RMIT UNIVERSITY

MATH1318
Time Series Analysis
Assignment 2

Student Name: Ancy Rex

Student Number : s3760058

Introduction

Lake Huron Bloaters are a species of whitefish(family of salmoniade) found in the freshwater. The silvery coloured fish with pink and purple reflecting skin mostly found in Great Lakes and in Lake Nipigon. This species of fish inhabits the underwater slopes. The 25.5cm herring like fish is also listed as exposed to global extinction by the IUCN Red List organisation(Coregonus hoyi, 2020).

Methodology

The egg deposition of the coregonus hoyi (in millions) between a time period (1981-1996) is used as the dataset for analysing. Using analysis methods to choose the best model to provide the forecast for the egg deposition for the next 5 years.

Time Series Plot

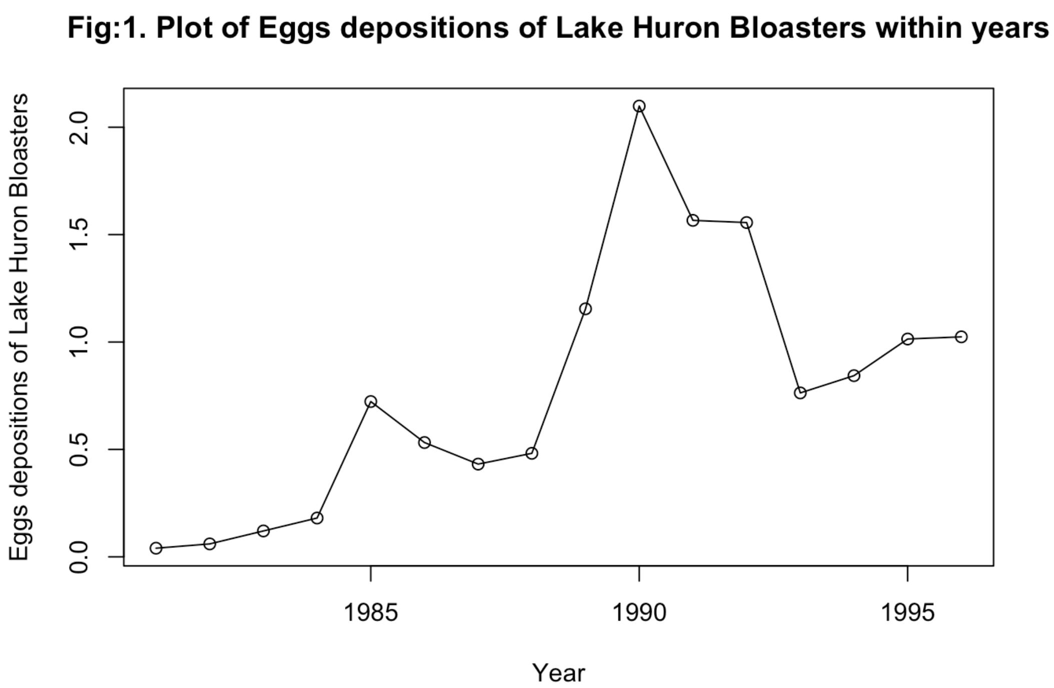


FIGURE : 1 TIME SERIES PLOT OF EGG DEPOSITION DATA

The time series graph for the egg deposition of Lake Huron bloaters were plotted using R code(Appendix 3.). By observing the graph (figure 1)for the egg deposition it seems as an auto regressive with an upward trend. There is a change in variance as there is a sudden peaking in the year 1990. Although there is no seasonability in the graph.

ACF and PACF

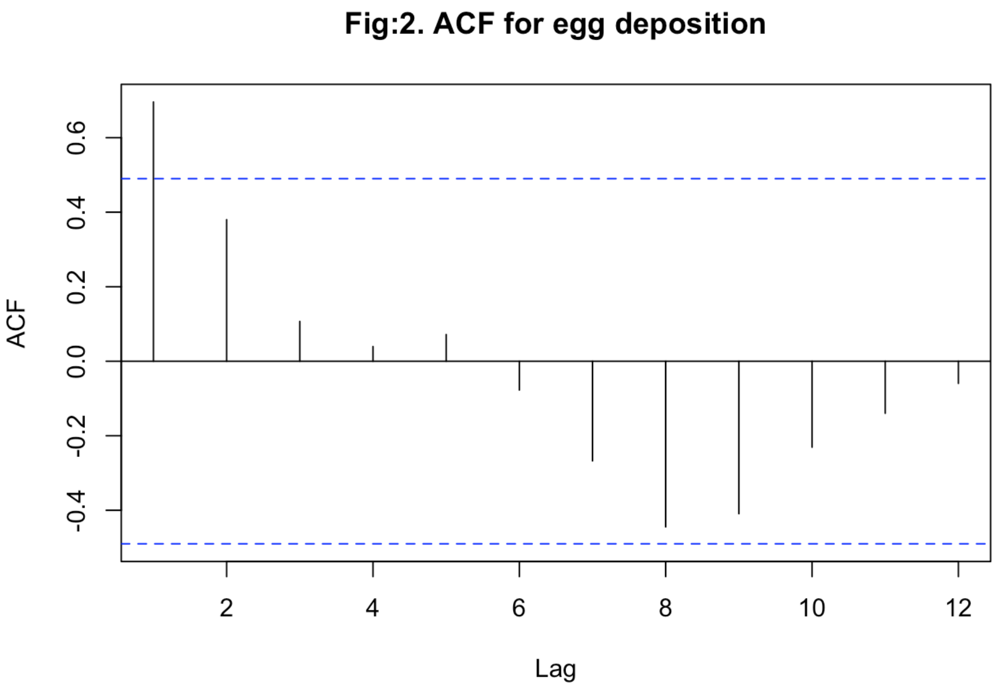


FIGURE 2: ACF OF EGG DEPOSITION DATA

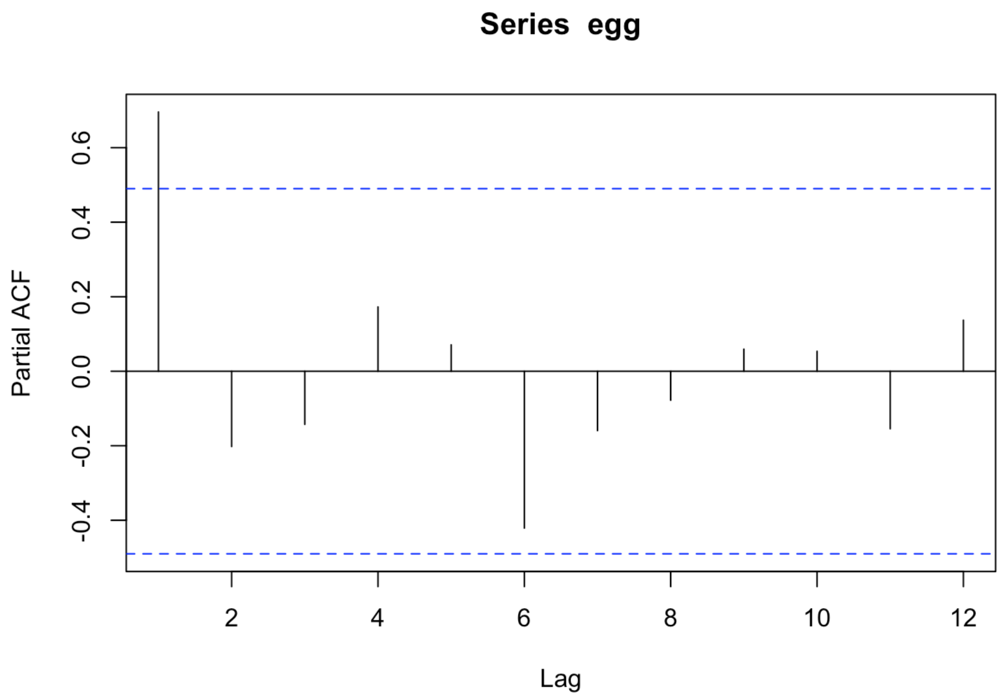


FIGURE 3 : PACF OF EGG DEPOSITION DATA

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

ADF TEST

The ACF and PACF graph for the egg deposition of Lake Huron bloaters were plotted using R code(Appendix 4). The ADF test was also done to find out if the series was stationary or not. The R code for the ADF test is also present in (Appendix 4). In the ACF and PACF graph it was found to only have returned one significant lag.

The ADF test gave a p-value of 0.546 which is greater than 0.05, and so we cannot reject the null hypothesis. Although the alternate hypothesis observes series to be stationary the null hypothesis proves the series to be non -stationary.

Scatter Plot

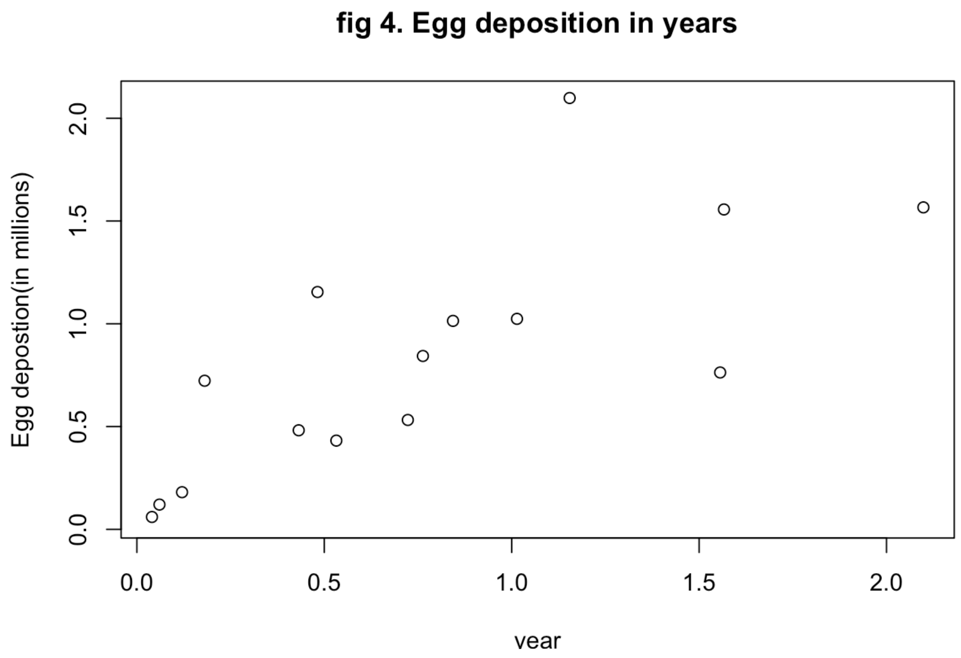


FIGURE 4: SCATTER PLOT OF EGG DEPOSITION DATA

```
## [1] 0.7445657
```

The scatter plot for egg deposition and correlation of the graph was found using the R code in Appendix 5. The scatter plot for the egg deposition was also found to be a positive upward trend. The correlation was found to be 75% which also proves the upward trend in the plot.

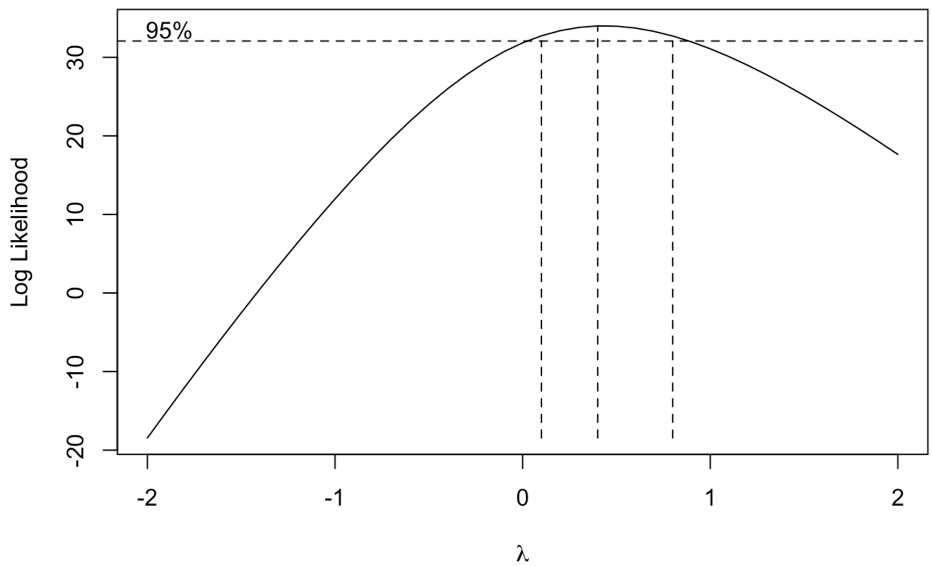


FIGURE 5: TRANSFORMING WAS DONE ON EGG DEPOSITION DATA

Transforming data

The data was transformed using Box-Cox transformation (Yule-Walker), where we obtained the interval for lambda values (Appendix 6) . The mid point of the interval was taken as the lambda value(0.45) . Later the QQplot that was plotted using R code(Appendix 6)does not seem to give normal distribution .

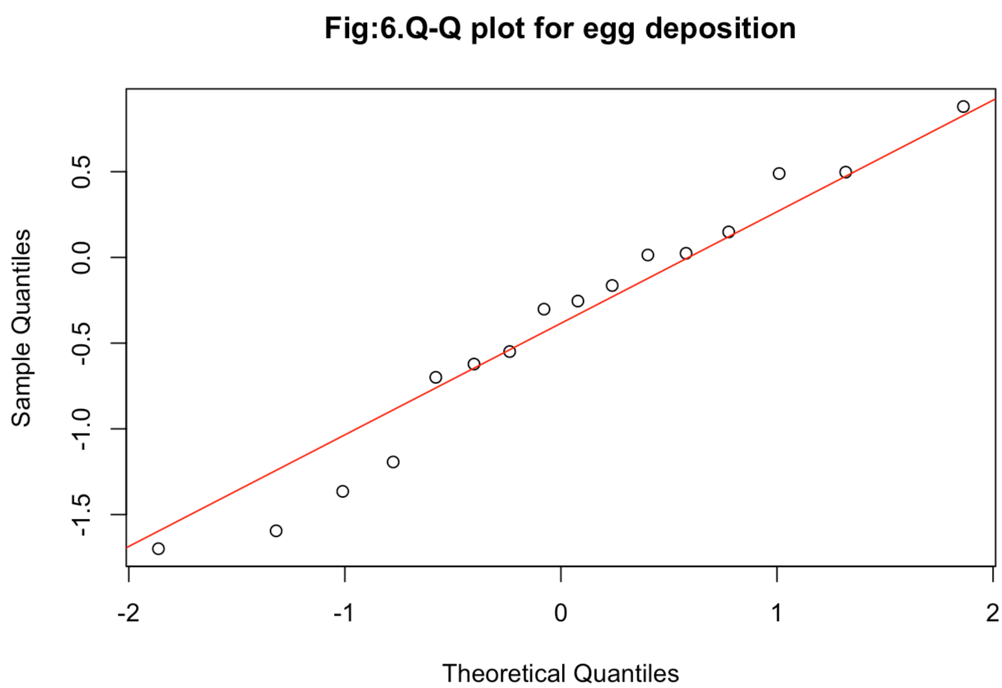


FIGURE 6: Q-Q PLOT FOR TRANSFORMED EGG DEPOSITION DATA

```
 Augmented Dickey-Fuller Test

data:  BC.egg
Dickey-Fuller = -1.6769, Lag order = 2, p-value = 0.6955
alternative hypothesis: stationary
```

ADF TEST FOR TRANSFORMED DATA

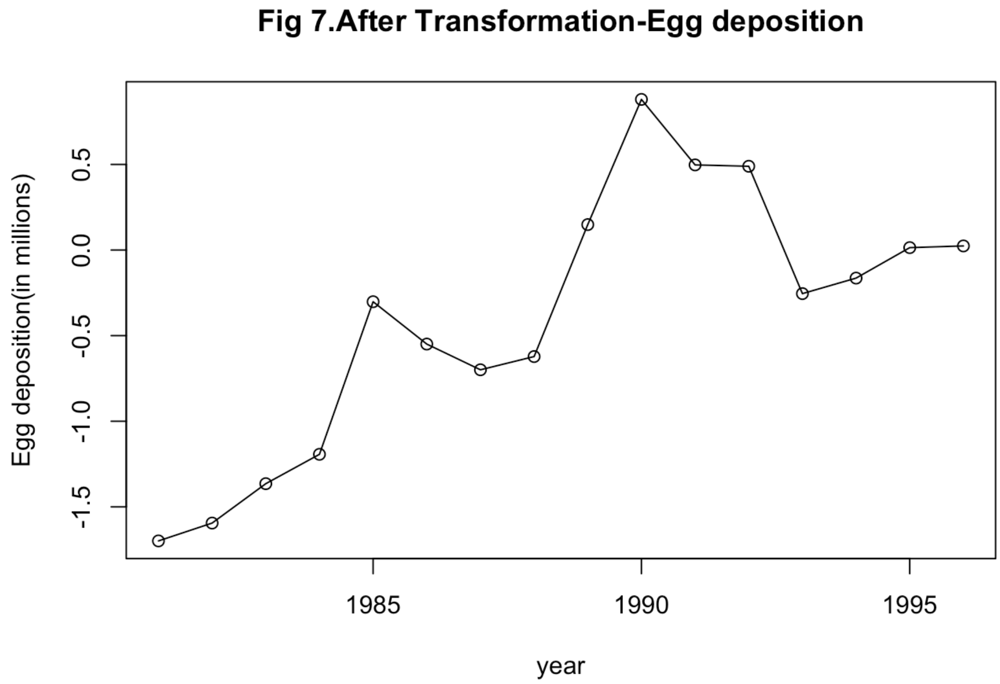


FIGURE 7: TIME SERIES AFTER TRANSFORMATION

ADF test was done (Appendix 6) to find the p-value and it was found to be 0.6955 which is greater than 0.05 and so we reject the null hypothesis. From the null hypothesis, the series was still found to be non-stationary. The time series plot (Appendix 6)does not seem to change even after the transformation and so it was necessary to difference the data.

Differencing the data

The differenced data was plotted and ADF test was done with R code(Appendix 7). The plot shows significant change after differencing. The p-value found to be 0.0443 which is less than 0.05 and so we fail to reject the null hypothesis. By observing the null hypothesis series was found to be stationary.

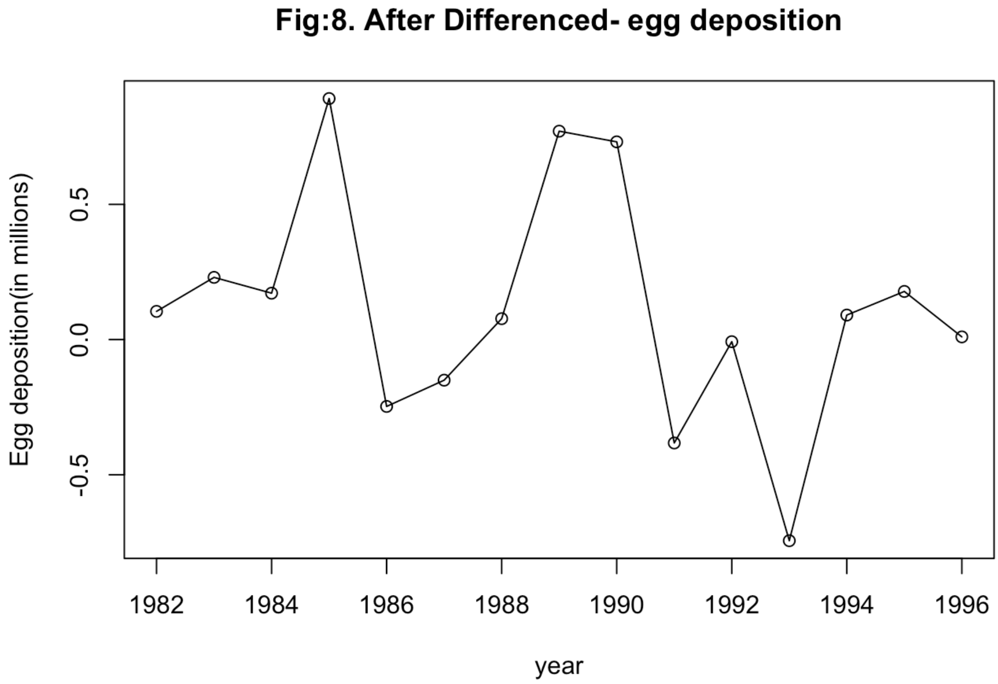


FIGURE 8:TIME SERIES FOR DATA AFTER DIFFERENCING

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff.BC.egg
## Dickey-Fuller = -3.6798, Lag order = 2, p-value = 0.0443
## alternative hypothesis: stationary
```

ADF TEST AFTER DIFFERING OF THE EGG DEPOSITION DATA

ACF and PACF was plotted on the differenced data using the code given in Appendix 7. Both ACF and PACF which were plotted with the differenced data does not show any significant lag. With the stationary series we proceed with the fitting.

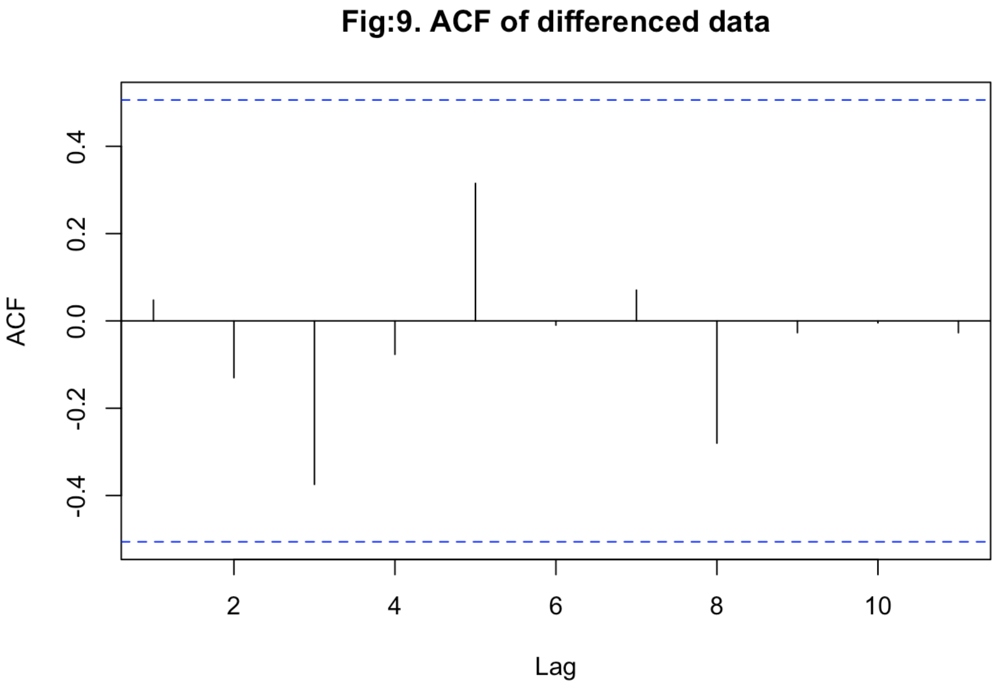


FIGURE 9: ACF FOR DIFFERENCED EGG DEPOSITION DATA

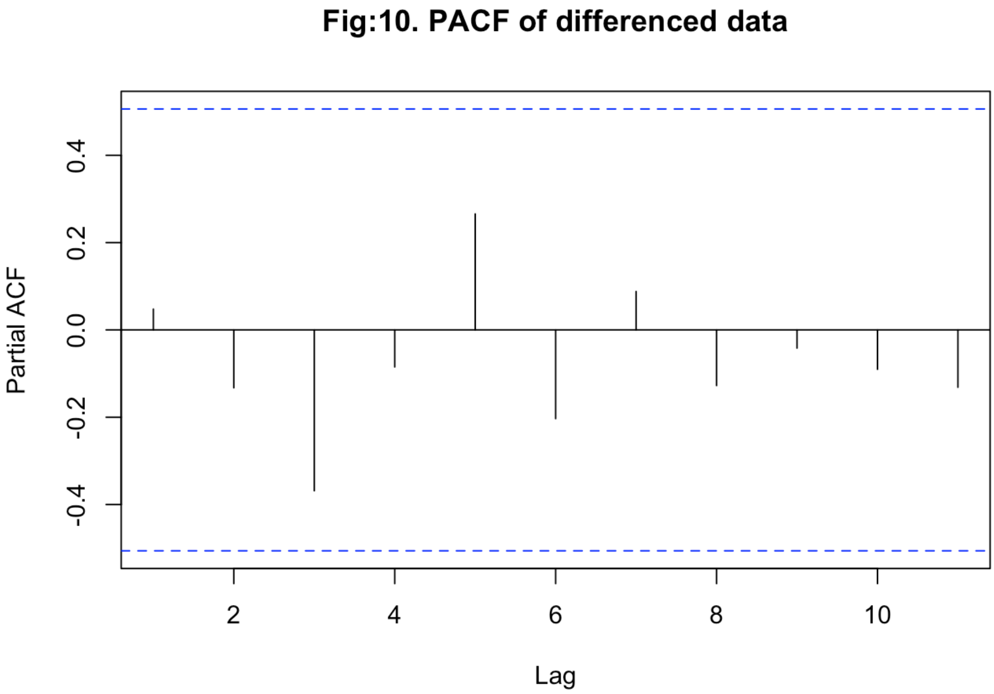


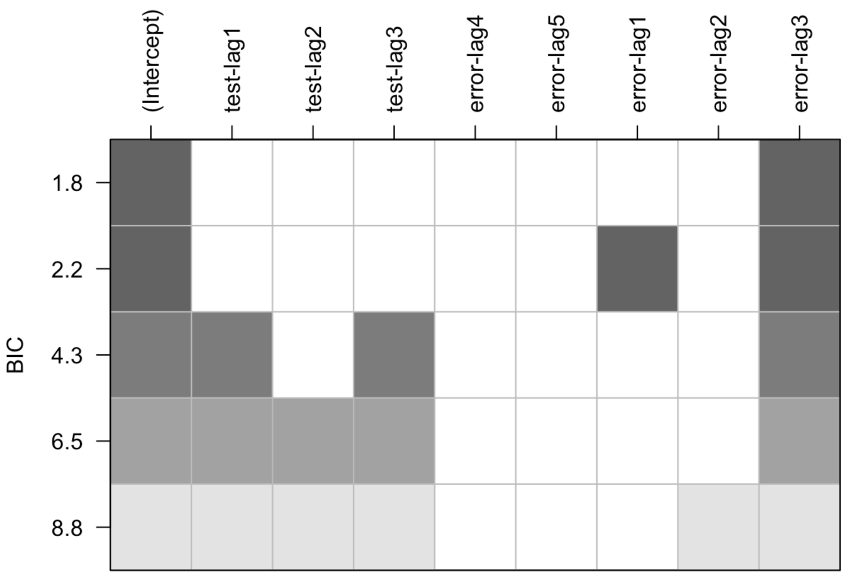
FIGURE 10: PACK FOR DIFFERENCED EGG DEPOSITION DATA

Model Fitting

In the eacf table, p = 0 and q = 0 were found and thus presence of white noise. The models observed from the eacf were

```
## 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
```

EACF FOR EGG DEPOSITION



BIC TABLE WAS CREATED FOR THE DIFFERENCED EGG DEPOSITION DATA

ARMIA(1,1,0),ARMIA(0,1,1) and ARMIA(0,1,0). The EACF table was constructed using the code in Appendix 8.

BIC was plotted using the yule-walker method (Appendix 8). The shaded columns present in the table corresponds to the coefficients AR(1),AR(2),AR(3),MA(1),MA(2),MA(3). The other possible cases are ARIMA(1,1,1),ARIMA(1,1,2),ARMIA(1,1,3),ARMIA(2,1,1),ARMIA(2,1,2),ARMIA(2,1,3),ARMIA(3,1,1),ARIMA(3,1,2),ARMIA(3,1,3).

Coefficient test was done on all the possible cases. Using R code (Appendix 9).

ARIMA(1,1,1)-CSS

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.104270   0.937663  0.1112   0.9115
## ma1 0.010553   0.985521  0.0107   0.9915
```

ARIMA(1,1,1)-ML

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 0.0016698  0.9762099  0.0017   0.9986
## ma1 0.1100801  0.9440422  0.1166   0.9072
```

ARIMA(1,1,2) - CSS

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.3216704  0.0098951 133.5688 < 2.2e-16 ***
## ma1 -1.6167108  0.1992406  -8.1144 4.883e-16 ***
## ma2 -0.1451761  0.2980578  -0.4871    0.6262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(1,1,2) - ML

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.48074    1.02030  0.4712   0.6375
## ma1 -0.39876    1.00089 -0.3984   0.6903
## ma2 -0.11072    0.21204 -0.5222   0.6016
```

ARIMA(1,1,3) - CSS

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -0.30535    0.30773 -0.9923 0.321054
## ma1  0.63953    0.23913  2.6744 0.007485 **
## ma2  0.29070    0.25582  1.1363 0.255812
## ma3 -0.64695    0.24329 -2.6592 0.007833 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(1,1,3) - ML

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.28276    0.36723 -0.7700   0.4413
## ma1  0.59223    0.48220  1.2282   0.2194
## ma2  0.35365    0.51319  0.6891   0.4907
## ma3 -0.56055    0.43277 -1.2953   0.1952
```

ARIMA(2,1,1) - CSS

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  1.050797   0.021500  48.8733   <2e-16 ***
## ar2  0.027527   0.043590   0.6315   0.5277
## ma1 -1.728384   0.071280 -24.2479   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,1,1) - ML

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.54117    0.69894  0.7743   0.4388
## ar2 -0.17120    0.24660 -0.6942   0.4875
## ma1 -0.43082    0.67707 -0.6363   0.5246
```

ARIMA(2,1,2) - CSS

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ar1  0.448873   0.033166 13.5340  < 2e-16 ***
## ar2  0.848674   0.074302 11.4219  < 2e-16 ***
## ma1 -0.998290   0.488128 -2.0451  0.04084 *
## ma2 -1.371110   0.779222 -1.7596  0.07848 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,1,2) - ML

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.41464    1.23676  0.3353 0.737426
## ar2 -0.90854    0.55015 -1.6514 0.098647 .
## ma1 -0.22562    1.56251 -0.1444 0.885185
## ma2  0.99971    0.35986  2.7781 0.005468 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,1,3) - CSS

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.48241    0.20386 -7.2718 3.549e-13 ***
## ar2 -0.76201    0.18765 -4.0607 4.892e-05 ***
## ma1  2.05225    0.11418 17.9738 < 2.2e-16 ***
## ma2  2.39688    0.19321 12.4058 < 2.2e-16 ***
## ma3  1.26282    0.19926  6.3376 2.333e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,1,3) - ML

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -0.49878    0.58800 -0.8483   0.3963
## ar2 -0.27760    0.43377 -0.6400   0.5222
## ma1  0.76951    0.59712  1.2887   0.1975
## ma2  0.60221    0.64615  0.9320   0.3513
## ma3 -0.35403    0.52187 -0.6784   0.4975
```

ARIMA(3,1,1) - CSS

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.146836   0.439287  0.3343   0.7382
## ar2 -0.046805   0.241328 -0.1939   0.8462
## ar3 -0.389182   0.245075 -1.5880   0.1123
## ma1 -0.167991   0.443927 -0.3784   0.7051
```

ARIMA(3,1,1) - ML

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.064090   0.553536  0.1158   0.9078
## ar2 -0.039712   0.242038 -0.1641   0.8697
## ar3 -0.250850   0.229731 -1.0919   0.2749
## ma1  0.031813   0.548900  0.0580   0.9538
```

ARIMA(3,1,2) - CSS

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.236391   0.259979   4.7557 1.977e-06 ***
## ar2 -0.783025   0.358715  -2.1829   0.02905 *
## ar3  0.097833   0.298755   0.3275   0.74331
## ma1 -1.770577   0.117307 -15.0936 < 2.2e-16 ***
## ma2  1.483194   0.128004  11.5871 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(3,1,2) - ML

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.51649    1.04689  0.4934  0.62176
## ar2 -0.84463    0.33909 -2.4909  0.01274 *
## ar3 -0.17694    0.36206 -0.4887  0.62506
## ma1 -0.54251    1.29037 -0.4204  0.67417
## ma2  0.99996    0.45441  2.2006  0.02777 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(3,1,3) - CSS

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.070013   0.157264 -0.4452 0.6561797
## ar2  0.772288   0.237952  3.2456 0.0011722 **
## ar3  1.067447   0.228729  4.6669 3.058e-06 ***
## ma1 -0.510946   0.140179 -3.6450 0.0002674 ***
## ma2 -1.375963   0.200337 -6.8682 6.500e-12 ***
## ma3 -2.211406   0.258208 -8.5644 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(3,1,3) - ML

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  0.33478    0.85519  0.3915   0.6954
## ar2 -0.66116    0.60223 -1.0978   0.2723
## ar3 -0.42614    0.74304 -0.5735   0.5663
## ma1 -0.38936    1.02179 -0.3811   0.7032
## ma2  0.81219    0.81831  0.9925   0.3209
## ma3  0.28031    0.85773  0.3268   0.7438
```

ARIMA(0,1,1) - CSS

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1  0.11857    0.24784  0.4784   0.6324
```

ARIMA(0,1,1) - ML

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 0.030394   0.254473  0.1194   0.9049
```

ARIMA(1,1,0) - CSS

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.11393    0.25603   0.445   0.6563
```

ARIMA(1,1,0) - ML

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.10720    0.24864  0.4311   0.6664
```

All the significant cases are ARIMA(1,1,0),ARMIA(1,1,0),ARMIA(1,1,1) and ARMIA(3,1,3). The AIC and BIC values are found for the following significant cases. ML gives better and most likely estimation when considering other parameters.

	df <dbl>	AIC <dbl>
model_011_ml	2	20.50570
model_110_ml	2	21.45089
model_111_ml	3	23.43715
model_313_ml	7	28.70119

4 rows

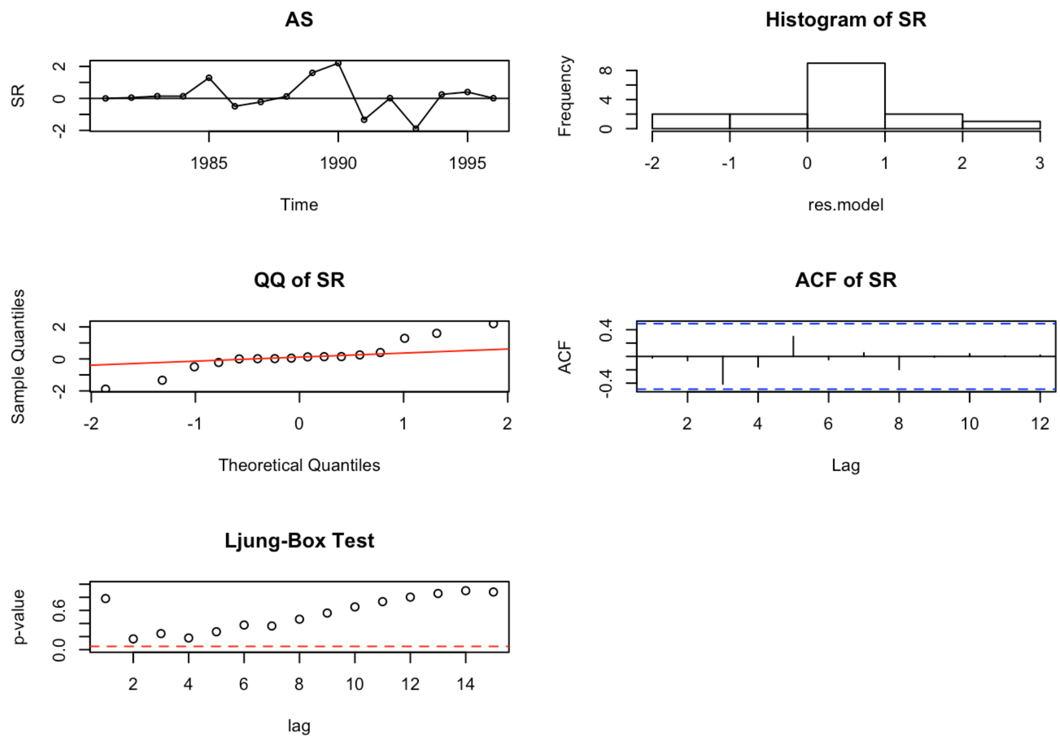
	df <dbl>	BIC <dbl>
model_011_ml	2	21.92180
model_110_ml	2	22.86699
model_111_ml	3	25.56131
model_313_ml	7	33.65755

4 rows

Using the sort score function we found the values of AIC and BIC, code in Appendix 10.The AIC and BIC values are the least for ARIMA(0,1,1) model(ML) ,thus making it the best model.

Residual Analysis

Later the residual analysis for the model ARIMA(0,1,1)(ML) was found and code is present in Appendix 11.



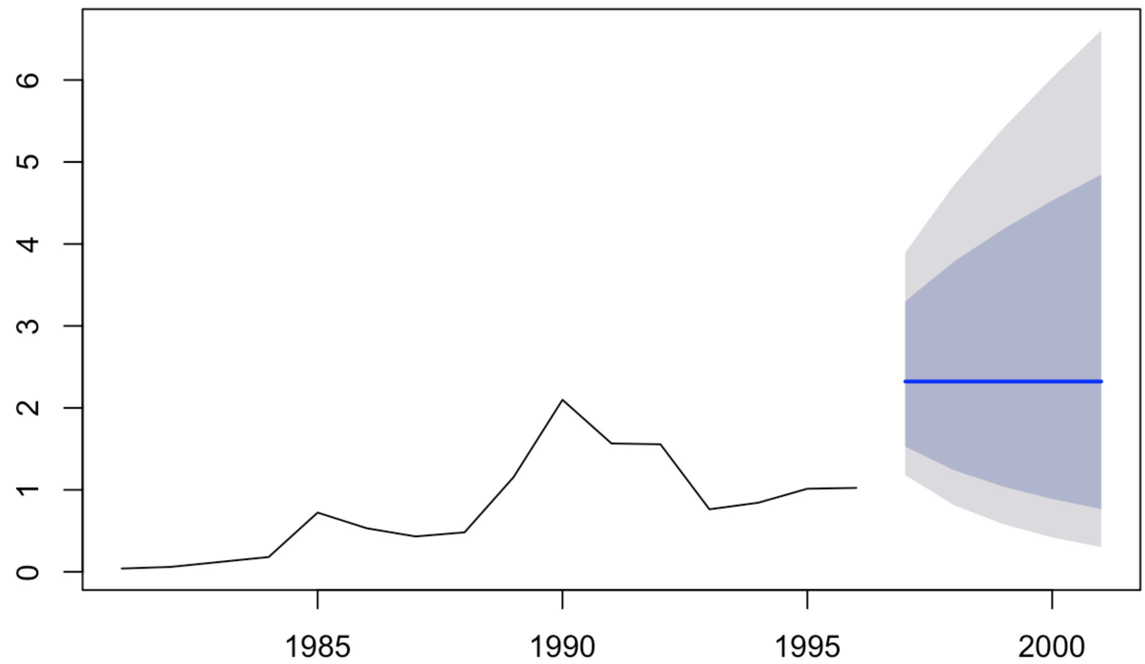
```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.91443, p-value = 0.1372
```

The residual analysis was found for ARIMA(0,1,1) appear normal. The model gives us a symmetric histogram and normally distributed QQ plot. P value present in the Shapiro-Wilk normality test was 0.1372 which is greater than 0.05 and so it was considered as normality cases.

Predicting and Forecasting

To forecast the egg deposition of Lake Huron Bloaters in millions for the next 5 years is done using the function predict (code in Appendix 12)

Fig 11: Forecast of egg deposition for next 5 years



Conclusion

The egg deposition of Lake Huron bloaters in millions between the years 1981-1996 was analysed using different analysing techniques. The data was converted to time series, using scatter plot and correlation it was found that there is an upward trend in the series. It was necessary to change the data from non stationary to stationary and so the data was transformed and differenced before modelling. Using EACF and BIC we were able to obtain certain models. Coefficient test was done on all models and those significant model cases were considered. The AIC and BIC values for all the significant models were found and model_011_ml had the least value which made it the best model out of all the others. Residual analysis was found for this specific model to obtain a symmetric histogram and normally distributed QQ plot. Using ARIMA(0,1,1) model predictions were done to forecast the next 5 years ,where it was found that it follows the same trend as the previous years of the egg deposition of Lake Huron bloaters.

Reference

1. En.wikipedia.org. 2020. Coregonus Hoyi. [online] Available at: <https://en.wikipedia.org/wiki/Coregonus_hoyi> [Accessed 10 May 2020].

Appendix

1. Setup

```
library(TSA)
library(forecast)
library(fUnitRoots)
library(ggplot2)
library(tseries)
library(tidyverse)
library(sandwich)
library(lmtest)
library(bestglm)
library(FitAR)
source("/Users/ancy_rex/Documents/S3\ /Time\ Series\ Analysis/Assignments/
Assignment\ 2/sort.score.R")
```

2. Reading the data

```
# Reading the egg deposition.
egg <- read.csv("/Users/ancy_rex/Documents/S3\ /Time\ Series\ Analysis/
Assignments/Assignment\ 2/eggs.csv")
 #class of dataset
egg
```

3. Time Series Graph

```
egg <- ts(as.vector(egg$eggs), start=1981 , end=1996)
#plotting time series graph
plot(egg ,type='o',xlab='Year',ylab='Eggs depositions of Lake Huron Bloasters',main =
"Fig:1. Plot of Eggs depositions of Lake Huron Bloasters within years")
```

4. ACF and PACF

```
# To plot both ACF and PACF
 acf(egg,main='Fig:2. ACF for egg deposition')
pacf(egg,main='Fig:3. PACF for egg deposition')

#adf test
adf.test(egg)
```

5. Scatter Plot

```
plot(y=egg,x=zlag(egg),ylab='Egg depostion(in millions)',xlab = 'year',main='Fig 4.
Egg deposition in years')
y = egg
x = zlag(egg)
index = 2:length(x)
cor(y[index],x[index])
```

6. Transforming data

```
# Box-Cox Transformation (yule-walker)
```

```
egg.transform1 = BoxCox.ar(egg,method = "yule-walker",main='fig: 5.Box-Cox
transformation')
egg.transform1$ci

#lambda value obtained
lambda = 0.45
BC.egg = (egg^lambda-1)/lambda
qqnorm(BC.egg,main='Fig:6.Q-Q plot for egg deposition')
qqline(BC.egg, col = 2)
adf.test(BC.egg)

#plotting time series after transforming data
plot(BC.egg,type='o',xlab='year',ylab='Egg deposition(in millions)', main='Fig 7.After
Transformation-Egg deposition')
```

7. Differencing the data

```
diff.BC.egg = diff(BC.egg)
plot(diff.BC.egg,type='o',xlab='year',ylab='Egg deposition(in millions)',main='Fig:8.
After Differenced- egg deposition')

#adf test
adf.test(diff.BC.egg)

#acf of differenced data
acf(diff.BC.egg,main='Fig:9. ACF of differenced data')

#pacf of differenced data
pacf(diff.BC.egg,,main='Fig:10. PACF of differenced data')
```

8. Model Fitting

```
#eacf
eacf(diff.BC.egg,ar.max = 3,ma.max = 3)

#BIC table
res=armasubsets(y=diff.BC.egg, nar = 3, nma = 5, y.name ='test', ar.method = 'yw')
plot(res)
```

9. Coefficient Test

```
# ARIMA(1,1,1)-CSS
model_111_css = arima(BC.egg,order=c(1,1,1),method='CSS')
coeftest(model_111_css)

# ARIMA(1,1,1)-ML
model_111_ml = arima(BC.egg,order=c(1,1,1),method='ML')
coeftest(model_111_ml)

# ARIMA(1,1,2) - CSS
model_112_css = arima(BC.egg,order=c(1,1,2),method='CSS')
coeftest(model_112_css)
```

```
# ARIMA(1,1,2) - ML
model_112_ml = arima(BC.egg,order=c(1,1,2),method='ML')
coeftest(model_112_ml)

# ARIMA(1,1,3) - CSS
model_113_css = arima(BC.egg,order=c(1,1,3),method='CSS')
coeftest(model_113_css)

# ARIMA(1,1,3) - ML
model_113_ml = arima(BC.egg,order=c(1,1,3),method='ML')
coeftest(model_113_ml)

# ARIMA(2,1,1) - CSS
model_211_css = arima(BC.egg,order=c(2,1,1),method='CSS')
coeftest(model_211_css)

# ARIMA(2,1,1) - ML
model_211_ml = arima(BC.egg,order=c(2,1,1),method='ML')
coeftest(model_211_ml)

# ARIMA(2,1,2) - CSS
model_212_css = arima(BC.egg,order=c(2,1,2),method='CSS')
coeftest(model_212_css)

# ARIMA(2,1,2) - ML
model_212_ml = arima(egg,order=c(2,1,2),method='ML')
coeftest(model_212_ml)

# ARIMA(2,1,3) - CSS
model_213_css = arima(BC.egg,order=c(2,1,3),method='CSS')
coeftest(model_213_css)

# ARIMA(2,1,3) - ML
model_213_ml = arima(BC.egg,order=c(2,1,3),method='ML')
coeftest(model_213_ml)

# ARIMA(3,1,1) - CSS
model_311_css = arima(egg,order=c(3,1,1),method='CSS')
coeftest(model_311_css)

# ARIMA(3,1,1) - ML
model_311_ml = arima(BC.egg,order=c(3,1,1),method='ML')
coeftest(model_311_ml)

# ARIMA(3,1,2) - CSS
model_312_css = arima(BC.egg,order=c(3,1,2),method='CSS')
coeftest(model_312_css)

# ARIMA(3,1,2) - ML
```

```
model_312_ml = arima(BC.egg,order=c(3,1,2),method='ML')
coeftest(model_312_ml)

# ARIMA(3,1,3) - CSS
model_313_css = arima(BC.egg,order=c(3,1,3),method='CSS')
coeftest(model_313_css)

# ARIMA(3,1,3) - ML
model_313_ml = arima(BC.egg,order=c(3,1,3),method='ML')
coeftest(model_313_ml)

# ARIMA(0,1,1) - CSS
model_011_css = arima(BC.egg,order=c(0,1,1),method='CSS')
coeftest(model_011_css)

# ARIMA(0,1,1) - ML
model_011_ml = arima(egg,order=c(0,1,1),method='ML')
coeftest(model_011_ml)

# ARIMA(1,1,0) - CSS
model_110_css = arima(BC.egg,order=c(1,1,0),method='CSS')
coeftest(model_110_css)

# ARIMA(1,1,0) - ML
model_110_ml = arima(BC.egg,order=c(1,1,0),method='ML')
coeftest(model_110_ml)
```

10. Sort Score

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

#AIC values
sort.score(AIC(model_011_ml,model_110_ml,model_111_ml,model_313_ml),score =
“aic")

#BIC
sort.score(BIC(model_011_ml,model_110_ml,model_111_ml,model_313_ml),score =
“bic")
```

11. Residual Analysis

```
#residual analysis function
residual.analysis <- function(model, std=TRUE){
if(std ==TRUE)
{
  res.model = rstandard(model)
}else
{
  res.model = residuals(model)
}
par(mfrow=c(3,2))
plot(res.model,type='o',ylab="SR",main="AS")
abline(h=0)
hist(res.model,main="Histogram of SR")
qqnorm(res.model,main="QQ of SR")
qqline(res.model,col = 2)
acf(res.model,main="ACF of SR")
print(shapiro.test(res.model))
k=0
LBQPlot(res.model,lag.max = length(model$residuals)-1, StartLag =
k+1,k=0,SquaredQ = FALSE)
par(mfrow=c(1,1))
}

# model_011_ml residual analysis
residual.analysis(model = model_011_ml)
par(mfrow=c(1,1))
```

12. Forecasting and Predicting

```
#predicting
predict(model_011_ml,n.ahead = 5,newxreg = NULL,se.fit = TRUE)
#plotting forecast for egg deposition
fit=Arima(egg,c(0,1,1))
plot(forecast(fit,h=5,lambda = 0.45),main='Fig 11: Forecast of egg deposition for
next 5 years')
fit
```