Assignment 2

Namita Chhibba

3 May 2018

Introduction

The following Time Series Analysis is concerning Egg depositions (in millions) of age-3 Lake Huron Bloaters (Coregonus hoyi) between years 1981 and 1996. The data will be analyzed using various analysis methods and goal is to choose the best model among a set of possible models for the eggs depositions and give forecasts for the next 5 years.

Loading Packages

```
library(readr)
library(TSA)
```

```
## Loading required package: leaps
```

```
## Loading required package: locfit
```

```
## locfit 1.5-9.1    2013-03-22
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-23. For overview type 'help("mgcv-package")'.
```

```
## Loading required package: tseries
```

```
##
## Attaching package: 'TSA'
```

```
## The following object is masked from 'package:readr':
##
##     spec
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(tseries)
library(fUnitRoots)
```

```
## Loading required package: timeDate
```

```
##
## Attaching package: 'timeDate'
```

```
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
```

```
## Loading required package: timeSeries
```

```
## Loading required package: fBasics
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following object is masked from 'package:timeSeries':
##
##     time<-
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(knitr)
library(forecast)
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:nlme':
##
##     getResponse
```

Importing data

```
##   year   eggs
## 1 1981 0.0402
## 2 1982 0.0602
## 3 1983 0.1205
## 4 1984 0.1807
## 5 1985 0.7229
## 6 1986 0.5321
```

Class of the data.

```
## [1] "data.frame"
```

Since the data is available as a data frame and not as time series object, so we will convert the eggs deposition series available in the dataset to a time series object.

Converting data to time series object

```
eggs.ts <- ts(as.vector(eggs$eggs), start=1981, end= 1996, frequency=1)
head(eggs.ts)
```

```
## Time Series:
## Start = 1981
## End = 1986
## Frequency = 1
## [1] 0.0402 0.0602 0.1205 0.1807 0.7229 0.5321
```

ts() function converts the given observations through time having starting date and end date. Frequency is set to 1 as it is annual data.

Descriptive Analysis

Class of series is:

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

time series. So we will plot it and get insights out of it.

```
plot(eggs.ts, type='o', ylab='egg depositions (in millions)')
```

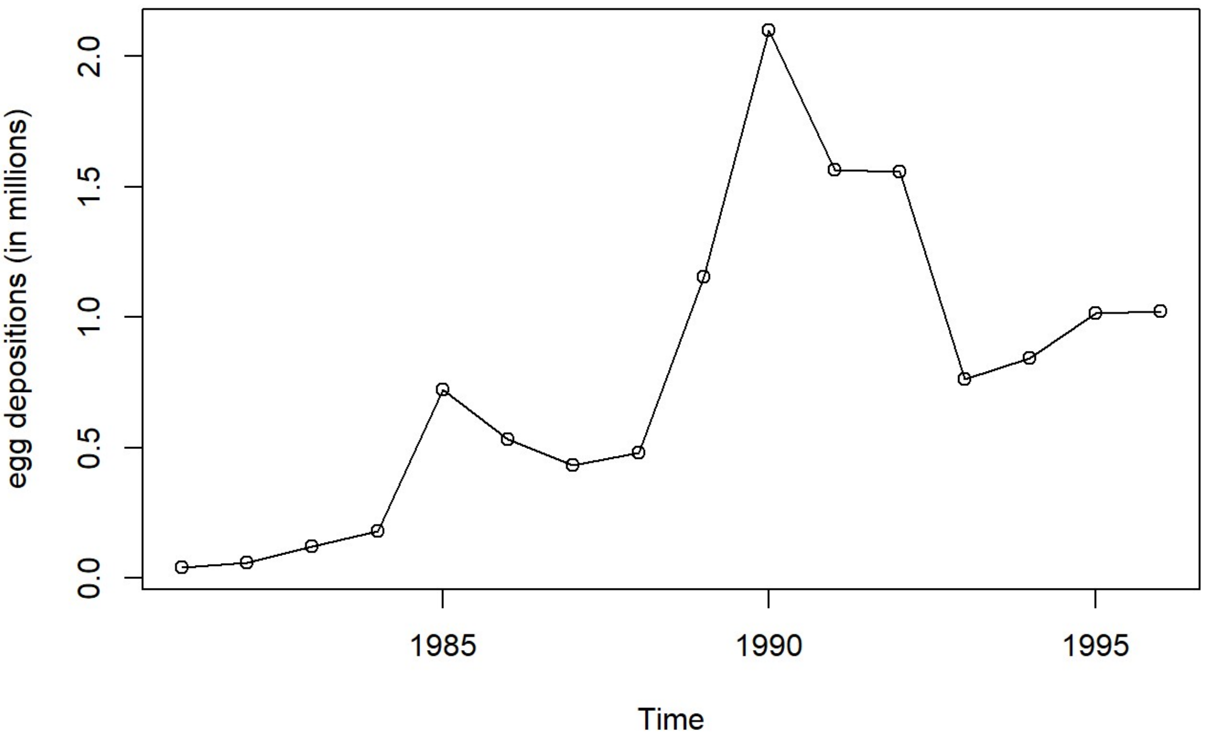


Figure 1.

Interpretation:

As we can see the class is time series object and thereby the graph plotted is the time series plot.The discussion of the produced plot is based on 4 aspects.

1. Trend: There is clear display of upward trend, implying that egg depositions has increased over the years.
2. Seasonality: There is no clear indication of seasonality. As there is no repeating pattern seen.
3. Changing variance: There is the presence of variance in the middle of series.

4. Autoregressive behaviour or Moving Average: It seems that succeeding measurements are related to one another as the values that are neighbors in time trend to be similar in size. If so, we might be able to use one year's eggs deposition value to help forecast next year's value. But there is no apparent presence of moving average.

The following code chunk generates a scatter plot to investigate the relationship between pairs of consecutive deposition values:

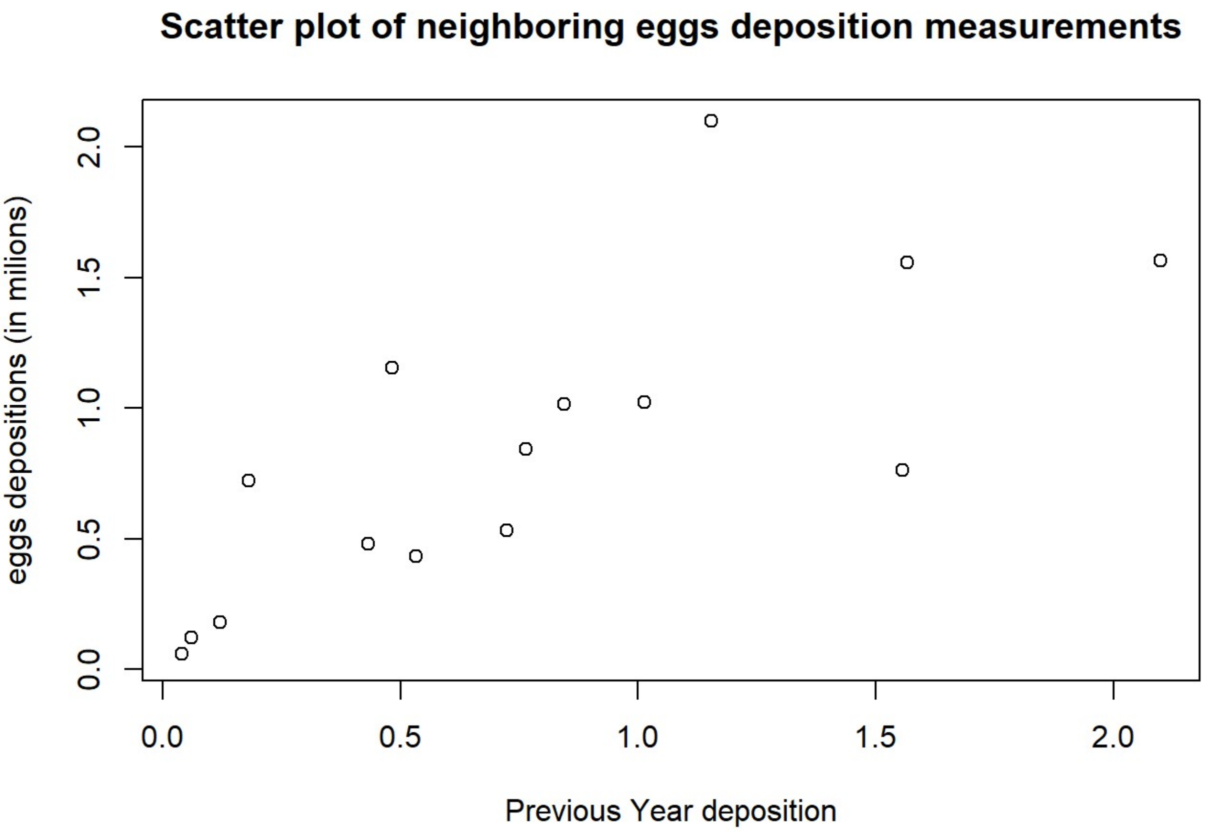


Figure 2.

We observe a strong upward trend. We observe from this plot that there is an indication of this years deposition from last year's value for many years but there are certain years for which there is high variation from the last year's value. Corrrelation seems to be good and can be calculated as follows.

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

As expected, we observe a high correlation between eggs deposition in one year with the succeeding year.

Modelling

Now to decide which models to apply. As the data displays the trend factor so let's begin by applying trend models

I) Linear trend model

```
##
## Call:
## lm(formula = eggs.ts ~ time(eggs.ts))
##
## 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(eggs.ts)    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
```

1. Both the coefficients (Intercept and linear component) are significant.
2. F-statistic value is significant implying the overall significance of the above linear trend model.
3. R-squared value is moderate implying the linear model is good but there is scope of improvement.

Let’s fit the trend line to check whether the model is good fit or not.

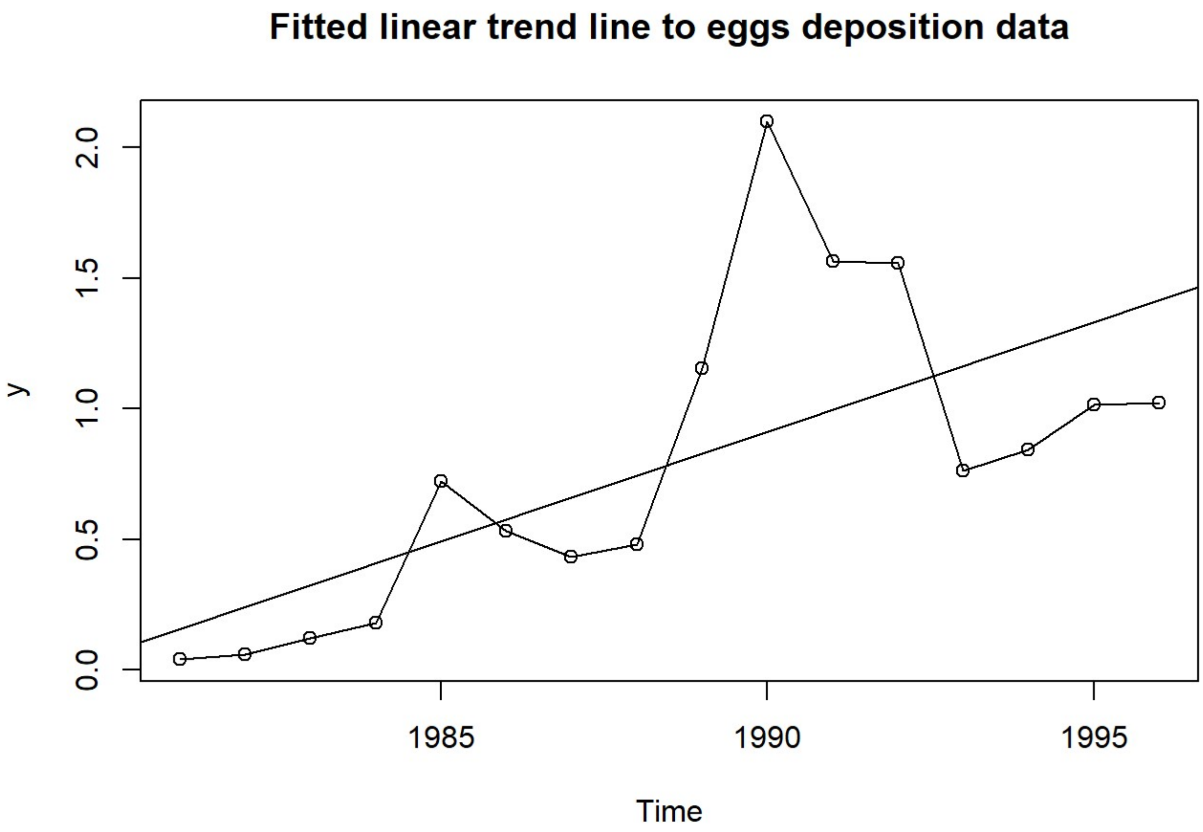


Figure 3.

The trend line fits fine but is not the optimal one as the distance between the data points and the trend line is quite considerable at many time points.

So we can think of trying the quadratic model next.

II) Quadratic Model:

```
##
## Call:
## lm(formula = eggs.ts ~ 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
```

1. All the coefficients are significant.
2. F-statistic value is also significant.
3. R-squared value has improved over linear model. So the quadratic model seems to be better fitting model.

Plotting the fitted quadratic curve along with the observed ozone width series.

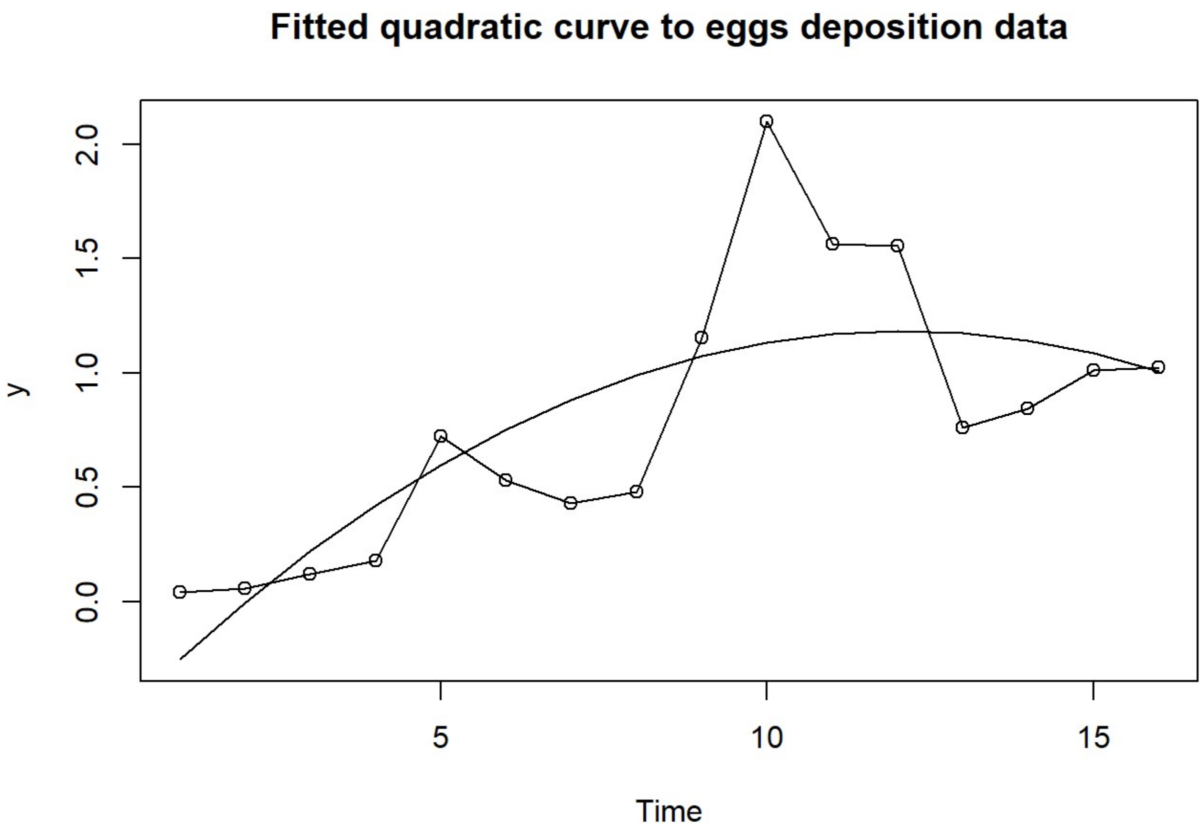


Figure 4.

The quadratic line seems to fit better to the data in comparison to the linear trend line.

Just to check if there is some presence or effect of cyclical or seasonal trend, we convert the annual data with frequency 1 to the one with higher frequency (=12). And try apply the seasonal models.

Converting annual data to seasonal data with frequency equal to 12 and plotting.

```
eggs2.ts <- ts(as.vector(eggs$eggs), start=c(1981,1), end= c(1996,12), frequency=12)
head(eggs2.ts)
```

```
##         Jan    Feb    Mar    Apr    May    Jun
## 1981 0.0402 0.0602 0.1205 0.1807 0.7229 0.5321
```

```
month.=season(eggs2.ts)
```

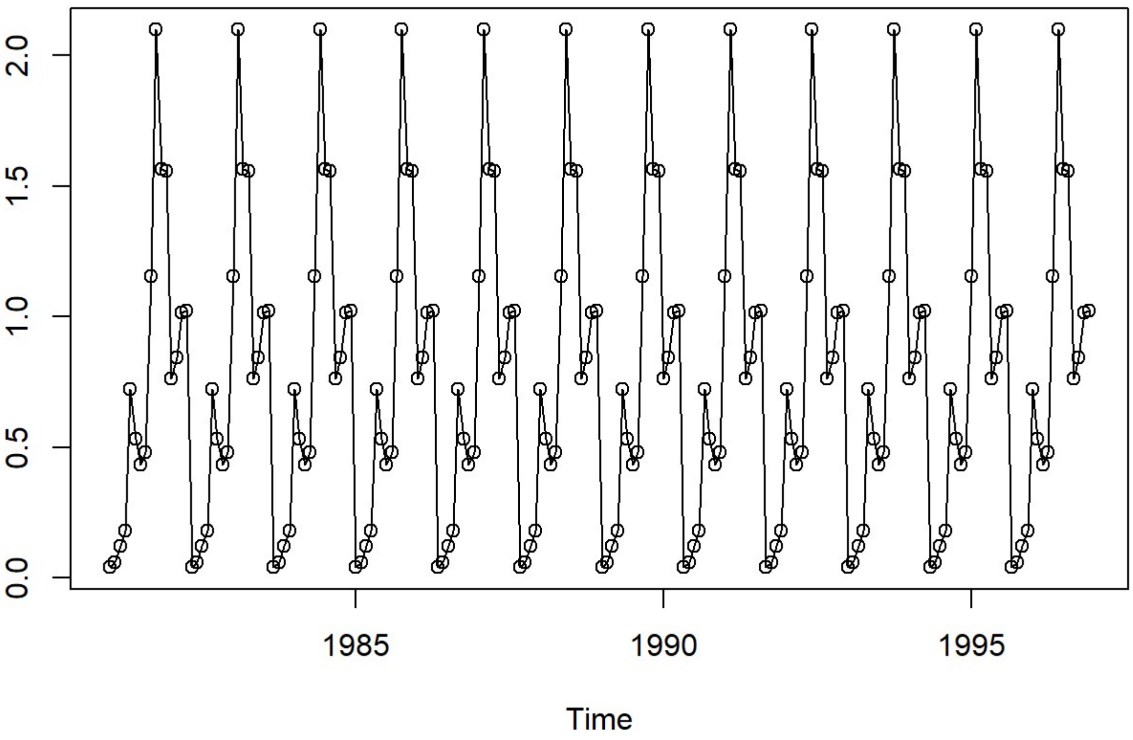


Figure 5.

III) Seasonal model

```
##
## Call:
## lm(formula = eggs2.ts ~ month. - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82332 -0.42109  0.00629  0.29431  1.21487
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## month.January     0.6702     0.1480   4.528 1.08e-05 ***
## month.February    0.8835     0.1480   5.969 1.24e-08 ***
## month.March       0.7831     0.1480   5.291 3.50e-07 ***
## month.April       0.8107     0.1480   5.478 1.44e-07 ***
## month.May         0.6702     0.1480   4.528 1.08e-05 ***
## month.June        0.8835     0.1480   5.969 1.24e-08 ***
## month.July        0.7832     0.1480   5.291 3.50e-07 ***
## month.August      0.8107     0.1480   5.478 1.44e-07 ***
## month.September   0.6702     0.1480   4.528 1.08e-05 ***
## month.October     0.8835     0.1480   5.969 1.24e-08 ***
## month.November    0.7832     0.1480   5.291 3.50e-07 ***
## month.December    0.8107     0.1480   5.478 1.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.592 on 180 degrees of freedom
## Multiple R-squared:  0.6555, Adjusted R-squared:  0.6325
## F-statistic: 28.54 on 12 and 180 DF,  p-value: < 2.2e-16
```

All monthly coefficients are significant. So effects of all months are significant in this model.

To see how this seasonal model fits the data, we plot the model along with series and check the results

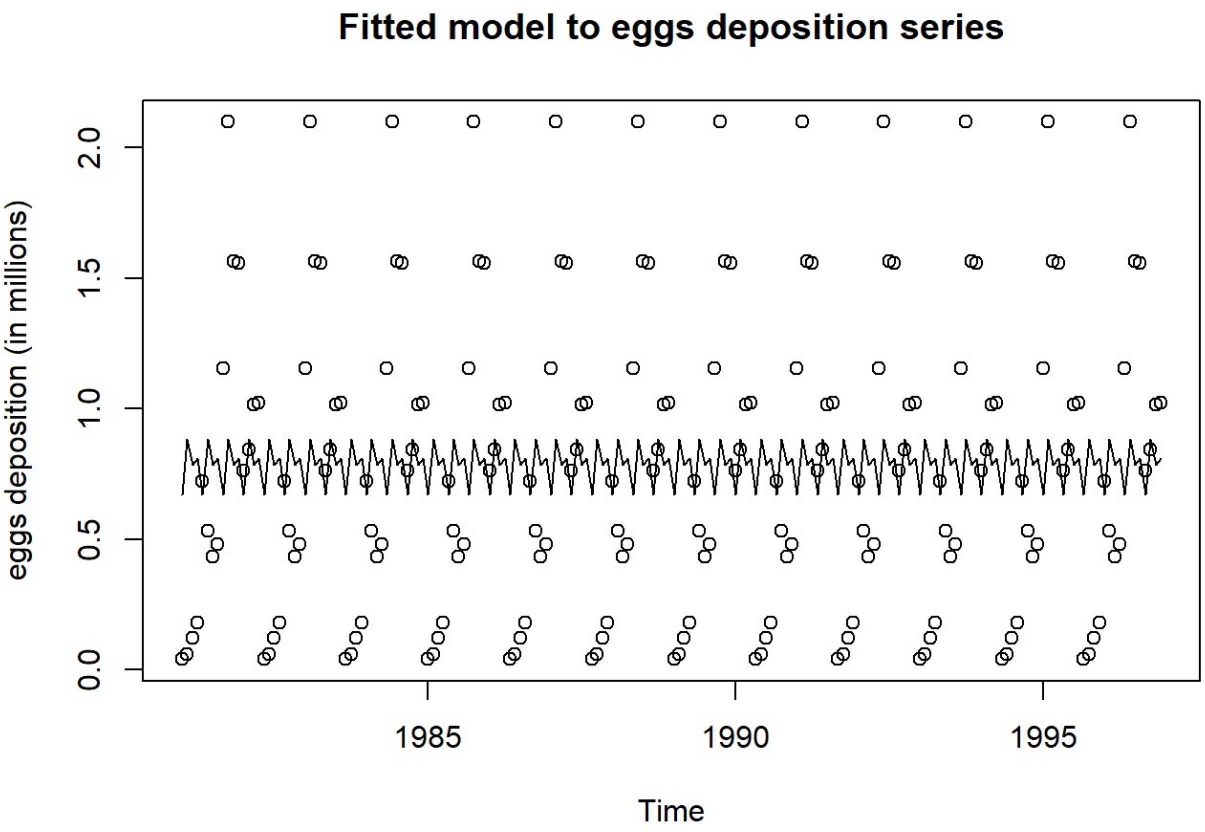


Figure 6.

As clearly seen that model doesn't befit the data plot implying that the cyclic model is not an appropriate model for modelling ozone depletion series.

Let's try fitting the harmonic model to the data.

IV) Harmonic Model

```
##
## Call:
## lm(formula = eggs2.ts ~ har.)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7467 -0.4179 -0.0439  0.2698  1.3115
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.869e-01  4.207e-02    18.7   <2e-16 ***
## har.cos(2*pi*t)  5.212e-16  5.949e-02     0.0        1
## har.sin(2*pi*t) -4.306e-15  5.949e-02     0.0        1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5829 on 189 degrees of freedom
## Multiple R-squared:  2.943e-29,  Adjusted R-squared:  -0.01058
## F-statistic: 2.781e-27 on 2 and 189 DF,  p-value: 1
```

The harmonic coefficients are not significant, implying that harmonic model is not the suitable one for modelling and forecasting eggs deposition value.

Inference from modelling results:

Considering the statistical results and plot outputs of the above models we choose linear trend model and quadratic model for the diagnostic testing, as seasonal model and harmonic model do not come up as the suitable models for the given data series.

Diagnostic Testing for trend models

A) Residual Analysis:

If the trend model is reasonably correct, then the residuals should behave roughly like the true stochastic component (white noise). Let’s check it for the models we applied above.

1. Residual Plot for linear model:

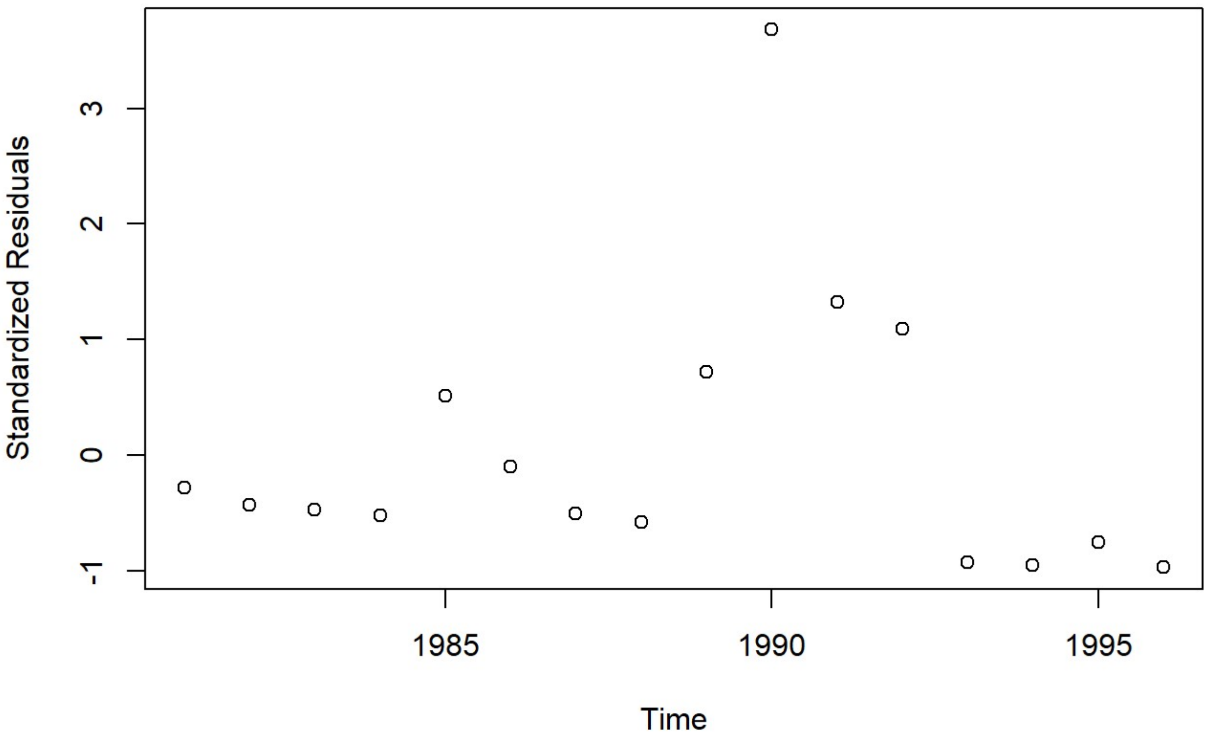


Figure 7.

As seen the residual plot is not random but it shows presence of trend and moving average autoregressive behaviour so let’s further check for quadratic model to see if it is better in terms of residuals and R^2 too.

2. Residual Plot for quadratic model

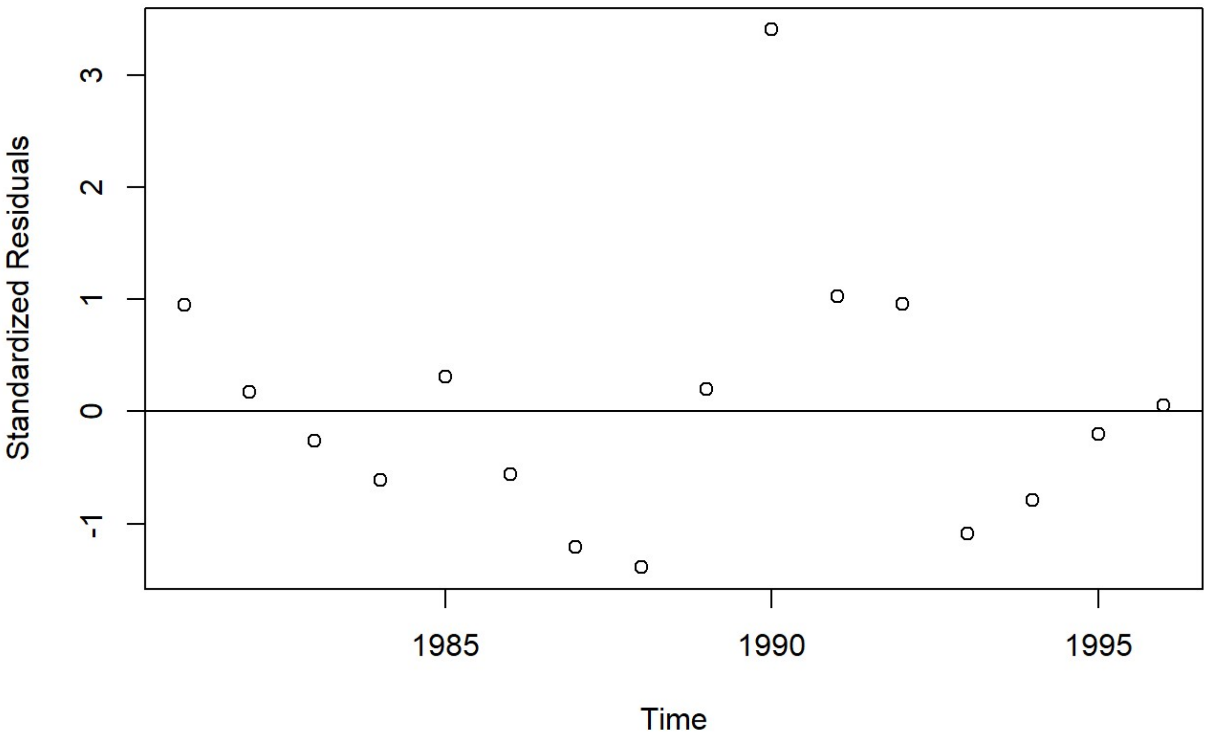


Figure 8.

Still the residuals are not random. There seems to be the presence of local trends and moving average.

B) R-squared value

Another measure of diagnostic testing is R-squared value

1. For linear model: 0.4074
2. For quadratic model: 0.5306

C) QQ plot

Third measure of diagnostic testig is quantile-quantile (QQ) plot.

1. QQ plot for linear model

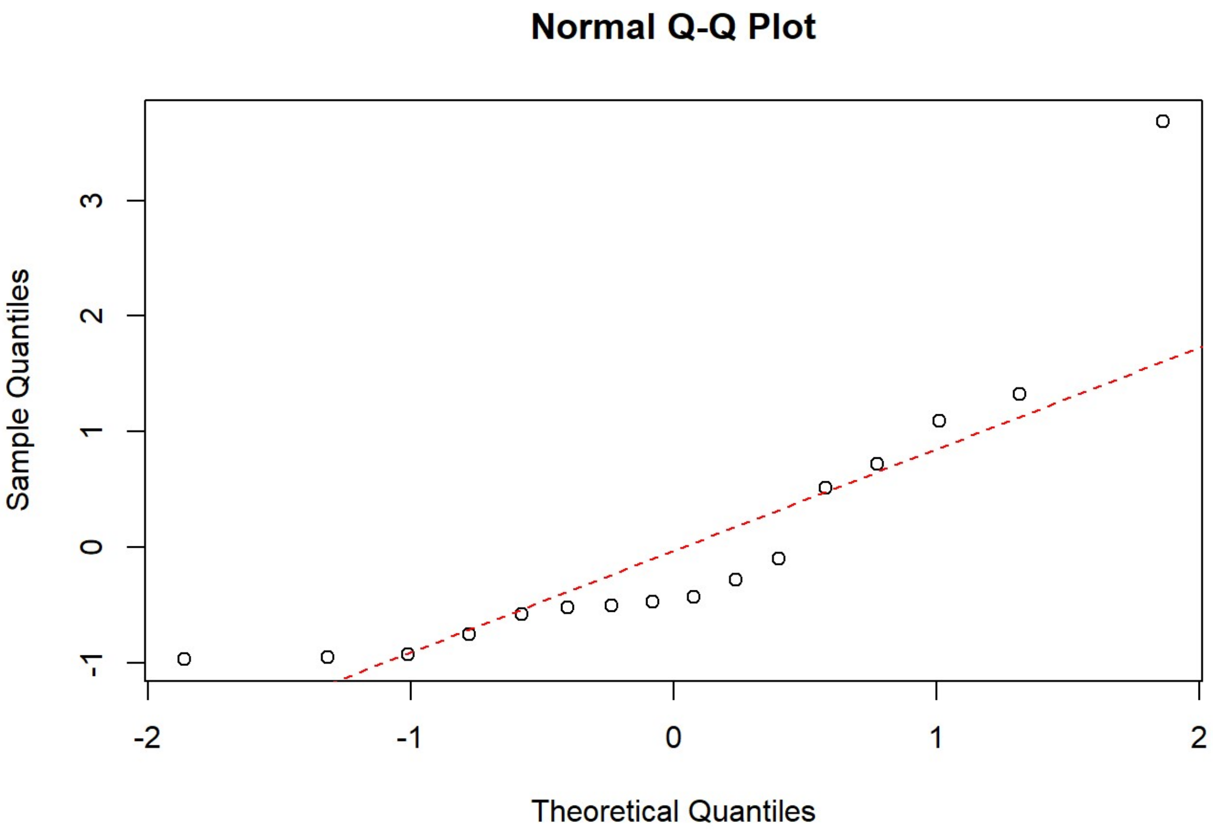


Figure 9.

The QQ plot for the linear model shows much variation from the normal at many points.

2. QQ plot for quadratic model

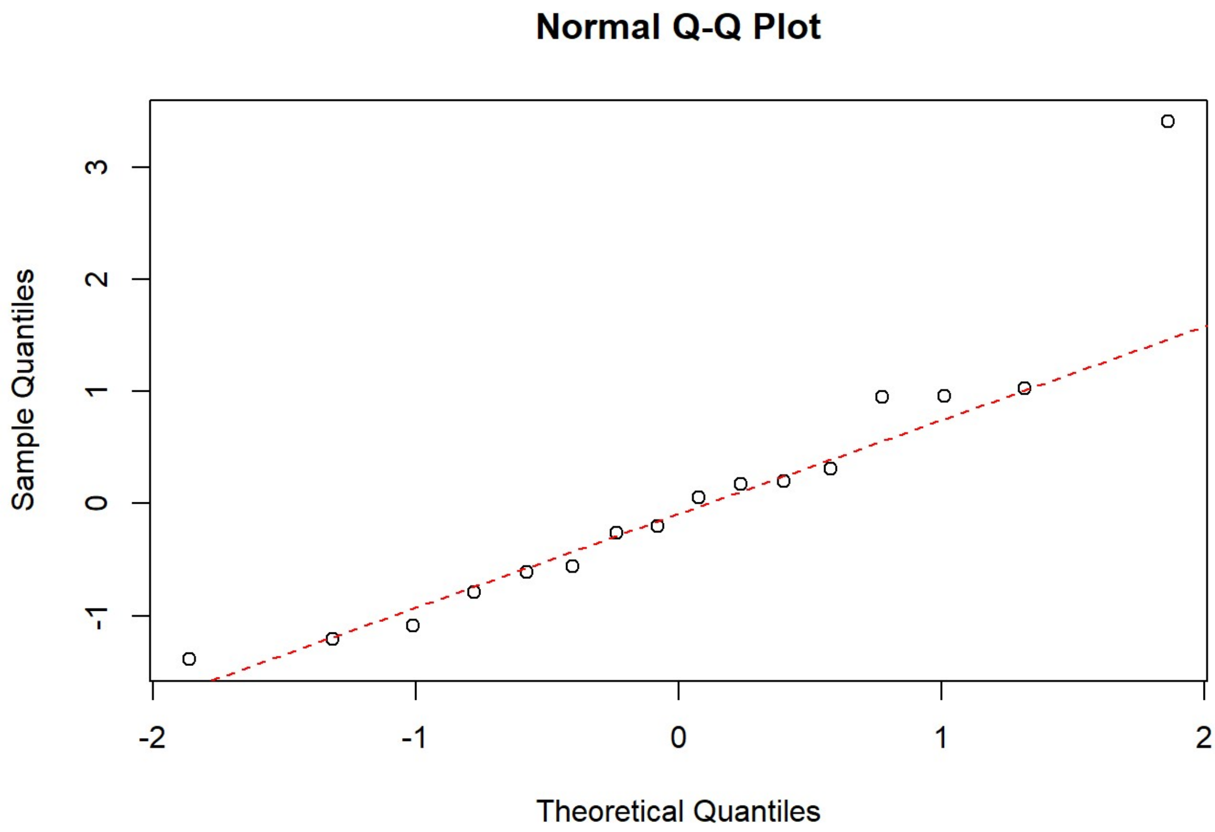


Figure 10.

The straight-line pattern here supports the assumption of a normally distributed stochastic component in this model. Much better than the linear model QQ plot.But at the ends there is huge variation.

D) Shapiro Wilk Test

Fourth measure of diagnostic testing is Shapiro Wilk normality test.

1. Shapiro Wilk Test for linear model

```
##
##  Shapiro-Wilk normality test
##
## data:  y
## W = 0.7726, p-value = 0.001205
```

As the p-value for the linear model is smaller than alpha, hence the data is not normally distributed.

2. Shapiro Wilk test for quadratic model

```
##
##  Shapiro-Wilk normality test
##
## data:  y
## W = 0.87948, p-value = 0.03809
```

As the p-value for the quadratic model is less than alpha , hence the data is not normally distributed. The p value is greater than the linear model case but still less than 0.05. Therefore, we even quadratic model is not the apt model considering normality of residuals.

E) ACF plot

The fifth measure of diagnosis is the acf plot.

1. ACF plot for linear model

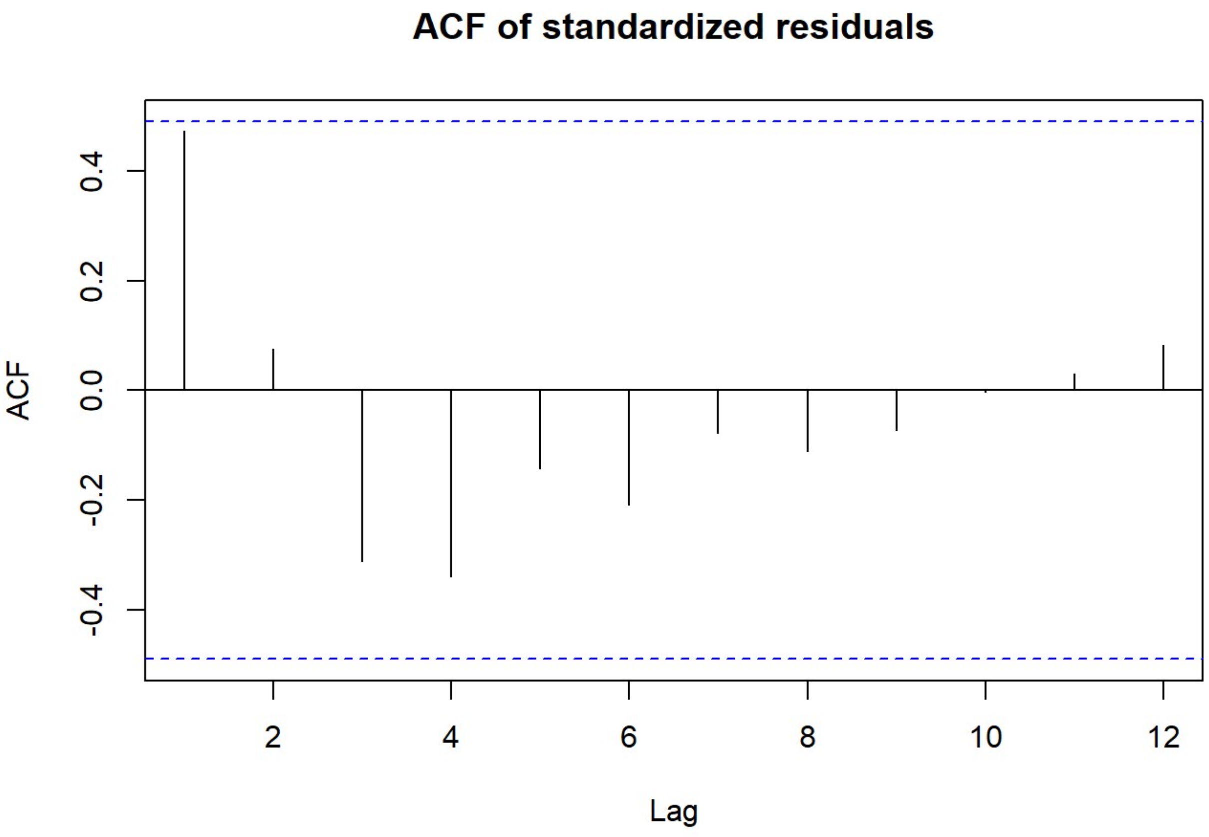


Figure 11.

There is no presence of significant auto correlation for any lag. So following the linear model we seem to have got rid of autoregressive behaviour.

2. ACF plot for quadratic model

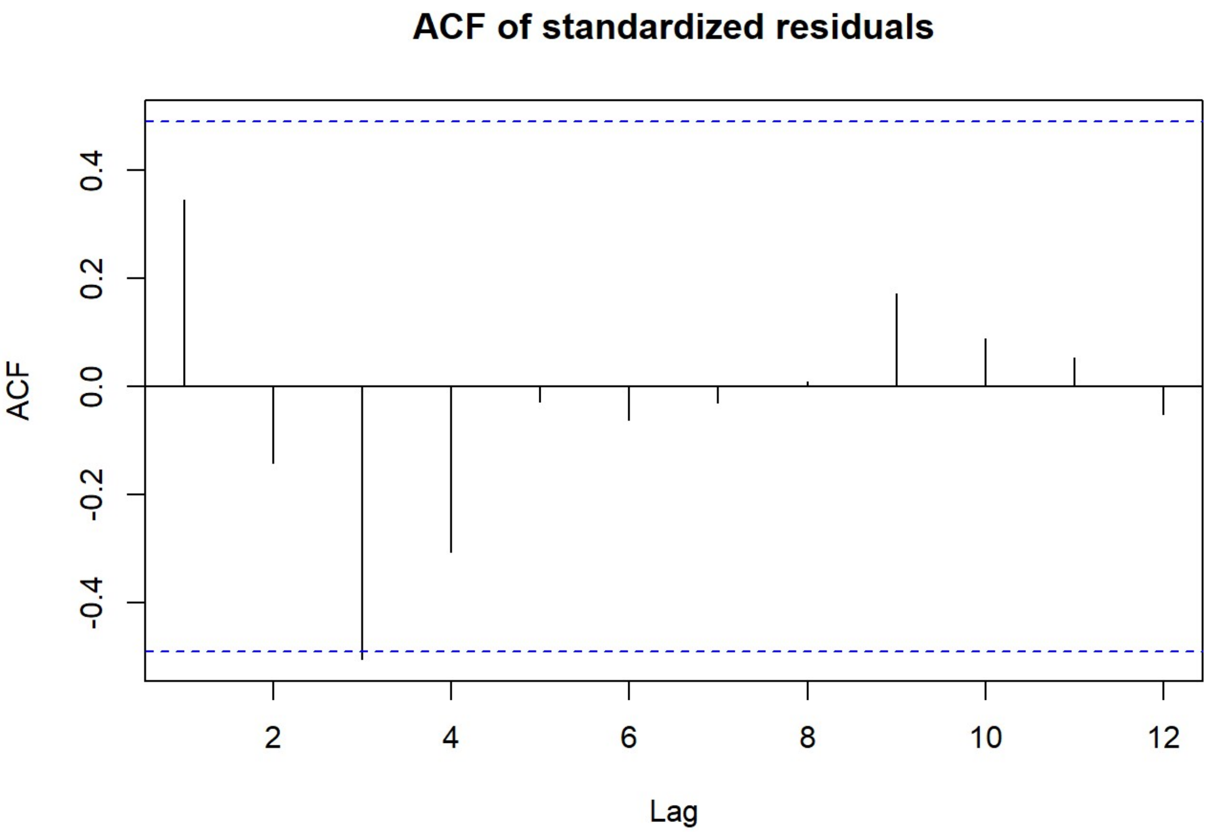


Figure 12.

There is presence of some significant auto correlation for the quadratic model. So besides moving average and trend there is an apparent presence of auto correlation also in the residuals.

Interpretation from Diagnostic Testing:

1. Based on residual testing for both linear and quadratic model there is presence of trend and moving average in the residuals.
2. R-square value of both linear and quadratic models is moderate.
3. Considering QQ plot and the Shapiro Wilk normality test, none of the two models is apt.
4. Autoregressive behaviour is captured very well by linear and quadratic models. Though there is an apparent indication of auto regressive behaviour for the residuals of quadratic model.

Conclusion: Since none of the linear or seasonal models are found to meet the criteria of the diagnostic tests so we need to look for other models.

Let’s begin the further analysis by plotting acf and pcf plot for the egg deposition series.

ACF and PACF plot for eggs deposition time series

```
par(mfrow=c(1,2))
acf(eggs.ts)
pacf(eggs.ts)
```

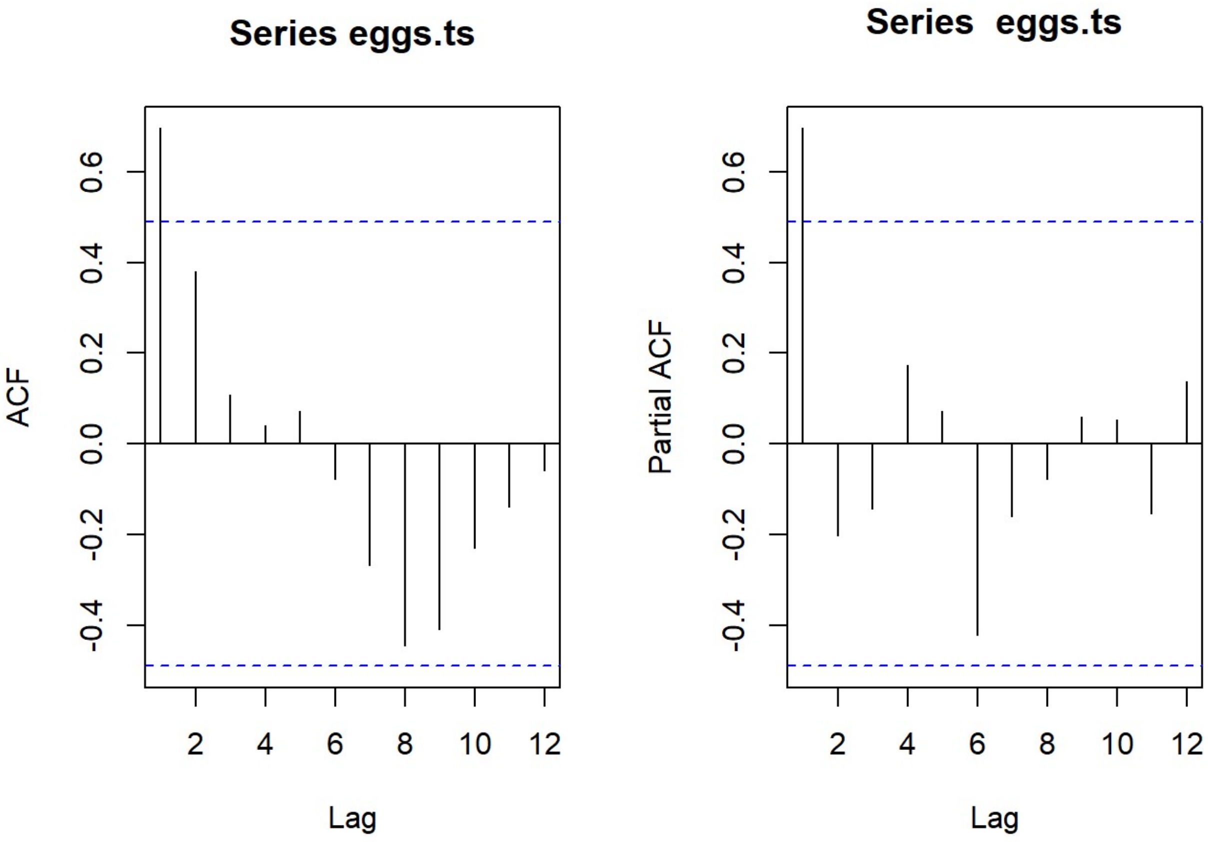


Figure 13.

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

Slowly decaying pattern in ACF and and very high first correlation in PACF implies the existence of trend and non-stationarity.

Apply ADF test: Hypothesis test to decide on the existence of a trend in the series.

```
adf.test(eggs.ts)
```

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

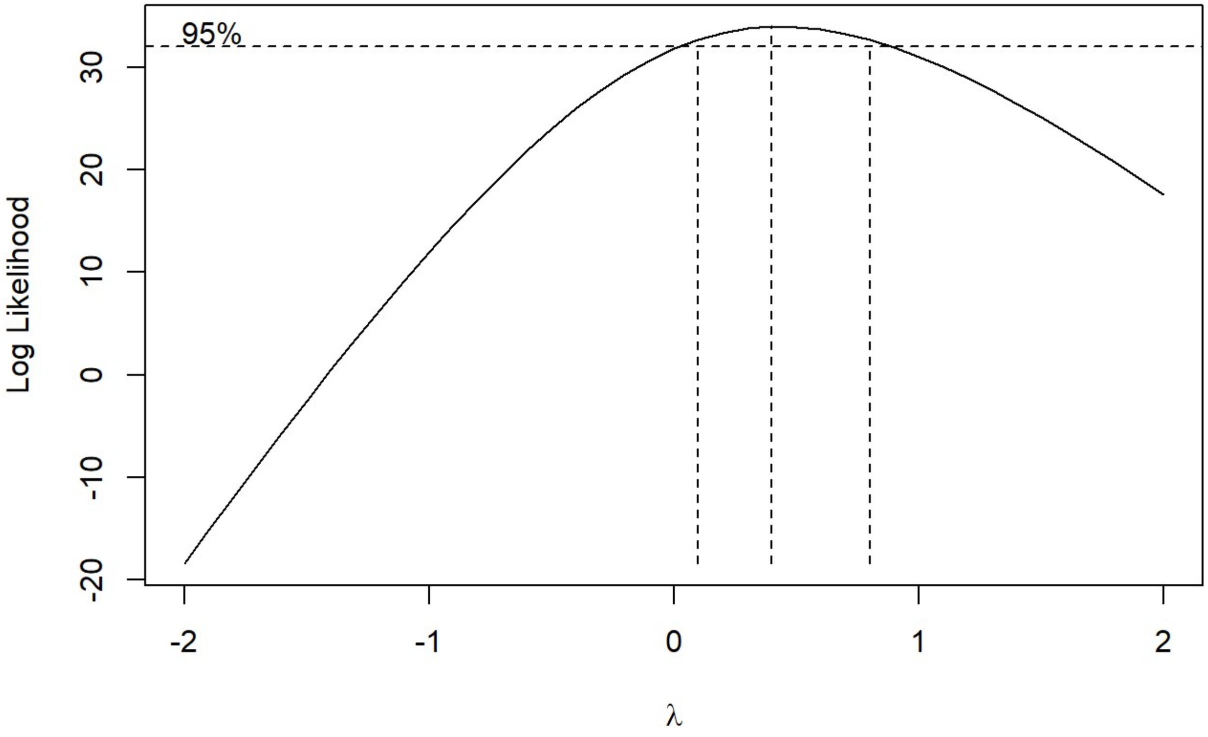
With p-value of 0.5469, we cannot reject the null hypothesis stating that the series is non- stationary.

Now, we will try to overcome the non-stationarity nature of the series by using suitable tools.

Let’s first apply the box-Cox transformation.

```
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible
## convergence problem: optim gave code = 1

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible
## convergence problem: optim gave code = 1
```



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

The confidence interval comes out to be (01, 0.8). so we choose lambda value as 0.45 and perform the analysis.

Plot for Box-Cox transformed series.

```
par(mfrow=c(1,2))
lambda=0.45 # midpoint of interval (0.1, 0.8)
eggs.bc= (eggs.ts^lambda-1)/lambda
plot(eggs.ts, type='o', ylab ='count for Original series')
plot(eggs.bc, type='o', ylab ="count for transformed series")
```

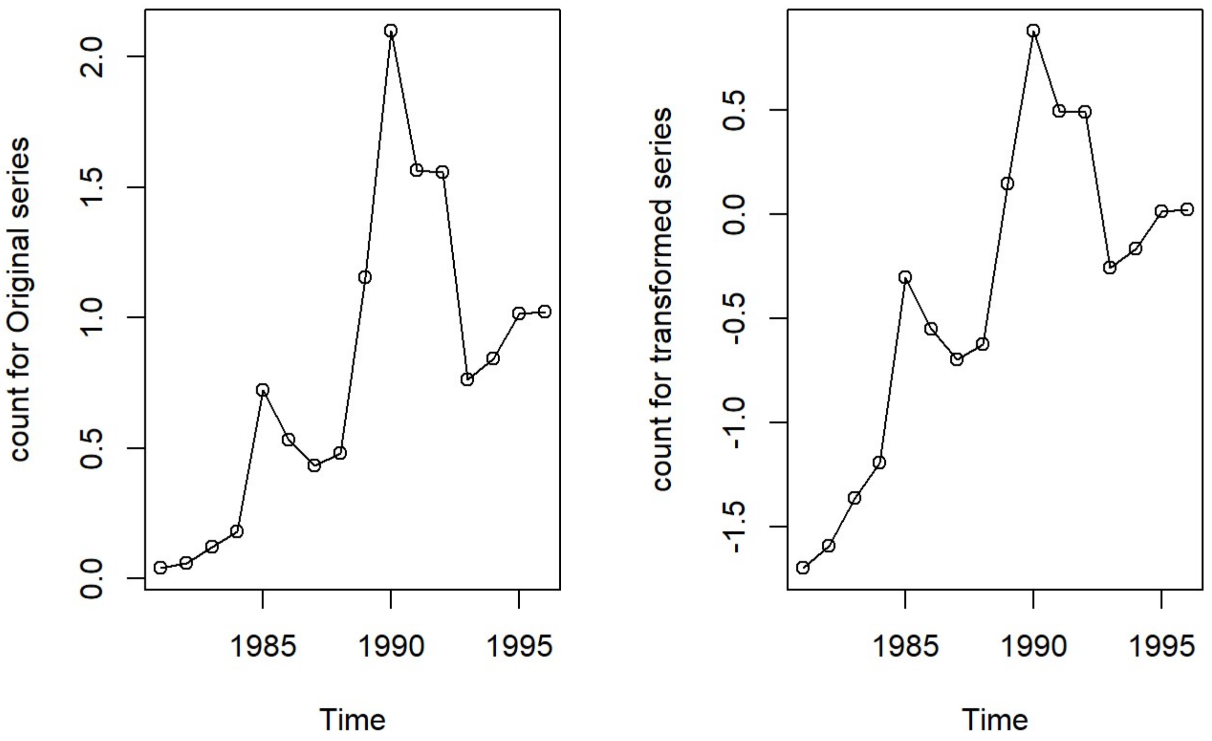


Figure 14.

We see that after applying the box-Cox transformation, the variance has decreased.

So, let’s try to detrend the series using differencing and check for stationarity of the resultant series.

We start with first differencing and plot the series.

Plot after first differencing

```
par(mfrow=c(1,1))
eggs.bc.diff1 <- diff(eggs.bc, differences=1)
plot(eggs.bc.diff1, ylab='Change in eggs depositions', type='l')
```

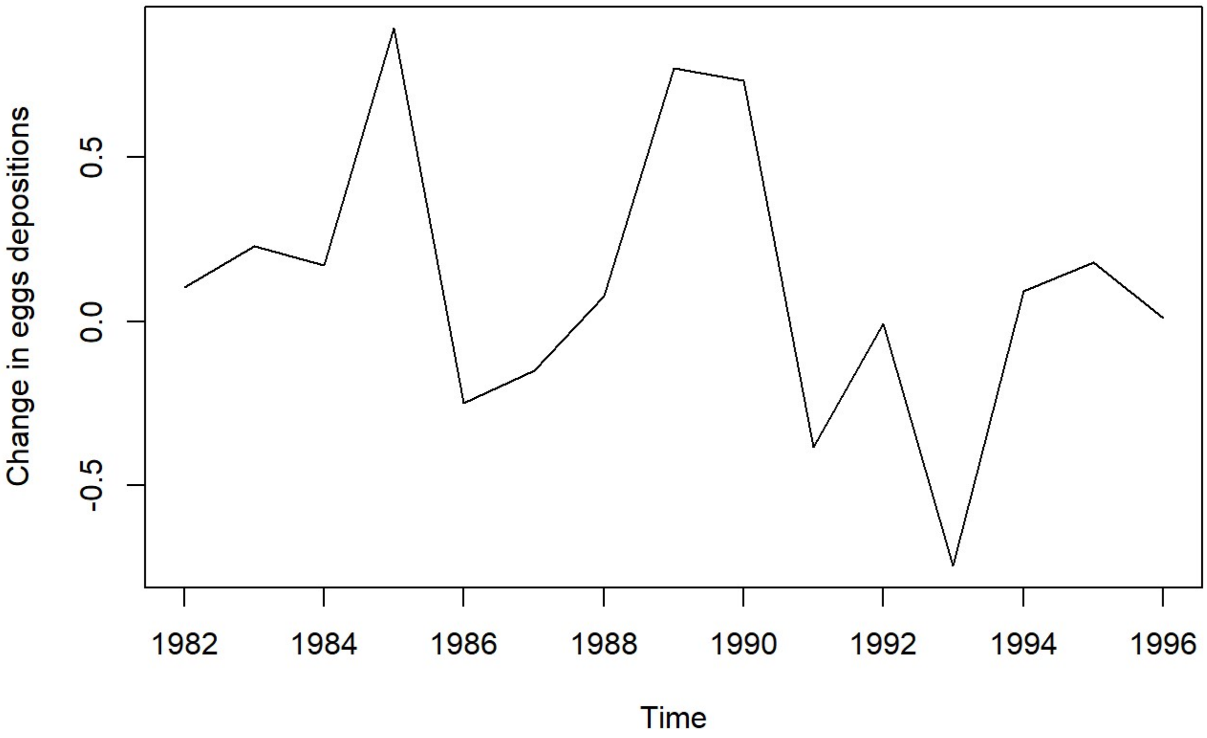


Figure 15.

We see that there is still some trend in the series. And series doesn’t appear to be completely stationary.

Let’s apply ADF unitroot test to the differenced series to test for stationarity.

```
order=ar(diff(eggs.bc.diff1))$order
adfTest(eggs.bc.diff1, lags = order,  title = NULL,description = NULL)
```

```
## 
## Title:
##  Augmented Dickey-Fuller Test
## 
## Test Results:
##   PARAMETER:
##     Lag Order: 4
##   STATISTIC:
##     Dickey-Fuller: -0.8222
##   P VALUE:
##     0.3469 
## 
## Description:
##  Sun May 06 22:11:38 2018 by user: user
```

With a p-value of 0.3469, we fail to reject the null hypothesis stating that the series is non-stationary.

So we apply the second differencing to check that if tis makes the original eggs deposition series stationary.

Plot after second differencing

```
par(mfrow=c(1,1))
eggs.bc.diff2 <- diff(eggs.bc, differences=2)
plot(eggs.bc.diff2, ylab='Change in eggs depositions', type='l')
```

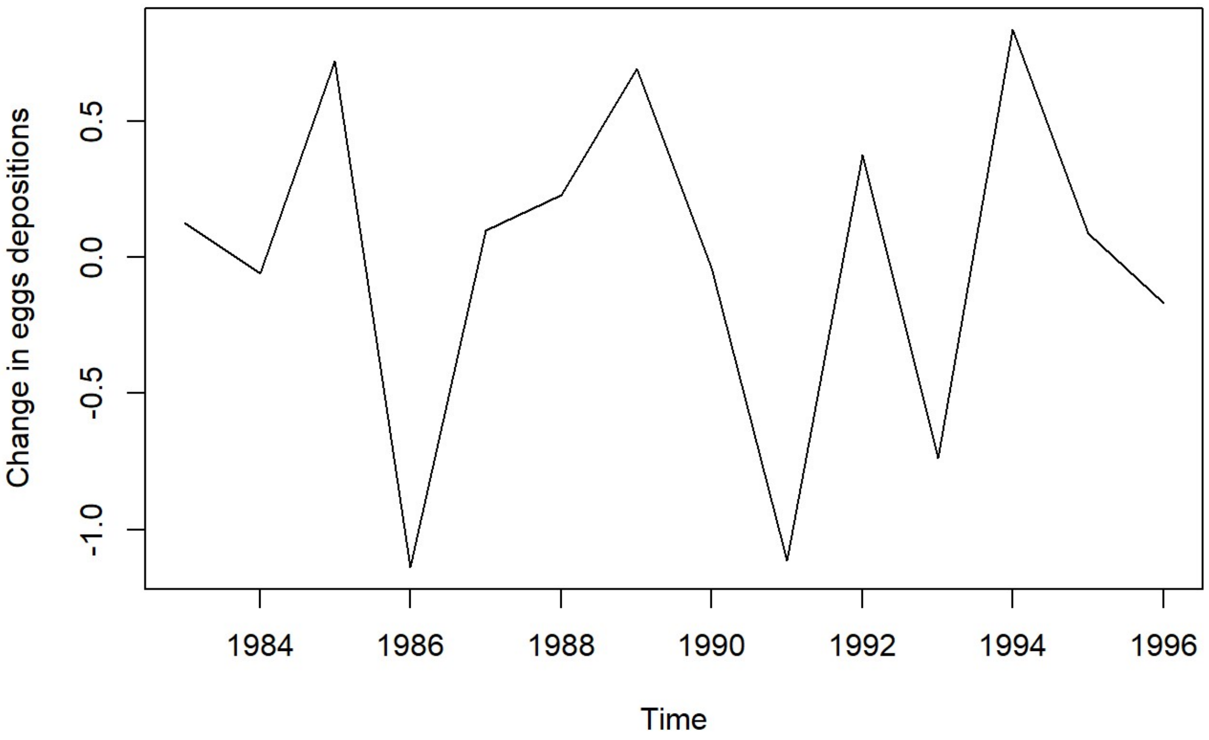


Figure 16.

The series seems to be detrended after second differencing but series do not seem to be symmetric around zero level.

Let's check for the stationarity by applying adf test on second differenced series.

```
order=ar(diff(eggs.bc.diff2))$order
adfTest(eggs.bc.diff2, lags = order,  title = NULL,description = NULL)
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 4
##   STATISTIC:
##     Dickey-Fuller: -1.5692
##   P VALUE:
##     0.1098
##
## Description:
##  Sun May 06 22:11:38 2018 by user: user
```

With a p-value of 0.1098, we fail to reject the null hypothesis stating that the series is non-stationary.

So, we go to third differencing.

Plot after third differencing

```
par(mfrow=c(1,1))
eggs.bc.diff3 <- diff(eggs.bc, differences=3)
plot(eggs.bc.diff3, ylab='Change in eggs depositions', type='l')
```

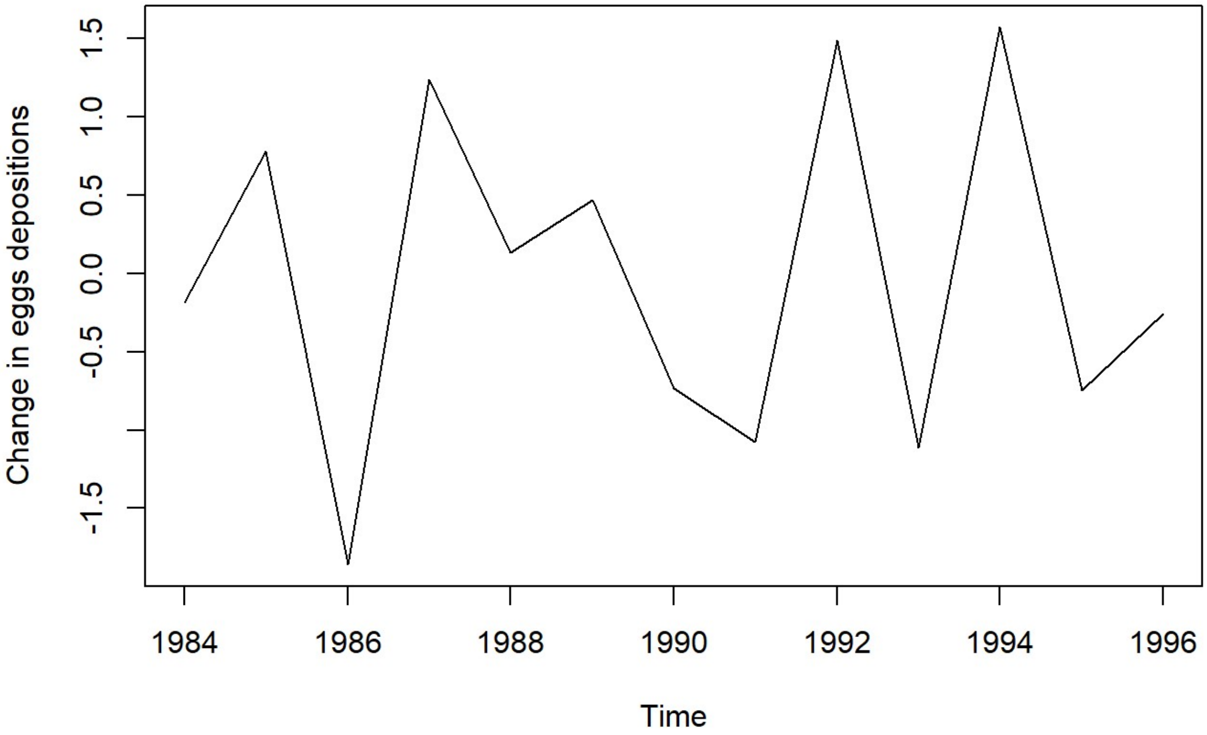


Figure 17.

After applying third differencing we witness a presence of trend which is the hint for non-stationarity.

We confirm our insight using adf test.

```
order=ar(diff(eggs.bc.diff3))$order
adfTest(eggs.bc.diff3, lags = order,  title = NULL,description = NULL)
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 4
##   STATISTIC:
##     Dickey-Fuller: -1.3368
##   P VALUE:
##     0.1836
##
## Description:
##  Sun May 06 22:11:38 2018 by user: user
```

With a p-value of 0.1836, we fail to reject the null hypothesis stating that the series is non-stationary. So even after third differencing the series is non-stationary.

So we consider fourth differencing.

Plot after fourth differencing

```
par(mfrow=c(1,1))
eggs.bc.diff4 <- diff(eggs.bc, differences=4)
plot(eggs.bc.diff4, ylab='Change in eggs depositions', type='l')
```

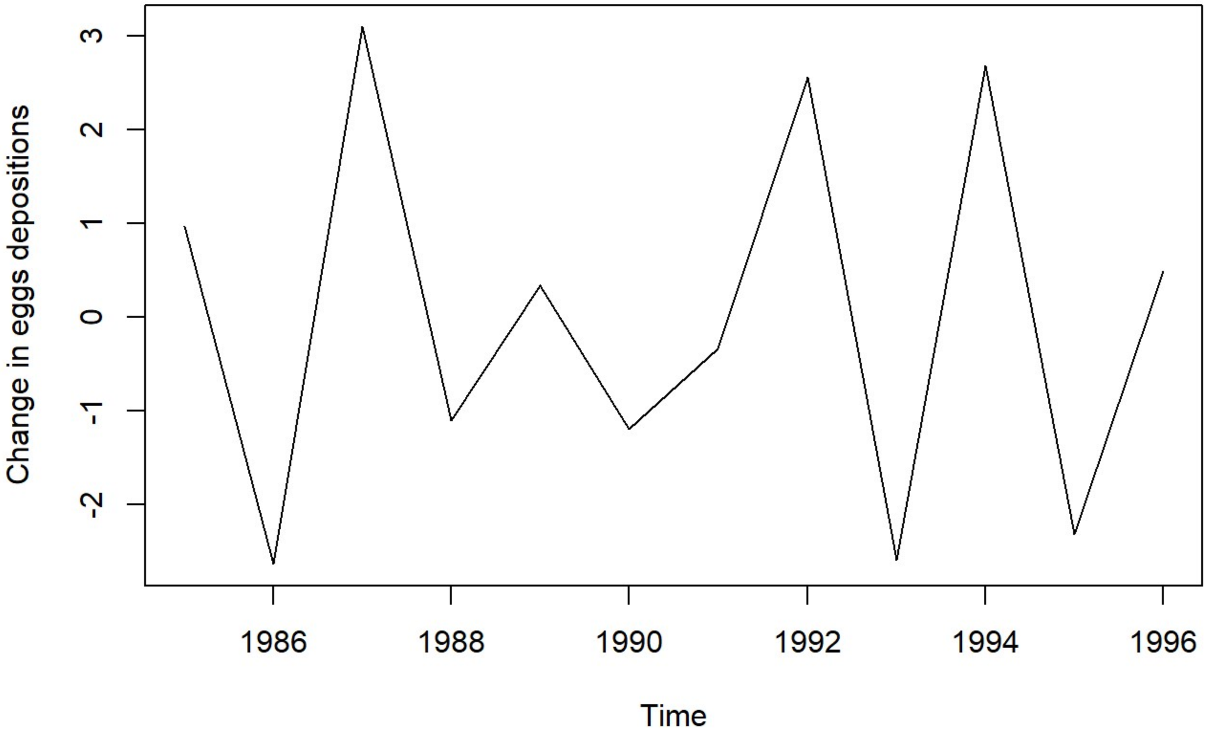


Figure 18.

The plot for fourth differencing is detrended and do not show variance, so, it seems to be a stationary series.

Let’s confirm it using adf test.

```
order=ar(diff(eggs.bc.diff4))$order
adfTest(eggs.bc.diff4, lags = order,  title = NULL,description = NULL)
```

```
## 
## Title:
##  Augmented Dickey-Fuller Test
## 
## Test Results:
##   PARAMETER:
##     Lag Order: 2
##   STATISTIC:
##     Dickey-Fuller: -2.3228
##   P VALUE:
##     0.02265 
## 
## Description:
##  Sun May 06 22:11:39 2018 by user: user
```

So, with a p-value of 0.02265 we reject the null hypothesis that the series is non-stationary. Therefore series has become stationary after fourth differencing.

Now we will check the acf and pacf plots for the fourth differenced series.

ACF and PACF plot for fourth differenced transformed series

```
par(mfrow=c(1,2))
acf(eggs.bc.diff4)
pacf(eggs.bc.diff4)
```

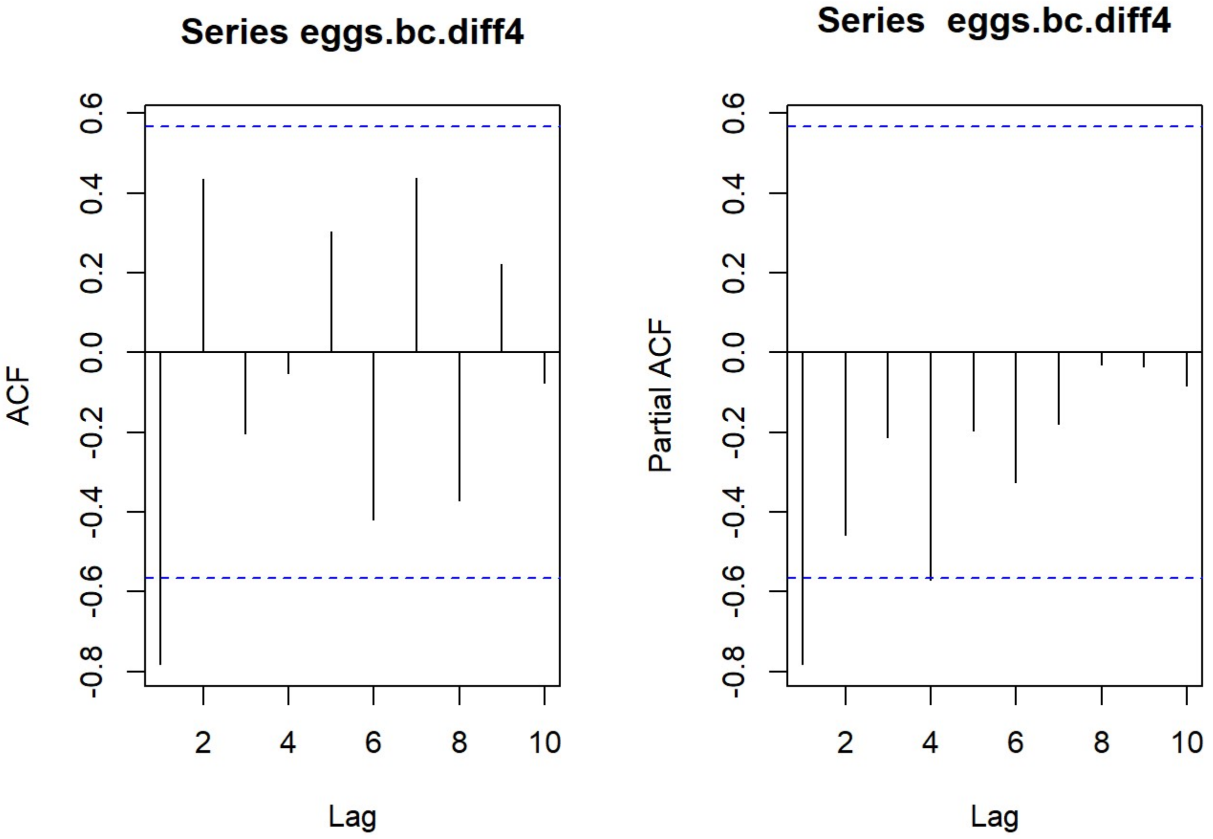


Figure 19.

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

There is one highly significant lag in ACF and two significant lags in PACF.

So we can include ARIMA(2,4,1) and (1,4,1) models among the tentative models.

Checking the eacf:

```
eacf(eggs.bc.diff4, ar.max=2, ma.max=2) # We put these arguments to limit the orders p
and q at 2. Otherwise, the eacf function returns error  and displays nothing.
```

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

From the output of the eacf we include ARIMA(0,4,1), ARIMA(1,4,1) and ARIMA(0,4,2) models in the set of possible models.

Let’s display the BIC table.

BIC table

```
par(mfrow=c(1,1))
res= armasubsets(y=eggs.bc.diff4, nar=2, nma=3, y.name='test', ar.method='ols')
```

```
## Warning in ar.ols(x, aic = aic, order.max = order.max, na.action =
## na.action, : model order: 6 singularities in the computation of the
## projection matrix results are only valid up to model order 5
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : nvmax reduced to 4
```

```
plot(res)
```

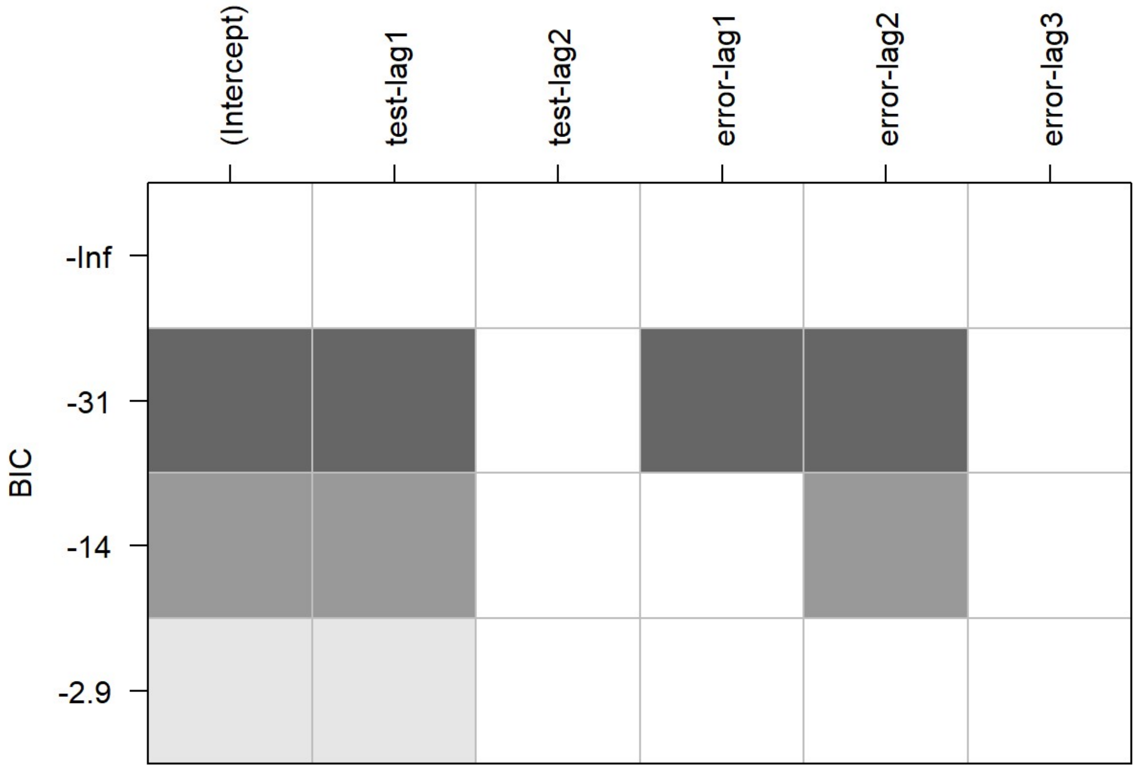


Figure 20.

In the BIC table shaded columns correspond to AR(1), MA(1) and MA(2) coefficients and so, from here we can include ARIMA(1,4,1), ARIMA(1,4,2) models in the set of candidate models.

In conclusion the set of candidate models is {ARIMA(0,4,1), ARIMA(0,4,2), ARIMA(1,4,1), ARIMA (1,4,2) and ARIMA(2,4,1)}

Now we will fit these models and select the best one according to the selection measures.

I) ARIMA(0,4,1)

ARIMA(0,4,1)-CSS method

```
model_041_css = arima(eggs.bc,order=c(0,4,1),method='CSS')
coeftest(model_041_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -1.07312    0.10255 -10.465 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For ARIMA (0,4,1) model coefficient for MA(1) component is significant using CSS method.

ARIMA(0,4,1)-MLE method

```
model_041_ml = arima(eggs.bc,order=c(0,4,1),method='ML')
coeftest(model_041_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.97867    0.20858  -4.692 2.705e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For ARIMA (0,4,1) model coefficient for MA(1) component is significant using MLE method.

II) ARIMA(0,4,2)

ARIMA(0,4,2)-CSS

```
model_042_css = arima(eggs.bc,order=c(0,4,2),method='CSS')
coeftest(model_042_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -1.86033    0.15221 -12.223 < 2.2e-16 ***
## ma2  0.98970    0.14778   6.697 2.127e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(0,4,2)-MLE

```
model_042_ml = arima(eggs.bc,order=c(0,4,2),method='ML')
coeftest(model_042_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ma1 -1.86524    0.32284 -5.7775 7.58e-09 ***
## ma2  0.94502    0.31884  2.9640 0.003037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For ARIMA (0,4,2) model coefficient for MA(2) is significant using both CSS and MLE methods.

III) ARIMA(1,4,1)

ARIMA(1,4,1)-CSS

```
model_141_css = arima(eggs.bc,order=c(1,4,1),method='CSS')
coeftest(model_141_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.68261    0.23574 -2.8956  0.003785 **
## ma1 -0.85228    0.13550 -6.2899 3.176e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(1,4,1)-MLE

```
model_141_ml = arima(eggs.bc,order=c(1,4,1),method='ML')
coeftest(model_141_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.59064    0.20856 -2.8319  0.004627 **
## ma1 -0.97232    0.23567 -4.1258 3.695e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For ARIMA(1,4,1) model AR(1) and MA(1) coefficients are significant using both CSS and MLE methods.

IV) ARIMA(1,4,2)

ARIMA(1,4,2)-CSS

```
model_142_css = arima(eggs.bc,order=c(1,4,2),method='CSS')
coeftest(model_142_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -0.73180    0.27154 -2.6950 0.007038 **
## ma1 -0.72655    0.46791 -1.5528 0.120477
## ma2 -0.13349    0.47970 -0.2783 0.780797
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(1,4,2)-MLE

```
model_142_ml = arima(eggs.bc,order=c(1,4,2),method='ML')
coeftest(model_142_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.26282    0.27976 -0.9394   0.34751
## ma1 -1.84547    0.39304 -4.6954 2.661e-06 ***
## ma2  0.90979    0.39746  2.2890   0.02208 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For ARIMA(1,4,2) model results are opposite for CSS and MLE mehods. By CSS method AR(1) component is significant while MA(2) is not. But by MLE method MA(2) component is significant while AR(1) is not.

V) ARIMA(2,4,1)

ARIMA(2,4,1)-CSS

```
model_241_css = arima(eggs.bc,order=c(2,4,1),method='CSS')
coeftest(model_241_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.96946    0.28457 -3.4067 0.0006575 ***
## ar2 -0.38226    0.29690 -1.2875 0.1979139
## ma1 -0.61123    0.32906 -1.8575 0.0632409 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(2,4,1)-MLE

```
model_241_ml = arima(eggs.bc,order=c(2,4,1),method='ML')
coeftest(model_241_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.74682    0.28775 -2.5954 0.0094479 **
## ar2 -0.21568    0.28846 -0.7477 0.4546411
## ma1 -0.96690    0.25623 -3.7736 0.0001609 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For ARIMA(2,4,1) AR(1) and MA(1) components are significant using both CSS and MLE components.

The models with all coefficients significant are ARIMA(0,4,1), ARIMA(0,4,2) and ARIMA(1,4,1).

Now we will use AIC and BIC to choose the best model.

```
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")')
  }
}
sort.score(AIC(model_041_ml,model_042_ml,model_141_ml, model_142_ml, model_241_ml), sc
ore = "aic")
```

```
##              df      AIC
## model_042_ml  3 42.61715
## model_142_ml  4 43.89656
## model_141_ml  3 44.96155
## model_241_ml  4 46.45431
## model_041_ml  2 48.39360
```

```
sort.score(BIC(model_041_ml,model_042_ml,model_141_ml, model_142_ml, model_241_ml), sc
ore = "bic")
```

```
##              df      BIC
## model_042_ml  3 44.07187
## model_142_ml  4 45.83618
## model_141_ml  3 46.41627
## model_241_ml  4 48.39393
## model_041_ml  2 49.36341
```

Both AIC and BIC select ARIMA(0,4,2) model for this series.

ARIMA(1,4,2) is an overfitting model for ARIMA(0,4,2) and we find AR(1) coefficient insignificant using ML method and MA(1) and MA(2) coefficients using CSS method.But this model is not in terms with AIC and BIC. Another overfitting model is ARIMA(0,4,3), which is fitted below to check the overfitting:

VI) ARIMA(0,4,3)

ARIMA(0,4,3)-CSS

```
model_043_css = arima(eggs.bc,order=c(0,4,3),method='CSS')
coeftest(model_043_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -2.04563    0.30687 -6.6661 2.627e-11 ***
## ma2  1.38897    0.59467  2.3357   0.01951 *
## ma3 -0.24057    0.34033 -0.7069   0.47966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARIMA(0,4,3)-MLE

```
model_043_ml = arima(eggs.bc,order=c(0,4,3),method='ML')
```

```
## Warning in stats:::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
coeftest(model_043_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -1.93497    0.42185 -4.5869 4.499e-06 ***
## ma2  1.03498    0.62442  1.6575   0.09742 .
## ma3 -0.01929    0.31165 -0.0619   0.95064
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using CSS method on ARIMA(0,4,3) model we get MA(3) coefficient to be insignificant. So we can say that ARIMA(0,4,3) is just an overfitted model.

So we stay with ARIMA(0,4,2) model.

We will go on with residual analysis of the models with sgnificant coefficients.

Diagnostics for- ARIMA(0,4,2)

```
residual.analysis <- function(model, std = TRUE,start = 2, class = c("ARIMA","GARCH","ARMA-GARCH")[1]){
  # If you have an output from arima() function use class = "ARIMA"
  # If you have an output from garch() function use class = "GARCH"
  # If you have an output from ugarchfit() function use class = "ARMA-GARCH"
  library(TSA)
  library(FitAR)
  if (class == "ARIMA"){
    if (std == TRUE){
      res.model = rstandard(model)
    }else{
      res.model = residuals(model)
    }
  }else if (class == "GARCH"){
    res.model = model$residuals[start:model$n.used]
  }else if (class == "ARMA-GARCH"){
      res.model = model@fit$residuals
  }else {
    stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")
  }
  par(mfrow=c(3,3))
  plot(res.model,type='o',ylab='Standardised residuals', main="Time series plot of standardised residuals")
  abline(h=0)
  hist(res.model,main="Histogram of standardised residuals")
  acf(res.model,main="ACF of standardised residuals")
  pacf(res.model,main="PACF of standardised residuals")
  qqnorm(res.model,main="QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = length(model$residuals)-1, StartLag = k + 1, k = 0, SquaredQ = FALSE)
}
residual.analysis(model = model_141_ml)
```

```
## Loading required package: lattice
```

```
## Loading required package: ltsa
```

```
## Loading required package: bestglm
```

```
##
## Attaching package: 'FitAR'
```

```
## The following object is masked from 'package:forecast':
##
##     BoxCox
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.91725, p-value = 0.1524
```

```
## Warning in (ra^2)/(n - (1:lag.max)): longer object length is not a multiple
## of shorter object length
```

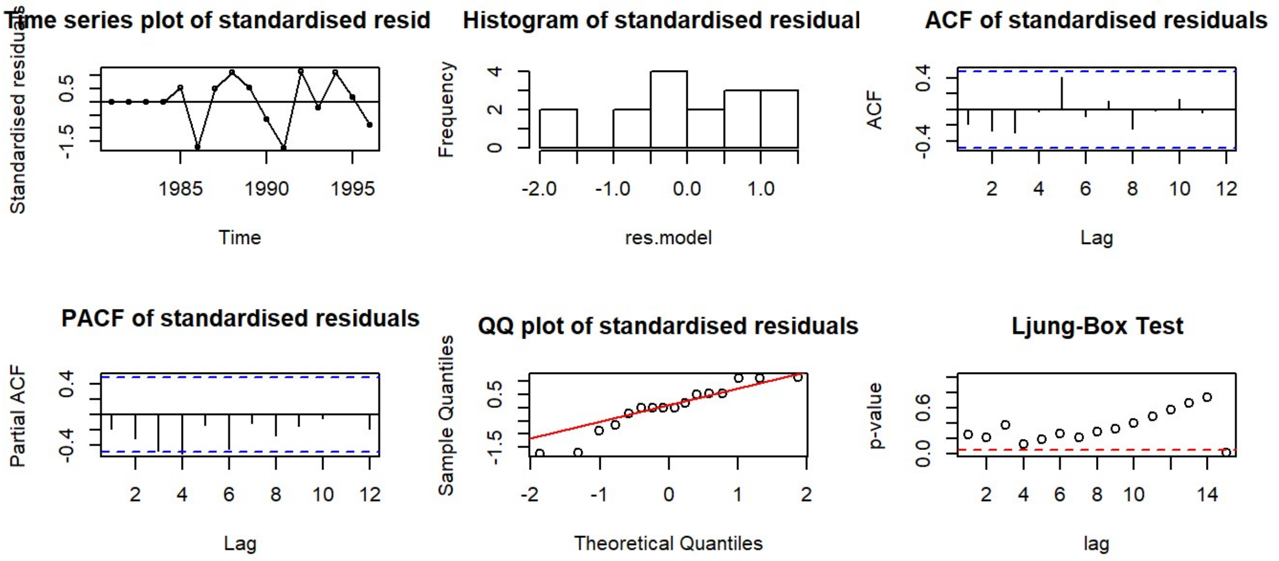


Figure 21.

There seems to be one slightly significant lag in PACF. So to confirm that does that lag has any affect on over all significance of model, we apply the Ljung-Box test for looking at the residual correlations as a whole.

```
Box.test(residuals(model_042_ml), lag=10, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  residuals(model_042_ml)
## X-squared = 13.613, df = 10, p-value = 0.1914
```

As p-value is greater than significance level, this implies we cannot reject the null hypothesis that the residuals are independent. There is no serial correlation between residuals.

Besides that, the residuals seem to be normally distributed in the qqplot but there is some deviation in the left so to check for normality we use shapiro-Wilk test.

```
shapiro.test(residuals(model_042_ml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_042_ml)
## W = 0.9356, p-value = 0.2986
```

As the p-value for the ARIMA(0,4,2) model is more than alpha , hence the data is normally distributed.

So based on the diagnostic testing we draw the following insights:

1. Residuals are randomly distributed between limit -3 to 3.
2. Histogram is normally distributed.
3. ACF and PACF do not show any significant lag. The patterns in ACF and PACF implies existence of white noise behaviour. So residuals are uncorrelated.And Ljug-Box test result supports it.
4. qqplot shows normal distribution around the straight line. And Shapiro-Wilk test confirms it.
5. All p-values are greater than 0.05 in the Ljung-Box plot, so this confirms that ARIMA(0,4,2) model successfully deals with serial correlation.

Hence, diagnostic tests reassure the suitability of ARIMA(0,4,2) model as a best fit model.

So now we can confidently use the selected model for forecasting the egg depositions for next 5 years.

Forecasting:

Forecast plot for egg deposition time series using ARIMA (0,4,2) (considering original series)

```
fit = Arima(eggs.ts,c(0,4,2))
plot(forecast(fit,h=5))
```

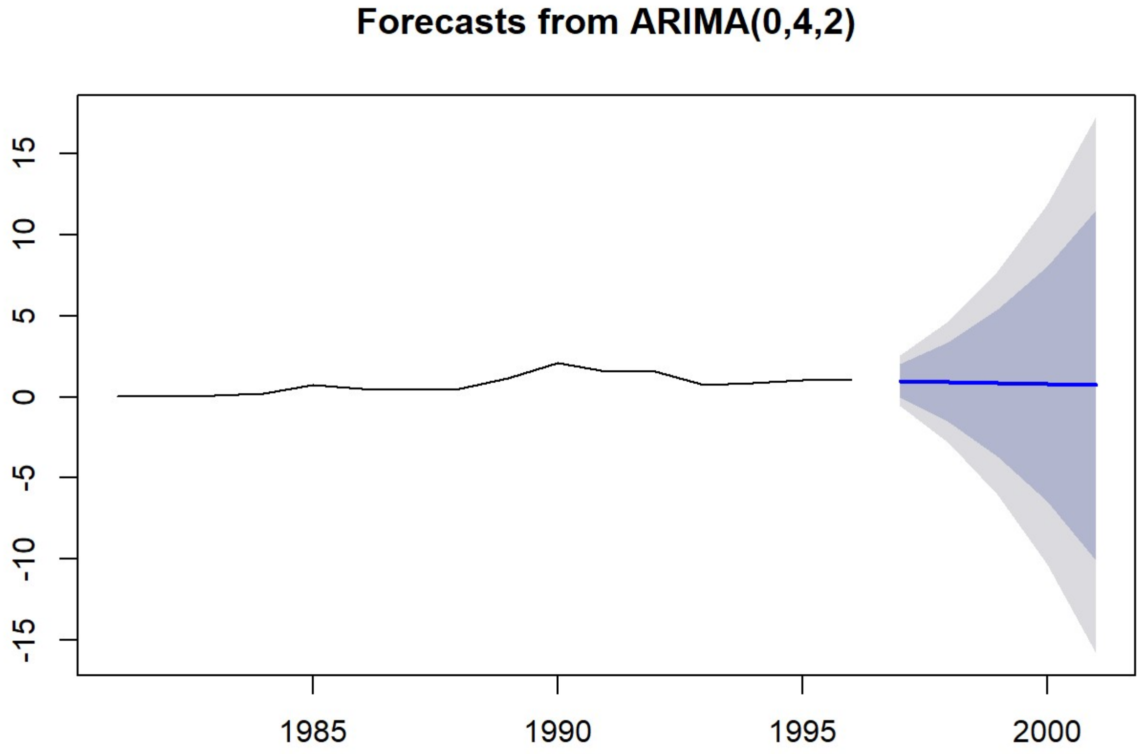


Figure 22.

Reducing the y-limit to see the exact trend in the forecast.

Forecast plot using ARIMA (0,4,2) model with trimmed y-limit

```
fit = Arima(eggs.ts,c(0,4,2))
plot(forecast(fit,h=5) , ylim=c(-5,5))
```

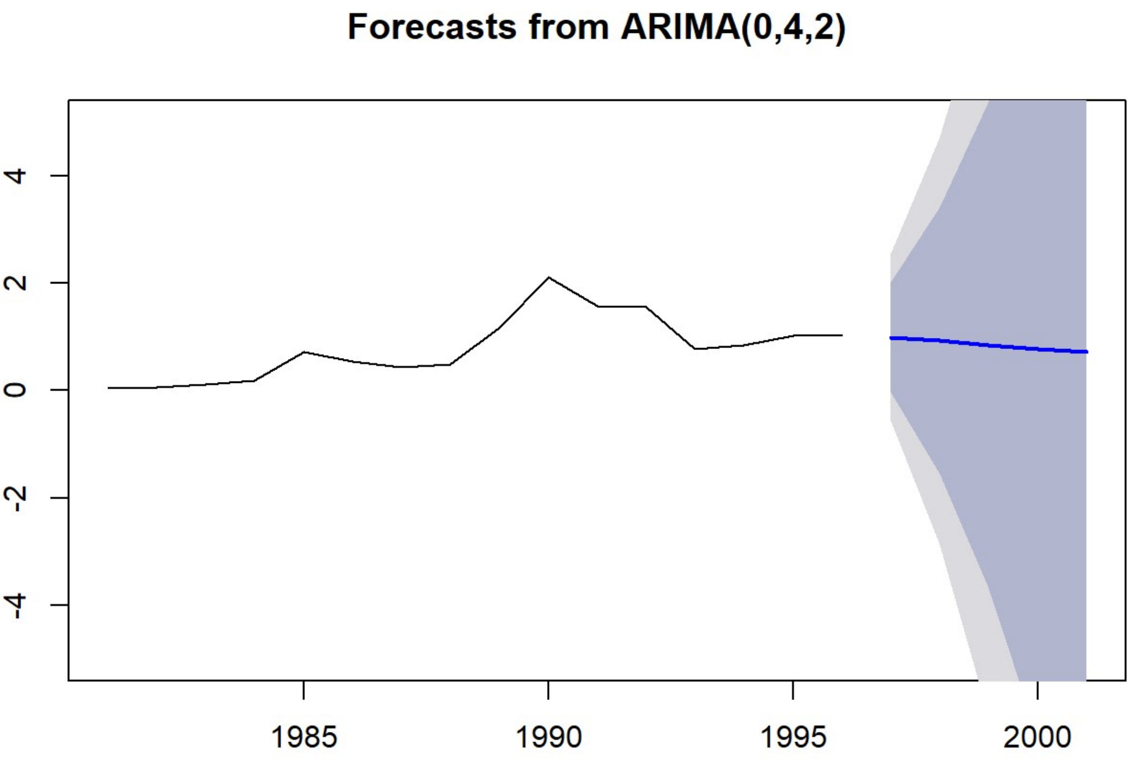


Figure 23.

Conclusion:

Looking a the forecasts it seems that the value of egg depositions is going to follow decreasing trend in coming 5 years.

Summary:

In this task we dealt with the dataset representing egg depositions (in millions) of age-3 Lake Huron Bloaters (Coregonus hoyi) between years 1981 and 1996. The goal was to find the best fitting model to the dataset and give predictions of yearly changes for the next 5 years. Trend, seasonal and ARIMA models were applied and their statistical outputs and the plots were obtained. Based on which the suitable model was picked and taken forward for the diagnostic testing. As seen, trend models were voted out in the initial phase only because they failed the diagnose testing. So we further based our analysis on the ARIMA models. After using various model specification methods and applying various diagnostics tests we chose ARIMA(0,4,2) as the most suitable model out of all proposed candidate models.Our diagnostic phase for chosen ARIMA model involved following tests-

1. Residual analysis
2. Histogram of residuals
3. ACF and PACF plots

4. Shapiro Wilk test
5. Ljung-Box test

ARIMA(0,4,2) successfuly clears all the diagnostic tests (Figure 21.) and comes up as the best fit model for the given dataset of egg deposition for age 3 Lake Huron Bloaters. Finally the forecast is shown for the series for next 5 years (Figure 22. /Figure 23.)

References:

https://stackoverflow.com/questions/36026131/knit-html-recover-from-error (https://stackoverflow.com/questions/36026131/knit-html-recover-from-error)

https://stackoverflow.com/questions/37116632/rmarkdown-html-number-figures (https://stackoverflow.com/questions/37116632/rmarkdown-html-number-figures)

Appendix:

```
library(readr) library(TSA) library(tseries) library(fUnitRoots) library(lmtest) library(knitr) library(forecast)

outputFormat = opts_knit$get(“rmarkdown.pandoc.to”)

capTabNo = 1; capFigNo = 1;

capTab = function(x){ if(outputFormat == ‘html’){ x = paste0(“Table”,capTabNo,“.”,x) capTabNo <<- capTabNo + 1 }; x }

capFig = function(x){ if(outputFormat == ‘html’){ x = paste0(“Figure”,capFigNo,“.”,x) capFigNo <<- capFigNo + 1 }; x }

eggs <- read.csv(“C:/Users/user/Desktop/Namita Chhibba/Semester III/Time Series/Assignment 2/eggs.csv”)

head(eggs)

class(eggs)

eggs.ts <- ts(as.vector(eggs$eggs), start=1981, end= 1996, frequency=1) head(eggs.ts)

class(eggs.ts)

plot(eggs.ts, type=‘o’, ylab=‘egg depositions’)

plot(y=eggs.ts,x=zlag(eggs.ts),ylab=‘eggs deposition’, xlab=‘Previous Year deposition’, main = “Scatter plot of neighboring eggs deposition measurements”)

y = eggs.ts # Read the eggs deposition data into y x = zlag(eggs.ts) # Generate first lag of the series index = 2:length(x) # Create an index to get rid of the first NA value in x cor(y[index],x[index]) # Calculate correlation between numerical values in x and y

model1 =lm(eggs.ts~time(eggs.ts)) summary(model1)

plot(eggs.ts, type=‘o’, ylab=‘y’, main = “Fitted linear trend line to eggs deposition data”) abline(model1)
```

```
t=time(eggs.ts) t2= t^2 model2 =lm(eggs.ts~t+t2) summary(model2)

plot(ts(fitted(model2)), ylim = c(min(c(fitted(model2), as.vector(eggs.ts))), max(c(fitted
(model2),as.vector(eggs.ts)))),ylab=‘y’ , main = “Fitted quadratic curve to eggs deposition data”) lines
(as.vector(eggs.ts),type=“o”)

eggs2.ts <- ts(as.vector(eggs$eggs), start=c(1981,1), end= c(1996,12), frequency=12) head(eggs2.ts)
month.=season(eggs2.ts)

plot(eggs2.ts, type=‘o’, ylab=’’)

model3=lm(eggs2.ts~month.-1) # -1 removes the intercept term summary(model3)
```

To see how this seasonal model fits the data, we plot the model along with series and check the results

```
plot(ts(fitted(model3),freq=12,start=c(1981,1)),ylab=‘eggs deposition (in millions)’,type=‘l’, ylim=range(c
(fitted(model3),eggs2.ts)),main=“Fitted model to eggs deposition series”) # ylim ensures that the y axis
range fits the raw data and the fitted values points(eggs2.ts)

har.=harmonic(eggs2.ts,1) # calculate cos(2pit) and sin(2pit) model4=lm(eggs2.ts~har.) summary
(model4)

res.model1=rstudent(model1) plot(y=res.model1, x=as.vector(time(eggs.ts)), xlab=‘Time’,
ylab=‘Standardized Residuals’, type=‘p’)

res.model2= rstudent(model2) plot(y=res.model2, x=as.vector(time(eggs.ts)), xlab=‘Time’,
ylab=‘Standardized Residuals’, type=‘p’) abline(h=0)

y = rstudent(model1) qqnorm(y) qqline(y, col = 2, lwd = 1, lty = 2)

y = rstudent(model2) qqnorm(y) qqline(y, col = 2, lwd = 1, lty = 2)

y = rstudent(model1) shapiro.test(y)

y = rstudent(model2) shapiro.test(y)

acf(rstudent(model1), main = “ACF of standardized residuals”)

acf(rstudent(model2), main = “ACF of standardized residuals”)

par(mfrow=c(1,2)) acf(eggs.ts) pacf(eggs.ts) par(mfrow=c(1,1))

adf.test(eggs.ts)

eggs.transform <- BoxCox.ar(eggs.ts, method=“yw”, plotit=TRUE) eggs.transform$ci

par(mfrow=c(1,2)) lambda=0.45 # midpoint of interval (0.1, 0.8) eggs.bc= (eggs.ts^lambda-1)/lambda
plot(eggs.ts, type=‘o’, ylab =‘count for Original series’) plot(eggs.bc, type=‘o’, ylab =“count for
transformed series”)

par(mfrow=c(1,1)) eggs.bc.diff1 <- diff(eggs.bc, differences=1) plot(eggs.bc.diff1, ylab=‘Change in eggs
depositions’, type=‘l’)

order=ar(diff(eggs.bc.diff1))$order adfTest(eggs.bc.diff1, lags = order, title = NULL,description = NULL)

par(mfrow=c(1,1)) eggs.bc.diff2 <- diff(eggs.bc, differences=2) plot(eggs.bc.diff2, ylab=‘Change in eggs
```

depositions’, type=‘l’)

order=ar(diff(eggs.bc.diff2))$order adfTest(eggs.bc.diff2, lags = order, title = NULL,description = NULL)

par(mfrow=c(1,1)) eggs.bc.diff3 <- diff(eggs.bc, differences=3) plot(eggs.bc.diff3, ylab=‘Change in eggs depositions’, type=‘l’)

order=ar(diff(eggs.bc.diff3))$order adfTest(eggs.bc.diff3, lags = order, title = NULL,description = NULL)

par(mfrow=c(1,1)) eggs.bc.diff4 <- diff(eggs.bc, differences=4) plot(eggs.bc.diff4, ylab=‘Change in eggs depositions’, type=‘l’)

order=ar(diff(eggs.bc.diff4))$order adfTest(eggs.bc.diff4, lags = order, title = NULL,description = NULL)

par(mfrow=c(1,2)) acf(eggs.bc.diff4) pacf(eggs.bc.diff4) par(mfrow=c(1,1))

eacf(eggs.bc.diff4, ar.max=2, ma.max=2) # We put these arguments to limit the orders p and q at 2. Otherwise, the eacf function returns error and displays nothing.

par(mfrow=c(1,1)) res= armasubsets(y=eggs.bc.diff4, nar=2, nma=3, y.name=‘test’, ar.method=‘ols’) plot(res)

model_041_css = arima(eggs.bc,order=c(0,4,1),method=‘CSS’) coeftest(model_041_css)

model_041_ml = arima(eggs.bc,order=c(0,4,1),method=‘ML’) coeftest(model_041_ml)

model_042_css = arima(eggs.bc,order=c(0,4,2),method=‘CSS’) coeftest(model_042_css)

model_042_ml = arima(eggs.bc,order=c(0,4,2),method=‘ML’) coeftest(model_042_ml)

model_141_css = arima(eggs.bc,order=c(1,4,1),method=‘CSS’) coeftest(model_141_css)

model_141_ml = arima(eggs.bc,order=c(1,4,1),method=‘ML’) coeftest(model_141_ml)

model_142_css = arima(eggs.bc,order=c(1,4,2),method=‘CSS’) coeftest(model_142_css)

model_142_ml = arima(eggs.bc,order=c(1,4,2),method=‘ML’) coeftest(model_142_ml)

model_241_css = arima(eggs.bc,order=c(2,4,1),method=‘CSS’) coeftest(model_241_css)

model_241_ml = arima(eggs.bc,order=c(2,4,1),method=‘ML’) coeftest(model_241_ml)

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”)’) } } sort.score(AIC(model_041_ml,model_042_ml,model_141_ml, model_142_ml, model_241_ml), score = “aic”) sort.score(BIC(model_041_ml,model_042_ml,model_141_ml, model_142_ml, model_241_ml), score = “bic”)

model_043_css = arima(eggs.bc,order=c(0,4,3),method=‘CSS’) coeftest(model_043_css)

model_043_ml = arima(eggs.bc,order=c(0,4,3),method=‘ML’) coeftest(model_043_ml)

residual.analysis <- function(model, std = TRUE,start = 2, class = c(“ARIMA”,“GARCH”,“ARMA-GARCH”)[1]){ # If you have an output from arima() function use class = “ARIMA” # If you have an output from garch() function use class = “GARCH” # If you have an output from ugarchfit() function use class = “ARMA-GARCH” library(TSA) library(FitAR) if (class == “ARIMA”){ if (std == TRUE){ res.model

= rstandard(model) }else{ res.model = residuals(model) } }else if (class == “GARCH”){ res.model = model$residuals[start : model$ n.used] }else if (class == “ARMA-GARCH”){ res.model = model@fit$residuals (mailto:model@fit$residuals) }else { stop(“The argument ‘class’ must be either ‘ARIMA’ or ‘GARCH’”) } par(mfrow=c(3,3)) 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”) acf(res.model,main=“ACF of standardised residuals”) pacf (res.model,main=“PACF of standardised residuals”) qqnorm(res.model,main=“QQ plot of standardised residuals”) qqline(res.model, col = 2) print(shapiro.test(res.model)) k=0 LBQPlot(res.model, lag.max = length(model$residuals)-1, StartLag = k + 1, k = 0, SquaredQ = FALSE) } residual.analysis(model = model_141_ml)

Box.test(residuals(model_042_ml), lag=10, type=“Ljung-Box”)

shapiro.test(residuals(model_042_ml))

fit = Arima(eggs.ts,c(0,4,2)) plot(forecast(fit,h=5), ylim=c(-5,5))