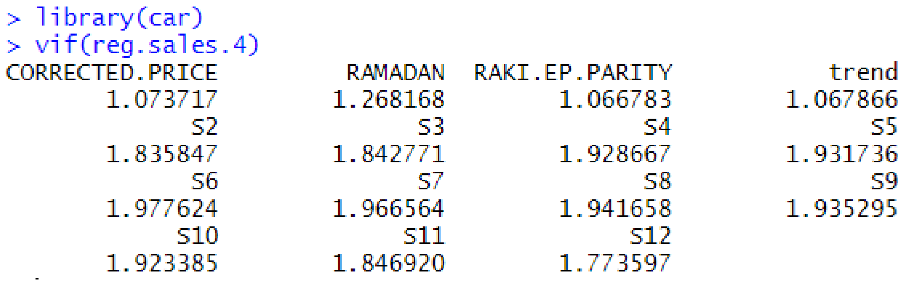
reg.sales.4=step(reg.sales.3,direction=c("backward"))
summary(reg.sales.4)



```
Call:
lm(formula = SALES ~ CORRECTED.PRICE + RAMADAN + RAKI.EP.PARITY + 
    trend + S2 + S3 + S4 + S5 + S6 + S7 + S8 + S9 + S10 + S11 + 
    S12, data = dataforsalesreg3)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.54218 -0.09622  0.00349  0.09724  0.62769 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     16.8257397  0.1408845 119.429  < 2e-16 ***
CORRECTED.PRICE -0.0010486  0.0001448  -7.241 3.78e-11 ***
RAMADAN         -0.0256733  0.0023849 -10.765  < 2e-16 ***
RAKI.EP.PARITY   0.0235382  0.0093747   2.511 0.013300 *  
trend            0.0072161  0.0003625  19.908  < 2e-16 ***
S2               0.2385147  0.0707594   3.371 0.000993 ***
S3               0.3025760  0.0708927   4.268 3.82e-05 ***
S4               0.5558265  0.0725261   7.664 4.07e-12 ***
S5               0.6313128  0.0725838   8.698 1.48e-14 ***
S6               0.7967868  0.0734409  10.849  < 2e-16 ***
S7               0.8263961  0.0732352  11.284  < 2e-16 ***
S8               0.5776868  0.0727700   7.939 9.33e-13 ***
S9               0.3861397  0.0726507   5.315 4.64e-07 ***
S10              0.1674269  0.0724268   2.312 0.022406 *  
S11              0.1031710  0.0709725   1.454 0.148503    
S12             -0.1230572  0.0723663  -1.700 0.091489 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1732 on 127 degrees of freedom
Multiple R-squared:  0.9012,	Adjusted R-squared:  0.8895 
F-statistic: 77.24 on 15 and 127 DF,  p-value: < 2.2e-16
```

5. Validity of Analysis

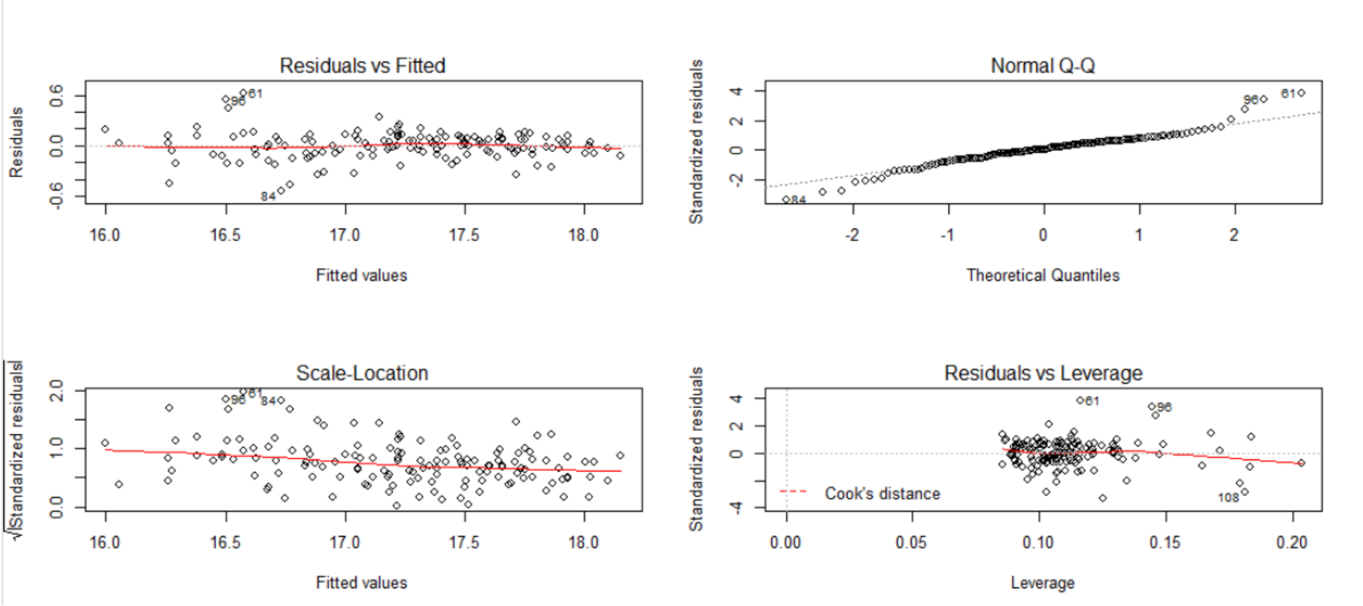
P value of F-statistic is low, therefore the regression variables together are significant to explain the “SALES” variable. The p value of T-distribution of all the variables are low and that means H0; Bi = 0; is rejected.



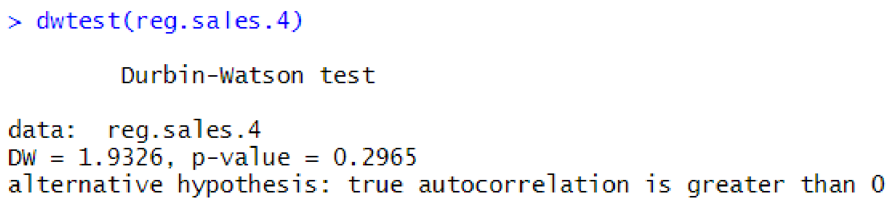
```
> library(car)
> vif(reg.sales.4)
CORRECTED.PRICE         RAMADAN  RAKI.EP.PARITY           trend 
       1.073717        1.268168        1.066783        1.067866 
             S2              S3              S4              S5 
       1.835847        1.842771        1.928667        1.931736 
             S6              S7              S8              S9 
       1.977624        1.966564        1.941658        1.935295 
            S10             S11             S12 
       1.923385        1.846920        1.773597 
```

There is no large vif values, that means there is no multicollinearity between them.

par(mfrow=c(2,2))
plot(reg.sales.4)



Residuals do not follow a pattern, normality assumption is satisfied, the diagnostic plot supports the validity of the model.

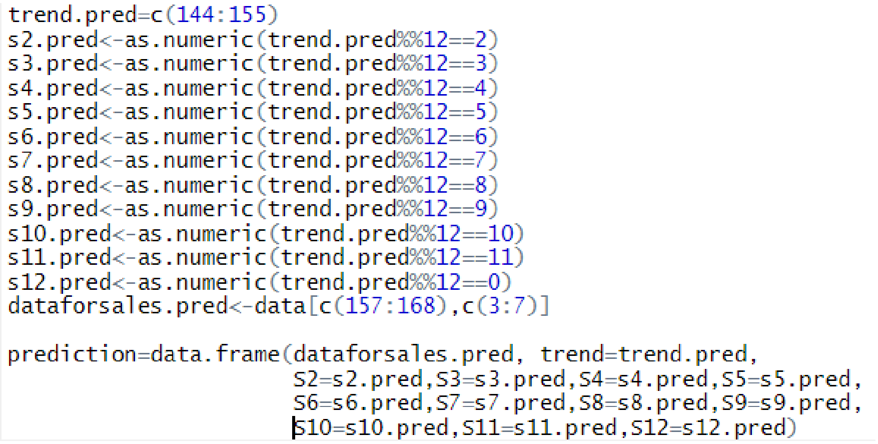


```
> dwtest(reg.sales.4)

        Durbin-Watson test

data:  reg.sales.4
DW = 1.9326, p-value = 0.2965
alternative hypothesis: true autocorrelation is greater than 0
```

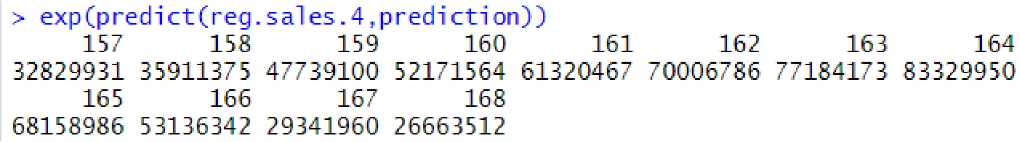
DW test shows that the hypothesis “The autocorrelation between residuals is greater than 0” is not significant.

6. Prediction of Beer Sales for 2001

The prediction data set for the next year is created. Since the model output is a logarithm value, the exponential of the value is the forecast for the corresponding month next year.



```
trend.pred=c(144:155)
s2.pred<-as.numeric(trend.pred%%12==2)
s3.pred<-as.numeric(trend.pred%%12==3)
s4.pred<-as.numeric(trend.pred%%12==4)
s5.pred<-as.numeric(trend.pred%%12==5)
s6.pred<-as.numeric(trend.pred%%12==6)
s7.pred<-as.numeric(trend.pred%%12==7)
s8.pred<-as.numeric(trend.pred%%12==8)
s9.pred<-as.numeric(trend.pred%%12==9)
s10.pred<-as.numeric(trend.pred%%12==10)
s11.pred<-as.numeric(trend.pred%%12==11)
s12.pred<-as.numeric(trend.pred%%12==0)
dataforsales.pred<-data[c(157:168),c(3:7)]

prediction=data.frame(dataforsales.pred, trend=trend.pred,
                      S2=s2.pred,S3=s3.pred,S4=s4.pred,S5=s5.pred,
                      S6=s6.pred,S7=s7.pred,S8=s8.pred,S9=s9.pred,
                      S10=s10.pred,S11=s11.pred,S12=s12.pred)
```



```
> exp(predict(reg.sales.4,prediction))
     157      158      159      160      161      162      163      164 
32829931 35911375 47739100 52171564 61320467 70006786 77184173 83329950 
     165      166      167      168 
68158986 53136342 29341960 26663512 
```

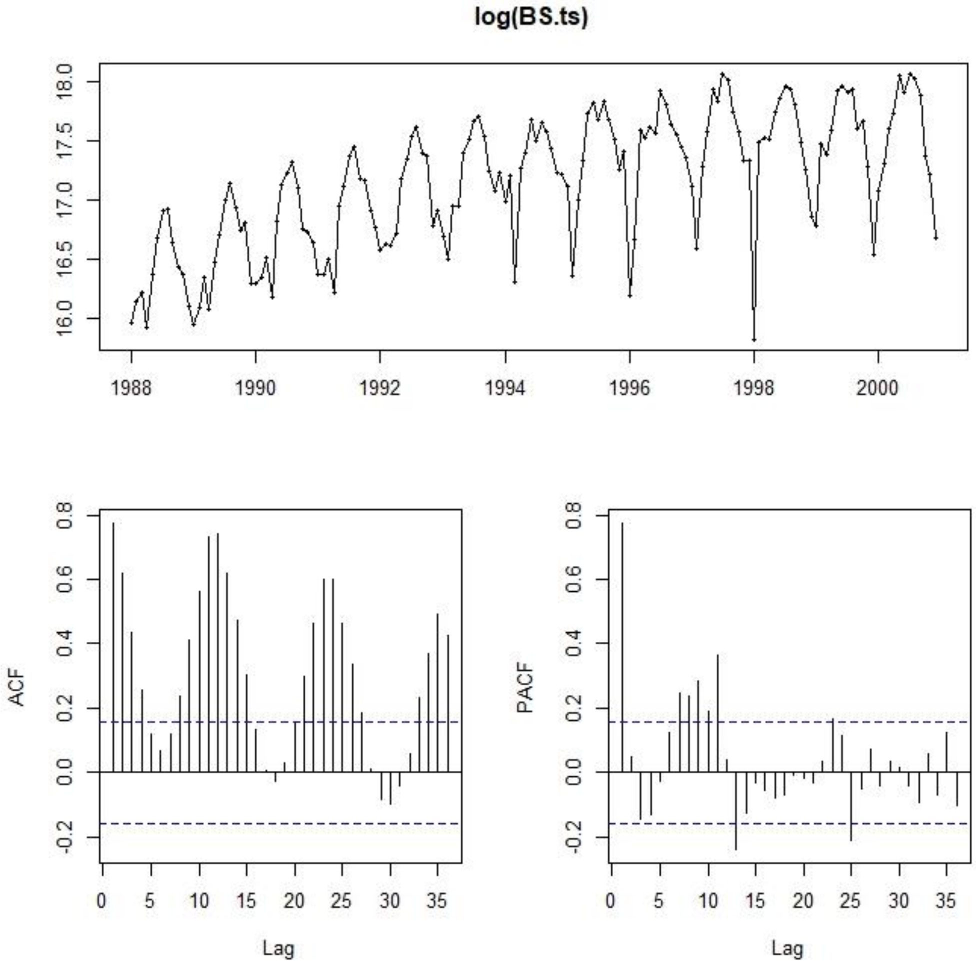
Method B: FORECASTING WITH TIME SERIES ANALYSIS

1- Preliminary transformation decision:

To induce stationarity, it is decided that preliminary transformation is needed. Logarithmic transformation should be applied to stabilise the variance of time series.

```
tsdisplay(log(BS.ts))
```

2- Utilization of time series plots with ACF and PACF

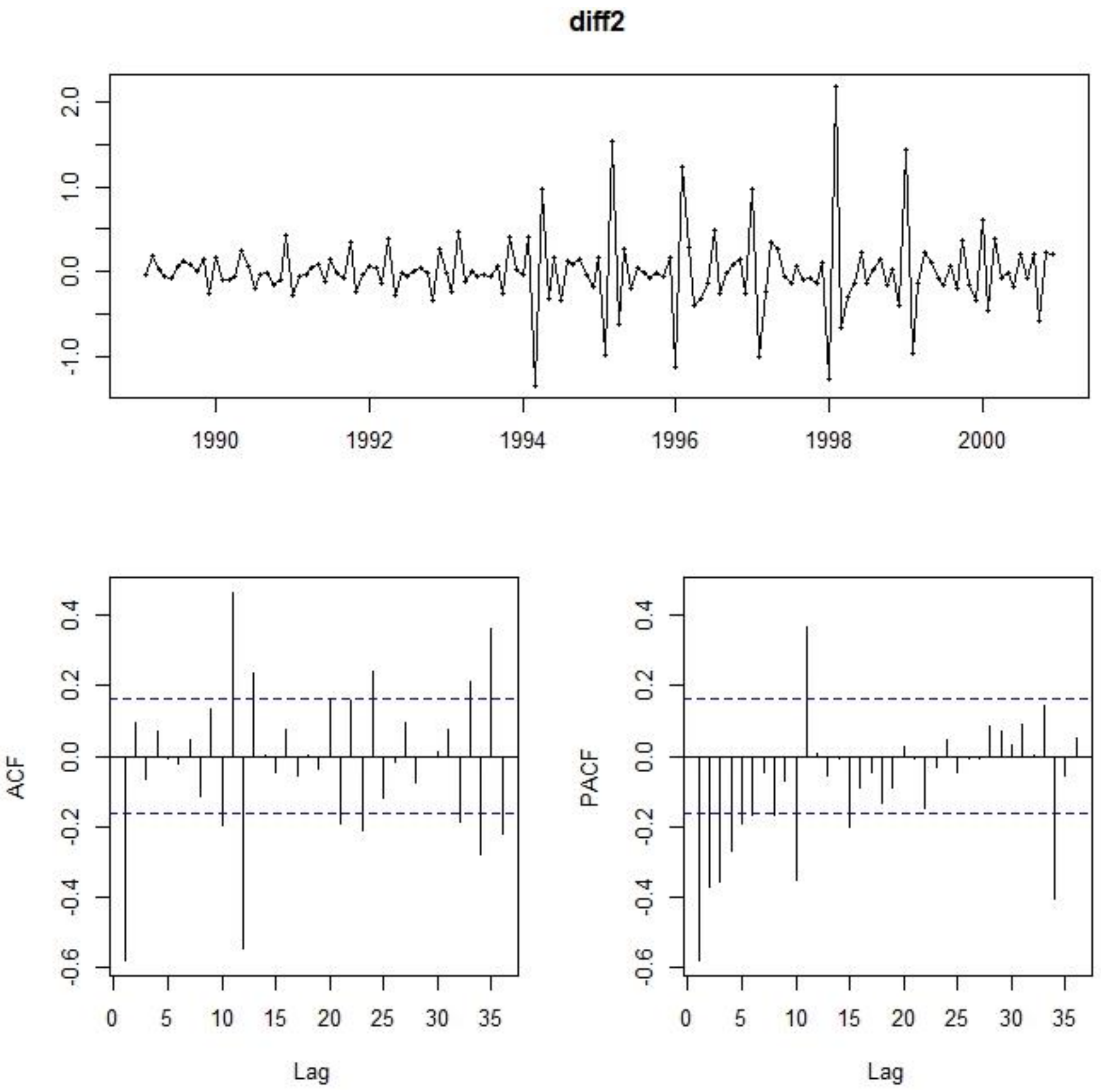


When the shapes of auto correlation functions of the time series of logarithmic Beer Sales is examined, it is clearly observed that there is a sinusodial relationship with 12 lags which implies that there is seasonality at lag 12.The auto correlation values at the other lags are also significant so there is a both trend and seasonality . Since the series has a strong and consistent seasonal pattern, then we should use an order of seasonal differencing.

```
#ndiffs(log(BS.ts))
[1] 1
#nsdiffs(log(BS.ts))
[1] 1
```

Additional proof to use seasonal and regular difference as appropiate number of differences are shown by KPSS test. These functions suggests that we should do both a seasonal difference and one regular difference on logaritmic beer sales data.

```
diff1 <- diff(log(BS.ts), 12)

acf(diff1)

pacf(diff1)

diff2 <- diff(diff1, 1)

plot(diff2)

acf(diff2, lag.max = 36)

pacf(diff2, lag.max = 36)

tsdisplay(diff2)
```

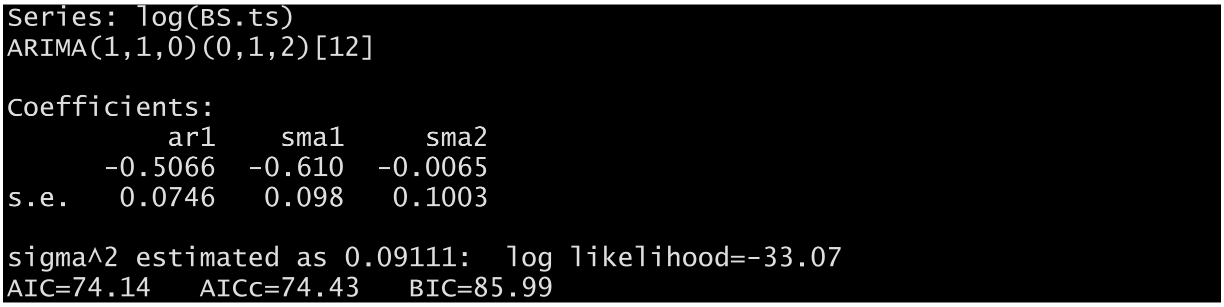


ACF cut off at lag=1 which is a sudden fall from significant level can be seen and PACF dies out slowly, in other words exponentially decaying. MA(1) might be suggested in regular terms. In seasonal terms, significant levels are seen on ACF plot which suggests seasonal AR(1). Since acf function has oscillation movements in significant exceedings while pacf has not, AR(1) is used instead of MA.

3- Initial ARIMA model

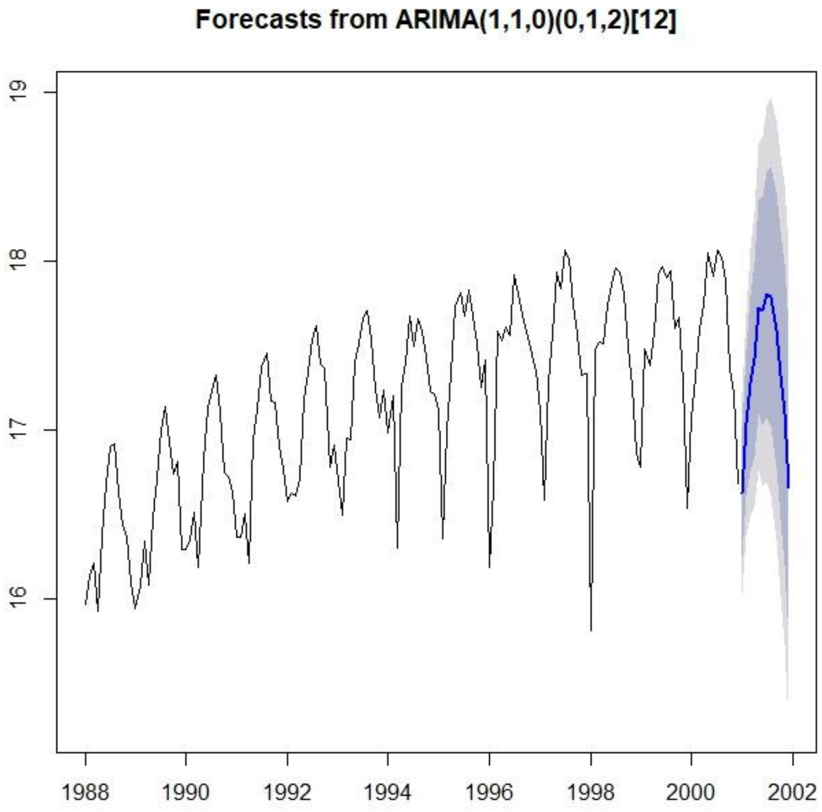
```
arimaBS<-Arima(log(BS.ts), order=c(1,1,0), seasonal=c(0,1,1))

arimaBS

plot(forecast(arimaBS,h=12))
```

The R code above is used for making ARIMA model with the reasons mentioned above. This is the initial ARIMA model based on the inspection on ACF and PACF plot.



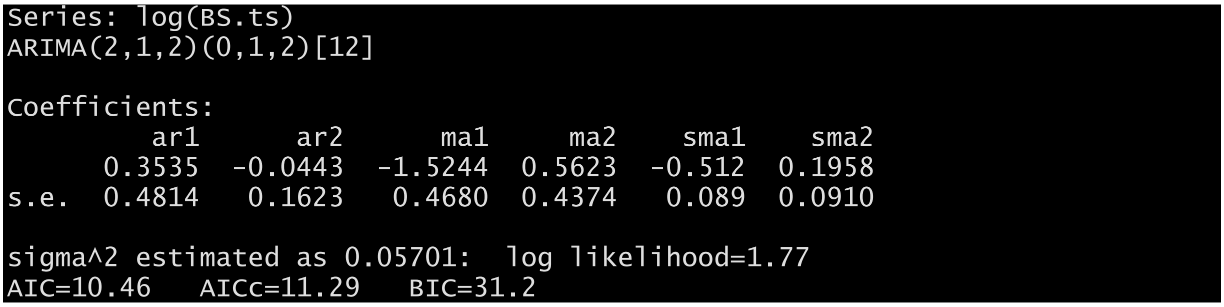
```
Series: log(BS.ts)
ARIMA(1,1,0)(0,1,2)[12]

Coefficients:
          ar1     sma1      sma2
      -0.5066  -0.610  -0.0065
s.e.   0.0746   0.098   0.1003

sigma^2 estimated as 0.09111:  log likelihood=-33.07
AIC=74.14   AICc=74.43   BIC=85.99
```

The output above gives the AIC, AICc and BIC values of the initial ARIMA model. The plot below gives the data with forecast values of the initial ARIMA model.



4- Neighborhood search of the initial model

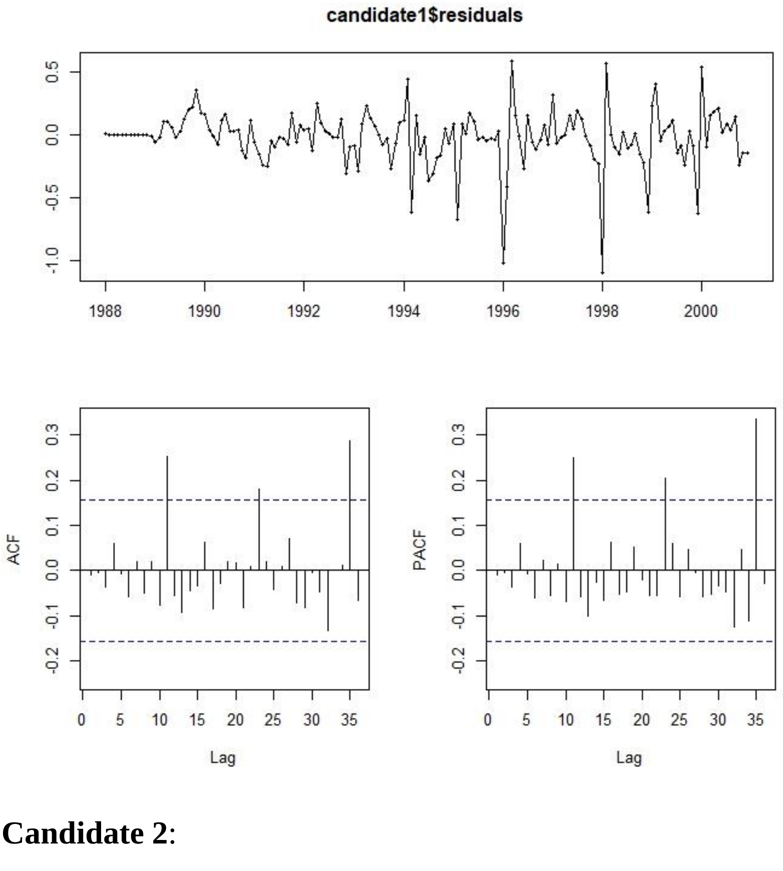
```
candidate1 <- auto.arima(log(BS.ts) , allowdrift = TRUE)

candidate1

tsdisplay(candidate1$residuals)
```



```
Series: log(BS.ts)
ARIMA(2,1,2)(0,1,2)[12]

Coefficients:
         ar1      ar2      ma1     ma2    sma1    sma2
      0.3535  -0.0443  -1.5244  0.5623  -0.512  0.1958
s.e.  0.4814   0.1623   0.4680  0.4374   0.089  0.0910

sigma^2 estimated as 0.05701:  log likelihood=1.77
AIC=10.46   AICc=11.29   BIC=31.2
```

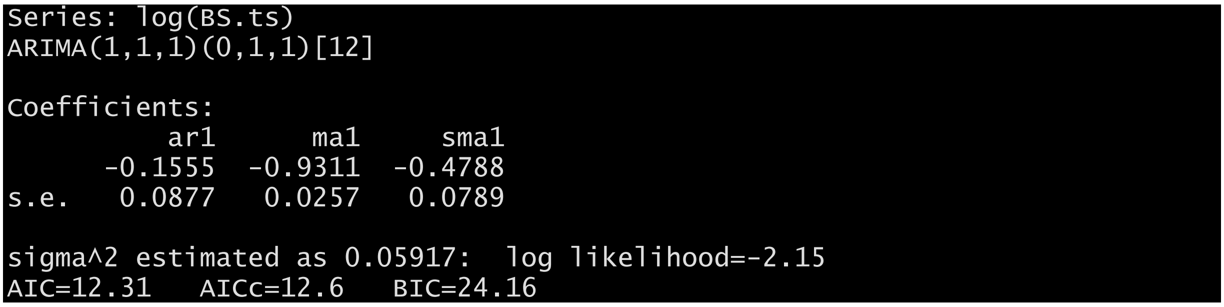
First candidate – which is determined by optimization process of auto.arima function and supposed to yield one of the best possible choice of variable selection in ARIMA model- has 1 regular and 1 seasonal difference just as our model and additionally regular AR(2)+MA(2) and a seasonal MA(2). AIC value is 10.46 which is far better than our AIC since lower values on AIC and BIC is preferred.

ACF and PACF of candidate 1 is shown below:



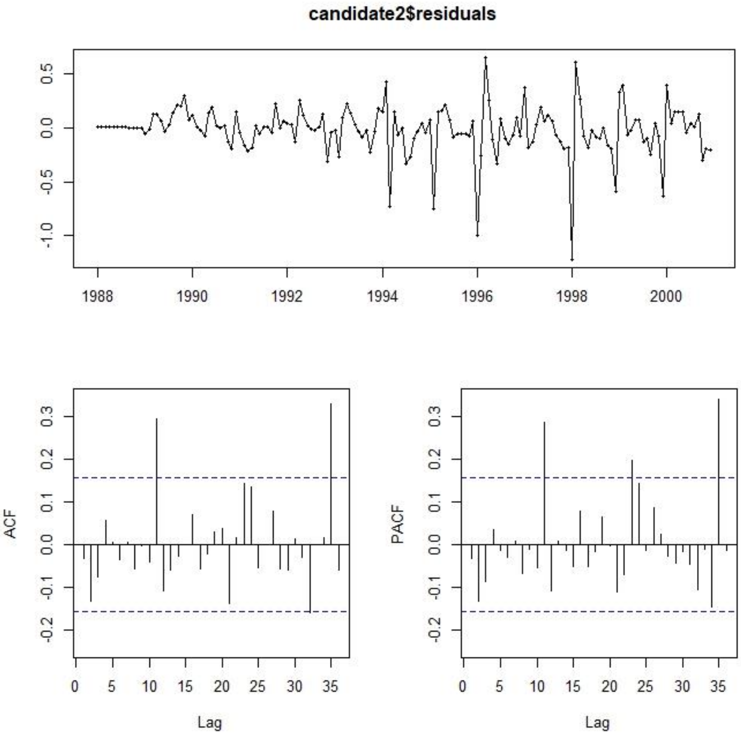
Candidate 2:

```
candidate2 <- Arima(log(BS.ts), order=c(1,1,1), seasonal=c(0,1,1))

candidate2

tsdisplay(candidate2$residuals)
```



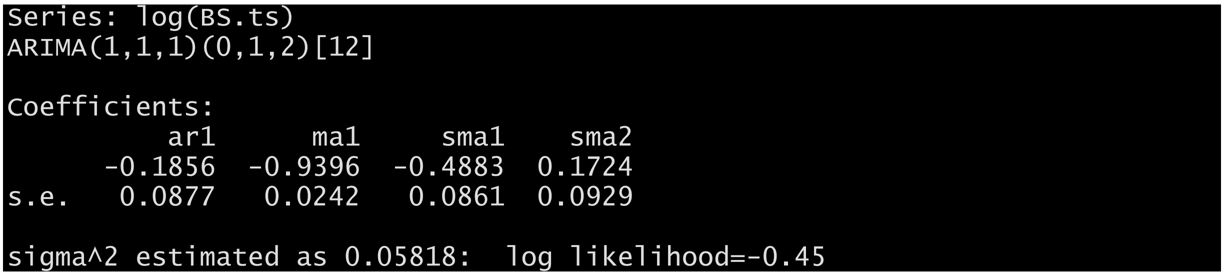
```
Series: log(BS.ts)
ARIMA(1,1,1)(0,1,1)[12]

Coefficients:
          ar1      ma1     sma1
      -0.1555  -0.9311  -0.4788
s.e.   0.0877   0.0257   0.0789

sigma^2 estimated as 0.05917:  log likelihood=-2.15
AIC=12.31   AICc=12.6   BIC=24.16
```

ACF&PACF plot:

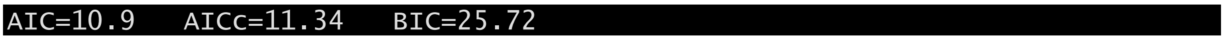


Candidate 3:

```
candidate3 <- Arima(log(BS.ts), order=c(1,1,1), seasonal=c(0,1,2))

candidate3

tsdisplay(candidate3$residuals)
```

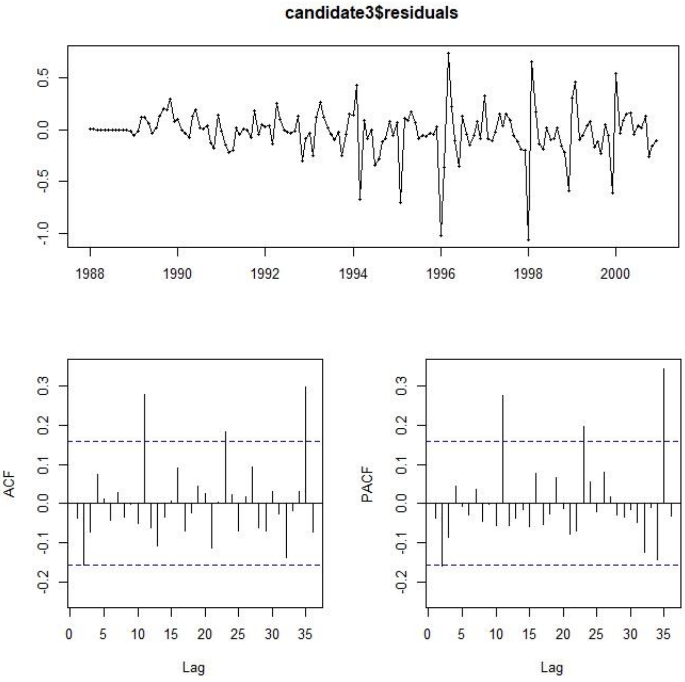


```
Series: log(BS.ts)
ARIMA(1,1,1)(0,1,2)[12]

Coefficients:
          ar1      ma1     sma1    sma2
      -0.1856  -0.9396  -0.4883  0.1724
s.e.   0.0877   0.0242   0.0861  0.0929

sigma^2 estimated as 0.05818:  log likelihood=-0.45
```



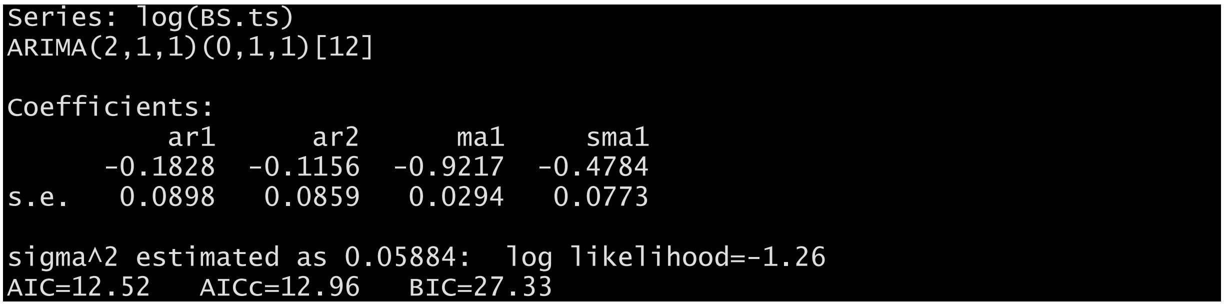
```
AIC=10.9   AICc=11.34   BIC=25.72
```

ACF&PACF plot:



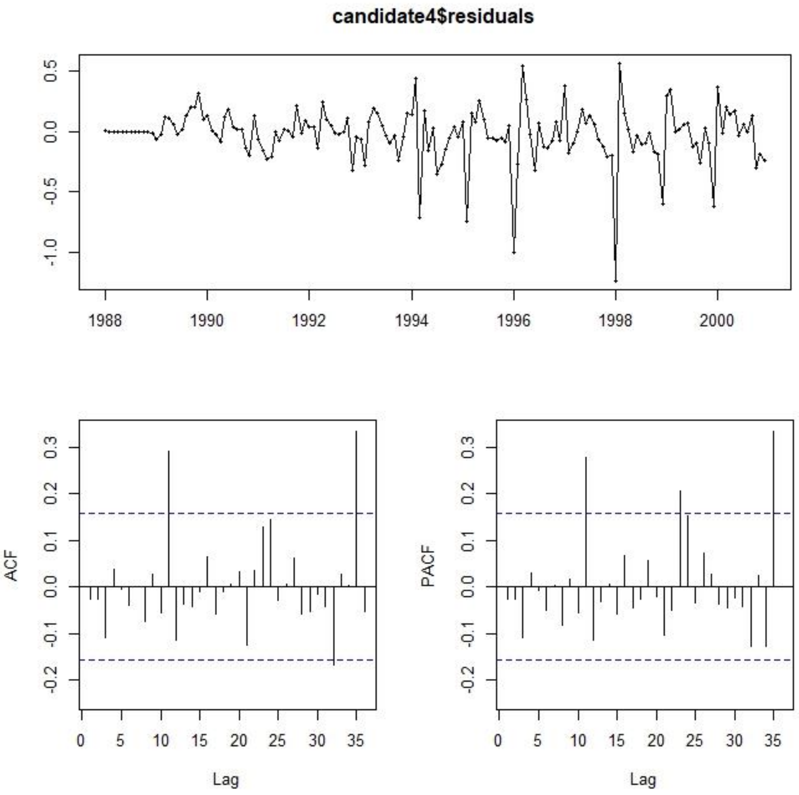
Candidate 4:

```
candidate4 <- Arima(log(BS.ts), order=c(2,1,1), seasonal=c(0,1,1))

candidate4

tsdisplay(candidate4$residuals)
```



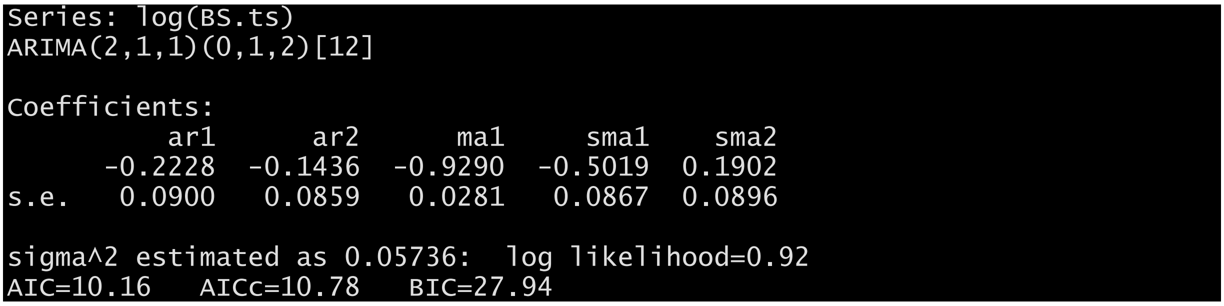
```
Series: log(BS.ts)
ARIMA(2,1,1)(0,1,1)[12]

Coefficients:
          ar1      ar2      ma1     sma1
      -0.1828  -0.1156  -0.9217  -0.4784
s.e.   0.0898   0.0859   0.0294   0.0773

sigma^2 estimated as 0.05884:  log likelihood=-1.26
AIC=12.52   AICc=12.96   BIC=27.33
```

ACF&PACF plot:



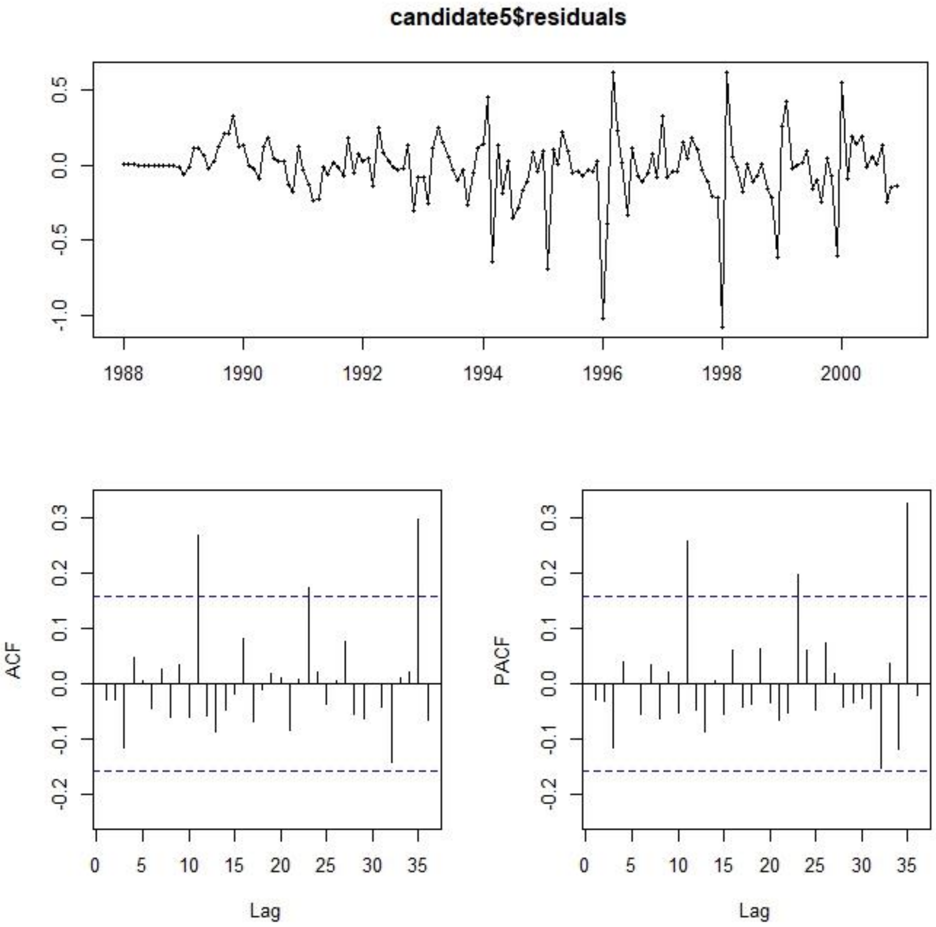
Candidate 5:

```
candidate5 <- Arima(log(BS.ts), order=c(2,1,1), seasonal=c(0,1,2))

candidate5

tsdisplay(candidate5$residuals)
```



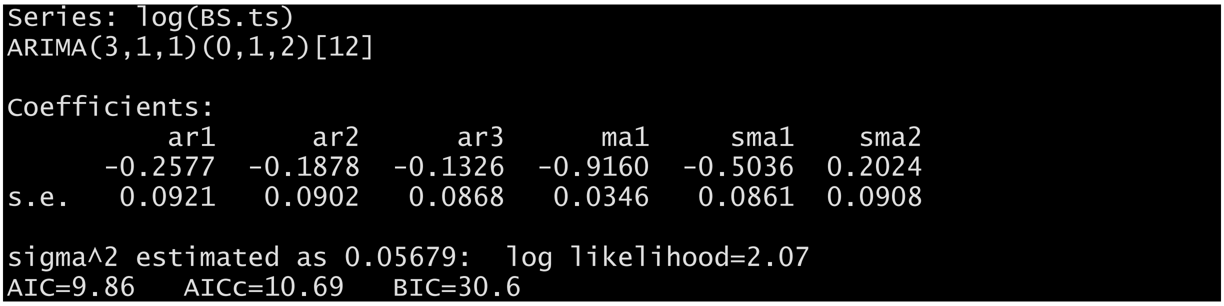
```
Series: log(BS.ts)
ARIMA(2,1,1)(0,1,2)[12]

Coefficients:
          ar1      ar2      ma1     sma1    sma2
      -0.2228  -0.1436  -0.9290  -0.5019  0.1902
s.e.   0.0900   0.0859   0.0281   0.0867  0.0896

sigma^2 estimated as 0.05736:  log likelihood=0.92
AIC=10.16   AICc=10.78   BIC=27.94
```

ACF&PACF plot:



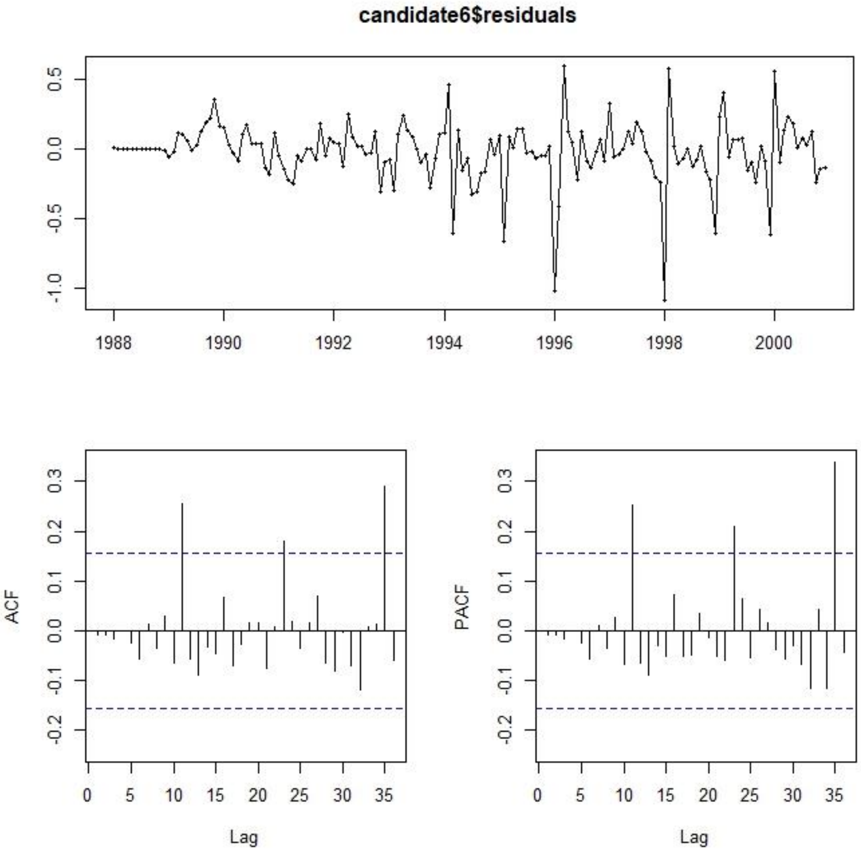
Candidate 6:

```
candidate6 <- Arima(log(BS.ts), order=c(3,1,1), seasonal=c(0,1,2))

candidate6

tsdisplay(candidate6$residuals)
```



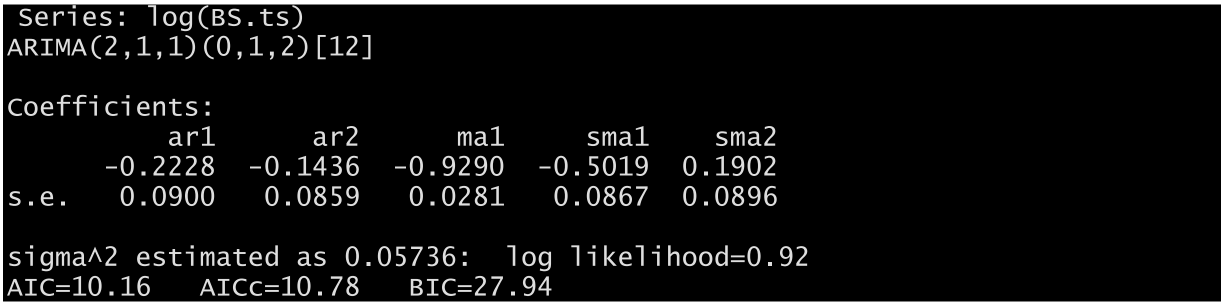
```
Series: log(BS.ts)
ARIMA(3,1,1)(0,1,2)[12]

Coefficients:
          ar1      ar2      ar3      ma1     sma1    sma2
      -0.2577  -0.1878  -0.1326  -0.9160  -0.5036  0.2024
s.e.   0.0921   0.0902   0.0868   0.0346   0.0861  0.0908

sigma^2 estimated as 0.05679:  log likelihood=2.07
AIC=9.86   AICc=10.69   BIC=30.6
```

Acf&pacf plot:

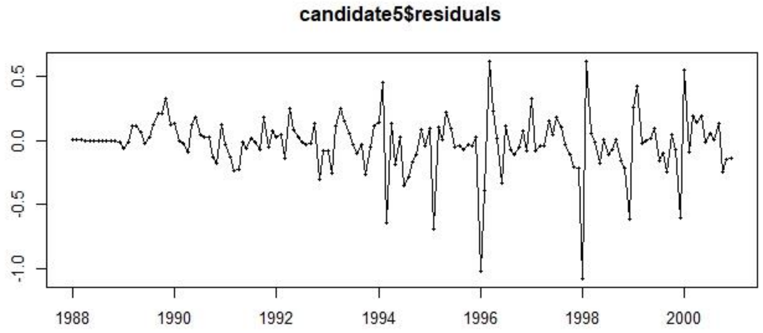


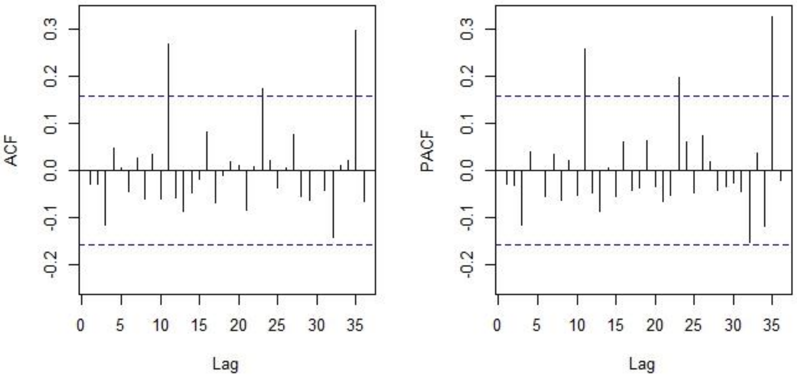
5- Best model choice

To choose the best model minimum aic and bic values and noncorrelated residuals are searched. Candidate6 has the minimum AIC value and candidate2 has the minimum BIC value. All of the ACF and PACF have significant values with similar plots. Overall, candidate 5 has the best AIC&BIC value among others, that’s why candidate5 is chosen.



```
 Series: log(BS.ts) 
ARIMA(2,1,1)(0,1,2)[12] 

Coefficients:
          ar1      ar2      ma1     sma1    sma2
      -0.2228  -0.1436  -0.9290  -0.5019  0.1902
s.e.   0.0900   0.0859   0.0281   0.0867  0.0896

sigma^2 estimated as 0.05736:  log likelihood=0.92
AIC=10.16   AICc=10.78   BIC=27.94
```





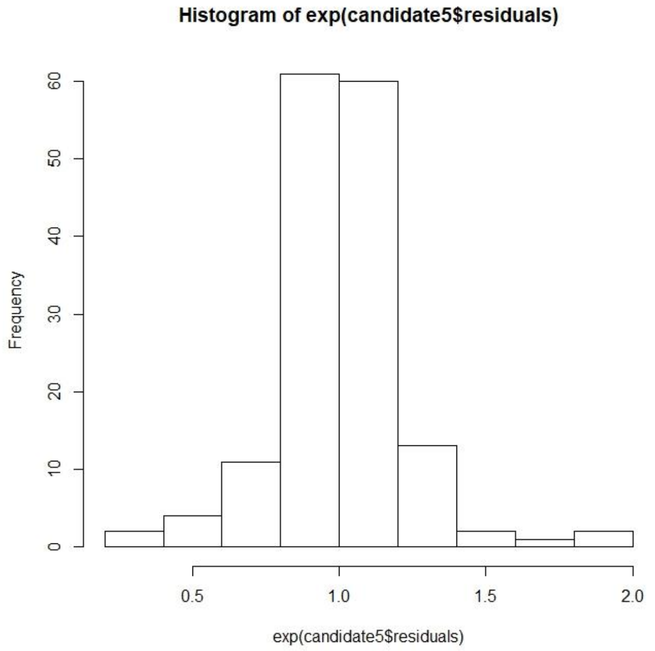
6- Validity of analysis

Residuals are should be checked for validity. They should have constant variance and mean that is close to 0.

```
mean(exp(candidate5$residuals))

[1] 1.004584
```

Mean of the residuals are close to one which is desirable for multiplicative process. In multiplicative processes residuals are normally distributed around one. So, this assumption holds.

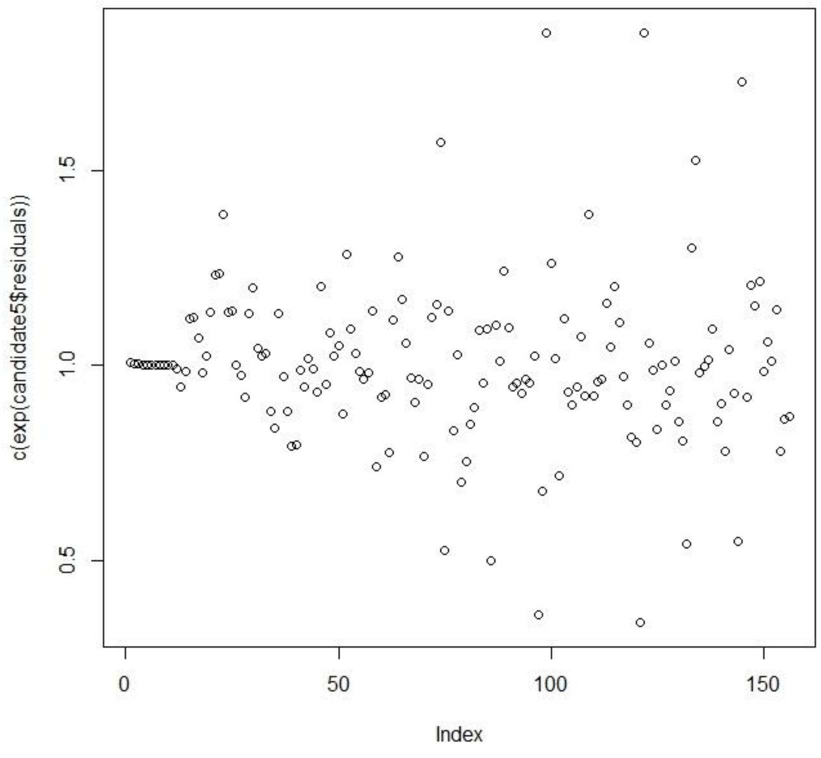
```
hist(exp(candidate5$residuals))
```



Histogram also shows residuals are normally distributed around 1.

```
plot(c(exp(candidate5$residuals)))
```

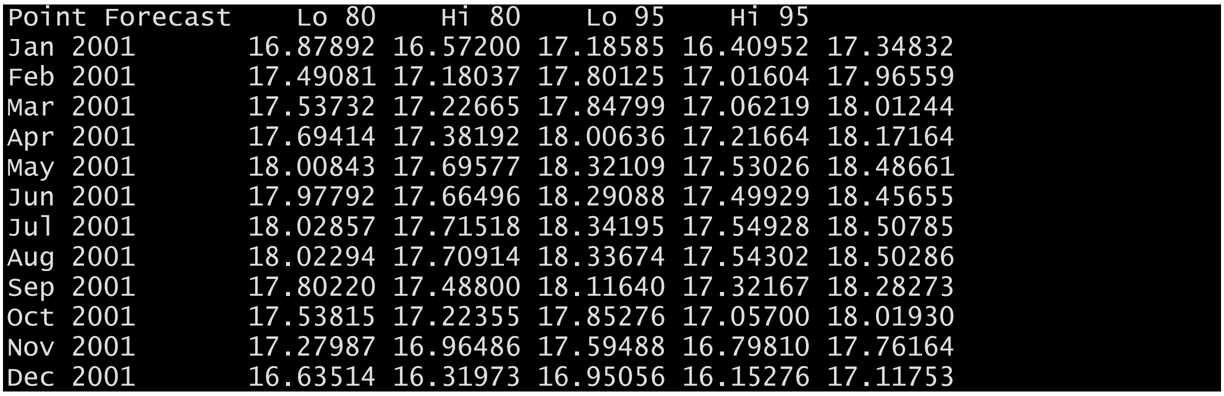
Residual plot is shown below and as can be inferred that residuals have constant variance



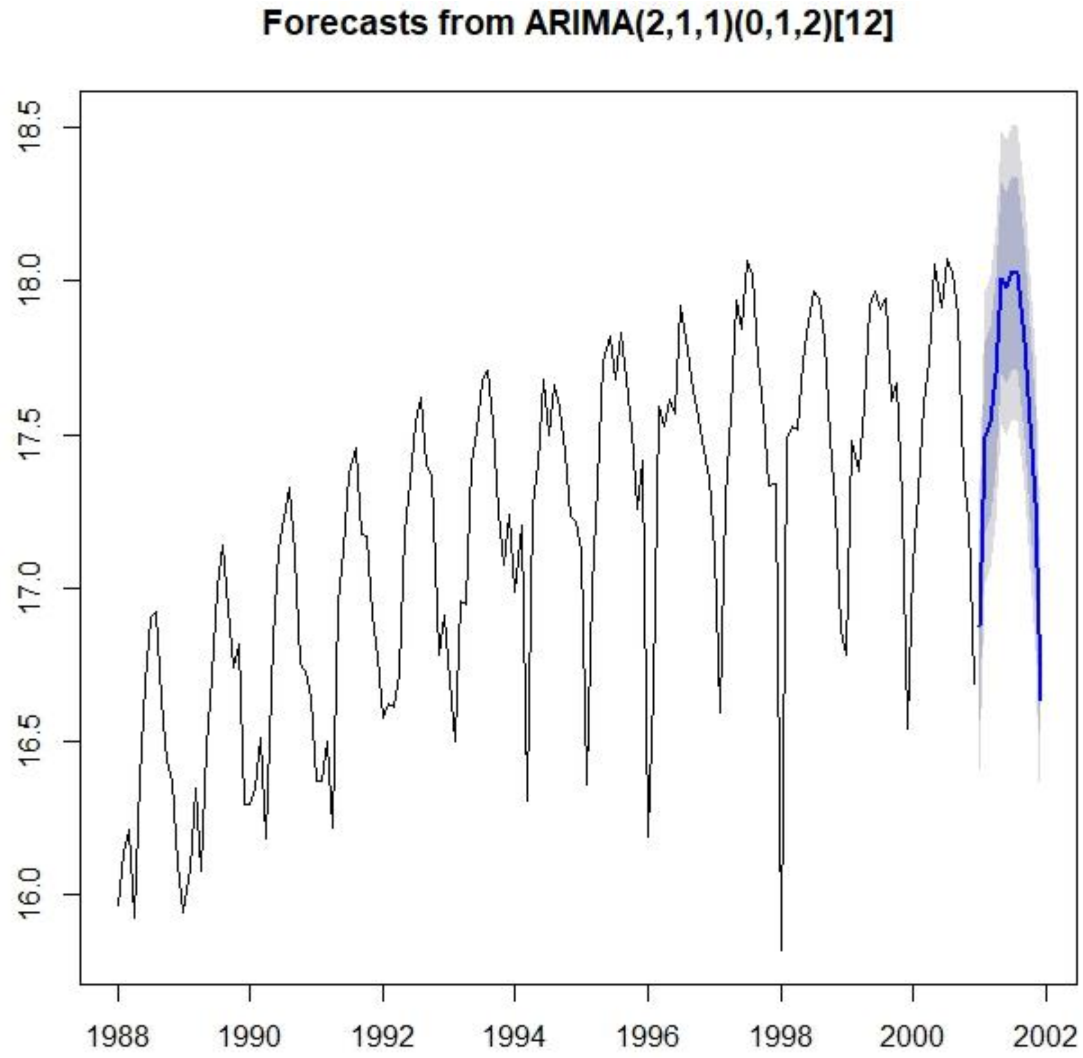
7- Prediction of beer sales for 2001

```
forecast(candidate5, h=12)

plot(forecast(candidate5, h=12))
```

The R code above is used in order to forecast beer sales by chosen model candidate 5. Outputs and forecast plot are listed below:



```
Point Forecast     Lo 80    Hi 80    Lo 95    Hi 95
Jan 2001       16.87892 16.57200 17.18585 16.40952 17.34832
Feb 2001       17.49081 17.18037 17.80125 17.01604 17.96559
Mar 2001       17.53732 17.22665 17.84799 17.06219 18.01244
Apr 2001       17.69414 17.38192 18.00636 17.21664 18.17164
May 2001       18.00843 17.69577 18.32109 17.53026 18.48661
Jun 2001       17.97792 17.66496 18.29088 17.49929 18.45655
Jul 2001       18.02857 17.71518 18.34195 17.54928 18.50785
Aug 2001       18.02294 17.70914 18.33674 17.54302 18.50286
Sep 2001       17.80220 17.48800 18.11640 17.32167 18.28273
Oct 2001       17.53815 17.22355 17.85276 17.05700 18.01930
Nov 2001       17.27987 16.96486 17.59488 16.79810 17.76164
Dec 2001       16.63514 16.31973 16.95056 16.15276 17.11753
```


Forecasts from ARIMA(2,1,1)(0,1,2)[12]