

Time Series Analysis

[Code ▼](#)

Analysis and Forecast of egg depositions of Lake Huron Bloaters

Student Details

- Saurabh Mallik

Introduction

The Lake Huron Bloater (*Coregonus hoyi*), is a species of fish, and it is a form of freshwater white fish. It is found commonly in great lakes. This species is on a decline in lake huron due to predation and pollution.

The purpose of this report is to analyse the Lake Huron Bloater's egg deposition from the eggs series provided in the FSAdat package. The egg series is a subset of the main dataset having two variables years and eggs. The years variable has years ranging from 1981 to 1996. The eggs deposition variable has egg deposition value in millions.

Methodology

To undertake this research, time series analysis and forecasting methods on R Studio are being used to infer from the dataset.

Analysis and Inferences

To check for the nature of the data in the "Lake Huron Bloater Eggs Deposition" dataset, we first load relevant packages and the dataset.

We next load in the data.

[Hide](#)

```
bleggs <- read_csv("eggs.csv")
```

```
Parsed with column specification:
cols(
  year = col_integer(),
  eggs = col_double()
)
```

We check for class of object and convert it to a ts function.

[Hide](#)

```
class(bleggs)
```

```
[1] "tbl_df"      "tbl"        "data.frame"
```

[Hide](#)

```
bleggs.ts <- ts(as.vector(bleggs[,2]), start=1981, end=1996)
class(bleggs.ts)
```

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

We display the eggs time series.

[Hide](#)

```
bleggs.ts
```

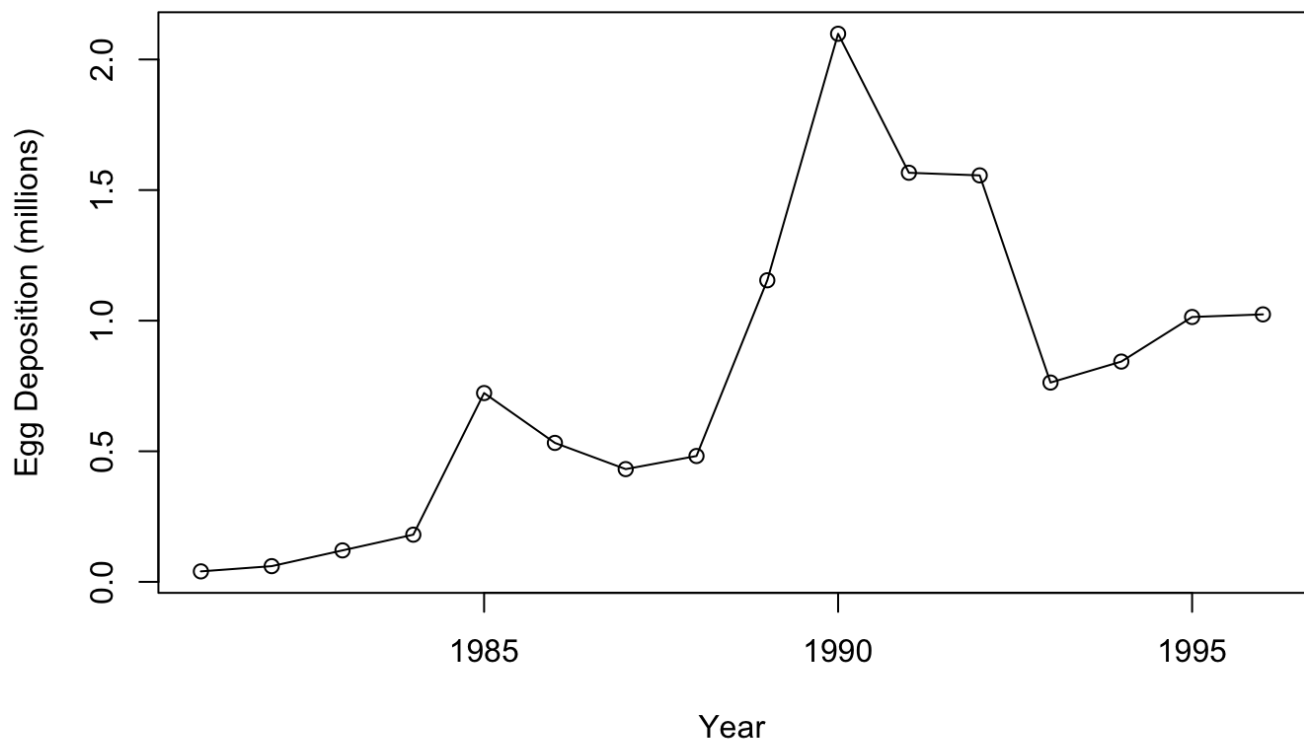
```
Time Series:
Start = 1981
End = 1996
Frequency = 1
      eggs
[1,] 0.0402
[2,] 0.0602
[3,] 0.1205
[4,] 0.1807
[5,] 0.7229
[6,] 0.5321
[7,] 0.4317
[8,] 0.4819
[9,] 1.1546
[10,] 2.0984
[11,] 1.5663
[12,] 1.5562
[13,] 0.7631
[14,] 0.8434
[15,] 1.0141
[16,] 1.0241
```

Next, we plot the time series and check for any insights that we can gain from it.

[Hide](#)

```
plot(bleggs.ts,type='o', main = 'Graph 1. Lake Huron Bloater Eggs Deposition time series', ylab = "Egg Deposition (millions)", xlab = "Year")
```

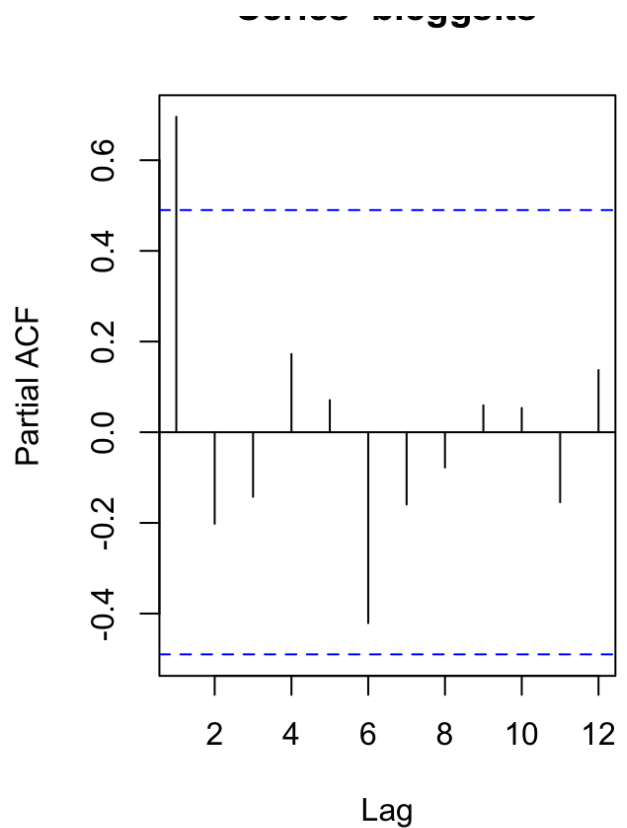
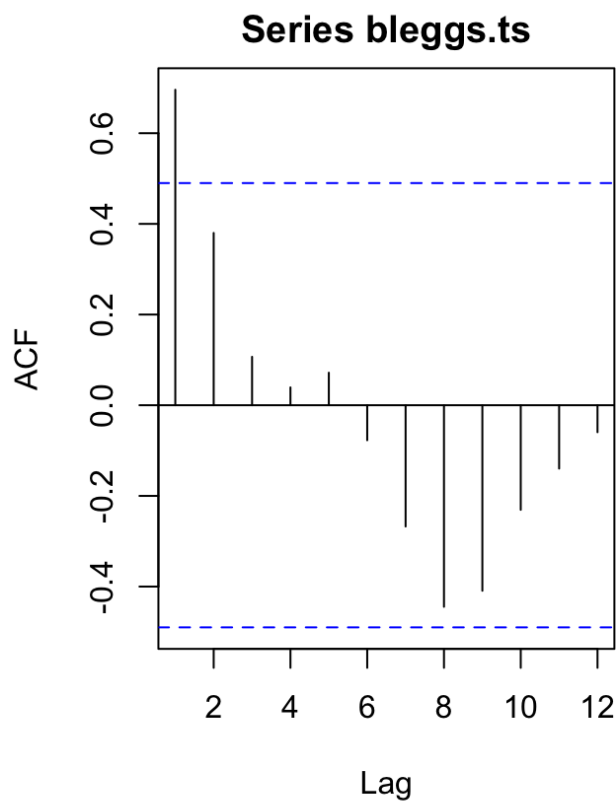
Graph 1. Lake Huron Bloater Eggs Deposition time series



From graph 1, there does not seem to be any evidence of seasonality. The time series shows evidence of autoregressive behaviour, and there are slight hints of changing variance over time. It is also visible that there is a hint of trend and that over time there has been a gradual linear increase in egg depositions by the Lake Huron Bloaters. The egg deposition hit a peak in 1990, then gradually reduced and maintained the upward trend.

Hide

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



The slowly decaying pattern in ACF and very high first correlation in PACF implies the existence of trend and nonstationarity in the series.

We next calculate correlation between one year's egg depositions with that of the successive year's.

Hide

```
y = bleggs.ts
x = zlag(bleggs.ts)
index = 2:length(x)
cor(y[index], x[index])
```

```
[1] 0.7445657
```

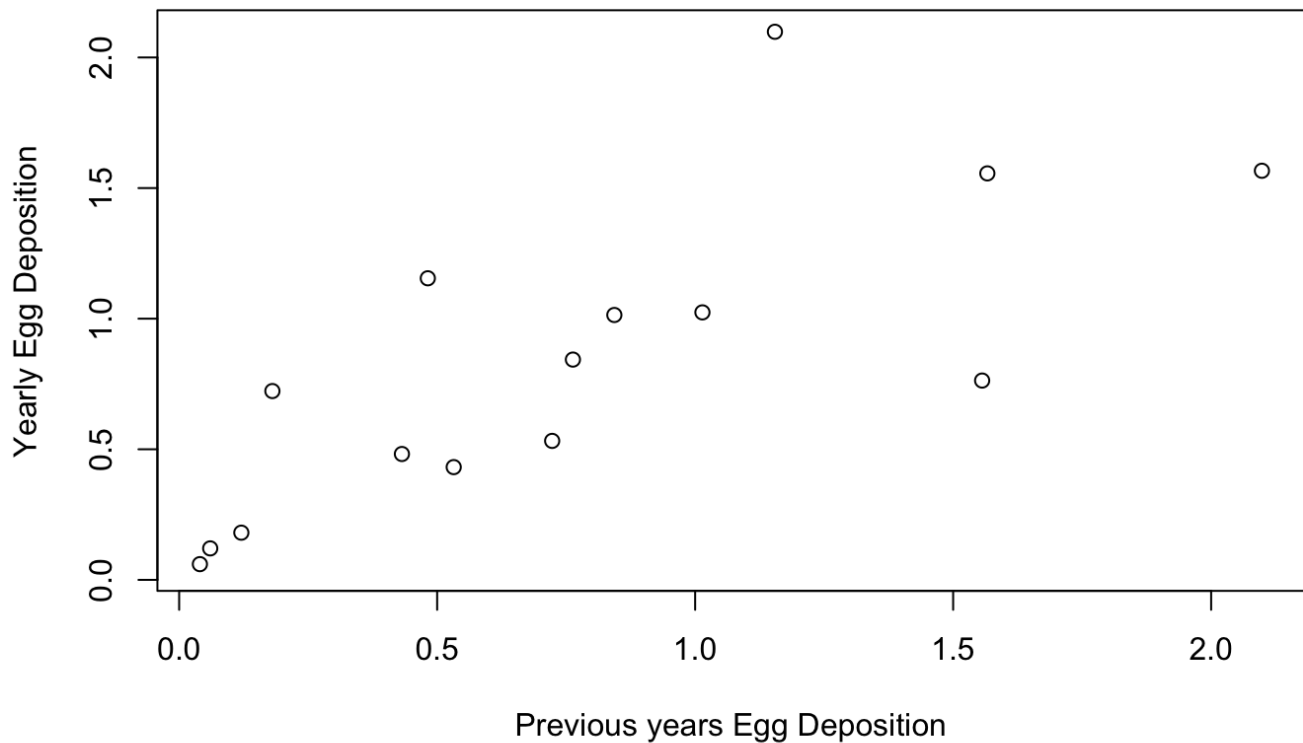
From the above test, we see a 74.45% correlation which is a very strong positive correlation.

We next visualise the correlation with the graph 2., confirming the above correlation test.

Hide

```
plot(y = bleggs.ts, x = zlag(bleggs.ts), ylab = "Yearly Egg Deposition", xlab = "Previous years Egg Deposition", main = "Graph 2. Scatter Plot Comparing Consecutive Year Egg Deposition Figures")
```

Graph 2. Scatter Plot Comparing Consecutive Year Egg Deposition Figure



Next, we start to model fit the data. We commence with a simple linear trend model first, and get a summary for it.

Hide

```
model.lm <- lm(bleggs.ts~time(bleggs.ts))
summary(model.lm)
```

Call:

```
lm(formula = bleggs.ts ~ time(bleggs.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(bleggs.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

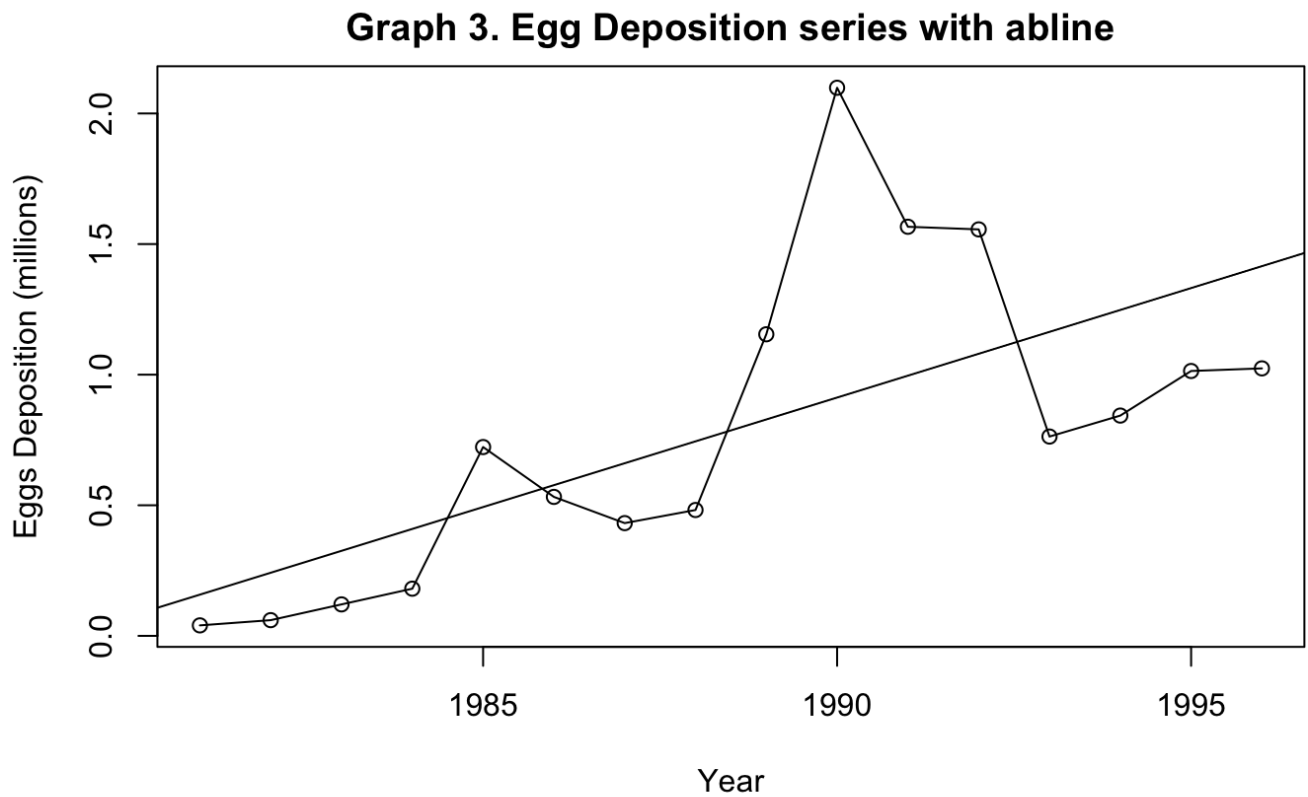
Multiple R-squared: 0.4469, Adjusted R-squared: 0.4074

F-statistic: 11.31 on 1 and 14 DF, p-value: 0.004642

We see that the model is significant, but does not have a good R squared value of .4074 which shows that the model fit is not very good. The coefficients are significant with extremely small p values. We again plot the time series and add an abline.

Hide

```
plot(bleggs.ts,type='o', main = 'Graph 3. Egg Deposition series with abline', ylab =
"Eggs Deposition (millions)", xlab = "Year")
abline(model.lm)
```

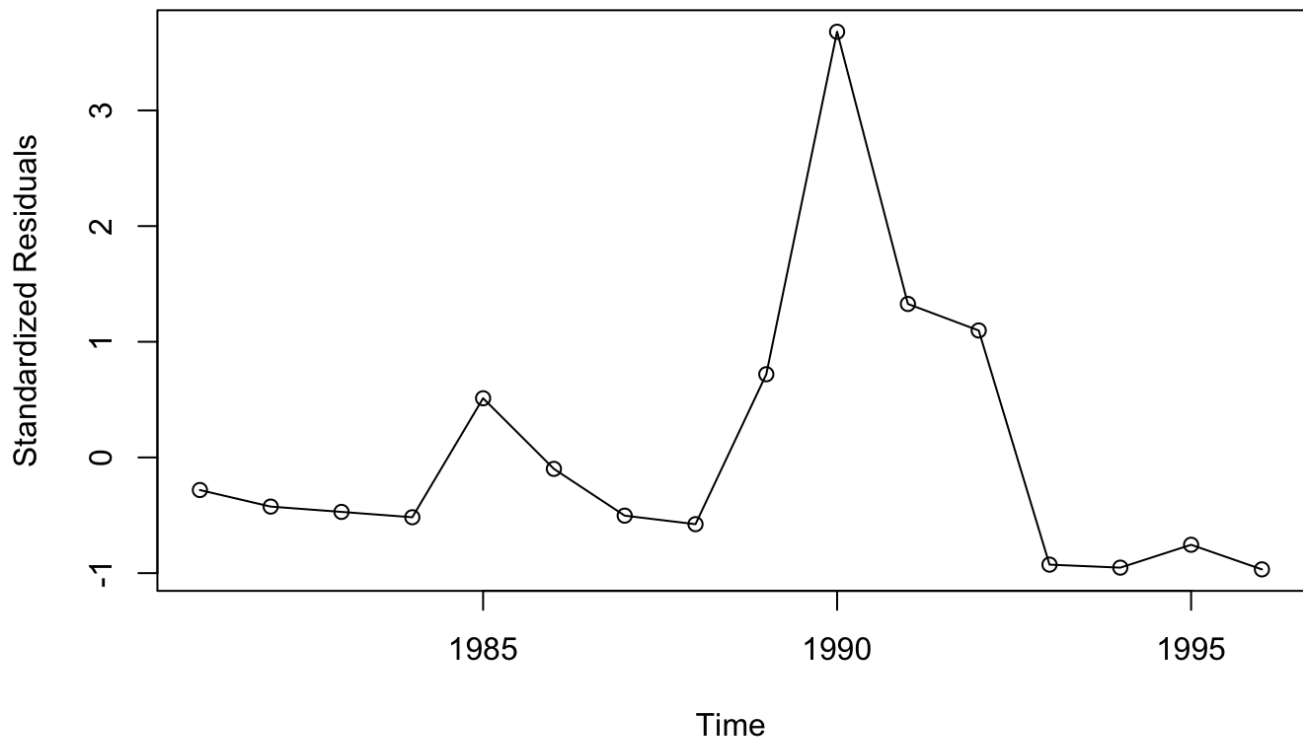


From Graph 3, we can see that the fitted least squares line follows the overall trend in the series. We next check the residuals and plot it to gain further inferences.

Hide

```
res.model.lm = rstudent(model.lm)
plot(y = res.model.lm, x = as.vector(time(bleggs.ts)), xlab = 'Time', ylab = 'Standardized Residuals', type = 'o', main = "Graph 4. Showing standard residuals of model.lm over time")
```

Graph 4. Showing standard residuals of model.lm over time

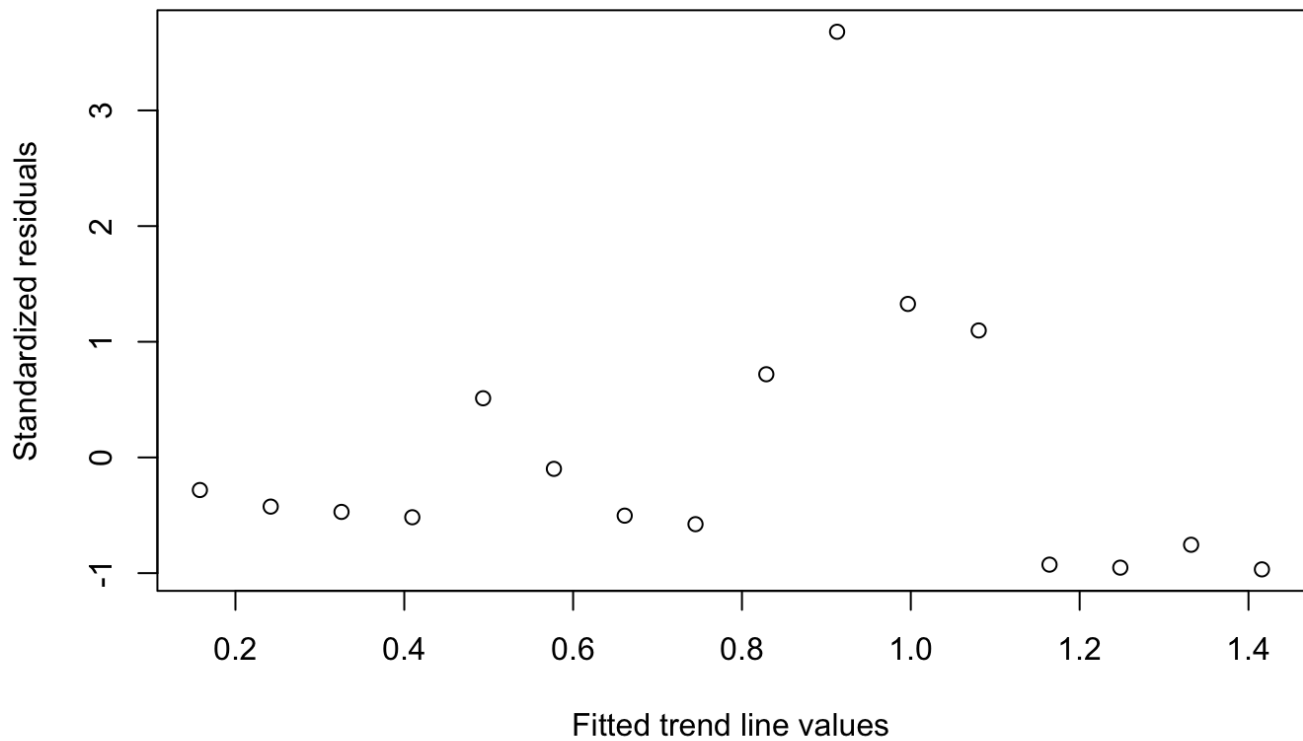


From the residuals plot, we can see that they hang together too much for white noise, the plot is slightly smooth. Also there seems to be more variance towards the last third of the series. We plot the fitted residuals next.

Hide

```
plot(y = rstudent(model.lm), x = fitted(model.lm),  
     ylab = "Standardized residuals",  
     xlab = "Fitted trend line values",  
     main = "Graph 5. Time series plot of the standardized residuals")
```

Graph 5. Time series plot of the standardized residuals

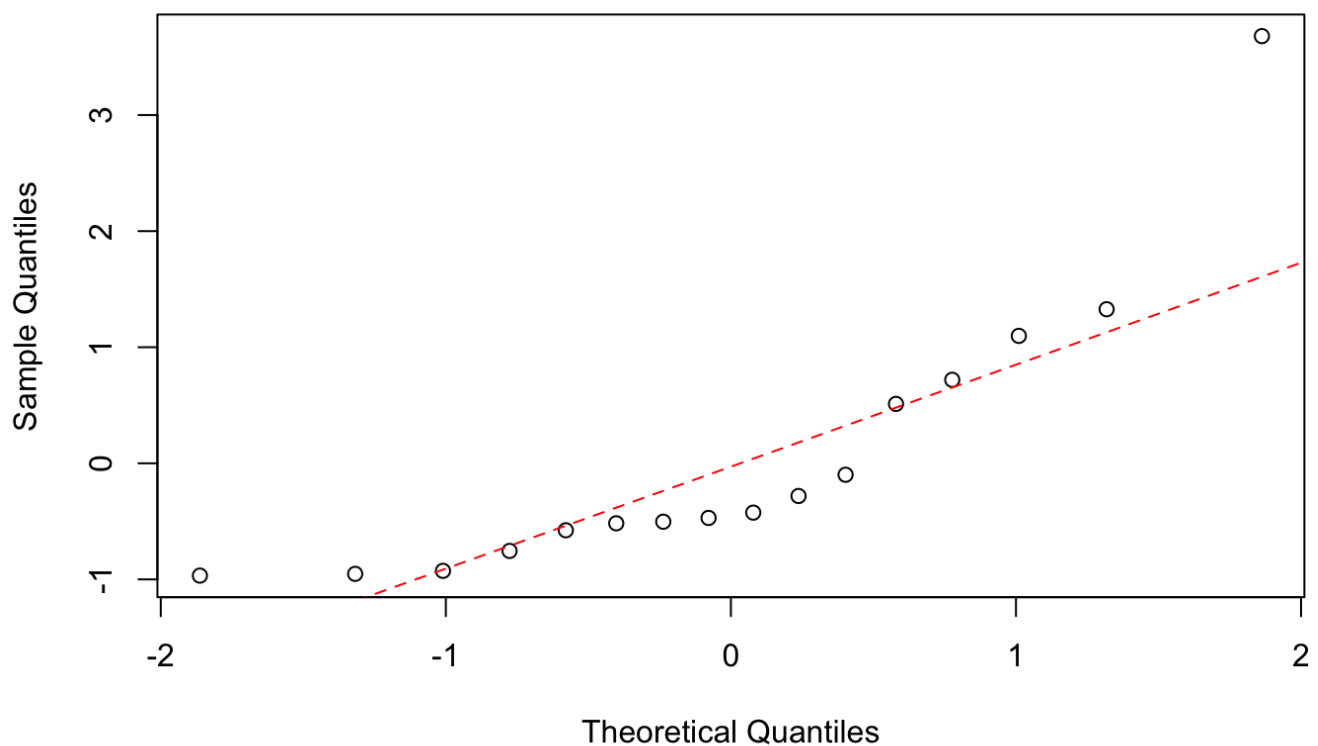


From graph 5, we can see that the points tend to hang together as well.

Hide

```
qqnorm(res.model.lm, main = "Graph 6. Normal Q-Q Plot")
qqline(res.model.lm, col = 2, lwd = 1, lty = 2)
```

Graph 6. Normal Q-Q Plot



Because we see a enough amount of variation away from the reference line, we find evidence that the normality assumption does not hold for the egg deposition dataset.

[Hide](#)

```
shapiro.test(res.model.lm)
```

Shapiro-Wilk normality test

```
data:  res.model.lm
W = 0.7726, p-value = 0.001205
```

The Shapiro-Wilk normality test shows that the model isn't normal.

We next try another model adding a time period, in the form of a quadratic trend model.

[Hide](#)

```
t = time(bleggs.ts)
t2 = t^2
model.quad = lm(bleggs.ts ~ t + t2)
summary(model.quad)
```

Call:

```
lm(formula = bleggs.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

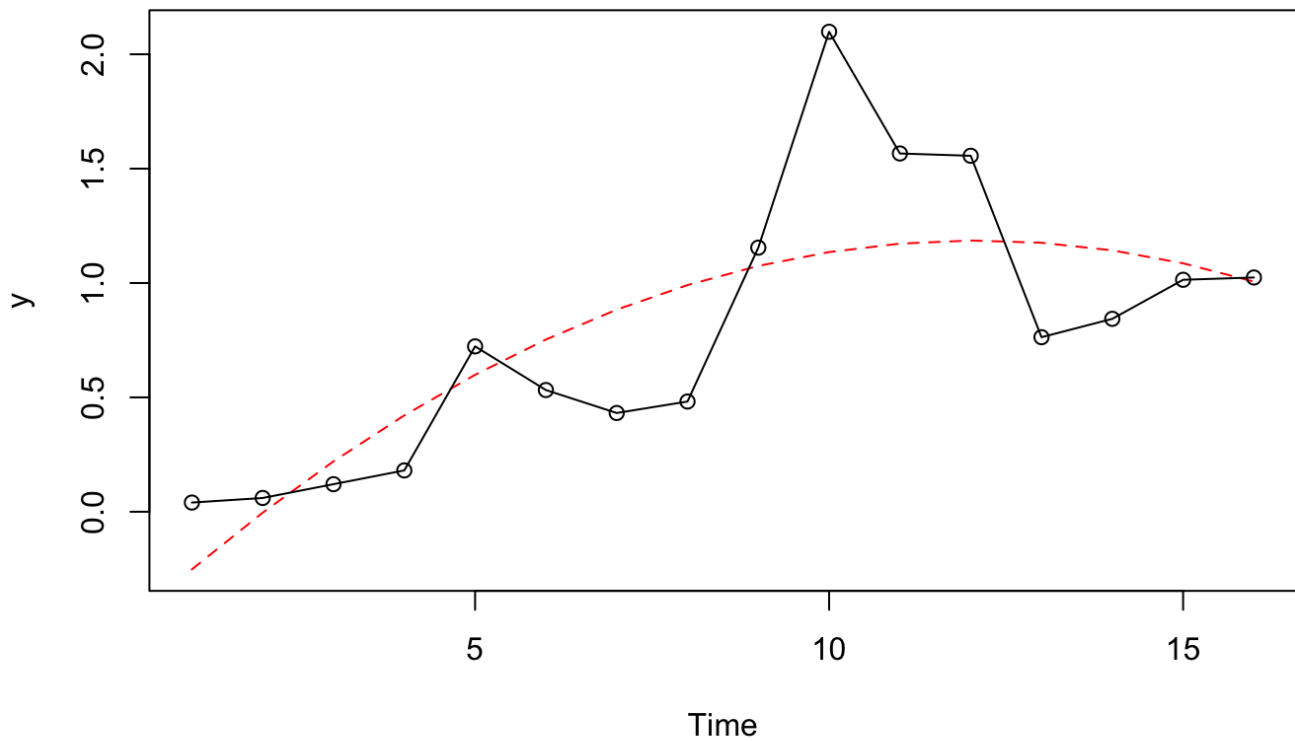
F-statistic: 9.479 on 2 and 13 DF, p-value: 0.00289

We see that the R squared value of the quadratic model is slightly better than the linear trend model at .5306, but it is still not very good. Although a semi decent fit, this model would not serve well for forecasting. We see that the trend coefficients are all significant as well. We proceed to diagnostic checking for the model.

[Hide](#)

```
plot(ts(fitted(model.quad)), ylim = c(min(c(fitted(model.quad),
as.vector(bleggs.ts))), max(c(fitted(model.quad),as.vector(bleggs.ts)))),
ylab='y' , main = "Graph 7. Fitted quadratic curve to egg deposition data", type=
"l",lty=2,col="red")
lines(as.vector(bleggs.ts),type="o")
```

Graph 7. Fitted quadratic curve to egg deposition data

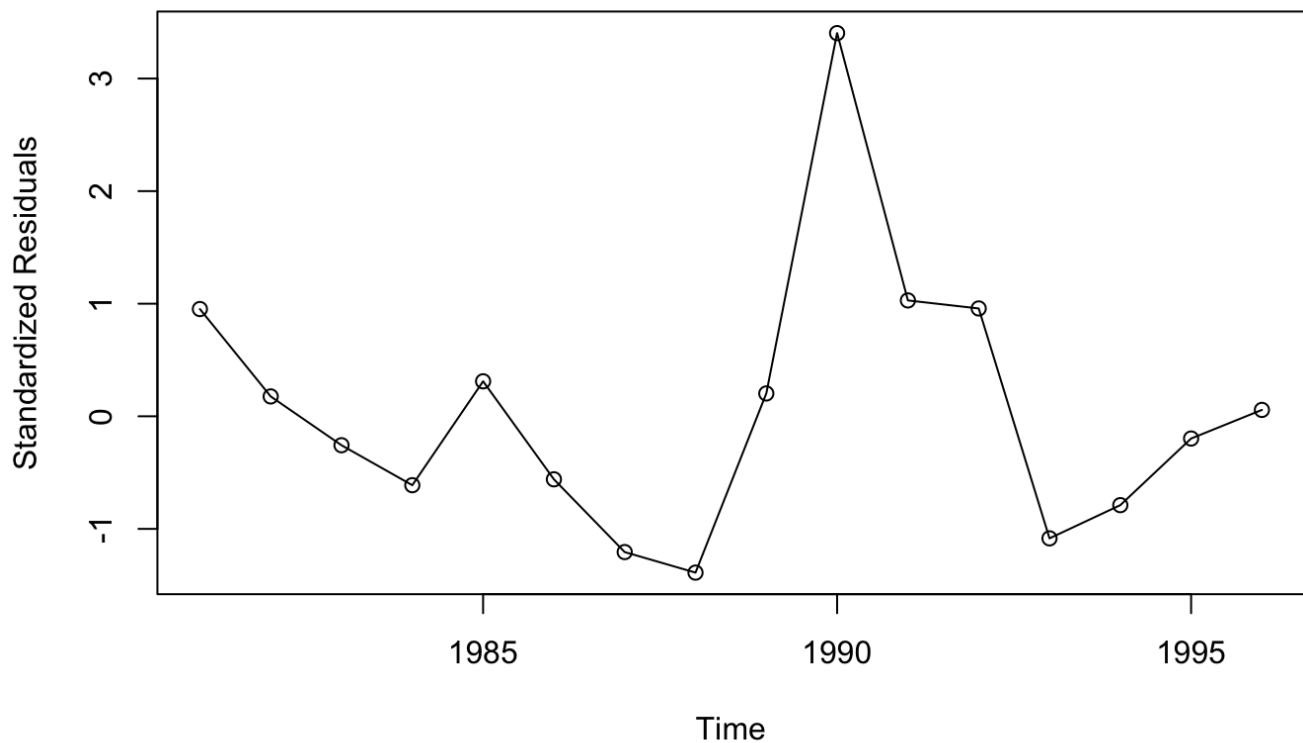


We can see the trend line is fitting the data decently, but not ideally. We next undergo diagnostic testing for this model.

Hide

```
res.model.quad = rstudent(model.quad)
plot(y = res.model.quad, x = as.vector(time(bleggs.ts)), xlab = 'Time', ylab = 'Standard
ized Residuals', type = 'o', main = "Graph 8. Standard residual model plot")
```

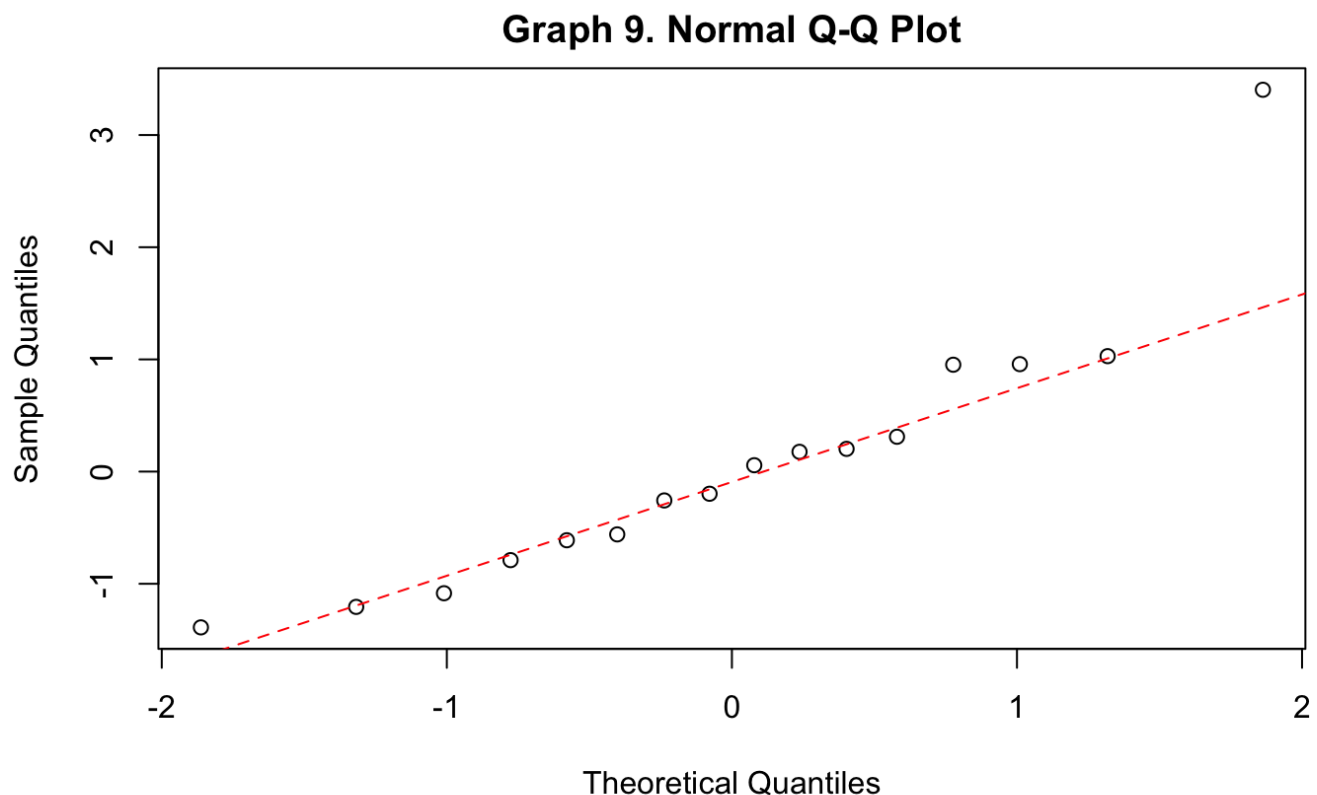
Graph 8. Standard residual model plot



We see that the values are still bunching together, but not as much as the previous model. We continue to do a normality test.

Hide

```
qqnorm(res.model.quad, main = "Graph 9. Normal Q-Q Plot")
qqline(res.model.quad, col = 2, lwd = 1, lty = 2)
```



The values are still moving away from the reference line.

Hide

```
shapiro.test(res.model.quad)
```

Shapiro-Wilk normality test

```
data:  res.model.quad
W = 0.87948, p-value = 0.03809
```

This model also has an insignificant p-value for the Shapiro-Wilk normality test.

For a non ARIMA model, we would choose the quadratic model as it had a higher R squared value. Forecast with this we see the following.

Hide

```
t = c(1997, 1998, 1999, 2000, 2001)
t2 = t^2
new = data.frame(t,t2)
forecasts = predict(model.quad, new, interval = "prediction")
print(forecasts)
```

	fit	lwr	upr
1	0.9025896	-0.2603314	2.065511
2	0.7756872	-0.5240481	2.075422
3	0.6253658	-0.8469331	2.097665
4	0.4516257	-1.2277437	2.130995
5	0.2544667	-1.6647222	2.173656

Hide

```
plot(bleggs.ts, main = "Graph 10. Forecasts of next 5 years of Egg Deposition (millions)", type = "o", xlim = c(1980, 2005), ylim = c(-2.0, 4.0))
lines(ts(as.vector(forecasts[,1]), start = 1997), col="red", type="l")
```

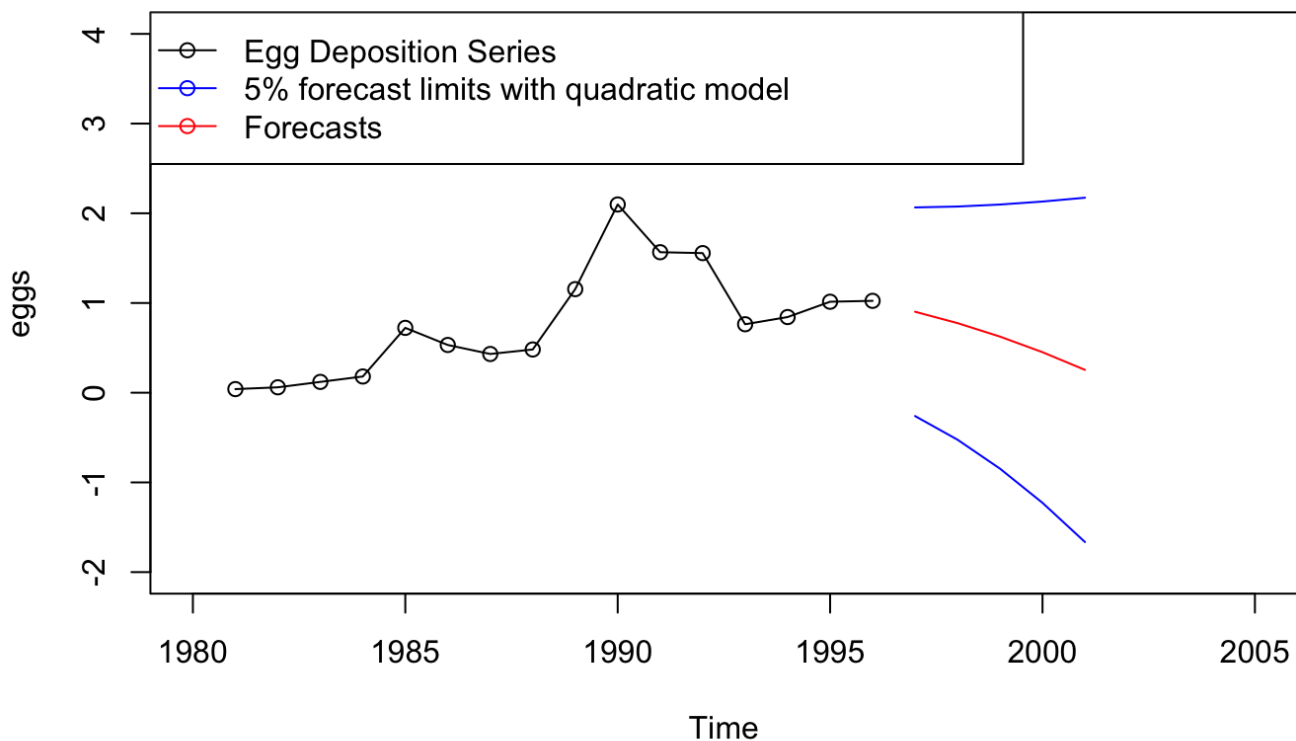
Hide

```
lines(ts(as.vector(forecasts[,2]), start = 1997), col="blue", type="l")
lines(ts(as.vector(forecasts[,3]), start = 1997), col="blue", type="l")
```

Hide

```
legend("topleft", lty=1, pch=1, col=c("black","blue","red"), text.width = 18,
      c("Egg Deposition Series", "5% forecast limits with quadratic model", "Forecast s"))
```

Graph 10. Forecasts of next 5 years of Egg Deposition (millions)



We next check to see whether the series is stationary.

Hide

```
adf.test(bleggs.ts)
```

Augmented Dickey-Fuller Test

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

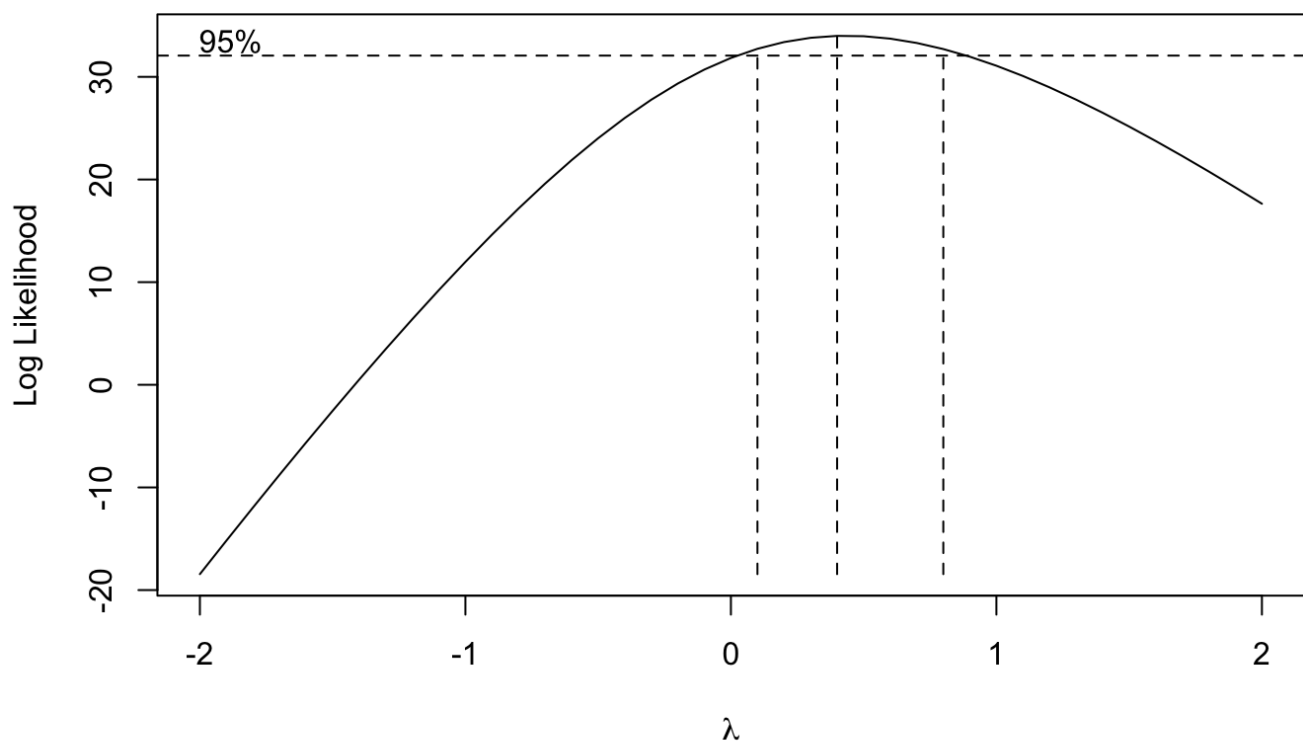
The value of p is insignificant, and hence we need to transform the series or difference it and make it stationary.

In order to make the series stationary and test out further ARIMA model fitting, we first use the first try transforming.

Hide

```
BC.eggs = BoxCox.ar(bleggs.ts, method = "yule-walker", main = "Graph 11. BC Transform  
ation")
```

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



Hide

```
BC.eggs$ci
```

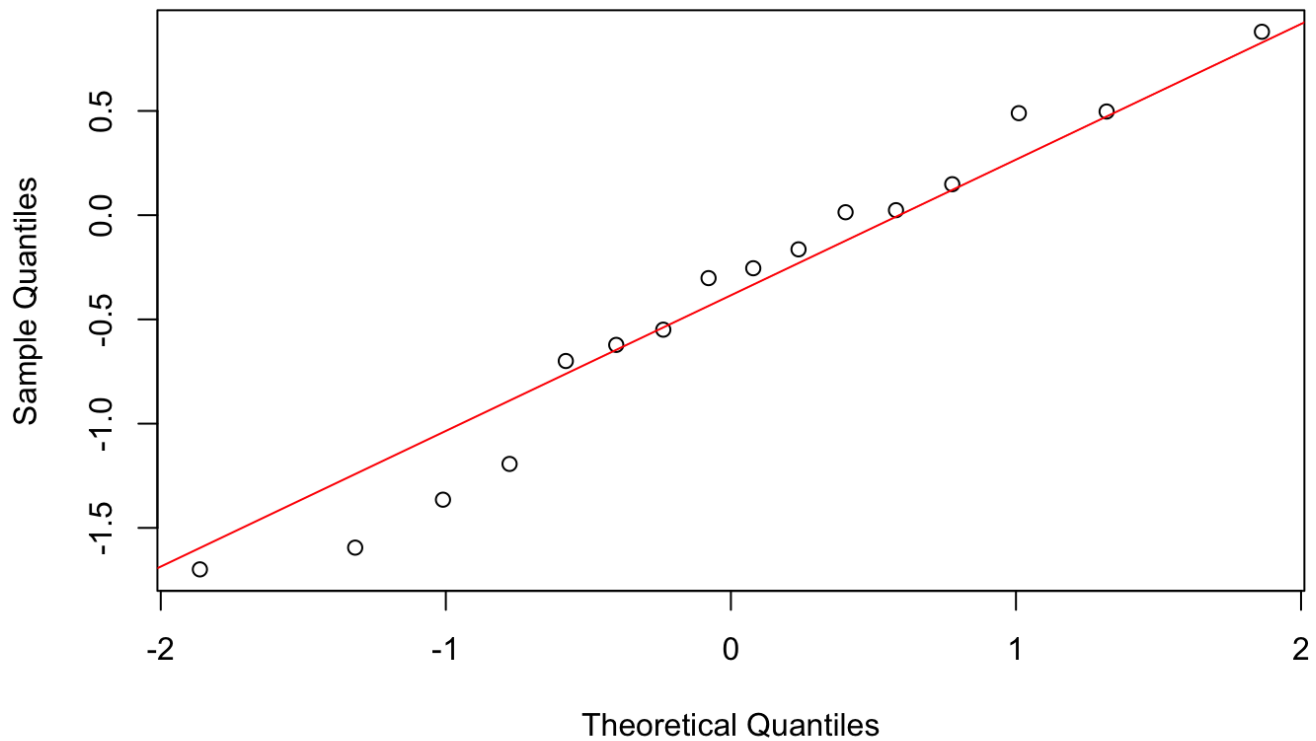
```
[1] 0.1 0.8
```

Mid-point of interval is 0.45. So we will take lambda as 0.45

Hide

```
lambda = 0.45
BC.eggs = (bleggs.ts^lambda-1)/lambda
qqnorm(BC.eggs, main = "Graph 12. Normal Q-Q Plot")
qqline(BC.eggs, col = 2)
```

Graph 12. Normal Q-Q Plot



We see that the data is not normal. We apply ADF unitroot test to the transformed series

Hide

```
adf.test(BC.eggs)
```

Augmented Dickey-Fuller Test

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

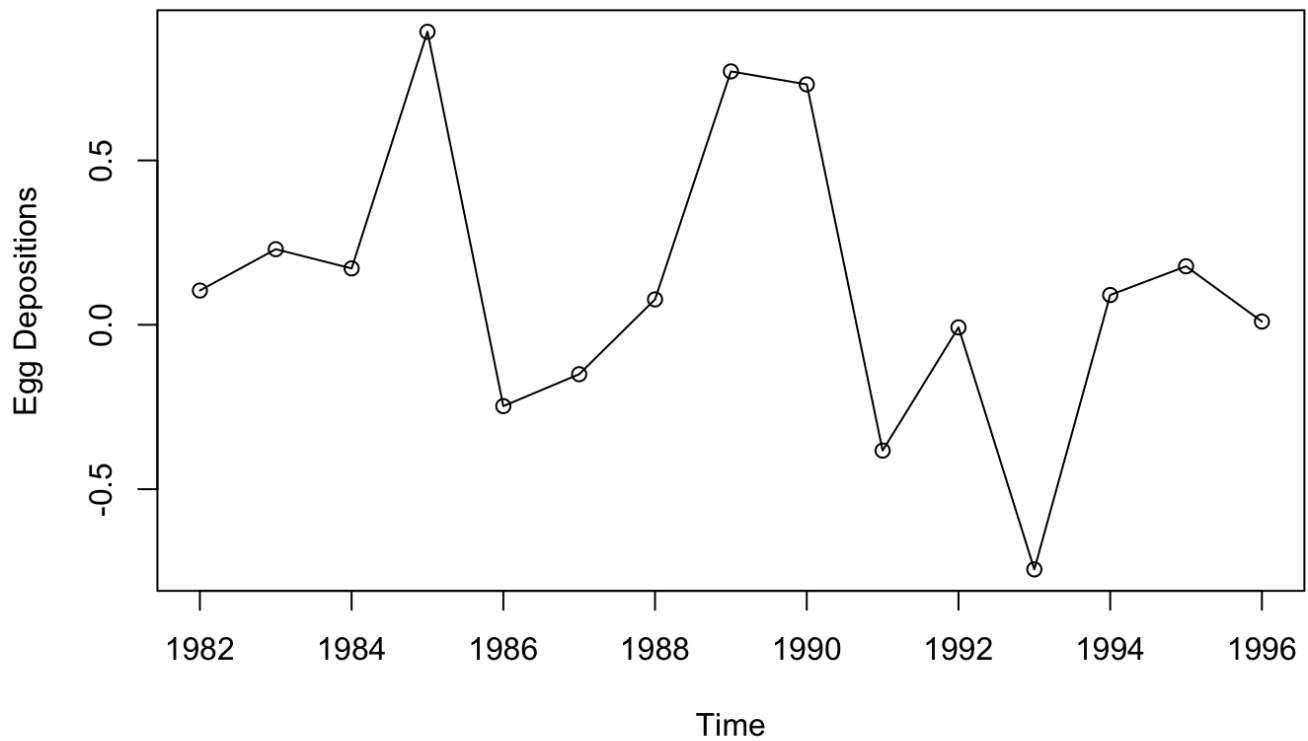
The transformed series is still not stationary.

We next move to differencing.

Hide

```
diff.eggs <- diff(BC.eggs)
plot(diff.eggs,type='o',ylab='Egg Depositions', main = "Graph 13. Differenced Series"
)
```

Graph 13. Differenced Series



After applying the the first difference the series became detrended and stationary. Now there is some changing variance come to the light.

We apply ADF unitroot test to the differenced series.

Hide

```
adf.test(diff.eggs)
```

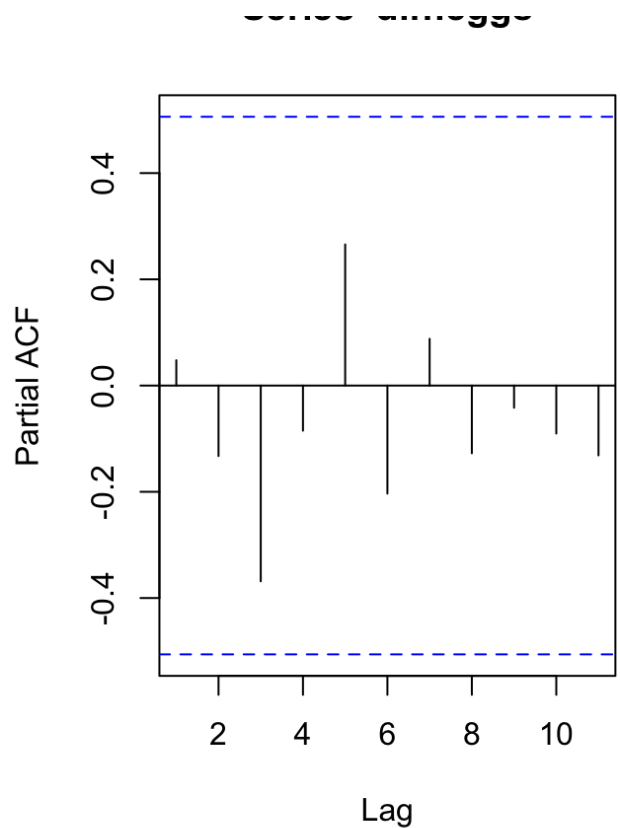
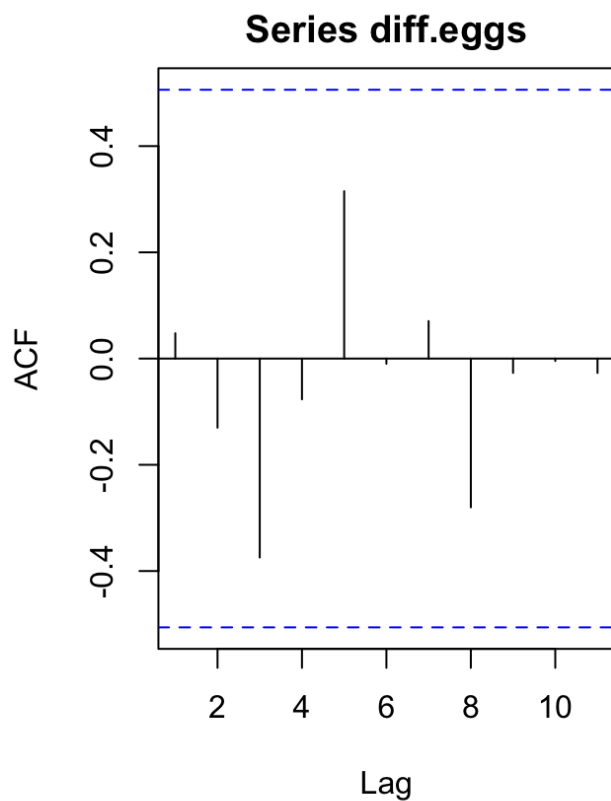
Augmented Dickey-Fuller Test

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

ADF test confirms with the p-value of 0.0443 that the series is stationary with the first differencing.

Hide

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



There is no significant lags in ACF or PACF hence we can include ARIMA(0,1,0) or the basic linear trend or quadratic models as suitable candidates for modelling.

We next check for EACF.

Hide

```
eacf(diff.eggs,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
```

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

Hide

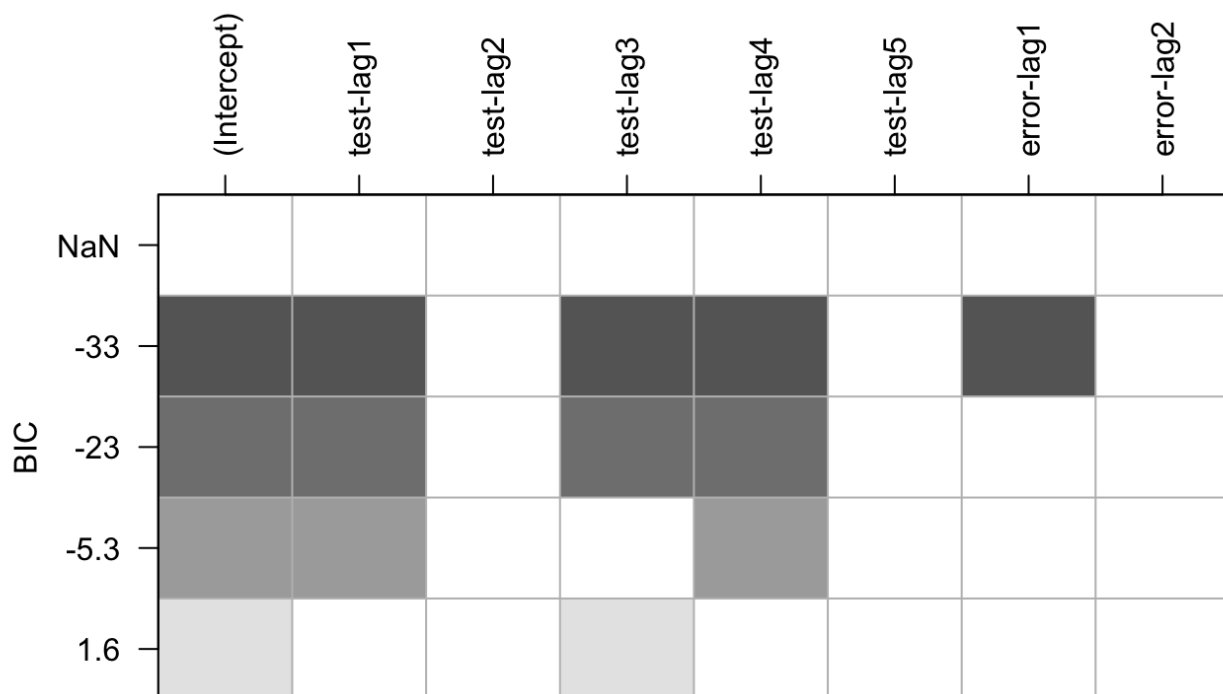
```
par(mfrow=c(1,1))
res = armasubsets(y=diff.eggs,nar=5,nma=2,y.name='test',ar.method='ols')
```

```
model order: 8 singularities in the computation of the projection matrix results are
only valid up to model order 72 linear dependencies foundnvmx reduced to 5
```

Hide

```
plot(res)
```


NaNs producedNaNs producedNaNs produced



In the BIC table, shaded columns correspond to AR(1), AR(3), AR(4), MA(1) coefficients and from here we can include ARIMA(1,1,0), ARIMA(3,1,0), ARIMA(1,1,1), ARIMA(3,1,1) and ARIMA(4,1,0) models as the other combinations give large models.

The final set of possible models is {ARIMA(1,1,0), ARIMA(0,1,1), ARIMA(3,1,0), ARIMA(1,1,1), ARIMA(3,1,1), ARIMA(4,1,0)}

Now we move on to coefficient testing for the models.

ARIMA(1,1,0)

Hide

```
model_110_css = arima(diff.eggs,order=c(1,1,0),method='CSS')
coeftest(model_110_css)
```

z test of coefficients:

```
      Estimate Std. Error z value Pr(>|z|)
ar1 -0.40609    0.24460  -1.6602  0.09687 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The intercept is not significant for 110 css model.

Hide

```
model_110_ml = arima(diff.eggs,order=c(1,1,0),method='ML')
coeftest(model_110_ml)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.38056	0.23493	-1.6199	0.1053

The intercept is not significant for 110 ml model.

ARIMA(0,1,1)

Hide

```
model_011_css = arima(diff.eggs,order=c(0,1,1),method='CSS')
coeftest(model_110_css)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.40609	0.24460	-1.6602	0.09687 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

The intercept is not significant for 011 css model.

Hide

```
model_011_ml = arima(diff.eggs,order=c(0,1,1),method='ML')
coeftest(model_011_ml)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ma1	-0.99994	0.66766	-1.4977	0.1342

The coefficient is insignificant for ml model.

ARIMA(3,1,0)

Hide

```
model_310_css = arima(diff.eggs,order=c(3,1,0),method='CSS')
coeftest(model_310_css)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.53415	0.21650	-2.4672	0.01362 *
ar2	-0.41735	0.24223	-1.7229	0.08490 .
ar3	-0.51729	0.23907	-2.1637	0.03049 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Two of the AR coefficients are significant in the css model.

[Hide](#)

```
model_310_ml = arima(diff.eggs,order=c(3,1,0),method='ML')
coefstest(model_310_ml)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.51836	0.22606	-2.2931	0.02184 *
ar2	-0.36849	0.24360	-1.5127	0.13036
ar3	-0.47004	0.23356	-2.0125	0.04417 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Two of the AR coefficients are significant for the ml model.

ARIMA(1,1,1)

[Hide](#)

```
model_111_css = arima(diff.eggs,order=c(1,1,1),method='CSS')
coefstest(model_111_css)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	0.030947	0.290460	0.1065	0.9152
ma1	-1.021947	0.201926	-5.0610	4.171e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

1 significant AR and MA in this model.

[Hide](#)

```
model_111_ml = arima(diff.eggs,order=c(1,1,1),method='ML')
coefstest(model_111_ml)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	0.11474	0.26992	0.4251	0.670764
ma1	-1.00000	0.33205	-3.0116	0.002599 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

No significant coefficients in this model.

ARIMA(3,1,1)

[Hide](#)

```
model_311_css = arima(diff.eggs,order=c(3,1,1),method='CSS')
coeftest(model_311_css)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.35630	0.27597	-1.2911	0.1967
ar2	-0.36505	0.23983	-1.5221	0.1280
ar3	-0.58811	0.23679	-2.4837	0.0130 *
ma1	-0.33814	0.34765	-0.9727	0.3307

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Only one significant coefficient AR3

Hide

```
model_311_ml = arima(diff.eggs,order=c(3,1,1),method='ML')
coeftest(model_311_ml)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.13108	0.29378	-0.4462	0.65546
ar2	-0.19970	0.24973	-0.7997	0.42390
ar3	-0.41663	0.24785	-1.6810	0.09276 .
ma1	-0.65245	0.30355	-2.1494	0.03160 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Few significant coefficients in AR2, and MA1 and AR3 almost significant.

ARIMA(4,1,0)

Hide

```
model_410_css = arima(diff.eggs,order=c(4,1,0),method='CSS')
coeftest(model_410_css)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)
ar1	-0.67298	0.18629	-3.6126	0.0003032 ***
ar2	-0.53298	0.18086	-2.9469	0.0032097 **
ar3	-0.74639	0.19086	-3.9107	9.202e-05 ***
ar4	-0.58294	0.19646	-2.9672	0.0030054 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

All coefficients significant.

Hide

```
model_410_ml = arima(diff.eggs,order=c(4,1,0),method='ML')
coeftest(model_410_ml)
```

z test of coefficients:

```
      Estimate Std. Error z value  Pr(>|z|)
ar1 -0.79683    0.19268 -4.1355 3.542e-05 ***
ar2 -0.55356    0.21190 -2.6124 0.0089913 **
ar3 -0.70979    0.19964 -3.5553 0.0003775 ***
ar4 -0.62789    0.21213 -2.9600 0.0030769 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All Coefficients significant.

The models with all coefficients significant at 5% are ARIMA(4,1,0) - CSS and ARIMA(4,1,0) - ML

Hide

```
sort.score(AIC(model_410_ml,model_011_ml, model_310_ml, model_311_ml,model_110_ml), s
core = "aic")
```

	df <dbl>	AIC <dbl>
model_011_ml	2	23.12446
model_410_ml	5	24.59229
model_311_ml	5	26.92807
model_110_ml	2	27.05798
model_310_ml	4	27.58888

5 rows

Hide

```
sort.score(BIC(model_410_ml,model_011_ml, model_310_ml, model_311_ml,model_110_ml), s
core = "bic" )
```

	df <dbl>	BIC <dbl>
model_011_ml	2	24.40258
model_410_ml	5	27.78757
model_110_ml	2	28.33609
model_311_ml	5	30.12336
model_310_ml	4	30.14511

5 rows

Both AIC and BIC models select ARIMA(0,1,1) as the best fit, however the coefficient was not significant. Hence, we use the next best model ARIMA(4,1,0).

We will try overfitting with ARIMA(4,1,1) and ARIMA (3,1,0) models.

In ARIMA(3,1,0) overfit models, we already saw that in both css and ml case, the ar2 becomes insignificant and hence we drop this model.

ARIMA(4,1,2)

Hide

```
model_412_css = arima(diff.eggs,order=c(4,1,2),method='CSS')
```

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

Hide

```
coeftest(model_412_css)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
ar1	-0.3598750	0.0053844	-66.8371	< 2.2e-16	***
ar2	-0.6419774	0.0080000	-80.2473	< 2.2e-16	***
ar3	-0.5515864	0.0913031	-6.0413	1.529e-09	***
ar4	-0.2670896	0.0800436	-3.3368	0.0008475	***
ma1	-0.6484179	0.4821119	-1.3450	0.1786404	
ma2	3.0443972	0.7689431	3.9592	7.520e-05	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

In ARIMA(0,1,2) ovefit, we sett that ma2 becomes insignificant.

Hide

```
model_012_ml = arima(diff.eggs,order=c(4,1,2),method='ML')  
coeftest(model_012_ml)
```

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
ar1	-1.34011	0.18849	-7.1098	1.162e-12	***
ar2	-0.96820	0.30917	-3.1316	0.001739	**
ar3	-0.92126	0.28112	-3.2771	0.001049	**
ar4	-0.76217	0.15272	-4.9907	6.016e-07	***
ma1	1.93876	1.58709	1.2216	0.221866	
ma2	0.99527	1.60001	0.6220	0.533918	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

In both cases - ma1 becomes insignificant, and in css ma2 also is insignificant.

Hence, we will stop at ARIMA(4,1,0) model and proceed to residual analysis.

[Hide](#)

```
residual.analysis(model = model_410_ml)
```

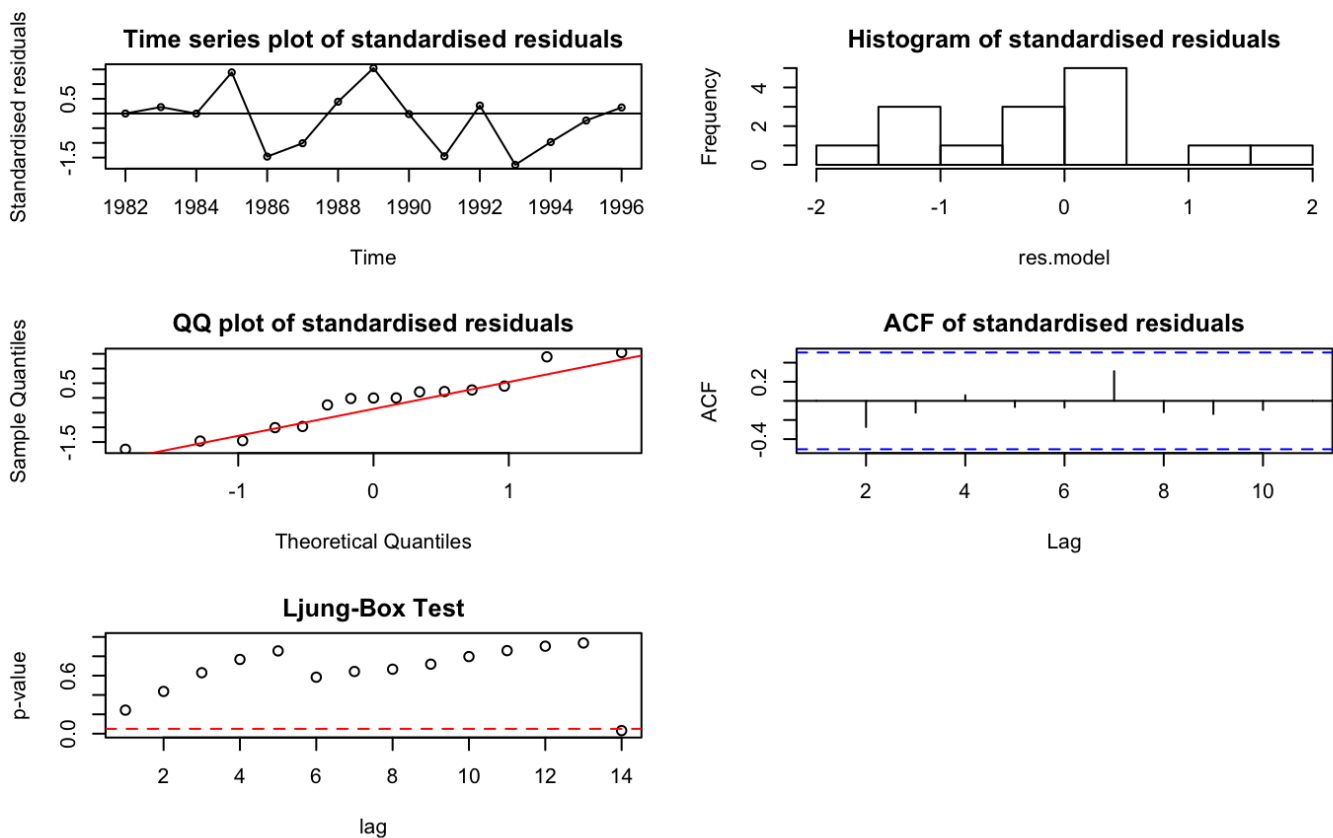
Shapiro-Wilk normality test

```
data: res.model
```

```
W = 0.93498, p-value = 0.3234
```

[Hide](#)

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

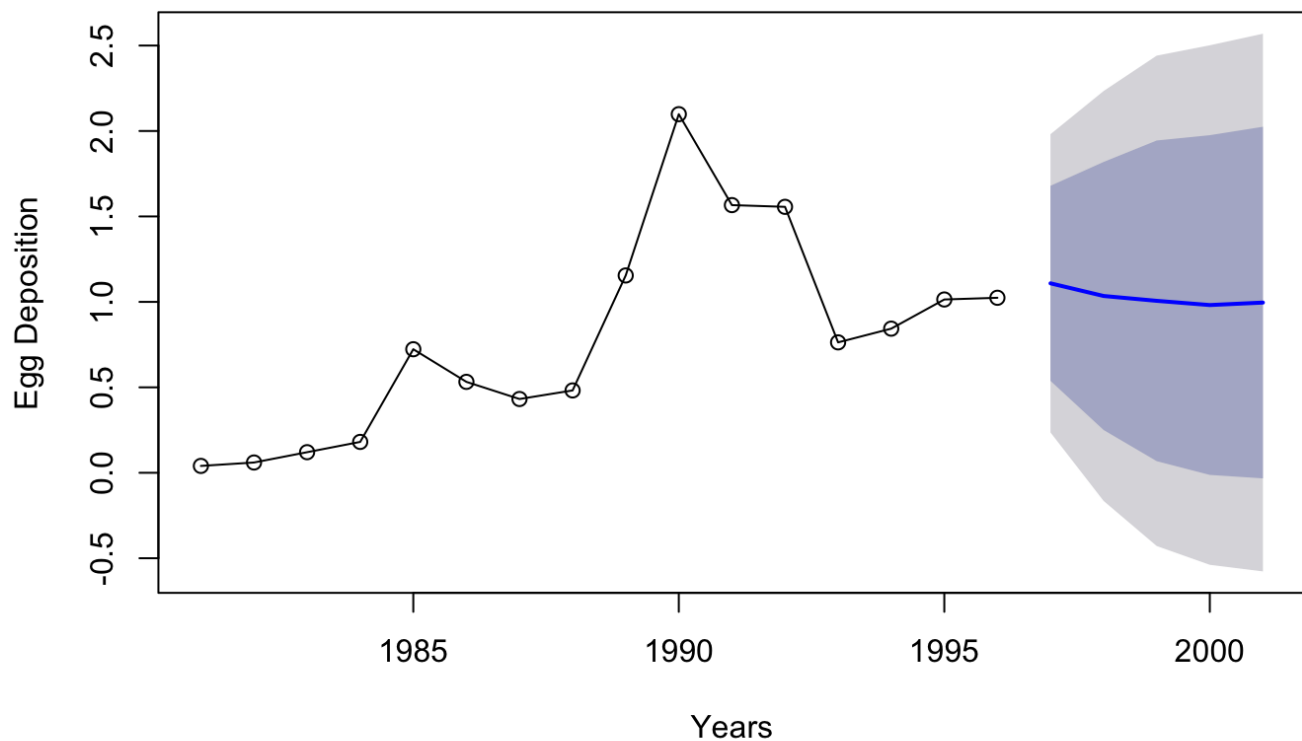


There appear to be no problems with the residual plots. No significant lags in the ACF as well.

[Hide](#)

```
fit = Arima(bleggs.ts,c(4,1,0))
plot(forecast(fit,h=5), type = "o", ylab = "Egg Deposition", xlab = "Years", main =
"Graph 14. 5 year ahead forecast of Egg Deposition Series")
```

Graph 14. 5 year ahead forecast of Egg Deposition Series



Conclusion

In case of a trend model, we chose the quadratic trend model, as it had a higher R squared value than the linear trend model, hence, we forecast with that.

Hide

```
plot(bleggs.ts, main = "Graph 15. Forecasts of next 5 years of Egg Deposition (millions)", type = "o", xlim = c(1980, 2005), ylim = c(-2.0, 4.0))
lines(ts(as.vector(forecasts[,1]), start = 1997), col="red", type="l")
```

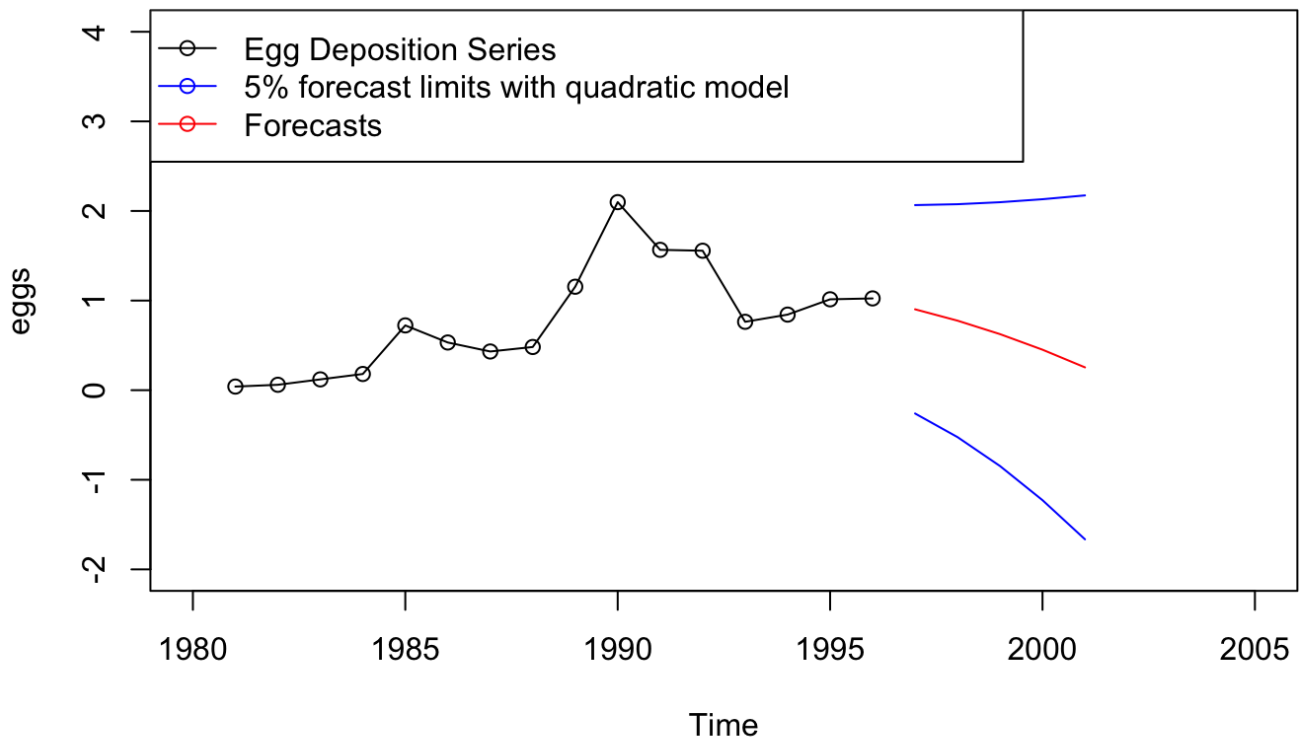
Hide

```
lines(ts(as.vector(forecasts[,2]), start = 1997), col="blue", type="l")
lines(ts(as.vector(forecasts[,3]), start = 1997), col="blue", type="l")
```

Hide

```
legend("topleft", lty=1, pch=1, col=c("black","blue","red"), text.width = 18,
      c("Egg Deposition Series","5% forecast limits with quadratic model", "Forecasts"))
```


Graph 15. Forecasts of next 5 years of Egg Deposition (millions)



As this model was not a very good fit, we chose the ARIMA approach. The results from analysis led us to the following conclusion.

After various model fitting and coefficient testing, the model that was selected for forecasting in this case due to having best BIC and AIC score and fit was ARIMA (4,1,0) as shown below in the BIC and AIC Tables, as ARIMA(0,1,1) had insignificant coefficients.

Hide

```
sort.score(AIC(model_410_ml,model_011_ml, model_310_ml, model_311_ml,model_110_ml), s
core = "aic")
```

	df <dbl>	AIC <dbl>
model_011_ml	2	23.12446
model_410_ml	5	24.59229
model_311_ml	5	26.92807
model_110_ml	2	27.05798
model_310_ml	4	27.58888

5 rows

Hide

```
sort.score(BIC(model_410_ml,model_011_ml, model_310_ml, model_311_ml,model_110_ml), s
core = "bic")
```

	df <dbl>	BIC <dbl>
--	-------------	--------------

	df <dbl>	BIC <dbl>
model_011_ml	2	24.40258
model_410_ml	5	27.78757
model_110_ml	2	28.33609
model_311_ml	5	30.12336
model_310_ml	4	30.14511
5 rows		

Hence, we used ARIMA(4,1,0), to forecast next 5 years egg deposition of Lake Hurron Bloaters. The plot of the forecast is below.

Hide

```
fit = Arima(bleggs.ts,c(4,1,0))
plot(forecast(fit,h=5), type = "o", ylab = "Egg Deposition", xlab = "Years", main =
"Graph 16. - 5 year forecast of Lake Huron Bloater - Egg Deposition ")
```

Graph 16. - 5 year forecast of Lake Huron Bloater - Egg Deposition

