

MATH1318 - Time Series - Assignment 2

Samuel Holt

May 10, 2019

Contents

Introduction	2
Packages	2
Import Data	2
Visualise Data	2
Auto Correlations of Lags	3
Deterministic Trends	4
Fitting and Evaluating a Linear Model	4
Hypothesis test of Normality of Residuals	6
ACF for Linear Model Residuals	6
Visualise Fitted Model	7
Fitting and Evaluating a Quadratic Model	8
Hypothesis test of Normality of Residuals	10
ACF for Quadratic Model Residuals	10
Visualise Fitted Model	11
Stochastic Trends	12
Data Transformation	12
Box Cox	12
ACF and PACF of Transformed Data	13
Transformed Data Visualisation	14
Differencing Transformed Data	15
Visualise Differenced Data	17
Models for Differenced Data	18
ACF and PACF	18
EACF	19
BIC Table	19
Fitting ARIMA Models	20
Running Hypothesis Tests for Coefficients	21
Refining Models with AIC and BIC Tests	23
Evaluating the Fit of the optimal ARIMA Model	23

Normality of Residuals	23
Significance of Lags	24
Forecast with ARIMA Model	26
Summary of Fitted ARIMA Model	27
Strengths and Limitations of the Model	27
Appendix	28

Introduction

The data provided for this assignment was the annual egg depositions (in millions) of age-3 Lake Huron Bloaters (Coregonous hoyi) between the years of 1981 and 1996. This data are available from the FSAdata package.

For this report I will be investigating deterministic and stochastic trends. First I will process and prepare the data to look at stationarity, existence of a trend, a change in variance, possible seasonality and intervention points, and behavior of the time series. Then I will fit and evaluate deterministic and stochastic models to the data. Once a valid model has been found, I will forecast the following 5 years of the data.

Packages

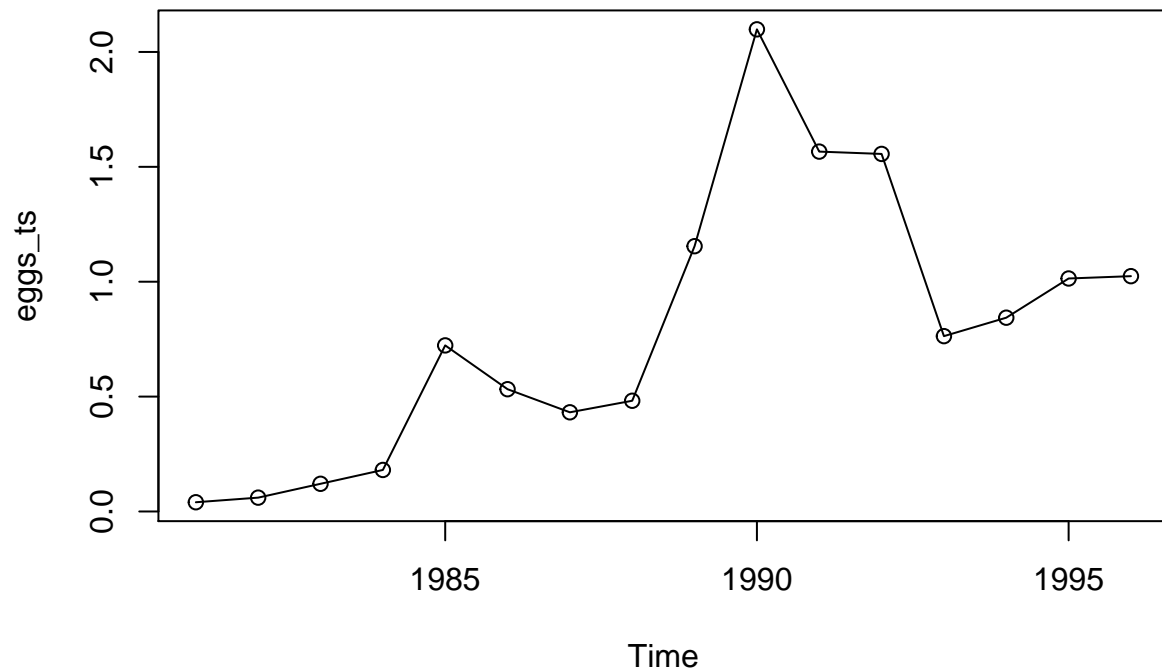
Import Data

```
setwd("C:/Users/samgh/Desktop/Masters of Statistics and Operations Research/Year 2/Sem 1/Time Series Analysis")
eggs <- read.csv('eggs.csv')
eggs_ts <- ts(eggs$eggs, start = 1981, frequency = 1)
```

Visualise Data

Visualise the time series data and investigate trend, change in variance, seasonality, intervention point, and behavior. These aspects will aid modelling decisions.

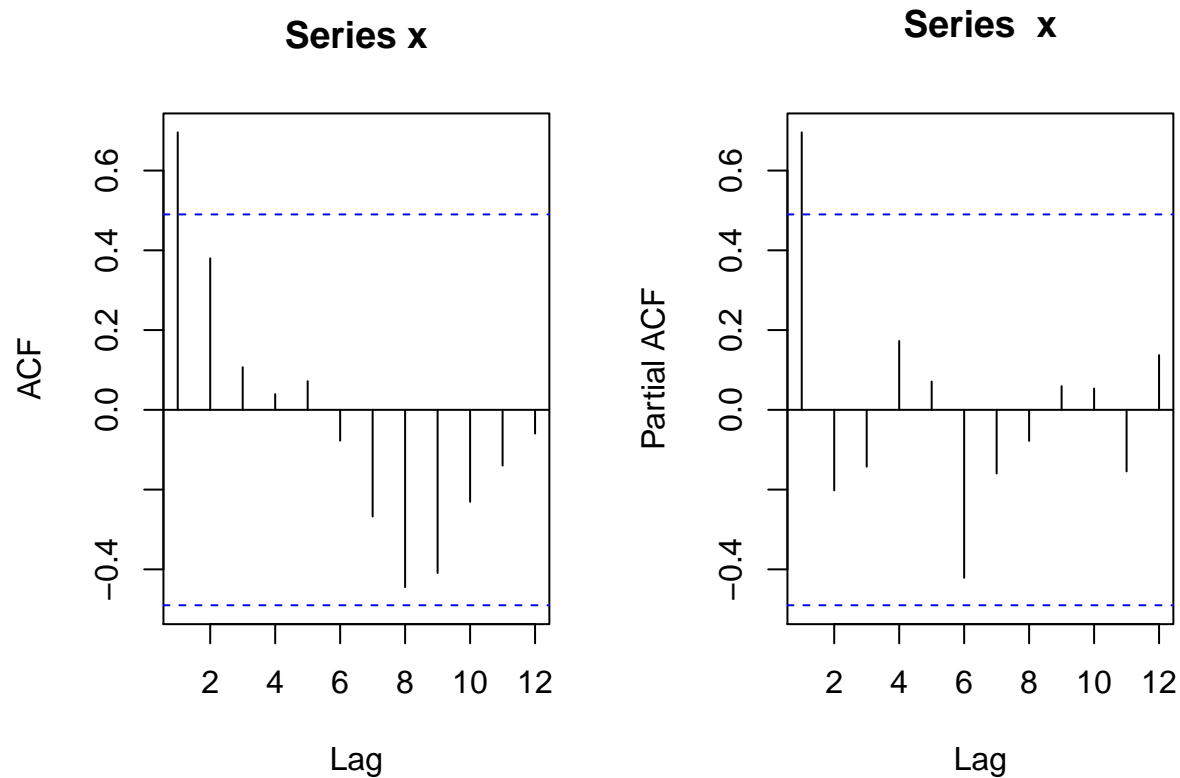
```
plot(eggs_ts, type = 'o')
```



There is a positive upward trend in the data. Perhaps a gradual increase in variance, although that may have been caused by the outlier in 1990. No seasonality is immediately or obviously present. A possible intervention point at 1988, causing a great increase over 2 year, and then perhaps again at 1990 bloater population control measures were implemented so that successive years saw either decrease or very slight increase in depositions. The time series seems to follow an auto regressive behavior.

Auto Correlations of Lags

```
acf_pacf(eggs_ts)
```



For the auto correlation and partial auto correlation functions, there are significant lags at lag 1. The ACF displays a sinusoidal pattern, coupled with a significant lag at 1 for the PACF, this confirms the presence of a trend in the time series data. Also indicates the non-stationarity of the data, of which will need to be transformed and differenced for ARMA/ARIMA models in the following section.

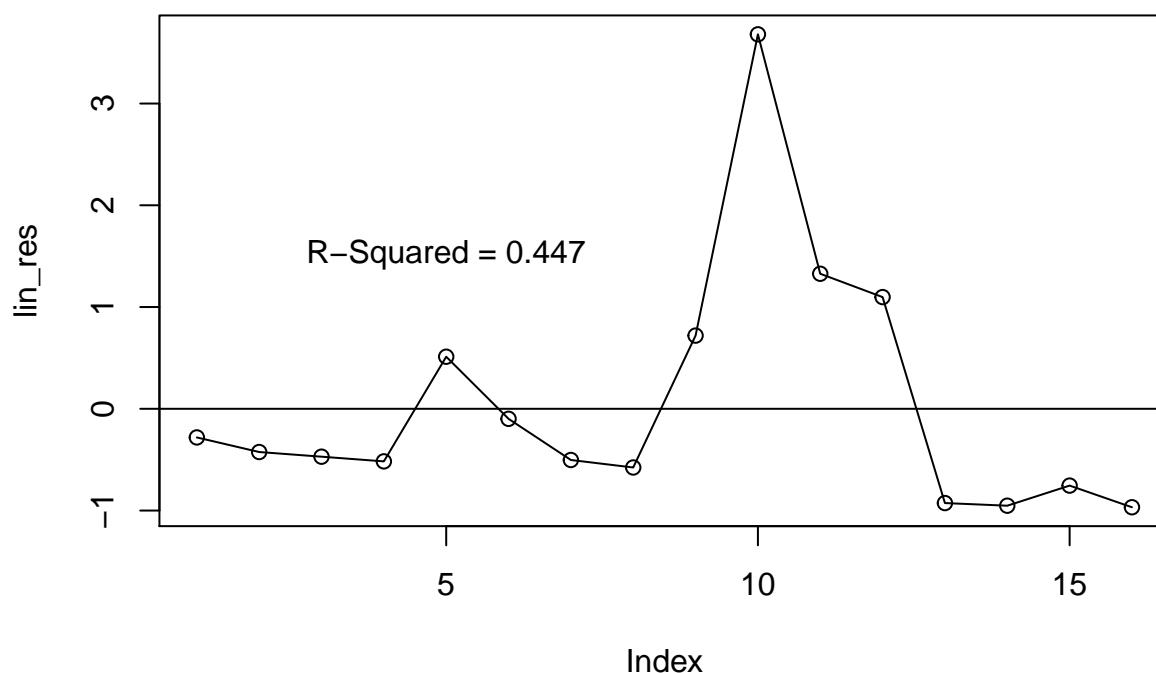
Deterministic Trends

For this section, I will be fitting and evaluating deterministic trends. Firstly with a linear model followed by a quadratic model. A harmonic model has not been considered as no seasonality nor harmonic properties are present in the time series. For each fitted model, I will be evaluating the standardised residuals of the model, looking for homoscedasticity, normality and the auto correlations. For this, I have created a function that fits a specified model and then displays the R-squared value alongside a plot of the residuals.

Fitting and Evaluating a Linear Model

```
model_ts_res(eggs_ts, method = 'linear')
```

Residuals of Linear Model Fitted

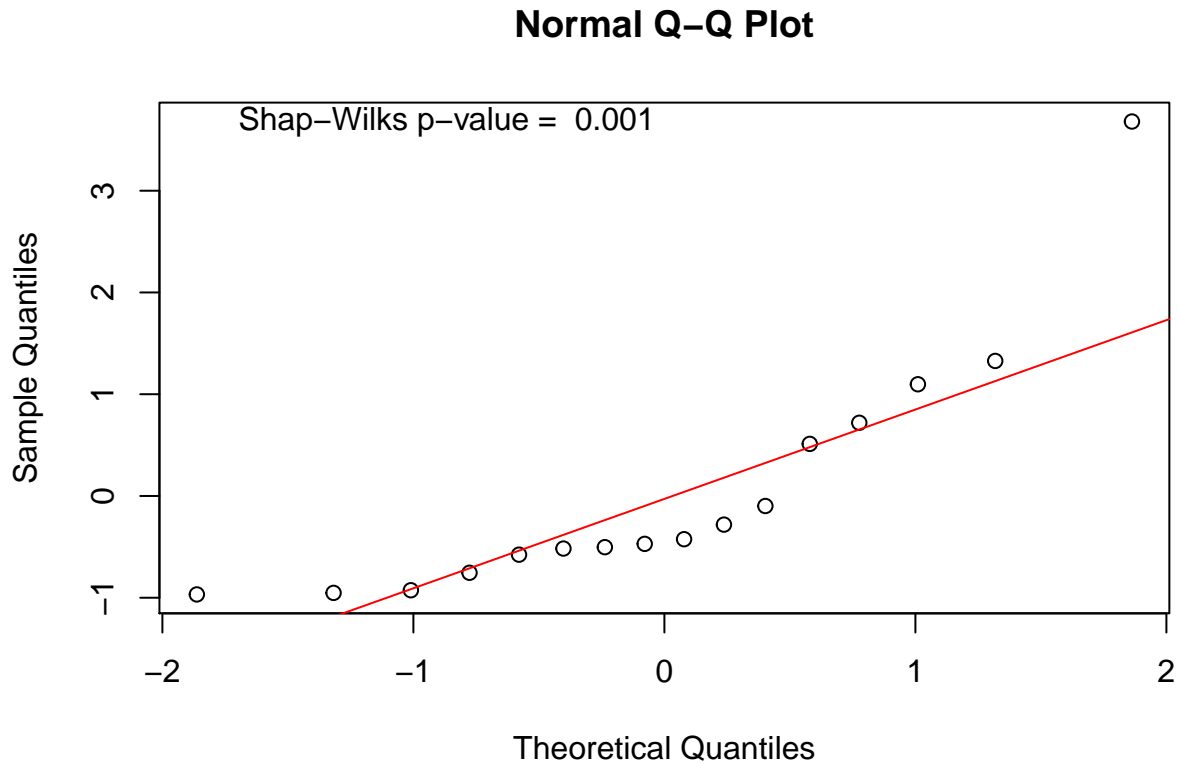


```
##
## Call:
## lm(formula = x ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.4048 -0.2768 -0.1933  0.2536  1.1857
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -165.98275   49.58836  -3.347  0.00479 **
## t              0.08387    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
```

Looking at the standardised residuals of the fitted linear model, we can see the outlier in 1990 has caused a great shift in variance of residuals. With the R-squared value being 0.447, we can say that 44.7% of the data are explained by the independent variable, time. Despite all coefficients holding statistically significant values, this model is misleading regarding its predictive capacity, as clearly the lack of homoscedasticity of the residuals indicates the model is not capturing some behaviors within the time series. We shall further investigate this by looking at the normality of standardised residuals.

Hypothesis test of Normality of Residuals

```
t <- time(eggs_ts)
lin_mod <- lm(eggs_ts ~ t)
lin_res <- rstudent(lin_mod)
normal_test(lin_res)
```

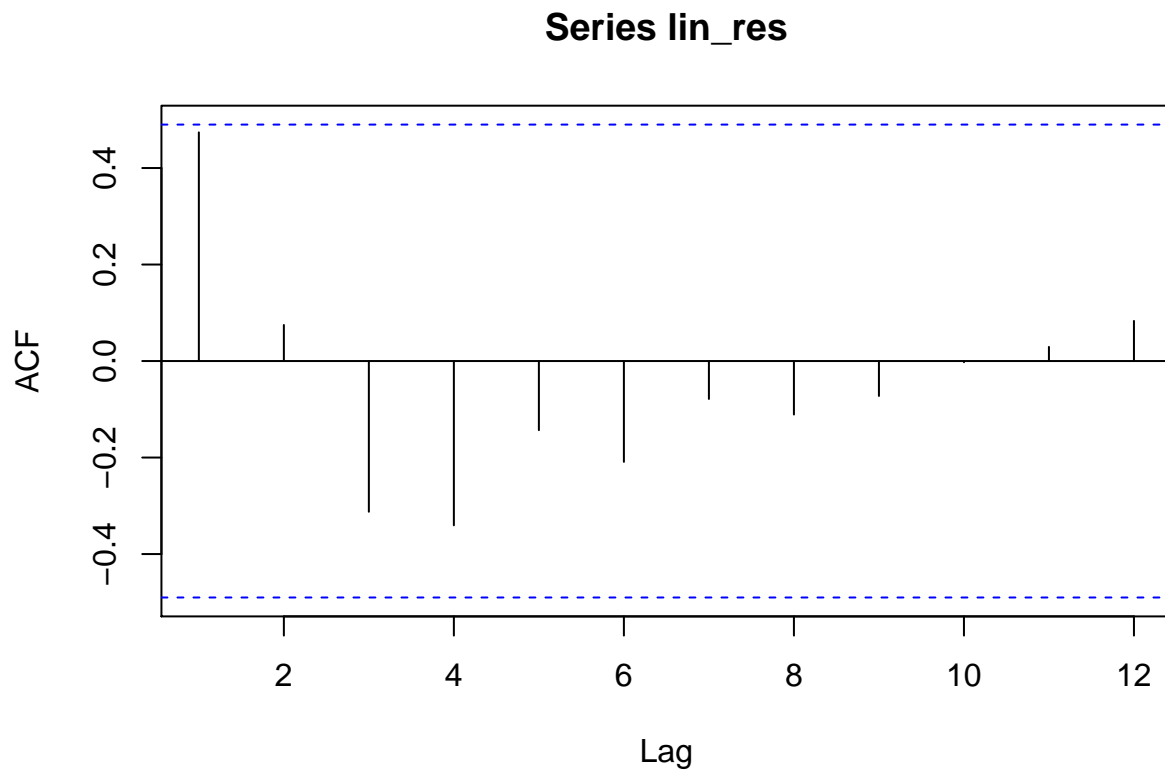


The distribution of the standardised residuals for the fitted linear model shows no signs of normality. With a p-value < 0.001 , we reject the null hypothesis of the Shapiro-Wilks test, and presume the residuals are not normally distributed.

Below an auto correlation function will be produced to investigate for white noise properties.

ACF for Linear Model Residuals

```
acf(lin_res)
```



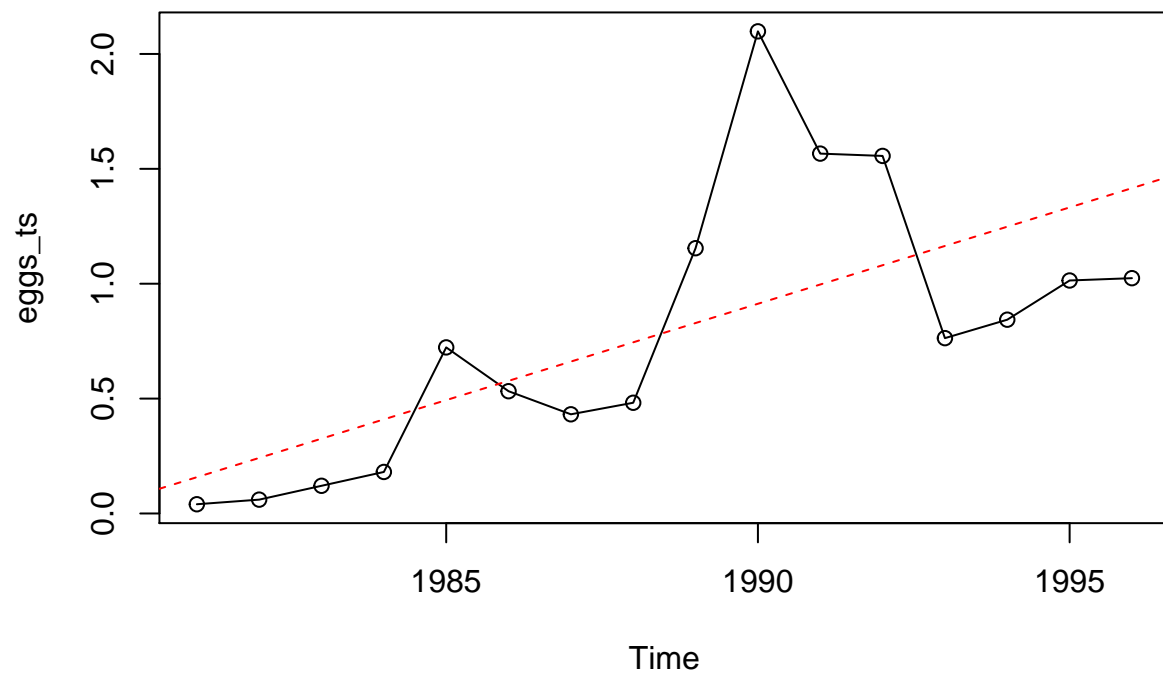
This ACF of the fitted linear model standardised residuals shows no significant lags. The first lag however is very close to the threshold and is a standout among the lag significance values. This is indicative of a model lacking in capturing all trend behaviors.

In conclusion, we have proven the linear model to be ineffectual of capturing all behaviors in the time series and thus would be limited in its predictive capabilities.

Below a visualisation of the linear model has been provided for reference.

Visualise Fitted Model

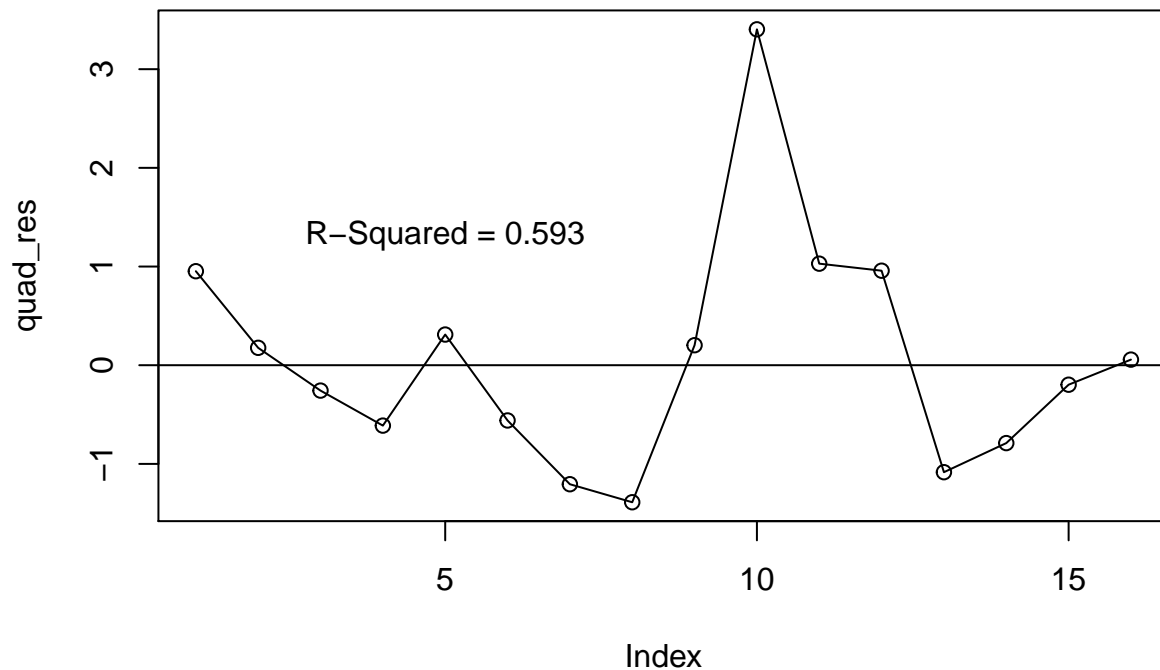
```
lin_mod <- lm(eggs_ts ~ t)
plot(eggs_ts, type = 'o')
abline(lin_mod, col = 2, lty = 2)
```



Fitting and Evaluating a Quadratic Model

```
model_ts_res(eggs_ts, method = 'quadratic')
```


Residuals of Fitted Quadratic Model

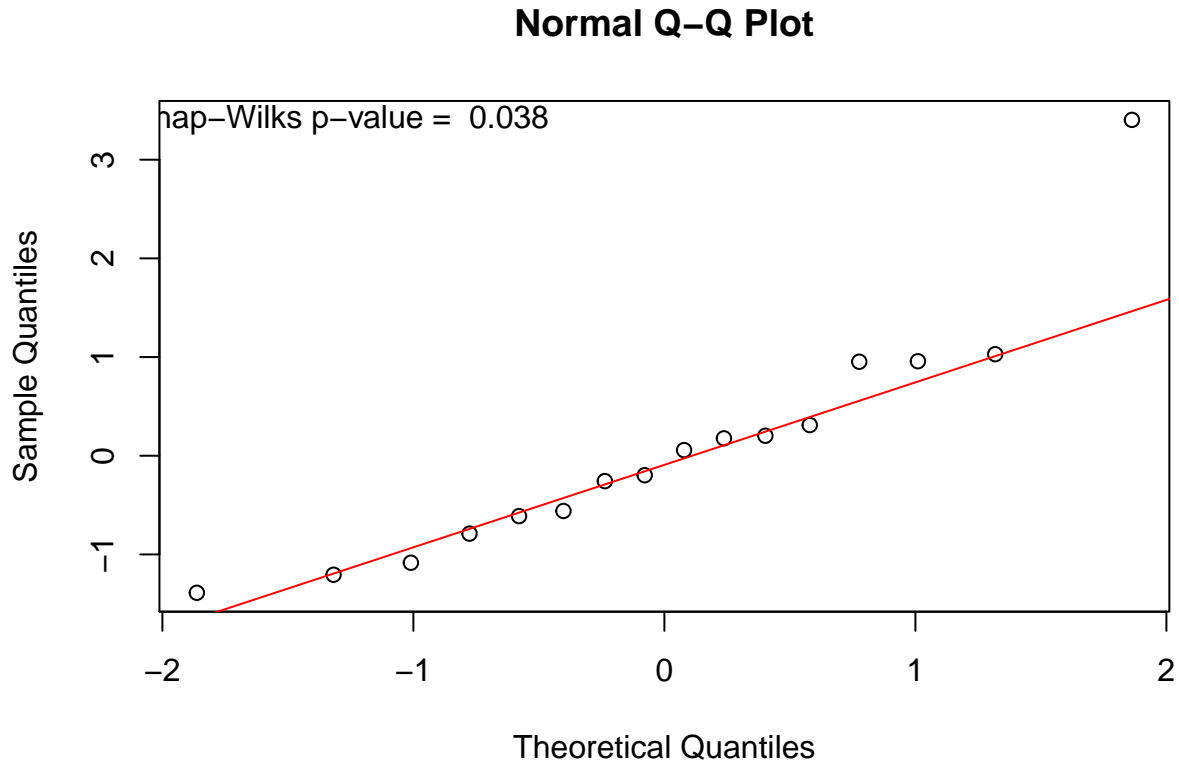


```
##
## Call:
## lm(formula = x ~ t + t2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50896 -0.25523 -0.02701  0.16615  0.96322
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.647e+04  2.141e+04  -2.170   0.0491 *
## t             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
```

Fitting a quadratic model to the time series shows an improved R-squared value of 0.593, compared to the previous linear fit. This indicates that a quadratic model explains 59.3% of the variability in the time series data. Although an improvement upon the linear model of 0.447 and all coefficients have been found to be statistically significant, this is still deemed as an ineffectual model for capturing all trends. Below I will investigate normality of the standardised residuals of the fitted quadratic model.

Hypothesis test of Normality of Residuals

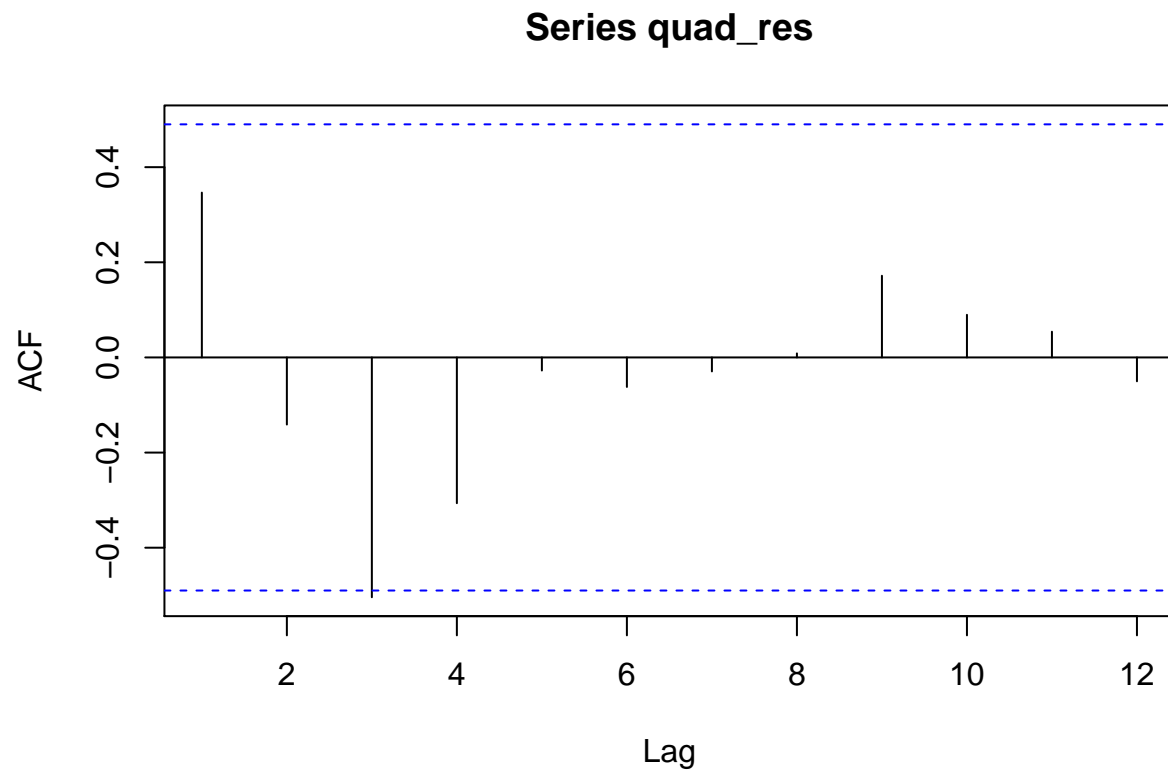
```
t <- time(eggs_ts)
t2 <- t ^ 2
quad_mod <- lm(eggs_ts ~ t + t2)
quad_res <- rstudent(quad_mod)
normal_test(quad_res)
```



The distribution of the standardised residuals for the fitted quadratic model shows no signs of normality. With a p-value = 0.038, we reject the null hypothesis of the Shapiro-Wilks test, and presume the residuals are not normally distributed.

ACF for Quadratic Model Residuals

```
acf(quad_res)
```



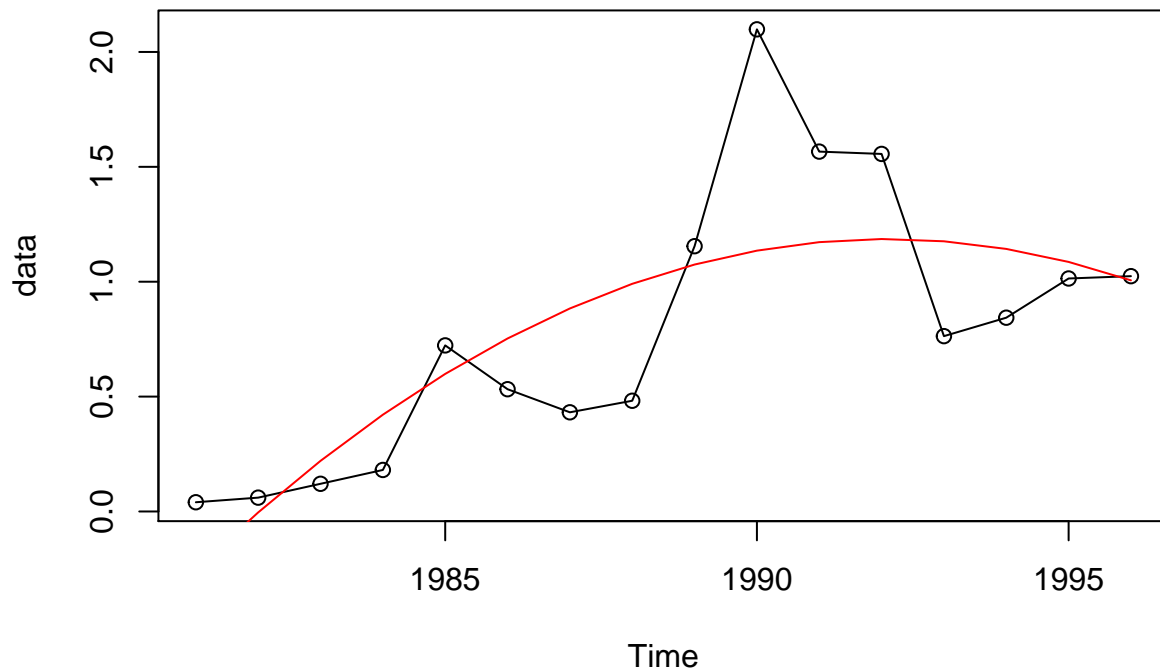
For the auto correlation function plot of the standardised residuals of the fitted quadratic model, there is a significant value for lag 3. There is also a slight presence of a pattern in the values, indicating not only an invalid model, but a further trend or behavior the model is not capturing.

Below I have visualised the quadratic model for reference. We can see that the outlier in 1990 has caused the model to believe there will be a continuing downward trend after 1992.

Visualise Fitted Model

```
fit_quad_mod(eggs_ts)
```

Fitted Quadractic Model Overlay



Stochastic Trends

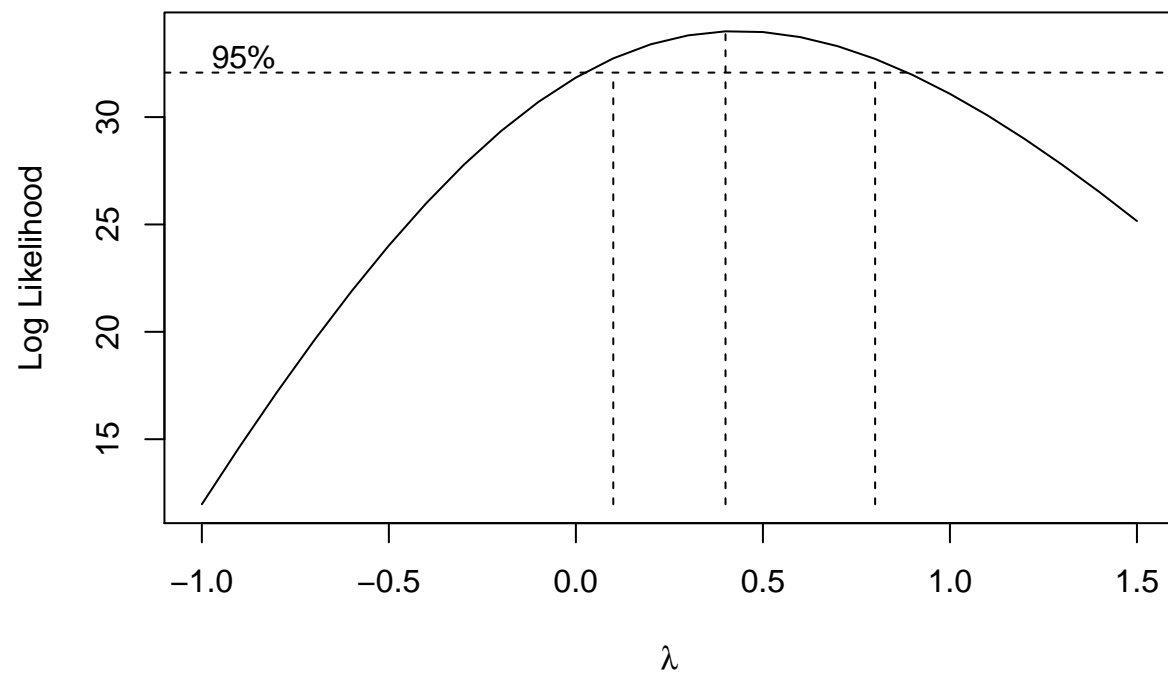
In this section I will be investigating the parameters of ARMA and ARIMA models, evaluating the parameters found and then forecasting with the most statistically significant model for the next 5 years.

I will investigate the possible transformation of the data by searching for possible values of lambda using the Box Cox transform function. Then I will look into possible differencing of the data to form a stationary time series to model with ARMA/ARIMA. I will then search for valid parameters of the ARMA/ARIMA model and then evaluate these models using BIC/AIC functions. The normality of these residuals will be explored

Data Transformation

Box Cox

```
box_this <-  
  TSA::BoxCox.ar(eggs_ts, lambda = seq(-1, 1.5, 0.1), method = 'yw')  
  
## 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
```



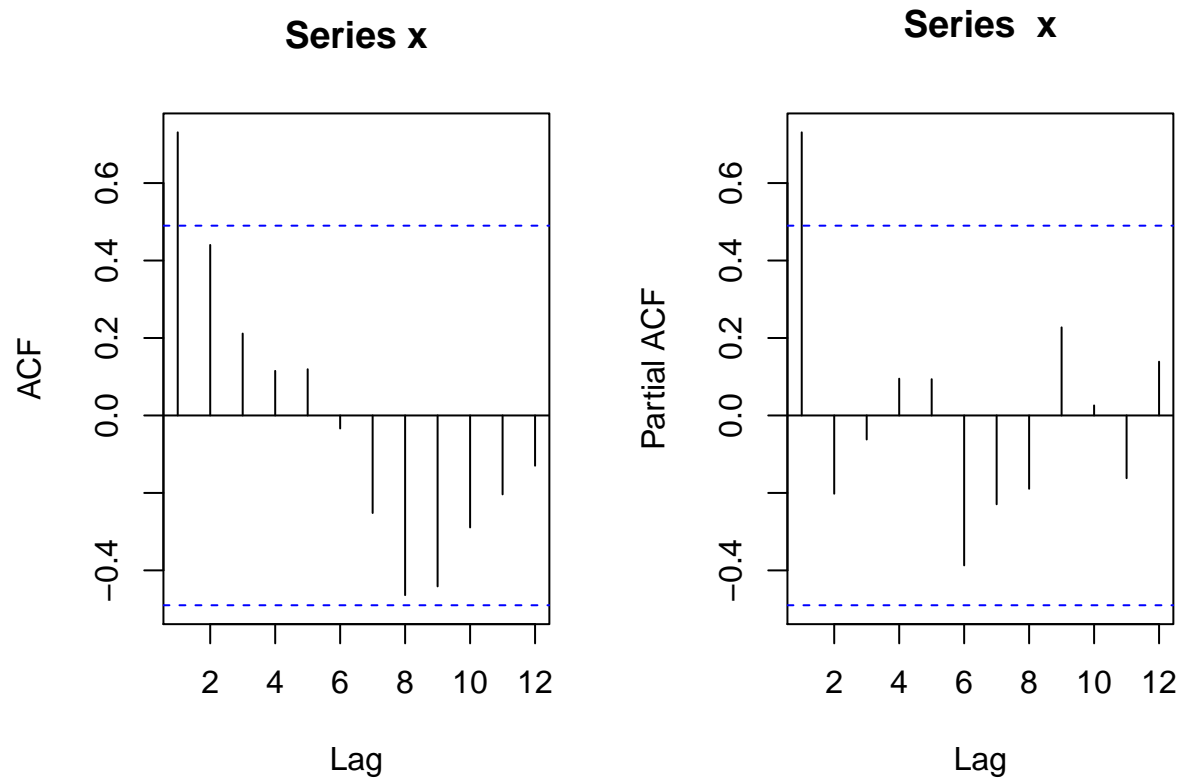
```
box_this$ci
```

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

```
eggs_sq <- sqrt(eggs_ts)
```

ACF and PACF of Transformed Data

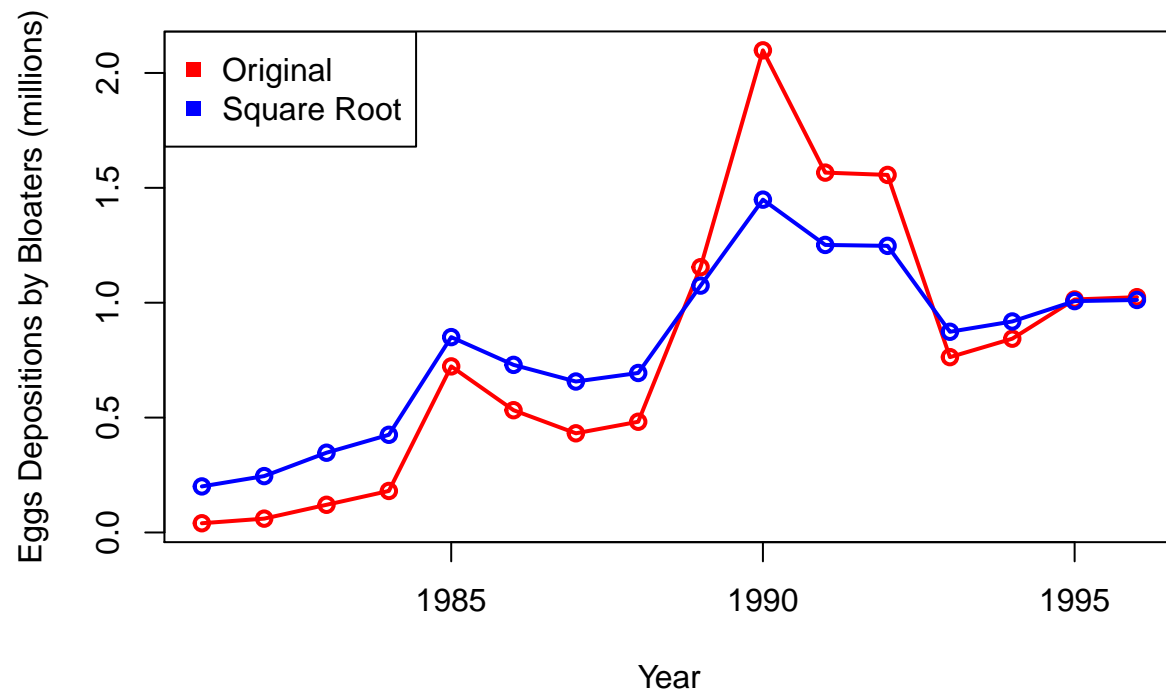
```
acf_pacf(eggs_sq)
```



Transformed Data Visualisation

```
plot(
  eggs_ts,
  col = 'red',
  lwd = 2,
  main = 'Bloaters Data Transformed',
  xlab = 'Year',
  ylab = "Eggs Depositions by Bloaters (millions)",
  type = 'o'
)
lines(eggs_sq, col = 'blue', lwd = 2, type = 'o')
legend('topleft', legend = c('Original', 'Square Root'), col = c('red', 'blue'), pch = 15)
```

Bloaters Data Transformed



Differencing Transformed Data

```
ar(diff(eggs_sq))$order
```

```
## [1] 0
```

```
adfTest(eggs_sq, lags = 0)
```

```
##  
## Title:  
## Augmented Dickey-Fuller Test  
##  
## Test Results:  
## PARAMETER:  
## Lag Order: 0  
## STATISTIC:  
## Dickey-Fuller: 0.192  
## P VALUE:  
## 0.6689  
##  
## Description:  
## Sat May 11 20:32:30 2019 by user: samgh
```

```
eggs_d1 <- diff(eggs_sq, difference = 1)
ar(diff(eggs_d1))$order
```

```
## [1] 4
```

```
adfTest(eggs_d1, lags = 4)
```

```
##
## Title:
## Augmented Dickey-Fuller Test
##
## Test Results:
## PARAMETER:
## Lag Order: 4
## STATISTIC:
## Dickey-Fuller: -0.8002
## P VALUE:
## 0.3539
##
## Description:
## Sat May 11 20:32:30 2019 by user: samgh
```

```
eggs_d2 <- diff(eggs_sq, difference = 2)
ar(diff(eggs_d2))$order
```

```
## [1] 4
```

```
adfTest(eggs_d2, lags = 4)
```

```
##
## Title:
## Augmented Dickey-Fuller Test
##
## Test Results:
## PARAMETER:
## Lag Order: 4
## STATISTIC:
## Dickey-Fuller: -1.5305
## P VALUE:
## 0.1221
##
## Description:
## Sat May 11 20:32:30 2019 by user: samgh
```

```
eggs_d3 <- diff(eggs_sq, difference = 3)
ar(diff(eggs_d3))$order
```

```
## [1] 4
```



```
adfTest(eggs_d3, lags = 4)
```

```
##
## Title:
## Augmented Dickey-Fuller Test
##
## Test Results:
## PARAMETER:
## Lag Order: 4
## STATISTIC:
## Dickey-Fuller: -1.2635
## P VALUE:
## 0.2068
##
## Description:
## Sat May 11 20:32:30 2019 by user: samgh
```

```
eggs_d4 <- diff(eggs_sq, difference = 4)
ar(diff(eggs_d4))$order
```

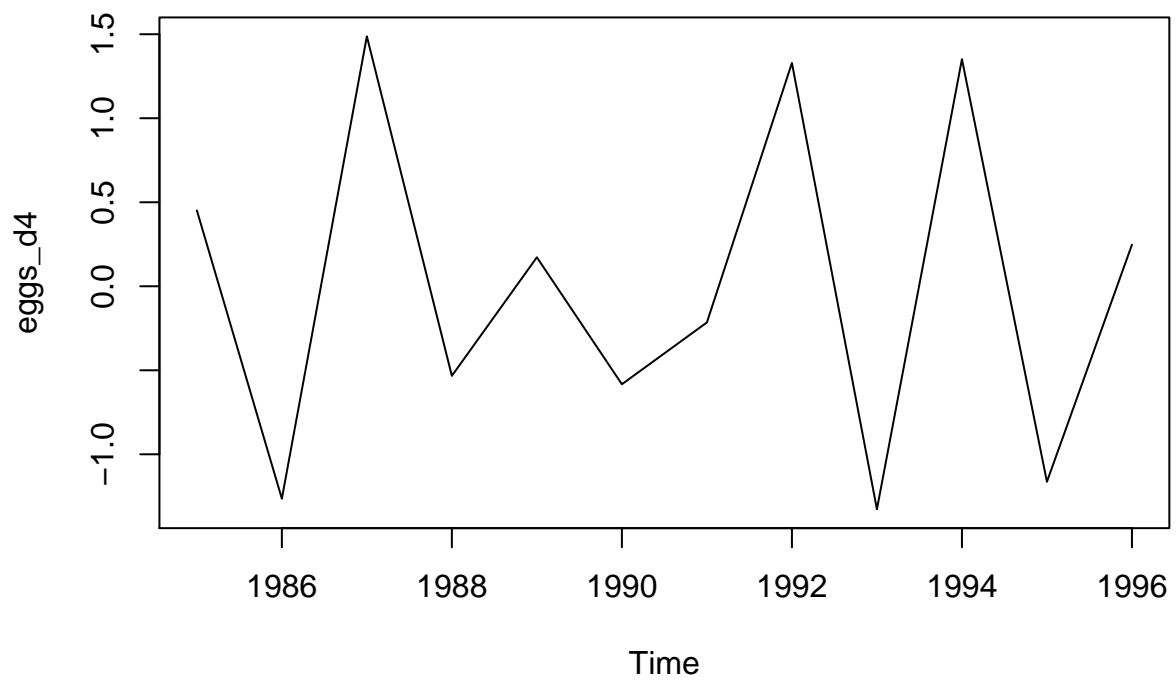
```
## [1] 2
```

```
adfTest(eggs_d4, lags = 2)
```

```
##
## Title:
## Augmented Dickey-Fuller Test
##
## Test Results:
## PARAMETER:
## Lag Order: 2
## STATISTIC:
## Dickey-Fuller: -2.2915
## P VALUE:
## 0.02382
##
## Description:
## Sat May 11 20:32:30 2019 by user: samgh
```

Visualise Differenced Data

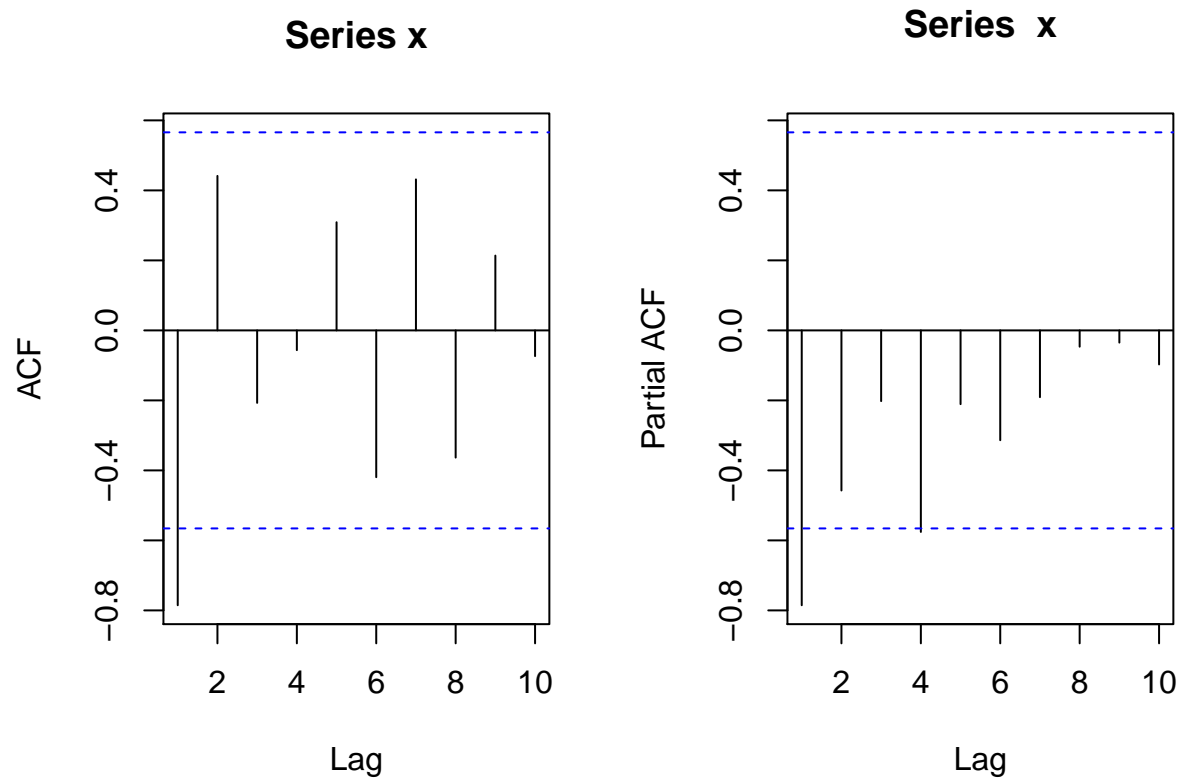
```
plot(eggs_d4)
```



Models for Differenced Data

ACF and PACF

```
acf_pacf(eggs_d4) # q = 1, p = 2
```



ARIMA(2,4,1) or ARIMA(1,4,1) since the second PACF signif lag is very close

EACF

```
eacf(eggs_d4, ar.max = 2, ma.max = 2)
```

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

ARIMA(1,4,0) or ARIMA(0,4,1)

BIC Table

```
res <- armasubsets(eggs_d4, nar = 3, nma = 3)
```

```
## 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, : 2 linear dependencies found
```

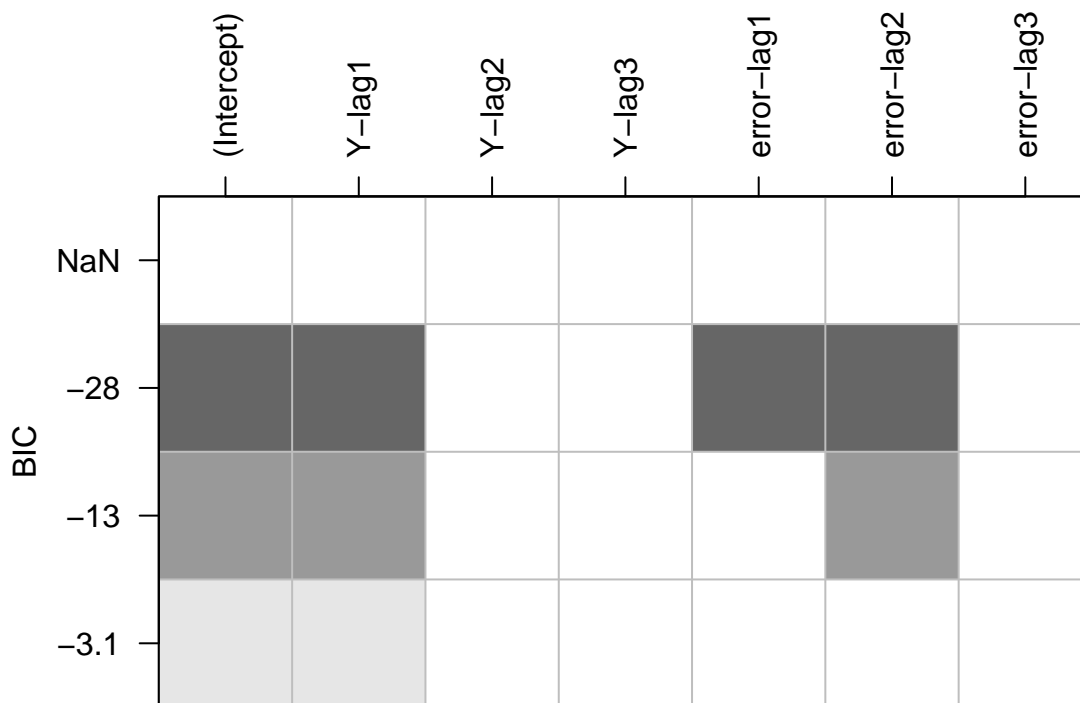
```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : nvmax reduced to 4
```

```
plot(res)
```

```
## Warning in log(vr): NaNs produced
```

```
## Warning in log(vr): NaNs produced
```

```
## Warning in log(vr): NaNs produced
```



Fitting ARIMA Models

In accordance with the ACF, PACF, EACF and the BIC table outputs, the following models will be fitted, have hypothesis test run for each of its respective coefficients and then compared using AIC and BIC functions:

- ARIMA(0,4,1)
- ARIMA(0,4,2)
- ARIMA(1,4,0)

- ARIMA(2,4,0)
- ARIMA(2,4,2)
- ARIMA(1,4,1)
- ARIMA(1,4,2)
- ARIMA(2,4,1)
- ARIMA(2,4,2)

The method argument will be set to maximum likelihood as it is the most accurate despite being CPU intensive, due to the small scale of the data, this should not cause issues.

```
model041 <- arima(eggs_sq, order = c(0,4,1), method = 'ML')
model140 <- arima(eggs_sq, order = c(1,4,0), method = 'ML')
model141 <- arima(eggs_sq, order = c(1,4,1), method = 'ML')
model142 <- arima(eggs_sq, order = c(1,4,2), method = 'ML')
model042 <- arima(eggs_sq, order = c(0,4,2), method = 'ML')
model240 <- arima(eggs_sq, order = c(2,4,0), method = 'ML')
model241 <- arima(eggs_sq, order = c(2,4,1), method = 'ML')
model242 <- arima(eggs_sq, order = c(2,4,2), method = 'ML')
```

Running Hypothesis Tests for Coefficients

```
coeftest(model041) # theta1 = -0.97, p-val < 0.001 stat signif
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.97860    0.20856 -4.6922 2.703e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model140) # phi1 = -0.753, p-val < 0.001 stat signif
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.75275    0.16065 -4.6856 2.792e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model141) # phi1 = -0.594, theta1 = -0.972, both stat signif
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.59355    0.20767 -2.8581 0.004262 **
## ma1 -0.97229    0.23585 -4.1225 3.748e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model142) # ph1 not stat sig, theta1 = -1.844, theta2 = 0.907, both MA parameters stat signif
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.26516    0.28008 -0.9467  0.34379
## ma1 -1.84443    0.39772 -4.6375 3.526e-06 ***
## ma2  0.90748    0.40244  2.2549  0.02414 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model042)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -1.86561    0.32297 -5.7765 7.629e-09 ***
## ma2  0.94485    0.31899  2.9620  0.003057 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model240)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -1.13789    0.26265 -4.3324 1.475e-05 ***
## ar2 -0.49761    0.28579 -1.7412  0.08165 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model241) # phi1 = -0.750, phi2 not stat signif, theta1 = -0.967 stat signif
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.74957    0.28850 -2.5982 0.009372 **
## ar2 -0.21426    0.28894 -0.7415 0.458369
## ma1 -0.96695    0.25614 -3.7751 0.000160 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(model242) # neither phi value stat signif, theta1 = -1.84, theta2 = 0.91, both stat signif
```

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

```
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.2640770  0.3192937 -0.8271  0.40820
## ar2  0.0021065  0.2964192  0.0071  0.99433
## ma1 -1.8447917  0.3994339 -4.6185 3.865e-06 ***
## ma2  0.9078541  0.4040854  2.2467  0.02466 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Only the models ARIMA(2,4,1) and ARIMA(2,4,2) display overfitting with insignificant coefficients, the rest all hold significant values. These remaining models will be funneled into AIC and BIC functions to determine the best fit for forecasting.

Refining Models with AIC and BIC Tests

```
AIC(model041,model140,model042,model240,model141,model142,model241,model242) # ARIMA(0,4,2) with 25.8
```

```
##      df      AIC
## model041  2 31.59691
## model140  2 32.44766
## model042  3 25.80334
## model240  3 32.05642
## model141  3 28.09207
## model142  4 27.07391
## model241  4 29.59189
## model242  5 29.07309
```

For the AIC ranking, the best fit model is the ARIMA(0,4,2), I will continue with the BIC function to see if it agrees with the AIC function output.

```
BIC(model041,model140,model240,model042,model141,model142,model241,model242) # ARIMA(0,4,2) with 27.01
```

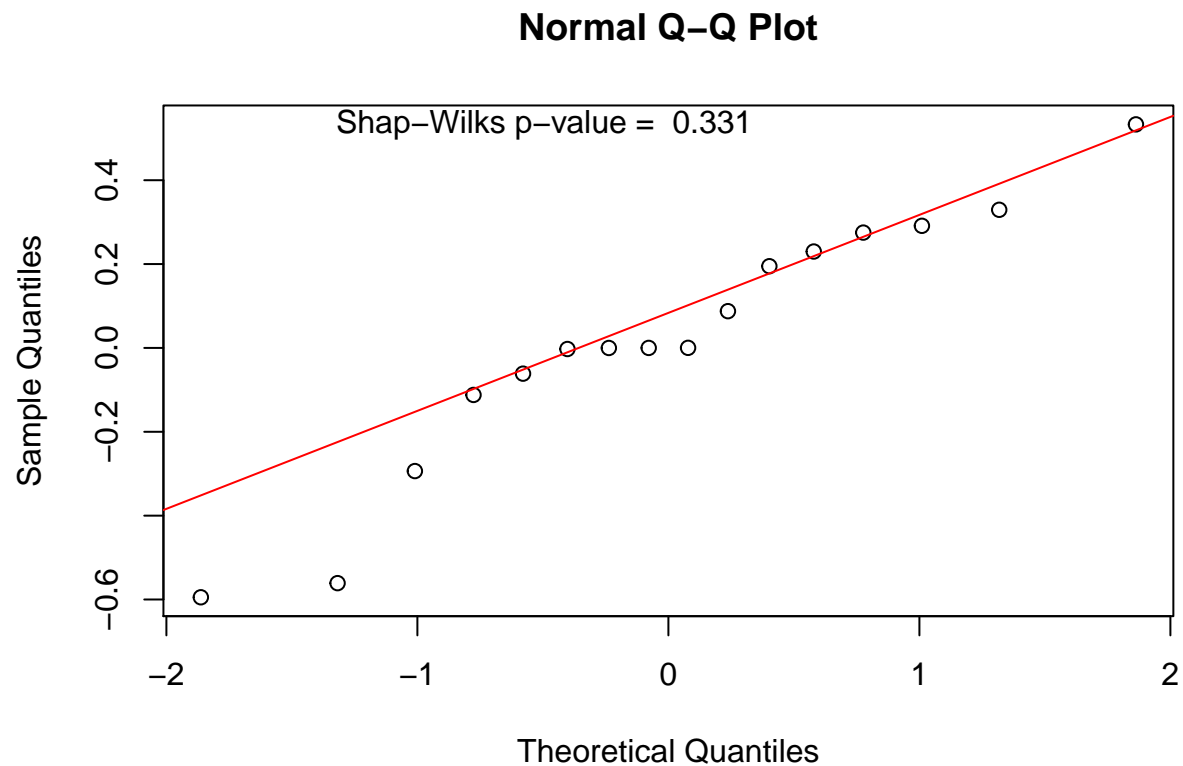
```
##      df      BIC
## model041  2 32.56673
## model140  2 33.41747
## model240  3 33.51114
## model042  3 27.25806
## model141  3 29.54679
## model142  4 29.01354
## model241  4 31.53152
## model242  5 31.49762
```

Here we see the BIC function agrees with the previous AIC function, the model ARIMA(0,4,2) will have its residuals investigated for normality and significance of lags to validate the model, if these tests are deemed satisfactory, the model will be used to forecast for the proceeding 5 years.

Evaluating the Fit of the optimal ARIMA Model

Normality of Residuals

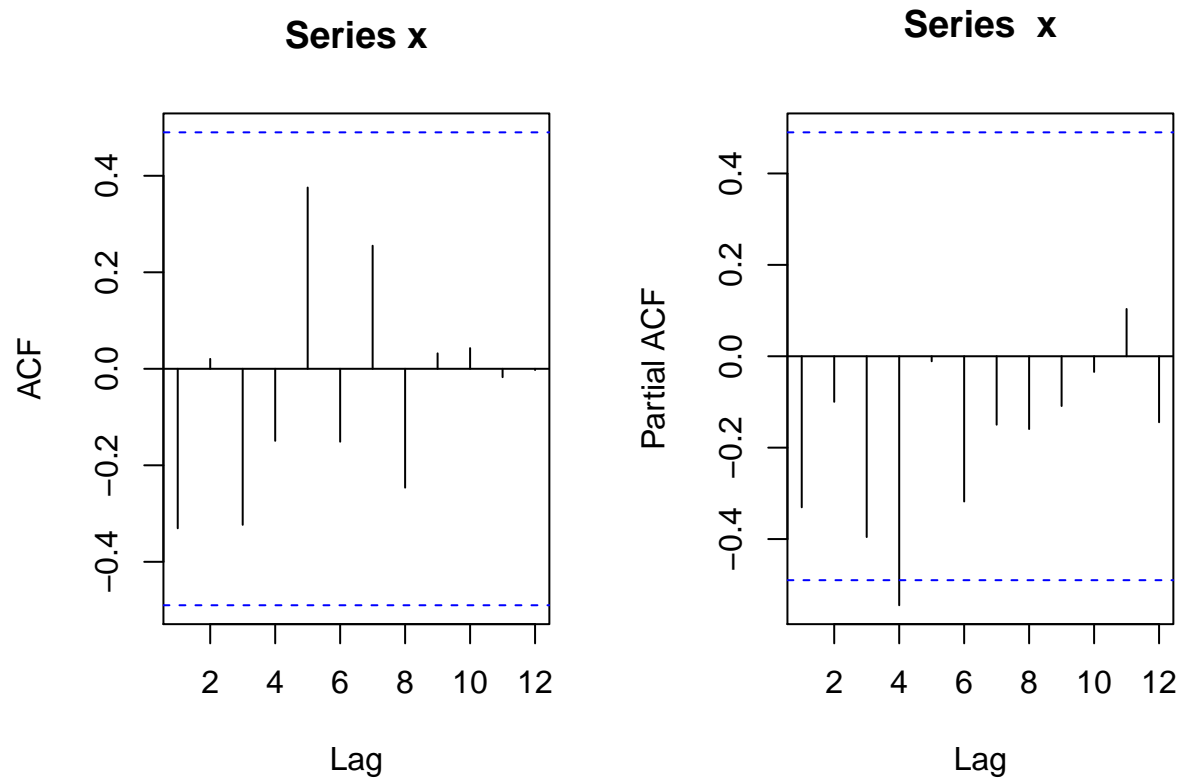
```
normal_test(residuals(model042))
```



The normality of the standardised residuals is proven to be evident with a p-value from the Shapiro-Wilks test for normality at 0.331. This means we fail to reject the null hypothesis and presume that the residuals are normally distributed for the ARIMA(0,4,2) model.

Significance of Lags

```
acf_pacf(residuals(model042)) # white noise present
```

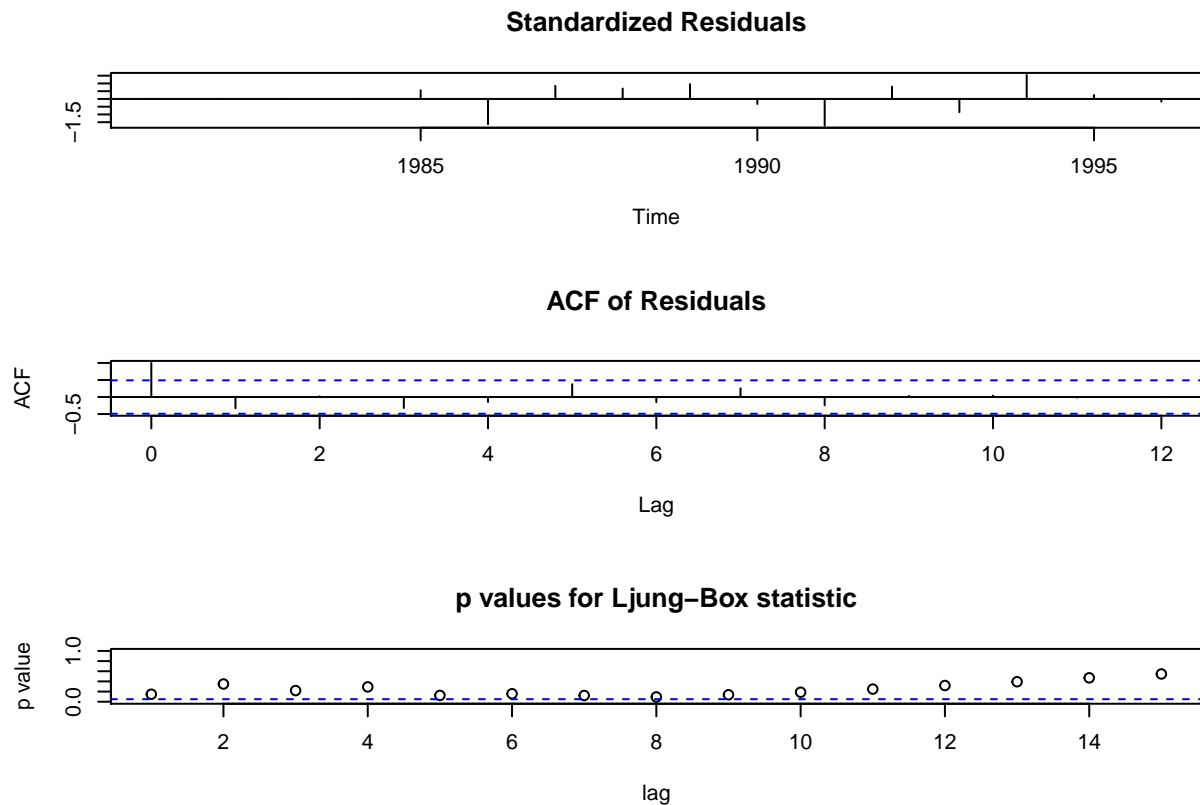



The ACF and PACF plots demonstrate white noise characteristics, a late significant value at lag 4, although the residual values for the ACF and PACF are clearly detrended and are without any noticeable pattern. The Ljung-Box test will be run for the residuals of the model to investigate significant lags further. The `tsdiag` function visualises this test, alongside an ACF plot and the standardised residuals.

```
Box.test(residuals(model042), lag = 10, type = 'Ljung-Box', fitdf = 0) # supports no autocorrelations u

##
## Box-Ljung test
##
## data: residuals(model042)
## X-squared = 13.704, df = 10, p-value = 0.1869

tsdiag(model042, gof = 15, omit.initial = F) # although very close on lag 5, no statistically significant
```



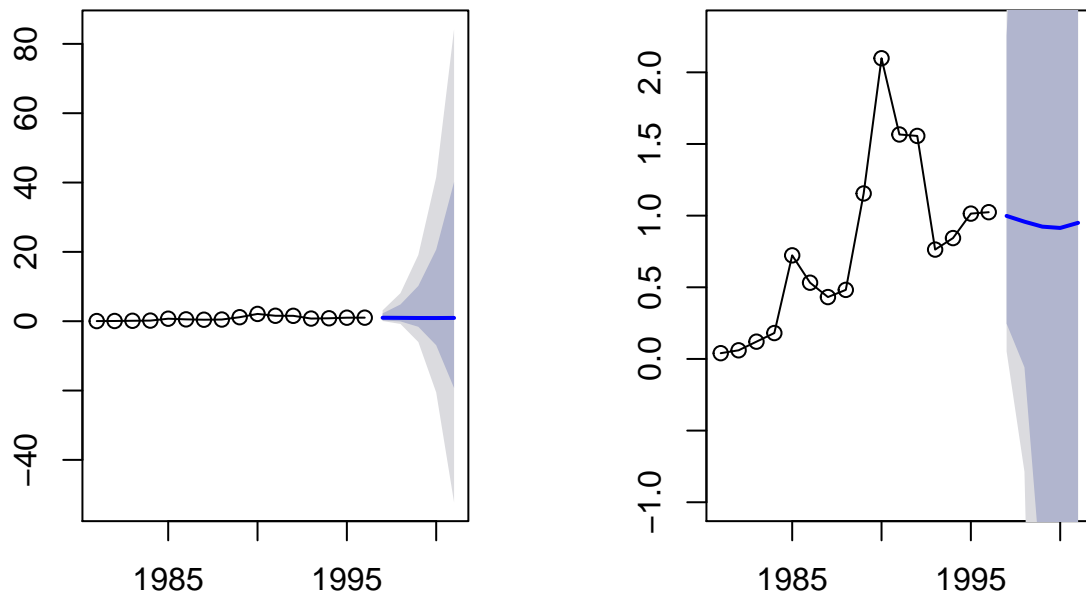
The Box-Ljung test p-value comes to 0.1869, therefore we fail to reject the null hypothesis and presume there are independently distributed and do not exhibit any significant lags. The graphics provided further demonstrate this claim as not p-value falls below the alpha level.

Now we will forecast the following 5 years of the time series data with the ARIMA(0,4,2) model. A

Forecast with ARIMA Model

```
fit = Arima(eggs_ts, order = c(0,4,2), lambda = 0.5)
par(mfrow = c(1,2))
plot(forecast(fit, h = 5), type = 'o', main = 'Full 5 Year Forecast ARIMA(0,4,2)')
plot(forecast(fit, h = 5), type = 'o', ylim = c(-1,2.3), main = 'Zoom In of ARIMA(0,4,2) Forecast')
```

Full 5 Year Forecast ARIMA(0,4,2) Zoom In of ARIMA(0,4,2) Forecast



Summary of Fitted ARIMA Model

An ARIMA model was fitted to predict the following 5 years of the bloater egg depositions time series data from 1981 to 1996. This ARIMA model's parameters were defined by investigating the transformed data with ACF, PACF, EACF and BIC tables. Each of the model options then had their parameter estimate coefficients tested for statistical significance. The models of which that held statistical significance ($p\text{-value} < \alpha \text{ level}$) and were without symptoms of overfitting would be evaluated together using AIC and BIC functions. The lowest value from this function would coincide with the most effective model. This model was found to be ARIMA(0,4,2). The following 5 years were then predicted, with a close up of the single predicted values for the time series.

Strengths and Limitations of the Model

The ARIMA(0,4,2) model, although holding greatest significance for all statistical tests compared to the other models presented, shows a vast range of possible values after only a few years. This would suggest that the moving average behavior of the fit may be too much for the time series data. I would go further in suggesting that the original data visually displayed autoregressive characteristics, but as shown $p = 0$, meaning no autoregressive behavior is being considered for the model. A further limitation of the predictive aspects of the model is that there are negative values present in the forecast, which is not helpful as egg depositions cannot have a negative count.

Below is an appendix of the custom functions created for the report.

Appendix

```
# Run both AutoCorrelation Function and Partial AutoCorreltion Functions and visualise together
acf_pacf <- function(x) {
  par(mfrow = c(1, 2))
  TSA::acf(x)
  stats::pacf(x)
  par(mfrow = c(1, 1))
}

# Fit a Model and then summarise and visualise Standardised Residuals
model_ts_res <- function(x, method) {
  if (method == 'linear') {
    t <- time(x)
    lin_mod <- lm(x ~ t)
    lin_res <- rstudent(lin_mod)
    sum_lin <- summary(lin_mod)
    r2 <- round(sum_lin$r.squared, 3)
    plot(lin_res, type = 'o', main = 'Residuals of Linear Model Fitted')
    abline(h = 0)
    text(5, lin_res[5] + 1, paste('R-Squared =', r2))
  }
  if (method == 'quadratic') {
    t <- time(x)
    t2 <- t ^ 2
    quad_mod <- lm(x ~ t + t2)
    quad_sum <- summary(quad_mod)
    r2_q <- round(quad_sum$r.squared, 3)
    quad_res <- rstudent(quad_mod)
    plot(quad_res, type = 'o', main = 'Residuals of Fitted Quadratic Model')
    abline(h = 0)
    text(5, quad_res[5] + 1, paste('R-Squared =', r2_q))
  }
}

# Visualise Fitted Quadratic Model
fit_quad_mod <- function(data){
  t <- time(data)
  t2 <- t ^ 2
  quad_mod <- lm(data ~ t + t2)
  newd <- data.frame(t = t, t2 = t2)
  pred <- predict(quad_mod, newd, interval = 'prediction')
  plot(data, type = 'o', main = 'Fitted Quadractic Model Overlay')
  lines(ts(as.vector(pred[,1]), start = min(t)), col = 2)
}

# Runs a Hypthesis Test for Normality fo Given Data Points
normal_test <- function(x) {
  shap <- shapiro.test(x)
  shap_p <- round(shap$p.value, 3)
  qqnorm(x)
  qqline(x, col = 'red')
  text('topleft', paste('p-value = ', shap_p))
}
```