# Assignment 2-Time Series (MATH 1318)

Code

Vamika Pardeshi-s3701024

May 11, 2019

# Setup-

- Installed and loaded the necessary package to reproduce the report.

Hide

```
install.packages("TSA")
library(TSA)
install.packages("forecast")
library(forecast)
install.packages("FSAdata")
library(FSAdata)
install.packages("tseries")
library(tseries)
install.packages("timeSeries")
library(timeSeries)
install.packages("lmtest")
library(lmtest)
install.packages("FitAR")
library(FitAR)
install.packages("dLagM")
library(dLagM)
install.packages("fUnitRoots")
library(fUnitRoots)
```

# Introduction-

Bloater (aka Coregonus hoyi) is a form/species of ‘freshwater whitefish’ that belongs to the family of Salmonidae and can be found in the ‘Great lakes’ and ‘Lake Nipigon’. The task is to carry out an analysis on the egg depositions of these ‘Lake Huron Bloaters’ between years 1981 and 1996, by using different methods of Time series analysis. The dataset is available in ‘BloaterLH’ of FSAdata package or the given ‘CSV’ file can also be used directly. This report comprises the analysis of the data while choosing the best model among a set of possible models for the provided dataset, followed by the forecasting of egg depositions for the coming/next 5 years.

# Data Import-

Imported the ‘eggs’ dataset into R by using the function read.csv().

Hide

```
setwd("C:\\Time series")
Bloater_egg <- read.csv("eggs.csv")
Bloater_egg
```

| year <int> | eggs <dbl> |
|---|---|
| 1981 | 0.0402 |
| 1982 | 0.0602 |
| 1983 | 0.1205 |
| 1984 | 0.1807 |
| 1985 | 0.7229 |
| 1986 | 0.5321 |
| 1987 | 0.4317 |
| 1988 | 0.4819 |
| 1989 | 1.1546 |
| 1990 | 2.0984 |

1-10 of 16 rows Previous **1** 2 Next

Since the ‘eggs’ column in the data contains the egg depositions, hence checking it’s class first.

Hide

```
class(Bloater_egg$eggs)
```

```
[1] "numeric"
```

# ANALYZING THE DATA- TIME SERIES PLOT

The class of the ‘eggs’ column came out to be numeric, hence first converting it into a ‘ts’ object and then plotting the time series plot for the data.

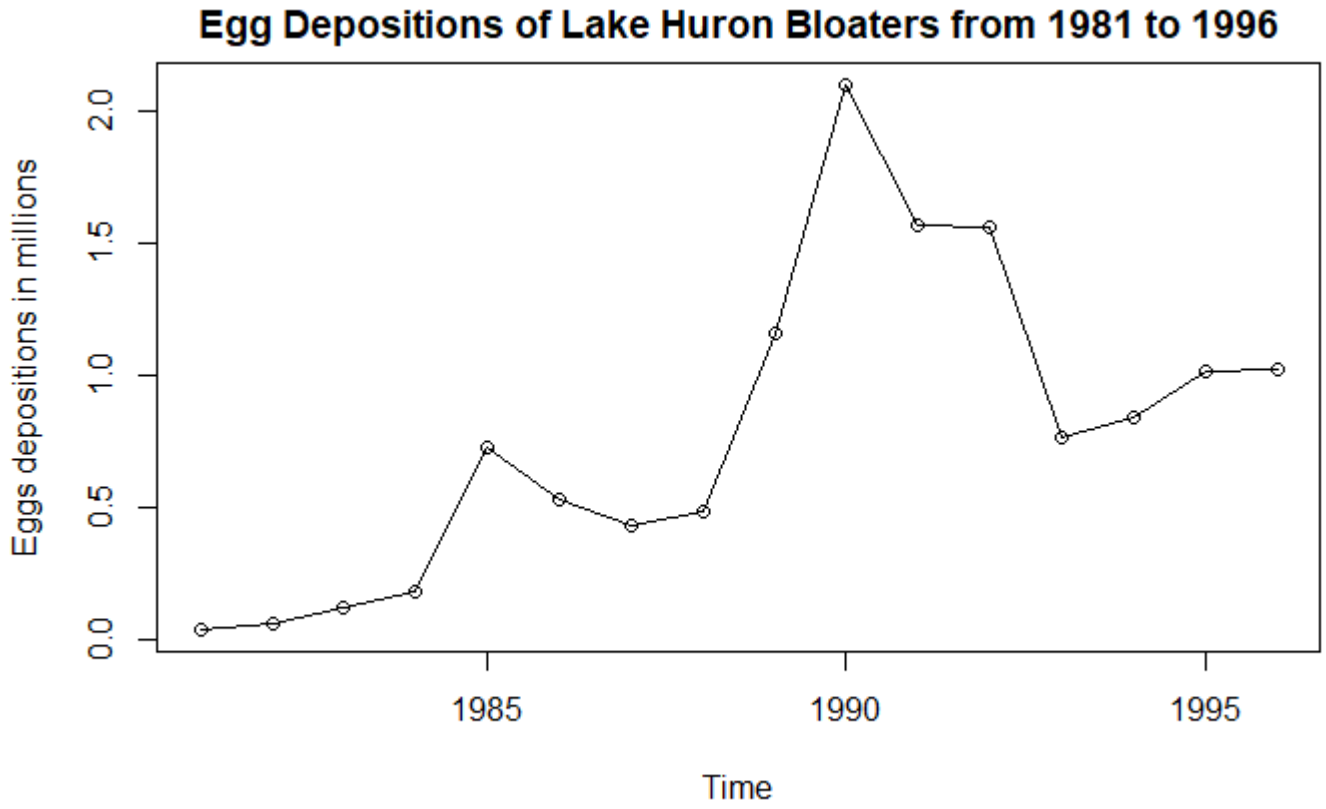
Hide

```
Bloater_egg <- ts(as.vector(Bloater_egg$eggs), start=1981, end=1996)
class(Bloater_egg)
```

```
[1] "ts"
```

Hide

```
plot(Bloater_egg,type='o',ylab='Eggs depositions in millions',main = "Egg Depositions of Lake Hu
ron Bloaters from 1981 to 1996")
```




The above time series plot represents an overall upward (positive) trend , i.e., the egg depositions of age-3 Lake Huron Bloaters has been increased over the years from 1981 to 1996. Although, the number of observations are only 16 and 2 peaks can be noticed, where the former is lower than the latter, hence, changing variance is present. From the succeeding observations, it can be interpreted that the series follows an auto-regressive behaviour. The egg depositions is at its peak, i.e., the highest in the year 1990, after which there is a decline in the trend, that is again followed by an upward trend after the year 1993.

# SELECTION OF BEST FITTING TREND MODEL-

Before further analysis, it is important to find the best fitted model for the given data.

# 1. Linear model

Hide

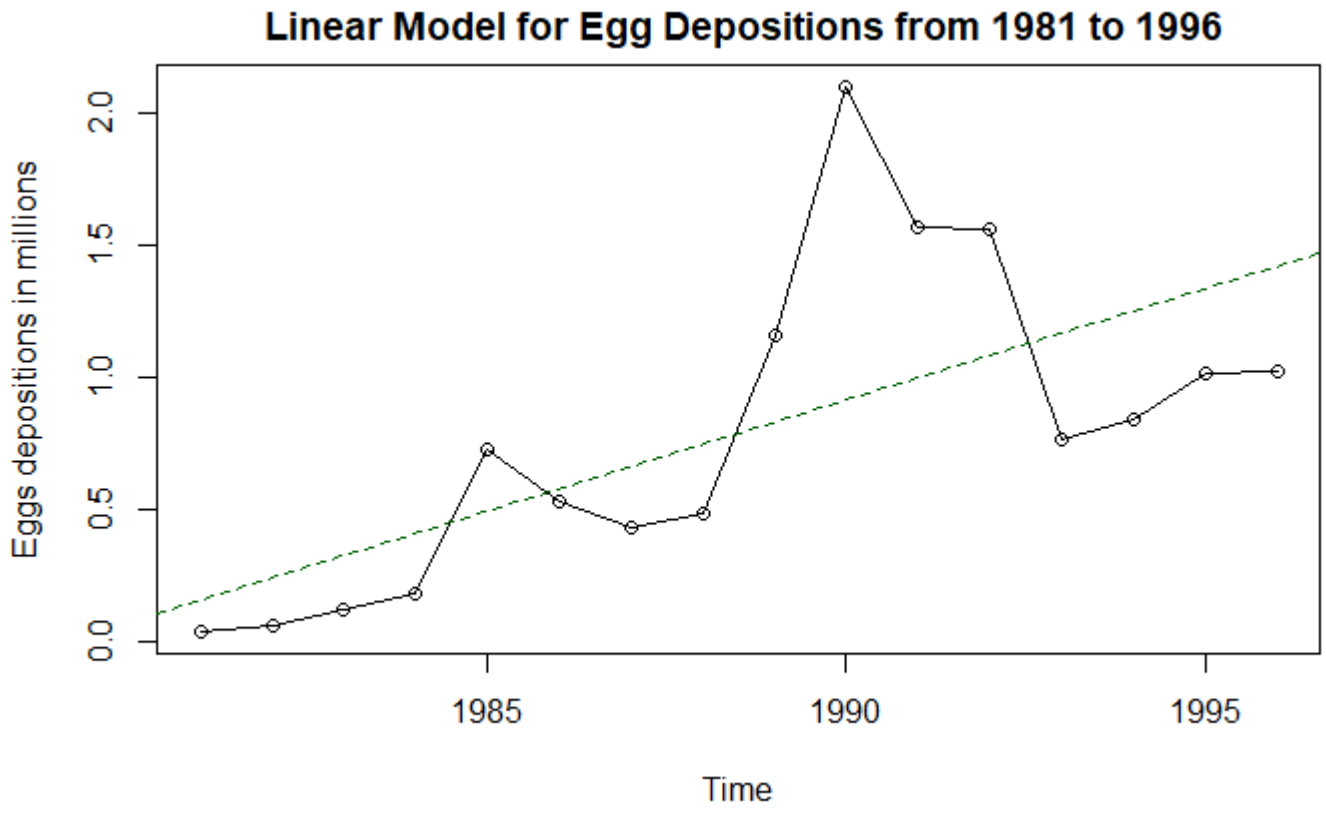
```
Linearmodel <- lm(Bloater_egg~time(Bloater_egg)) # label the linear trend model as linear_model
summary(Linearmodel)
```

```
Call:
lm(formula = Bloater_egg ~ time(Bloater_egg))

Residuals:
    Min      1Q  Median      3Q     Max
-0.4048 -0.2768 -0.1933  0.2536  1.1857

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -165.98275   49.58836  -3.347  0.00479 **
time(Bloater_egg)    0.08387    0.02494   3.363  0.00464 **
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1

Residual standard error: 0.4598 on 14 degrees of freedom
Multiple R-squared:  0.4469,    Adjusted R-squared:  0.4074
F-statistic: 11.31 on 1 and 14 DF,  p-value: 0.004642
```

*SUPERIMPOSING LINE OF BEST FIT FOR LINEAR MODEL-

Hide

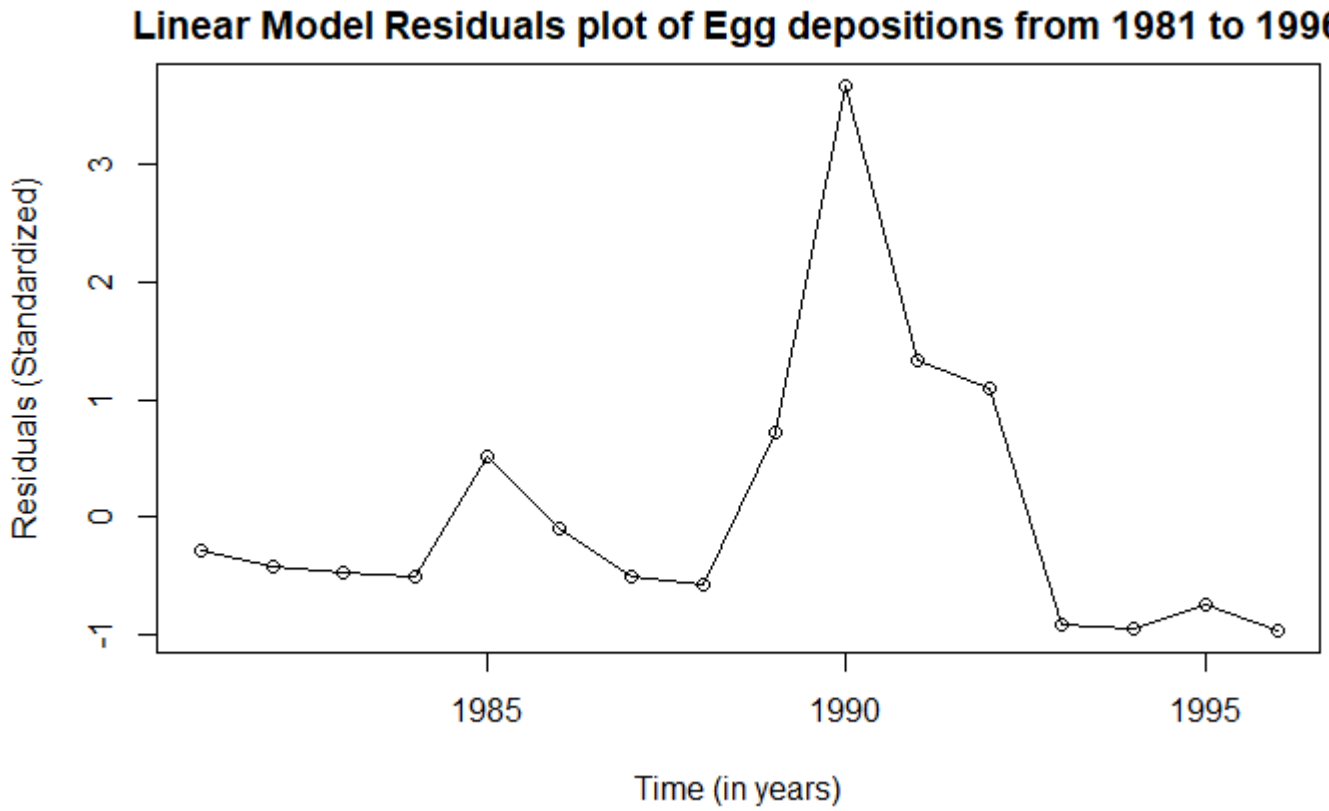
```
plot(Bloater_egg,type='o',ylab='Eggs depositions in millions',main = " Linear Model for Egg Depo
sitions from 1981 to 1996")
abline(Linearmodel, lty=2, col="dark green")
```




*LINEAR MODEL RESIDUALS

Hide

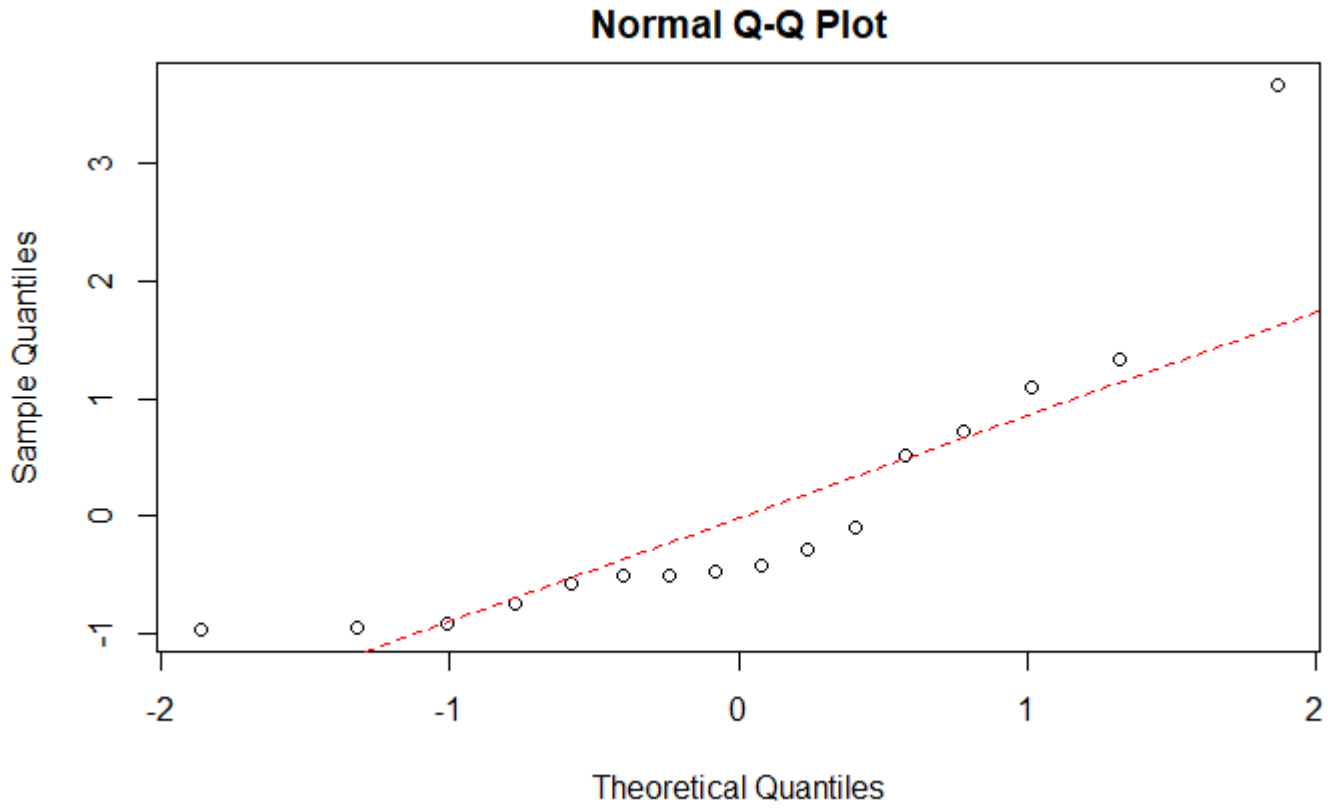
```
residual_linear = rstudent(Linearmodel)
plot(y = residual_linear, x = as.vector(time(Bloater_egg)),xlab = 'Time (in years)', ylab='Resid
uals (Standardized)',type='o', main = "Linear Model Residuals plot of Egg depositions from 1981
 to 1996")
```




*NORMALITY CHECK-LINEAR MODEL RESIDUALS (QQ PLOT)-

Hide

```
qqnorm(residual_linear)
qqline(residual_linear, col = 2, lwd = 1, lty = 2)
```




*NORMALITY CHECK-LINEAR MODEL RESIDUALS (SHAPIRO-WILK TEST)-

Hide

```
shapiro.test(residual_linear)
```

```
    Shapiro-Wilk normality test

data:  residual_linear
W = 0.7726, p-value = 0.001205
```

# 2. Quadratic Model

Hide

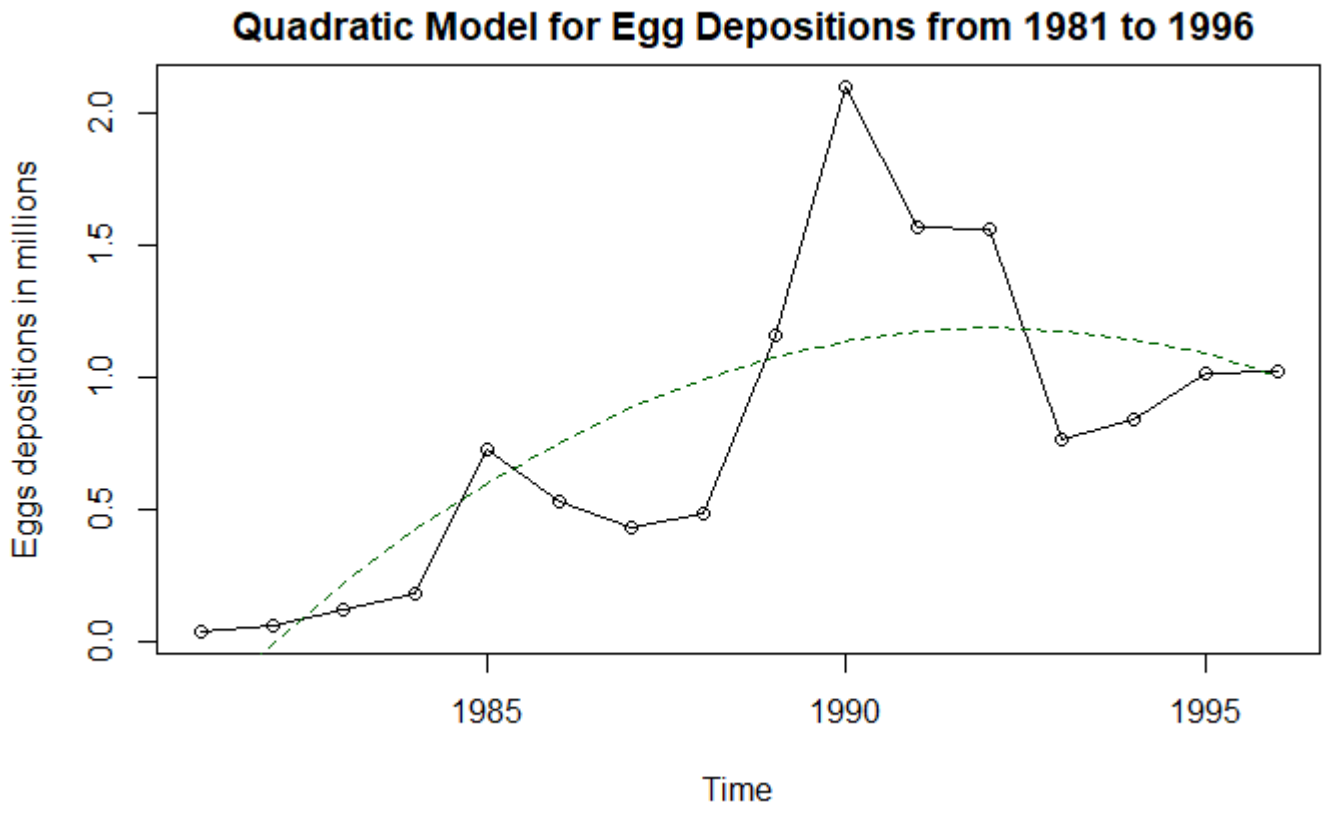
```
t = time(Bloater_egg)
t2 = t^2
QuadraticModel = lm(Bloater_egg~t + t2)
summary(QuadraticModel)
```

```
Call:
lm(formula = Bloater_egg ~ t + t2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.50896 -0.25523 -0.02701  0.16615  0.96322

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.647e+04  2.141e+04  -2.170   0.0491 *
t            4.665e+01  2.153e+01   2.166   0.0494 *
t2          -1.171e-02  5.415e-03  -2.163   0.0498 *
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1

Residual standard error: 0.4092 on 13 degrees of freedom
Multiple R-squared:  0.5932,    Adjusted R-squared:  0.5306
F-statistic: 9.479 on 2 and 13 DF,  p-value: 0.00289
```

*SUPERIMPOSING LINE OF BEST FIT FOR QUADRATIC MODEL-

Hide

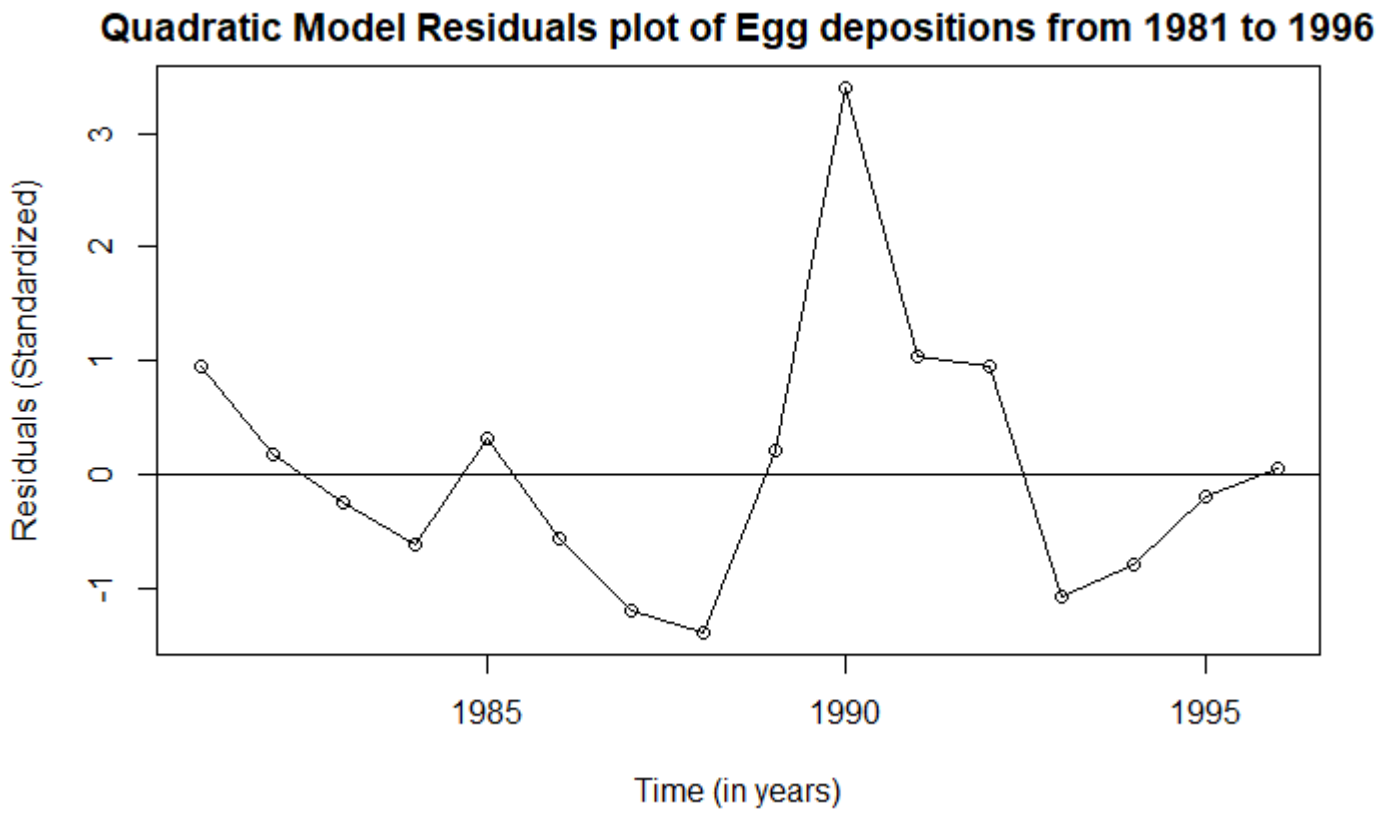
```
plot(Bloater_egg,type='o',ylab='Eggs depositions in millions',main = " Quadratic Model for Egg D
epositions from 1981 to 1996")
points(t,predict.lm(QuadraticModel), type="l", lty=2,col="dark green")
```




*QUADRATIC MODEL RESIDUALS-

Hide

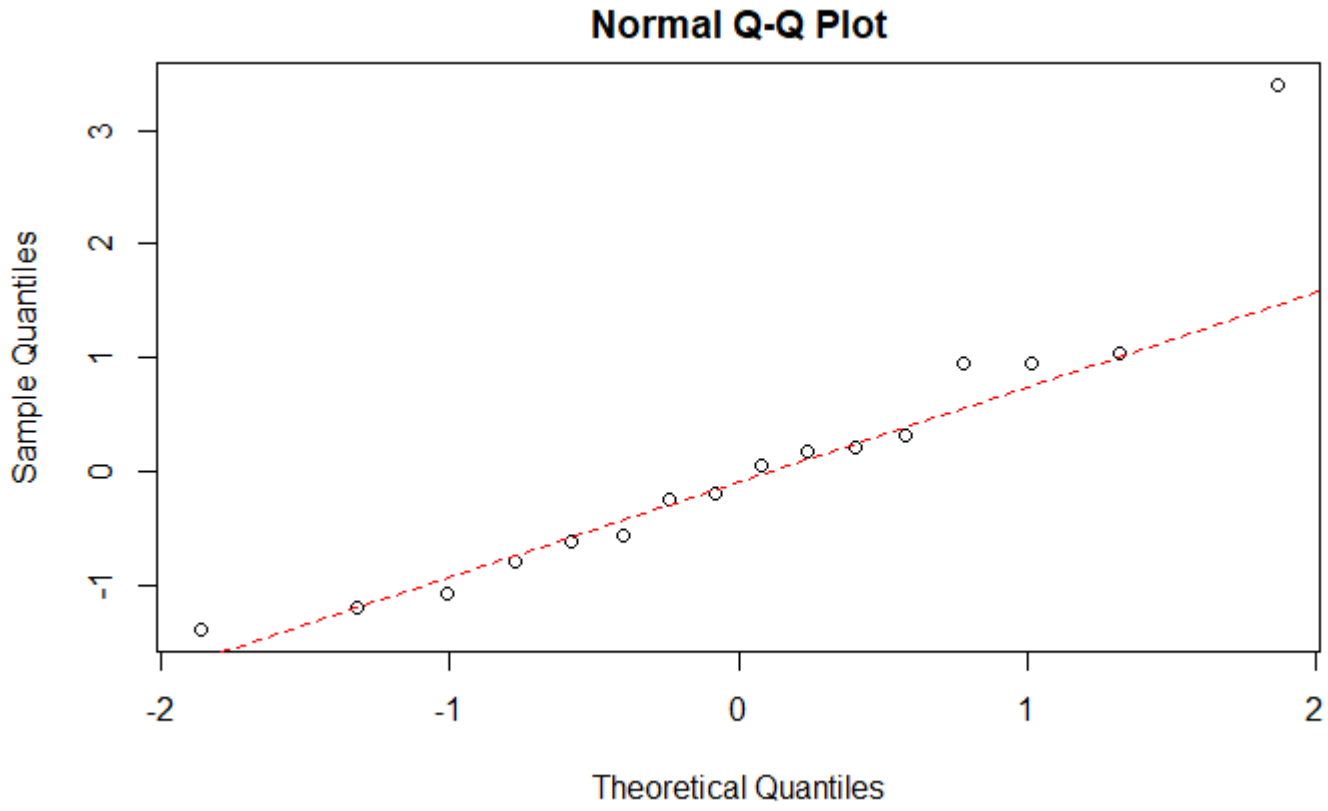
```
residual_quadratic = rstudent(QuadraticModel)
plot(y = residual_quadratic, x = as.vector(time(Bloater_egg)),xlab = 'Time (in years)', ylab='Residuals (Standardized)',type='o', main = "Quadratic Model Residuals plot of Egg depositions from 1981 to 1996")
abline(h=0)
```




*NORMALITY CHECK-QUADRATIC MODEL RESIDUALS (QQ PLOT)-

Hide

```
qqnorm(residual_quadratic)
qqline(residual_quadratic, col = 2, lwd = 1, lty = 2)
```




*NORMALITY CHECK-QUADRATIC MODEL RESIDUALS (SHAPIRO-WILK TEST)-

Hide

```
shapiro.test(residual_quadratic)
```

```
    Shapiro-Wilk normality test

data:  residual_quadratic
W = 0.87948, p-value = 0.03809
```

Based on all the above analysis for the linear and quadratic models, the following can be interpreted- 1. With the R-squared value, it can be said that around 53% of the variation in the data can be explained by the “QUADRATIC TREND”, whereas only around 40% of the variation is captured bythe “LINEAR TREND”. 2. The QQ-PLOT for “QUADRATIC MODEL” had lesser deviation from the normality as compared to “LINEAR MODEL”. 3. Although, from the shapiro-wilk test, the p-value for both the Quadratic and Linear models are less than 0.05, hence the null hypothesis can be rejected indicating the data is not normally distributed. 4.With no seasonality present, harmonic model cannot be used.

Overall, it can be said from the results that, QUADRATIC MODEL is the best fit trend model for the given time series.

# DATA PREPARATION-

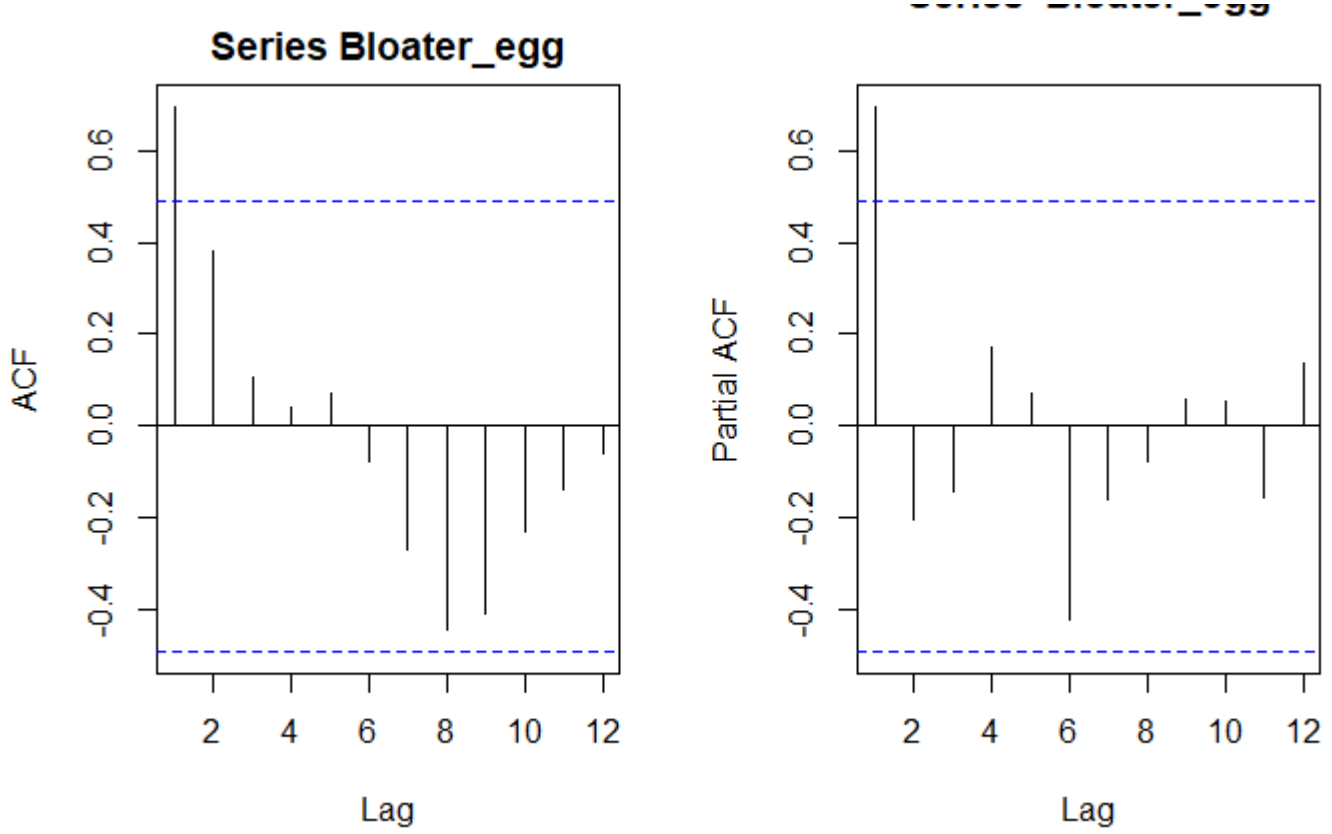
ACF and PACF have been performed in order to check the TREND and STATIONARITY-

Hide

```
par(mfrow=c(1,2))
acf(Bloater_egg)
pacf(Bloater_egg)
```

Hide

```
par(mfrow=c(1,1))
```




In ACF, there is a slowly decaying pattern and in PACF, the first correlation is very high, that implies the presence of non-stationarity and trend. And hence, implying the need for transformation and differencing in order to make the it stationary.

# Transformation-

It is always better to first make the data stationary while preparing the data. Although, the ACF and PACF have already shown that the data is non-stationary, but 'adf.test()' will confirm that result. Also, to check the normality, shapiro.test() function is used.

Hide

```
adf.test(Bloater_egg)
```

```
    Augmented Dickey-Fuller Test

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

Hide

```
shapiro.test(Bloater_egg)
```

```
    Shapiro-Wilk normality test

data:  Bloater_egg
W = 0.94201, p-value = 0.3744
```

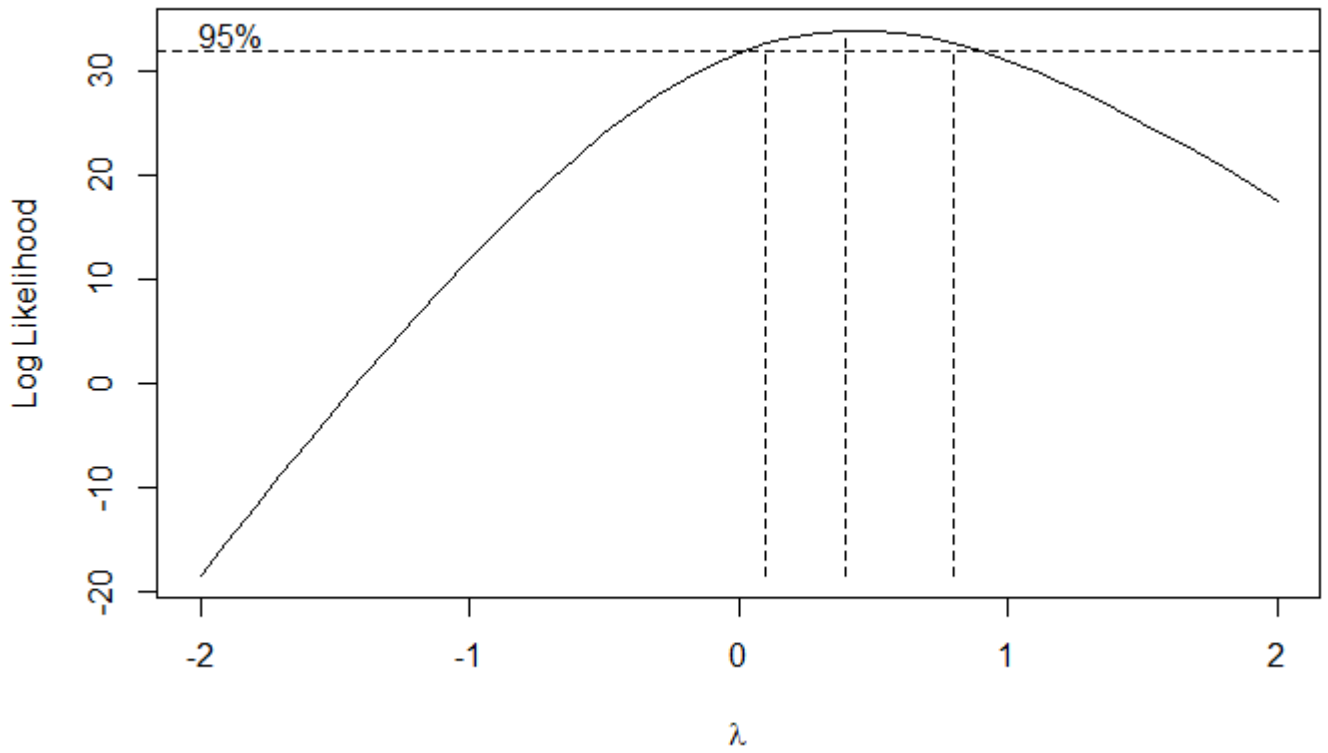
From the ‘Dickey-Fuller Test’, it can be observed that the p-value of 0.5469 is greater than alpha=0.05, hence, the null hypothesis cannot be rejected, proving that the data is non-stationary. Similar is the case with the ‘Shapiro-Wilk normality test’, the p-value of 0.3744 is greater than alpha, therefore the null hypothesis, that the data is normally distributed, cannot be rejected, proving the normality of the data.

*BOX-COX TRANSFORMATION-

Hide

```
Bloater_egg_T <- BoxCox.ar(Bloater_egg,method = "yule-walker")
```

```
possible convergence problem: optim gave code = 1possible convergence problem: optim gave code =
1
```




Hide

```
Bloater_egg_T$ci
```

```
[1] 0.1 0.8
```

The values of lambda contained by the 95% confidence interval, lies between 0.1 and 0.8. The mid-point/center of this confidence interval comes out to be 0.45 approximately. This mid-point value of lambda will be used for the box-cox transformation.

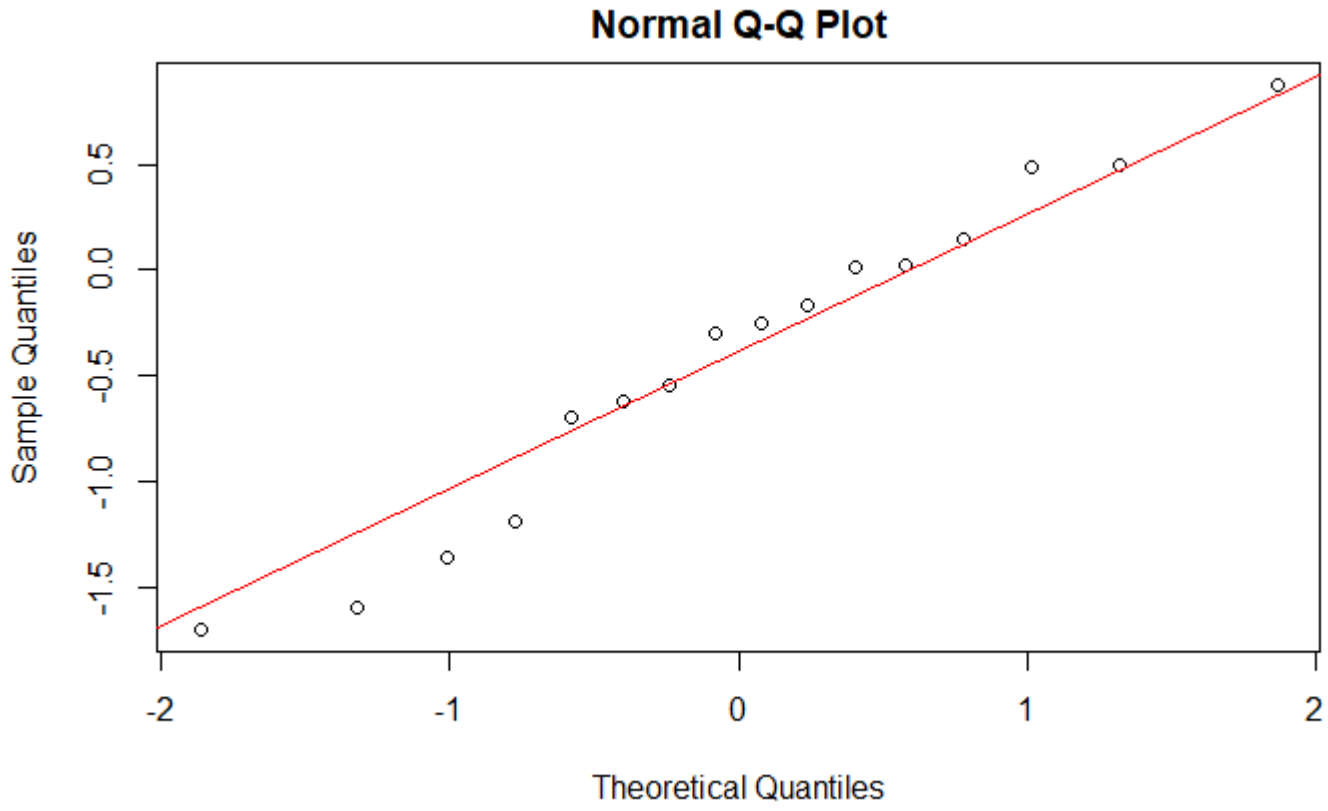
Hide

```
lambda=0.45
Bloater_egg_BC <- (Bloater_egg^lambda-1)/lambda
```

*Normality check after applying the Box-Cox transformation-

Hide

```
qqnorm(Bloater_egg_BC)
qqline(Bloater_egg_BC, col = 2)
```




Hide

```
shapiro.test(Bloater_egg_BC)
```

```
    Shapiro-Wilk normality test

data:  Bloater_egg_BC
W = 0.96269, p-value = 0.7107
```

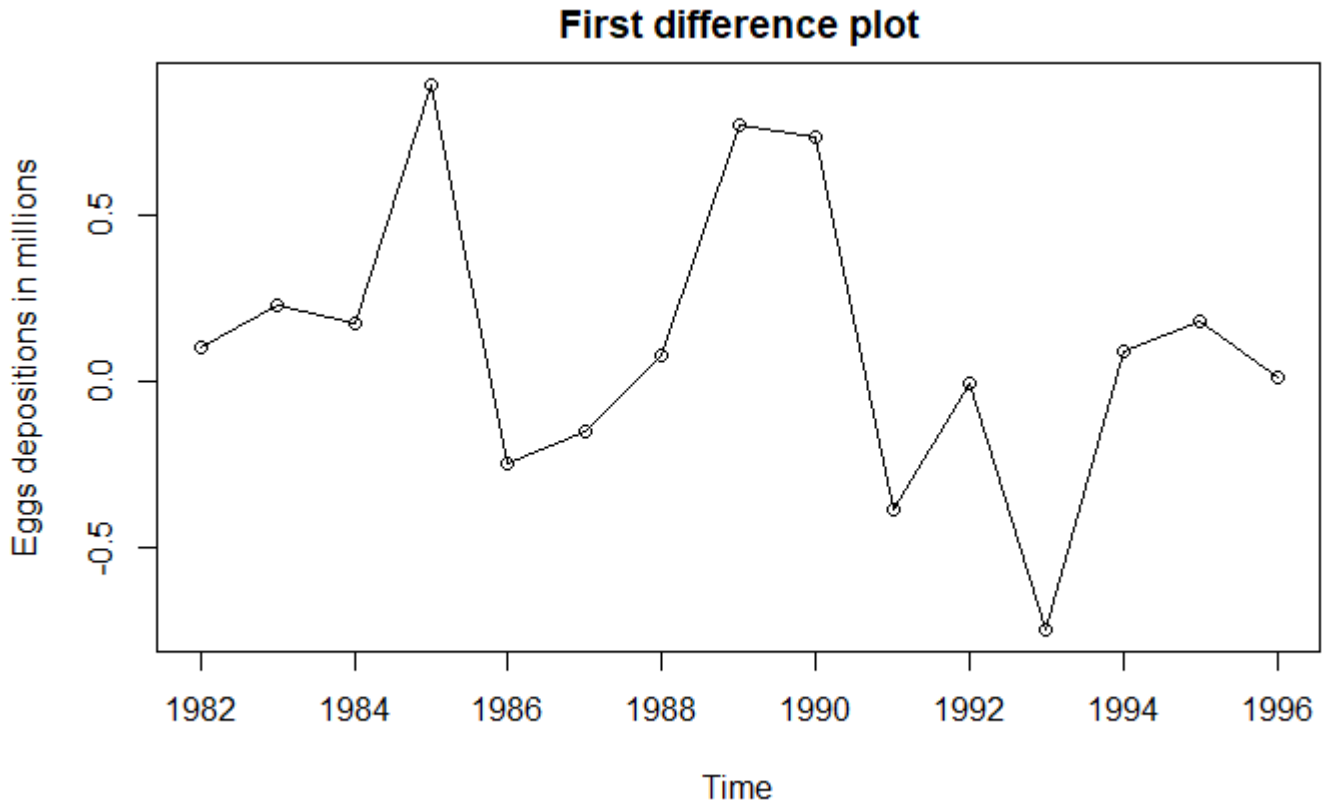
It can be noticed that the Box_Cox transformation has improved the normality of the series as the deviation of the dot-points from the red-line has decreased. Additionally, the p-value from the Shapiro-wilk test has also increased to 0.71 from 0.37, hence confirming that the data is normally distributed.

# DIFFERENCING

*CALCULATING THE FIRST DIFFERENCE

Hide

```
Bloater_egg_BC_diff = diff(Bloater_egg_BC)
plot(Bloater_egg_BC_diff,type='o',ylab='Eggs depositions in millions',main = "First difference p
lot")
```




A trend can still be seen from the above plot. The stationarity needs to be checked.

Hide

```
adf.test(Bloater_egg_BC_diff)
```

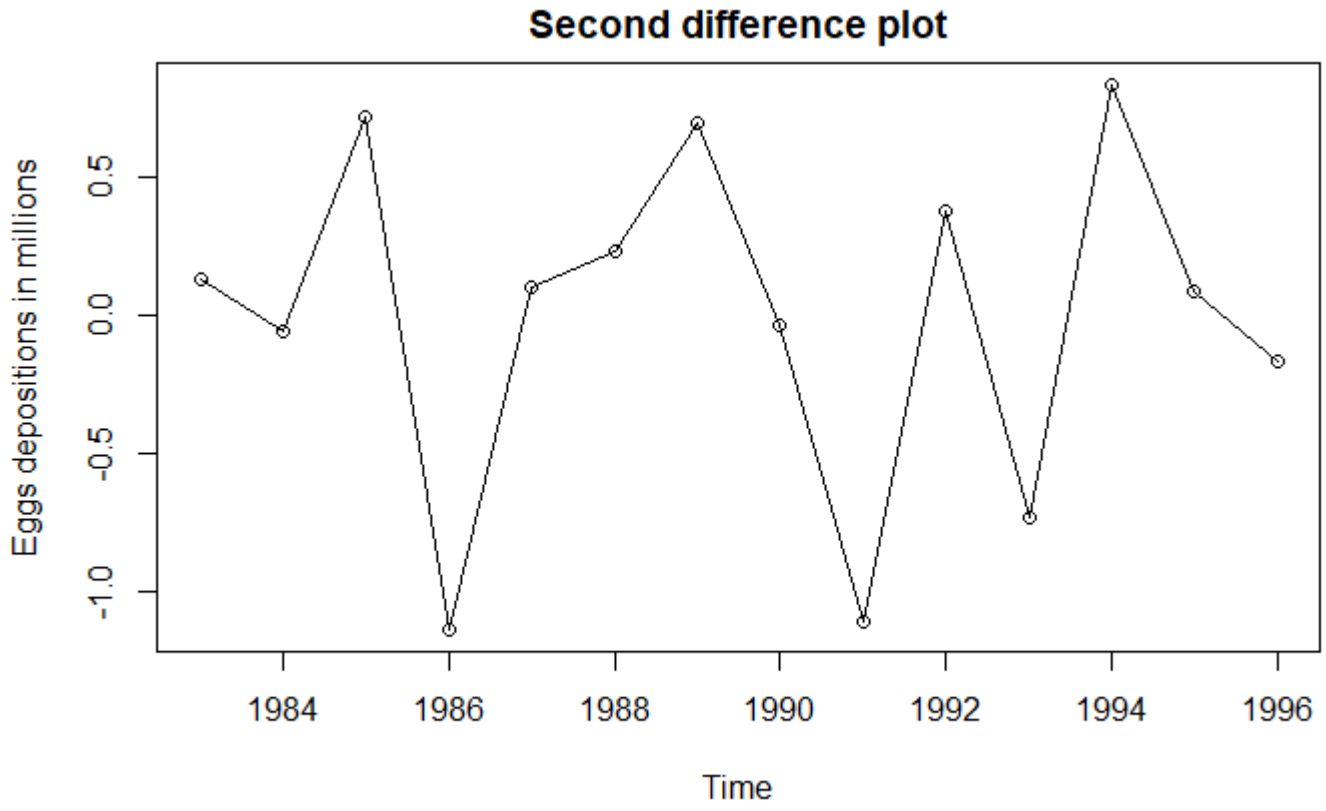
```
    Augmented Dickey-Fuller Test

data:  Bloater_egg_BC_diff
Dickey-Fuller = -3.6798, Lag order = 2, p-value = 0.0443
alternative hypothesis: stationary
```

The p-value of 0.04 is less than alpha=0.05, hence the null hypothesis of non-stationarity can be rejected, i.e., the series has become stationary after the first differencing.

*CALCULATING THE SECOND DIFFERENCE

Hide

```
Bloater_egg_BC_diff2 = diff(Bloater_egg_BC, differences= 2)
plot(Bloater_egg_BC_diff2,type='o',ylab='Eggs depositions in millions',main = "Second difference
plot")
```




Now, the trend can no longer be seen in the baove series, hence the series canbe considered as stationary. Checking it with adf.test.

Hide

```
adf.test(Bloater_egg_BC_diff2)
```

```
    Augmented Dickey-Fuller Test

data:  Bloater_egg_BC_diff2
Dickey-Fuller = -3.1733, Lag order = 2, p-value = 0.1254
alternative hypothesis: stationary
```

The p-value here has increased after the second differencing, this is generally not the case. The trend has been disappeared here, though the p-value is no longer small enough, but considering the trend, second differencing will be used.

# Model specification using ACf, PACF, EACF and BIC over the differenced series

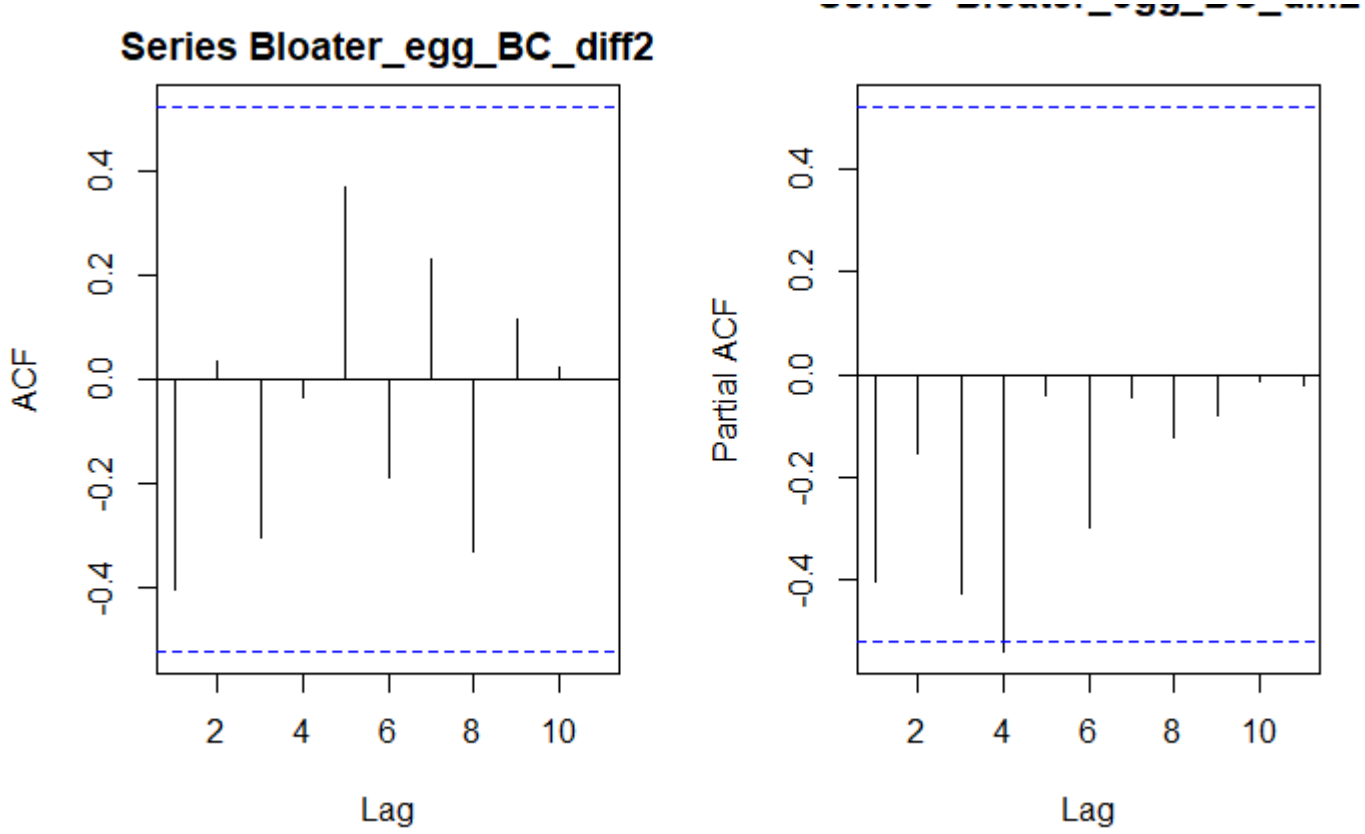
*ACF and PACF FOR THE DIFFERENCED SERIES

Hide

```
par(mfrow=c(1,2))
acf(Bloater_egg_BC_diff2)
pacf(Bloater_egg_BC_diff2)
```

Hide

```
par(mfrow=c(1,1))
```




No significant lags are present in the above ACF plot, although there is one significant lag present in the PACF plot. So, ARIMA(1,2,0) model can be considered here.

*EACF (EXTENDED ACF)

To know the order of AR (auto-regressive) and MA (moving average) components of an ARMA model, EACF can also be used.

Hide

```
eacf(Bloater_egg_BC_diff2, 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
```

With the upper-left point of the above Extended autocorrelation (EACF) method as (0,0), confirming the existence of a white noise behaviour. By taking into account the neighbouring points, models ARIMA(0,2,1) and ARIMA(1,2,0), ARIMA(1,2,1) can be added to the set of possible models. ARIMA(1,2,0) has already been found from ACF and PACF plots.

*BAYESIAN INFORMATION CRITERION (BIC)

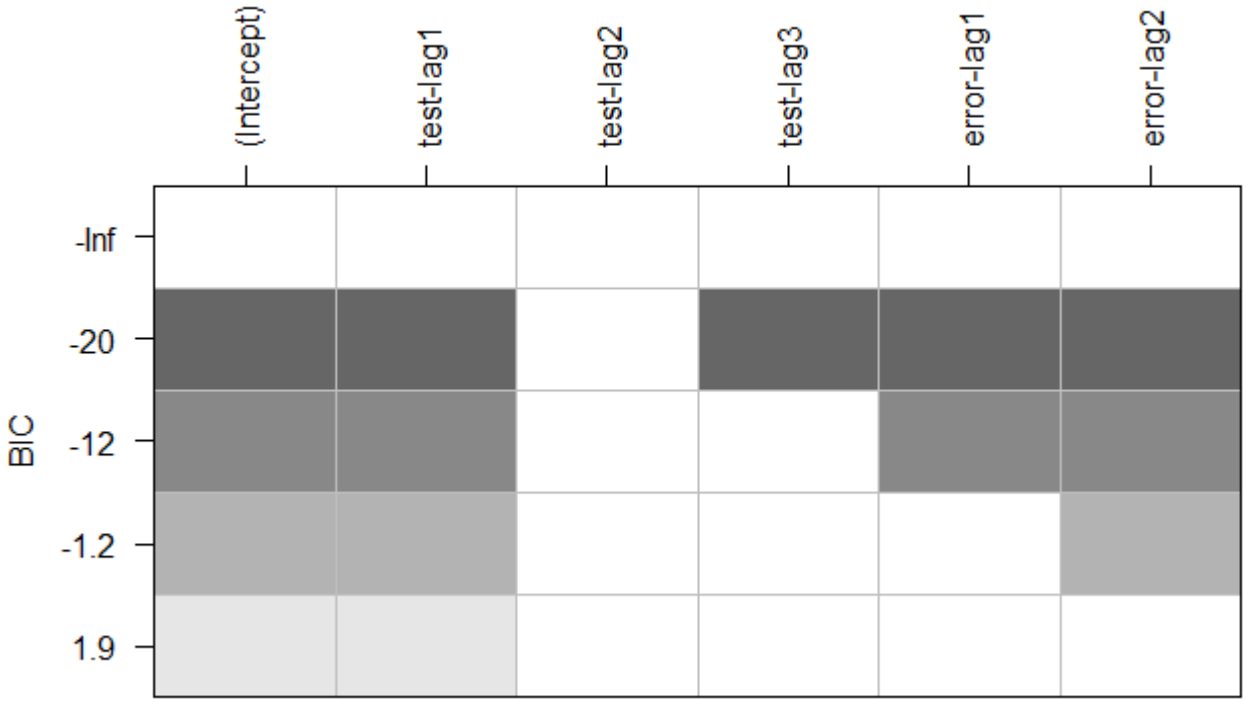
Hide

```
res = armasubsets(y=Bloater_egg_BC_diff2,nar=3,nma=2,y.name='test',ar.method='ols')
```

```
model order:  7 singularities in the computation of the projection matrix results are only valid up to model order 6
```

Hide

```
plot(res)
```




In the above BIC table, the shaded columns are corresponding to AR(1), AR(3) coefficients and there are two MA effects,i.e., MA(1) and MA(2). So, the models from the above output, that can be included in the set of possible models are ARIMA(1,2,1), ARIMA(1,2,2), ARIMA(3,2,1) and ARIMA(3,2,2).

So, the set of all the candidate models are- ARIMA(0,2,1), ARIMA(1,2,0), ARIMA(1,2,1), ARIMA(1,2,2), ARIMA(3,2,1) and ARIMA(3,2,2).

# PARAMETER ESTIMATION (Model Testing)

After it is ensured that the series is staionary, and the specifications of orders of the AR and MA elements for ARMA model have been calculated, the next step is the estimation of parameters of the above specified tentative models. For this least squares estimation and maximum likelihood estimation will be applied. At last, the selelction of the best model will be established from AIC and BIC.

- ARIMA(0,2,1)

Hide

```
model_021_css = arima(Bloater_egg,order=c(0,2,1),method='CSS')
coeftest(model_021_css)
```

```
z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ma1 -1.066739   0.071847 -14.847 < 2.2e-16 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_021_ml = arima(Bloater_egg,order=c(0,2,1),method='ML')
coeftest(model_021_ml)
```

```
z test of coefficients:

    Estimate Std. Error z value  Pr(>|z|)
ma1 -1.00000    0.25823 -3.8725 0.0001077 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

MA(1) coefficient is significant for both CSS and ML estimations.

- ARIMA(1,2,0)

Hide

```
model_120_css = arima(Bloater_egg,order=c(1,2,0),method='CSS')
coeftest(model_120_css)
```

```
z test of coefficients:

    Estimate Std. Error z value Pr(>|z|)
ar1 -0.45944    0.23810 -1.9296  0.05365 .
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_120_ml = arima(Bloater_egg,order=c(1,2,0),method='ML')
coeftest(model_120_ml)
```

```
z test of coefficients:

    Estimate Std. Error z value Pr(>|z|)
ar1 -0.42966    0.22743 -1.8892  0.05886 .
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

AR(1) coefficient is insignificant for both CSS and ML estimations

- ARIMA(1,2,1)

Hide

```
model_121_css = arima(Bloater_egg,order=c(1,2,1),method='CSS')
coeftest(model_121_css)
```

```
z test of coefficients:

     Estimate Std. Error  z value Pr(>|z|)
ar1  0.073817   0.284315   0.2596   0.7951
ma1 -1.132556   0.074796 -15.1419   <2e-16 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_121_ml = arima(Bloater_egg,order=c(1,2,1),method='ML')
coeftest(model_121_ml)
```

```
z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1  0.071764   0.269251  0.2665    0.7898
ma1 -0.999999   0.236872 -4.2217 2.425e-05 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

AR(1) is insignificant in both CSS and ML estimations, whereas MA(1) is significant for both.

- ARIMA(1,2,2)

Hide

```
model_122_css = arima(Bloater_egg,order=c(1,2,2),method='CSS')
coeftest(model_122_css)
```

```
z test of coefficients:

     Estimate Std. Error z value  Pr(>|z|)
ar1  1.005671   0.049778  20.203 < 2.2e-16 ***
ma1 -2.824099   0.125344 -22.531 < 2.2e-16 ***
ma2  1.838559   0.114620  16.040 < 2.2e-16 ***
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_122_ml = arima(Bloater_egg,order=c(1,2,2),method='ML')
coeftest(model_122_ml)
```

```
z test of coefficients:

     Estimate Std. Error z value Pr(>|z|)
ar1  0.058932   1.166803  0.0505   0.9597
ma1 -0.987060   1.155631 -0.8541   0.3930
ma2 -0.012925   1.130998 -0.0114   0.9909
```

AR(1) and both the coefficients of MA are significant for CSS, whereas they all are insignificant for ML.

- ARIMA(3,2,1)

Hide

```
model_321_css = arima(Bloater_egg,order=c(3,2,1),method='CSS')
coeftest(model_321_css)
```

```
z test of coefficients:

    Estimate Std. Error z value Pr(>|z|)
ar1 -0.17209    0.26139 -0.6584 0.510286
ar2 -0.19198    0.25364 -0.7569 0.449106
ar3 -0.52748    0.24922 -2.1165 0.034300 *
ma1 -0.64906    0.24639 -2.6343 0.008432 **
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_321_ml = arima(Bloater_egg,order=c(3,2,1),method='ML')
coeftest(model_321_ml)
```

```
z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  0.0042103  0.3147996  0.0134   0.9893
ar2 -0.0431425  0.2891200 -0.1492   0.8814
ar3 -0.3350403  0.2798031 -1.1974   0.2311
ma1 -0.9018704  0.6489515 -1.3897   0.1646
```

For CSS, only AR(3) and MA(1) are significant.

- ARIMA(3,2,2)

Hide

```
model_322_css = arima(Bloater_egg,order=c(3,2,2),method='CSS')
coeftest(model_322_css)
```

```
z test of coefficients:

     Estimate Std. Error z value Pr(>|z|)
ar1 -0.139922   0.348205 -0.4018  0.68780
ar2 -0.198602   0.258520 -0.7682  0.44235
ar3 -0.544728   0.277057 -1.9661  0.04928 *
ma1 -0.683391   0.360419 -1.8961  0.05795 .
ma2  0.045948   0.329455  0.1395  0.88908
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_322_ml = arima(Bloater_egg,order=c(3,2,2),method='ML')
coeftest(model_322_ml)
```

```
z test of coefficients:

     Estimate Std. Error z value Pr(>|z|)
ar1  0.132356   0.472334  0.2802  0.77931
ar2 -0.099038   0.323387 -0.3063  0.75941
ar3 -0.384399   0.314004 -1.2242  0.22088
ma1 -0.996970   0.589359 -1.6916  0.09072 .
ma2  0.199121   0.472894  0.4211  0.67371
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Only AR(3) is significant from CSS, whereas all the coefficients of AR and MA are insignificant for ML.

So, it can be now concluded that, the model with all the significant coefficients is ARIMA(1,2,2) with CSS method.

# SORTING THE MODEL WITH AIC AND BIC

Creating 'sort.score' function.

Hide

```
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")')
  }
}
```

This function wil sort all the tentative models by their respective AIC and BIC, in order to get the best model among all the possible models.

Hide

```
sort.score(AIC(model_021_ml,model_120_ml,model_121_ml,model_122_ml,model_321_ml,model_322_ml), s
core = "aic")
```

| | df<br><dbl> | AIC<br><dbl> |
|---|---|---|
| model_021_ml | 2 | 22.74602 |
| model_121_ml | 3 | 24.67428 |
| model_120_ml | 2 | 26.57611 |
| model_122_ml | 4 | 26.67412 |
| model_321_ml | 5 | 26.90919 |
| model_322_ml | 6 | 28.75165 |

6 rows

Hide

```
sort.score(BIC(model_021_ml,model_120_ml,model_121_ml,model_122_ml,model_321_ml,model_322_ml), s
core = "bic" )
```

| | df<br><dbl> | BIC<br><dbl> |
|---|---|---|
| model_021_ml | 2 | 24.02413 |
| model_121_ml | 3 | 26.59145 |
| model_120_ml | 2 | 27.85423 |

| | df<br><dbl> | BIC<br><dbl> |
|---|---|---|
| model_122_ml | 4 | 29.23035 |
| model_321_ml | 5 | 30.10448 |
| model_322_ml | 6 | 32.58599 |

6 rows

With the results of both AIC and BIC, it can be infered that ARIMA(0,2,1) is the best model among all the tentative models.

# Overfitting

ARIMA(1,2,1) model overfits the best model ARIMA(0,1,1) and also, from the previous results AR(1) has been noticed as insignificant from both CSS and ML estimation.

Another overfitting model can be ARIMA(0,2,2). It is now being fitted, in order to check it's overfitting.

- ARIMA(0,2,2)

Hide

```
model_022_css = arima(Bloater_egg,order=c(0,2,2),method='CSS')
coeftest(model_022_css)
```

```

z test of coefficients:

    Estimate Std. Error z value  Pr(>|z|)
ma1 -1.01988    0.26306 -3.8769 0.0001058 ***
ma2 -0.05353    0.29008 -0.1845 0.8535938
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hide

```
model_022_ml = arima(Bloater_egg,order=c(0,2,2),method='ML')
coeftest(model_022_ml)
```

```

z test of coefficients:

     Estimate Std. Error z value Pr(>|z|)
ma1 -0.932377   0.345781 -2.6964 0.007009 **
ma2 -0.067623   0.250360 -0.2701 0.787081
---
Signif. codes:  0 □***□ 0.001 □**□ 0.01 □*□ 0.05 □.□ 0.1 □ □ 1
```

Hence, MA(2) is insignificant for both the CSS nd ML.

# SELECTING THE BEST MODEL

Hide

```
sort.score(AIC(model_021_ml,model_120_ml,model_121_ml,model_122_ml,model_321_ml,model_322_ml,model_022_ml), score = "aic")
```

| | df<br><dbl> | AIC<br><dbl> |
|---|---|---|
| model_021_ml | 2 | 22.74602 |
| model_121_ml | 3 | 24.67428 |
| model_022_ml | 3 | 24.67638 |
| model_120_ml | 2 | 26.57611 |
| model_122_ml | 4 | 26.67412 |
| model_321_ml | 5 | 26.90919 |
| model_322_ml | 6 | 28.75165 |

7 rows

Hide

```
sort.score(BIC(model_021_ml,model_120_ml,model_121_ml,model_122_ml,model_321_ml,model_322_ml,model_022_ml), score = "bic" )
```

| | df<br><dbl> | BIC<br><dbl> |
|---|---|---|
| model_021_ml | 2 | 24.02413 |
| model_121_ml | 3 | 26.59145 |
| model_022_ml | 3 | 26.59355 |
| model_120_ml | 2 | 27.85423 |
| model_122_ml | 4 | 29.23035 |
| model_321_ml | 5 | 30.10448 |
| model_322_ml | 6 | 32.58599 |

7 rows

After fitting the overfitting model, it can still be noticed that ARIMA(0,2,1) is the best model by both AIC and BIc results. Hence, it is the most suitable model to forecast about egg depositions.

# MODEL DIAGNOSTICS

For model diagnostics, the best model, i.e., ARIMA(0,2,1) is selected first and it is to be checked whether it works fine with the residuals. If it won't, then we will switch to a different model. Standardised residual behaviour is analyzed on the basis of autocorrelation and normality. Ljung-Box test has also been tested. Based on all these outputs, the feasibility of the best model will be verified.

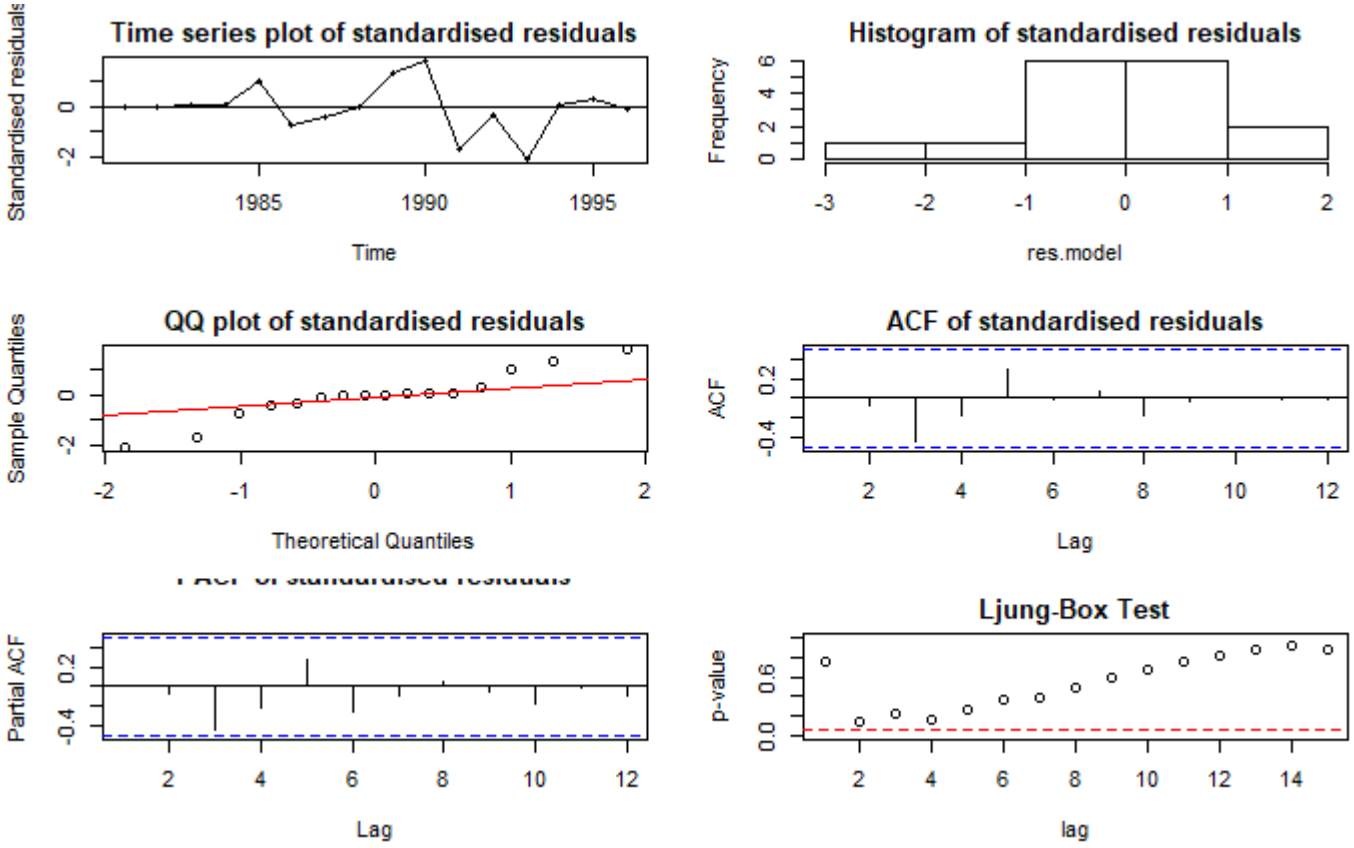
Hide

```
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 standardised
 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")
  pacf(res.model,main="PACF of standardised residuals")
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = length(model$residuals)-1 , StartLag = k + 1, k = 0, SquaredQ = F
ALSE)
  par(mfrow=c(1,1))
}
```

Hide

```
residual.analysis(model = model_021_ml)
```

```

    Shapiro-Wilk normality test

data:  res.model
W = 0.92478, p-value = 0.2013
```




- The very first plot is representing the standardised residuals. No trend, no change in variance are present in the plot, supporting ARIMA(0,2,1).
- The second plot of histogram for standardised residuals is displaying a similar pattern as of the normal distribution, hence proving the presence of normality.
- The third QQ plot, it can be observed that maximum number of data points are aligned with the line of normality. Although, at both the end tails a deviation from the normality can be observed, reason being lesser number of observations in the dataset. Moreover, the p-value of 0.20 from the shapiro-wilk test indicating that the null hypothesis cannot be rejected, confirming the existence of normality.
- From the 4th and 5th plot of ACF and PACF, showing no significant lags,hence, indicating the presence of white noise and supporting the model ARIMA(0,2,1).
- The last Ljung-box test represents that all the data points are above the dashed red line at 5%, confirming that the model ARIMA(0,2,1) is feasible.

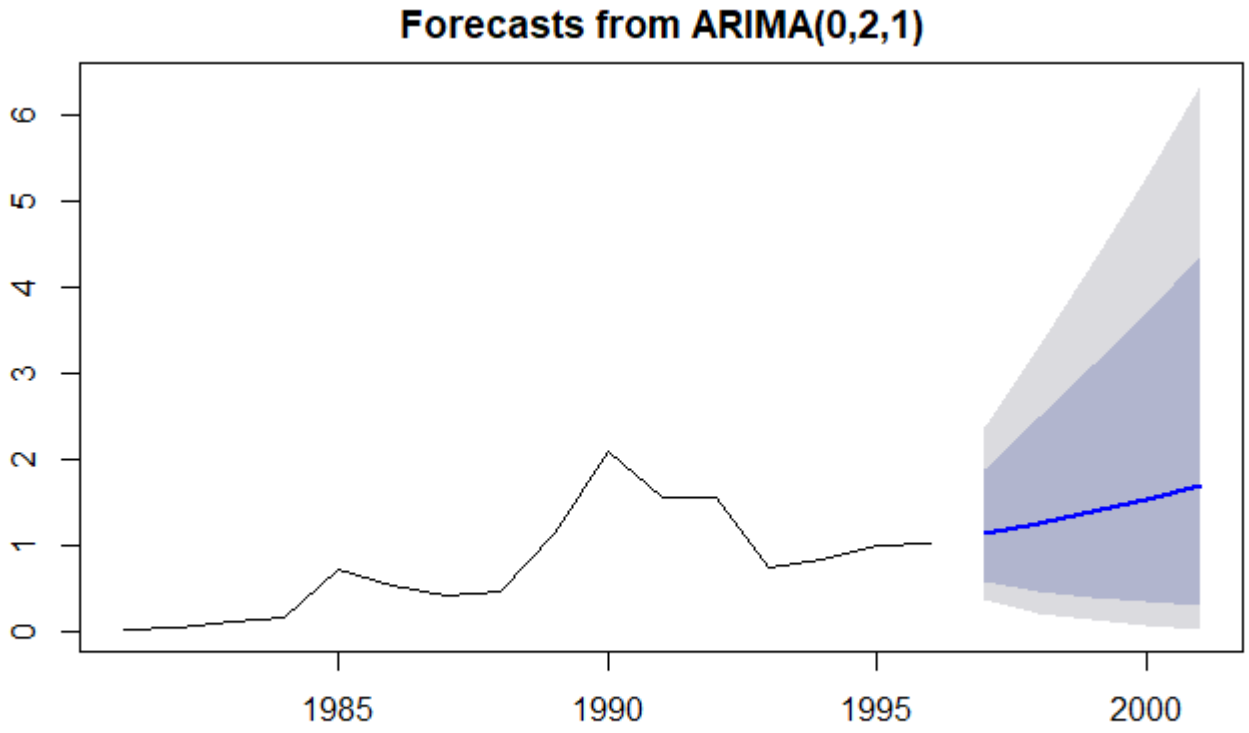
So the conclusion here is, from all the outputs of model diagnostics, it is confirmed that the model ARIMA(0,2,1) works very well with the given data of egg depositions of Lake Huron Bloaters between the years 1981 and 1996.

# FORECASTING FOR THE NEXT 5 YEARS-

This section covers the forecasting of egg depositions, for the next five years after 1996 by fitting the model ARIMA(0,2,1).

Hide

```
forecast_fit = Arima(Bloater_egg,c(0,2,1), lambda = 0.45)
plot(forecast::forecast(forecast_fit,h=5))
```




# SUMMARY

In this report, in order to achieve the final goal of predicting the egg depositions for the next 5 year, firstly, the series is made stationary by using transformation and differencing. Then, the order specifications of the models have been found by using ACF, PACF, EACF and BIC methods. The parameter estimation is performed by least squares estimation and maximum likelihood estimation and the best model is confirmed with the results of AIC and BIC. With the help of model diagnostic, ARIMA(0,2,1) came out to be the most feasible model for forecasting the egg depositions of Lake Huron Bloaters for the next 5 years. The forecast indicates the increase in egg depositions of Bloaters in the Next 5 years after 1996.