Forecasting ASX Monthly Price

Alex Hudson

Introduction

The aim of this report will be to investigate Australian Stock Exchange (ASX) price data and develop appropriate forecasts. Firstly, the data series and related data series will be analised to gain further understanding of them. This analysis will include examination of autocorrelation, stationarity, effects of transformations to reduce changing variance and nonstationarity and examination of seasonality. Secondly, models will be fit to the ASX price data using related time series as independent variables. The best models will be decided based on evaluation metrics, with it being plotted as a forecast alongside the best of each type of model. The data used includes the price of the ASX, of gold, of oil by barrel and of copper per ton by month from January 2004 to May 2017.

```
knitr::opts_chunk$set(
    echo = TRUE,
    message = FALSE,
    warning = FALSE
)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(expsmooth)
library(TSA)
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

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

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

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

```r
library(urca)
library(dynlm)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
library(x12)
```

```
## Loading required package: x13binary

## x12 is ready to use.

## Use the package x12GUI for a Graphical User Interface.

## By default the X13-ARIMA-SEATS binaries provided by the R package x13binary

## are used but this can be changed with x12path(validpath)

## ---------------

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/x12/issues
```

```r
library(tseries)
library(dLagM)
```

```
## Loading required package: nardl

##
## Attaching package: 'dLagM'

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

```r
library(stringr)
library(car)
```

```
## Loading required package: carData
```

```
asx.data <- read.csv("~/Projects/ASX Prediction/data/ASX_data.csv")
# gold and copper had a comma to demonstrate the thousands unit, this needed
# to be removed and the data needed to be changed into a numeric vector
asx.data$Gold.price <- as.numeric(str_remove(asx.data$Gold.price, ","))
asx.data$Copper_USD.tonne <- as.numeric(str_remove(asx.data$Copper_USD.tonne, ","))


asx.all <- ts(asx.data, start = c(2004, 1), frequency = 12)
colnames(asx.all) <- c("asx.price", "gold.price", "oil.price", "copper.price")

asx <- ts(asx.data$ASX.price, start = c(2004, 1), frequency = 12)
gold <- ts(as.numeric(str_remove(asx.data$Gold.price, ",")), start = c(2004, 1), frequency = 12)
oil <- ts(asx.data$Crude.Oil..Brent._USD.bbl, start = c(2004, 1), frequency = 12)
copper <- ts(as.numeric(str_remove(asx.data$Copper_USD.tonne, ",")), start = c(2004, 1), frequency = 12
```

```
head(asx)
```

```
##         Jan    Feb    Mar    Apr    May    Jun
## 2004 2935.4 2778.4 2848.6 2970.9 2979.8 2999.7
```

```
class(asx)
```

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

Time Series Evaluation

From a visual inspection, all four variables appear to have an increasing trend. This is further demonstrated by the ACF plots of each. Finally, the ADF test for each failed to reject the null, meaning that they are nonstationary. This means that the data needs to be differenced to make it more stationary before the data analysis can proceed.

The visual inspection for each time series will examine trend, seasonality changing variance, presence of moving average or autoregressive behaviour and intervention points.

ASX

- The series appears to positive trend with little changing variance throughout the series with a possible intervention point around 2008. Additionally, it demonstrates mostly autoregressive behaviour and seasonality may be present.

Gold

- The series demonstrates a clear positive trend with little changing variance and no clear intervention point. It shows autoregressive behaviour and lack of strong seasonality

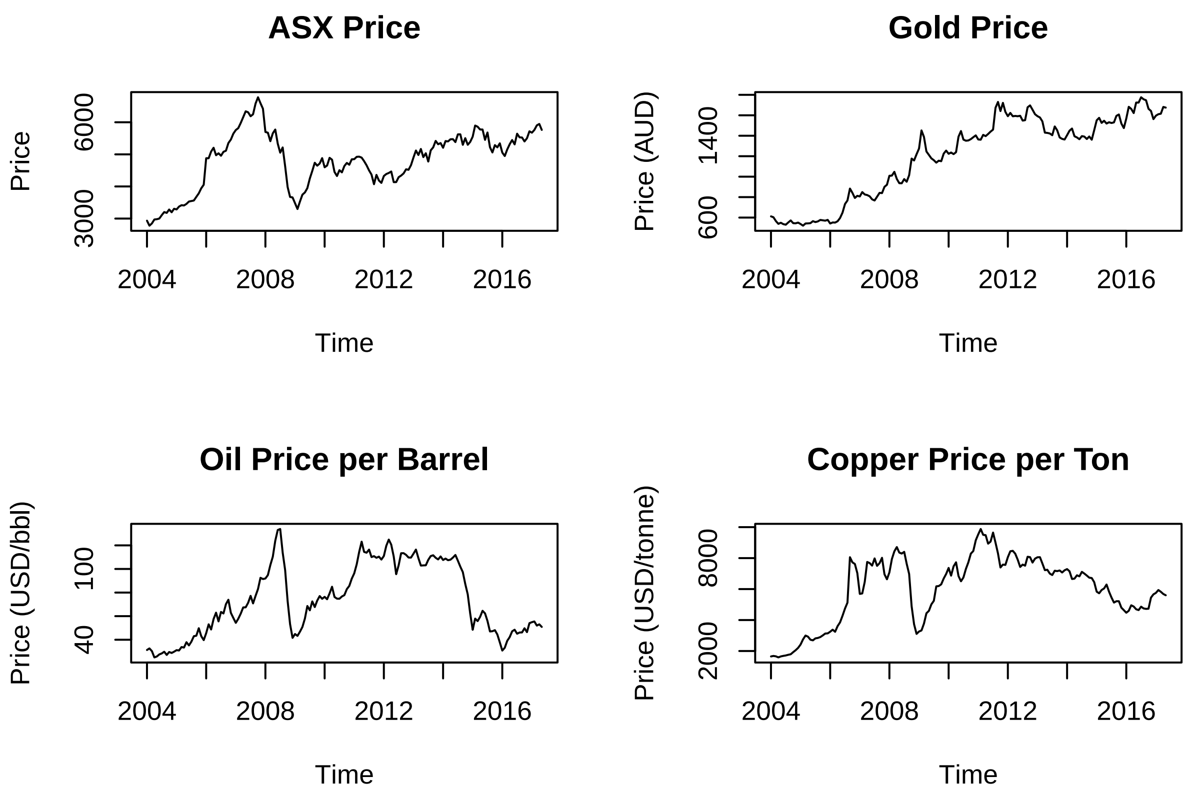
Oil

- The series may have a positive trend, although it is not clear. Little changing variance is seen but a clear intervention point is present around 2008. The series demonstrates autoregressive behaviour and possible seasonality.

Copper

- The series shows a slight positive trend with some changing variance andf an intervention point around 2008. It demonstrates autoregressive behaviour and lack of strong seasonality.

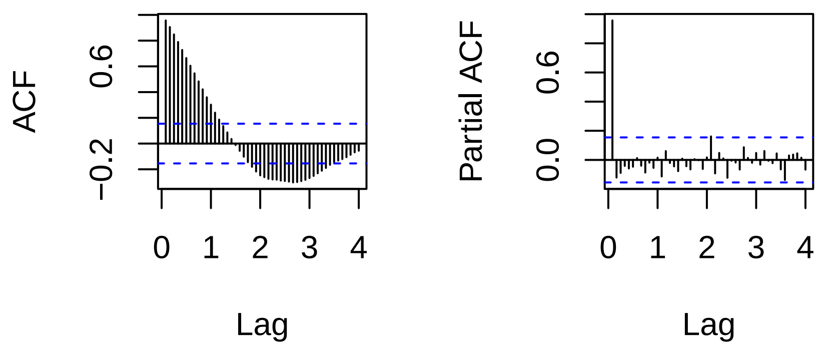
```
par(mfrow = c(2, 2))
plot(asx, main = "ASX Price", ylab = "Price")
plot(gold, main = "Gold Price", ylab = "Price (AUD)")
plot(oil, main = "Oil Price per Barrel", ylab = "Price (USD/bbl)")
plot(copper, main = "Copper Price per Ton", ylab = "Price (USD/tonne)")
```




```
acf.pacf <- function(data, name, figure){
  par(mfrow = c(1, 2))
  acf(data, lag.max = 48, main = NA)
  pacf(data, lag.max = 48, main = NA)
  title(paste0("\n", "Figure ", figure, ". ", name, " ACF and PACF"), outer = TRUE)
  par(mfrow = c(1, 1))
}
```

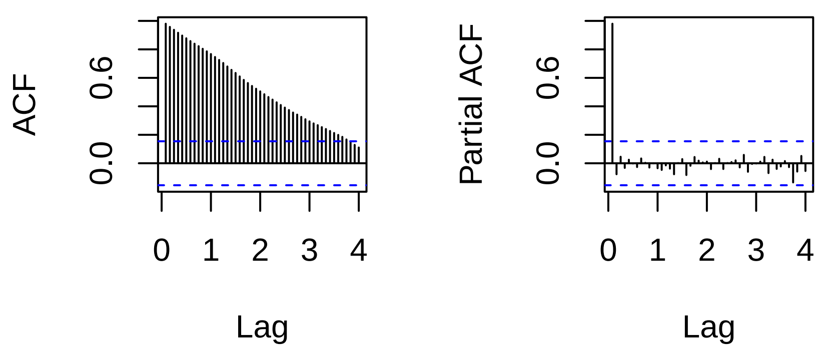
```
acf.pacf(asx, "ASX", 1)
```

Figure 1. ASX ACF and PACF



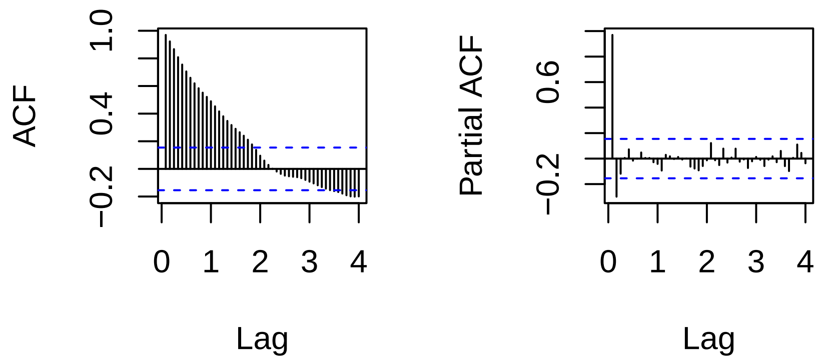

```r
acf.pacf(gold, "Gold", 2)
```

Figure 2. Gold ACF and PACF



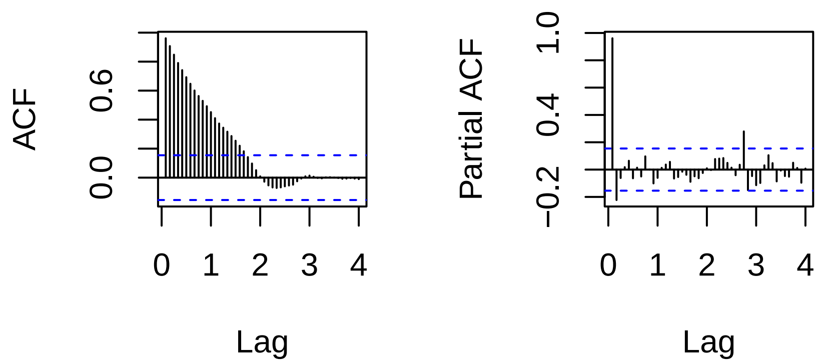

```r
acf.pacf(oil, "Oil", 3)
```

Figure 3. Oil ACF and PACF




```
acf.pacf(copper, "Copper", 4)
```

Figure 4. Copper ACF and PACF



Stationarity of Data

The stationarity was tested for the 4 time series, with all being described as nonstationary by the ADF and PP test. For the ASX series, the `adf.test()` from the `tseries` packaged produced a p-value of `0.285` which fails to reject the null hypothesis of nonstationarity, meaning it is nonstationary. Additional tests were performed from the `urca` package. The ADF test from this package produced a test-statistic of `0.453`, which was higher than the critical value at 1pct of `-2.58`, 5pct of `-1.95` and 10pct of `-1.62`. This means that the null hypothesis of nonstationarity was rejected. The PP test from this package returned a test-statistic of `-2.203`, with this being higher than the critical value at 5pct of `-2.880`. This means the null hypothesis of nonstationarity was not rejected and the series is nonstationary. Finally, the KPSS test was conducted, producing a test statistic of `0.873`. This was higher than the critical value at 5pct of `0.463`, meaning that the null hypothesis of stationarity was rejected and that the series is nonstationary. A similar analysis was undertaken for the gold, oil and copper series, finding nonstationarity in all three time series.

```r
checkstationarity <- function(data){
  print(adf.test(data))
  k = ar(data)$order
  adf <- ur.df(data, type = "none", lags = k, selectlags = "AIC")
  print(summary(adf))
  pp <- ur.pp(data, type = "Z-tau", lags = "short")
  print(summary(pp))
  kpss <- ur.kpss(data)
  print(summary(kpss))
}

checkstationarity(asx)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -2.6995, Lag order = 5, p-value = 0.2846
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -696.15 -108.92   34.02  133.30  792.06
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1     0.001504   0.003321   0.453    0.651
## z.diff.lag1 0.076549   0.080116   0.955    0.341
## z.diff.lag2 0.130992   0.079970   1.638    0.103
##
## Residual standard error: 202.7 on 155 degrees of freedom
## Multiple R-squared:  0.02775,    Adjusted R-squared:  0.008932
## F-statistic: 1.475 on 3 and 155 DF,  p-value: 0.2236
##
##
## Value of test-statistic is: 0.4529
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
```

```
## ###################################
## # Phillips-Perron Unit Root Test #
## ###################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -683.35 -104.89   15.86  135.94  782.18
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 191.38104   87.09761   2.197   0.0295 *
## y.l1          0.96386    0.01781  54.107   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 158 degrees of freedom
## Multiple R-squared:  0.9488, Adjusted R-squared:  0.9485
## F-statistic:  2928 on 1 and 158 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -2.2025
##
##          aux. Z statistics
## Z-tau-mu            2.3366
##
## Critical values for Z statistics:
##                       1pct      5pct     10pct
## critical values -3.472136 -2.879539 -2.576262
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.8725
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
checkstationarity(gold)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
```

```
## Dickey-Fuller = -1.8369, Lag order = 5, p-value = 0.6444
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -138.655  -26.586   -2.016   22.726  206.439
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    0.003218   0.003322   0.969    0.334
## z.diff.lag 0.122451   0.079603   1.538    0.126
##
## Residual standard error: 52.57 on 157 degrees of freedom
## Multiple R-squared:  0.02351,    Adjusted R-squared:  0.01107
## F-statistic:  1.89 on 2 and 157 DF,  p-value: 0.1545
##
##
## Value of test-statistic is: 0.9687
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -146.558  -28.725   -7.533   20.322  209.729
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.85676   13.11322   1.514    0.132
## y.l1         0.98898    0.01038  95.318   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.43 on 158 degrees of freedom
## Multiple R-squared:  0.9829, Adjusted R-squared:  0.9828
## F-statistic:  9085 on 1 and 158 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -1.0701
##
##          aux. Z statistics
## Z-tau-mu            1.5151
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.472136 -2.879539 -2.576262
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.9312
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
checkstationarity(oil)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -1.8523, Lag order = 5, p-value = 0.6379
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.943  -3.549   1.248   3.982  15.176
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.003793   0.005835  -0.650    0.517
## z.diff.lag  0.399011   0.073572   5.423  2.2e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.863 on 155 degrees of freedom
## Multiple R-squared:  0.1601, Adjusted R-squared:  0.1493
## F-statistic: 14.78 on 2 and 155 DF,  p-value: 1.339e-06
##
##
## Value of test-statistic is: -0.65
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.6715  -3.5936   0.4549   3.9455  14.3505
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.07981    1.32129   1.574    0.117
## y.l1         0.97347    0.01659  58.686   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.302 on 158 degrees of freedom
## Multiple R-squared:  0.9561, Adjusted R-squared:  0.9559
## F-statistic:  3444 on 1 and 158 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -2.0125
##
##          aux. Z statistics
## Z-tau-mu            1.9323
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.472136 -2.879539 -2.576262
##
##
```

```
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.8196
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
checkstationarity(copper)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -2.2502, Lag order = 5, p-value = 0.472
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1861.62  -167.44    35.95   248.39  2829.64
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.002248   0.005792  -0.388    0.698
## z.diff.lag  0.307584   0.076382   4.027  8.8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 464.6 on 156 degrees of freedom
## Multiple R-squared:  0.09423,    Adjusted R-squared:  0.08262
## F-statistic: 8.115 on 2 and 156 DF,  p-value: 0.000444
##
##
## Value of test-statistic is: -0.3881
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
```

```
##
## ###################################
## # Phillips-Perron Unit Root Test #
## ###################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.88  -213.60   -21.83   225.07  2876.96
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 255.00356  112.61252   2.264   0.0249 *
## y.l1          0.96156    0.01771  54.301   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 477.4 on 158 degrees of freedom
## Multiple R-squared:  0.9491, Adjusted R-squared:  0.9488
## F-statistic:  2949 on 1 and 158 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -2.3641
##
##          aux. Z statistics
## Z-tau-mu             2.406
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.472136 -2.879539 -2.576262
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 1.0012
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

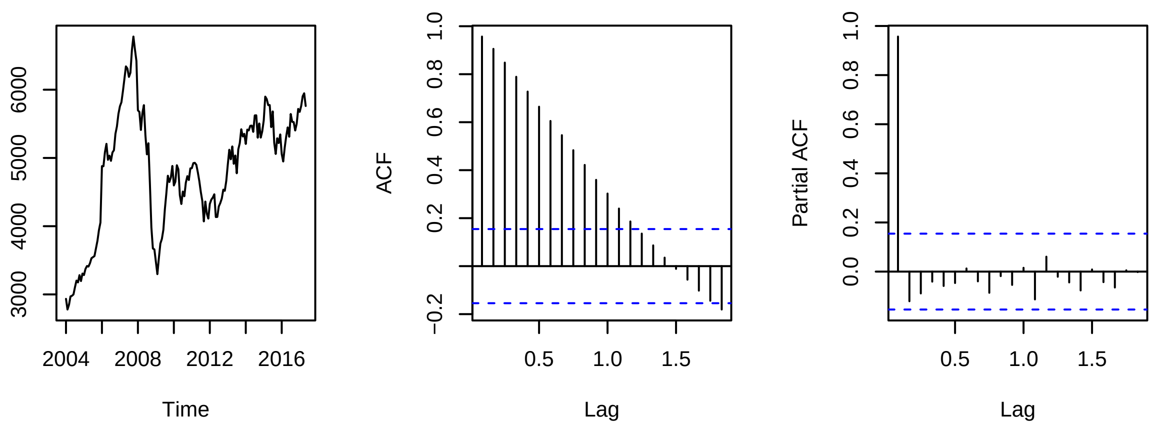
Examination of Effects of Differencing and Transformations

ASX Price

For the ASX series, a log transformation did not appear to make a significant difference, which may be due to the initial low changing variance. Differencing demonstrated a large reduction in nonstationarity such that by the 4 stationarity tests agreed that the data was stationary.

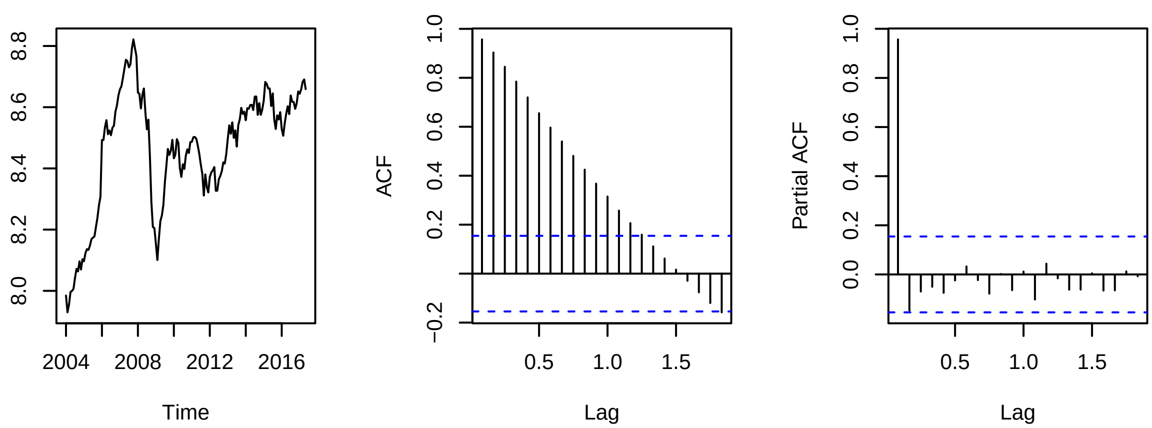
```
display <- function(data, name = NA, figure){
  par(mfrow = c(1, 3))
  plot(data, main = NA, cex.main = .9, ylab = NA)
  acf(data, main = NA, cex.main = .9)
  pacf(data, main = NA, cex.main = .6)
  title(paste0("\n", "Figure ", figure, ". ", name, " Plot, ACF and PACF"), outer = TRUE)
  par(mfrow = c(1, 1))
}

display(asx, "ASX", 9)
```

Figure 9. ASX Plot, ACF and PACF




```
display(log(asx), "Transformed ASX", 5)
```

Figure 5. Transformed ASX Plot, ACF and PACF



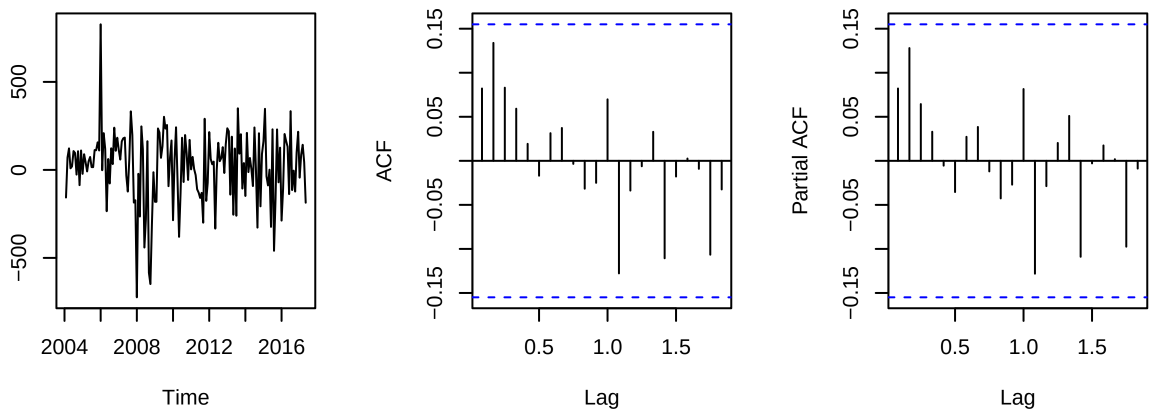
```r
asx.diff <- diff(asx)
checkstationarity(asx.diff)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -4.5543, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -708.38  -93.22   48.40  147.29  817.12
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1 -0.90948    0.07929  -11.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202.9 on 158 degrees of freedom
## Multiple R-squared:  0.4544, Adjusted R-squared:  0.4509
## F-statistic: 131.6 on 1 and 158 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -11.4702
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726.94 -110.96   30.36  130.50  800.79
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.19528   16.15436   1.064    0.289
## y.l1         0.08261    0.07960   1.038    0.301
##
## Residual standard error: 202.8 on 157 degrees of freedom
## Multiple R-squared:  0.006813,   Adjusted R-squared:  0.0004874
## F-statistic: 1.077 on 1 and 157 DF,  p-value: 0.301
##
##
## Value of test-statistic, type: Z-tau  is: -11.6769
##
##          aux. Z statistics
## Z-tau-mu            1.0785
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.472387 -2.879651 -2.576321
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.1054
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```r
display(asx.diff, "Differenced ASX", 6)
```

Figure 6. Differenced ASX Plot, ACF and PACF



Gold Price

The Boxcox transformation appears to make some difference to the gold dataset, with the main difference being at the start of the series. The differencing makes a more significant difference to the dataset, with the differenced series being confirmed by the 4 stationarity tests as stationary.

```
boxcox.auto <- function(data){
  bc <-BoxCox.ar(data)
  lambda <- mean(bc$ci)
  transf <- forecast::BoxCox(data, lambda = lambda)
}

display(gold, "Gold", 7)
```

Figure 7. Gold Plot, ACF and PACF

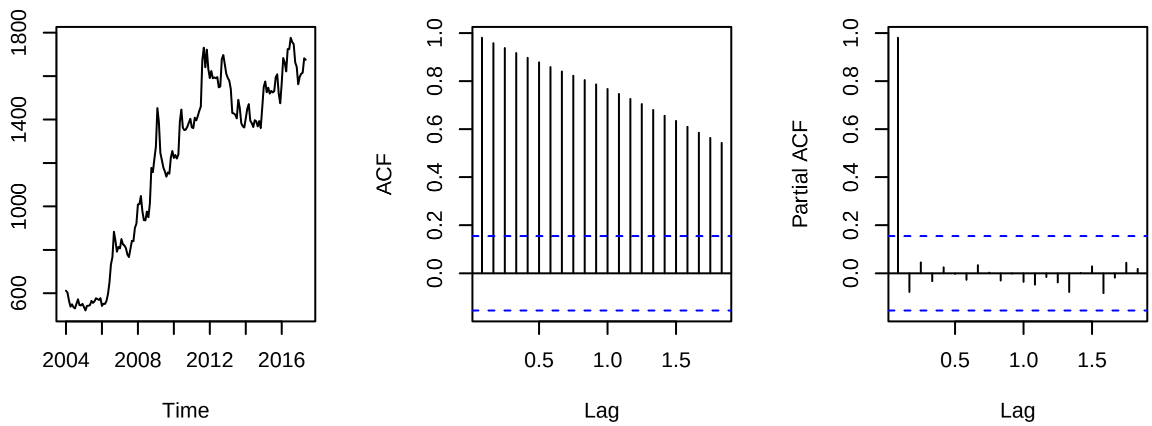
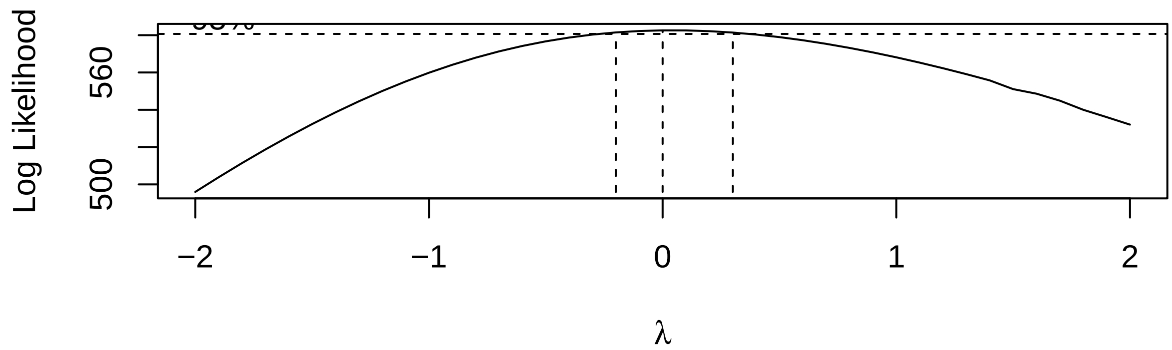


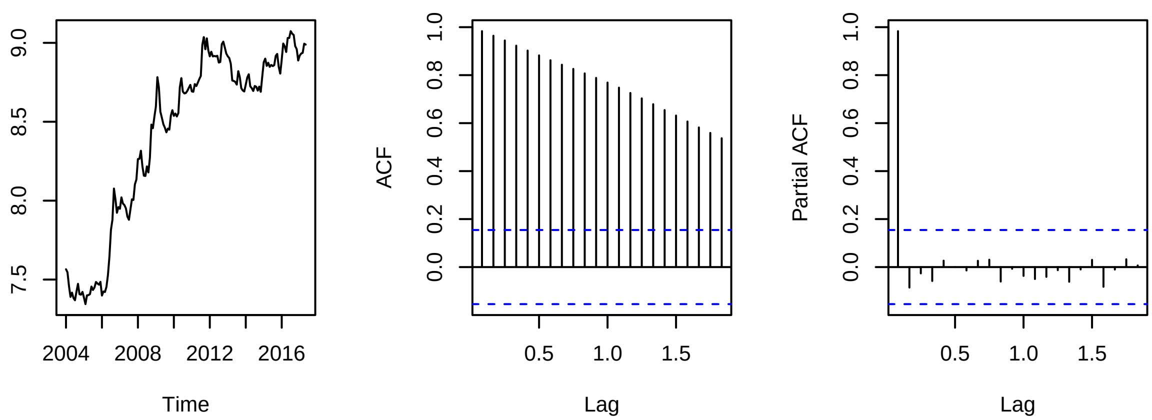
Figure 8. BoxCox Plot

```
gold.tr <- boxcox.auto(gold)
```




```r
display(gold.tr, "Gold", 9)
```

Figure 9. Gold Plot, ACF and PACF




```r
gold.diff <- diff(gold)
checkstationarity(gold.diff)
```

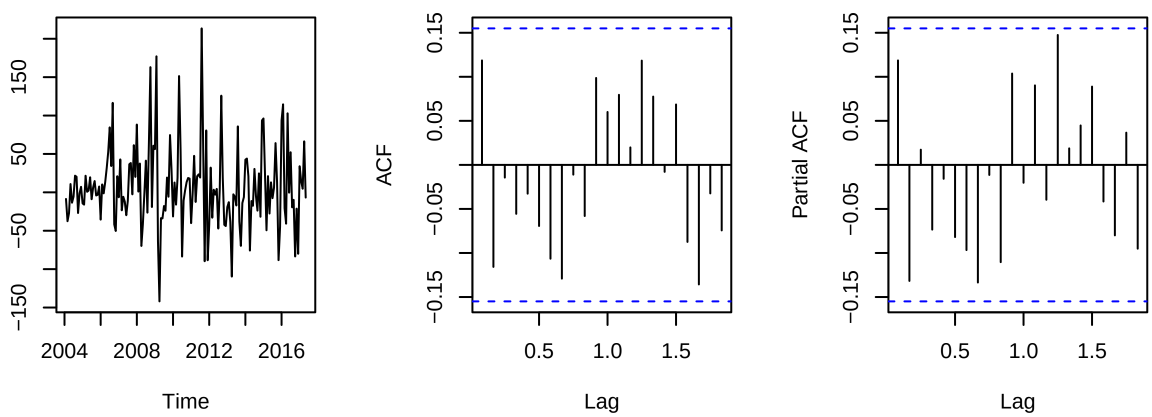
```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -5.8718, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
```

```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.536  -21.619    2.742   28.198  213.468
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -0.9730     0.1051  -9.258   <2e-16 ***
## z.diff.lag   0.1193     0.0800   1.491    0.138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.58 on 155 degrees of freedom
## Multiple R-squared:  0.4432, Adjusted R-squared:  0.436
## F-statistic: 61.69 on 2 and 155 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -9.258
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.373  -28.889   -3.725   21.022  205.268
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.94193    4.18926   1.418    0.158
## y.l1         0.11864    0.07924   1.497    0.136
##
## Residual standard error: 52.39 on 157 degrees of freedom
## Multiple R-squared:  0.01408,    Adjusted R-squared:  0.007797
## F-statistic: 2.242 on 1 and 157 DF,  p-value: 0.1363
##
##
## Value of test-statistic, type: Z-tau  is: -11.0448
##
```

```
##          aux. Z statistics
## Z-tau-mu            1.4085
##
## Critical values for Z statistics:
##                       1pct      5pct     10pct
## critical values -3.472387 -2.879651 -2.576321
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0719
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
display(gold.diff, "Gold", 10)
```

Figure 10. Gold Plot, ACF and PACF

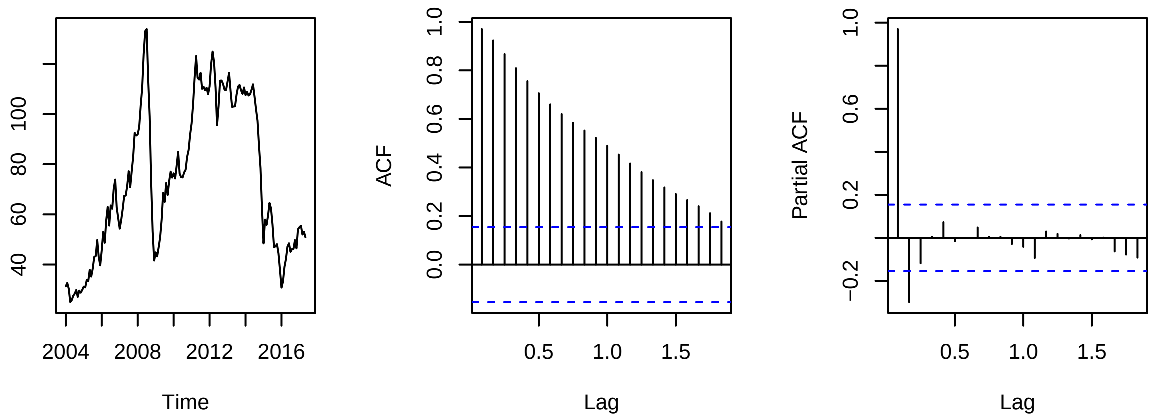


Oil Price

The oil series seems to benefit from a log transformation to reduce changing variance, which is seen in both the undifferenced and differenced series. The differenced series is confirmed to be stationary by the 4 stationarity tests and with the transformation appears to demonstrate more consistent variance.

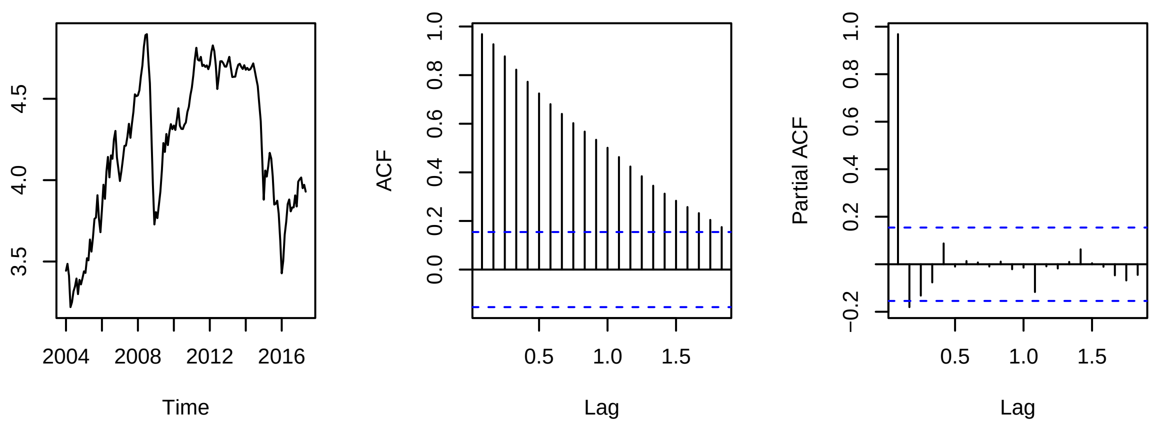
```
display(oil, "Oil", 27)
```

Figure 27. Oil Plot, ACF and PACF




```r
display(log(oil), "Oil", 11)
```

Figure 11. Oil Plot, ACF and PACF




```r
oil.diff <- diff(oil)
checkstationarity(oil)
```

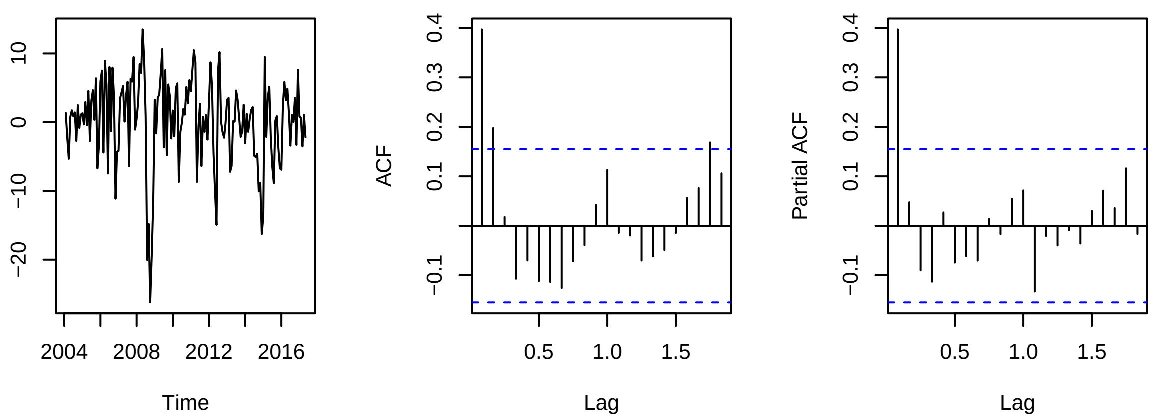
```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -1.8523, Lag order = 5, p-value = 0.6379
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
```

```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.943  -3.549   1.248   3.982  15.176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.003793   0.005835  -0.650    0.517
## z.diff.lag  0.399011   0.073572   5.423  2.2e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.863 on 155 degrees of freedom
## Multiple R-squared:  0.1601, Adjusted R-squared:  0.1493
## F-statistic: 14.78 on 2 and 155 DF,  p-value: 1.339e-06
##
##
## Value of test-statistic is: -0.65
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.6715  -3.5936   0.4549   3.9455  14.3505
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.07981    1.32129   1.574    0.117
## y.l1         0.97347    0.01659  58.686   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.302 on 158 degrees of freedom
## Multiple R-squared:  0.9561, Adjusted R-squared:  0.9559
## F-statistic:  3444 on 1 and 158 DF,  p-value: < 2.2e-16
##
##
```

```
## Value of test-statistic, type: Z-tau  is: -2.0125
##
##          aux. Z statistics
## Z-tau-mu            1.9323
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.472136 -2.879539 -2.576262
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.8196
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

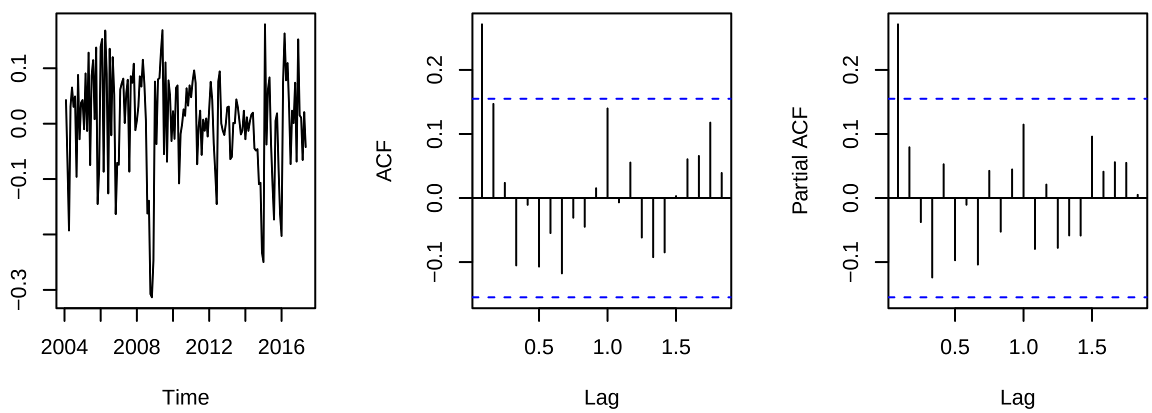
```
display(oil.diff, "Oil", 12)
```

Figure 12. Oil Plot, ACF and PACF




```
oil.diff.tr <- diff(log(oil))
display(oil.diff.tr, "Oil", 13)
```

Figure 13. Oil Plot, ACF and PACF



Copper

The Boxcox transformation appears to make the variance in the copper series more consistent, although the effect is not large. Additionally, the differencing is confirmed to make the series stationary by the 4 stationarity test. The differenced and transformed series seems to be an improvement on the original differenced series, although some large changed from one point to another remain.

```
display(copper, "Copper", 14)
```

Figure 14. Copper Plot, ACF and PACF

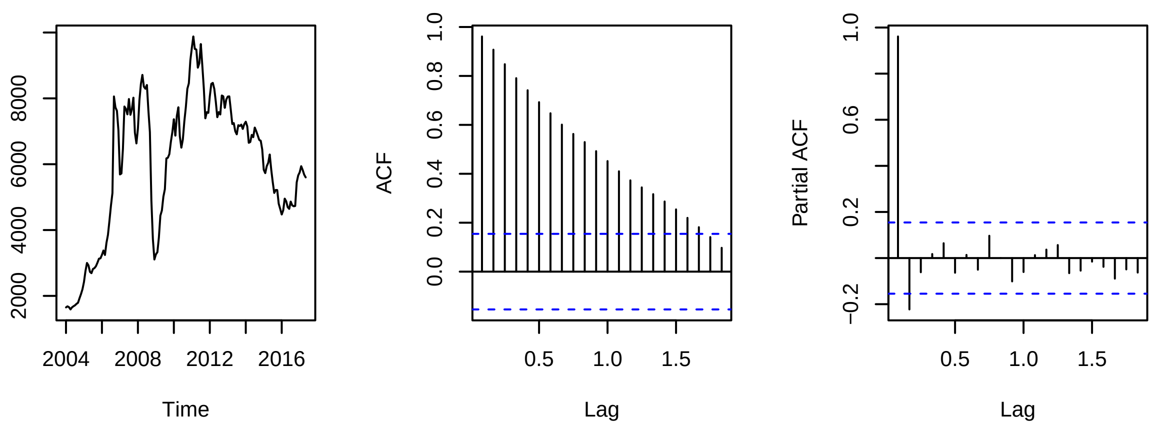
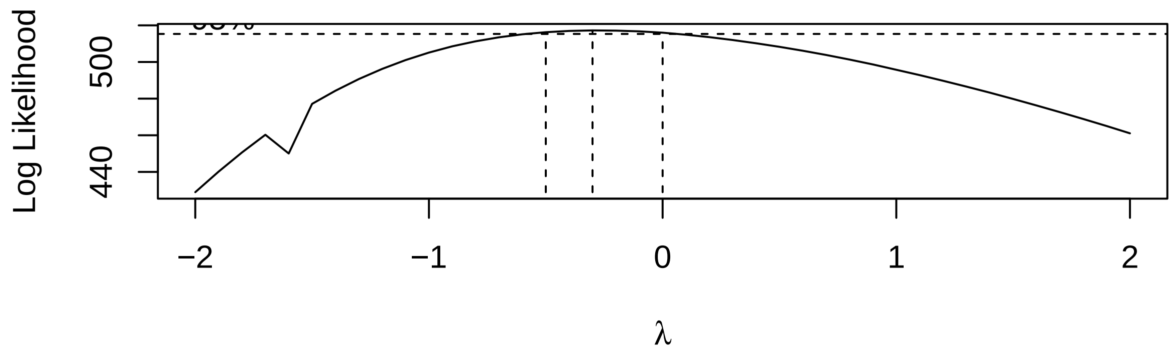


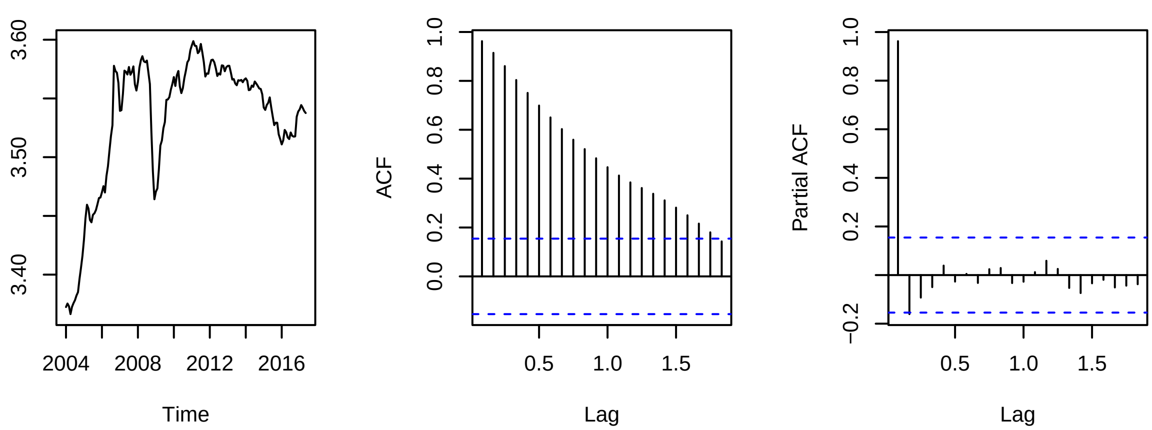
Figure 15. BoxCox Plot

```
copper.tr <- boxcox.auto(copper)
```




```
display(copper.tr, "Copper", 16)
```

Figure 16. Copper Plot, ACF and PACF




```
copper.diff <- diff(copper)
checkstationarity(copper.diff)
```

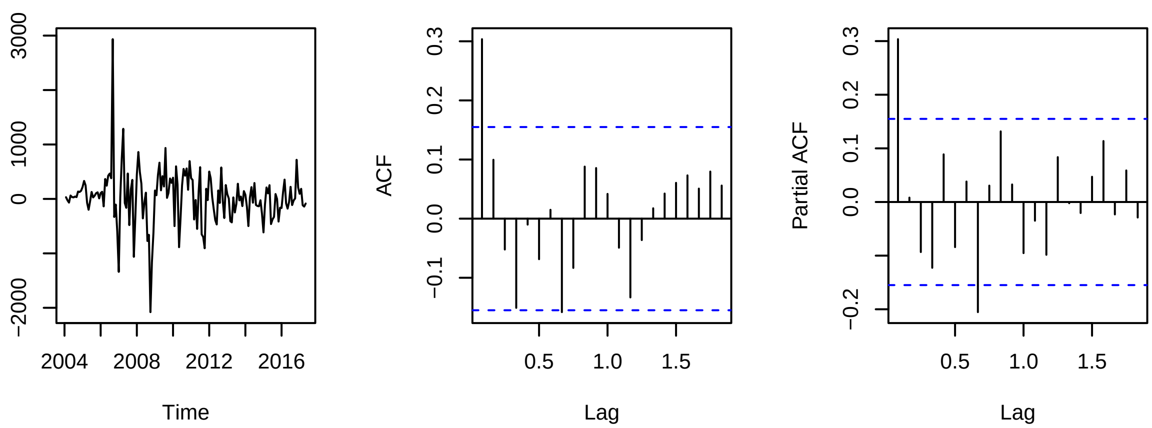
```
##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -5.478, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
##
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
```

```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1873.02  -176.79    25.31   233.05  2815.45
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.687611   0.094381  -7.285 1.49e-11 ***
## z.diff.lag -0.009816   0.080081  -0.123    0.903
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 464.8 on 156 degrees of freedom
## Multiple R-squared:  0.3472, Adjusted R-squared:  0.3389
## F-statistic: 41.49 on 2 and 156 DF,  p-value: 3.556e-15
##
##
## Value of test-statistic is: -7.2855
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
##
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1896.8  -191.6     4.5   218.4  2802.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.93190   36.77216   0.460    0.646
## y.l1         0.30370    0.07605   3.993 9.98e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 463 on 157 degrees of freedom
## Multiple R-squared:  0.09221,    Adjusted R-squared:  0.08642
## F-statistic: 15.95 on 1 and 157 DF,  p-value: 9.984e-05
##
##
```

```
## Value of test-statistic, type: Z-tau  is: -9.0691
##
##          aux. Z statistics
## Z-tau-mu           0.4559
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.472387 -2.879651 -2.576321
##
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.1707
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

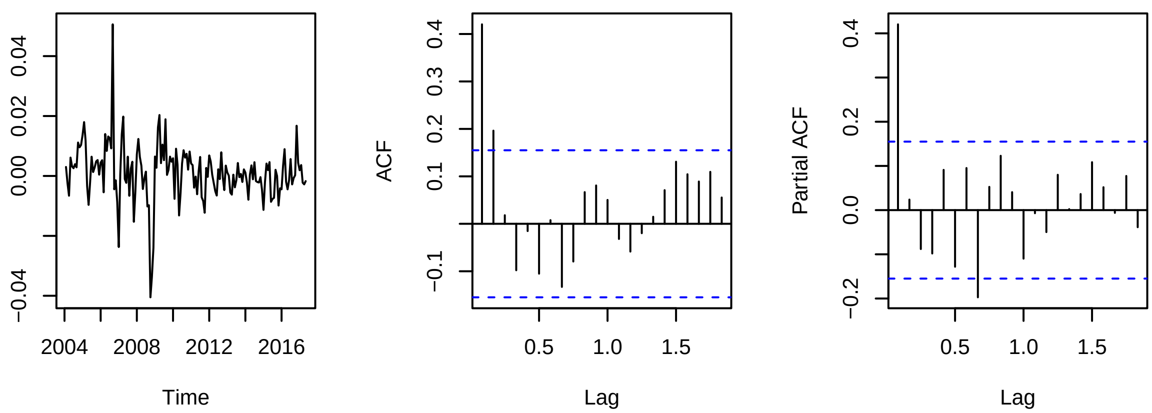
```
display(copper.diff, "Copper", 17)
```

Figure 17. Copper Plot, ACF and PACF



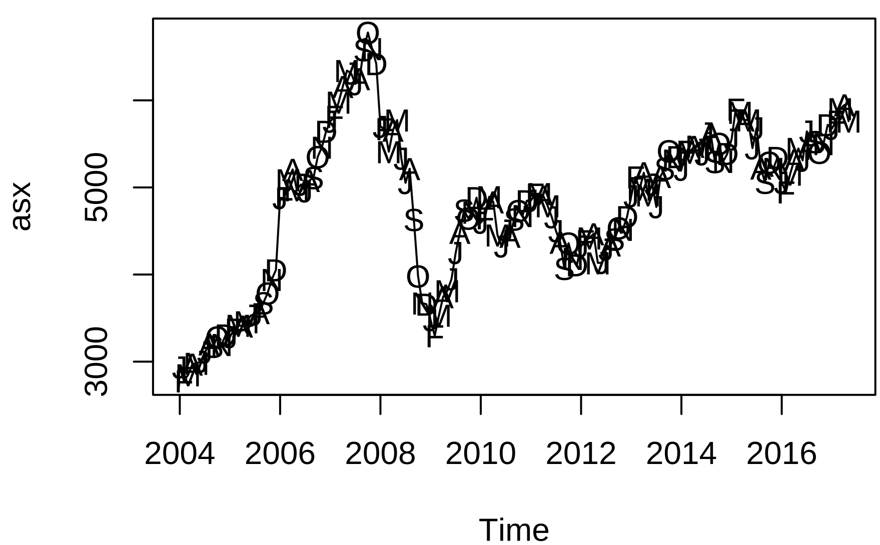

```
copper.diff.tr <- diff(copper.tr)
display(copper.diff.tr, "Copper", 18)
```

Figure 18. Copper Plot, ACF and PACF




```
plot(asx, main = "Figure 19. ASX Plot with Seasonal Markers")
points(y=asx, x=time(asx), pch=as.vector(season(asx)), main = "Figure 17. ASX Plot with Seasonal Marker
```

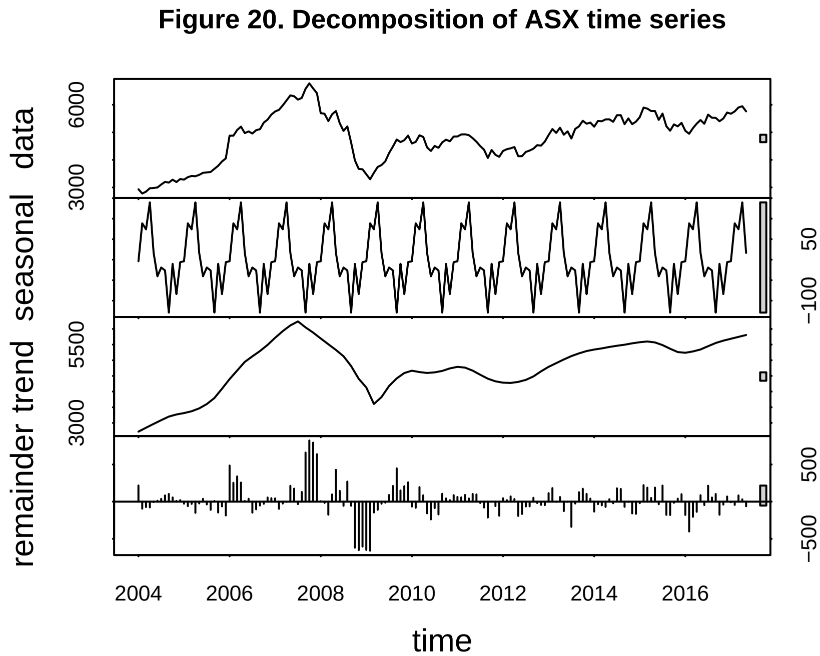
Figure 19. ASX Plot with Seasonal Markers



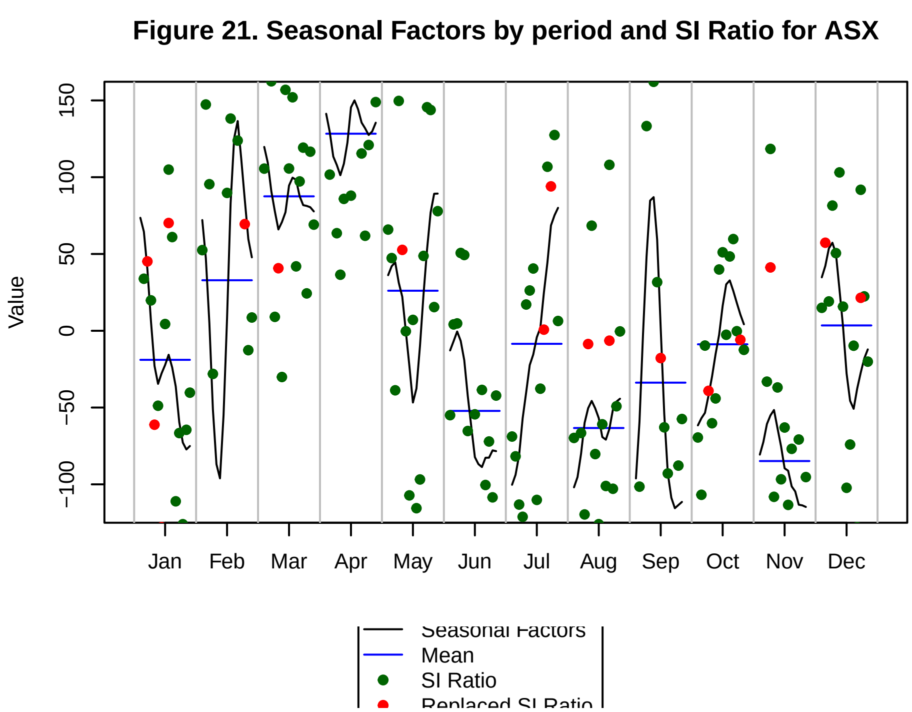
Decomposition and Forecasting

Decomposition demonstrates the trend in the ASX series. A pattern in the seasonal component of the series is less clear, the plot of seasonal factors demonstrates a large difference in seasonal component between months. For this reason the series was forecasted with seasonal adjustment. The forecast predicts that the ASX price will continue to rise as it has in the past.

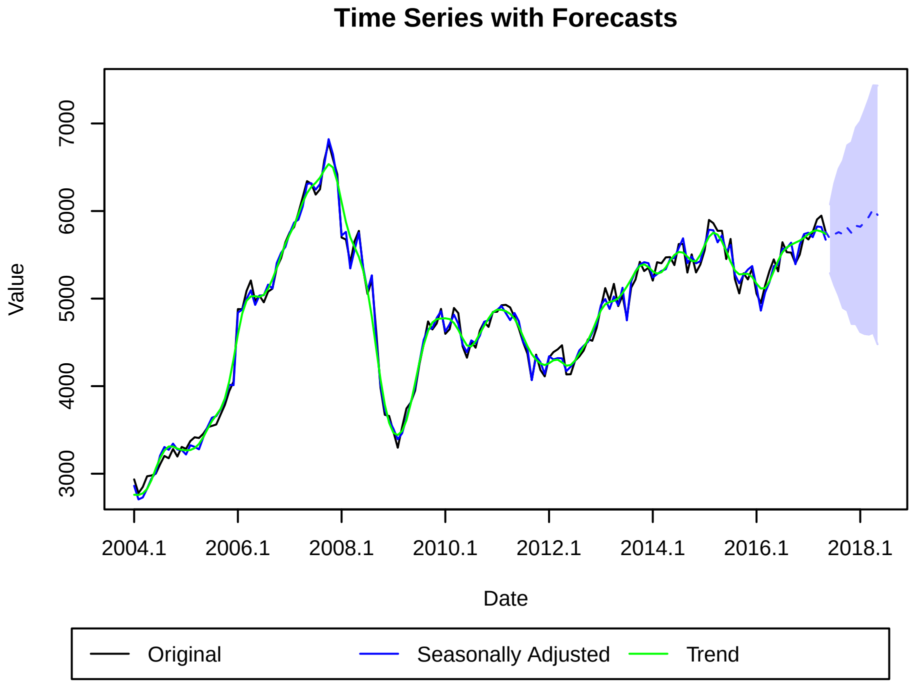
```
asx.decom <- stl(asx, t.window = 15, s.window = "periodic", robust = TRUE)
plot(asx.decom, main = "Figure 20. Decomposition of ASX time series")
```




```
asx.decomp12 <- x12(asx)
plotSeasFac(asx.decomp12, main = "Figure 21. Seasonal Factors by period and SI Ratio for ASX")
```

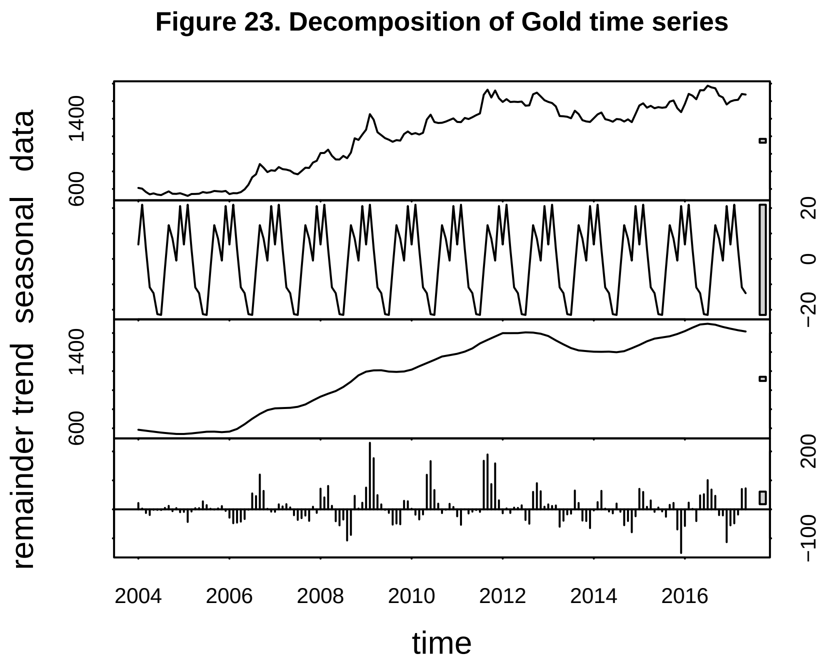


```r
x12::plot(asx.decomp12, sa = TRUE, trend = TRUE, forecast = TRUE, main = "Figure 22. ASX Series with Fo
```

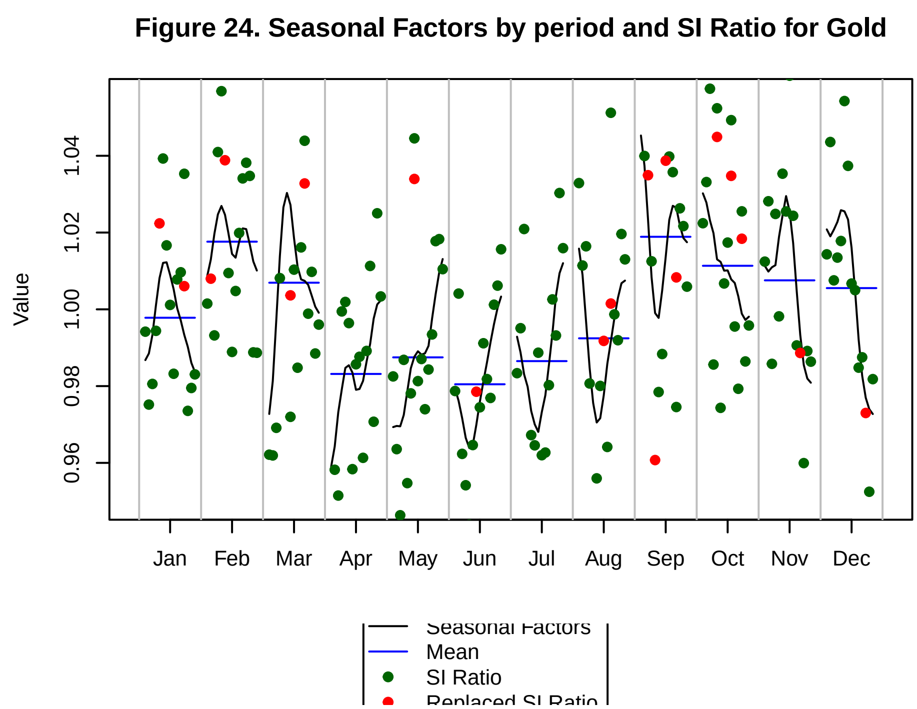


The trend of the gold series was demonstrated by the decomposition. The seasonal factors may suggest some seasonality in the series, although the seasonal factors are small. The forecast without seasonal adjustment predicts that the series will remain at its current level where it finished the series.

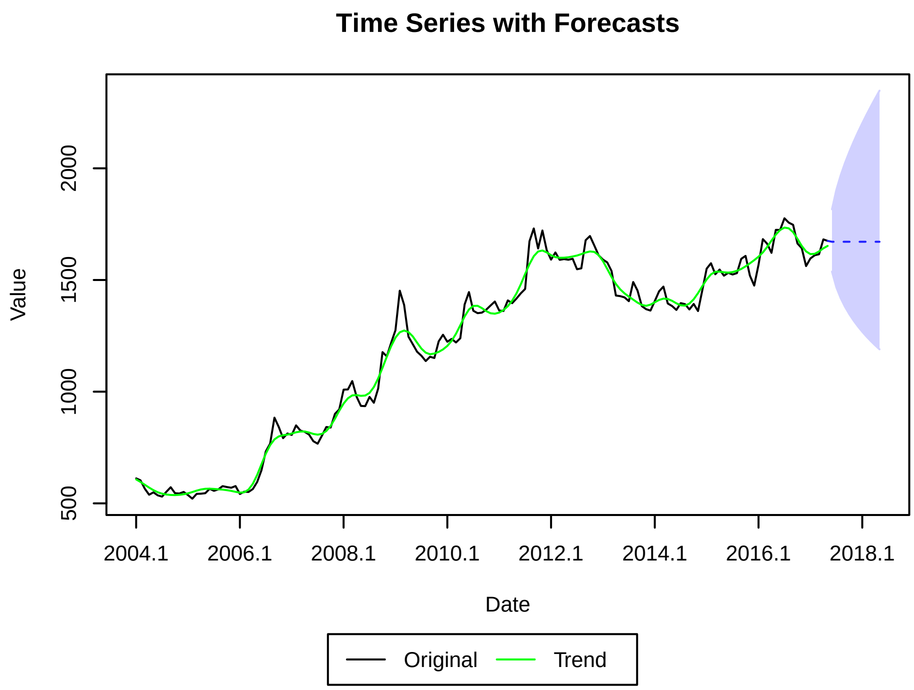
```r
gold.decom <- stl(gold, t.window = 15, s.window = "periodic", robust = TRUE)
plot(gold.decom, main = "Figure 23. Decomposition of Gold time series")
```




```r
gold.decomp12 <- x12(gold)
plotSeasFac(gold.decomp12, main = "Figure 24. Seasonal Factors by period and SI Ratio for Gold")
```

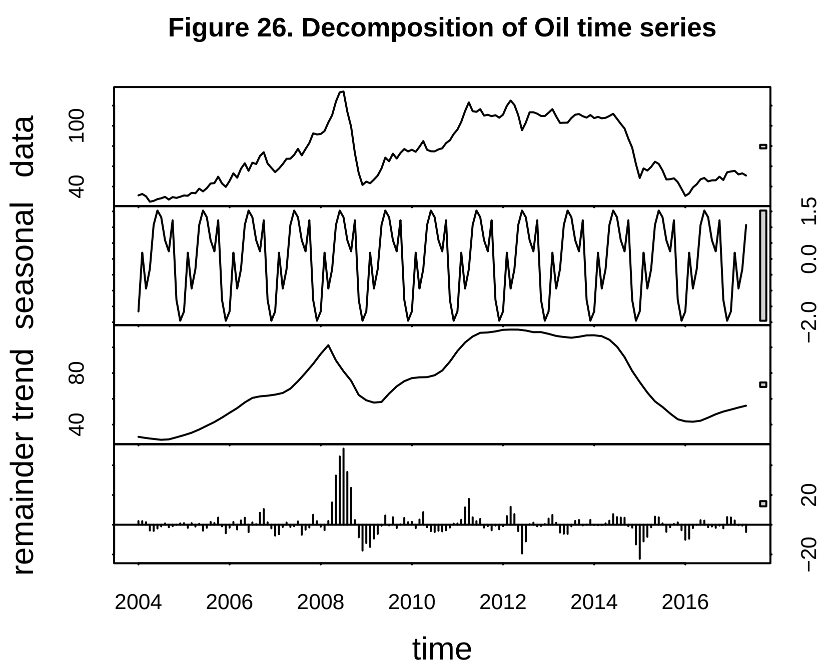



```r
x12::plot(gold.decomp12, sa = FALSE, trend = TRUE, forecast = TRUE, main = "Figure 25. Gold Series with
```

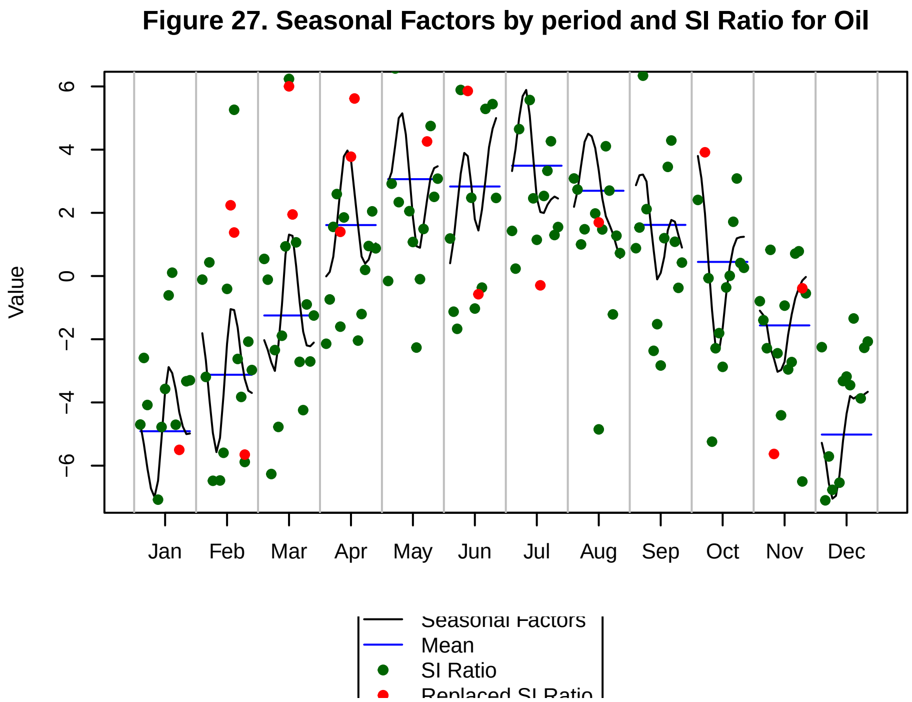


Decomposition of the oil time series demonstrates the trend overtime. Additionally, the plot of seasonal factors shows the pattern of seasonality throughout the data. Therefore, the forecast used the trend and seasonal adjustment. It predicts that the series will remain relatively stable in the future.

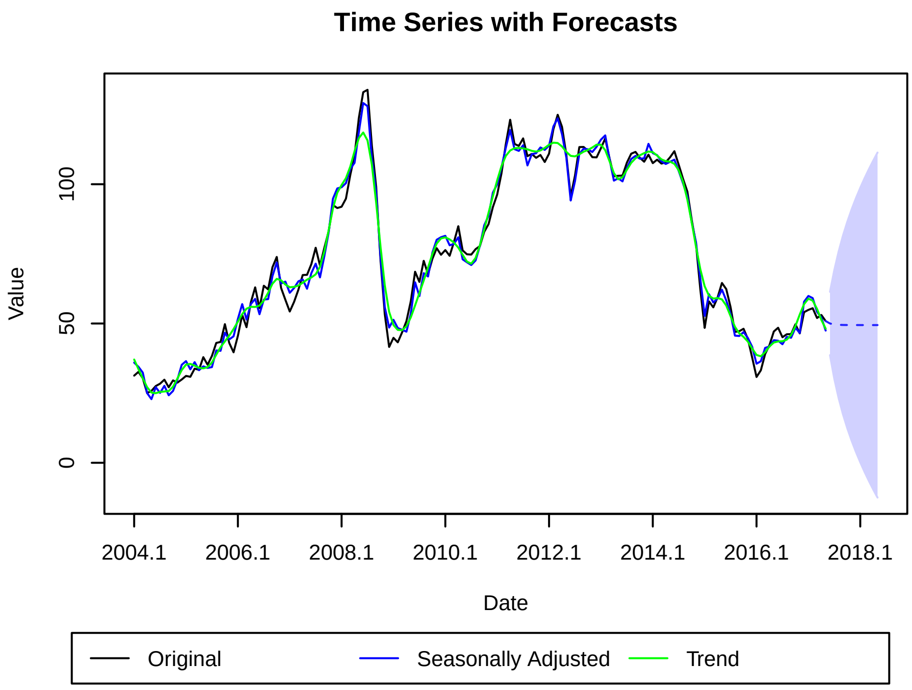
```
oil.decom <- stl(oil, t.window = 15, s.window = "periodic", robust = TRUE)
plot(oil.decom, main = "Figure 26. Decomposition of Oil time series")
```



```
oil.decomp12 <- x12(oil)
plotSeasFac(oil.decomp12, main = "Figure 27. Seasonal Factors by period and SI Ratio for Oil")
```

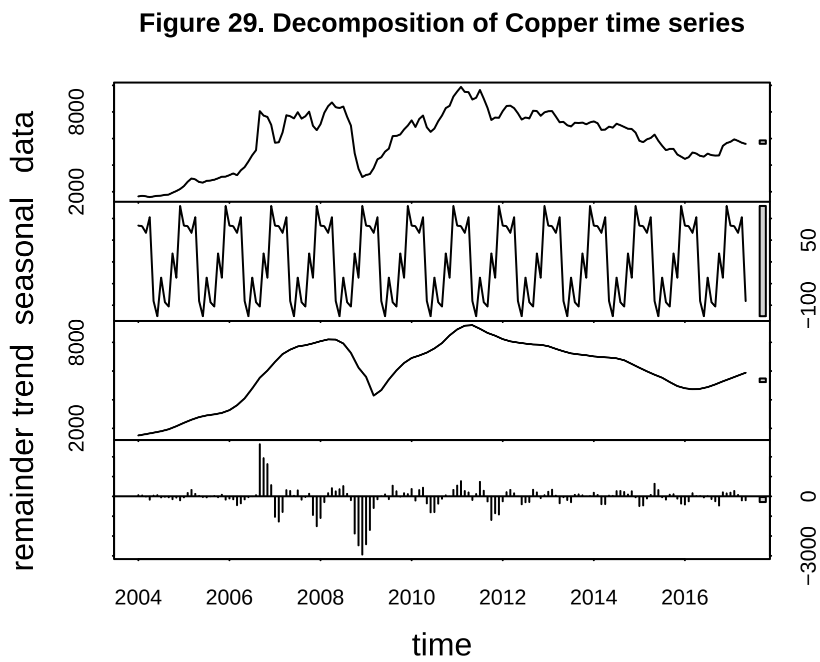



```
x12::plot(oil.decomp12, sa = TRUE, trend = TRUE, forecast = TRUE, main = "Figure 28. Gold Series with Fo
```



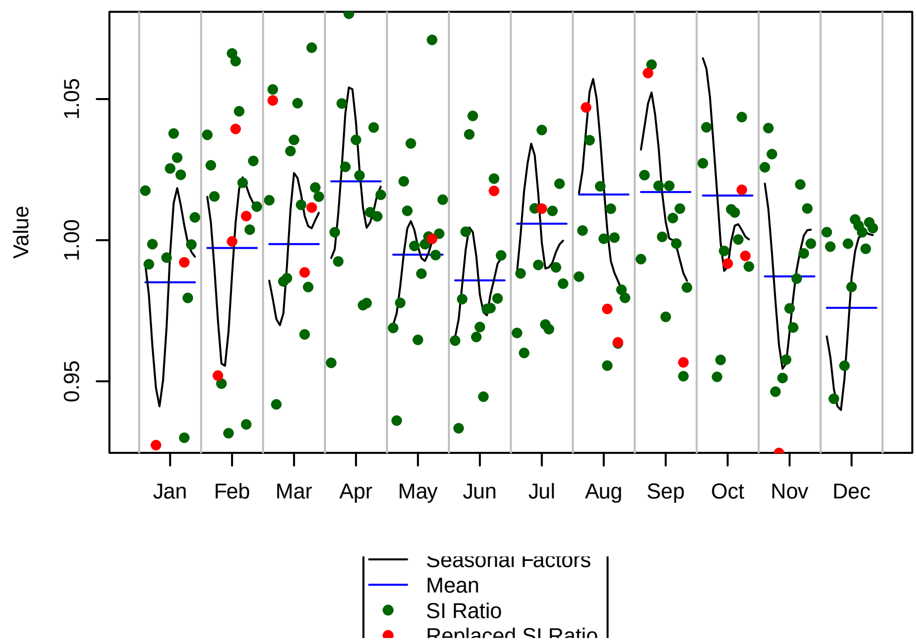
The trend is demonstrated by the decomposition of the time series. The plot of seasonal factors suggests little seasonal component, therefore, it will not be used in the forecast. The forecast predicts that the series will remain consistent in the future.

```
copper.decom <- stl(copper, t.window = 15, s.window = "periodic", robust = TRUE)
plot(copper.decom, main = "Figure 29. Decomposition of Copper time series")
```



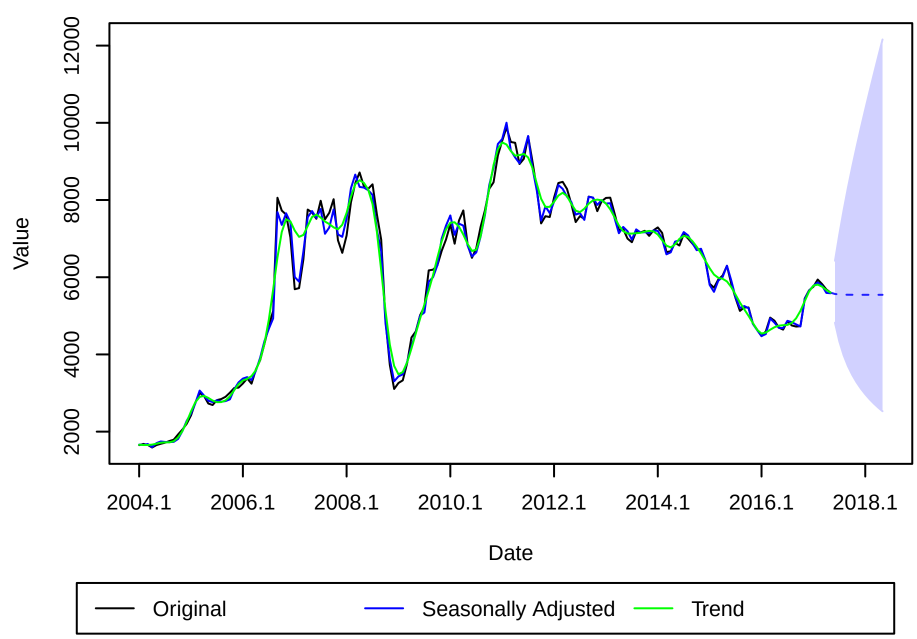

```
copper.decomp12 <- x12(copper)
plotSeasFac(copper.decomp12, main = "Figure 30. Seasonal Factors by period and SI Ratio for Copper")
```

Figure 30. Seasonal Factors by period and SI Ratio for Copper




```
x12::plot(copper.decomp12, sa = TRUE, trend = TRUE, forecast = TRUE, main = "Figure 31. Gold Series wit
```

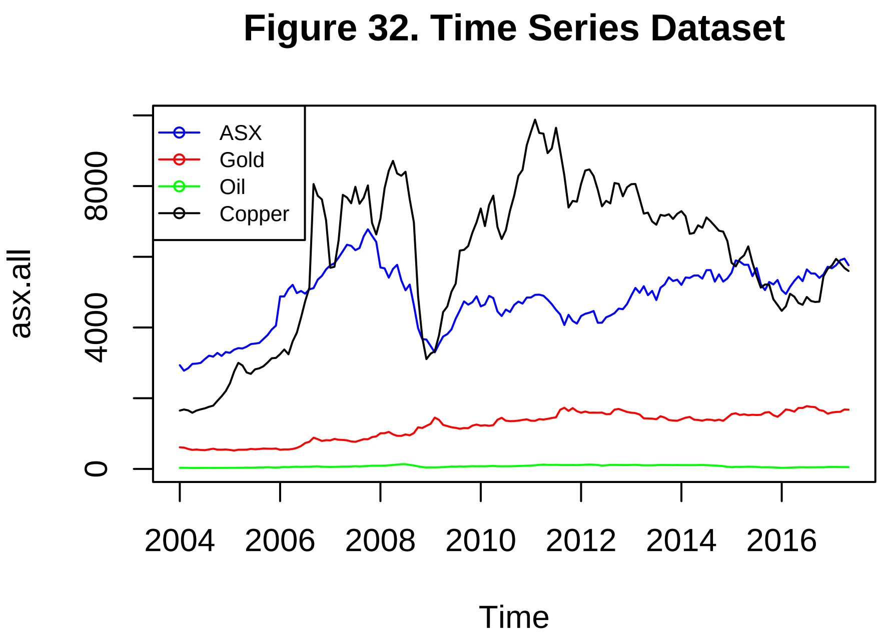
Time Series with Forecasts



Model Fitting

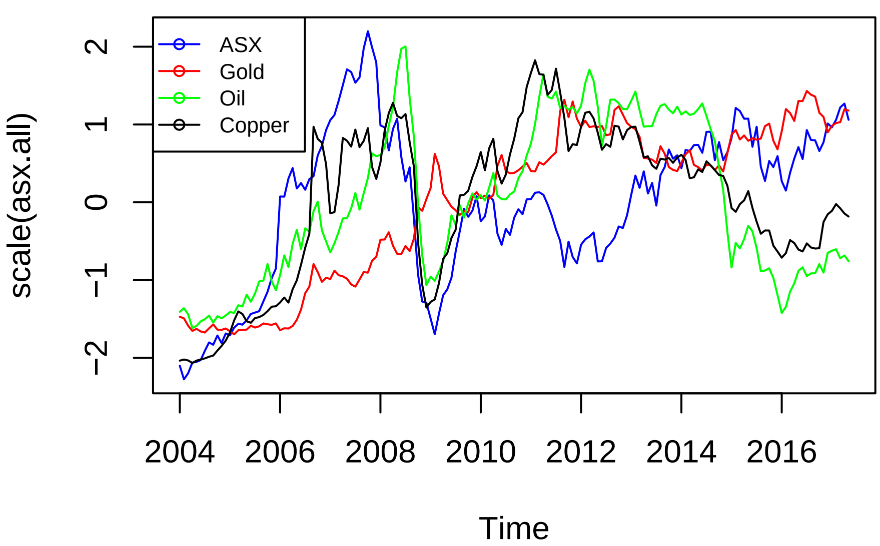
To predict and forecast the price of the ASX in the future models will be fit and evaluated. The models used will be the distributed lag models, the polynomial distributed lag model, the Koyck model and the autoregressive distributed lag model. To evaluate these model using AIC/BIC, the adjusted R squared and model and coefficient significance values will be used. Furthermore, residuals will be examined to evaluate the effectiveness of the model fit and variance inflation factors (VIFs) will be used to examine whether the model has satisfied the multicollinearity assumption.

```
plot(asx.all, plot.type = "s", type = 'l', col = c("blue", "red", "green", "black"), main = "Figure 32.
legend("topleft", lty = 1, pch = 1, text.width = 1.5, col = c("blue", "red", "green", "black"), c("ASX"
```




```
plot(scale(asx.all), plot.type = "s", type = 'l', col = c("blue", "red", "green", "black"), main = "Fig
legend("topleft", lty = 1, pch = 1, text.width = 1.5, col = c("blue", "red", "green", "black"), c("ASX"
```

Figure 33. Scaled Time Series Dataset



Distributed Lag Models (DLMs)

To predict ASX price, gold, oil and copper were used used as independent variables to fit distributed lag models (DLMs). Six models were attempted, 1 with each variable and 3 with combinations of them. However, none of these models demonstrated high success, with each model leaving high serial correlation in the residuals as demonstrated by the Breusch-Godfrey test and ACF visual inspection. This implies that the models struggled to account for the variance in the data. The very low to moderate adjusted R squared values found by the model further supports this. Additionally, all models violated the multicollinearity assumption. In terms of statistical significance, all models were significant at alpha 0.05. However, 4 of the models did not have a significant coefficient, with model 5 and 6 having the first lag of copper as significant (`p < 0.05`). The models appeared to favour a higher number of lags with the AIC/BIC values lowering as these increased. However, since the first lags were not significant, adding more lags resulted in adding nonsignificant coefficients. Additionally, adjusted R squared decreased as lags were increased. Therefore, a lag of 1 was considered to be optimal for all models. Finally, AIC/BIC were used to diferentiate between the final 6 models and demonstrated that model 5 was the best performing model.

Model 5 used oil and copper price to explain ASX price using 1 lag. The model was statistically significant at `p < 0.05`, with the first lag of copper as a significant parameter. Additionally, it demonstrated an adjusted R squared of 0.404. Although the residuals from the model demonstrated serial correlation and nonnormality of distribution. Finally, as the other models, the model violated the multicollinearity assumption. Therefore, the model was the best of the DLM models but was not concluded to be a suitable model. To attempt to produce a model which does not violate the multicollinearity assumption, a polynomial DLM model will be implemented.

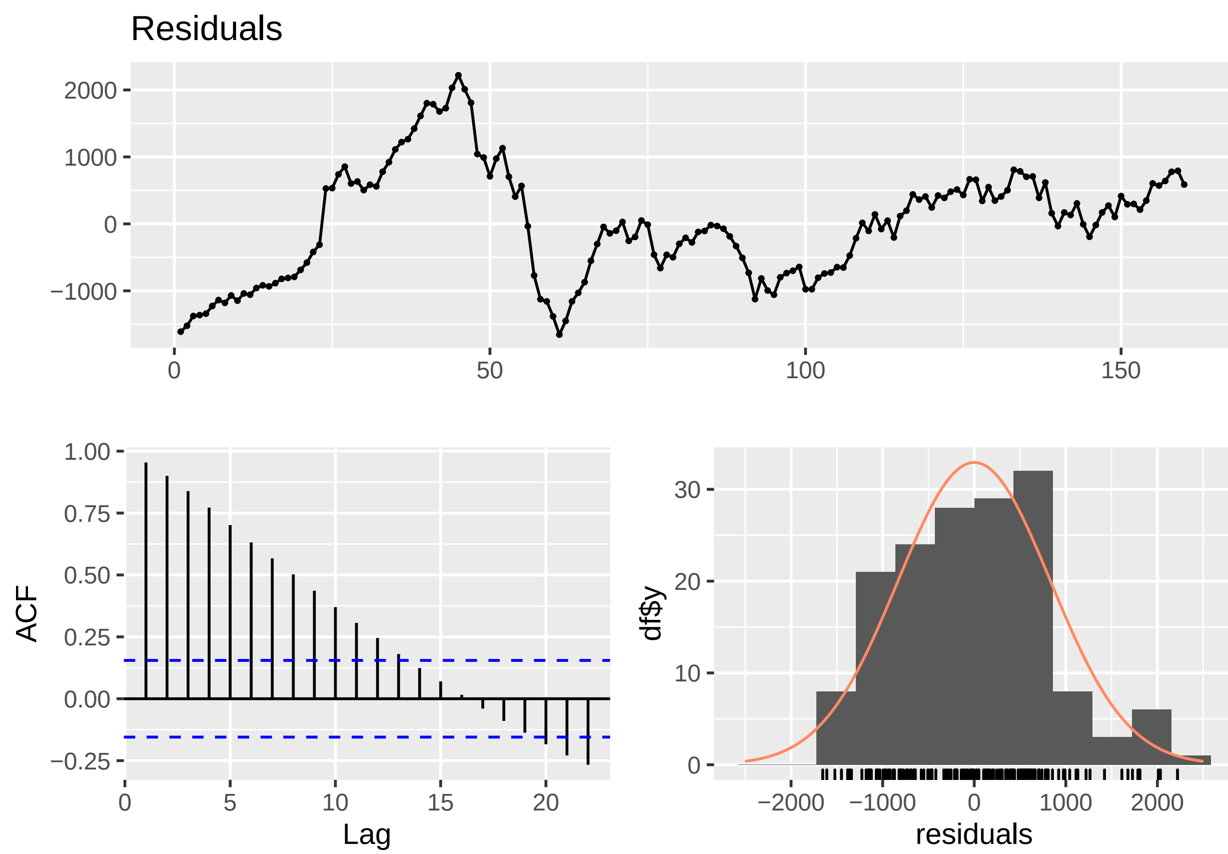
DLM outputs

```
modeldlm1 <- dlm(formula = asx.price ~ gold.price, data = data.frame(asx.all), q = 1)
summary(modeldlm1)
```

```
##
## Call:
## lm(formula = as.formula(model.formula), data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1654.13  -706.62    -9.17   561.27  2219.74
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3943.0338   211.5468  18.639   <2e-16 ***
## gold.price.t    0.4117     1.2742   0.323    0.747
## gold.price.1    0.3216     1.2711   0.253    0.801
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 839.7 on 157 degrees of freedom
## Multiple R-squared:  0.1097, Adjusted R-squared:  0.09832
## F-statistic: 9.669 on 2 and 157 DF,  p-value: 0.0001097
##
## AIC and BIC values for the model:
##        AIC     BIC
## 1 2613.609 2625.91
```

Figure 34. Residuals of DLM 1

```
checkresiduals(modeldlm1$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 148.3, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modeldlm1$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modeldlm1$model$fitted.values
## W = 0.88545, p-value = 8.871e-10
```

```
finiteDLMauto(x = as.vector(gold), y = as.vector(asx), q.min = 1, q.max = 10,
              model.type = "dlm", error.type = "AIC", trace = TRUE)
```

```
##    q - k    MASE      AIC      BIC   GMRAE   MBRAE R.Adj.Sq Ljung-Box
## 10    10 3.91844 2460.345 2499.570 4.32190 0.69116 -0.02199         0
## 9      9 3.97365 2476.983 2513.270 4.43213 0.75893 -0.00887         0
## 8      8 4.02150 2493.885 2527.220 4.45929 0.61156  0.00491         0
## 7      7 4.07437 2510.535 2540.905 4.47998 0.92888  0.01807         0
## 6      6 4.11652 2527.575 2554.966 4.40970 1.22132  0.03067         0
## 5      5 4.16217 2544.887 2569.286 4.45961 0.93030  0.04375         0
```

```
## 4      4 4.22171 2562.296 2583.690 4.65274 1.30422  0.05548         0
## 3      3 4.27118 2579.215 2597.590 4.64464 1.61370  0.06908         0
## 2      2 4.31239 2596.292 2611.637 4.68194 2.08243  0.08409         0
## 1      1 4.35631 2613.609 2625.910 4.56013 1.06887  0.09832         0
```

```
vif(modeldlm1$model)
```

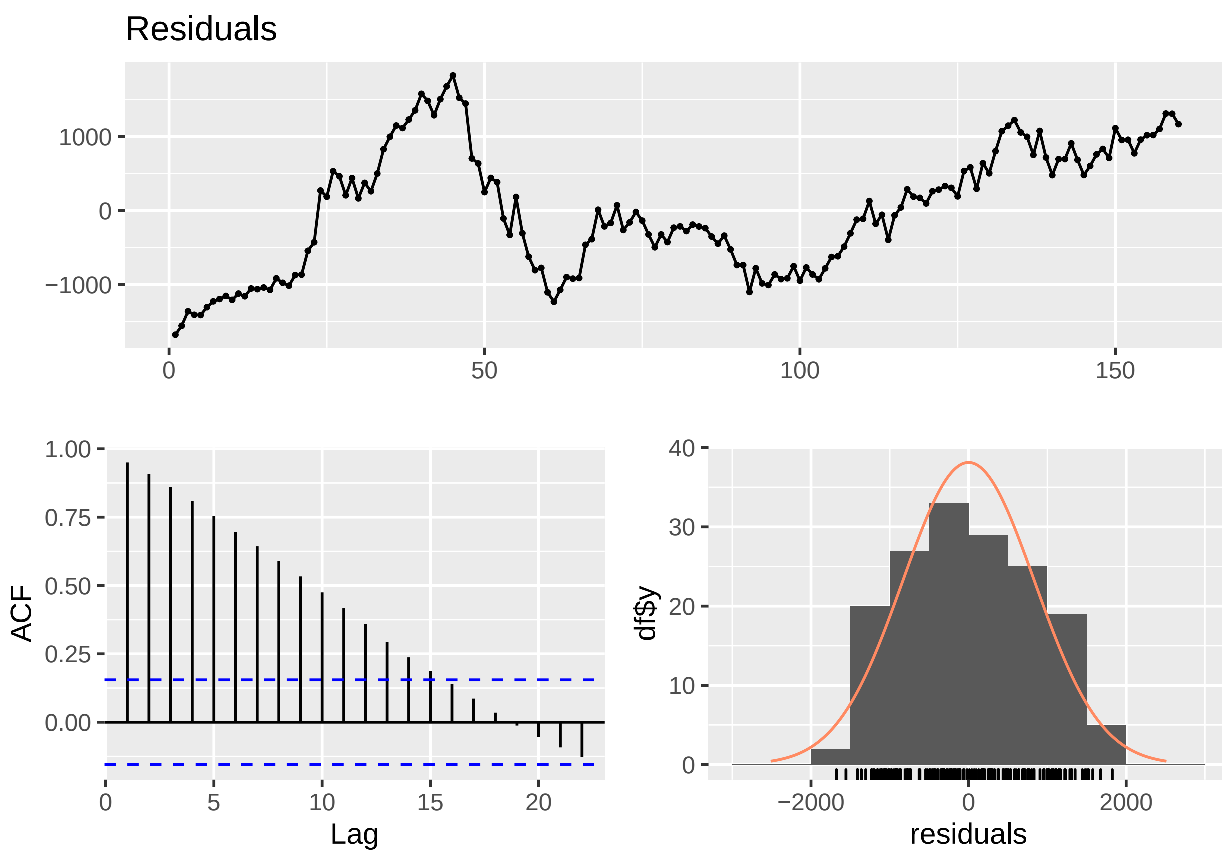
```
## gold.price.t gold.price.1
##     58.50301     58.50301
```

```
modeldlm2 <- dlm(formula = asx.price ~ oil.price, data = data.frame(asx.all), q = 1)
summary(modeldlm2)
```

```
##
## Call:
## lm(formula = as.formula(model.formula), data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1677.08  -770.94   -61.63   694.56  1823.95
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4144.338    178.033  23.278   <2e-16 ***
## oil.price.t   17.140     10.636   1.611    0.109
## oil.price.1   -7.941     10.589  -0.750    0.454
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 842.6 on 157 degrees of freedom
## Multiple R-squared:  0.1036, Adjusted R-squared:  0.09217
## F-statistic: 9.071 on 2 and 157 DF,  p-value: 0.000187
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2614.698 2626.998
```

Figure 35. Residuals of DLM 2

```
checkresiduals(modeldlm2$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 147.28, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modeldlm2$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modeldlm2$model$fitted.values
## W = 0.95055, p-value = 2.008e-05
```

```
finiteDLMauto(formula = asx.price ~ oil.price, data = data.frame(asx.all), q.min = 1, q.max = 10,
              model.type = "dlm", error.type = "AIC", trace = TRUE)
```

```
##     q - k    MASE      AIC      BIC   GMRAE   MBRAE R.Adj.Sq Ljung-Box
## 10    10 4.00878 2459.842 2499.066 4.39830 0.94829 -0.01859         0
## 9      9 4.07657 2476.871 2513.158 4.59976 0.86566 -0.00812         0
## 8      8 4.13953 2493.914 2527.249 4.76106 0.78221  0.00472         0
## 7      7 4.20880 2510.754 2541.124 4.79551 0.02285  0.01667         0
## 6      6 4.26700 2527.701 2555.091 4.84721 0.56283  0.02988         0
## 5      5 4.32339 2544.936 2569.335 4.87419 1.47404  0.04345         0
```

```
## 4      4 4.39825 2561.888 2583.281 5.11093 1.10612  0.05794         0
## 3      3 4.47189 2579.101 2597.477 5.24099 1.43485  0.06975         0
## 2      2 4.53543 2596.715 2612.059 5.37682 0.74046  0.08165         0
## 1      1 4.60656 2614.698 2626.998 5.46829 0.63464  0.09217         0
```

```
vif(modeldlm2$model)
```

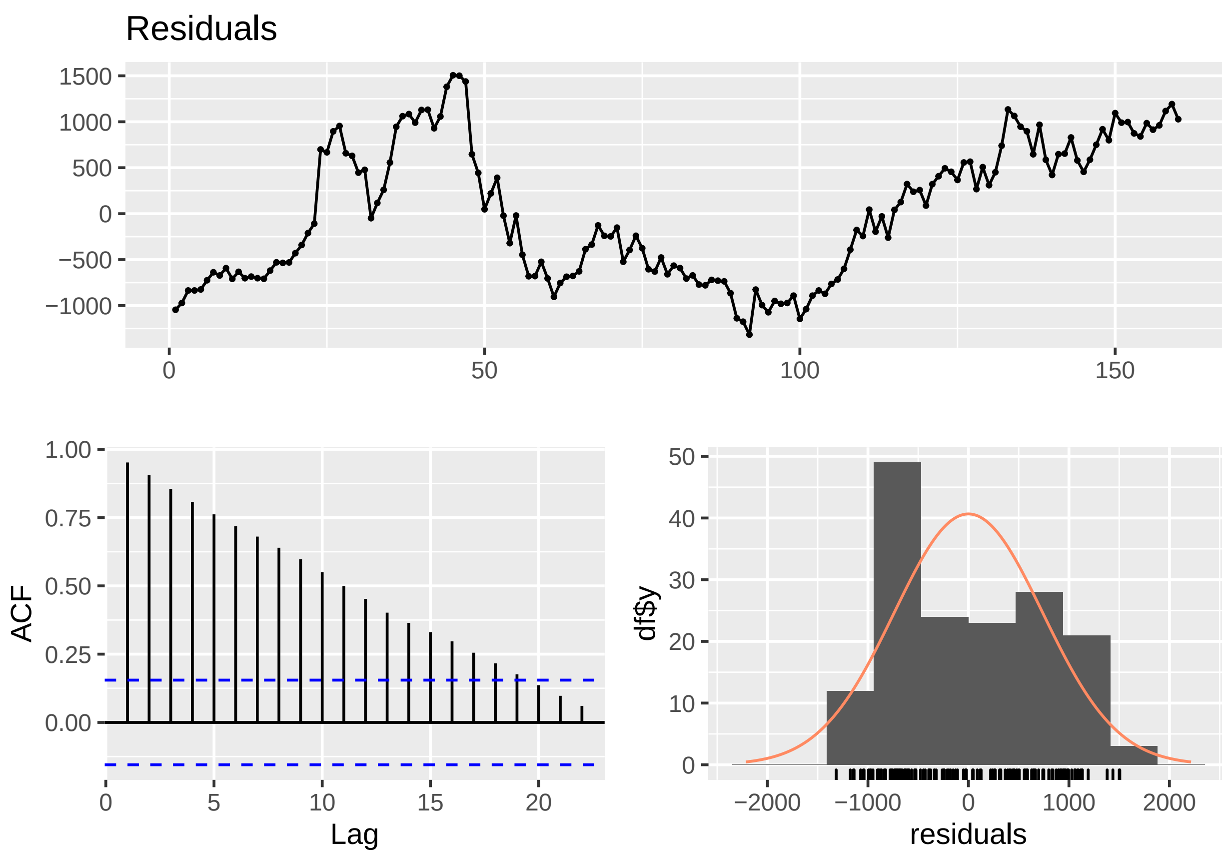
```
## oil.price.t oil.price.1
##    22.79751    22.79751
```

```
modeldlm3 <- dlm(formula = asx.price ~ copper.price, data = data.frame(asx.all), q = 1)
summary(modeldlm3)
```

```
##
## Call:
## lm(formula = as.formula(model.formula), data = design)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1315.8  -679.5  -117.9   646.4  1506.1
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.437e+03  1.781e+02  19.295   <2e-16 ***
## copper.price.t 1.847e-01  1.238e-01   1.492    0.138
## copper.price.1 4.624e-02  1.222e-01   0.378    0.706
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 743.1 on 157 degrees of freedom
## Multiple R-squared:  0.3028, Adjusted R-squared:  0.2939
## F-statistic: 34.09 on 2 and 157 DF,  p-value: 5.06e-13
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2574.488 2586.789
```

Figure 36. Residuals of DLM 3

```
checkresiduals(modeldlm3$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 147.66, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modeldlm3$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modeldlm3$model$fitted.values
## W = 0.94527, p-value = 7.221e-06
```

```
finiteDLMauto(formula = asx.price ~ copper.price, data = data.frame(asx.all), q.min = 1, q.max = 10,
              model.type = "dlm", error.type = "AIC", trace = TRUE)
```

```
##     q - k    MASE      AIC      BIC   GMRAE    MBRAE R.Adj.Sq Ljung-Box
## 10     10 3.97380 2436.164 2475.389 5.07172  0.93641  0.12924         0
## 9       9 4.00524 2451.016 2487.302 5.07902  0.99736  0.14957         0
## 8       8 4.03682 2466.511 2499.846 5.19900  0.04304  0.16793         0
## 7       7 4.07595 2481.988 2512.357 5.19017  0.78489  0.18422         0
## 6       6 4.10610 2497.775 2525.166 5.29119  0.78994  0.20021         0
## 5       5 4.12565 2513.265 2537.664 5.36259  0.90478  0.21921         0
```

```
## 4      4 4.16156 2528.895 2550.289 5.38475  0.88789  0.23649         0
## 3      3 4.19532 2544.155 2562.531 5.23016  0.67751  0.25434         0
## 2      2 4.21688 2559.356 2574.700 5.34802  0.67093  0.27395         0
## 1      1 4.24262 2574.488 2586.789 5.38993 -0.49453  0.29391         0
```

```
vif(modeldlm3$model)
```

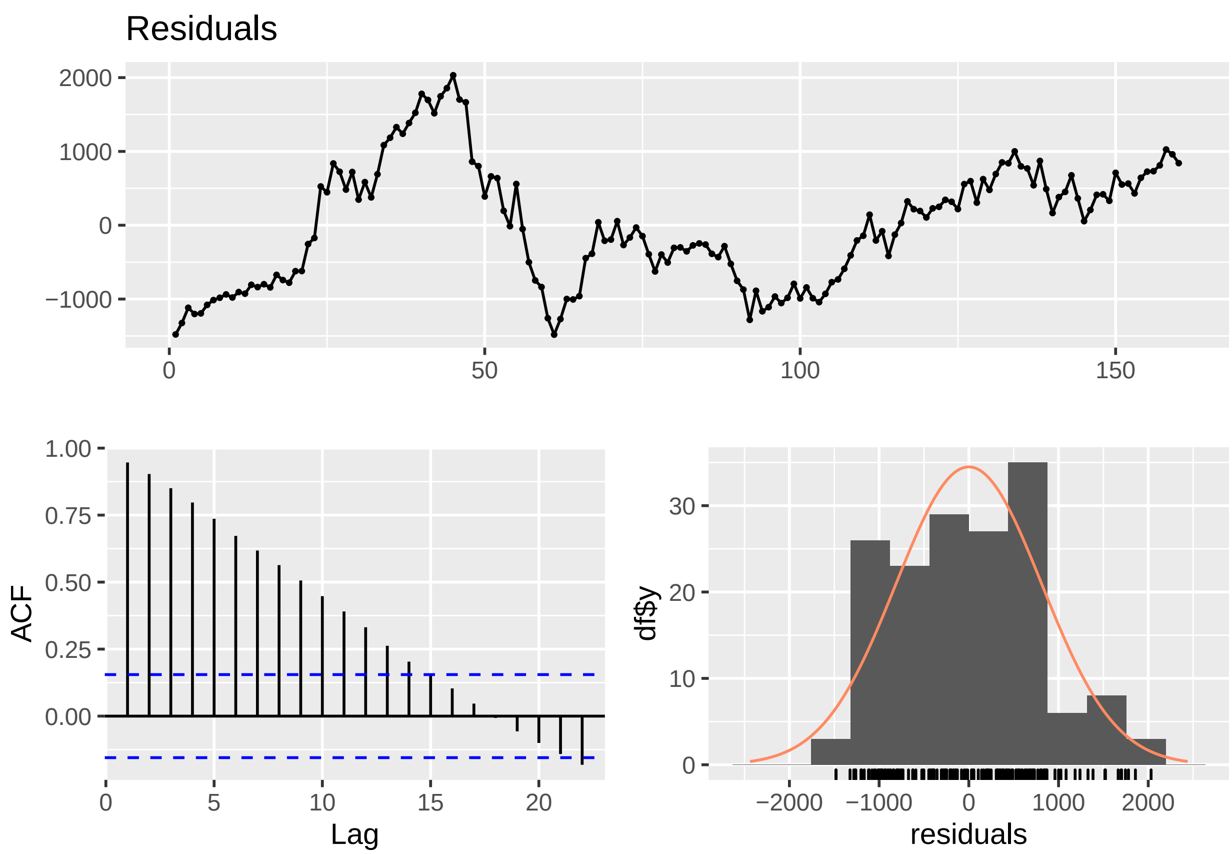
```
## copper.price.t copper.price.1
##       19.66222       19.66222
```

```
modeldlm4 <- dlm(formula = asx.price ~ gold.price + oil.price, data = data.frame(asx.all), q = 1)
summary(modeldlm4)
```

```
##
## Call:
## lm(formula = as.formula(model.formula), data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1482.20  -749.39   -22.02   587.36  2031.87
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3710.9425   224.2137  16.551   <2e-16 ***
## gold.price.t    0.6760     1.2729   0.531   0.5962
## gold.price.1   -0.1166     1.2671  -0.092   0.9268
## oil.price.t    18.7135    10.5765   1.769   0.0788 .
## oil.price.1   -12.7946    10.6189  -1.205   0.2301
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 823.4 on 155 degrees of freedom
## Multiple R-squared:  0.1548, Adjusted R-squared:  0.133
## F-statistic: 7.097 on 4 and 155 DF,  p-value: 2.84e-05
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2609.286 2627.737
```

Figure 37. Residuals of DLM 4

```
checkresiduals(modeldlm4$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 147.4, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modeldlm4$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modeldlm4$model$fitted.values
## W = 0.94887, p-value = 1.441e-05
```

```
finiteDLMauto(formula = asx.price ~ gold.price + oil.price, data = data.frame(asx.all), q.min = 1, q.ma
              model.type = "dlm", error.type = "AIC", trace = TRUE)
```

```
##    q - k    MASE      AIC      BIC   GMRAE    MBRAE R.Adj.Sq Ljung-Box
## 10    10 3.85968 2472.752 2545.167 4.51817  0.80771 -0.04150         0
## 9      9 3.91880 2487.041 2553.566 4.59091 14.71560 -0.01713         0
## 8      8 3.97758 2501.647 2562.255 4.72289  1.69754  0.00681         0
## 7      7 4.03980 2516.044 2570.710 4.88556  0.30391  0.02917         0
## 6      6 4.08550 2531.088 2579.782 4.79178  1.42973  0.04879         0
## 5      5 4.13881 2546.306 2589.004 4.91945 -0.75355  0.06897         0
```

```
## 4      4 4.20872 2561.334 2598.008 5.09480  0.77841  0.08902         0
## 3      3 4.28929 2577.210 2607.836 5.00974 -6.04016  0.10274         0
## 2      2 4.35728 2593.043 2617.594 5.24889  1.33144  0.11879         0
## 1      1 4.42238 2609.286 2627.737 5.26525  1.44174  0.13299         0
```

```
vif(modeldlm4$model)
```

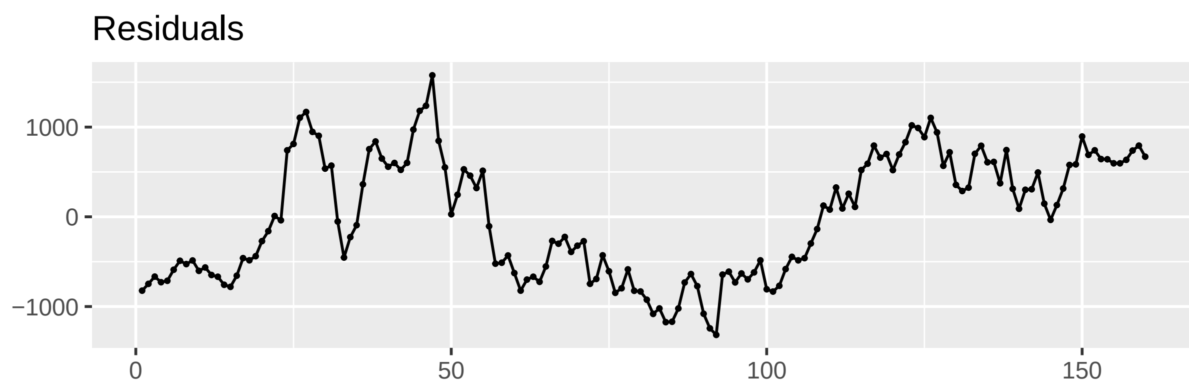
```
## gold.price.t gold.price.1  oil.price.t  oil.price.1
##     60.71945     60.46120     23.60226     24.00544
```

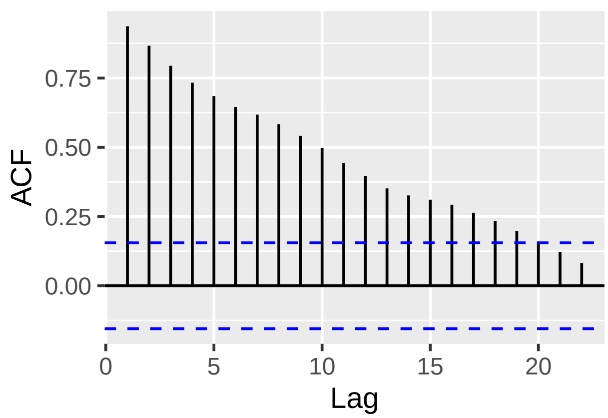
```
modeldlm5 <- dlm(formula = asx.price ~ oil.price + copper.price, data = data.frame(asx.all), q = 1)
summary(modeldlm5)
```

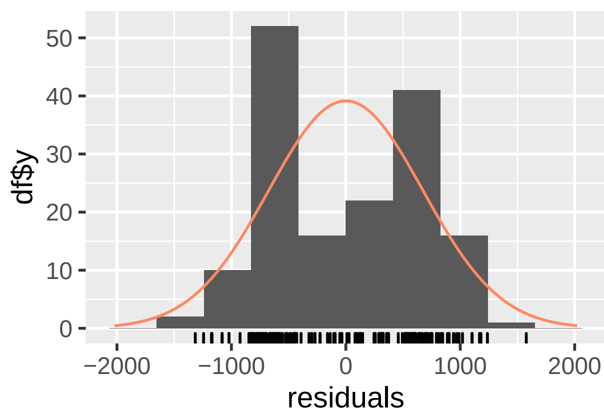
```
##
## Call:
## lm(formula = as.formula(model.formula), data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1315.64  -621.12   -12.58   600.44  1577.04
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3448.5518   164.1457  21.009   <2e-16 ***
## oil.price.t      -4.9220    10.4942  -0.469   0.6397
## oil.price.1     -15.5468    10.3062  -1.508   0.1335
## copper.price.t    0.1925     0.1365   1.410   0.1605
## copper.price.1    0.2885     0.1377   2.096   0.0377 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 682.5 on 155 degrees of freedom
## Multiple R-squared:  0.4194, Adjusted R-squared:  0.4044
## F-statistic: 27.99 on 4 and 155 DF,  p-value: < 2.2e-16
##
## AIC and BIC values for the model:
##        AIC     BIC
## 1 2549.209 2567.66
```

Figure 38. Residuals of DLM 5

```
checkresiduals(modeldlm5$model)
```








```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 143.45, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modeldlm5$model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modeldlm5$model$residuals
## W = 0.94312, p-value = 4.829e-06
```

```
finiteDLMauto(formula = asx.price ~ oil.price + copper.price, data = data.frame(asx.all), q.min = 1, q.
              model.type = "dlm", error.type = "AIC", trace = TRUE)
```

```
##     q - k    MASE      AIC      BIC   GMRAE   MBRAE R.Adj.Sq Ljung-Box
## 10    10 3.63201 2430.576 2502.990 4.48494 0.85265  0.21231         0
## 9      9 3.65878 2443.222 2509.748 4.75033 0.74957  0.23761         0
## 8      8 3.67126 2455.995 2516.604 4.65954 0.61318  0.26303         0
## 7      7 3.69432 2468.719 2523.385 4.67690 0.90865  0.28603         0
## 6      6 3.70478 2480.947 2529.642 4.67984 0.09549  0.31168         0
## 5      5 3.71299 2493.210 2535.908 4.63363 1.46809  0.33756         0
```

```
## 4      4 3.73284 2505.371 2542.046 4.65201 0.96352  0.36217         0
## 3      3 3.77678 2518.556 2549.182 4.76978 1.06979  0.38099         0
## 2      2 3.82120 2532.714 2557.265 4.80688 0.19570  0.39703         0
## 1      1 3.89540 2549.209 2567.660 5.04160 1.11662  0.40439         0
```

```
vif(modeldlm5$model)
```

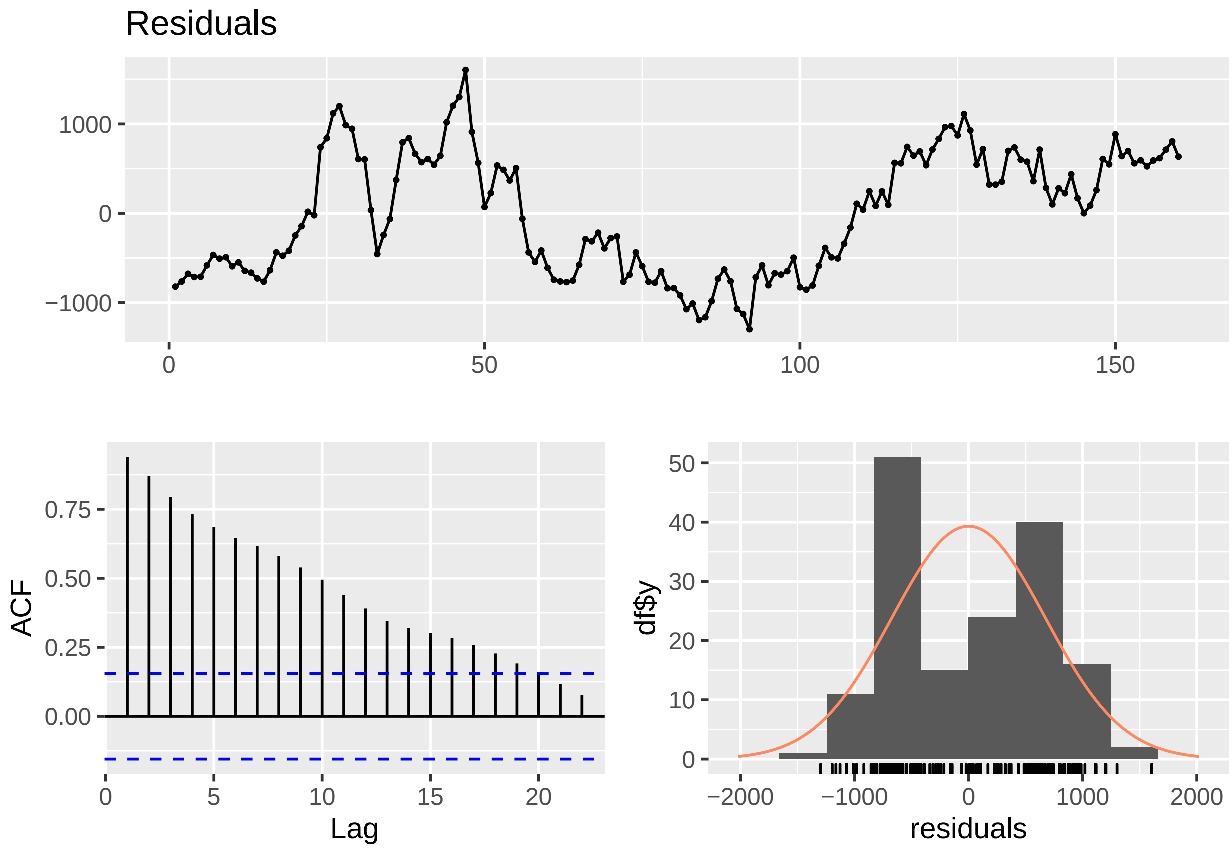
```
##    oil.price.t    oil.price.1 copper.price.t copper.price.1
##       33.82536       32.91676       28.33366       29.56691
```

```
modeldlm6 <- dlm(formula = asx.price ~ gold.price + oil.price + copper.price, data = data.frame(asx.all
summary(modeldlm6)
```

```
##
## Call:
## lm(formula = as.formula(model.formula), data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1296.50  -630.93    26.13   600.90  1603.94
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3419.8564   191.2024  17.886   <2e-16 ***
## gold.price.t     -0.5698     1.0708  -0.532    0.595
## gold.price.1      0.6204     1.0592   0.586    0.559
## oil.price.t      -5.6065    10.7857  -0.520    0.604
## oil.price.1     -14.9472    10.5135  -1.422    0.157
## copper.price.t    0.1920     0.1374   1.398    0.164
## copper.price.1    0.2854     0.1391   2.051    0.042 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 685.9 on 153 degrees of freedom
## Multiple R-squared:  0.421,  Adjusted R-squared:  0.3983
## F-statistic: 18.54 on 6 and 153 DF,  p-value: 3.895e-16
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2552.754 2577.355
```

Figure 39. Residuals of DLM 6

```
checkresiduals(modeldlm6$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 144.4, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modeldlm6$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modeldlm6$model$fitted.values
## W = 0.96781, p-value = 0.0008735
```

```
finiteDLMauto(formula = asx.price ~ gold.price + oil.price + copper.price, data = data.frame(asx.all),
              model.type = "dlm", error.type = "AIC", trace = TRUE)
```

```
##     q - k    MASE      AIC      BIC   GMRAE    MBRAE R.Adj.Sq Ljung-Box
## 10    10 3.45015 2440.112 2545.716 4.17908 -0.09314  0.20653         0
## 9      9 3.51306 2452.691 2549.455 4.32629  0.92363  0.22986         0
## 8      8 3.54824 2464.680 2552.563 4.33153  1.13867  0.25664         0
## 7      7 3.58483 2476.180 2555.141 4.18334  1.59915  0.28265         0
## 6      6 3.61856 2488.492 2558.491 4.36746  1.11068  0.30501         0
## 5      5 3.63787 2500.567 2561.564 4.45774  0.93643  0.32883         0
```

```
## 4      4 3.67485 2513.098 2565.055 4.26210  1.02182  0.34904         0
## 3      3 3.75026 2525.161 2568.037 4.80762  0.61340  0.36951         0
## 2      2 3.80734 2538.017 2571.775 4.87590  0.68063  0.38758         0
## 1      1 3.87272 2552.754 2577.355 4.84385  0.91834  0.39832         0
```

```r
vif(modeldlm6$model)
```

```
##   gold.price.t   gold.price.1    oil.price.t    oil.price.1 copper.price.t
##       61.91094       60.88159       35.36972       33.90848       28.40662
## copper.price.1
##       29.90281
```

```r
AIC(modeldlm1$model, modeldlm2$model, modeldlm3$model, modeldlm4$model, modeldlm5$model, modeldlm6$model
```

```
##                  df      AIC
## modeldlm1$model  4 2613.609
## modeldlm2$model  4 2614.698
## modeldlm3$model  4 2574.488
## modeldlm4$model  6 2609.286
## modeldlm5$model  6 2549.209
## modeldlm6$model  8 2552.754
```

Polynomial Distributed Lag Models

Three polynomial distributed lag models (PDLMs), one for each independent variable, were implemented to attempt to explain variance similarly to the DLM models without violating multicollinearity. All models were statistically significant at `p < 0.05`, although all the coefficients in all models were not significant at `p < 0.05`. Additionally, all models had nonnormaly distributed residuals with serial correlation and all models violated multicollinearity. This demonstrates that the models are not suitable to explain the data and do so poorly. A `q = 1` and `k = 1` was chosen to maximise adjusted R squared since these model encountered the same problem as DLM models, in that AIC/BIC would be optimised only through adding nonsignificant lags. With these specifications, the third model performed the best, demonstrating a adjusted R squared value of `0.294` and the lowest AIC value.

The third polynomial model was decided by the evaluative measures as the best model. It had the lowest AIC score, in addition to having significantly higher adjusted R squared than the other models.

Polynomial DLM output

```r
modelpoly1 <- polyDlm(x = as.vector(gold), y = as.vector(asx), q = 1, k = 1)
```

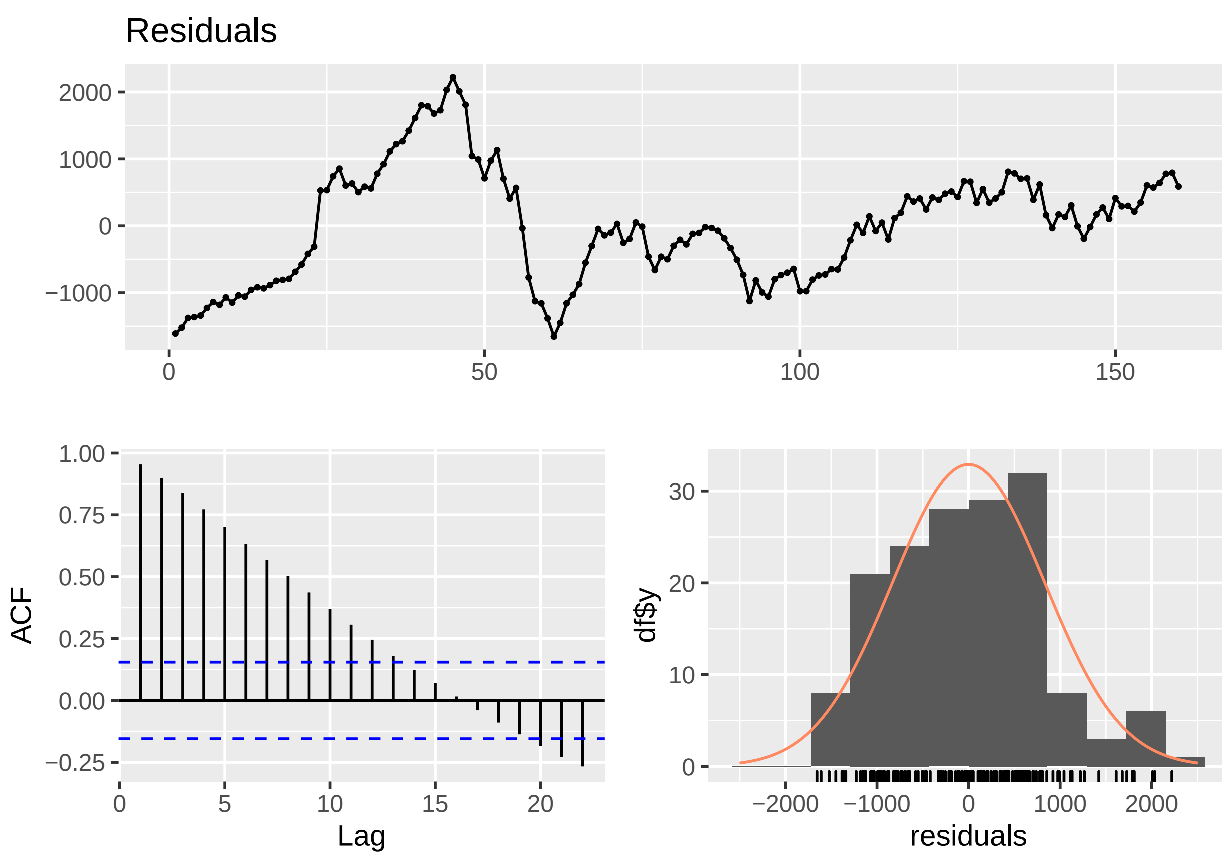
```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    0.412       1.27   0.323   0.747
## beta.1    0.322       1.27   0.253   0.801
```

```r
summary(modelpoly1)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1654.13  -706.62    -9.17   561.27  2219.74
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3943.03376  211.54676  18.639   <2e-16 ***
## z.t0           0.41169    1.27421   0.323    0.747
## z.t1          -0.09005    2.53981  -0.035    0.972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 839.7 on 157 degrees of freedom
## Multiple R-squared:  0.1097, Adjusted R-squared:  0.09832
## F-statistic: 9.669 on 2 and 157 DF,  p-value: 0.0001097
```

Figure 40. Residuals of PDLM 1

```
checkresiduals(modelpoly1$model)
```



```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 148.3, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modelpoly1$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelpoly1$model$fitted.values
## W = 0.88545, p-value = 8.871e-10
```

```
finiteDLMauto(x = as.vector(gold), y = as.vector(asx), data = data.frame(asx.all), q.min = 1, q.max = 5
              model.type = "poly", error.type = "AIC", trace = TRUE)
```

```
##   q - k    MASE      AIC      BIC   GMRAE    MBRAE R.Adj.Sq Ljung-Box
## 5 5 - 1 4.16400 2537.143 2549.342 4.42732  4.82889  0.06723         0
## 4 4 - 1 4.21758 2556.384 2568.609 4.57720  0.96264  0.07336         0
## 3 3 - 1 4.26750 2575.233 2587.484 4.52980 -0.55732  0.08099         0
## 2 2 - 1 4.30551 2594.354 2606.629 4.47648  4.03215  0.08961         0
## 1 1 - 1 4.35631 2613.609 2625.910 4.56013  1.06887  0.09832         0
```

```
vif(modelpoly1$model)
```

```
##     z.t0     z.t1
## 233.5819 233.5819
```

```
modelpoly2 <- polyDlm(x = as.vector(oil), y = as.vector(asx), q = 1, k = 1)
```

```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    17.10       10.6    1.61   0.109
## beta.1    -7.94       10.6   -0.75   0.454
```

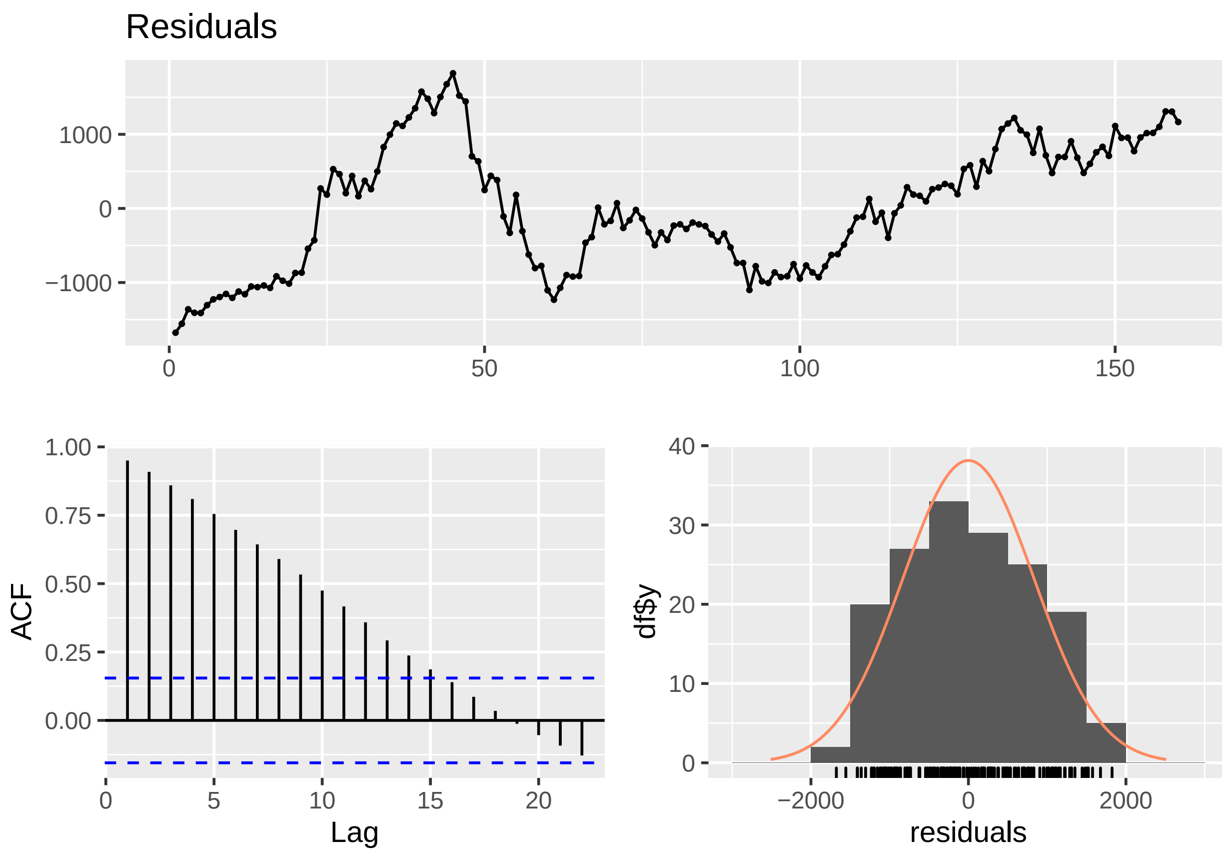
```
summary(modelpoly2)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1677.08  -770.94   -61.63   694.56  1823.95
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4144.34     178.03  23.278   <2e-16 ***
## z.t0           17.14      10.64   1.611    0.109
```

```
## z.t1          -25.08      21.11  -1.188    0.237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 842.6 on 157 degrees of freedom
## Multiple R-squared:  0.1036, Adjusted R-squared:  0.09217
## F-statistic: 9.071 on 2 and 157 DF,  p-value: 0.000187
```

Figure 41. Residuals of PDLM 2

```
checkresiduals(modelpoly2$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 147.28, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modelpoly2$model$fitted.values)
```

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

```
## data:  modelpoly2$model$fitted.values
## W = 0.95055, p-value = 2.008e-05
```

```
finiteDLMauto(x = as.vector(oil), y = as.vector(asx), data = data.frame(asx.all), q.min = 1, q.max = 5,
              model.type = "poly", error.type = "AIC", trace = TRUE)
```

```
##   q - k    MASE      AIC      BIC   GMRAE   MBRAE R.Adj.Sq Ljung-Box
## 5 5 - 1 4.32409 2537.158 2549.357 4.97791 2.10458  0.06714         0
## 4 4 - 1 4.39655 2556.008 2568.233 5.10561 0.99246  0.07558         0
## 3 3 - 1 4.47072 2575.112 2587.362 5.19574 5.89283  0.08169         0
## 2 2 - 1 4.53561 2594.715 2606.991 5.36922 0.85959  0.08754         0
## 1 1 - 1 4.60656 2614.698 2626.998 5.46829 0.63464  0.09217         0
```

```
vif(modelpoly2$model)
```

```
##     z.t0     z.t1
## 90.58266 90.58266
```

```
modelpoly3 <- polyDlm(x = as.vector(copper), y = as.vector(asx), q = 1, k = 1)
```

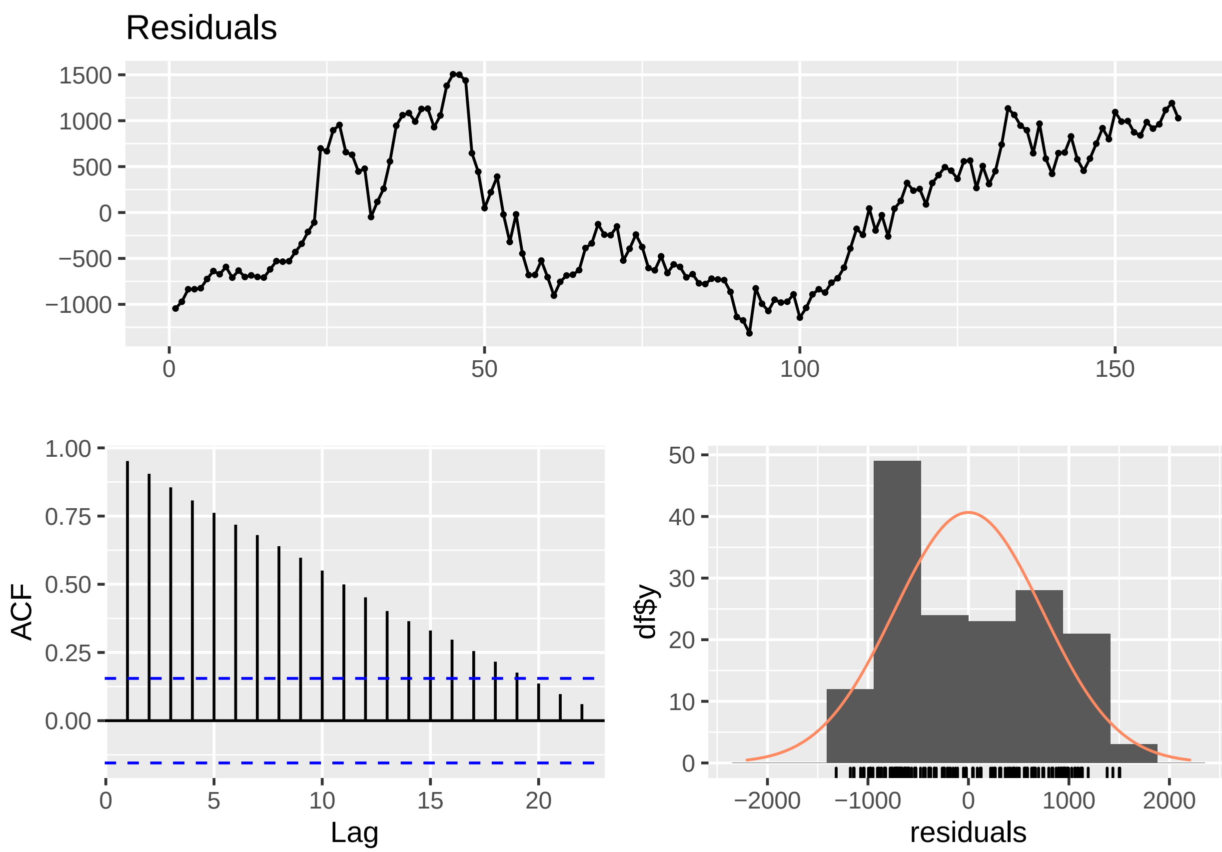
```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1850      0.124   1.490   0.138
## beta.1   0.0462      0.122   0.378   0.706
```

```
summary(modelpoly3)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1315.8  -679.5  -117.9   646.4  1506.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3436.7102   178.1110  19.295   <2e-16 ***
## z.t0           0.1847     0.1238   1.492    0.138
## z.t1          -0.1385     0.2445  -0.566    0.572
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 743.1 on 157 degrees of freedom
## Multiple R-squared:  0.3028, Adjusted R-squared:  0.2939
## F-statistic: 34.09 on 2 and 157 DF,  p-value: 5.06e-13
```

Figure 42. Residuals of PDLM 3

```
checkresiduals(modelpoly3$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 147.66, df = 10, p-value < 2.2e-16
```

```
shapiro.test(modelpoly3$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelpoly3$model$fitted.values
## W = 0.94527, p-value = 7.221e-06
```

```
finiteDLMauto(x = as.vector(copper), y = as.vector(asx), data = data.frame(asx.all), q.min = 1, q.max =
              model.type = "poly", error.type = "AIC", trace = TRUE)
```

```
##    q - k    MASE      AIC      BIC   GMRAE    MBRAE R.Adj.Sq Ljung-Box
## 5 5 - 1 4.11367 2505.646 2517.846 5.08586  0.76203  0.23776         0
## 4 4 - 1 4.15203 2523.100 2535.325 5.05822 -0.05088  0.25039         0
## 3 3 - 1 4.19205 2540.208 2552.459 5.15270  0.42877  0.26371         0
```

```
## 2 2 - 1 4.21449 2557.367 2569.643 5.30198  0.73653  0.27856         0
## 1 1 - 1 4.24262 2574.488 2586.789 5.38993 -0.49453  0.29391         0
```

```
vif(modelpoly3$model)
```

```
##     z.t0     z.t1
## 78.66294 78.66294
```

```
AIC(modelpoly1$model, modelpoly2$model, modelpoly3$model)
```

```
##                   df      AIC
## modelpoly1$model  4 2613.609
## modelpoly2$model  4 2614.698
## modelpoly3$model  4 2574.488
```

Koyck distributed lag models

Similar to the PDLM models, 3 Koyck models were implemented, one for each independent variable. All 3 models are statistically significant, although each model only had the first lag as a significant coefficient. Additionally, each model had an adjusted R squared of between 0.94 and 0.95, demonstrating good ability to explain variance. Although the residuals are not completely random, they look more random than the previous models, demonstrating an improvement. The residuals were confirmed through a shapiro-wilks test to be nonnormally distributed. Finally, all the models satisfied the assumption of multicollinearity.

The second Koyck model using oil as the independent variable was decided to be the best model. This model was evaluated to be the best by the AIC with it also having the highest adjusted R squared value. However, all the Koyck models were relatively close in when evaluated and the second model. Although the second model was by evaluative measures the best, it was not a significant improvement on the other two.

Koyck DLM output

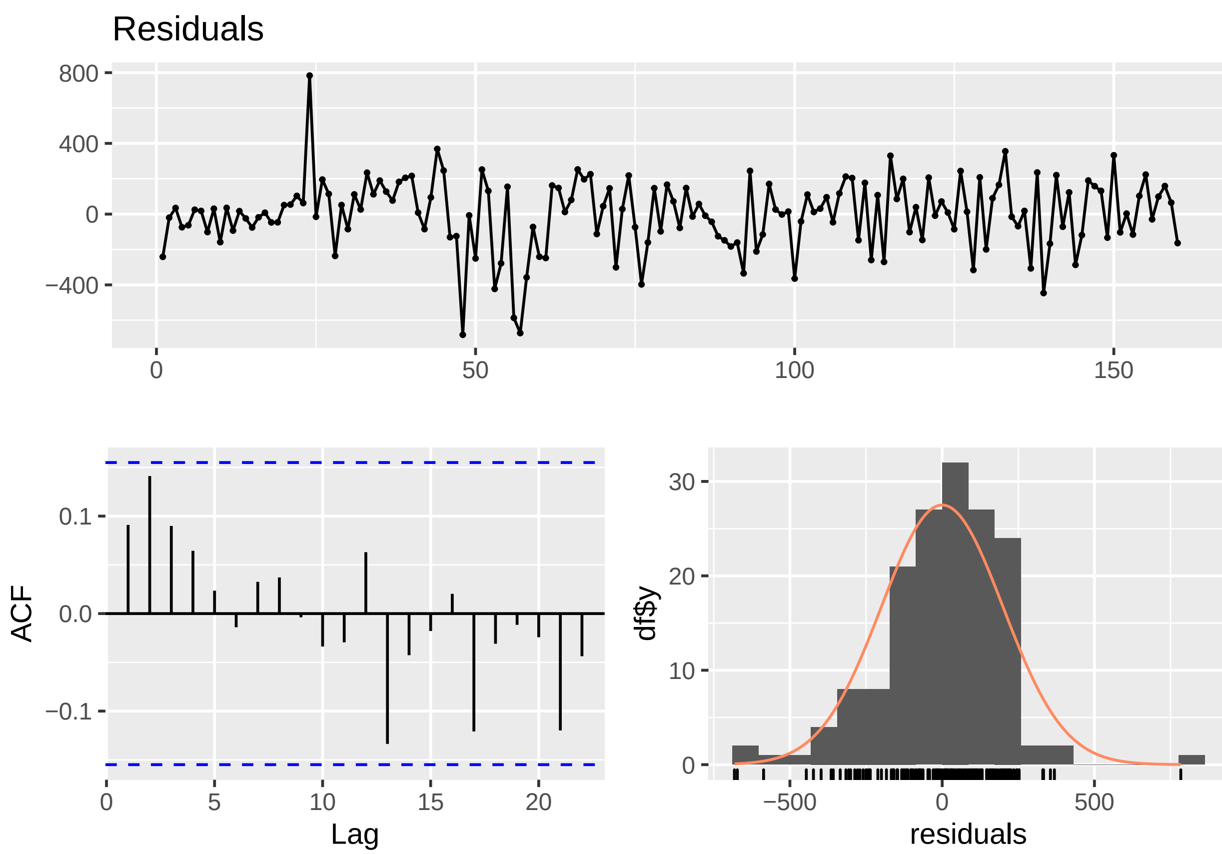
```
modelk1 <- koyckDlm(x = as.vector(gold), y = as.vector(asx))
summary(modelk1)
```

```
##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -682.19 -105.44   15.86  135.04  783.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.902e+02  8.958e+01   2.123   0.0353 *
## Y.1         9.635e-01  1.909e-02  50.469   <2e-16 ***
## X.t         2.595e-03  4.304e-02   0.060   0.9520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 201.4 on 157 degrees of freedom
## Multiple R-Squared: 0.9488,  Adjusted R-squared: 0.9481
## Wald test:  1454 on 2 and 157 DF,  p-value: < 2.2e-16
##
## Diagnostic tests:
## NULL
##
##                              alpha        beta       phi
## Geometric coefficients:  5205.15 0.002595168 0.9634602
```

Figure 43. Residuals of Koyck DLM 1

```
checkresiduals(modelk1$model)
```




```
shapiro.test(modelk1$model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelk1$model$residuals
## W = 0.96064, p-value = 0.0001672
```

```r
vif(modelk1$model)
```

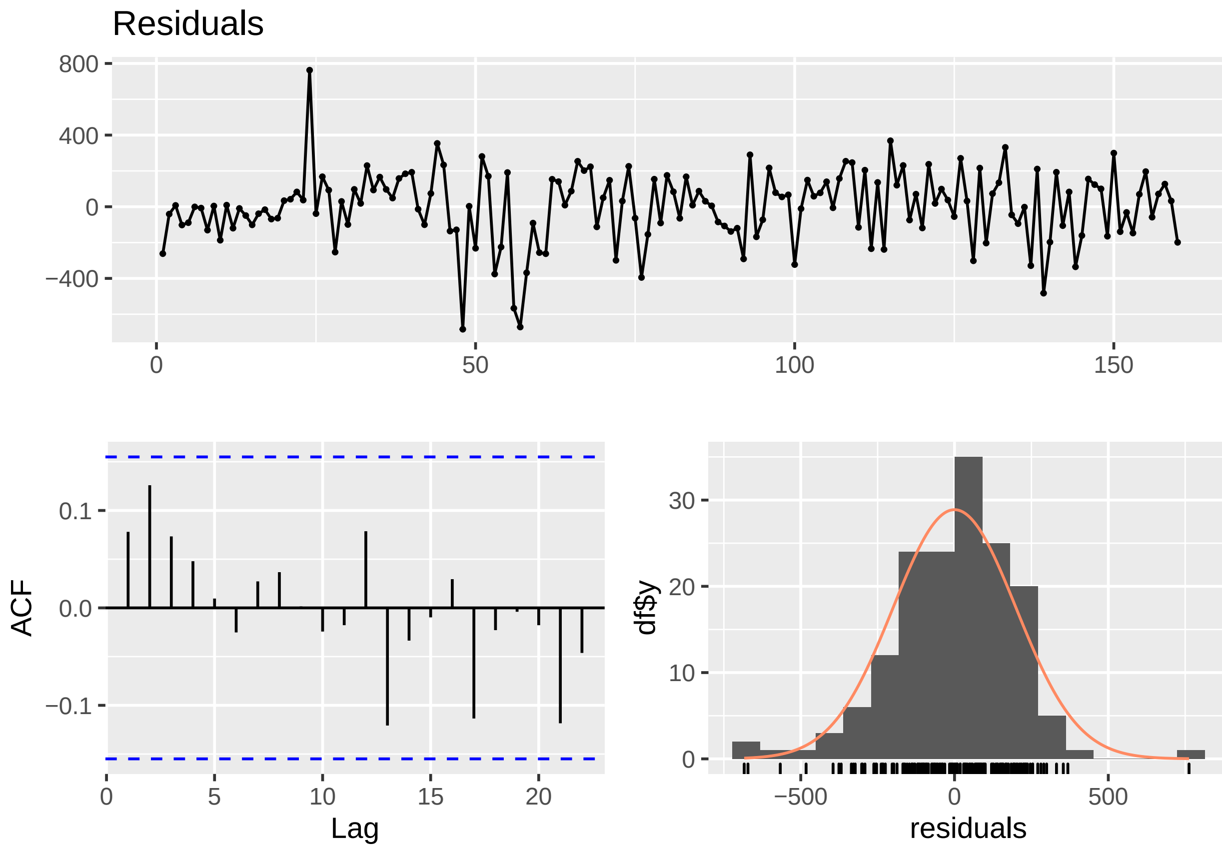
```
##      Y.1      X.t
## 1.140648 1.140648
```

```r
modelk2 <- koyckDlm(x = as.vector(oil), y = as.vector(asx))
summary(modelk2)
```

```
##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -683.91 -108.66   13.68  139.77  762.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 209.89536   87.89368   2.388   0.0181 *
## Y.1           0.97537    0.01905  51.193   <2e-16 ***
## X.t          -0.99907    0.58045  -1.721   0.0872 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.1 on 157 degrees of freedom
## Multiple R-Squared: 0.949,   Adjusted R-squared: 0.9483
## Wald test:  1461 on 2 and 157 DF,  p-value: < 2.2e-16
##
## Diagnostic tests:
## NULL
##
##                              alpha       beta       phi
## Geometric coefficients:  8522.034 -0.9990694 0.9753703
```

Figure 44. Residuals of Koyck DLM 2

```r
checkresiduals(modelk2$model)
```




```
vif(modelk2$model)
```

```
##      Y.1      X.t
## 1.14038 1.14038
```

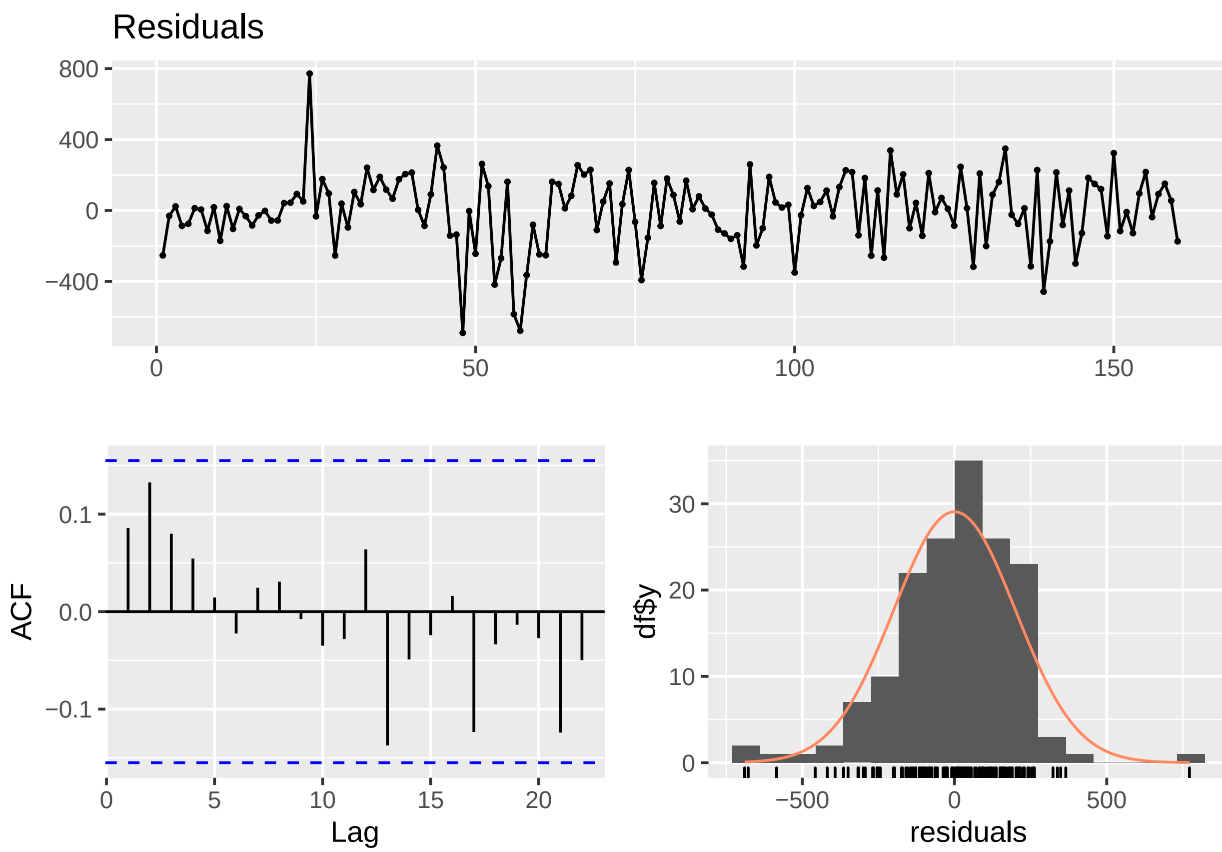
```
modelk3 <- koyckDlm(x = as.vector(copper), y = as.vector(asx))
summary(modelk3)
```

```
##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -689.64 -108.62   12.78  140.20  771.79
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 189.368812  87.644648   2.161   0.0322 *
## Y.1           0.971621   0.021895  44.376   <2e-16 ***
## X.t          -0.005864   0.009517  -0.616   0.5387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.9 on 157 degrees of freedom
```

```
## Multiple R-Squared: 0.9485,  Adjusted R-squared: 0.9479
## Wald test:  1448 on 2 and 157 DF,  p-value: < 2.2e-16
##
## Diagnostic tests:
## NULL
##
##                              alpha         beta       phi
## Geometric coefficients:  6672.885 -0.005863623 0.9716211
```

Figure 45. Residuals of Koyck DLM 3

```
checkresiduals(modelk3$model)
```




```
vif(modelk3$model)
```

```
##      Y.1      X.t
## 1.493966 1.493966
```

```
AIC(modelk1); AIC(modelk2); AIC(modelk3)
```

```
## [1] 2156.757
```

```
## [1] 2156.172
```

```
## [1] 2157.456
```

Autoregressive Distributed Lag Models

The autoregressive distributed lag models (ARDLM) demonstrated a significant improvement on the DLM and PDLM models. In addition to all the models being significant, the models demonstrated adjusted R squared between `0.95` and `0.96` with most models having all coefficients being significant at `q = 1` and `p = 1`. Additionally, these models seemed to effectively eliminate autocorrelation in the residuals. The residuals from these models appeared more normally distributed than previous models, although Shapiro Wilk tests rejected the null of normality. For these models similar pattern presented as in the DLM models, such that AIC/BIC favour more parameters. Finally, these modeled violated the multicollinearity assumption, with the components of the independent series violating the assumption.

The best model of the ARDL models was the second model, which used oil price as the dependent variable for ASX price. This model had $q = 1$ and $p = 1$, and demonstrated an adjusted R squared of `0.952` with each coefficient significant. Additionally, it had the second lowest AIC score, only being higher than model, which had more parameters with not all significant.

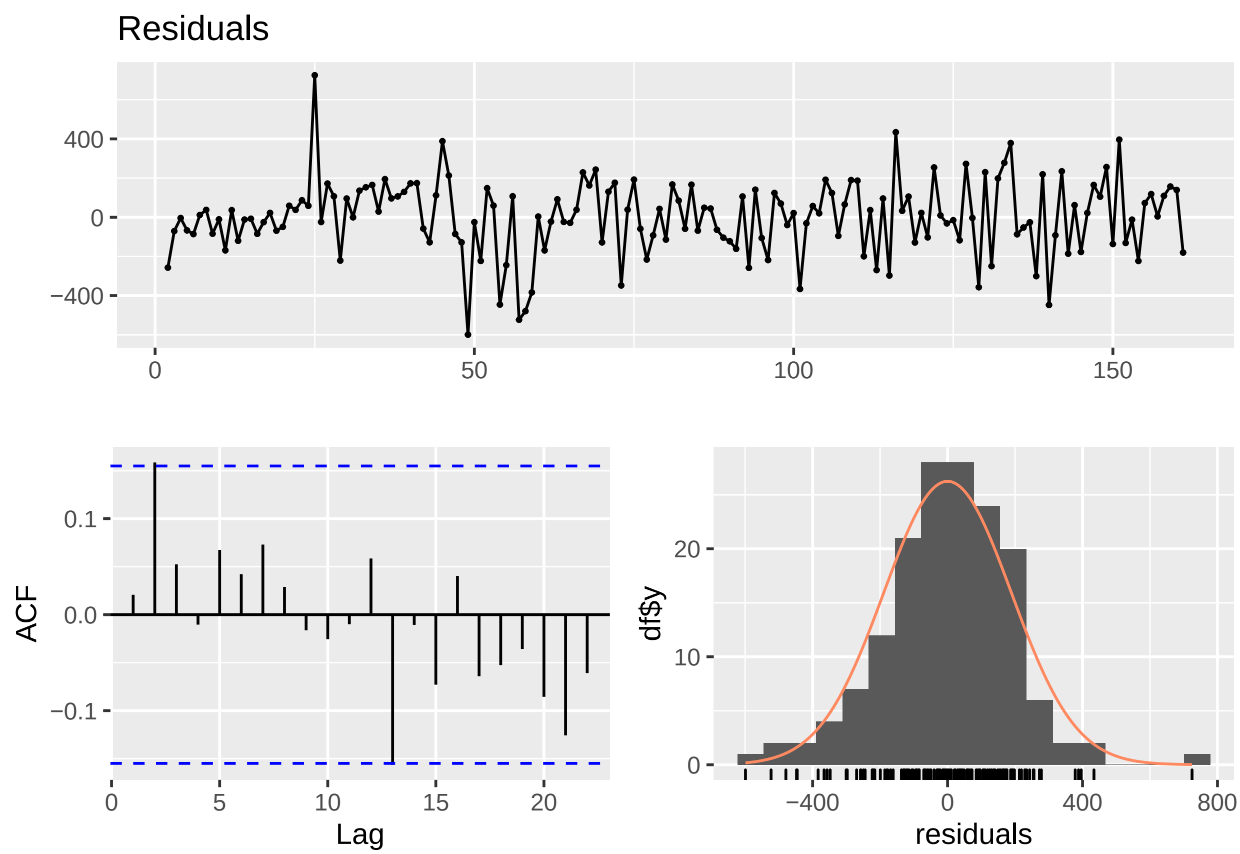
ARDLM Output

```
modelar1 <- ardlDlm(formula = asx.price ~ gold.price, data = data.frame(asx.all), q = 1, p = 1)
summary(modelar1)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -598.9 -102.9   10.4  119.5  724.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  182.80452   85.05812   2.149   0.0332 *
## gold.price.t  -1.24911    0.29162  -4.283 3.21e-05 ***
## gold.price.1   1.23169    0.28976   4.251 3.66e-05 ***
## asx.price.1    0.97172    0.01812  53.624  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191.1 on 156 degrees of freedom
## Multiple R-squared:  0.9542, Adjusted R-squared:  0.9533
## F-statistic:  1083 on 3 and 156 DF,  p-value: < 2.2e-16
```

Figure 46. Residuals of ARDLM 1

```
checkresiduals(modelar1$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 7.9182, df = 10, p-value = 0.6368
```

```
shapiro.test(modelar1$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelar1$model$fitted.values
## W = 0.97281, p-value = 0.003015
```

```
vif(modelar1$model)
```

```
##       gold.price L(gold.price, 1)  L(asx.price, 1)
##         59.17040         58.70438          1.14186
```

```
ardlm.search <- function(formula, data, p, q){
  for (i in 1:p){
    for (j in 1:q){
      modelx <- ardlDlm(formula, data = data.frame(data), p = i, q = j)
      cat("q = ", i, "q = ", j, "AIC = ", AIC(modelx$model), "BIC = ", BIC(modelx$model),"\n")
```

```r
    }
  }
}

ardlm.search(formula = asx.price ~ gold.price, asx.all, 5, 5)
```

```
## q =  1 q =  1 AIC =  2140.897 BIC =  2156.273
## q =  1 q =  2 AIC =  2128.524 BIC =  2146.938
## q =  1 q =  3 AIC =  2113.99 BIC =  2135.428
## q =  1 q =  4 AIC =  2102.754 BIC =  2127.204
## q =  1 q =  5 AIC =  2092.194 BIC =  2119.643
## q =  2 q =  1 AIC =  2128.627 BIC =  2147.04
## q =  2 q =  2 AIC =  2130.523 BIC =  2152.005
## q =  2 q =  3 AIC =  2115.89 BIC =  2140.39
## q =  2 q =  4 AIC =  2104.694 BIC =  2132.2
## q =  2 q =  5 AIC =  2094.14 BIC =  2124.639
## q =  3 q =  1 AIC =  2118.109 BIC =  2139.547
## q =  3 q =  2 AIC =  2120.027 BIC =  2144.528
## q =  3 q =  3 AIC =  2117.305 BIC =  2144.868
## q =  3 q =  4 AIC =  2105.731 BIC =  2136.293
## q =  3 q =  5 AIC =  2095.264 BIC =  2128.812
## q =  4 q =  1 AIC =  2107.002 BIC =  2131.452
## q =  4 q =  2 AIC =  2108.914 BIC =  2136.42
## q =  4 q =  3 AIC =  2106.276 BIC =  2136.839
## q =  4 q =  4 AIC =  2107.456 BIC =  2141.074
## q =  4 q =  5 AIC =  2097.01 BIC =  2133.608
## q =  5 q =  1 AIC =  2094.908 BIC =  2122.357
## q =  5 q =  2 AIC =  2096.86 BIC =  2127.359
## q =  5 q =  3 AIC =  2094.144 BIC =  2127.692
## q =  5 q =  4 AIC =  2095.425 BIC =  2132.023
## q =  5 q =  5 AIC =  2097.324 BIC =  2136.972
```

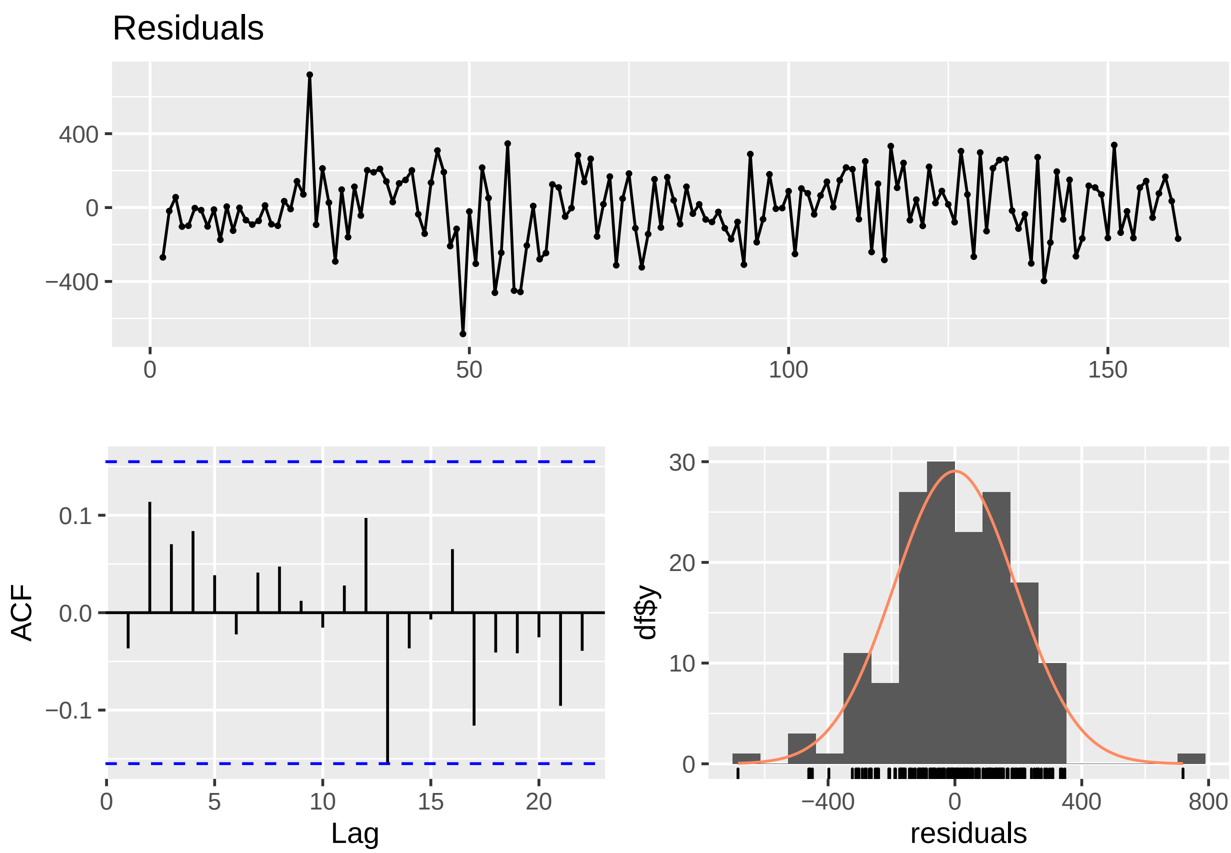
```r
modelar2 <- ardlDlm(formula = asx.price ~ oil.price, data = data.frame(asx.all), q = 1, p = 1)
summary(modelar2)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -684.06 -111.57   -1.77  138.90  719.35
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 211.45397   85.03705   2.487  0.01395 *
## oil.price.t   7.45111    2.46199   3.026  0.00290 **
## oil.price.1  -8.17917    2.44422  -3.346  0.00103 **
## asx.price.1   0.97067    0.01837  52.827  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 194.5 on 156 degrees of freedom
## Multiple R-squared:  0.9525, Adjusted R-squared:  0.9516
## F-statistic:  1044 on 3 and 156 DF,  p-value: < 2.2e-16
```

Figure 47. Residuals of ARDLM 2

```
checkresiduals(modelar2$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 6.0252, df = 10, p-value = 0.8131
```

```
shapiro.test(modelar2$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelar2$model$fitted.values
## W = 0.97667, p-value = 0.00826
```

```r
vif(modelar2$model)
```

```
##       oil.price L(oil.price, 1) L(asx.price, 1)
##       22.924733       22.797583        1.133443
```

```r
ardlm.search(formula = asx.price ~ oil.price, asx.all, 5, 5)
```

```
## q =  1 q =  1 AIC =  2146.524 BIC =  2161.9
## q =  1 q =  2 AIC =  2134.107 BIC =  2152.521
## q =  1 q =  3 AIC =  2121.07 BIC =  2142.508
## q =  1 q =  4 AIC =  2109.4 BIC =  2133.85
## q =  1 q =  5 AIC =  2098.335 BIC =  2125.784
## q =  2 q =  1 AIC =  2132.312 BIC =  2150.726
## q =  2 q =  2 AIC =  2134.235 BIC =  2155.718
## q =  2 q =  3 AIC =  2122.356 BIC =  2146.857
## q =  2 q =  4 AIC =  2110.793 BIC =  2138.299
## q =  2 q =  5 AIC =  2099.752 BIC =  2130.251
## q =  3 q =  1 AIC =  2121.919 BIC =  2143.357
## q =  3 q =  2 AIC =  2123.835 BIC =  2148.335
## q =  3 q =  3 AIC =  2124.324 BIC =  2151.887
## q =  3 q =  4 AIC =  2112.401 BIC =  2142.963
## q =  3 q =  5 AIC =  2101.35 BIC =  2134.899
## q =  4 q =  1 AIC =  2111.383 BIC =  2135.832
## q =  4 q =  2 AIC =  2113.294 BIC =  2140.8
## q =  4 q =  3 AIC =  2113.805 BIC =  2144.367
## q =  4 q =  4 AIC =  2114.384 BIC =  2148.003
## q =  4 q =  5 AIC =  2103.342 BIC =  2139.94
## q =  5 q =  1 AIC =  2097.076 BIC =  2124.525
## q =  5 q =  2 AIC =  2099.041 BIC =  2129.54
## q =  5 q =  3 AIC =  2099.518 BIC =  2133.066
## q =  5 q =  4 AIC =  2099.845 BIC =  2136.443
## q =  5 q =  5 AIC =  2100.917 BIC =  2140.566
```

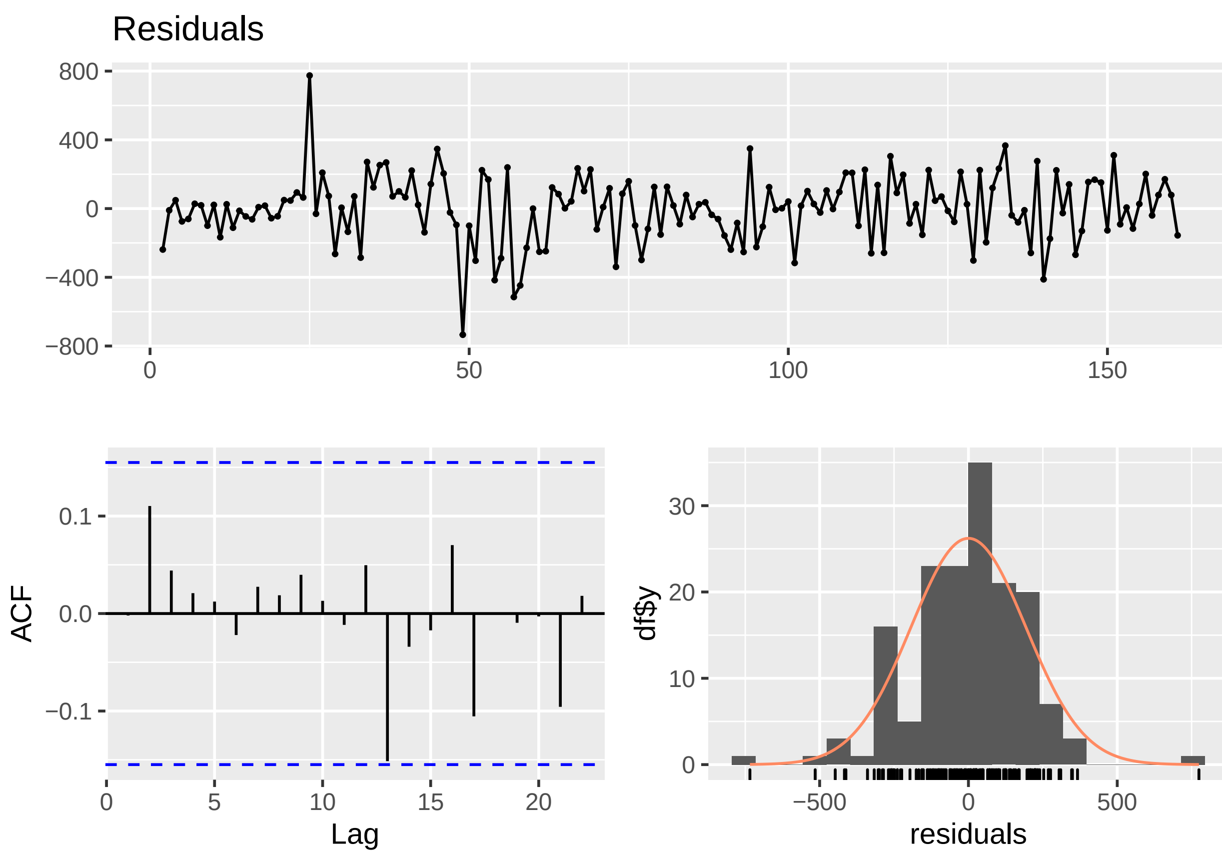
```r
modelar3 <- ardlDlm(formula = asx.price ~ copper.price, data = data.frame(asx.all), q = 1, p = 1)
summary(modelar3)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -734.34 -101.82   16.56  123.05  774.55
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     173.61066   84.94572   2.044  0.04266 *
## copper.price.t    0.10629    0.03258   3.263  0.00136 **
## copper.price.1   -0.10695    0.03228  -3.313  0.00115 **
```

```
## asx.price.1      0.96784    0.02103  46.027  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.2 on 156 degrees of freedom
## Multiple R-squared:  0.9522, Adjusted R-squared:  0.9513
## F-statistic:  1035 on 3 and 156 DF,  p-value: < 2.2e-16
```

Figure 48. Residuals of ARDLM 3

```
checkresiduals(modelar3$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 3.6178, df = 10, p-value = 0.9629
```

```
shapiro.test(modelar3$model$fitted.values)
```

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

```
## data:  modelar3$model$fitted.values
## W = 0.97128, p-value = 0.002049
```

```r
vif(modelar3$model)
```

```
##       copper.price L(copper.price, 1)    L(asx.price, 1)
##          19.716144          19.873449           1.473109
```

```r
for (i in 1:5){
  for (j in 1:5){
    modelx <- ardlDlm(formula = asx.price ~ copper.price, data = data.frame(asx.all), p = i, q = j)
    cat("q = ", i, "q = ", j, "AIC = ", AIC(modelx$model), "BIC = ", BIC(modelx$model),"\n")
  }
}
```

```
## q =  1 q =  1 AIC =  2147.741 BIC =  2163.116
## q =  1 q =  2 AIC =  2135.4 BIC =  2153.813
## q =  1 q =  3 AIC =  2121.12 BIC =  2142.558
## q =  1 q =  4 AIC =  2109.759 BIC =  2134.209
## q =  1 q =  5 AIC =  2099.056 BIC =  2126.505
## q =  2 q =  1 AIC =  2130.043 BIC =  2148.456
## q =  2 q =  2 AIC =  2132.038 BIC =  2153.52
## q =  2 q =  3 AIC =  2119.241 BIC =  2143.741
## q =  2 q =  4 AIC =  2107.649 BIC =  2135.155
## q =  2 q =  5 AIC =  2097.021 BIC =  2127.52
## q =  3 q =  1 AIC =  2117.307 BIC =  2138.745
## q =  3 q =  2 AIC =  2119.247 BIC =  2143.748
## q =  3 q =  3 AIC =  2119.696 BIC =  2147.259
## q =  3 q =  4 AIC =  2108.537 BIC =  2139.1
## q =  3 q =  5 AIC =  2097.832 BIC =  2131.38
## q =  4 q =  1 AIC =  2105.916 BIC =  2130.366
## q =  4 q =  2 AIC =  2107.774 BIC =  2135.28
## q =  4 q =  3 AIC =  2108.608 BIC =  2139.17
## q =  4 q =  4 AIC =  2110.085 BIC =  2143.704
## q =  4 q =  5 AIC =  2099.454 BIC =  2136.052
## q =  5 q =  1 AIC =  2095.118 BIC =  2122.566
## q =  5 q =  2 AIC =  2096.96 BIC =  2127.459
## q =  5 q =  3 AIC =  2097.887 BIC =  2131.436
## q =  5 q =  4 AIC =  2099.497 BIC =  2136.095
## q =  5 q =  5 AIC =  2101.419 BIC =  2141.067
```

```r
ardlm.search(formula = asx.price ~ copper.price, asx.all, 5, 5)
```

```
## q =  1 q =  1 AIC =  2147.741 BIC =  2163.116
## q =  1 q =  2 AIC =  2135.4 BIC =  2153.813
## q =  1 q =  3 AIC =  2121.12 BIC =  2142.558
## q =  1 q =  4 AIC =  2109.759 BIC =  2134.209
## q =  1 q =  5 AIC =  2099.056 BIC =  2126.505
## q =  2 q =  1 AIC =  2130.043 BIC =  2148.456
## q =  2 q =  2 AIC =  2132.038 BIC =  2153.52
## q =  2 q =  3 AIC =  2119.241 BIC =  2143.741
## q =  2 q =  4 AIC =  2107.649 BIC =  2135.155
```

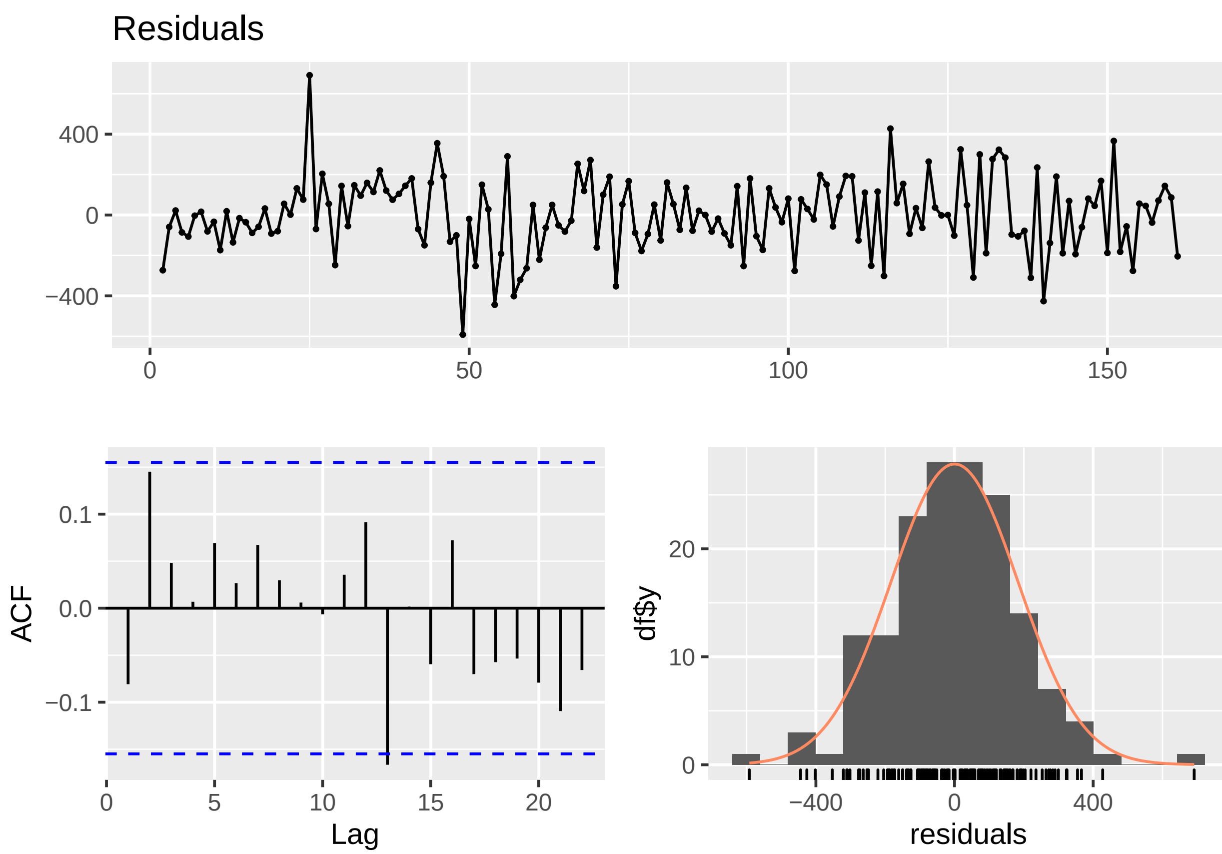
```
## q =  2 q =  5 AIC =  2097.021 BIC =  2127.52
## q =  3 q =  1 AIC =  2117.307 BIC =  2138.745
## q =  3 q =  2 AIC =  2119.247 BIC =  2143.748
## q =  3 q =  3 AIC =  2119.696 BIC =  2147.259
## q =  3 q =  4 AIC =  2108.537 BIC =  2139.1
## q =  3 q =  5 AIC =  2097.832 BIC =  2131.38
## q =  4 q =  1 AIC =  2105.916 BIC =  2130.366
## q =  4 q =  2 AIC =  2107.774 BIC =  2135.28
## q =  4 q =  3 AIC =  2108.608 BIC =  2139.17
## q =  4 q =  4 AIC =  2110.085 BIC =  2143.704
## q =  4 q =  5 AIC =  2099.454 BIC =  2136.052
## q =  5 q =  1 AIC =  2095.118 BIC =  2122.566
## q =  5 q =  2 AIC =  2096.96 BIC =  2127.459
## q =  5 q =  3 AIC =  2097.887 BIC =  2131.436
## q =  5 q =  4 AIC =  2099.497 BIC =  2136.095
## q =  5 q =  5 AIC =  2101.419 BIC =  2141.067
```

```
modelar4 <- ardlDlm(formula = asx.price ~ gold.price + oil.price, data = data.frame(asx.all), q = 1, p =
summary(modelar4)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -591.99 -101.14    0.73  123.49  691.48
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  189.95627   83.20892   2.283 0.023806 *
## gold.price.t  -1.09844    0.29064  -3.779 0.000224 ***
## gold.price.1   1.11722    0.28834   3.875 0.000158 ***
## oil.price.t    5.76679    2.41128   2.392 0.017982 *
## oil.price.1   -6.67290    2.41143  -2.767 0.006347 **
## asx.price.1    0.97475    0.01823  53.464  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 186.8 on 154 degrees of freedom
## Multiple R-squared:  0.9568, Adjusted R-squared:  0.9554
## F-statistic:   682 on 5 and 154 DF,  p-value: < 2.2e-16
```

Figure 49. Residuals of ARDLM 4

```
checkresiduals(modelar4$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 7.4292, df = 10, p-value = 0.6844
```

```
shapiro.test(modelar4$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelar4$model$fitted.values
## W = 0.97657, p-value = 0.008026
```

```
vif(modelar4$model)
```

```
##       gold.price L(gold.price, 1)        oil.price  L(oil.price, 1)
##        61.521659        60.850992        23.842725        24.059680
##  L(asx.price, 1)
##         1.209923
```

```
ardlm.search(formula = asx.price ~ gold.price + oil.price, asx.all, 5, 5)
```

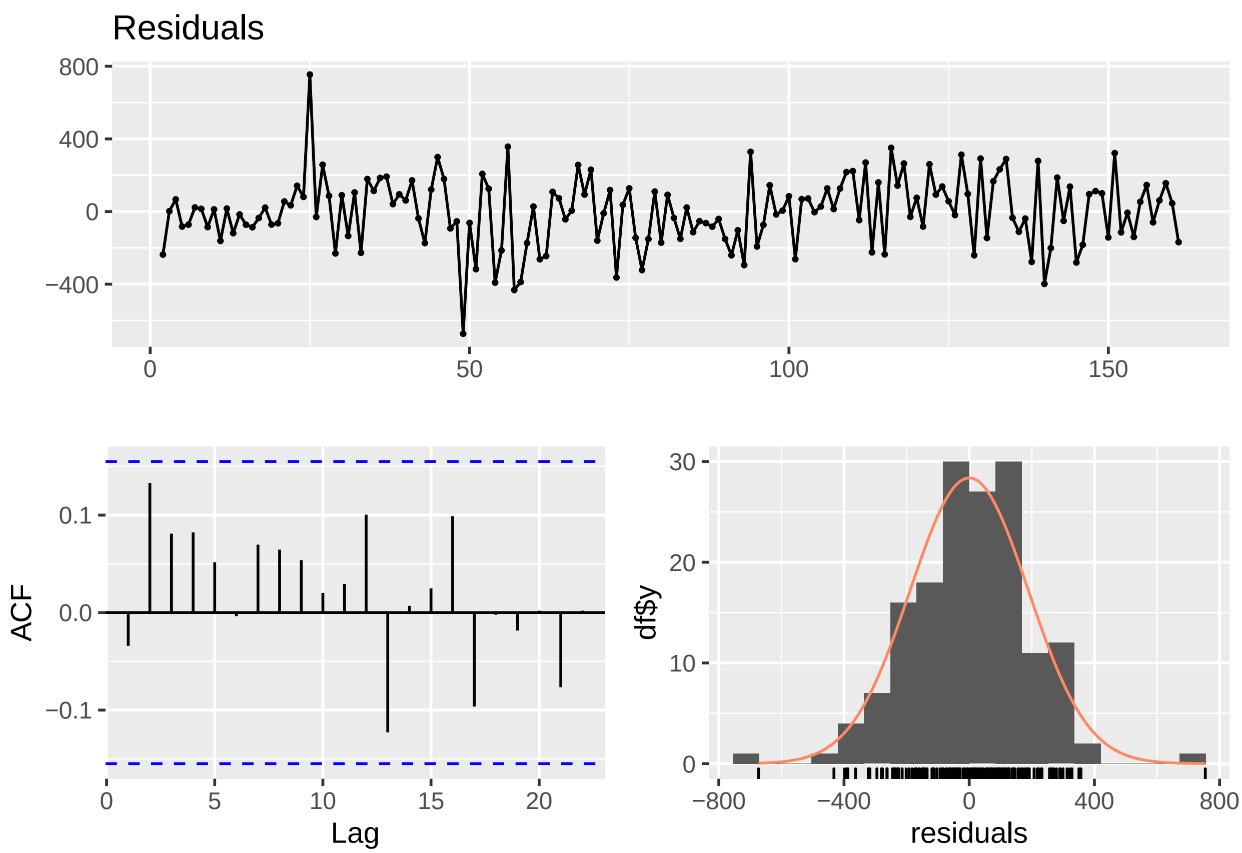
```
## q =  1 q =  1 AIC =  2135.516 BIC =  2157.043
## q =  1 q =  2 AIC =  2122.425 BIC =  2146.977
## q =  1 q =  3 AIC =  2108.476 BIC =  2136.039
## q =  1 q =  4 AIC =  2097.076 BIC =  2127.638
## q =  1 q =  5 AIC =  2086.505 BIC =  2120.054
## q =  2 q =  1 AIC =  2123.279 BIC =  2150.899
## q =  2 q =  2 AIC =  2124.498 BIC =  2155.187
## q =  2 q =  3 AIC =  2111.846 BIC =  2145.534
## q =  2 q =  4 AIC =  2100.502 BIC =  2137.177
## q =  2 q =  5 AIC =  2089.957 BIC =  2129.605
## q =  3 q =  1 AIC =  2114.768 BIC =  2148.457
## q =  3 q =  2 AIC =  2115.917 BIC =  2152.668
## q =  3 q =  3 AIC =  2115.256 BIC =  2155.07
## q =  3 q =  4 AIC =  2103.125 BIC =  2145.913
## q =  3 q =  5 AIC =  2092.688 BIC =  2138.436
## q =  4 q =  1 AIC =  2105.24 BIC =  2144.971
## q =  4 q =  2 AIC =  2106.343 BIC =  2149.13
## q =  4 q =  3 AIC =  2105.506 BIC =  2151.35
## q =  4 q =  4 AIC =  2106.298 BIC =  2155.198
## q =  4 q =  5 AIC =  2095.836 BIC =  2147.683
## q =  5 q =  1 AIC =  2089.622 BIC =  2135.37
## q =  5 q =  2 AIC =  2090.112 BIC =  2138.91
## q =  5 q =  3 AIC =  2089.523 BIC =  2141.371
## q =  5 q =  4 AIC =  2089.979 BIC =  2144.877
## q =  5 q =  5 AIC =  2091.945 BIC =  2149.892
```

```
modelar5 <- ardlDlm(formula = asx.price ~ oil.price + copper.price, data = data.frame(asx.all), q = 1, p
summary(modelar5)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -673.02 -115.41    8.56  118.89  754.25
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    242.96987   88.44334   2.747  0.00673 **
## oil.price.t      3.53327    2.95813   1.194  0.23415
## oil.price.1     -5.89072    2.90745  -2.026  0.04448 *
## copper.price.t   0.07880    0.03849   2.047  0.04234 *
## copper.price.1  -0.04708    0.03951  -1.192  0.23528
## asx.price.1      0.94928    0.02234  42.492  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191.9 on 154 degrees of freedom
## Multiple R-squared:  0.9544, Adjusted R-squared:  0.9529
## F-statistic: 644.2 on 5 and 154 DF,  p-value: < 2.2e-16
```

Figure 50. Residuals of ARDLM 5

```
checkresiduals(modelar5$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 9.7685, df = 10, p-value = 0.461
```

```
shapiro.test(modelar5$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelar5$model$fitted.values
## W = 0.97351, p-value = 0.003606
```

```
vif(modelar5$model)
```

```
##           oil.price    L(oil.price, 1)       copper.price L(copper.price, 1)
##           33.979105          33.119080          28.471320          30.797649
##    L(asx.price, 1)
##            1.720222
```

```r
ardlm.search(formula = asx.price ~ oil.price + copper.price, asx.all, 5, 5)
```

```
## q =  1 q =  1 AIC =  2144.242 BIC =  2165.768
## q =  1 q =  2 AIC =  2132.271 BIC =  2156.823
## q =  1 q =  3 AIC =  2118.394 BIC =  2145.957
## q =  1 q =  4 AIC =  2106.579 BIC =  2137.141
## q =  1 q =  5 AIC =  2095.772 BIC =  2129.32
## q =  2 q =  1 AIC =  2131.019 BIC =  2158.639
## q =  2 q =  2 AIC =  2132.896 BIC =  2163.585
## q =  2 q =  3 AIC =  2120.331 BIC =  2154.019
## q =  2 q =  4 AIC =  2108.36 BIC =  2145.035
## q =  2 q =  5 AIC =  2097.631 BIC =  2137.279
## q =  3 q =  1 AIC =  2120.156 BIC =  2153.845
## q =  3 q =  2 AIC =  2121.813 BIC =  2158.564
## q =  3 q =  3 AIC =  2121.911 BIC =  2161.725
## q =  3 q =  4 AIC =  2109.001 BIC =  2151.788
## q =  3 q =  5 AIC =  2098.153 BIC =  2143.9
## q =  4 q =  1 AIC =  2110.535 BIC =  2150.266
## q =  4 q =  2 AIC =  2111.935 BIC =  2154.722
## q =  4 q =  3 AIC =  2112.421 BIC =  2158.264
## q =  4 q =  4 AIC =  2112.269 BIC =  2161.169
## q =  4 q =  5 AIC =  2101.282 BIC =  2153.13
## q =  5 q =  1 AIC =  2092.53 BIC =  2138.278
## q =  5 q =  2 AIC =  2093.554 BIC =  2142.351
## q =  5 q =  3 AIC =  2094.753 BIC =  2146.601
## q =  5 q =  4 AIC =  2095.032 BIC =  2149.929
## q =  5 q =  5 AIC =  2096.161 BIC =  2154.109
```

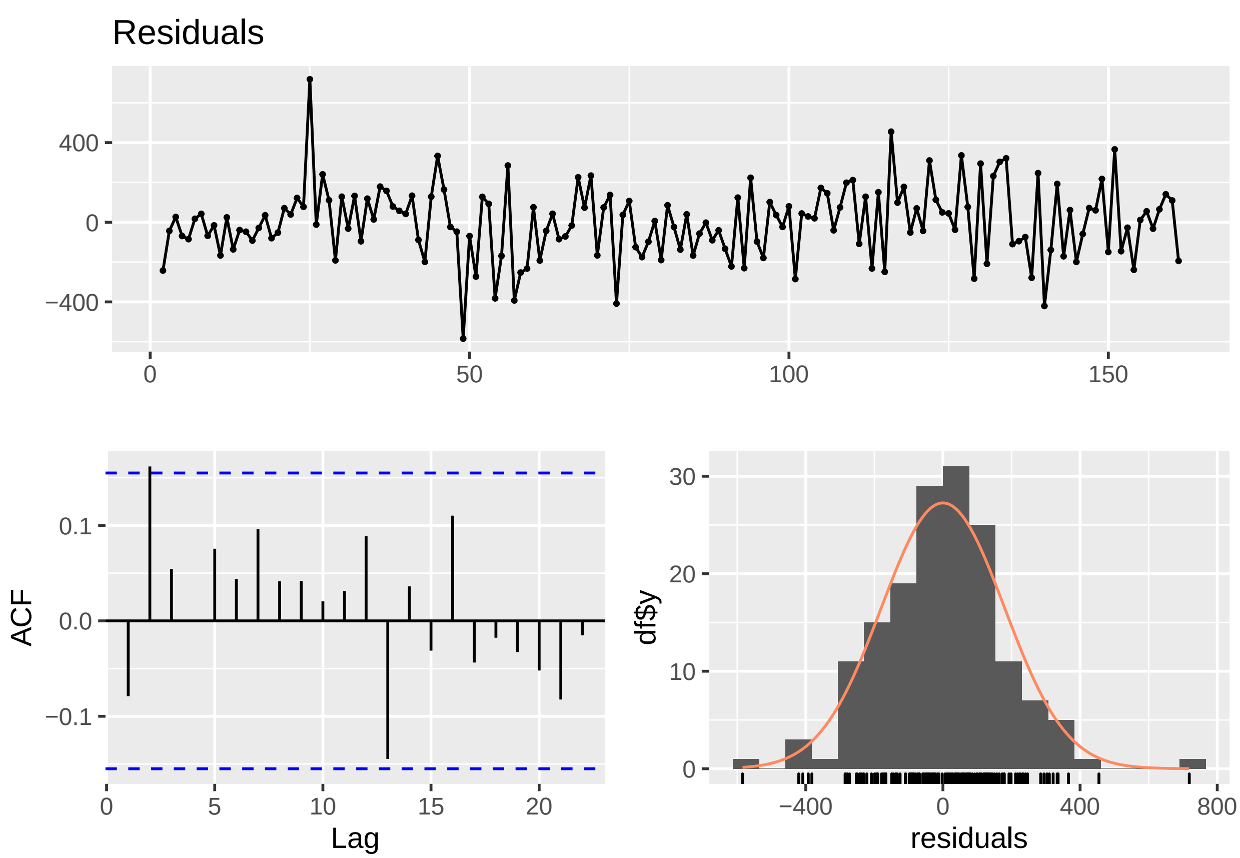
```r
modelar6 <- ardlDlm(formula = asx.price ~ gold.price + oil.price + copper.price, data = data.frame(asx.a
summary(modelar6)
```

```
##
## Time series regression with "ts" data:
## Start = 2, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -584.47 -100.37    8.75  109.63  718.43
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     236.09034   87.89610   2.686  0.00804 **
## gold.price.t     -1.15337    0.28684  -4.021 9.10e-05 ***
## gold.price.1      1.14735    0.28369   4.044 8.32e-05 ***
## oil.price.t       1.51259    2.89071   0.523  0.60155
## oil.price.1      -4.16243    2.82385  -1.474  0.14254
## copper.price.t    0.08140    0.03685   2.209  0.02868 *
## copper.price.1   -0.04539    0.03797  -1.196  0.23371
## asx.price.1       0.95299    0.02139  44.548  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 183.6 on 152 degrees of freedom
## Multiple R-squared:  0.9588, Adjusted R-squared:  0.9569
## F-statistic: 505.5 on 7 and 152 DF,  p-value: < 2.2e-16
```

Figure 51. Residuals of ARDLM 6

```
checkresiduals(modelar6$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 11.069, df = 11, p-value = 0.4375
```

```
shapiro.test(modelar6$model$fitted.values)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelar6$model$fitted.values
## W = 0.97402, p-value = 0.004118
```

```r
vif(modelar6$model)
```

```
##         gold.price   L(gold.price, 1)          oil.price    L(oil.price, 1)
##          62.040350          60.987610          35.478154          34.159543
##       copper.price L(copper.price, 1)    L(asx.price, 1)
##          28.536233          31.091988           1.724689
```

```r
ardlm.search(formula = asx.price ~ gold.price + oil.price + copper.price, asx.all, 5, 5)
```

```
## q =  1 q =  1 AIC =  2131.866 BIC =  2159.542
## q =  1 q =  2 AIC =  2119.288 BIC =  2149.977
## q =  1 q =  3 AIC =  2104.541 BIC =  2138.23
## q =  1 q =  4 AIC =  2093.199 BIC =  2129.874
## q =  1 q =  5 AIC =  2082.841 BIC =  2122.489
## q =  2 q =  1 AIC =  2120.906 BIC =  2157.733
## q =  2 q =  2 AIC =  2121.905 BIC =  2161.801
## q =  2 q =  3 AIC =  2108.425 BIC =  2151.301
## q =  2 q =  4 AIC =  2096.84 BIC =  2142.683
## q =  2 q =  5 AIC =  2086.518 BIC =  2135.316
## q =  3 q =  1 AIC =  2112.595 BIC =  2158.534
## q =  3 q =  2 AIC =  2113.144 BIC =  2162.145
## q =  3 q =  3 AIC =  2111.997 BIC =  2164.061
## q =  3 q =  4 AIC =  2098.84 BIC =  2153.852
## q =  3 q =  5 AIC =  2088.589 BIC =  2146.536
## q =  4 q =  1 AIC =  2103.367 BIC =  2158.379
## q =  4 q =  2 AIC =  2103.338 BIC =  2161.406
## q =  4 q =  3 AIC =  2102.338 BIC =  2163.463
## q =  4 q =  4 AIC =  2102.448 BIC =  2166.629
## q =  4 q =  5 AIC =  2092.135 BIC =  2159.232
## q =  5 q =  1 AIC =  2084.699 BIC =  2148.746
## q =  5 q =  2 AIC =  2082.796 BIC =  2149.892
## q =  5 q =  3 AIC =  2082.928 BIC =  2153.074
## q =  5 q =  4 AIC =  2082.852 BIC =  2156.048
## q =  5 q =  5 AIC =  2084.746 BIC =  2160.992
```

```r
AIC(modelar1$model, modelar2$model, modelar3$model, modelar4$model, modelar5$model, modelar6$model)
```

```
##                df      AIC
## modelar1$model  5 2140.897
## modelar2$model  5 2146.524
## modelar3$model  5 2147.741
## modelar4$model  7 2135.516
## modelar5$model  7 2144.242
## modelar6$model  9 2131.866
```

Overall AIC

```r
AIC <- c(AIC(modeldlm5$model), AIC(modelpoly3$model), AIC(modelk2), AIC(modelar4$model))
```

```
## [1] 2156.172
```

```r
data.frame(AIC, row.names = c("DLM", "PolyDLM", "KoyckDLM", "ARDLM"))
```

```
##               AIC
## DLM      2549.209
## PolyDLM  2574.488
## KoyckDLM 2156.172
## ARDLM    2135.516
```

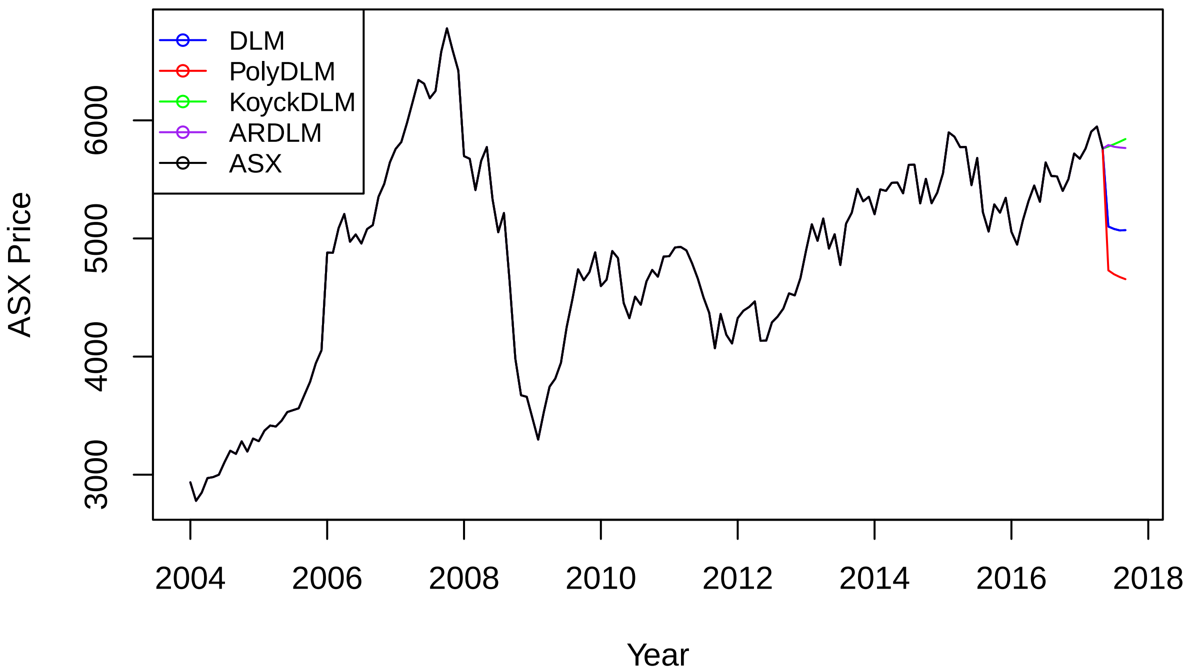
Forecasting

The best model from each type will be used to visualise forecasts. Of the models tested the Koyck DLM model appears to be the best model. It demonstrated a high adjusted R squared, overall significance, significance of coefficients and did not violate the multicollinearity assumption. The second best models were from the ARDLMs, which demonstrated models which could combine two dependent variables to produce a model with high adjusted R squared and significant parameters. However, these models violated the assumption of multicollinearity so the results cannot be trusted in the same way that Koyck models can. Worth noting is that ARDLM received the lowest AIC scores of the 4 model types, with the Koyck model receiving the second lowest.

The forecast in Figure 52. demonstrates the predictions for each model selected. The DLM and PDLM predicted that the ASX would significantly decrease in price and then at a slower rate continue to decrease. The Koyck DLM and ARDLM, which were decided to be the best two models, demonstrate predictions closer to where the ASX series finishes. The Koyck DLM predicts a steady increase of the ASX price over the next 4 month. This is slightly different to the ARDLM, which predicts an slight increase then a continued slight decrease in the ASX price.

```r
forecast.data <- matrix(c(1675, 1700, 1725, 1750, 50.87, 48, 46, 44, 5599, 5420, 5340, 5260), nrow = 3,
model1Frc <- dLagM::forecast(modeldlm5, x = forecast.data[2:3,], h = 4)
model2Frc <- dLagM::forecast(modelpoly3, x = forecast.data[3,], h = 4)
model3Frc <- dLagM::forecast(modelk2, x = forecast.data[2,], h = 4)
model4Frc <- dLagM::forecast(modelar4, x = forecast.data[1:2,], h = 4)


plot(ts(c(as.vector(asx), model1Frc$forecasts), start = c(2004, 1), frequency = 12), col = "blue", type
forecasts <- list(model1Frc$forecasts, model2Frc$forecasts, model3Frc$forecasts, model4Frc$forecasts, r
for (i in 1:5){
  colour <- c("blue", "red", "green", "purple", "black")
  lines(ts(c(as.vector(asx), forecasts[[i]]), start = c(2004, 1), frequency = 12), col = colour[i], typ
}
legend("topleft", lty = 1, pch = 1, text.width = 1.8, cex = .8, col = c("blue", "red", "green", "purple
```

Figure 52. Forecast of ASX Price




```
forecasts <- data.frame(model1Frc$forecasts, model2Frc$forecasts, model3Frc$forecasts, model4Frc$foreca
forecasts
```

```
##   model1Frc.forecasts model2Frc.forecasts model3Frc.forecasts
## 1            5100.842            4729.830            5778.473
## 2            5080.506            4696.766            5798.091
## 3            5067.917            4673.711            5819.224
## 4            5070.368            4655.235            5841.834
##   model4Frc.forecasts
## 1            5791.165
## 2            5776.263
## 3            5769.826
## 4            5765.832
```

Summary

The aim of this investigation was to examine the ASX price data from January 2004 to May 2017 with similar data to use for forecasting. It was split into two parts with the first being the examination of the dataset and the second the model fitting and forecasting.

For the first part, the examination analysed stationarity, changing variance and seasonality predominantly. All 4 time series were concluded to be nonstationary, although a single differecing was sufficient to convert the series into stationary series. In terms of changing variance, the oil and copper series benefitted the most from either log or Boxcox transformations to reduce its effect. Finally, decomposition of trend and seasonality was undertaken for each series. ASX and oil series appeared to have seasonal components based on the seasonal factors. Therefore, the forecasts based on the decompositions for these 2 series used seasonal adjusting, with gold and copper not needing the adjustment.

For the second part, 4 types of models were fit to the ASX data with different combinations of the possible independent variables. The first model, the distributed lag model (DLM), appeared to struggle with the data. It had relatively explanatory power and violated the multicollinearity assumption of the model. To fix this, the polynomial distributed lag model (PDLM) was attempted. This model had similar success to the DLM models and did not fix the violation of the assumption. Koyck distributed lag models were then attempted, with greater success than the 2 previous models. It demonstrated a strong ability to explain variance, with the model being overall significant as well as its coefficients. The Koyck model also demonstrated an improvement in its residuals, with them not demonstrating serial correlation. Importantly, this model also satisfied the assumption of multicollinearity. Finally, the autoregressive distributed lag model (ARDLM) was implimented. This model demonstrated similar improvements over the DLMs and PDLMs that Koyck models showed. However, the ARDLMs failed to satisfy the assumption of multicollinearity. The best of each model type were then evaluated with the AIC. The ARDLM model ranked as the best in this, with the Koyck model ranking as second. However, since the ARDLM model failed to satisfy the multicollinearity assumption the Koyck model was selected as the best model.

Four month forecasts of ASX price were run for the best of each model. The DLM and PDLM showed forecasts that dramatically deviated from the last point in the time series. The Koyck model and ARDLM demonstrated predicted closer to the final point of the time series. The main difference between the two was the increasing trend of the Koyck prediction and the decreasing trend of the ARDLM.

Conclusion

Based on the investigation and analyses conducted, the ASX price is predicted to increase over the next four months if oil prices continue in their trend. Although an alternate model suggests the ASX to decrease, the selected model was evaluated to be the best of the models available and should be the model used.