MATH1318 Time Series

Assignment 2 - Semester 1, 2019

Phil Steinke s3725547

Data

Egg depositions (in millions) of age-3 Lake Huron Bloaters (Coregonus hoyi) between years 1981 and 1996 are available in BloaterLH dataset of FSAdata package.

Goals

- [x] Analyse the egg depositions of Lake Huron Bloasters
- [x] Use the analysis methods covered in the **modules 1 − 7** of MATH1318 Time Series Analysis
- [x] Choose the best model among a set of possible models for this dataset
- [x] Give forecasts of egg depositions for the next 5 years

R code

```
cat("\014") # clear everything
```

```
setwd("./.")
dataFilename <- "eggs.csv"
startYear = 1981 # get start year
endYear = 1996 # get end year
data <- read_csv(dataFilename)
# convert to timeseries, ignoring year column:
data.ts <- ts(as.vector(data[,2]), start = startYear, end = endYear)
# source('~/code/data-science/uni/time-series/common/sort.score.R')
# sort.score function from Haydar
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")')
  }
}
data.ts.raw <- data.ts # make a copy for forecasting
```

```
# class(data) -> ts
# data %>% dim() # 16 x 2
data.ts %>% dim() # 16 x 1 because we removed the year column
```

```
## [1] 16  1
```

```
default_ylab = 'Lake Huron Bloaters'
default_xlab = 'Year'
fig_nums <- captioner()

doDiffAndPlot <- function(df.ts, diffCount, showPlot = T, showEacf=F) {
  ifelse(
    diffCount != 0, (df.ts = diff(df.ts, differences = diffCount)),
```

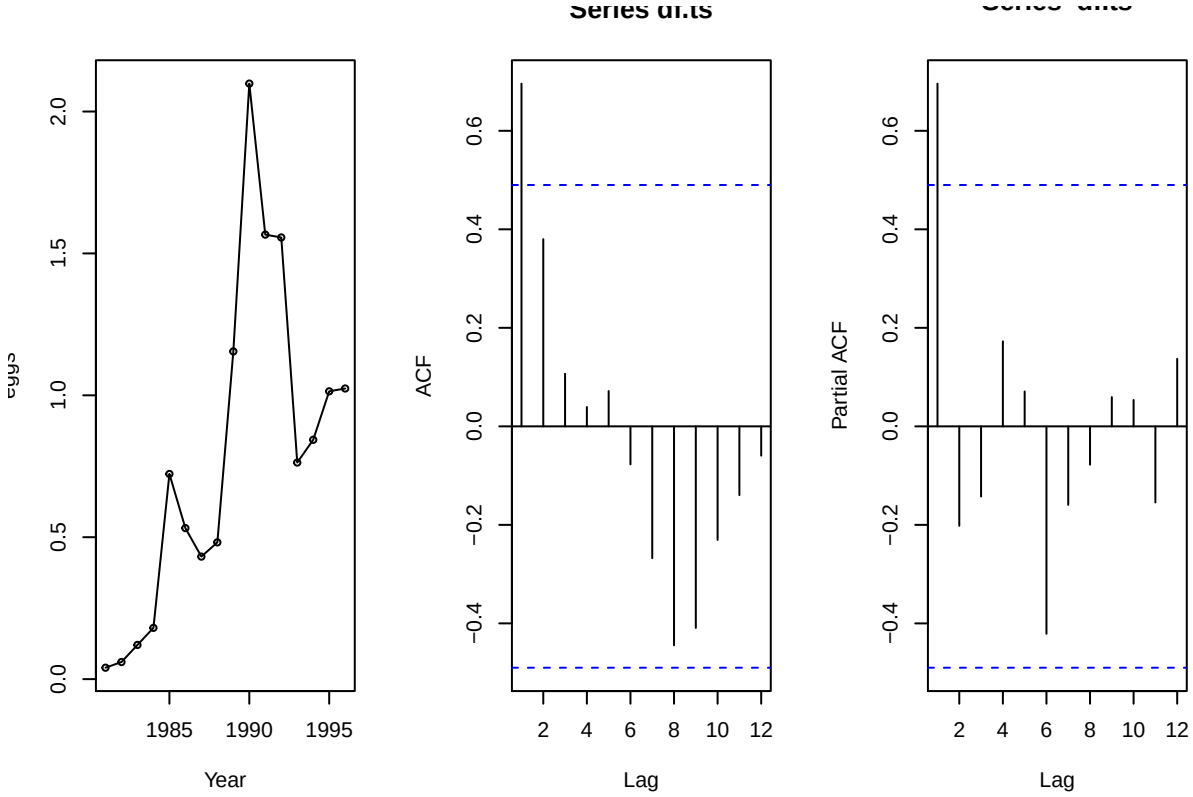
```
    'there is no diff\n')
  paste('diff: ', diffCount) %>% print()
  order = ar(diff(df.ts))$order
  paste('order count (k): ', order)
  diffAdfTest = adfTest(df.ts, lags = order, title = NULL, description = NULL)
  p <- diffAdfTest@test$p.value
  paste(
    'adf p-value:',
    p, (ifelse(p < 0.05, '< 0.05 significant', '> 0.05 insignificant')
    )) %>% print()
  if(showPlot) {
    par(mfrow = c(1,3))
    plot(
      df.ts,
      type='o',
      xlab=default_xlab,
      # ylab=fig_nums(figureName,
      #   paste('diff #', diffCount, ' of ', default_ylab, sep = '')
      # )
    )
    acf(df.ts)
    pacf(df.ts)
    par(mfrow=c(1,1))
  }

}
```

```
# The initial ts data, without any transformation:
fig_nums('initial ts data', 'initial timeseries data with no transformation') %>% cat()
```

```
## Figure  1: initial timeseries data with no transformation
```

```
doDiffAndPlot(data.ts, 0, T, F) # p = 0.4455 lag = 1 - has a trend = nope
```

```
## [1] "diff:  0"
## [1] "adf p-value: 0.452033577643409 > 0.05 insignificant"
```



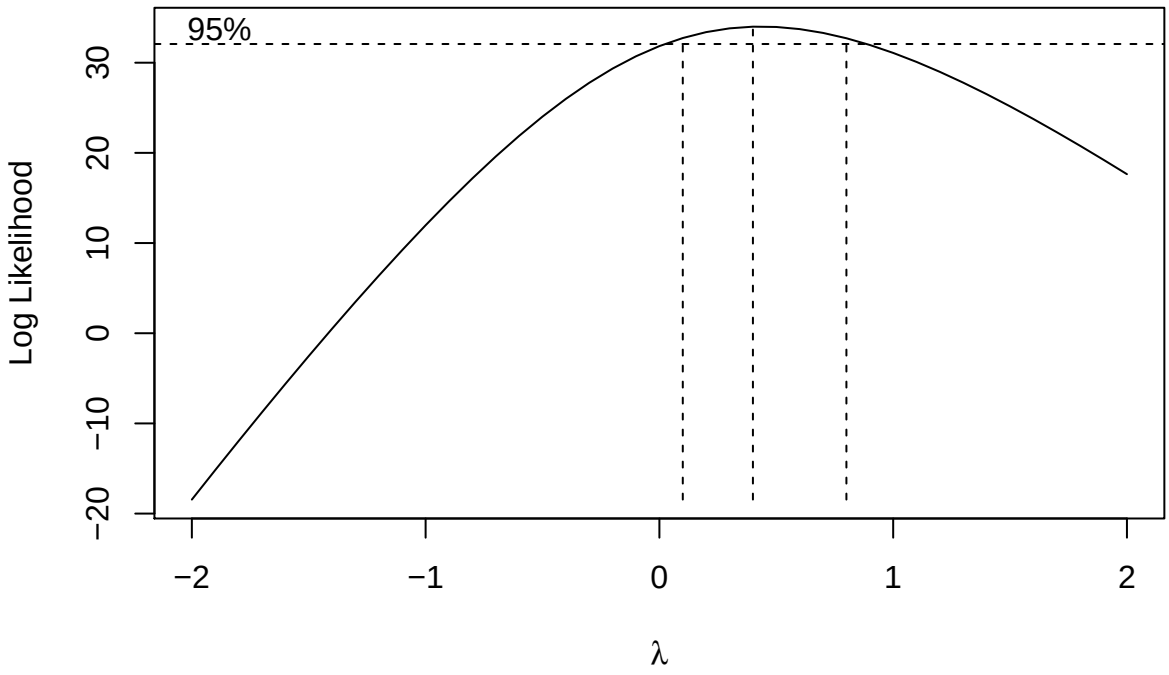
- Plot has visual positive trend
- Small dataset, so it is hard to visually observe any significant seasonality
- ACF and PACF have a waveform (similar to sin/cosine wave)
- Has a clear lag of 1 in ACF and PACF

Confidence interval

```
# check the confidence interval of BoxCox
fig_nums("confidence-interval", paste('Confidence Interval for ',
                                      default_ylab, sep = '')) %>% cat()
```

```
## Figure  2: Confidence Interval for Lake Huron Bloaters
```

```
boxcoxCi <- BoxCox.ar(data.ts, method = "yule-walker")$ci
```

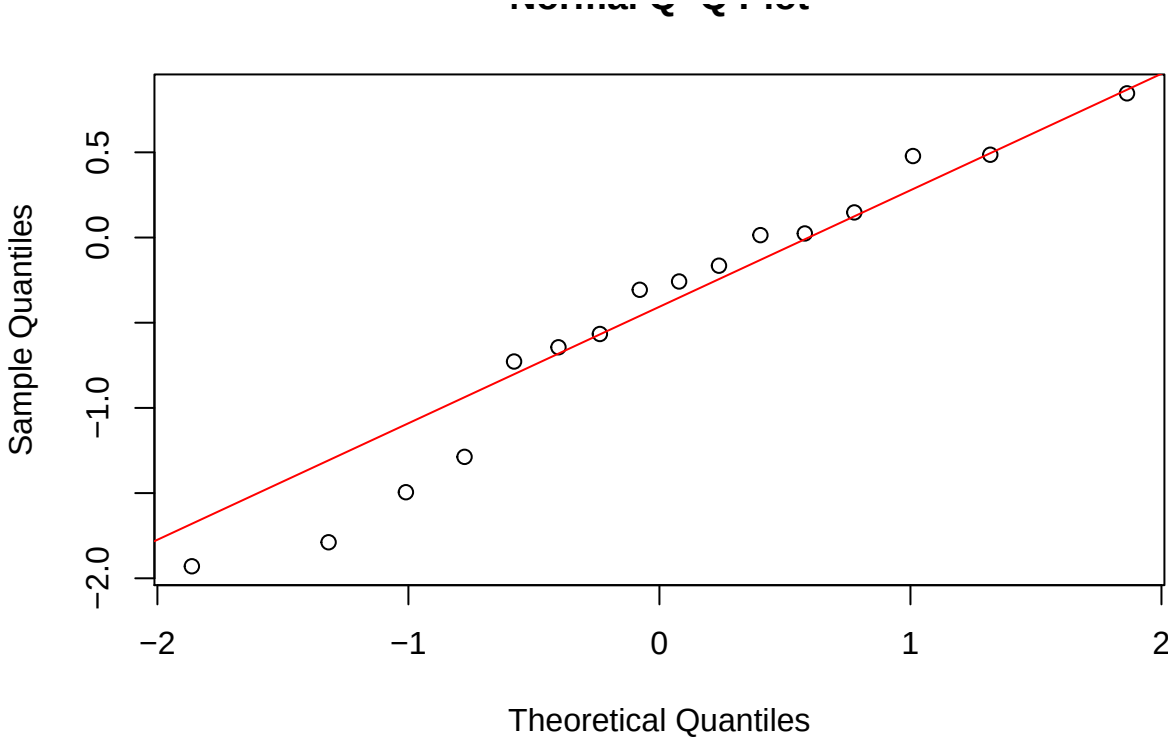


- The confidence interval `0.1 - 0.8` does not include 0, so we *cannot* do a log transform

```
doTestNormality <- function(df) {
  qqnorm(df)
  qqline(df, col = 2)
  shapiro.test(df)
}
lambda <- (max(boxcoxCi)-min(boxcoxCi))/2
# lambda=0.35 ~midpoint betwen confidence interval (0.8-0.1)/2
data.ts.boxcox = (data.ts^lambda-1) / lambda
fig_nums("test-normality", paste('Test Normality of ', default_ylab, sep = '')) %>% cat()
```

```
## Figure  3: Test Normality of Lake Huron Bloaters
```

```
doTestNormality(data.ts.boxcox)
```




```
##
##  Shapiro-Wilk normality test
##
## data:  df
## W = 0.95345, p-value = 0.5463
```

The Box-Cox transformation **did** help to improve the normality of the series because: - the dots are moreso aligned with the red line in the QQ plot - However, it isn't a perfect fit with QQ between ~(-2:0.5) does not fitting the QQ plot line - The Shapiro test p-value(0.5463) > 0.05 - Possible intonation point or bimodal data in the data

Diff

Without Box-Cox

```
doDiffAndPlot(data.ts, 1, F, F) # p = 0.3601 no lags, p is insignificant
```

```
## [1] "diff:  1"
## [1] "adf p-value: 0.360073649761517 > 0.05 insignificant"
```

```
doDiffAndPlot(data.ts, 2, F) # p = 0.1643 lag = 4 p is insignificant
```

```
## [1] "diff:  2"
## [1] "adf p-value: 0.164314242031066 > 0.05 insignificant"
```

```
doDiffAndPlot(data.ts, 3, F, F) # p = 0.3768 lag = 1, 4 = p is significant
```

```
## [1] "diff:  3"
## [1] "adf p-value: 0.376694692418674 > 0.05 insignificant"
```

- Our third diff doesn't add any further value
- Best is second diff with p 0.1643
- EACF We can't see any meaningful shelf, which indicates white noise series and possible non-stationarity

Diff

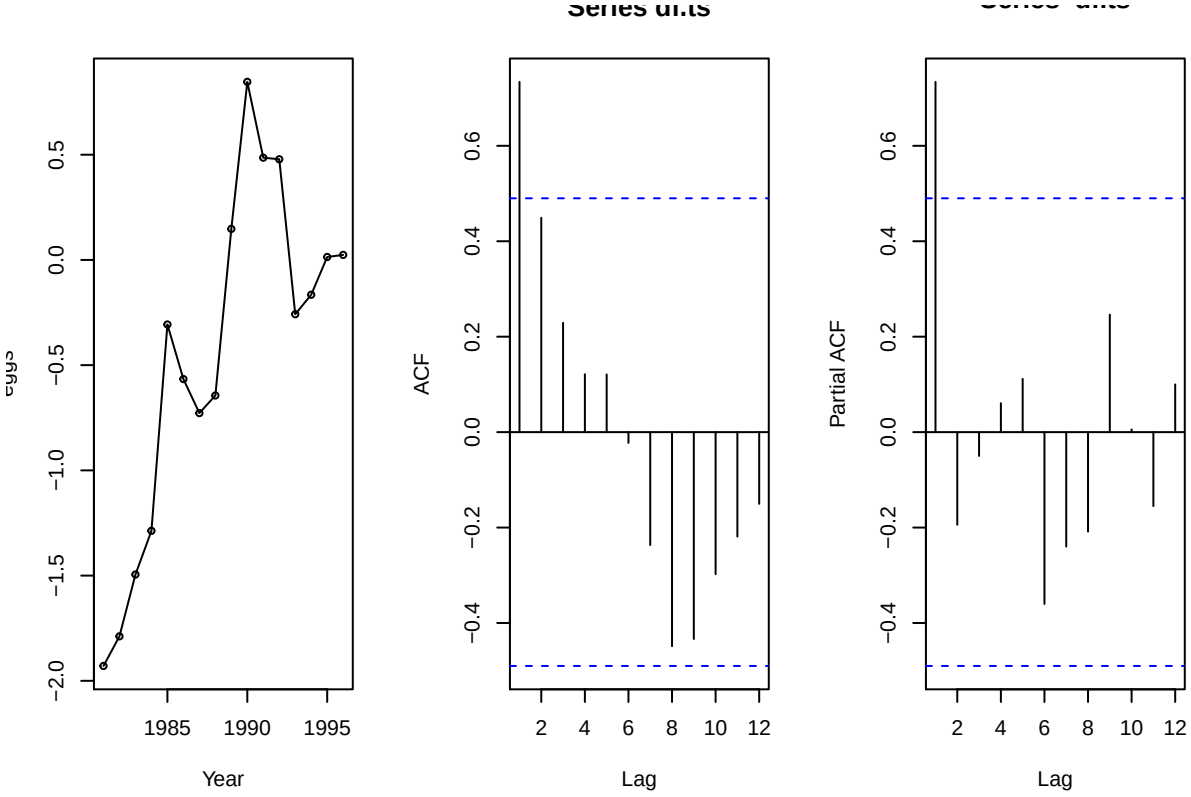
With Box-Cox

```
fig_nums('boxCox diff 0', 'BoxCox without any diff') %>% cat()
```

```
## Figure  4: BoxCox without any diff
```

```
doDiffAndPlot(data.ts.boxcox, 0, T) # p < 0.05 = 0.01941, lag = 1 but it still has trend
```

```
## [1] "diff:  0"
## [1] "adf p-value: 0.0224250337343733 < 0.05 significant"
```




```
doDiffAndPlot(data.ts.boxcox, 1, F) # p is higher 0.3282, better trend! Looks stationary
```

```
## [1] "diff:  1"
## [1] "adf p-value: 0.328249580787965 > 0.05 insignificant"
```

```
doDiffAndPlot(data.ts.boxcox, 2, F) # p is better 0.09166 pacf lag of 1 @4 pacf
```

```
## [1] "diff:  2"
## [1] "adf p-value: 0.0916630459959721 > 0.05 insignificant"
```

```
doDiffAndPlot(data.ts.boxcox, 3, F) # p is better 0.14 pacf lag of 1 @4 pacf
```

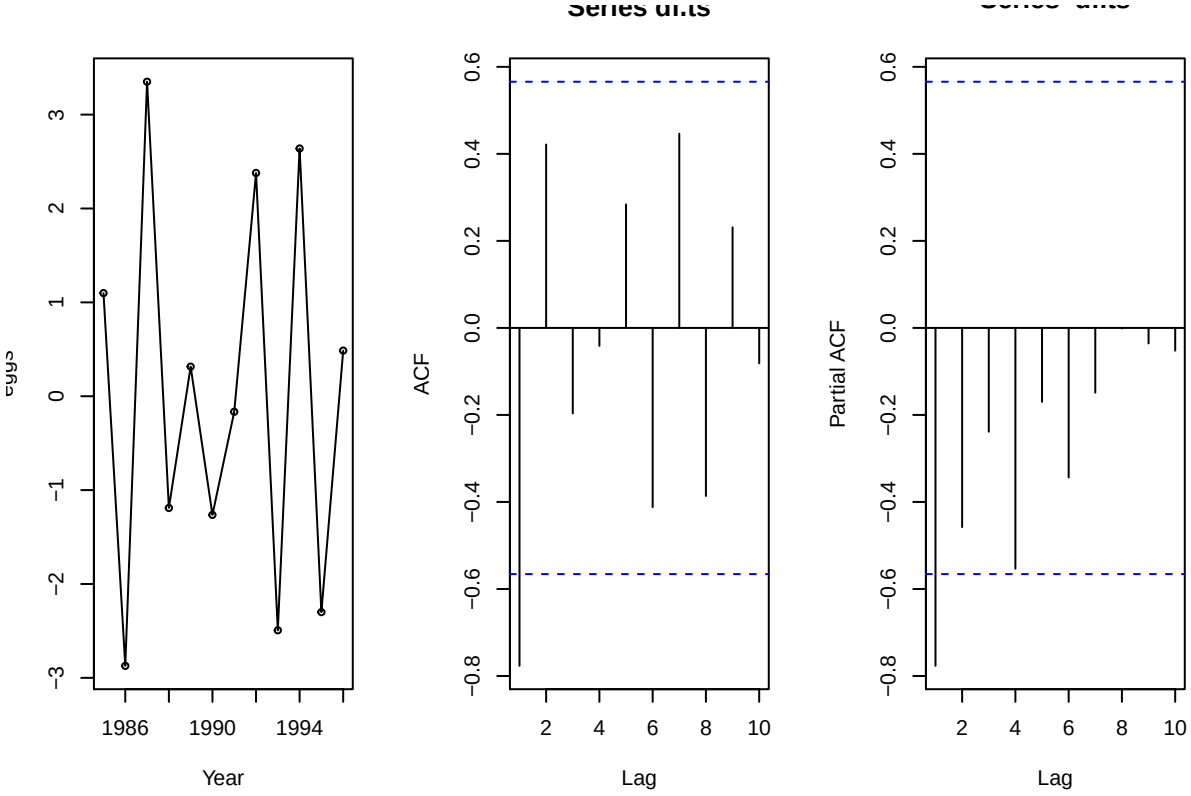
```
## [1] "diff:  3"
## [1] "adf p-value: 0.141211776292504 > 0.05 insignificant"
```

```
fig_nums('boxCox diff 4', 'BoxCox with diff of 4') %>% cat()
```

```
## Figure  5: BoxCox with diff of 4
```

```
doDiffAndPlot(data.ts.boxcox, 4, T, F) # p is better 0.02 pacf lag of 1 @4 pacf
```

```
## [1] "diff:  4"
## [1] "adf p-value: 0.0199702709567182 < 0.05 significant"
```

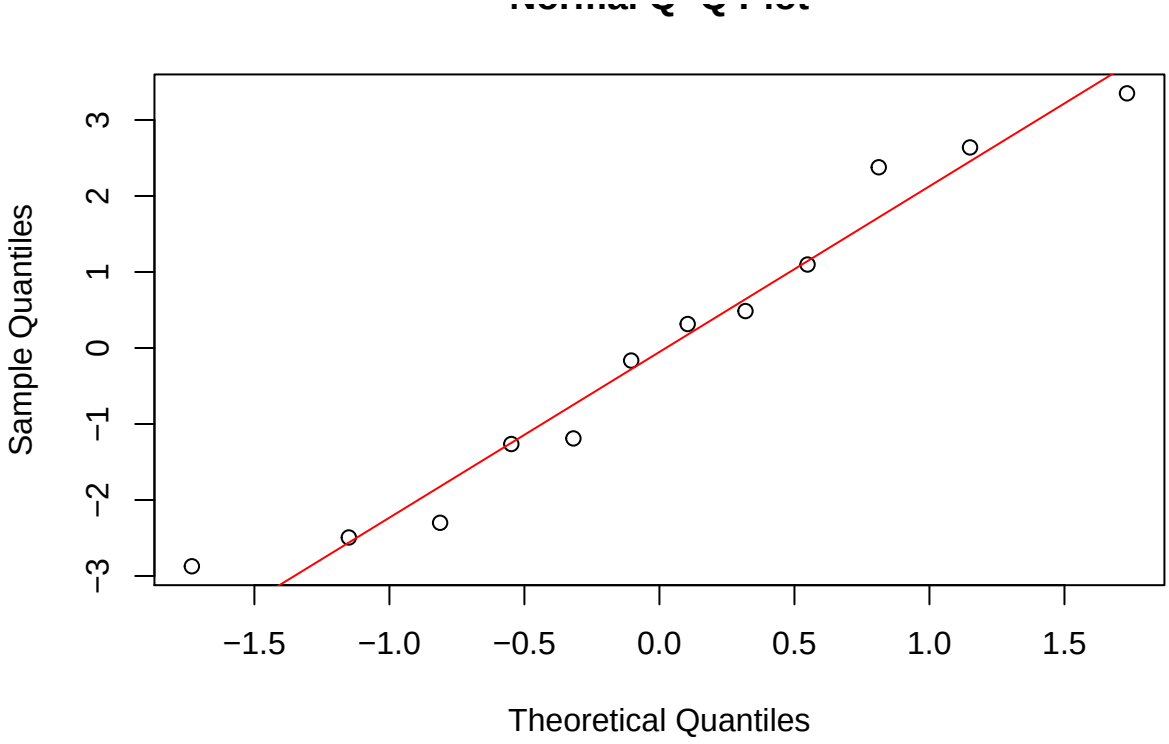


- Box-Cox transform has significance `p-value < 0.05 = 0.01941`
- Box-Cox transform with 4 differences has significance `p-value < 0.05 = 0.01997`
- Box-Cox with no difference still shows a trend **Therefore we select Box-Cox with a diff(4)**

```
# set variable for model testing:
data.ts.boxcox4 <-diff(data.ts.boxcox, difference = 4)
fig_nums('Normal Q-Q Plot test', 'Normal Q-Q Plot test') %>% cat()
```

```
## Figure  6: Normal Q-Q Plot test
```

```
doTestNormality(data.ts.boxcox4)
```




```
##
##  Shapiro-Wilk normality test
##
## data:  df
## W = 0.94709, p-value = 0.5948
```

The Box-Cox transformation **did** help to improve the normality of the series because: - the dots are more aligned with the red line in the QQ plot (than solely with the Box-Cox transformation) - the Shapiro test p-value(0.5948) > 0.05

From ACF and PACF: `{arima(p,d,q)} = {arima(1,4,1)}`

```
fig_nums('Eacf test', 'Eacf test') %>% cat()
```

```
## Figure  7: Eacf test
```

```
eacf(data.ts.boxcox, ar.max=3, ma.max=3)
```

```
## AR/MA
##   0 1 2 3
## 0 x o o o
## 1 o o o o
## 2 o o o o
## 3 o o o o
```

From the eacf plot we can select: - `{arima(0,4,1)}` - `{arima(1,4,0)}`

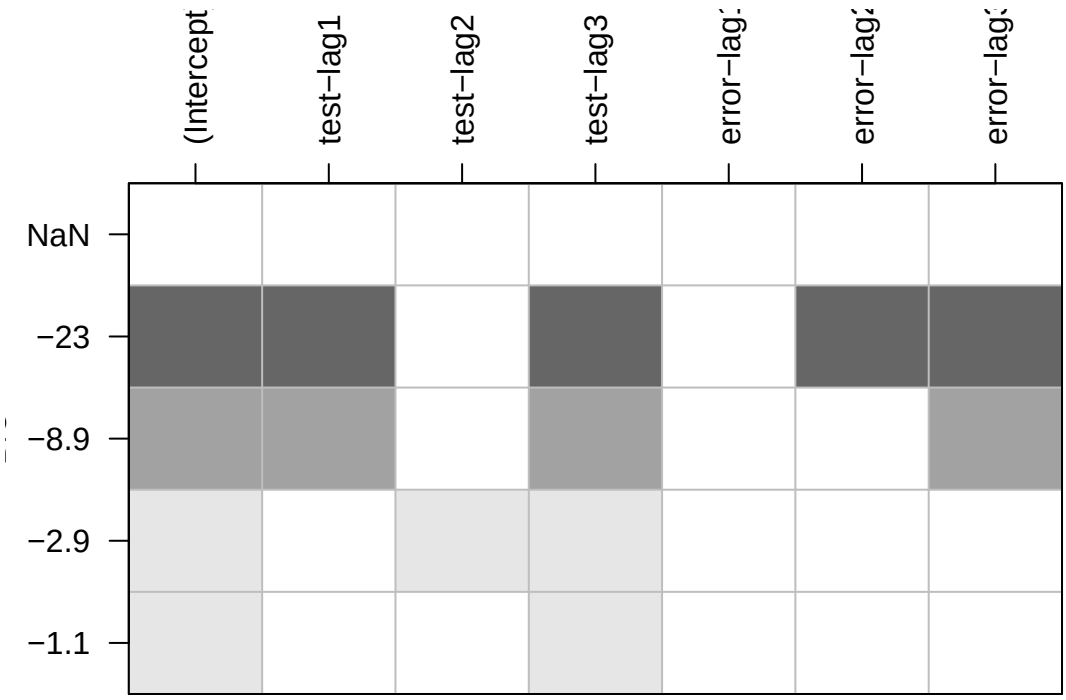
We can also select the following values which are close to the shelf: - `{arima(1,4,1)}` - `{arima(2,4,0)}`

Residuals: `BIC` Table

```
fig_nums('Residual BIC Table', 'Residual BIC table') %>% cat()
```

```
## Figure  8: Residual BIC table
```

```
par(mfrow=c(1,1))
armasubsets(y=data.ts,nar=3,nma=3,y.name='test',ar.method='ols') %>% plot()
```



From the BIC residual plot, we can also extract the models: {arima(1,4,2)}, {arima(1,4,1)}, {arima(0,4,2)}, {arima(0,4,1)} Note: arima(0,4,1) is duplicated in eacf plot

The final set of possible models {arima(p,d,q)} is: {arima(1,4,1), arima(0,4,1), arima(1,4,0), arima(1,4,2)}, {arima(0,4,2)}, {arima(2,4,0)}

Arima Model

```
getModelCoef <- function(pdq) {
  cat('\nmodel: arima(', pdq, ')\n')
  methods=c('CSS','ML')
  for (i in methods) {
    cat(i, '\n')
    model = arima(data.ts,order=pdq,method=i)
    coef = coeftest(model)
    modelName <- paste("model", i, sep = "")
    modelToScore <- assign(modelName, model)
    totalResultLines <- pdq[1] + pdq[3]
    startResult <- totalResultLines*3# because coeftest() returns a s4 object
    pValues <- 1:totalResultLines %>%
      map(~ {coeftest(model)[(.x + startResult)] %>% round(6) %>% paste()})
    isPSignifcant <- function(p) {
      ifelse(p < 0.05, 'p < 0.05 significant', 'p > 0.05 insignificant')
    }
    pValues %>% rbind(isPSignifcant(pValues), '\n') %>% paste() %>% cat()
  }
}
cat('Arima Model coeff p-values (rounded to 6 decimal places):\n')
```

```
## Arima Model coeff p-values (rounded to 6 decimal places):
```

```
model_041_p <- getModelCoef(pdq=c(0,4,1))
```

```
##
## model: arima( 0 4 1 )
## CSS
## 0 p < 0.05 significant
## ML
## 2e-06 p < 0.05 significant
```

```
model_042_p <- getModelCoef(pdq=c(0,4,2))
```

```
##
## model: arima( 0 4 2 )
## CSS
## 0 p < 0.05 significant
##  0 p < 0.05 significant
## ML
## 0 p < 0.05 significant
##  0.003161 p < 0.05 significant
```

```
model_140_p <- getModelCoef(pdq=c(1,4,0))
```

```
##
## model: arima( 1 4 0 )
## CSS
## 1e-06 p < 0.05 significant
## ML
## 0 p < 0.05 significant
```

```
model_141_p <- getModelCoef(pdq=c(1,4,1))
```

```
##
## model: arima( 1 4 1 )
## CSS
## 0.001172 p < 0.05 significant
##  0 p < 0.05 significant
## ML
## 0.001061 p < 0.05 significant
##  4.3e-05 p < 0.05 significant
```

```
model_142_p <- getModelCoef(pdq=c(1,4,2))
```

```
##
## model: arima( 1 4 2 )
## CSS
## 0.147627 p > 0.05 insignificant
##  0.007886 p < 0.05 significant
##  0.429623 p > 0.05 insignificant
## ML
## 0.239717 p > 0.05 insignificant
##  5.1e-05 p < 0.05 significant
##  0.056878 p > 0.05 insignificant
```

```
model_240_p <- getModelCoef(pdq=c(2,4,0))
```

```
##
## model: arima( 2 4 0 )
## CSS
## 1e-06 p < 0.05 significant
##  0.045272 p < 0.05 significant
## ML
## 2e-06 p < 0.05 significant
##  0.053114 p > 0.05 insignificant
```

```
model_242_p <- getModelCoef(pdq=c(2,4,1))
```

```
##
## model: arima( 2 4 1 )
## CSS
## 0.000637 p < 0.05 significant
##  0.215751 p > 0.05 insignificant
##  5e-06 p < 0.05 significant
## ML
## 0.005466 p < 0.05 significant
##  0.456372 p > 0.05 insignificant
##  0.00017 p < 0.05 significant
```

- arima(1,4,2), arima(2,4,1), arima(2,4,0) show insignificant p-values > 0.05,
- These are no longer models we will consider
- The remaining models are: `model_041, model_042, model_140, model_141`

```
model_041_ml = arima(data.ts,order=c(0,4,1),method='ML') # ARIMA(0,4,1)
model_042_ml = arima(data.ts,order=c(0,4,2),method='ML') # ARIMA(0,4,2)
model_140_ml = arima(data.ts,order=c(1,4,0),method='ML') # ARIMA(1,4,0)
model_141_ml = arima(data.ts,order=c(1,4,1),method='ML') # ARIMA(1,4,1)

# AIC and BIC values
sort.score(AIC(
  model_041_ml, model_042_ml, model_140_ml, model_141_ml ),
  score = "aic")
```

```
##              df      AIC
## model_042_ml  3 42.84286
## model_141_ml  3 44.37815
## model_140_ml  2 48.87321
## model_041_ml  2 49.07553
```

```
sort.score(BIC(
  model_041_ml, model_042_ml, model_140_ml, model_141_ml ),
  score = "bic")
```

```
##              df      BIC
## model_042_ml  3 44.29758
## model_141_ml  3 45.83287
## model_140_ml  2 49.84302
## model_041_ml  2 50.04535
```

- arimia(0,4,2) ranks highest according to he `sort.score` formula in both AIC and BIC

```r
# arima(0,4,2) entire coeficient's output for for completeness:
arima(data.ts,order=c(0,4,2),method='ML') %>% coeftest()
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -1.86751    0.32411 -5.7619 8.317e-09 ***
## ma2  0.94443    0.31996  2.9517  0.003161 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residuals

```r
residual.analysis <- function(model, std = TRUE){
  if (std == TRUE){
    res.model = rstandard(model)
  }else{
    res.model = residuals(model)
  }
  par(mfrow=c(3,2),
      oma = c(1,1,0,0) + 0.1,
      mar = c(2,2,2,2) + 0.1)
  plot(
    res.model,
    type='o',
    xlab='years',
    ylab='Standardised residuals',
    main="Time series plot of standardised residuals")
  abline(h=0)
  hist(res.model,main="Histogram of standardised residuals")
  qqnorm(res.model,main="QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  acf(res.model,main="ACF of standardised residuals")
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = length(model$residuals)-1 ,
          StartLag = k + 1, k = 0, SquaredQ = FALSE)
  #resPlotsTitle <- paste('Figure', figureCount, '. Standardised residuals of ', default_ylab, sep = ''
  #title(resPlotsTitle, side = 3, line = -33, outer = TRUE, cex.adj = 3)
  #figureCount <- figureCount + 1
  par(mfrow=c(1,1))
}
fig_nums('Residual Analysis', 'Residual Analysis') %>% cat()
```

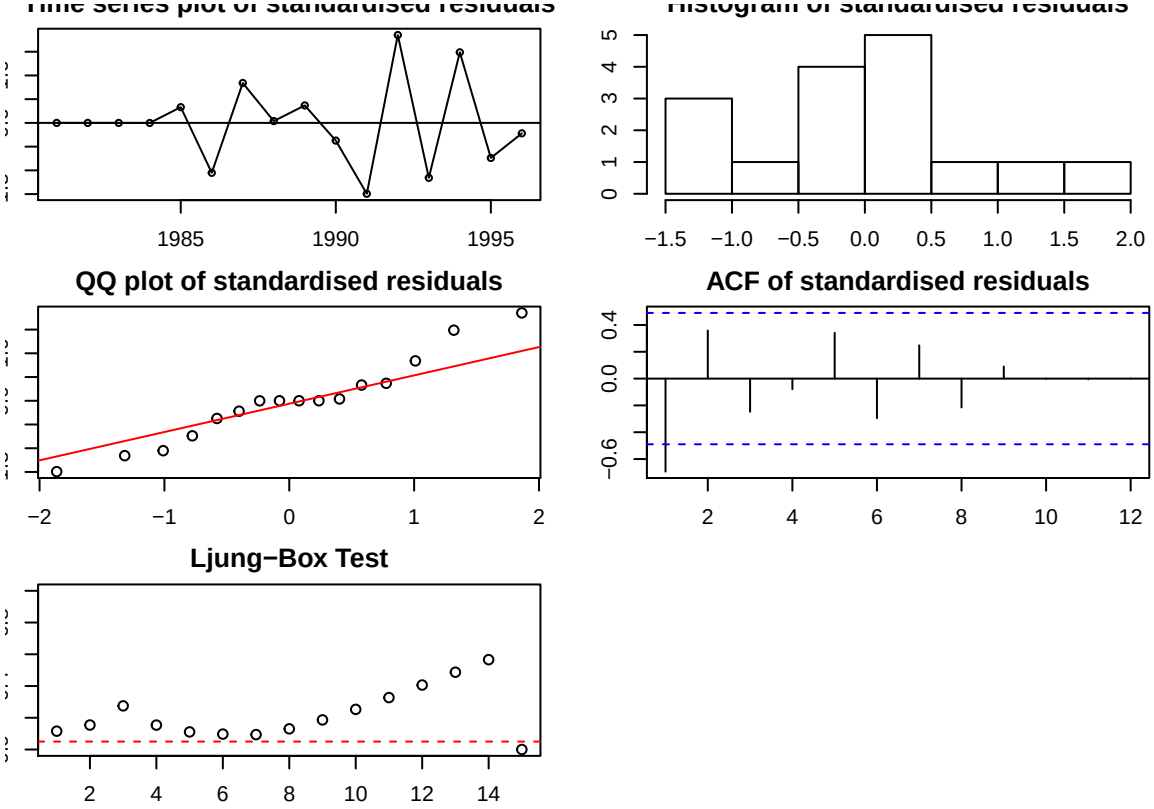
```
## Figure  9: Residual Analysis
```

Residuals for arima(0,4,1)

```r
residual.analysis(model = model_041_ml)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  res.model
## W = 0.95241, p-value = 0.5288
```



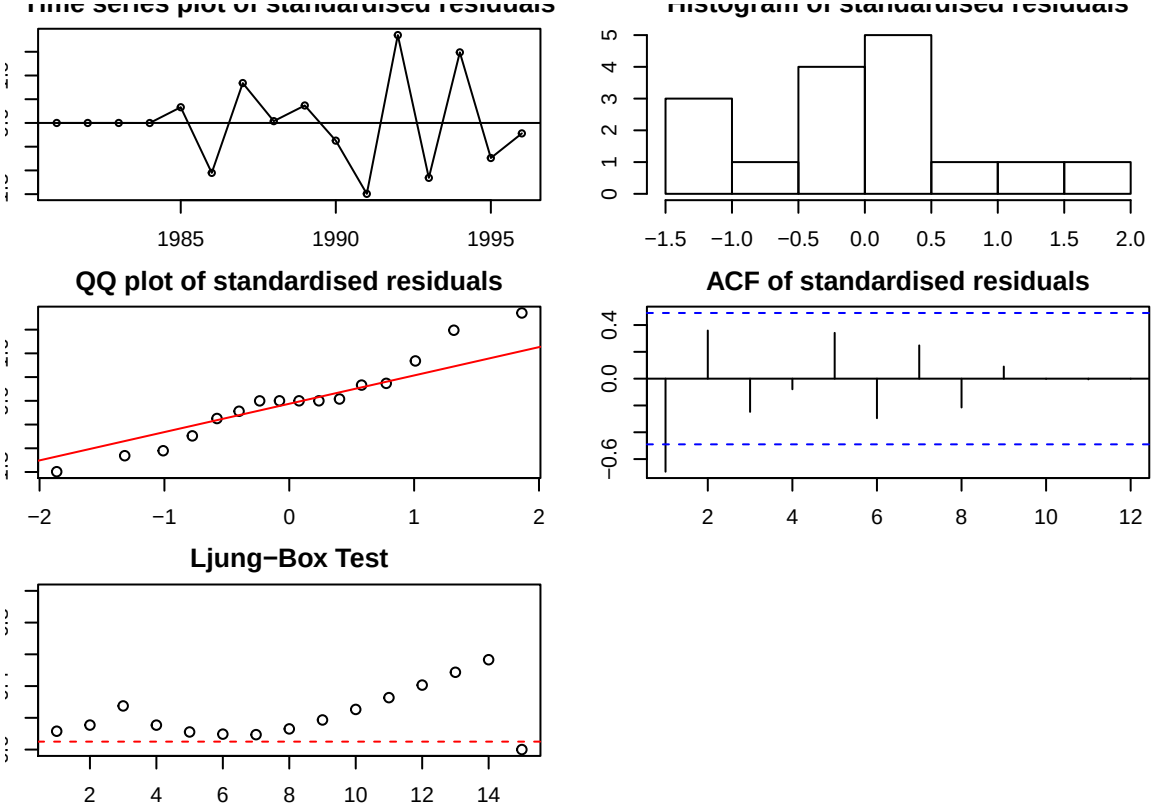
- Time-series standardised residuals: cannot observe a trend
- histogram: data is not normal and left skewed
- Ljung-Box test: One dots below red line
- Q-Q plot: tailed, doesn't fit
- ACF: lag of 1, indicates we have not modelled all of the data
- Shapiro-Wilk: significant

ACF and Ljung-Box shows a timeseries patern not captured in this model

Residuals for arima(0,4,1)

```
residual.analysis(model = model_041_ml)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.95241, p-value = 0.5288
```



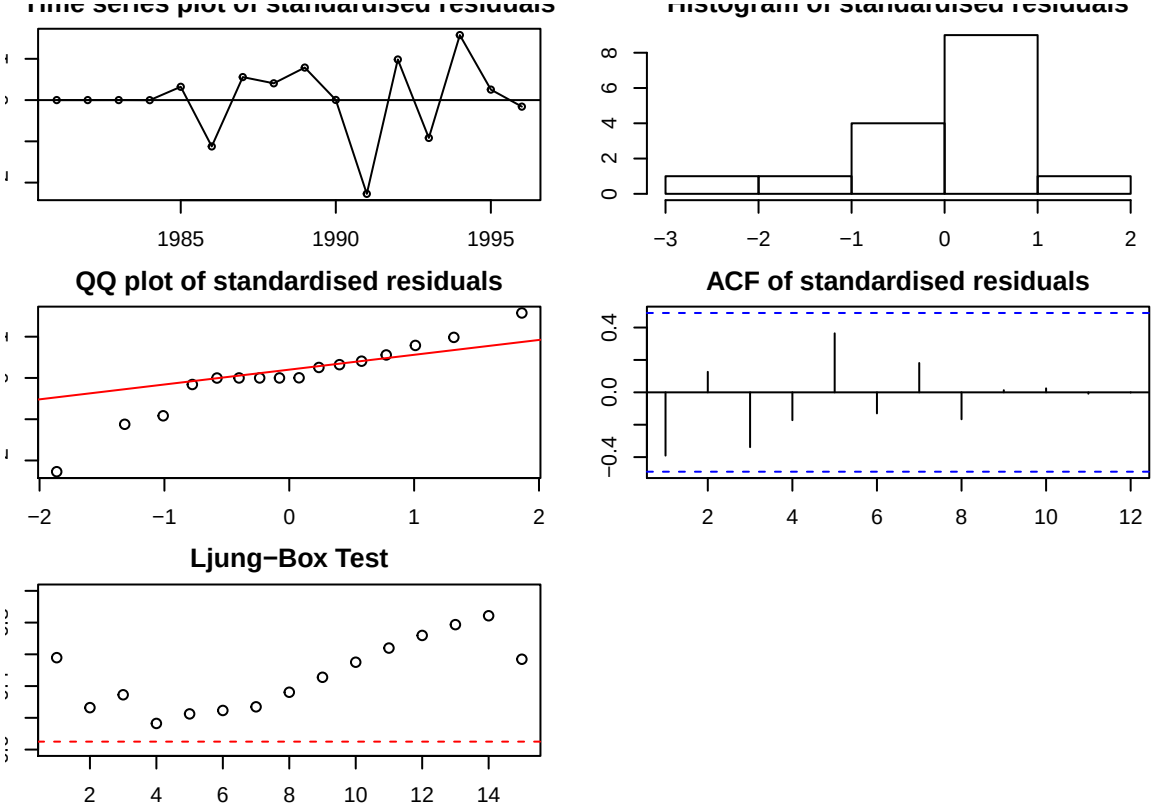
- Time-series standardised residuals: cannot observe a trend
- histogram: data is not normal and left skewed, possibly bimodal
- Ljung-Box test: One dots below red line
- Q-Q plot: tailed, doesn't fit
- ACF: lag of 1, indicates we have not modelled all of the data
- Shapiro-Wilk: significant

ACF and Ljung-Box shows a timeseries patern not captured in this model

Residuals for arima(0,4,2)

```
residual.analysis(model = model_042_ml)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.91548, p-value = 0.1426
```



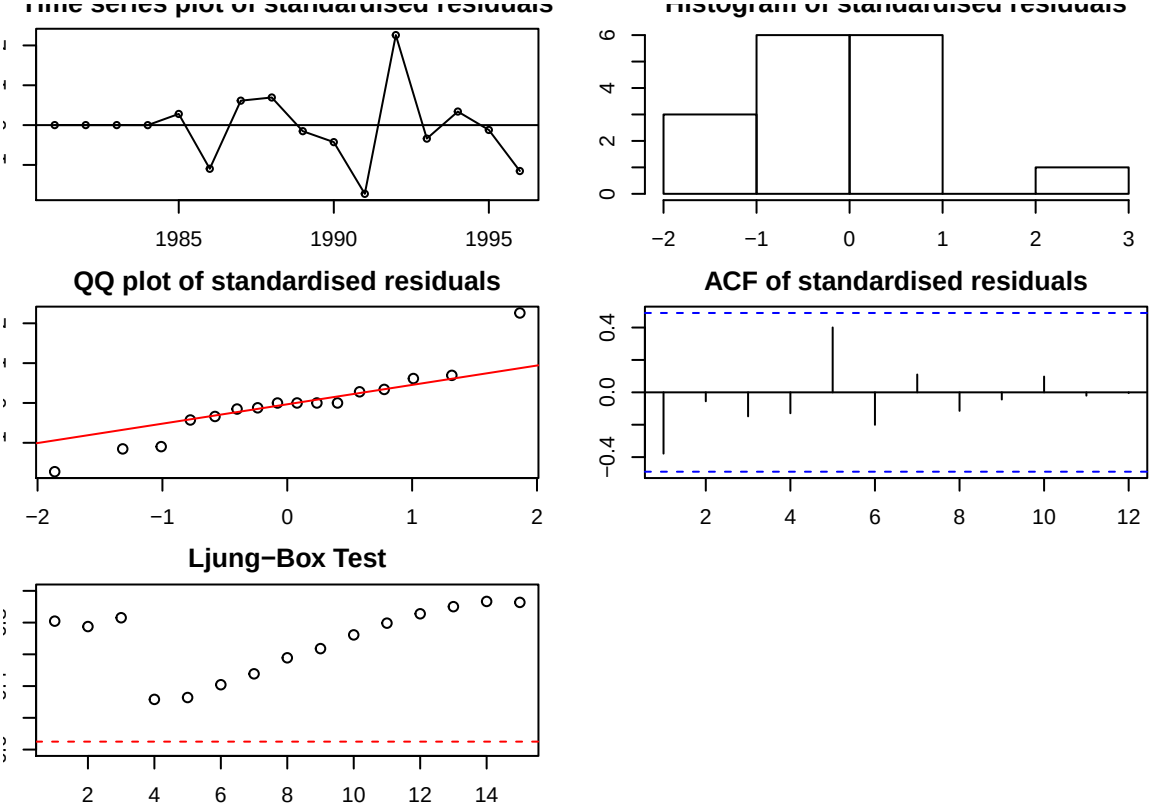
- Time-series standardised residuals: cannot observe a trend
- histogram: data is not normal and left skewed
- Ljung-Box test: all dots > red line
- Q-Q plot: left-tailed, doesn't fit
- ACF: no lags, indicating white noise
- Shapiro-Wilk: significant

We cannot find any meaningful timeseries observations from the residuals, indicating a good model fit

Residuals for arima(1,4,0)

```
residual.analysis(model = model_140_ml)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.9185, p-value = 0.1595
```



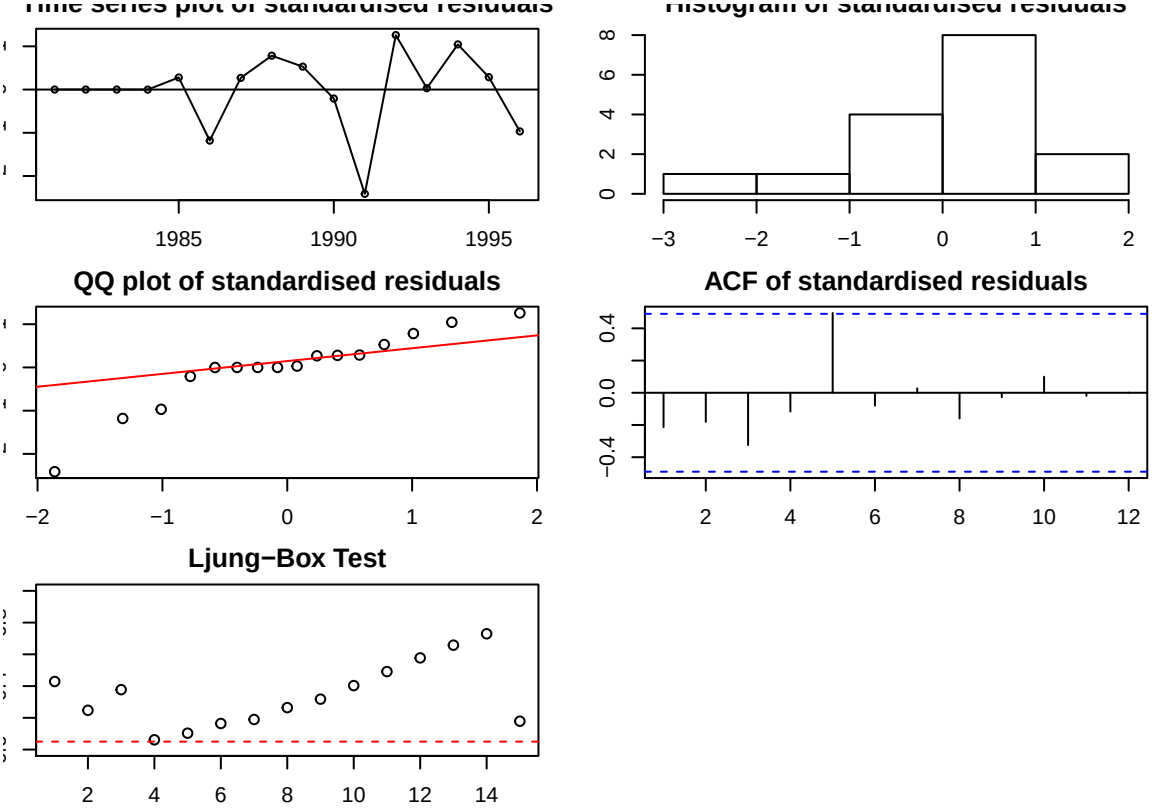
- Time-series standardised residuals: cannot observe a trend
- histogram: data is not normal and left skewed
- Ljung-Box test: all dots > red line
- Q-Q plot: left-tailed, doesn't fit
- ACF: no lags, indicating white noise
- Shapiro-Wilk: significant

We cannot find any meaningful timeseries observations from the residuals, indicating a good model fit

Residuals for arima(1,4,1)

```
residual.analysis(model = model_141_ml)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.8858, p-value = 0.04783
```



- Time-series standardised residuals: cannot observe a trend
- histogram: data is not normal and left skewed
- Ljung-Box test: one dot on red line
- Q-Q plot: left-tailed, doesn't fit
- ACF: Possible lag at 5, may not be significant
- Shapiro-Wilk: insignificant

ACF and Shapiro-Wilk shows a timeseries patern not captured in this model

Residual analysis conclusion

Residuals are not apparent in `arima(1,4,0)` and `arima(0,4,2)`

Forecast

```
library(forecast)
```

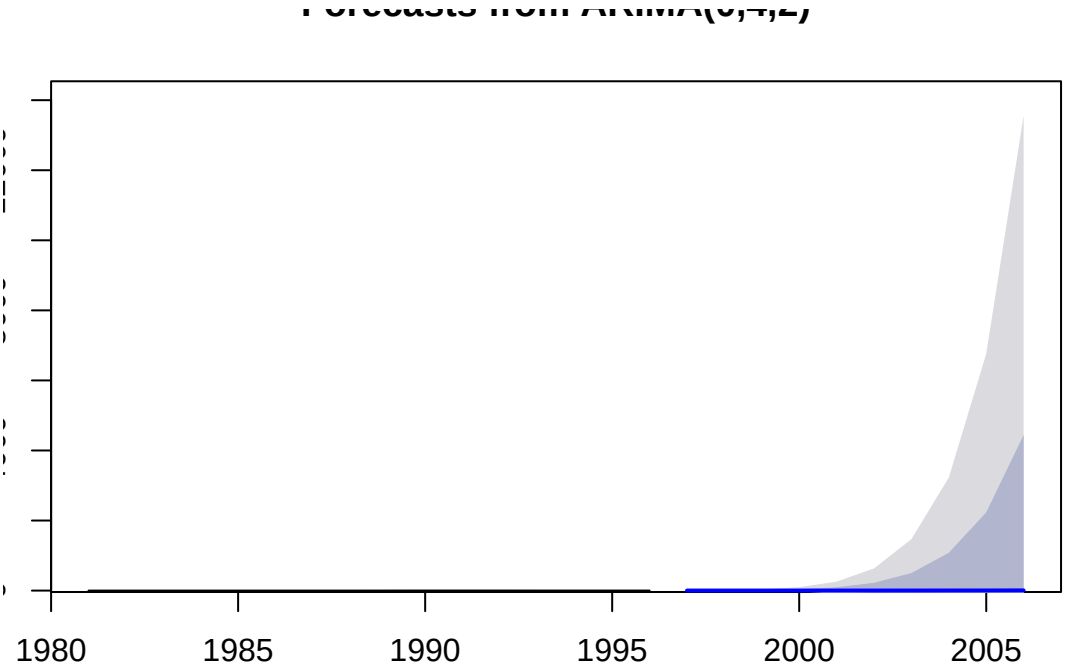
```
## Warning: package 'forecast' was built under R version 3.4.4
```

```
figureName <- fig_nums('Forecast of growth', 'Forecast of growth') %>% cat()
```

```
## Figure  10: Forecast of growth
```

```
# lambda of 0.35 is from Box-Cox transformation:
fit = Arima(data.ts.raw,c(0,4,2), lambda = lambda)
# hack to hide negative values:
ylim <- c(500, 14000)
plot(
  forecast(fit,h=10),
   ylim=ylim,
```

```
    ylab=figureName
    )
```



- The forecast shows an almost exponential positive trend in growth, with quite a wide confidence interval

NOTE: negative trend has been hidden on this plot, because it's not possible for fish to lay a negative number of eggs. This is a plot side-effect generated because lambda doesn't equal 0

Conclusion

The results of the above tests summarised:

- `p-value(0.02) < 0.05` in our adfTest
- `ma2 and ma1 p < 0.05` in coefficient test
- Residuals are not apparent in `arima(1,4,0)` and `arima(0,4,2)`

The AIC and BIC score for is higher for `arima(0,4,2)`, which is why we have selected `{arima(0,4,2)}` as our best fitting time-series model

Please note that the dataset is quite small, so we have limited most of our decisions to programatic output

The forcast shows a possible 3,000,000,000 eggs laid in 2006. Given the dataset is limited to 16 values, a more detailed dataset should be examined to check accuracy of fit. The model may also be adjusted because of the scarcity of food and high mortality rate of larval Bloaters

If the model still fits, recommendations include investing in the cavier industry in Lake Huron