Forecasting Assignment 1 - Phil Steinke

our analysis must explore and elaborate on:

Description

- ☑ The existence of **non-stationarity** in dataset,
- ☑ The impact of the components of a time series data on the given dataset
- ☑ The most accurate and suitable distributed lag model for the ASX price index.

Dataset

- The dataset includes monthly averages of ASX All Ordinaries (Ords) Price Index, Gold price (AUD), Crude Oil (Brent, USD/bbl) and copper (USD/tonne) between January 2004 and May 2017. .
- All Ordinaries (Ords) Price Index
- Gold price (AUD)
- Crude Oil (Brent, USD/bbl) and copper (USD/tonne)
- When: Between January 2004 and May 2017.

Goals:

- the existence of non-stationarity in dataset
- the impact of the components of a time series data on the given dataset
- the most accurate and suitable distributed lag model for the ASX price index

Create timeSeries objects

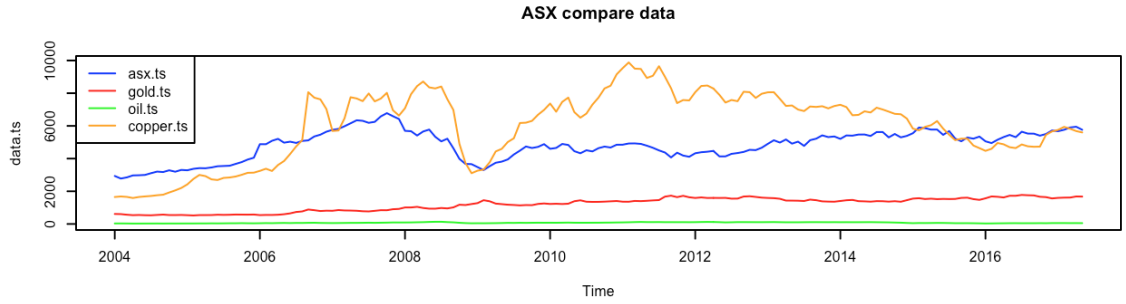
```
# `data.ts` - a time-series dataframe with all columns
data.ts <- convertToTimeseries(data, tsStart)

# Create a timeSeries object for each feature:
# LATER: convertMultivariateTimeseriesObjects(data, tsStart)

asx.ts <- convertToTimeseries(data, tsStart, colName='asx')
gold.ts <- convertToTimeseries(data, tsStart, colName='gold')
oil.ts <- convertToTimeseries(data, tsStart, colName='oil')
copper.ts <- convertToTimeseries(data, tsStart, colName='copper')

asx.ts %>% head(12)
```

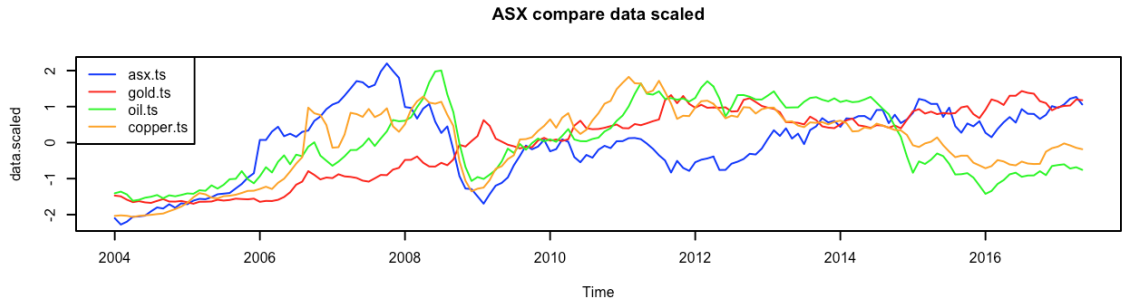
```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2004 2935.4 2778.4 2848.6 2970.9 2979.8 2999.7 3106.7 3202.9 3176.2 3282.4
##         Nov    Dec
## 2004 3195.7 3306.0
```

Plot individual features

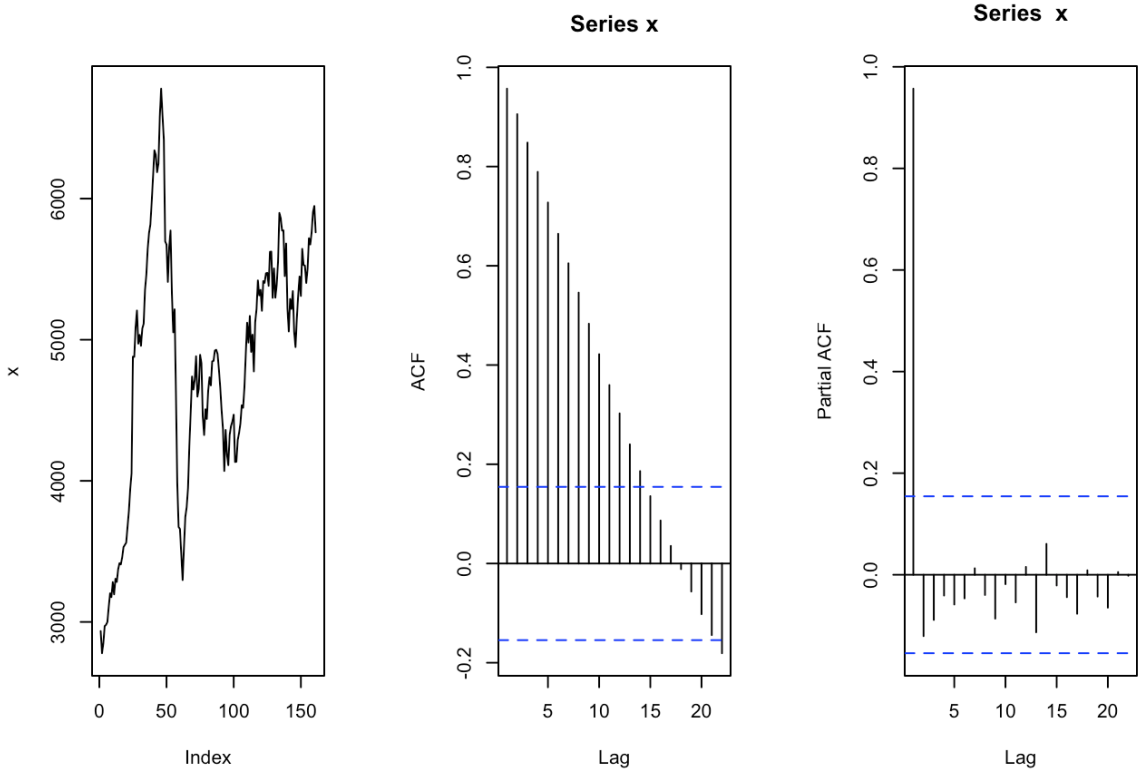
- [x] Display and infer time series, ACF and PACF plots (pt1)

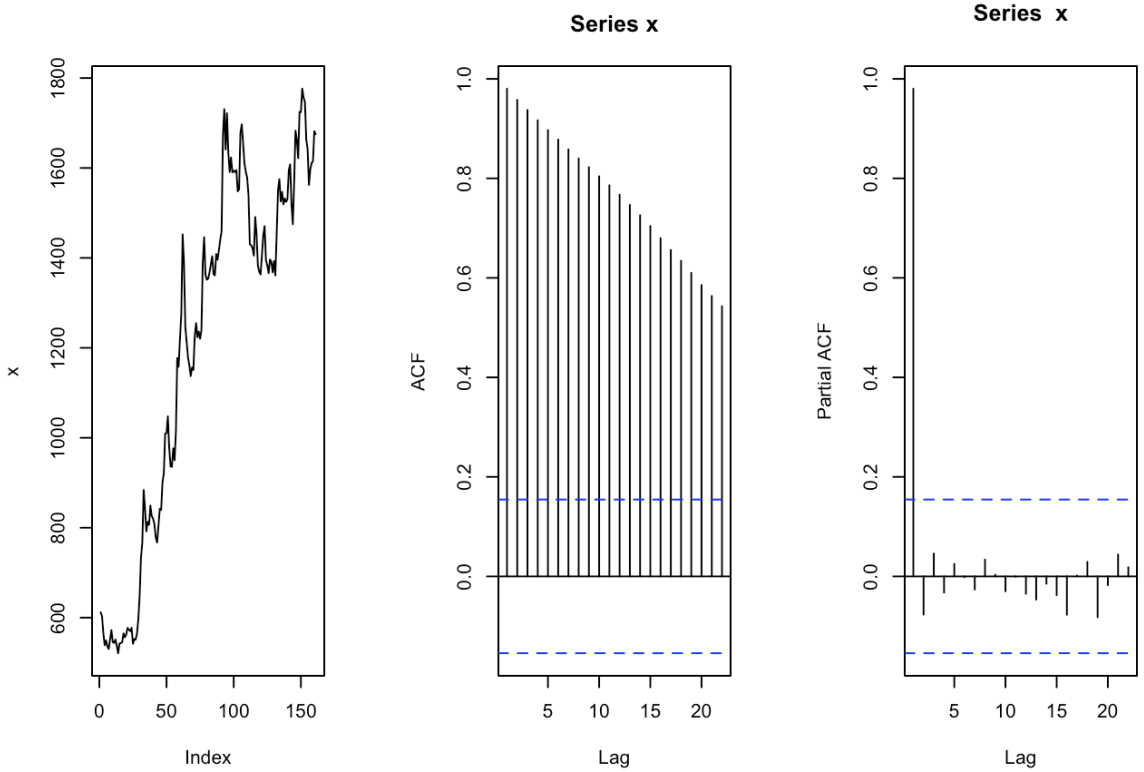
```
plotLayout(rows=2, cols=1)
colourGraphs <- c("blue", "red", "green", "orange")

plot(data.ts, plot.type="s",col = colourGraphs, main = "ASX compare data")
legend("topleft", lty=1, col=colourGraphs, legend = featuresData)

# Scale and centre each series
data.scaled <- scale(data.ts)
plot(data.scaled, plot.type="s", col = colourGraphs, main = "ASX compare data
scaled")
legend("topleft", lty=1, col= colourGraphs, legend = featuresData)
```

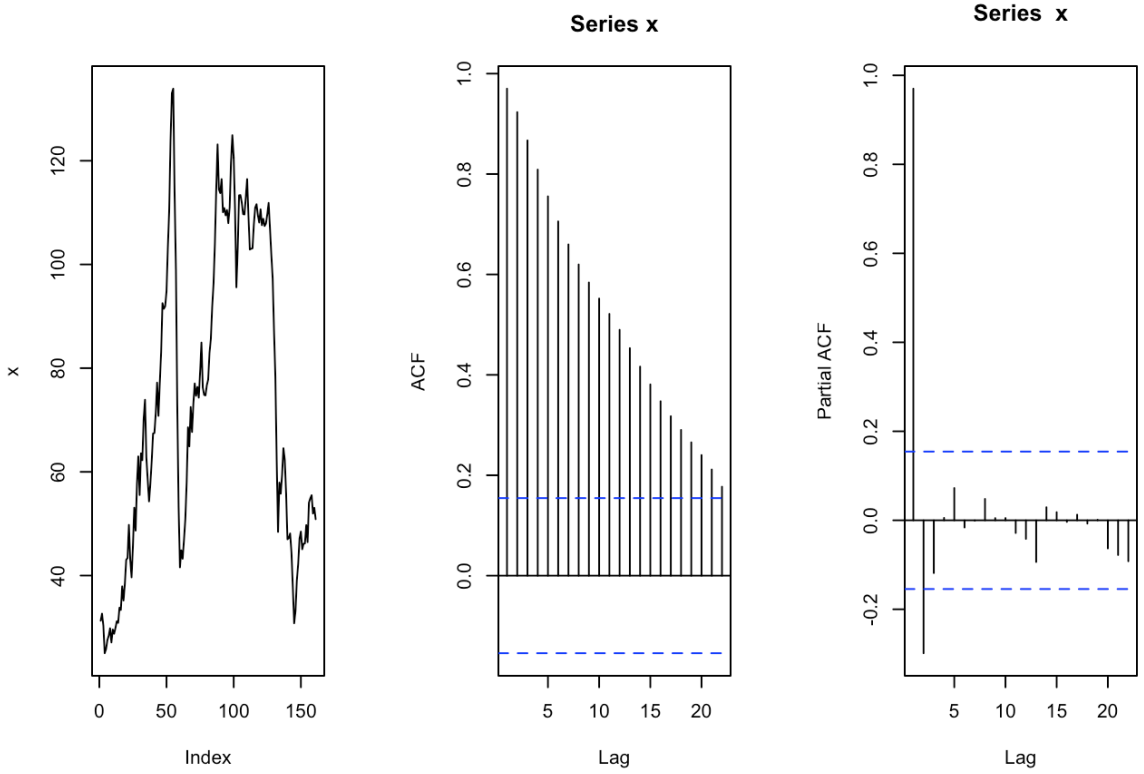


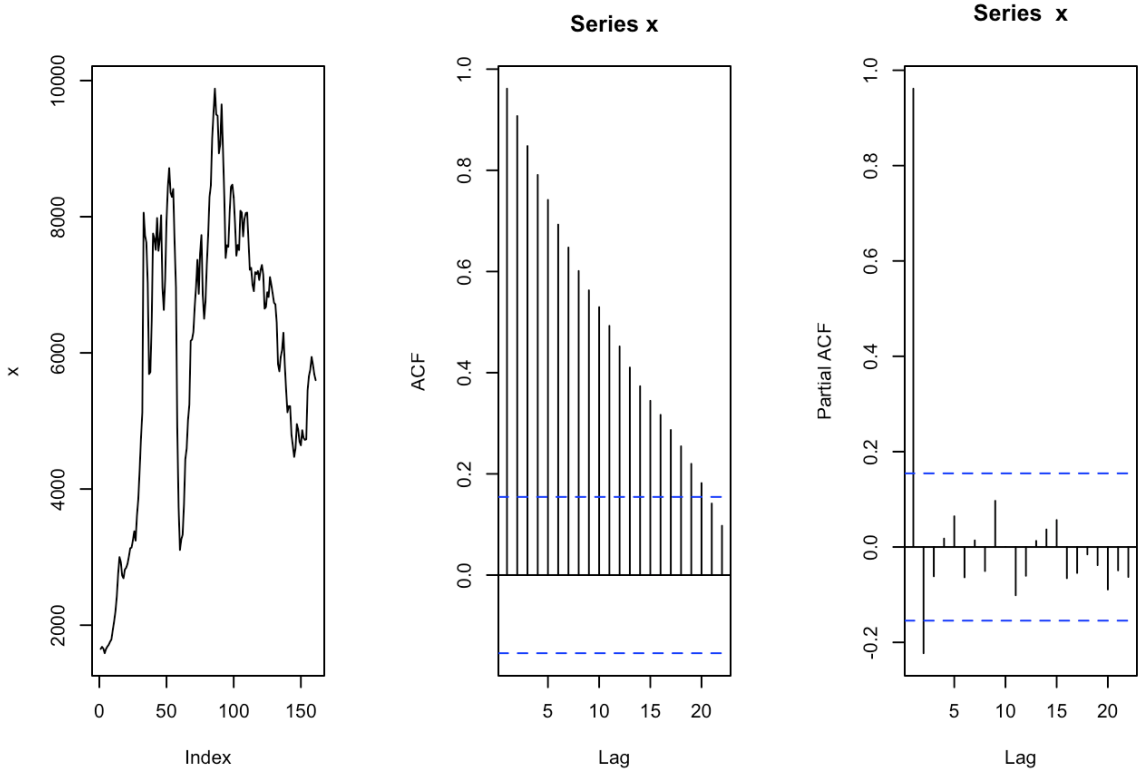



```
apply(data.ts, 2, doPlot)
```


x
Index
Series x
ACF
Lag
Series x
Partial ACF
Lag


x
Index
Series x
ACF
Lag
Series x
Partial ACF
Lag


x
Index
Series x
ACF
Lag
Series x
Partial ACF
Lag


x
Index
Series x
ACF
Lag
Series x
Partial ACF
Lag

```
## $asx
##
## ################################
## # 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-alpha  is: -7.4616
##
##          aux. Z statistics
## Z-tau-mu            2.3366
##
##
## $gold
##
## ################################
## # 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-alpha  is: -1.8005
##
##          aux. Z statistics
## Z-tau-mu            1.5151
##
##
## $oil
##
## ###################################
## # 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-alpha  is: -7.2988
##
##          aux. Z statistics
## Z-tau-mu            1.9323
##
##
## $copper
##
```

```
## ##################################
## # 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-alpha  is: -8.2854
##
##          aux. Z statistics
## Z-tau-mu             2.406
```

Check correlation of each of the features

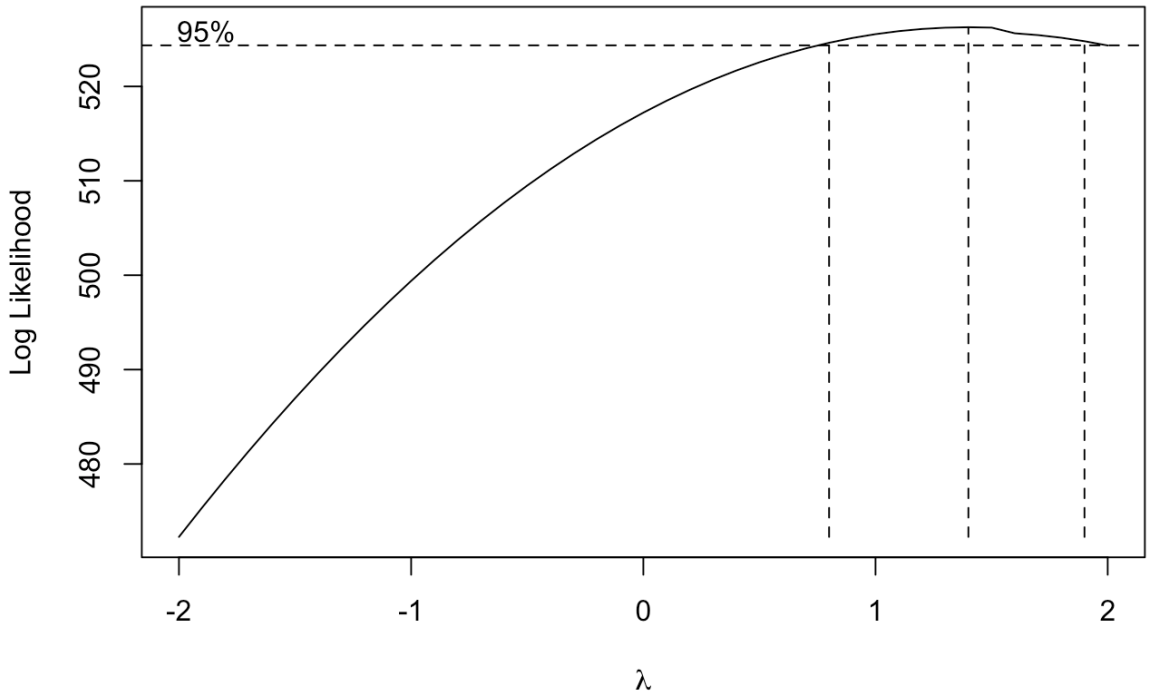
```
cor(data.ts)
```

```
##              asx      gold       oil    copper
## asx    1.0000000 0.3431908 0.3290338 0.5617864
## gold   0.3431908 1.0000000 0.4366382 0.5364213
## oil    0.3290338 0.4366382 1.0000000 0.8664296
## copper 0.5617864 0.5364213 0.8664296 1.0000000
```

- High positive correlation between:
  - `copper` and `oil` of `0.87`
  - `copper` and `asx` of `0.56`
  - `copper` and `gold` of `0.54`
- **Therefore:** features are not independent
- `copper` USD correlates with most other variables, so we can *test* a model for `data.ts` that's primarily based on `copper.ts`

Apply a Suitable Transformation

```
confidenceInterval(asx.ts)
```




```
## [1] 0.8 1.9
```

- No confidence intervals do not capture `0`, so we'll try a `BoxCox` transformation
- The Box-Cox transformation **did not** help to improve the normality of the series because:
- the dots are still not aligned with the red line in the QQ plot
- The Shapiro test p-value < 0.05
- Actually makes data less normal
- Therefore we discard the Box-Cox transformation

For other features

- For brevity; Q-Q plots are not included for other features

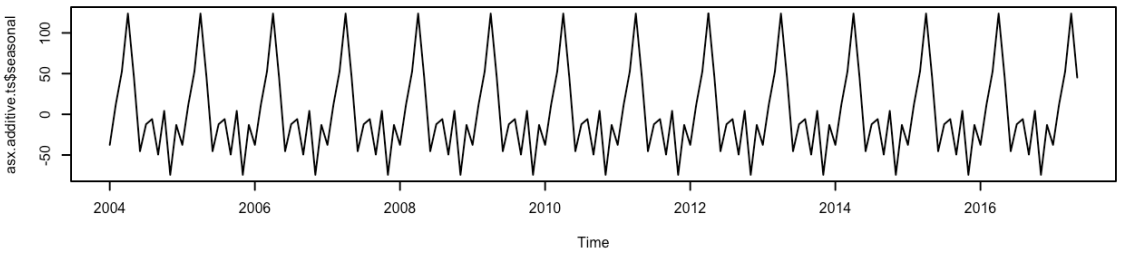
```
# cat('Gold: ', shapiro.test(gold.ts)$p, 'BoxCox: ', shapiro.test(gold.boxcox.
ts)$p, '\n')
# cat('Crude Oil: ', shapiro.test(oil.ts)$p, 'BoxCox: ', shapiro.test(oil.boxc
ox.ts)$p, '\n')
# cat('Copper: ', shapiro.test(copper.ts)$p, 'BoxCox: ', shapiro.test(copper.b
oxcox.ts)$p, '\n')
```

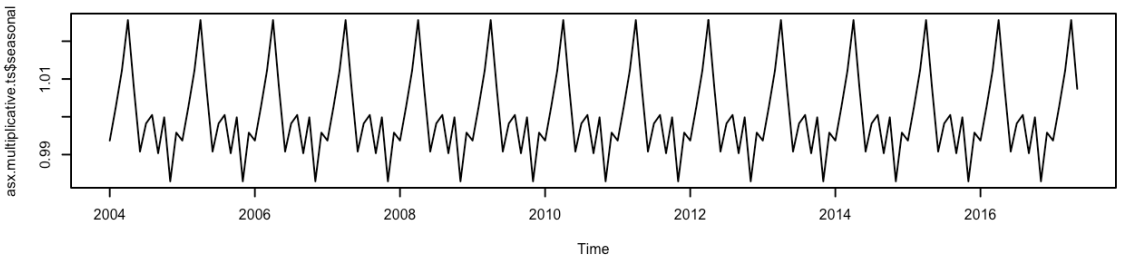
- For other features, the Box-Cox transformation **did not** help to improve the normality of the series because the Shapiro test p-value < 0.05

Check if there is existing seasonality

- Should we diff with or without seasonality?

```
plotLayout(rows=2, cols=1)
# additive seasonality:
asx.additive.ts <- decompose(asx.ts, type="additive")
plot(asx.additive.ts$seasonal) # April - max every, min every June
# asx.additive.ts$seasonal %>% head(24)
# multiplicative seasonality:
asx.multiplicative.ts <- decompose(
  asx.ts,
  type="multiplicative",
)
plot(asx.multiplicative.ts$seasonal)
```






```
cat('isSeasonal: ', isSeasonal(asx.ts), '\n')
```

```
## isSeasonal:  FALSE
```

```
cat('Calculated frequency: ', findSeasonalFreq(asx.ts), '\n')
```

```
## Calculated frequency:  1
```

- From decomposition, it appears there is seasonality every 12 months
  - However, the `isSeasonal` and `findSeasonalFreq` don't find the seasonality
- So, we will attempt to diff both with and without seasonality

The existence of **Non-Stationary** in the dataset

- [x] The existence of **Non-Stationary** in the dataset
- [x] Display and infer time series, ACF and PACF plots (pt1)

H0:μΔ=0 HA:μΔ≠0

```
adf.test(asx.ts)$p.value %>% round(2)
```

```
## [1] 0.28
```

```
adf.test(gold.ts)$p.value %>% round(2)
```

```
## [1] 0.64
```

```
adf.test(oil.ts)$p.value %>% round(2)
```

```
## [1] 0.64
```

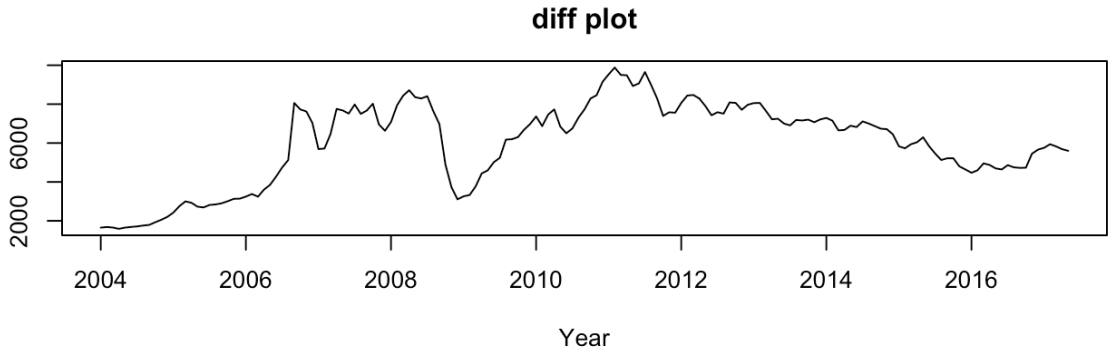
```
adf.test(copper.ts)$p.value %>% round(2)
```

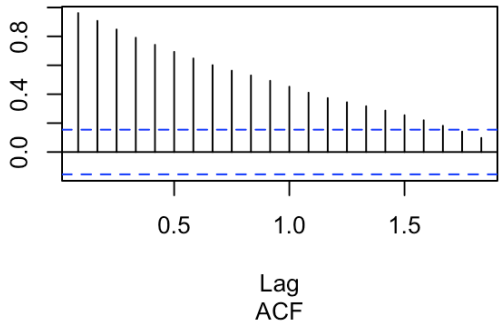
```
## [1] 0.47
```

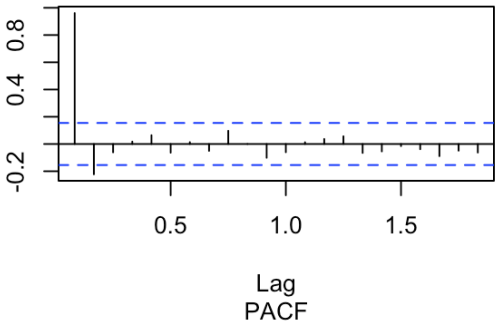
- With p-value's for all data of `0.28, 0.64, 0.64, 0.47`, we cannot reject the null hypothesis stating that the series is non-stationar
- The stochastic component of `data.ts` is not normally distributed
- We found that as the p-value was less than 5% and the null mean (i.e. zero) fell within the 95% confidence interval. As such the null hypothesis was rejected.

```
# LATER: just show ACF and PACF plots
doDiffAndPlot(copper.ts, 0, T, T)
```

```
## diff:  0
##
##
## adf p-value:
## > 0.05 insignificant
##  0.48777673286537
```








```
doDiffAndPlot(gold.ts, 0, F, F)
```

```
## diff:  0
##
##
## adf p-value:
## > 0.05 insignificant
##  0.909223920426874
```

```
doDiffAndPlot(oil.ts, 0, F, F)
```

```
## diff:  0
##
##
## adf p-value:
## > 0.05 insignificant
##  0.419617550809062
```

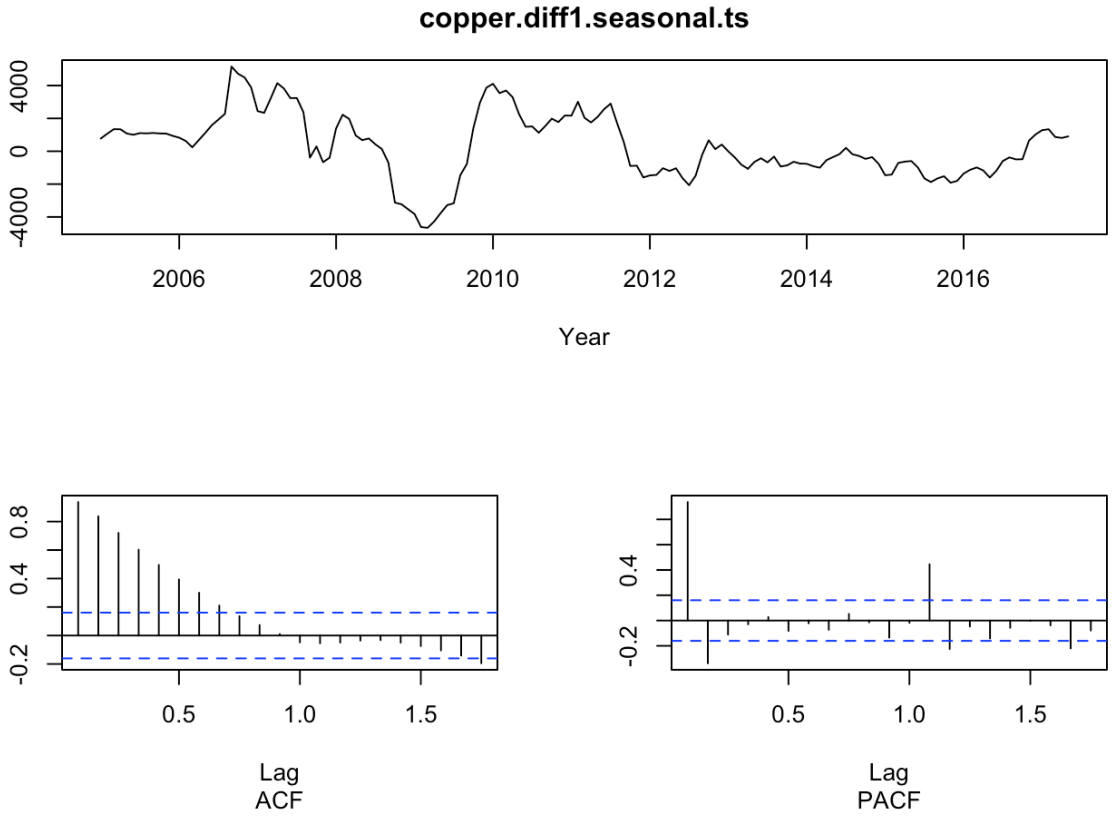
```
doDiffAndPlot(asx.ts, 0, F, F)
```

```
## diff:  0
##
##
## adf p-value:
## > 0.05 insignificant
##  0.758781510251436
```

Find best transformation:

```
### Diff #1 with ACF PACF plots
copper.diff1.seasonal.ts <- doDiffAndPlot(copper.ts, 1, T, lag=12, out=TRUE, p
lotTitle='copper.diff1.seasonal.ts')
```

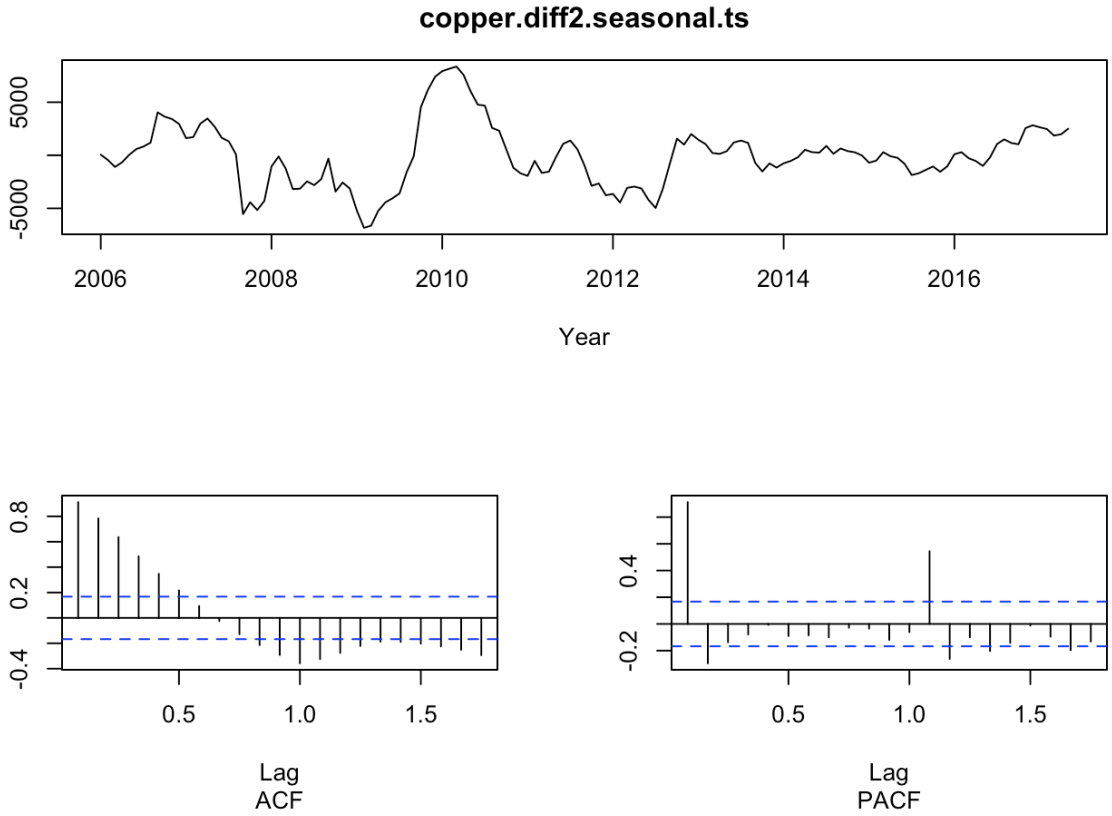
```
## diff:  1
##
##
## adf p-value:
## 0.0248276088311791 < 0.05 significant
```



- ACF shows decay

```
copper.diff2.seasonal.ts <- doDiffAndPlot(copper.ts, 2, T, lag=12, out = TRUE,
plotTitle='copper.diff2.seasonal.ts')
```

```
## diff:  2
##
##
## adf p-value:
## 0.01 < 0.05 significant
```



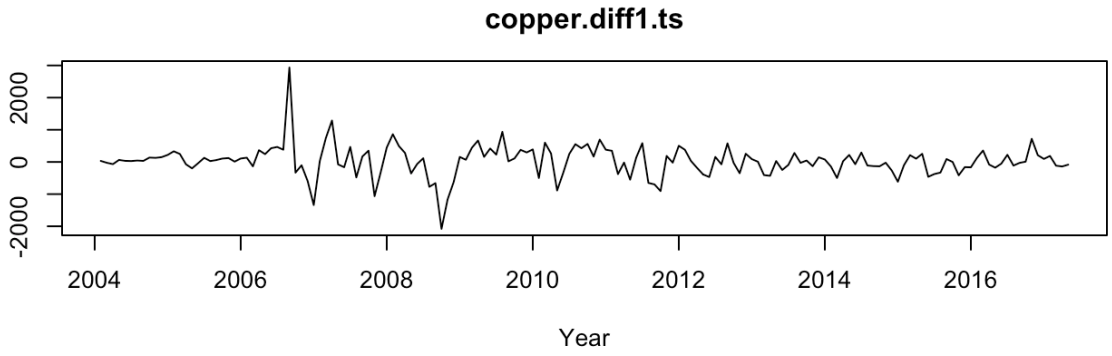
Further diff doesn't remove decay in ACF

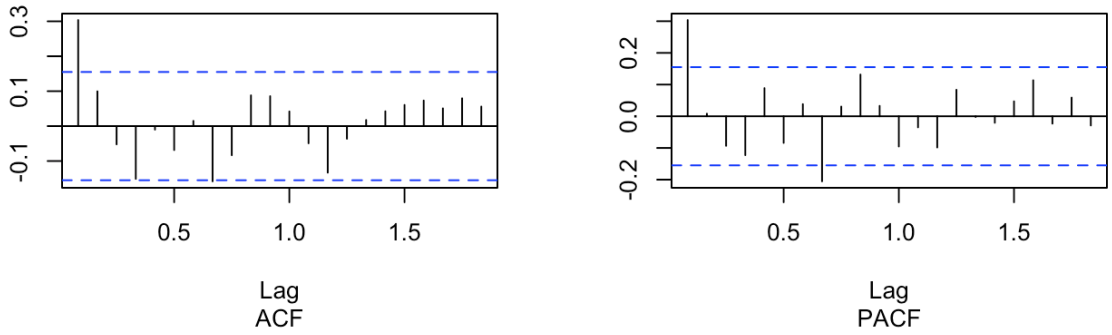
```
# copper.boxcox.diff1.seasonal.ts <- doDiffAndPlot(copper.boxcox.ts, 1, lag=1
2, out=TRUE, plotTitle='copper.boxcox.diff1.seasonal.ts')
```

This is also true when we consider the BoxCox Transform

```
copper.diff1.ts <- doDiffAndPlot(copper.ts, 1, T, F, out=TRUE, plotTitle='copp
er.diff1.ts')
```

```
## diff:  1
##
##
## adf p-value:
## 0.01 < 0.05 significant
```





- Not including the seasonal diff gives the best output
- First diff is stationary
- ACF and PACF plots show normality
- So, we will choose `diff = 1, no seasonal differencing, no boxcox transform`

Create dataframe for each feature with `diff = 1`

```
asx.diff1.ts <- doDiffAndPlot(asx.ts, 1, F, F, out=TRUE)
```

```
## diff:  1
```

```
## Warning in adfTest(df.afterDiff.ts, lags = order, title = NULL, description
## = NULL): p-value smaller than printed p-value
```

```
##
## adf p-value:
## 0.01 < 0.05 significant
```

```
gold.diff1.ts <- doDiffAndPlot(gold.ts, 1, F, F, out=TRUE)
```

```
## diff:  1
```

```
## Warning in adfTest(df.afterDiff.ts, lags = order, title = NULL, description
## = NULL): p-value smaller than printed p-value
```

```
##
## adf p-value:
## 0.01 < 0.05 significant
```

```
oil.diff1.ts <- doDiffAndPlot(oil.ts, 1, F, F, out=TRUE)
```

```
## diff:  1
```

```
## Warning in adfTest(df.afterDiff.ts, lags = order, title = NULL, description
## = NULL): p-value smaller than printed p-value
```

```
##
## adf p-value:
## 0.01 < 0.05 significant
```

- First difference is significant for all features - Augmented Dickey Fuller `p < 0.5`

Phillips-Perron Unit Root Test

```
PP.test(copper.ts, lshort = TRUE)$p.value %>% round(2)
```

```
## [1] 0.55
```

```
PP.test(copper.ts, lshort = FALSE)$p.value %>% round(2)
```

```
## [1] 0.62
```

```
cat('\ndiff = 1: ')
```

```
##
## diff = 1:
```

```
PP.test(copper.diff1.ts, lshort = TRUE)$p.value %>% round(2)
```

```
## [1] 0.01
```

```
PP.test(copper.diff1.ts, lshort = FALSE)$p.value %>% round(2)
```

```
## [1] 0.01
```

According to Phillips-Perron Unit Root Test:

- data.ts is not stationary `p(0.55, 0.62) > 0.05`
- data with diff=1 is stationary `p(.01) < 0.05`

Model

```
model.asxVsCopperDiff1 = dlm(
  x = as.vector(asx.diff1.ts),
  y = as.vector(copper.diff1.ts),
  q = 4
)
summarySummary(model.asxVsCopperDiff1)
```

```
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1487.56  -229.60    12.74   204.94  2901.11
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.2709    38.0810   0.138  0.89010
## x.t           0.6034     0.1867   3.231  0.00151 **
## x.1           0.3863     0.1878   2.057  0.04138 *
## x.2          -0.1749     0.1885  -0.928  0.35502
## x.3           0.0600     0.1877   0.320  0.74972
## x.4           0.2384     0.1870   1.275  0.20434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 468.6 on 150 degrees of freedom
## Multiple R-squared:  0.1113, Adjusted R-squared:  0.0817
## F-statistic: 3.758 on 5 and 150 DF,  p-value: 0.003103
##
## AIC and BIC values for the model:
##       AIC      BIC
## 1 2369.33 2390.679
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1487.56  -229.60    12.74   204.94  2901.11
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.2709    38.0810   0.138  0.89010
## x.t           0.6034     0.1867   3.231  0.00151 **
## x.1           0.3863     0.1878   2.057  0.04138 *
## x.2          -0.1749     0.1885  -0.928  0.35502
## x.3           0.0600     0.1877   0.320  0.74972
## x.4           0.2384     0.1870   1.275  0.20434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 468.6 on 150 degrees of freedom
## Multiple R-squared:  0.1113, Adjusted R-squared:  0.0817
## F-statistic: 3.758 on 5 and 150 DF,  p-value: 0.003103
##
## AIC and BIC values for the model:
##       AIC      BIC
## 1 2369.33 2390.679
```

```
# checkresiduals(model.asxVsCopperDiff1)
```

- `Adjusted R-squared: 0.0817` (residuals skipped)
- `AIC: 2369.33`

Linear Model

```
model.lm <- lm(asx ~ gold, oil, copper, data=data.ts)
summarySummary(model.lm)
```

```
## adjusted r-squared: 0.37
## p-value: 8.198425e-18
```

```
# checkresiduals(model.lm$residuals) #
```

- residuals skipped because of low `adjusted r-squared: 0.37`

H_0: $y_t \neq \beta_0 + \beta_1 x_t + \varepsilon_t$ H_a: $y_t = \beta_0 + \beta_1 x_t + \varepsilon_t$

Dynamic LM - `dynlm`

```
model.dynlm = dynlm(
  asx ~ gold + oil + copper,
  data=data.ts)
summarySummary(model.dynlm)
```

```
## adjusted r-squared: 0.41
## p-value: 2.934075e-18
```

- Residuals skipped because of low `Adjusted r-squared: 0.41`
- p-value is significant `< 0.05`

```
model.dlm = dlm(x = as.vector(asx.ts) , y = as.vector(copper.ts), q = 8)
summarySummary(model.dlm)
```

```
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2796.3 -1625.3    -4.4  1222.8  3826.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 499.29609  919.70613   0.543    0.588
## x.t           0.34271    0.71768   0.478    0.634
## x.1           0.31190    1.02537   0.304    0.761
## x.2          -0.16873    1.02759  -0.164    0.870
## x.3           0.10334    1.02838   0.100    0.920
## x.4           0.26490    1.02870   0.258    0.797
## x.5           0.09931    1.02881   0.097    0.923
## x.6          -0.09981    1.03089  -0.097    0.923
## x.7           0.08831    1.02924   0.086    0.932
## x.8           0.23596    0.71124   0.332    0.741
##
## Residual standard error: 1758 on 143 degrees of freedom
## Multiple R-squared:  0.2249, Adjusted R-squared:  0.1762
## F-statistic: 4.611 on 9 and 143 DF,  p-value: 2.322e-05
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2732.224 2765.559
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2796.3 -1625.3    -4.4  1222.8  3826.3
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 499.29609  919.70613   0.543    0.588
## x.t           0.34271    0.71768   0.478    0.634
## x.1           0.31190    1.02537   0.304    0.761
## x.2          -0.16873    1.02759  -0.164    0.870
## x.3           0.10334    1.02838   0.100    0.920
## x.4           0.26490    1.02870   0.258    0.797
## x.5           0.09931    1.02881   0.097    0.923
## x.6          -0.09981    1.03089  -0.097    0.923
## x.7           0.08831    1.02924   0.086    0.932
## x.8           0.23596    0.71124   0.332    0.741
##
## Residual standard error: 1758 on 143 degrees of freedom
## Multiple R-squared:  0.2249, Adjusted R-squared:  0.1762
## F-statistic: 4.611 on 9 and 143 DF,  p-value: 2.322e-05
```

```
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2732.224 2765.559
```

- Since copper has a high correlation with most variables, I've looked at a dlm between asx and copper. A good fit between these would ideally correlate well with other factors.
- However, with dlm, the adjusted r-squared is only `0.17`

Model our data vs model

```
# AIC
model.finiteDlmauto <- finiteDLMauto(
  x = as.vector(asx.ts),
  y = as.vector(copper.ts),
)
summarySummary(model.finiteDlmauto)
```

```
##      q - k         MASE            AIC            BIC
##  Min.   :10   Min.   :4.363   Min.   :2699   Min.   :2738
##  1st Qu.:10   1st Qu.:4.363   1st Qu.:2699   1st Qu.:2738
##  Median :10   Median :4.363   Median :2699   Median :2738
##  Mean   :10   Mean   :4.363   Mean   :2699   Mean   :2738
##  3rd Qu.:10   3rd Qu.:4.363   3rd Qu.:2699   3rd Qu.:2738
##  Max.   :10   Max.   :4.363   Max.   :2699   Max.   :2738
##     R.Adj.Sq        Ljung-Box
##  Min.   :0.1266   Min.   :0
##  1st Qu.:0.1266   1st Qu.:0
##  Median :0.1266   Median :0
##  Mean   :0.1266   Mean   :0
##  3rd Qu.:0.1266   3rd Qu.:0
##  Max.   :0.1266   Max.   :0
```

- Only provides an adjusted R squared of 0.12, much lower than previous models

```
model.polyDlm = polyDlm(
  x = as.vector(asx.ts),
  y = as.vector(copper.ts),
  q = 2,
  k = 2,
  show.beta = TRUE)
```

```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    0.189      0.703   0.268   0.789
## beta.1    0.361      1.020   0.355   0.723
## beta.2    0.766      0.695   1.100   0.272
```

```
summarySummary(model.polyDlm)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -2931  -1509    -75   1277   3774
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -283.93754  790.71444  -0.359    0.720
## z.t0           0.18852    0.70292   0.268    0.789
## z.t1           0.05628    3.10110   0.018    0.986
## z.t2           0.11617    1.53158   0.076    0.940
##
## Residual standard error: 1757 on 155 degrees of freedom
## Multiple R-squared:  0.3058, Adjusted R-squared:  0.2924
## F-statistic: 22.76 on 3 and 155 DF,  p-value: 2.908e-12
##
## adjusted r-squared: 0.29
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -2931  -1509    -75   1277   3774
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -283.93754  790.71444  -0.359    0.720
## z.t0           0.18852    0.70292   0.268    0.789
## z.t1           0.05628    3.10110   0.018    0.986
## z.t2           0.11617    1.53158   0.076    0.940
##
## Residual standard error: 1757 on 155 degrees of freedom
## Multiple R-squared:  0.3058, Adjusted R-squared:  0.2924
## F-statistic: 22.76 on 3 and 155 DF,  p-value: 2.908e-12
##
## p-value: 2.908403e-12
```

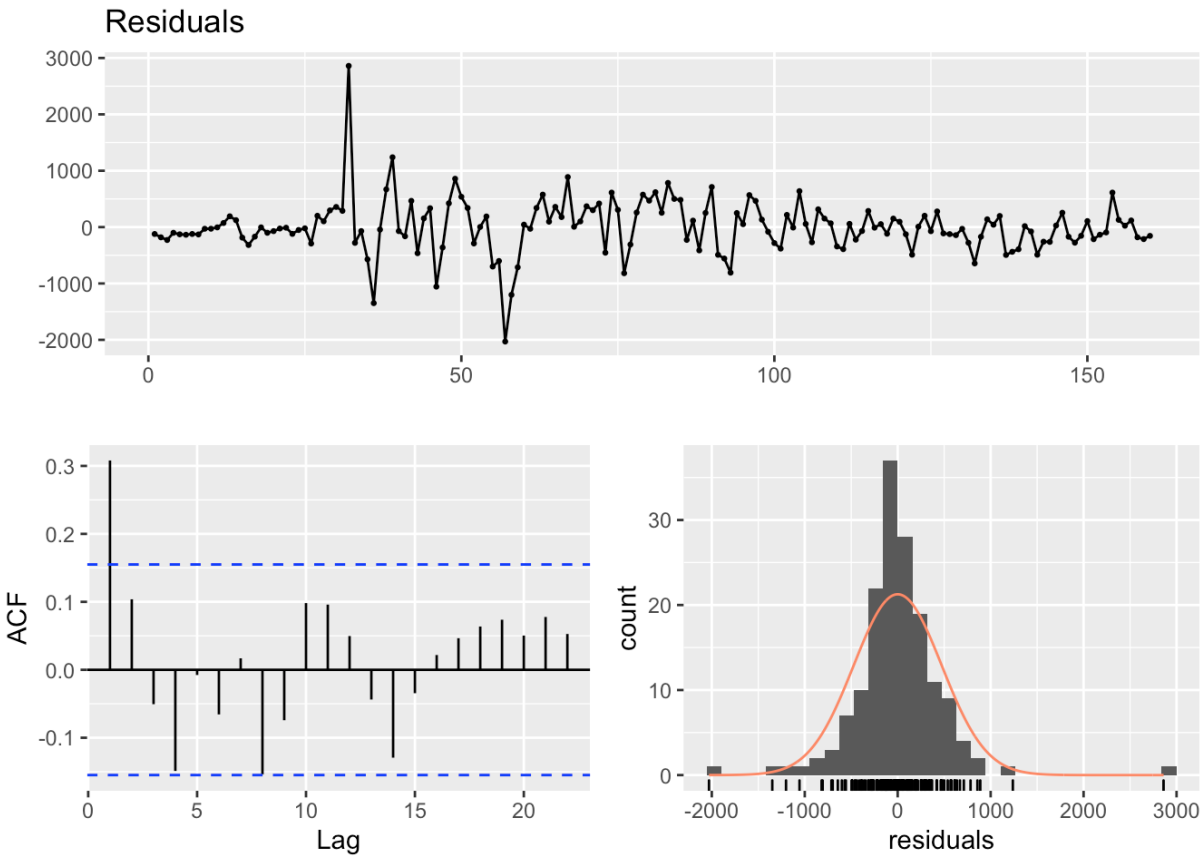
- Rejected: because this only has an adjusted r-squared of `0.2924`

Koyck

```
model.koyck = koyckDlm(
  x = as.vector(asx.ts),
  y = as.vector(copper.ts)
  )
summarySummary(model.koyck)
```

```
##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2029.96  -212.90   -16.14   224.58  2860.21
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 133.95044  215.46356   0.622    0.535
## Y.1           0.95378    0.02126  44.865   <2e-16 ***
## X.t           0.03475    0.05276   0.659    0.511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 476.5 on 157 degrees of freedom
## Multiple R-Squared: 0.9496,  Adjusted R-squared: 0.949
## Wald test:  1480 on 2 and 157 DF,  p-value: < 2.2e-16
##
##                             alpha       beta       phi
## Geometric coefficients:  2898.194 0.03474731 0.9537814
##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2029.96  -212.90   -16.14   224.58  2860.21
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 133.95044  215.46356   0.622    0.535
## Y.1           0.95378    0.02126  44.865   <2e-16 ***
## X.t           0.03475    0.05276   0.659    0.511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 476.5 on 157 degrees of freedom
## Multiple R-Squared: 0.9496,  Adjusted R-squared: 0.949
## Wald test:  1480 on 2 and 157 DF,  p-value: < 2.2e-16
##
##                             alpha       beta       phi
## Geometric coefficients:  2898.194 0.03474731 0.9537814
```

```
checkresiduals(model.koyck$model)
```

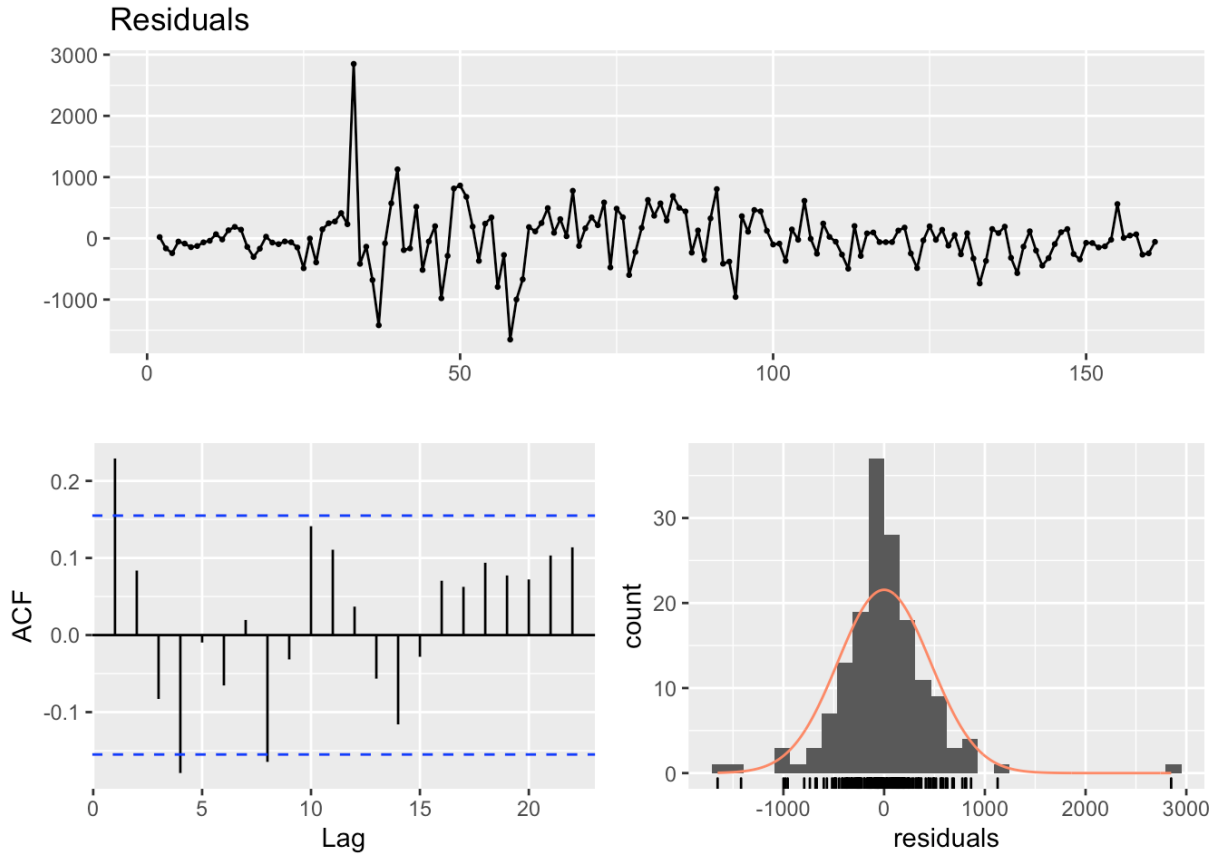


- And we have a winner, with an adjusted R-squared of `0.949`
- So high that it's possibly an overfit

```
# Fit autoregressive distributed lag models
model.ardlDlm = ardlDlm(
  x = as.vector(asx.ts),
  y = as.vector(copper.ts),
  p = 1,
  q = 1
)
summarySummary(model.ardlDlm)
```

```
##
## 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
## -1652.99  -242.96   -23.53   193.07  2849.75
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.1971   204.6539   0.133  0.89445
## X.t           0.6009     0.1842   3.263  0.00136 **
## X.1          -0.5500     0.1858  -2.961  0.00355 **
## Y.1           0.9569     0.0209  45.797  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 464.2 on 156 degrees of freedom
## Multiple R-squared:  0.9525, Adjusted R-squared:  0.9516
## F-statistic:  1043 on 3 and 156 DF,  p-value: < 2.2e-16
##
## adjusted r-squared: 0.95
##
## 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
## -1652.99  -242.96   -23.53   193.07  2849.75
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.1971   204.6539   0.133  0.89445
## X.t           0.6009     0.1842   3.263  0.00136 **
## X.1          -0.5500     0.1858  -2.961  0.00355 **
## Y.1           0.9569     0.0209  45.797  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 464.2 on 156 degrees of freedom
## Multiple R-squared:  0.9525, Adjusted R-squared:  0.9516
## F-statistic:  1043 on 3 and 156 DF,  p-value: < 2.2e-16
##
## p-value: 5.728645e-103
```

```
checkresiduals(model.ardlDlm$model)
```



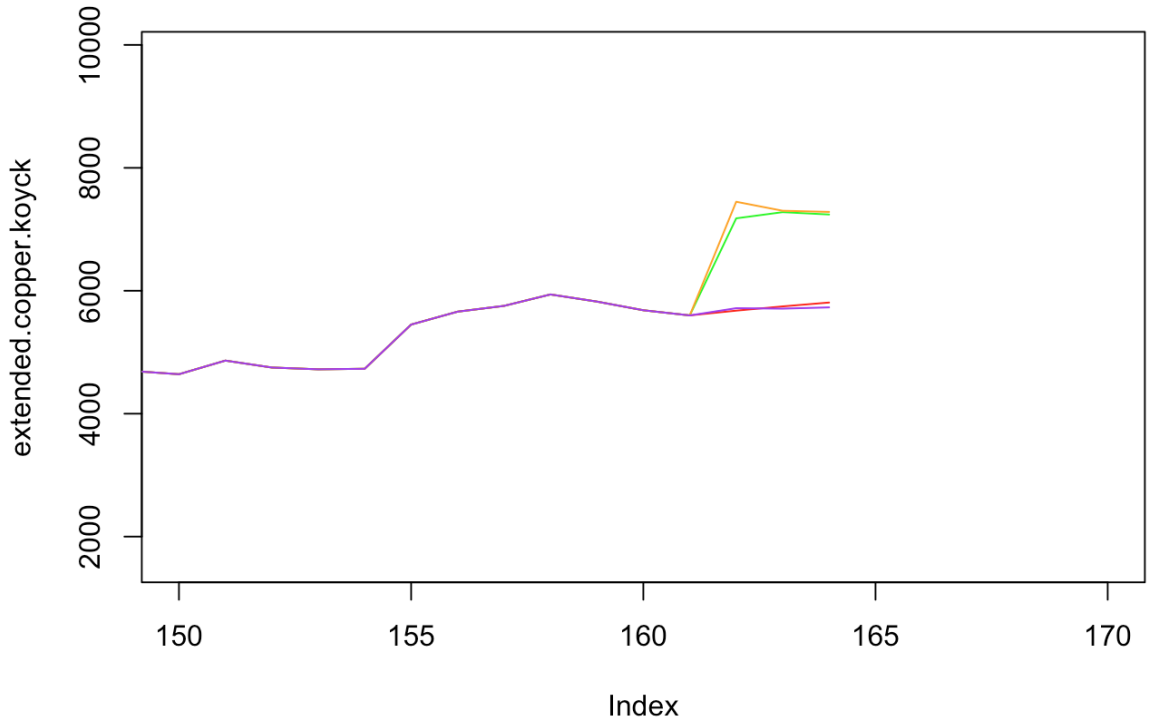

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 27.009, df = 10, p-value = 0.002596
```

- Adjusted R-squared: `0.9516`
- Fairly normal residuals
- Also a great model

Forecasting

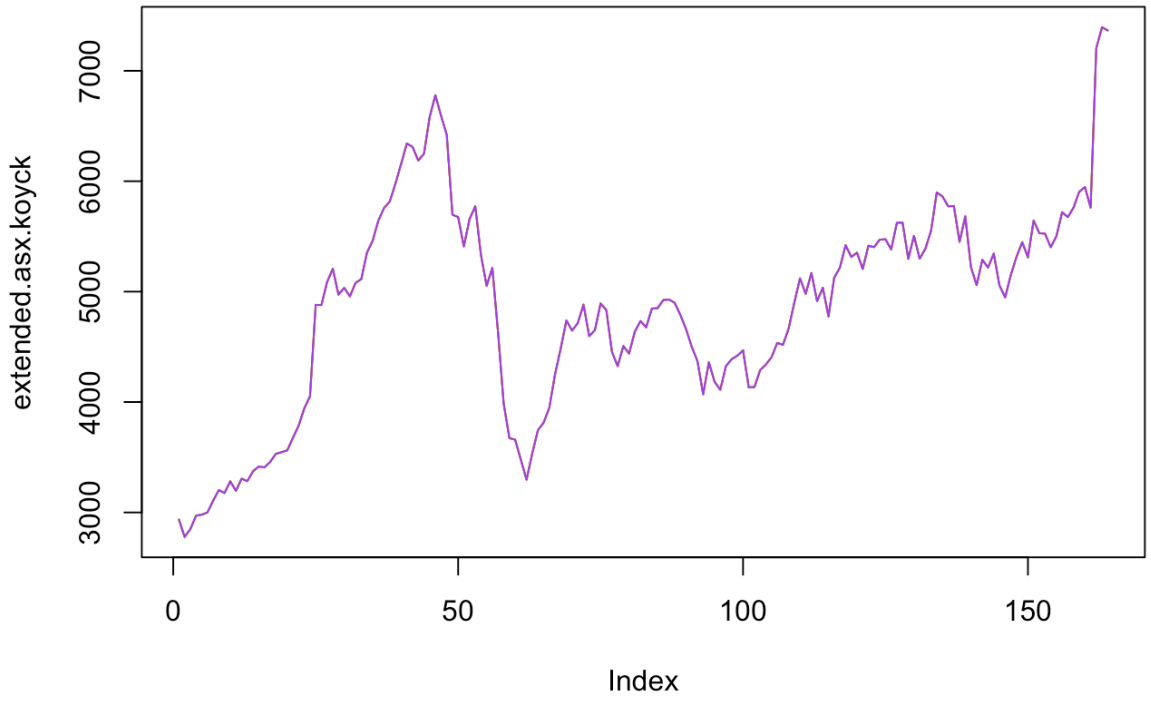
```
forecast.copper.dlm <- doForecast(model.dlm, copper.ts)
extended.copper.dlm <- c(copper.ts , forecast.copper.dlm)
forecast.copper.polyDlm <- doForecast(model.polyDlm, copper.ts)
extended.copper.polyDlm <- c(copper.ts, forecast.copper.polyDlm)
forecast.copper.koyck <- doForecast(model.koyck, copper.ts)
extended.copper.koyck <- c(copper.ts, forecast.copper.koyck)
forecast.copper.ardlDlm <- doForecast(model.ardlDlm, copper.ts)
extended.copper.ardlDlm <- c(copper.ts, forecast.copper.ardlDlm)

forecast.asx.dlm <- doForecast(model.dlm, asx.ts)
extended.asx.dlm <- c(asx.ts, forecast.asx.dlm)
forecast.asx.polyDlm <- doForecast(model.polyDlm, asx.ts)
extended.asx.polyDlm <- c(asx.ts, forecast.asx.dlm)
forecast.asx.koyck <- doForecast(model.koyck, asx.ts)
extended.asx.koyck <- c(asx.ts, forecast.asx.dlm)
forecast.asx.ardlDlm <- doForecast(model.ardlDlm, asx.ts)
extended.asx.ardlDlm <- c(asx.ts, forecast.asx.dlm)

forecast.oil.dlm <- doForecast(model.dlm, oil.ts)
extended.oil.dlm <- c(oil.ts, forecast.oil.dlm)
forecast.oil.polyDlm <- doForecast(model.polyDlm, oil.ts)
extended.oil.polyDlm <- c(oil.ts, forecast.oil.polyDlm)
forecast.oil.koyck <- doForecast(model.koyck, oil.ts)
extended.oil.koyck <- c(oil.ts, forecast.oil.koyck)
forecast.oil.ardlDlm <- doForecast(model.ardlDlm, oil.ts)
extended.oil.ardlDlm <- c(oil.ts, forecast.oil.ardlDlm)

forecast.gold.dlm <- doForecast(model.dlm, gold.ts)
extended.gold.dlm <- c(gold.ts, forecast.gold.dlm)
forecast.gold.polyDlm <- doForecast(model.polyDlm, gold.ts)
extended.gold.polyDlm <- c(gold.ts, forecast.gold.polyDlm)
forecast.gold.koyck <- doForecast(model.koyck, gold.ts)
extended.gold.koyck <- c(gold.ts, forecast.gold.koyck)
forecast.gold.ardlDlm <- doForecast(model.ardlDlm, gold.ts)
extended.gold.ardlDlm <- c(gold.ts, forecast.gold.ardlDlm)
```

```
plot(
  extended.copper.koyck,
  col='Red',
  type='l',
  xlim=c(150,170)
)
lines(extended.copper.dlm, col='Green', type='l')
lines(extended.copper.polyDlm, col='Orange', type='l')
lines(extended.copper.ardlDlm, col='Purple', type='l')
```



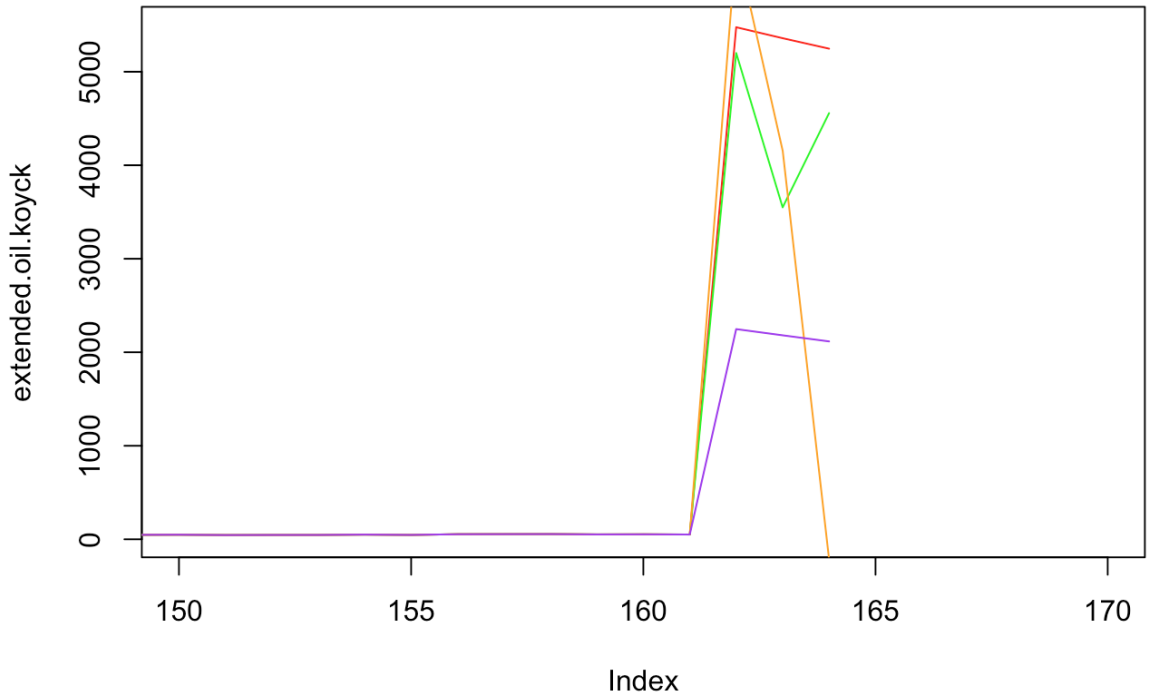
ASX Forecast

```
plot(
  extended.asx.koyck,
  col='Red',
  type='l',
  # xlim=c(150,170)
)
lines(extended.asx.dlm, col='Green', type='l')
lines(extended.asx.polyDlm, col='Orange', type='l')
lines(extended.asx.ardlDlm, col='Purple', type='l')
```



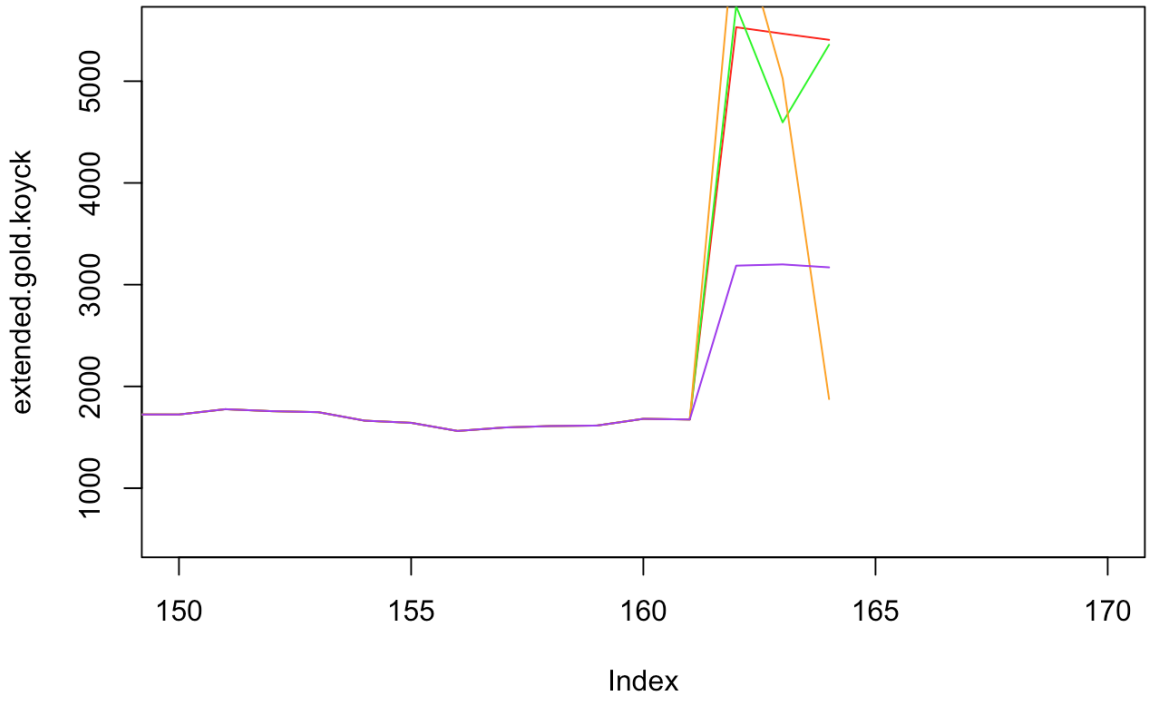
Oil Forecast

```
plot(
  extended.oil.koyck,
  col='Red',
  type='l',
  xlim=c(150,170)
)
lines(extended.oil.dlm, col='Green', type='l')
lines(extended.oil.polyDlm, col='Orange', type='l')
lines(extended.oil.ardlDlm, col='Purple', type='l')
```



Gold Forecast

```
plot(
  extended.gold.koyck,
  col='Red',
  type='l',
  xlim=c(150,170)
)
lines(extended.gold.dlm, col='Green', type='l')
lines(extended.gold.polyDlm, col='Orange', type='l')
lines(extended.gold.ardlDlm, col='Purple', type='l')
```



Conclusion

- We haven't rated them by AIC or BIC because of how poor the dlm fits are
- Both `model.ardlDlm = 0.9516` and `model.koyck = 0.949` have the best adjusted R-squared
- We select ardlDlm as the slightly better model

Source

- Monthly averages of ASX All Ordinaries (Ords) (https://www.asx.com.au/products/index-charts.htm) - Price Index, Gold price (AUD), Crude Oil (Brent, USD/bbl) and Copper (USD/tonne) starting from January 2004]()

Criteria	Not acceptable (0)	Needs Improvement (0-2]	Under expectation (2-4]	Pass (4-6]	Meets expectation (6-7]	Creates Distinction (7-8]	Creates High Distinction (8-10]
Reporting (20%)	Lacks the basic elements of a precise report. Some of the outputs have not been associated with inferences or there are inferences without supporting evidence.	Inferences are not addressed by relevant outputs and given under relevant sections. Some of the output remains without being inferred. There are language and formatting issues.	Inferences are addressed by relevant outputs and given under relevant sections. There are missing sections. There are minor language and formatting issues.	All inferences are addressed by relevant outputs and given under relevant sections. There is no missing section. There are no language and formatting issues.	All inferences are addressed by relevant outputs and given under relevant sections. There is no missing section. There are no language and formatting issues.	All inferences are clearly addressed by relevant outputs and given under relevant sections. All bits of the presented output are used for the inferences. There is no missing section. There are no language and formatting issues.	All inferences are clearly addressed by relevant outputs and given under relevant sections. All bits of the presented output are used for the inferences in a way to provide a mature discussion of analysis results. There is no missing section. There are no language and formatting issues.
R codes (25%)	Some of the scripts to run analyses are generating error messages. It is unclear what does each line do in the R codes.	All scripts to run analyses are working properly. However, it is hard to follow codes to identify what each line does.	All scripts to run analyses are working properly and suitable explanations are given on the codes.	Scripts to run analyses are working properly and suitable explanations are given on the code. Suitable object names are used to make it easy to follow the code.	Scripts to run analyses are working properly and suitable explanations are given on the code. Suitable object names are used to make it easy to follow the code. The script is arranged in a way not to repeat the same code chunks.	Scripts to run analyses are working properly and suitable explanations are given on the code. Suitable object names are used to make it easy to follow the code. The scripts include user-created functions to implement all analyses without using the same code chunks repeatedly.	Scripts to run analyses are working properly and suitable explanations are given on the code. Suitable object names are used to make it easy to follow the code. The scripts include user-created functions to implement all analyses in a dynamical way.
Descriptive analysis (%10)	Nearly no descriptive analysis has done and/or the descriptive analysis is insufficient.	Some descriptive plots/statistics are used and/or presented plots/statistics have interpretations.	Some descriptive plots/statistics are used and/or presented all plots/statistics have interpretations.	Uses all suitable plots and descriptive statistics and some plots and descriptive statistics have a suitable interpretation in one of the tasks.	Uses all suitable plots and descriptive statistics and each plot and the descriptive statistic have a suitable interpretation.	Uses all suitable plots and descriptive statistics are displayed. All plots and descriptive statistics are interpreted in a way to shed light on further analyses.	Uses all suitable plots and descriptive statistics are displayed. All plots and descriptive statistics are interpreted in a way to demonstrate an outstanding understating of concept targeted by the descriptive statistic/plot.
Choice of variables/model (15%)	An unsuitable model is chosen or all variables in a model are unsuitable for the aim of analysis.	A suitable model is chosen but some or all variables in a model are unsuitable for the aim of analysis.	A suitable model is chosen but some variables in a model are unsuitable for the aim of analysis.	A suitable model is chosen and all variables are suitable for the aim of analysis.	A suitable model is chosen and all variables are suitable for the aim of analysis. The reason behind the selection of the model is explained.	A suitable model is chosen and all variables are suitable for the aim of analysis. The reason behind the selection of the model is articulated based on the results of descriptive statistics.	A suitable model is chosen and all variables are suitable for the aim of analysis. Insightful reasons given about the selection of the model.
Implementation of models (15%)	Specified models are not fitted and there is no attempt to move forward with the analysis.	Some models fitted but there is no attempt to draw inferences from the model outputs.	Some models fitted but there is a limited amount of effort to draw inferences from the model outputs.	All possible models are fitted but there is a limited amount of effort to draw inferences from the model outputs.	All possible models are fitted and there is some effort to draw inferences from the model outputs.	All possible models are fitted and all the relevant outputs are included in the comments made on the fitted models.	All possible models are fitted and all the relevant outputs are clearly interpreted in a way to inform the next step of the analysis.
Diagnostic checking (15%)	There is no effort for diagnostic checking and validation of the assumptions of applied approaches.	Some aspects of diagnostic checking are applied and some of the assumptions of applied approaches are validated.	Some aspects of diagnostic checking are applied and most of the assumptions of applied approaches are validated.	Some aspects of diagnostic checking are applied and all assumptions of applied approaches are validated.	All aspects of diagnostic checking are applied and all assumptions of applied approaches are validated.	All aspects of diagnostic checking are applied and all assumptions of applied approaches are validated. The results are articulated in a way to demonstrate a good understanding of diagnostic checking.	All aspects of diagnostic checking are applied and all assumptions of applied approaches are validated. A mature discussion of the information gathered by diagnostic checking is given.

The dataset you will analyse in this assignment includes monthly averages of ASX: