Forecasting of Eggs Deposition

Tarun.S.Sarode

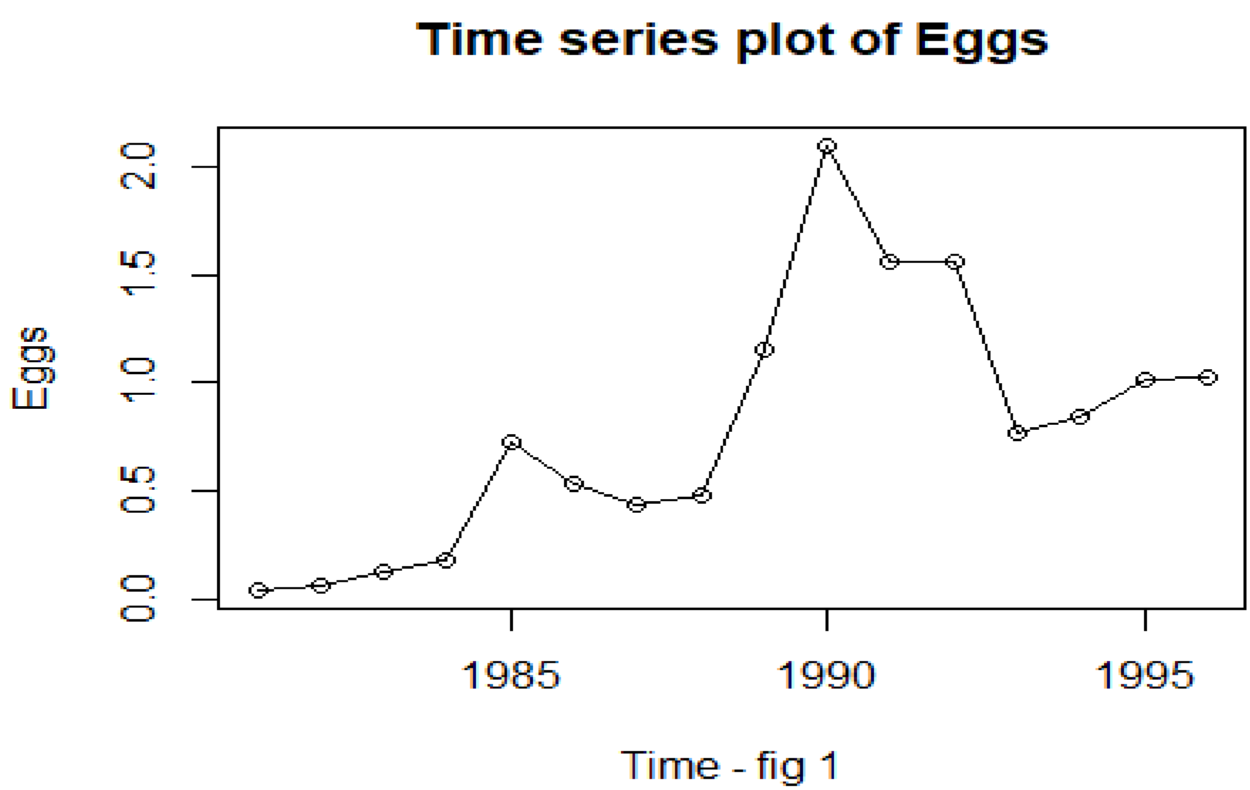
Table Of Contents

Introduction

The dataset is related to a fish called Coregonus hoyi which is found in freshwater bodies like lakes. The dataset has a count of the egg deposition (in millions) over the years from 1981 to 1996. The analysing is carried on this data set using relevant model fitting and functions in R studio. The data set has two variables 'Year' and 'Eggs'.

Data Analysis & Interpretation

Time series plot of the Eggs dataset.

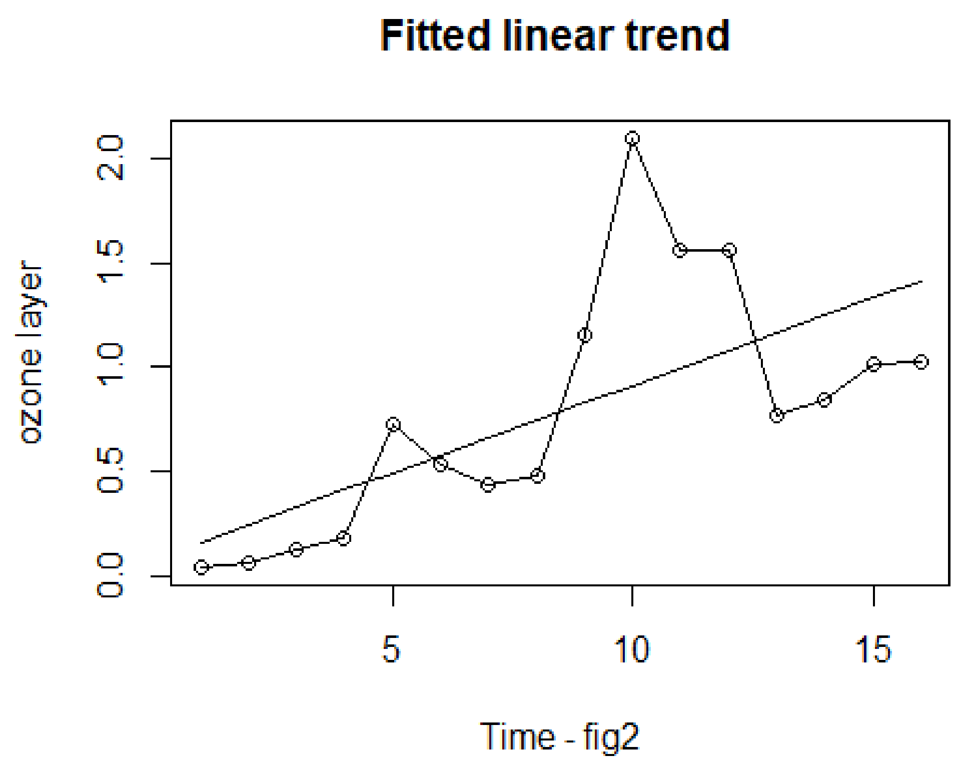


From the fig 1 it can be seen that:

1. a slight upward trend.
2. there is no seasonality.
3. there is also co variance.
4. there is an intervention point in the year 1988.
5. it has both moving average and auto regressive behaviour.

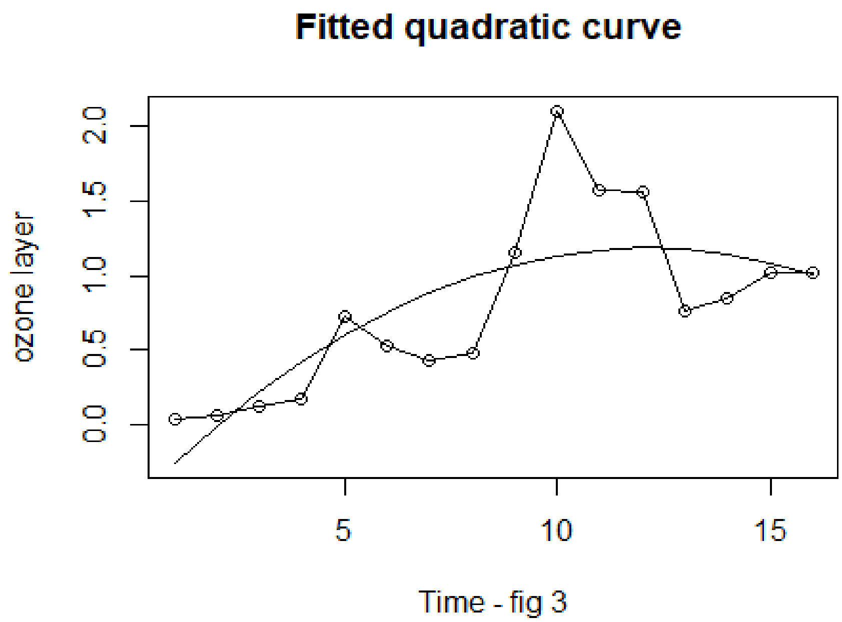
Firstly, stochastic models such as linear and quadratic are applied and checked if it is a good fit for the seiries.

- **Linear Trend Model**



From the fig 2 it can be observed that the linear model is not a right fit for the series.

❖ Quadratic Trend Model

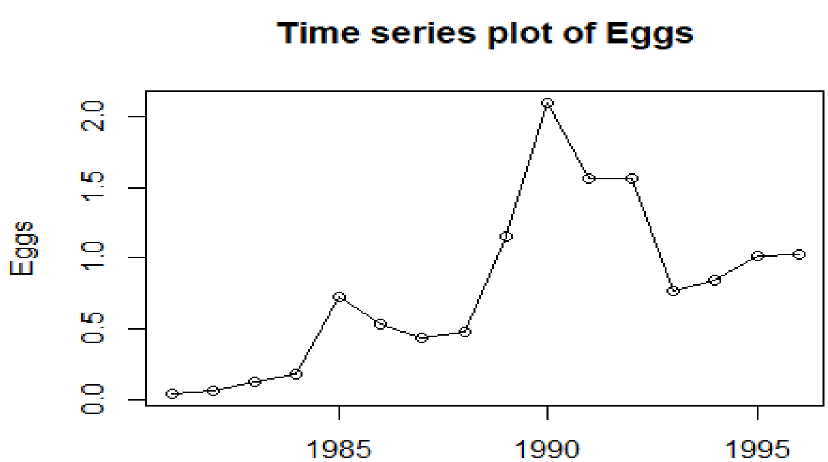


From the fig 3 which is a quadratic model does not fit the series and is not the right model.

- ❖ Harmonic Trend Model The harmonic trend models are applied to cyclic data set it is not appled to the eggs data set.

As the stochastic models are not a right fit we will apply the deterministic trend models.

Time series plot of the Egg data set.



It is nessary to check the stationarity of the series, the QQ-plot and the shapiro-wilk test is used to check the stationarity of the series.

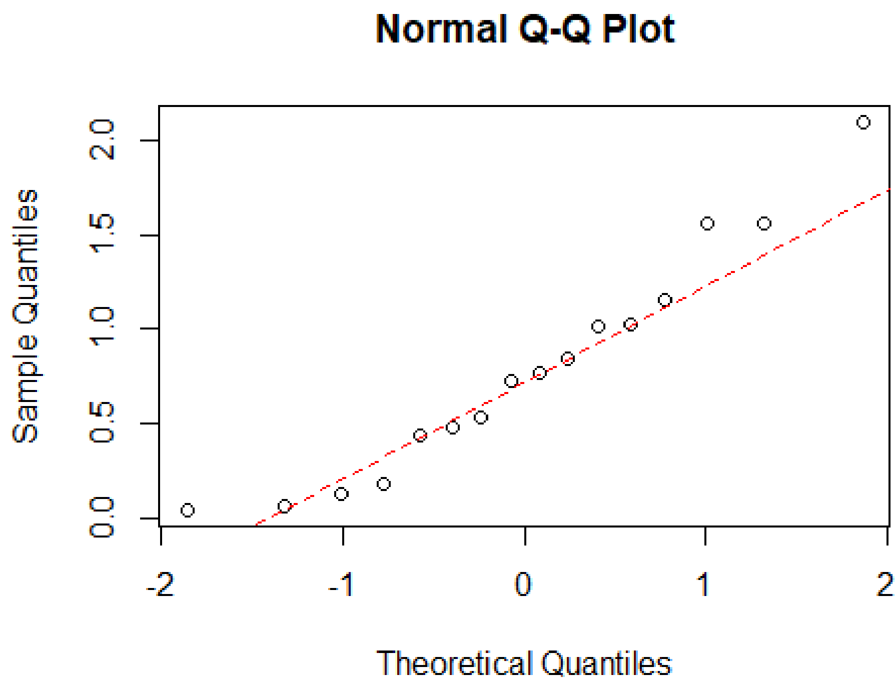


Fig 4

Output 1

```
##
## | Shapiro-Wilk normality test
##
## data:  egg2
## W = 0.94201, p-value = 0.3744
```

QQ-plot- it can be observed that from the fig 4 the series has low norrmality and shapiro-wilk (output 1) test gives p-value of 0.3744 which is >0.05 and hence we reject the null hypothesis which implying that the distribution of the data is not significantly different from normal distribution. In other words, we can assume the normality

As the QQ-plot and Shapiro-wilk test do not match we will confirm it with the ACF and Pacf plot

- **ACF Plot**

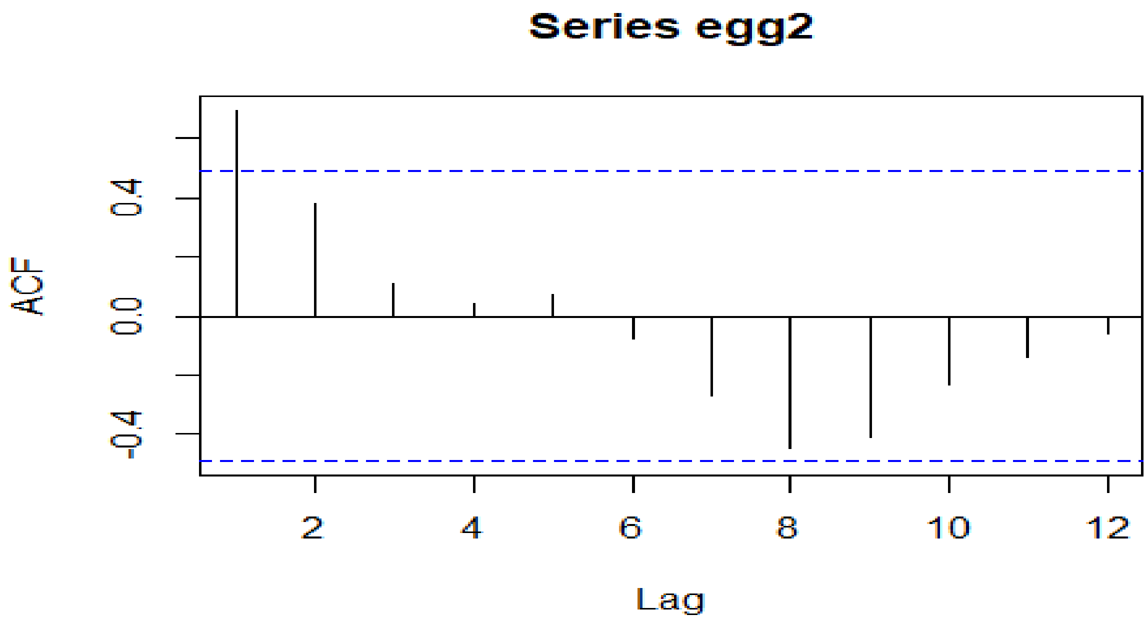


Fig 5

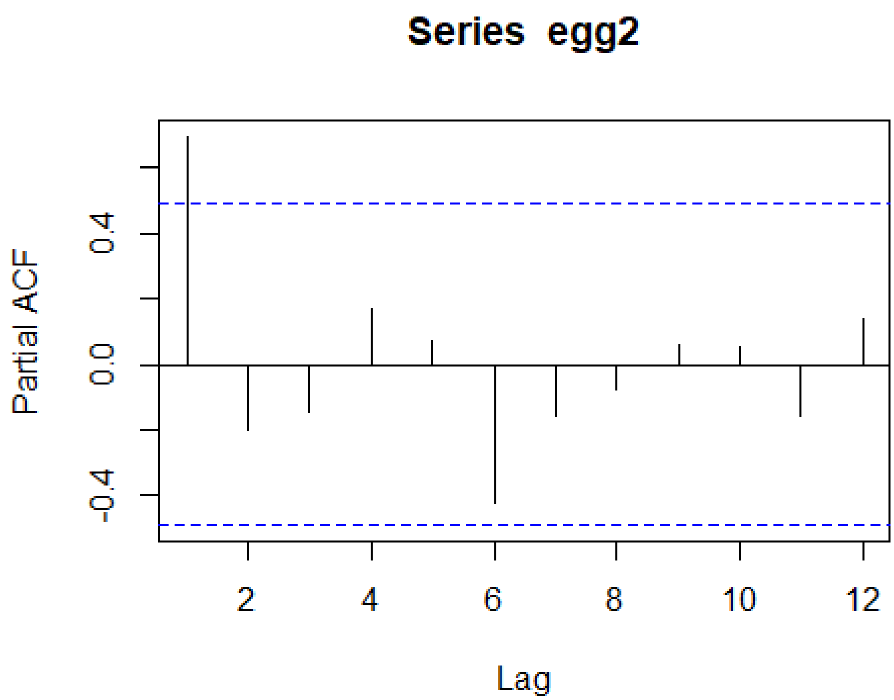


Fig 6

From the ACF plot (fig 5) and pcaf (fig 6) it can be observed that there is a slowly decaying pattern in ACF and very high first correlation shows the existence of trend and stationarity it can also be seen that there seasonality in the series from the wave pattern.

- ## Adf Test

Output 2

```
##
##  Augmented Dickey-Fuller Test
##
## data:  egg2
## Dickey-Fuller = -2.0669, Lag order = 2, p-value = 0.5469
## alternative hypothesis: stationary
```

From the output 2 which is the adf test the p-value is 0.5469 which is >0.05 and implies that the series is non stationary and fail to reject the null hypothesis.

As the series is not stationary the further analysis cannot be done as it is nessary to ensure the series is free of the trend hence transformation and differencing is applied to make the series stationary.

- **BoxCox Transformation**

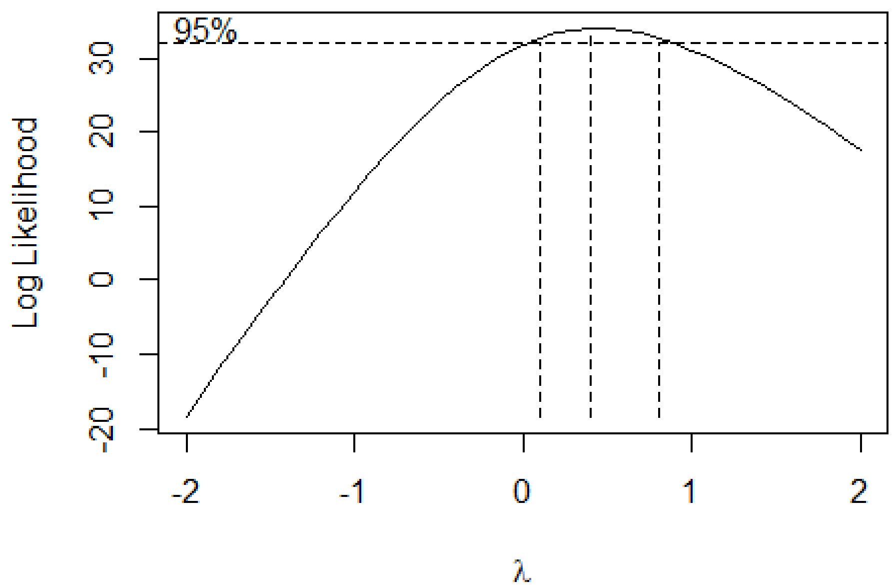


Fig 7

Output 3

```
## [1] 0.1 0.8
```

From the BoxCox transformation the class interval (output 3) is between 0.1 and 0.8 and considering the lambda value to be 0.4 and use this value to apply the transformation.

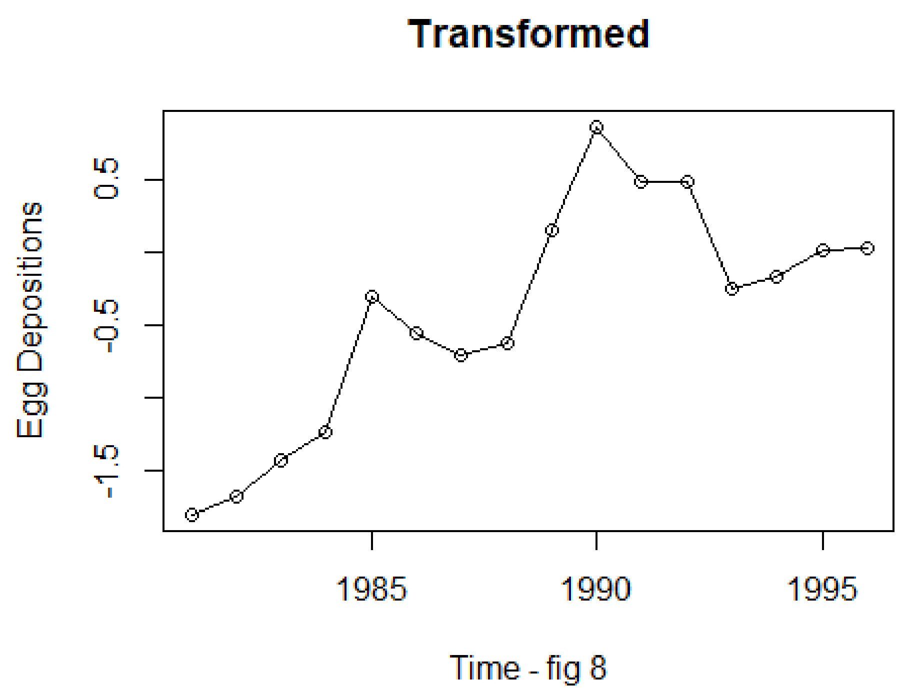
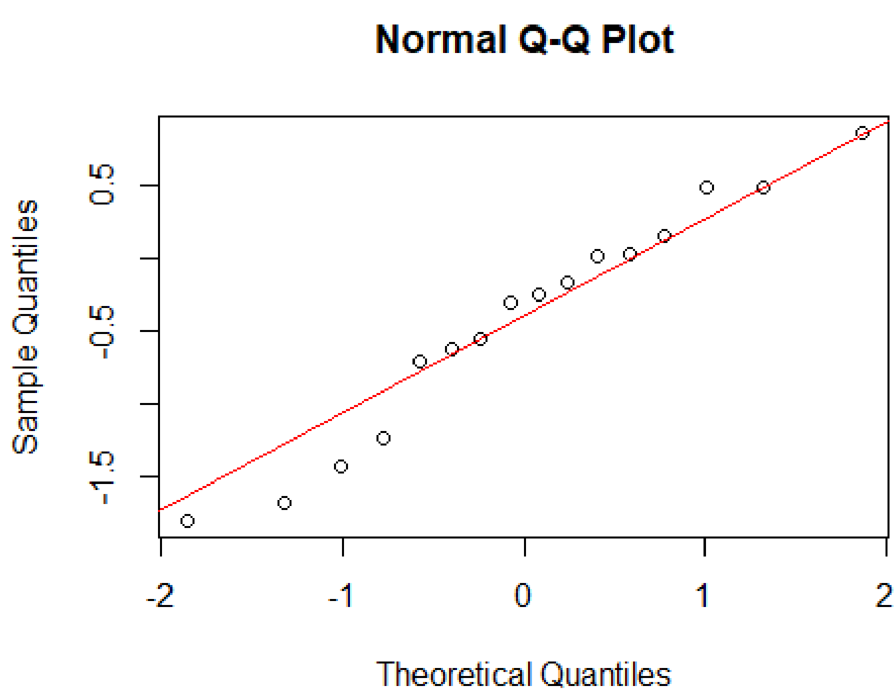


Fig 8

From the BoxCox Transformed Time series plot - fig 8 and fig 1, it can be seen that there is no much change in the series and can also check from the following

Output 4



QQ-plot- from the output 4 it can be concluded that the series is not stationary

Output 5

```
##
##  Shapiro-Wilk normality test
##
## data:  BC.egg2
## W = 0.95863, p-value = 0.6371
```

Shapiro-wilk- from the output 5 the p-value is 0.6371 which is >0.05 and hence we reject the null hypothesis.

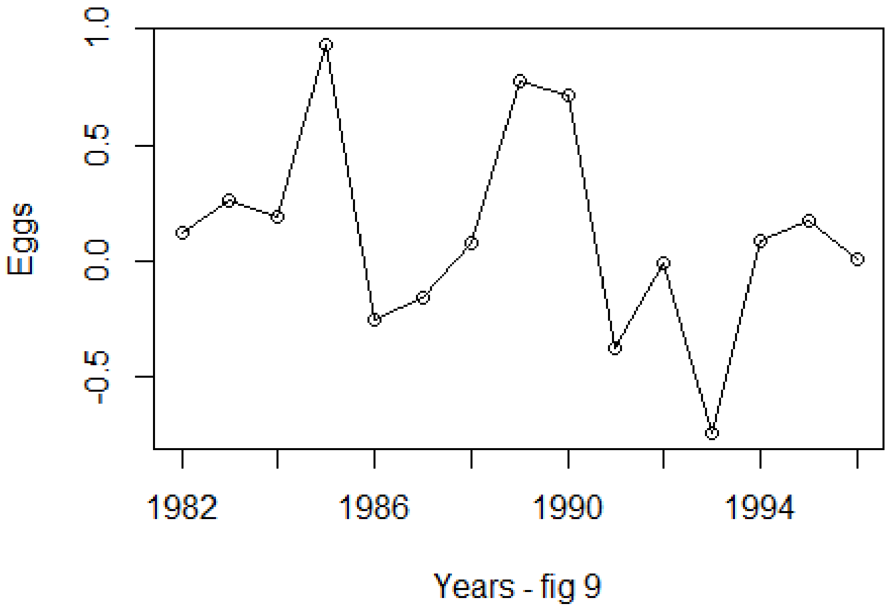
Output 6

```
##
##  Augmented Dickey-Fuller Test
##
## data:  BC.egg2
## Dickey-Fuller = -1.6454, Lag order = 2, p-value = 0.7075
## alternative hypothesis: stationary
```

Adf test- from the output 6 The p-value is 0.7075 which is >0.05 and conclude that to reject the null hypothesis.

So from the tests it can be are concluded that futher differencing is required to reduce the non-stationarity of the series.

- **Differencing**



After the first differencing is applied it can be observed that the series has been detrended when compared to fig 1, but there is a downward trend present in the series

As the plot is showing trend it is best to apply the stastical test which “adf.test()” or “ar()” , “adfTest()” to check if the series is stationary as the data is small and the plot can mislead.

<u>Output 7</u>

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1.BC.egg2
## Dickey-Fuller = -3.7522, Lag order = 2, p-value = 0.03913
## alternative hypothesis: stationary
```

ADF test for the differenced series From the Output 7 the p-value is 0.03913 which is <0.05 and can be concluded that the series is stationary. Since the series is stationary it is not required to further difference the series.

After appliying the tests, the models can be obtained from the acf and pacf plots

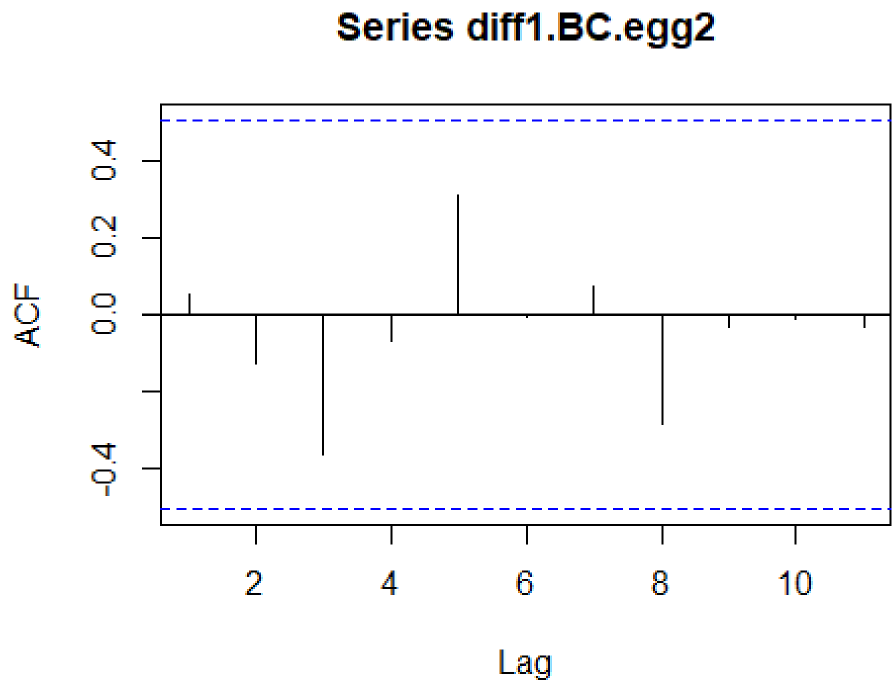


Fig 10

AFC Plot From the acf plot the q can be obtained, from the fig 10 it can be seen that the acf plot has white noise.

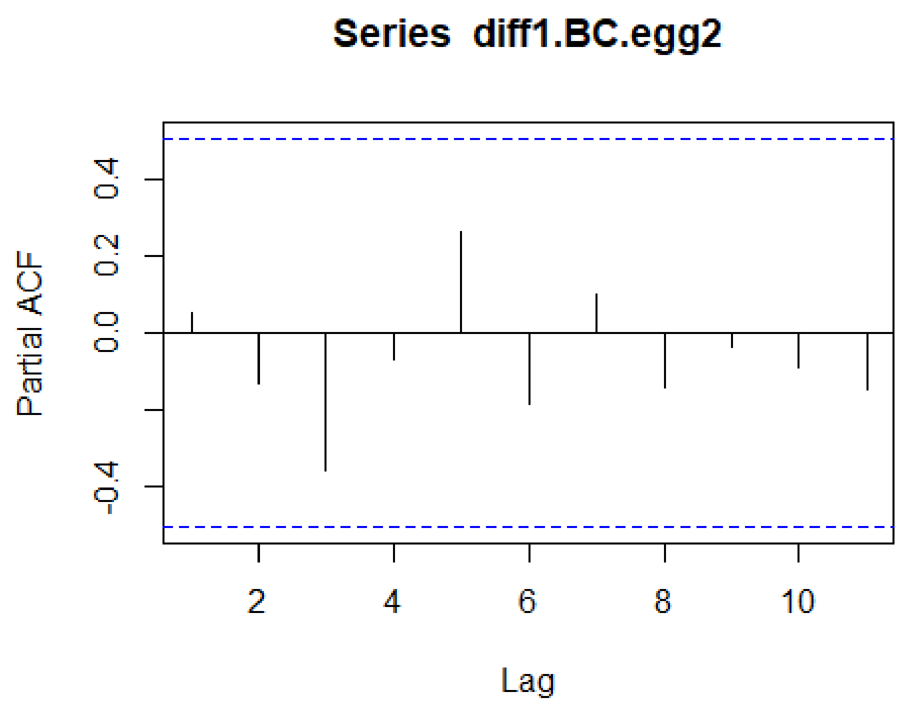


Fig 11

PACF plot from the pacf plot the p can be obtained, from the fig 11 it can be seen that it has a white noise.

To conclude we have no models from the ACF and PACF plots as whiote noise is present and implies that there are no significant lags in it and also says that the series is stationary and has no covariance.

- **EACF**

Next, we plot the EACF plot to find suitable models for the data set. EACF Eacf is another way to obtain models

<u>Output 8</u>

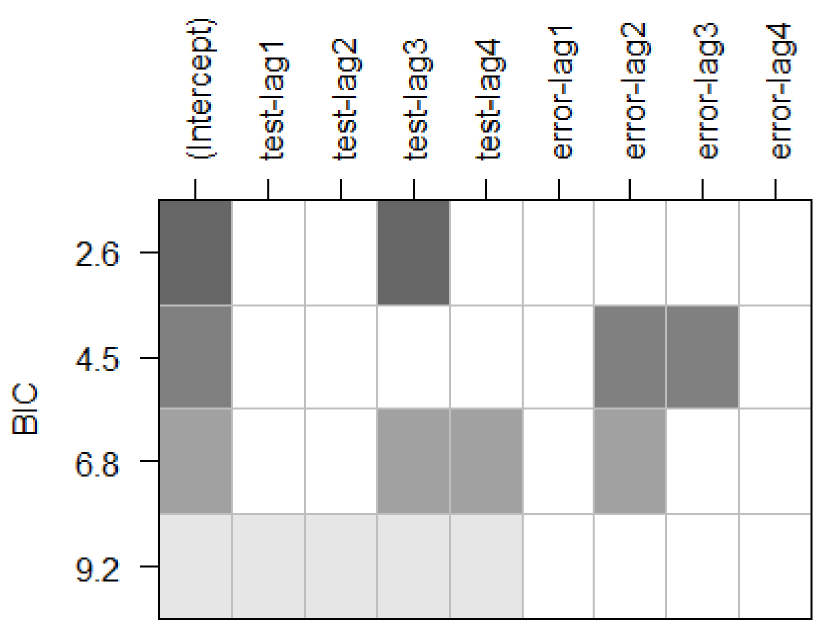
```
## AR/MA
##   0 1 2 3
## 0 o o o o
## 1 o o o o
## 2 o o o o
## 3 o o o o
```

From the output 8 EACF plot the following models can be obtained -

ARIMA (0,1,0), ARIMA (0,1,1), ARIMA (1,1,1), ARIMA (1,1,0)

Next we use the Bayesian Information Criteria to obtain some more models. Here the yule wakers method is applied to obtain the bic.

- **BIC**



<u>Fig 12</u>

From the fig 12 BIC table we can consider the below models are good and from the table it is observed that all the models are significant.

ARIMA(3,1,2), ARIMA(3,1,3), ARIMA(0,1,2), ARIMA(0,1,3), ARIMA(3,1,0)

Finally the following models can be considered for analysis.

ARIMA (0,1,1), ARIMA (0,1,2), ARIMA (0,1,3), ARIMA (1,1,0), ARIMA (1,1,1), ARIMA (3,1,0), ARIMA (3,1,2), ARIMA (3,1,3).

The above models are tested for the significance and then the best model is considered to forecast.The model fitted should have values $p < 0.05$ which means it is a suitable model.

Fitting of the Models

We are using both the 'css' and 'ml' methods and checking for the best fit model.

ARIMA(0,1,1) - Output 9.1

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.12664    0.24577  0.5153   0.6064
```

ARIMA(0,1,1) - Output 9.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.11948    0.23992   0.498   0.6185
```

ARIMA(0,1,2) - Output 10.1

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 0.145001   0.368813  0.3932   0.6942
## ma2 0.030648   0.442854  0.0692   0.9448
```

ARIMA(0,1,2) - Output 10.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 0.129177   0.320053  0.4036   0.6865
## ma2 0.016932   0.360021  0.0470   0.9625
```

ARIMA(0,1,3) - Output 11.1

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.42468    0.19305  2.1998 0.027819 *
## ma2  0.14240    0.18315  0.7775 0.436882
## ma3 -0.65373    0.21024 -3.1094 0.001875 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(0,1,3) - Output 11.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.22194    0.37290  0.5952   0.5517
## ma2  0.13924    0.23941  0.5816   0.5609
## ma3 -0.37961    0.42459 -0.8940   0.3713
```

ARIMA(1,1,0) - Output 12.1

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.12332    0.25558  0.4825   0.6294
```

ARIMA(1,1,0) - Output 12.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.11617    0.24846  0.4676   0.6401
```

ARIMA(1,1,1) - Output 13.1

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.142493   0.994614  0.1433   0.8861
## ma1 -0.021365   1.077051 -0.0198   0.9842
```

ARIMA(1,1,1) - Output 13.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.015043   0.947858  0.0159   0.9873
## ma1 0.105441   0.916084  0.1151   0.9084
```

ARIMA(3,1,0) - Output 14.1

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.087518   0.246775  0.3546   0.7229
## ar2 -0.042376   0.246586 -0.1719   0.8636
## ar3 -0.274669   0.245108 -1.1206   0.2625
```

ARIMA(3,1,0) - Output 14.2

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.104276   0.240440  0.4337   0.6645
## ar2 -0.038795   0.236748 -0.1639   0.8698
## ar3 -0.236277   0.230058 -1.0270   0.3044
```

ARIMA(3,1,2) - Output 15.1

```
##
## z test of coefficients:
##
##     Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.24342    0.25908   4.7993 1.592e-06 ***
## ar2 -0.78634    0.36037  -2.1820   0.02911 *
## ar3  0.10957    0.29753   0.3683   0.71268
```

ARIMA(3,1,2) - Output 15.2

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.142493   0.994614  0.1433   0.8861
## ma1 -0.021365   1.077051 -0.0198   0.9842
```

ARIMA(3,1,3) - Output 16.1

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.028538   0.224922  0.1269  0.899036
## ar2  0.720783   0.265172  2.7182  0.006564 **
## ar3  0.855584   0.270990  3.1573  0.001593 **
## ma1 -0.549303   0.154757 -3.5495  0.000386 ***
## ma2 -1.262396   0.251078 -5.0279 4.959e-07 ***
## ma3 -2.029212   0.285493 -7.1077 1.180e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(3,1,3) - Output 16.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.35535    0.84421  0.4209   0.6738
## ar2 -0.65043    0.64694 -1.0054   0.3147
## ar3 -0.43144    0.77973 -0.5533   0.5800
## ma1 -0.40731    0.97076 -0.4196   0.6748
## ma2  0.78275    0.84468  0.9267   0.3541
## ma3  0.30430    0.88849  0.3425   0.7320
```

It is also considered good to overfit the p and q to check if we can get a better model which is. ARIMA(1,1,1), ARIMA(0,1,2, ARIMA(2,1,1) and ARIMA(2,1,2). The models ARIMA(1,1,1), ARIMA(0,1,2) are already checked hence checking the significance of model ARIMA(2,1,1) and ARIMA(2,1,2)

ARIMA(2,1,1) - Output 17.1

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  0.989249   0.020679  47.8377  < 2e-16 ***
## ar2  0.072170   0.039757   1.8153  0.06948 .
## ma1 -1.719144   0.063919 -26.8958  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,1,1) - Output 17.2

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.53076    0.73180  0.7253   0.4683
## ar2 -0.16383    0.25007 -0.6551   0.5124
## ma1 -0.40953    0.70997 -0.5768   0.5641
```

ARIMA(2,1,2) - Output 18.1

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.238449   0.033315   7.1573 8.226e-13 ***
## ar2  1.076549   0.069153  15.5676 < 2.2e-16 ***
## ma1 -0.660590   0.059157 -11.1667 < 2.2e-16 ***
## ma2 -1.952721   0.130845 -14.9239 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,1,2) - Output 18.2

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  0.8889279         NA      NA        NA
## ar2 -0.9986042  0.0030488 -327.54 < 2.2e-16 ***
## ma1 -0.9037186         NA      NA        NA
## ma2  0.9771132         NA      NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the outputs we can see that the models ARIMA(3,1,3) and ARIMA(2,1,2) gives all significant coeffecients

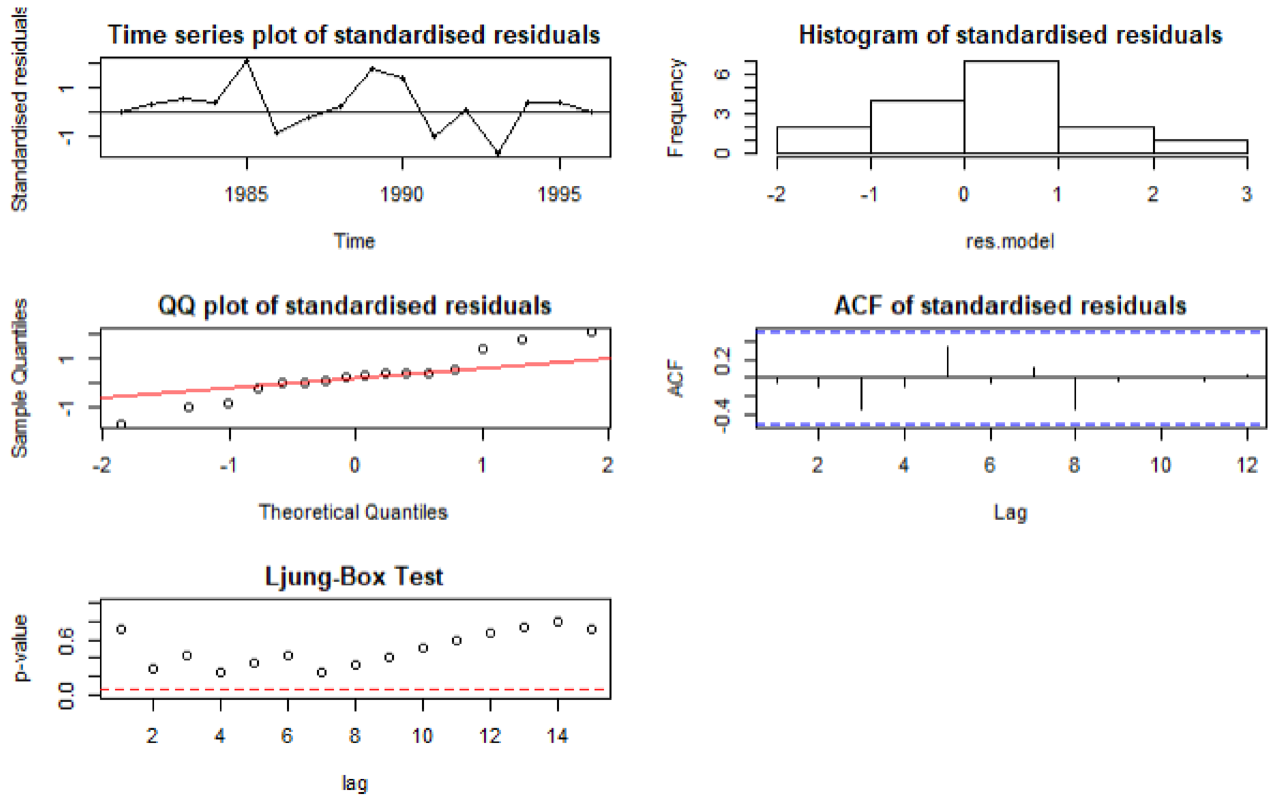
The best way to analyse the models is by sorting out the aic and bic score and conclude the best model. We will use the sort.score function to do the sorting. output 19 output 20

From the output 19 & 20 it can be seen the best lowest model is ARIMA(0,1,1) And hence the models ARIMA(0,1,1), ARIMA(2,1,2) and ARIMA(3,1,3) is used for residual analysis.

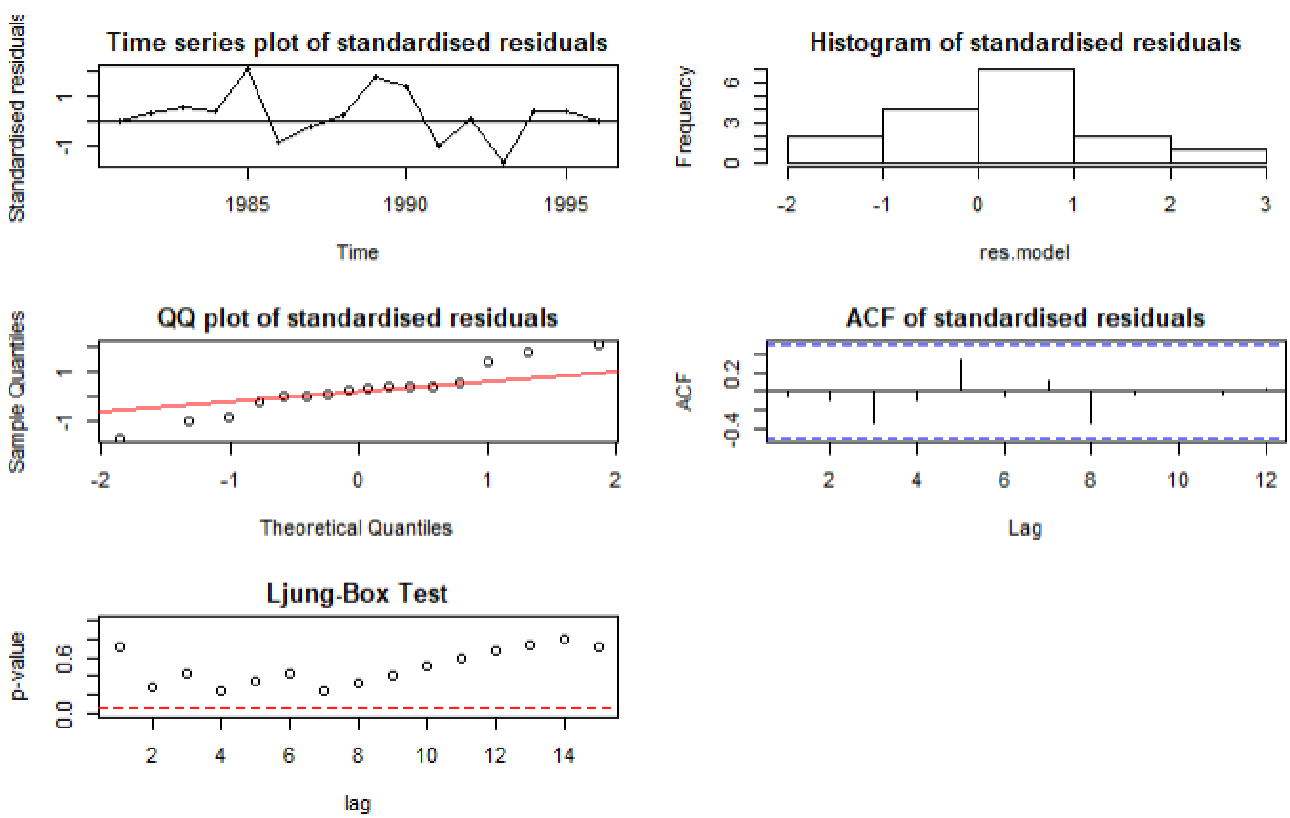
- ## Residual Analysis

Residual analysis is the best way to check if the model is fitted in the best possible way.

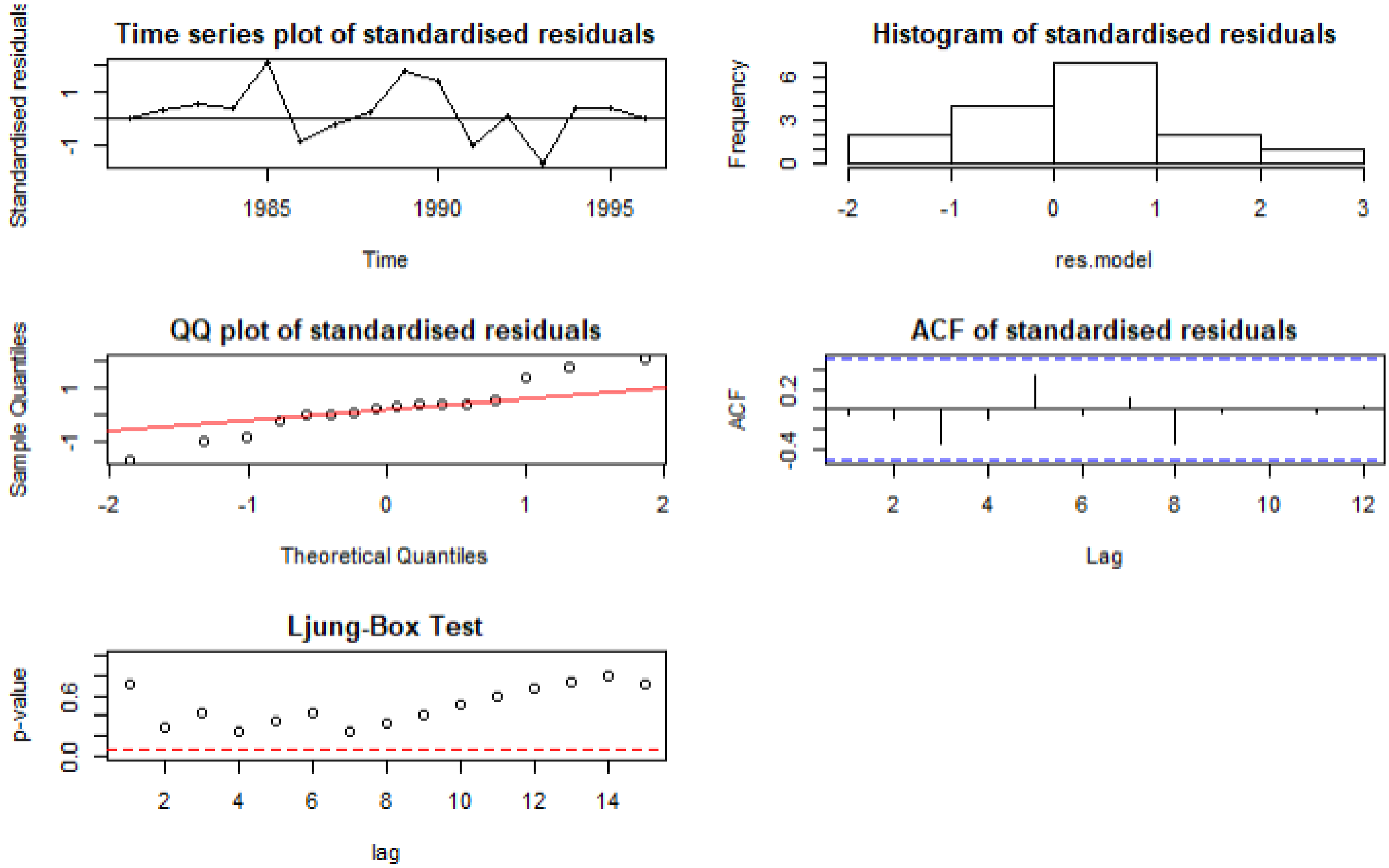
<u>Output 21</u>



<u>Output 22</u>



Output 23



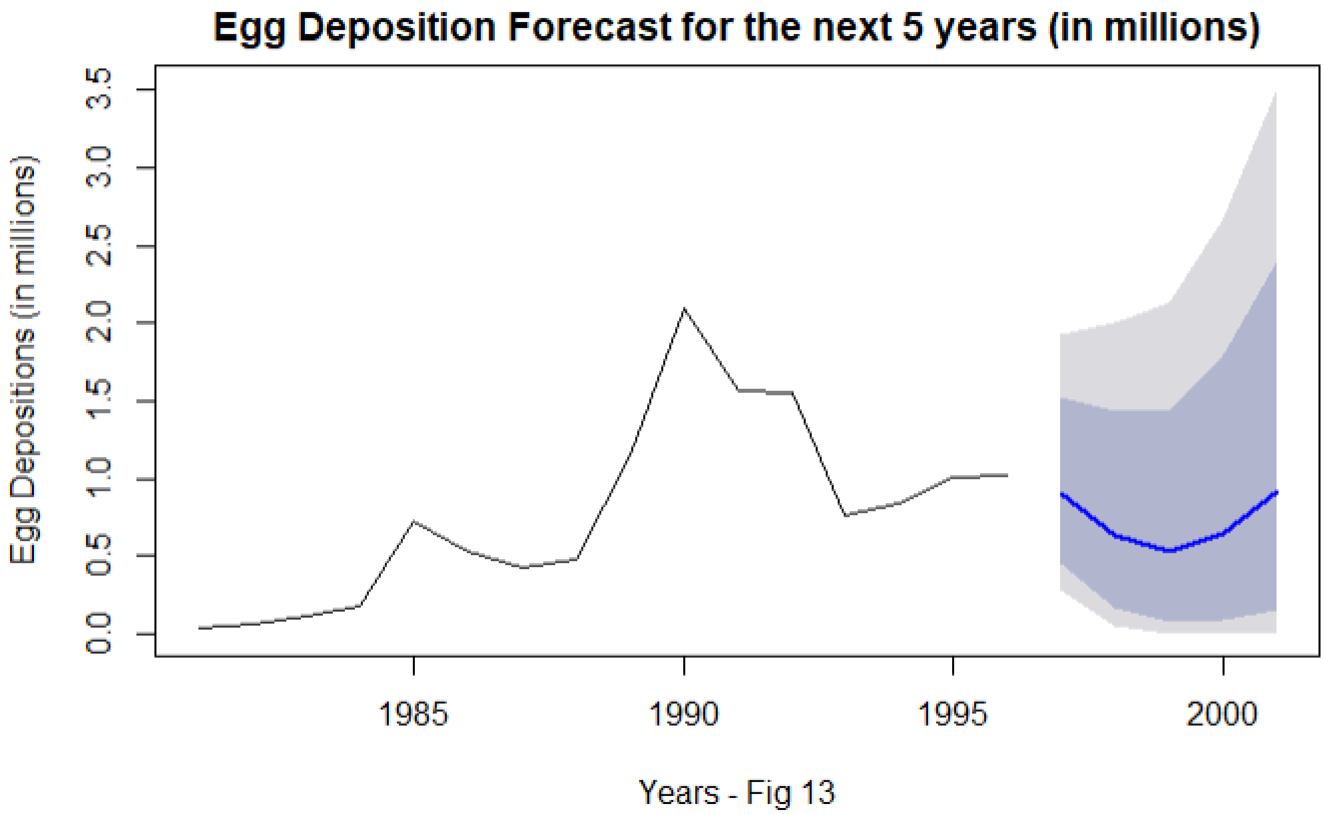
From the residual analysis (outputs 21,22,23) it can be observed that

The Shapiro-wilk test give the first best model with normality to be ARIMA(2,1,2) whith the highest p-value of 0.8819 followed by ARIMA(3,1,3) with p-value = 0.6635 and followed by ARIMA(0,1,1) with p-value = 0.5109.

Based on the Residual Analysis of the five plots, it can be seen that model ARIMA(2,1,2) in terms of all the plots, The Histogram looks symmetric except for the model ARIMA(3,1,3) and hence excluding the model. On the other hand all the plots are similar between the models ARIMA(2,1,2) and ARIMA(0,1,1) with the Acf plot showing white noise, the histogram are symmetric,the time series plot after differencing is having the points between 3 and -3 which is accepted and the trend is reduced, the L-jung box has all the points above the red line but the QQ-plot of the model ARIMA(2,1,2) is much better as the points are near to the red line. And finally considering the best model to be ARIMA(2,1,2) and hence forecasting with this model.

Forecasting

Based on the chosen model ARIMA(2,1,2) the forecast have been plotted for the next 5 years



Conclusion

Based on the models fitted and the residual analysis carried on the models it is colcluded that ARIMA(2,1,2) seems to forecast that there is going to be decrease and then raise in the deposition of eggs. The blue line shows the forecast line and the blue area shous the 85% interval and the grey area shows the 95% interval.

Appendix

```
# reading the required packages
library(readr)
library(dplyr)
library(TSA)
library(tseries)
library(fUnitRoots)
library(lmtest)
library(forecast)

#importing the data and preprocessing.

eggs <- read.csv("eggs.csv")

rownames(eggs)<- seq(from=1981,to=1996)

egg1<-select (eggs,-c('year'))

egg2<-(ts(as.vector(egg1),start = 1981))

class(egg2)

## [1] "ts"

plot(egg2, type='o', ylab='Eggs', xlab = 'Time - fig 1', main="Time series plot of Eggs")
```

```
#fit the linear model
model1 = lm(egg2~time(egg2))

#ploting the model

plot(ts(fitted(model1)),
     ylim = c(min(c(fitted(model1),
                    as.vector(egg2))),
              max(c(fitted(model1),
                    as.vector(egg2)))),
     ylab='ozone layer',
     xlab= 'Time - fig2' ,
     main ="Fitted linear trend ")
lines(as.vector(egg2),type="o")
```

```
#fit the quadratic model
t = time(egg2)
t2 = t^2
model2 = lm(egg2~t+t2)# label the model as model2

#ploting the model
plot(ts(fitted(model2)),
```

```r
     ylim = c(min(c(fitted(model2),
                    as.vector(egg2))),
              max(c(fitted(model2),
                    as.vector(egg2)))),
     ylab='ozone layer',
     xlab = 'Time - fig 3',
     main ="Fitted quadratic curve ")

lines(as.vector(egg2),type="o")
```

```r
# ploting the series
plot(egg2,type='o',ylab="Eggs", xlab = 'Time - fig 1', main='Time series plot of Eggs')
```

we will check if the series has stationarity

```r
#qq plot for the data
qqnorm(egg2)
qqline(egg2, col =2, lwd =1, lty =2, xlab = "Theoretical Quantiles - Fig 4")
```

```r
# output 1
#shapiro wilk test
shapiro.test(egg2)

##
##  Shapiro-Wilk normality test
##
## data:  egg2
## W = 0.94201, p-value = 0.3744
```

the p value is > 0.05 and is not normal

```r
#par(mfrow=c(1,2))

# fig 5
acf(egg2) #Trend is apparent from ACF and PACf plots
```

```r
# from pacf we get AR
#AR(1)

#fig 6
pacf(egg2)
```

```r
#output 2
#Apply ADF test with default settings
adf.test(egg2)

##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  egg2
## Dickey-Fuller = -2.0669, Lag order = 2, p-value = 0.5469
## alternative hypothesis: stationary

#fig 7
# Let's first apply the box-Cox transformation.
egg2.transform = BoxCox.ar(egg2,method = "yw")
```

```
# output 3
egg2.transform$ci

## [1] 0.1 0.8
```

The class interval is between 0.1 and 0.8, and considering 0.4 as it is above the 95% ci.

```
# appliying the transformation using lamda
lambda = 0.4
BC.egg2 = (egg2^lambda-1)/lambda

plot(BC.egg2, type = "o", ylab = "Egg Depositions",xlab= 'Time - fig 8', main = "Transfor
med")
```

```
#output_4
#qq plot
qqnorm(BC.egg2)
qqline(BC.egg2, col = 2)
```

```
#output_5
shapiro.test(BC.egg2)

##
##  Shapiro-Wilk normality test
##
## data:  BC.egg2
## W = 0.95863, p-value = 0.6371

adf.test(BC.egg2)

##
##  Augmented Dickey-Fuller Test
##
## data:  BC.egg2
## Dickey-Fuller = -1.6454, Lag order = 2, p-value = 0.7075
## alternative hypothesis: stationary

diff1.BC.egg2 = diff(BC.egg2)
#fig 9
plot(diff1.BC.egg2,type='o',ylab='Eggs', xlab='Years - fig 9')
```

```r
# output 7
#adf.test() should have p < 0.05
adf.test(diff1.BC.egg2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1.BC.egg2
## Dickey-Fuller = -3.7522, Lag order = 2, p-value = 0.03913
## alternative hypothesis: stationary
```

```r
#fig 10
acf(diff1.BC.egg2)
```

```r
# fig 11
pacf(diff1.BC.egg2)
```

```r
# output 8
eacf(diff1.BC.egg2, ar.max=3,ma.max=3)
```

```
## AR/MA
##   0 1 2 3
## 0 o o o o
## 1 o o o o
## 2 o o o o
## 3 o o o o
```

```r
#BCI
#from the plot we get
# test = ar = p
# error = ma = q

bic = armasubsets(y=diff1.BC.egg2,nar=4,nma=4,y.name='test',ar.method="yw")
#bic = armasubsets(y=Eggs_BoxCox.diff,nar=3,nma=6,y.name='test',ar.method='yw')
plot(bic)
```

```r
# ARIMA(0,1,1)
# output 9.1
model_011_css = arima(BC.egg2,order=c(0,1,1),method='CSS')
coeftest(model_011_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.12664    0.24577  0.5153   0.6064
```

```r
# ARIMA(0,1,1)
# output 9.2
model_011_ml = arima(BC.egg2,order=c(0,1,1),method='ML')
coeftest(model_011_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.11948    0.23992   0.498   0.6185
```

```
# ARIMA(0,1,2)
#output 10.1
model_012_css = arima(BC.egg2,order=c(0,1,2),method='CSS')
coeftest(model_012_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 0.145001   0.368813  0.3932   0.6942
## ma2 0.030648   0.442854  0.0692   0.9448
```

```
# ARIMA(0,1,2)
#output 10.2
model_012_ml = arima(BC.egg2,order=c(0,1,2),method='ML')
coeftest(model_012_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 0.129177   0.320053  0.4036   0.6865
## ma2 0.016932   0.360021  0.0470   0.9625
```

```
# ARIMA(0,1,3)
#output 11.1
model_013_css = arima(BC.egg2,order=c(0,1,3),method='CSS')
coeftest(model_013_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.42468    0.19305  2.1998 0.027819 *
## ma2  0.14240    0.18315  0.7775 0.436882
## ma3 -0.65373    0.21024 -3.1094 0.001875 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ARIMA(0,1,3)
#output 11.2
model_013_ml = arima(BC.egg2,order=c(0,1,3),method='ML')
coeftest(model_013_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1  0.22194    0.37290  0.5952   0.5517
```

```
## ma2  0.13924    0.23941  0.5816   0.5609
## ma3 -0.37961    0.42459 -0.8940   0.3713
```

```r
# ARIMA(1,1,0)
#output 12.1
model_110_css = arima(BC.egg2,order=c(1,1,0),method='CSS')
coeftest(model_110_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.12332    0.25558  0.4825   0.6294
```

```r
# ARIMA(1,1,0)
#output 12.2
model_110_ml = arima(BC.egg2,order=c(1,1,0),method='ML')
coeftest(model_110_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.11617    0.24846  0.4676   0.6401
```

```r
# ARIMA(1,1,1)
#output 13.1
model_111_css = arima(BC.egg2,order=c(1,1,1),method='CSS')
coeftest(model_111_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.142493   0.994614  0.1433   0.8861
## ma1 -0.021365   1.077051 -0.0198   0.9842
```

```r
# ARIMA(1,1,1)
#output 13.2
model_111_ml = arima(BC.egg2,order=c(1,1,1),method='ML')
coeftest(model_111_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.015043   0.947858  0.0159   0.9873
## ma1 0.105441   0.916084  0.1151   0.9084
```

```r
# ARIMA(3,1,0)
#output 14.1
model_310_css = arima(BC.egg2,order=c(3,1,0),method='CSS')
coeftest(model_310_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.087518   0.246775  0.3546   0.7229
## ar2 -0.042376   0.246586 -0.1719   0.8636
## ar3 -0.274669   0.245108 -1.1206   0.2625
```

```
# ARIMA(3,1,0)
#output 14.2
model_310_ml = arima(BC.egg2,order=c(3,1,0),method='ML')
coeftest(model_310_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.104276   0.240440  0.4337   0.6645
## ar2 -0.038795   0.236748 -0.1639   0.8698
## ar3 -0.236277   0.230058 -1.0270   0.3044
```

```
# ARIMA(3,1,2)
#output 15.1
model_312_css = arima(BC.egg2,order=c(3,1,2),method='CSS')
coeftest(model_312_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.24342    0.25908   4.7993 1.592e-06 ***
## ar2 -0.78634    0.36037  -2.1820   0.02911 *
## ar3  0.10957    0.29753   0.3683   0.71268
## ma1 -1.74773    0.12398 -14.0972 < 2.2e-16 ***
## ma2  1.45176    0.13565  10.7021 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ARIMA(3,1,2)
#output 15.2
model_312_ml = arima(BC.egg2,order=c(3,1,2),method='ML')
coeftest(model_312_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.60464    0.95999  0.6298  0.52880
## ar2 -0.87296    0.36010 -2.4242  0.01534 *
## ar3 -0.14520    0.39995 -0.3631  0.71656
## ma1 -0.63511    1.12517 -0.5645  0.57245
## ma2  0.99982    0.42150  2.3721  0.01769 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# ARIMA(3,1,3)
#output 16.1
model_313_css = arima(BC.egg2,order=c(3,1,3),method='CSS')
coeftest(model_313_css)

##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  0.028538   0.224922  0.1269  0.899036
## ar2  0.720783   0.265172  2.7182  0.006564 **
## ar3  0.855584   0.270990  3.1573  0.001593 **
## ma1 -0.549303   0.154757 -3.5495  0.000386 ***
## ma2 -1.262396   0.251078 -5.0279 4.959e-07 ***
## ma3 -2.029212   0.285493 -7.1077 1.180e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# ARIMA(3,1,3)
#output 16.2
model_313_ml = arima(BC.egg2,order=c(3,1,3),method='ML')
coeftest(model_313_ml)

##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.35535    0.84421  0.4209   0.6738
## ar2 -0.65043    0.64694 -1.0054   0.3147
## ar3 -0.43144    0.77973 -0.5533   0.5800
## ma1 -0.40731    0.97076 -0.4196   0.6748
## ma2  0.78275    0.84468  0.9267   0.3541
## ma3  0.30430    0.88849  0.3425   0.7320

# ARIMA(2,1,1)
# output 17.1
model_211_css = arima(BC.egg2,order=c(2,1,1),method='CSS')
coeftest(model_211_css)

##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1  0.989249   0.020679  47.8377  < 2e-16 ***
## ar2  0.072170   0.039757   1.8153  0.06948 .
## ma1 -1.719144   0.063919 -26.8958  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# ARIMA(2,1,1)
# output 17.2
model_211_ml = arima(BC.egg2,order=c(2,1,1),method='ML')
coeftest(model_211_ml)

##
## z test of coefficients:
```

```
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.53076    0.73180  0.7253   0.4683
## ar2 -0.16383    0.25007 -0.6551   0.5124
## ma1 -0.40953    0.70997 -0.5768   0.5641
```

```r
# ARIMA(2,1,2)
# output 18.1
model_212_css = arima(BC.egg2,order=c(2,1,2),method='CSS')
coeftest(model_212_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.238449   0.033315   7.1573 8.226e-13 ***
## ar2  1.076549   0.069153  15.5676 < 2.2e-16 ***
## ma1 -0.660590   0.059157 -11.1667 < 2.2e-16 ***
## ma2 -1.952721   0.130845 -14.9239 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# ARIMA(2,1,2)
# output 18.2
model_212_ml = arima(BC.egg2,order=c(2,1,2),method='ML')
coeftest(model_212_ml)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value  Pr(>|z|)
## ar1  0.8889279         NA      NA        NA
## ar2 -0.9986042  0.0030488 -327.54 < 2.2e-16 ***
## ma1 -0.9037186         NA      NA        NA
## ma2  0.9771132         NA      NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Aic and Bic

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

# output 18
sort.score(AIC(model_011_ml,model_111_ml,model_110_ml,model_312_ml,model_313_ml,model_012
_ml,model_013_ml,model_310_ml,model_211_ml,model_212_ml), score = "aic")
```

```
##              df      AIC
## model_011_ml  2 21.85969
```

```
## model_110_ml  2 21.87276
## model_012_ml  3 23.85741
## model_111_ml  3 23.85945
## model_013_ml  4 24.35319
## model_310_ml  4 24.80620
## model_212_ml  5 25.43231
## model_211_ml  4 25.60749
## model_312_ml  6 27.32993
## model_313_ml  7 29.23900

# output 19
sort.score(BIC(model_011_ml,model_111_ml,model_110_ml,model_312_ml,model_313_ml,model_012
_ml,model_013_ml,model_310_ml,model_211_ml,model_212_ml), score = "bic" )

##              df      BIC
## model_011_ml  2 23.27579
## model_110_ml  2 23.28886
## model_012_ml  3 25.98156
## model_111_ml  3 25.98360
## model_013_ml  4 27.18539
## model_310_ml  4 27.63840
## model_211_ml  4 28.43969
## model_212_ml  5 28.97256
## model_312_ml  6 31.57823
## model_313_ml  7 34.19535

#Residual Analysis

residual.analysis <- function(model, std = TRUE){
  library(TSA)
  library(FitAR)
  if (std == TRUE){
    res.model = rstandard(model)
  }else{
    res.model = residuals(model)
  }
  par(mfrow=c(3,2))
  plot(res.model,type='o',ylab='Standardised residuals', main="Time series plot of standa
rdised residuals")
  abline(h=0)
  hist(res.model,main="Histogram of standardised residuals")
  qqnorm(res.model,main="QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  acf(res.model,main="ACF of standardised residuals")
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = length(model$residuals)-1 , StartLag = k + 1, k = 0, Squar
edQ = FALSE)
}

# output 21
residual.analysis(model = model_011_ml)

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

```
##
## data:  res.model
## W = 0.95132, p-value = 0.5109
```

```r
# output 22
residual.analysis(model = model_011_ml)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.95132, p-value = 0.5109
```

```r
# output 23
residual.analysis(model = model_011_ml)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.95132, p-value = 0.5109
```

```r
# forecasting
# Fig 13
fit = Arima(egg2,c(2,1,2),lambda = 0.45)
plot(forecast(fit,h=5), xlab ="Years - Fig 13", ylab ='Egg Depositions (in millions)', ma
in = 'Egg Deposition Forecast for the next 5 years (in millions)')
```