Forecasting Assignment1

Maitreya Milind Kadam (s4087536)

2025-08-28

Introduction

The aim of this assignment is to analyze the time series and to check the existence of non-stationarity present in the variables. It is also expected to talk about the impact of the components of a time series. In addition to this, the most accurate and distributed lag models are also to be explored based on the relation between ASX Price Index and the other variables.

Data Description

I have loaded and inspected the structure of the dataset provided. After that I have converted the dataframe into a time series object for analysis based on visualization.

```
a1_data=read_csv('ASX_data.csv')
```

```
## Rows: 161 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (2): ASX price, Crude Oil (Brent)_USD/bbl
## num (2): Gold price, Copper_USD/tonne
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(a1_data)
```

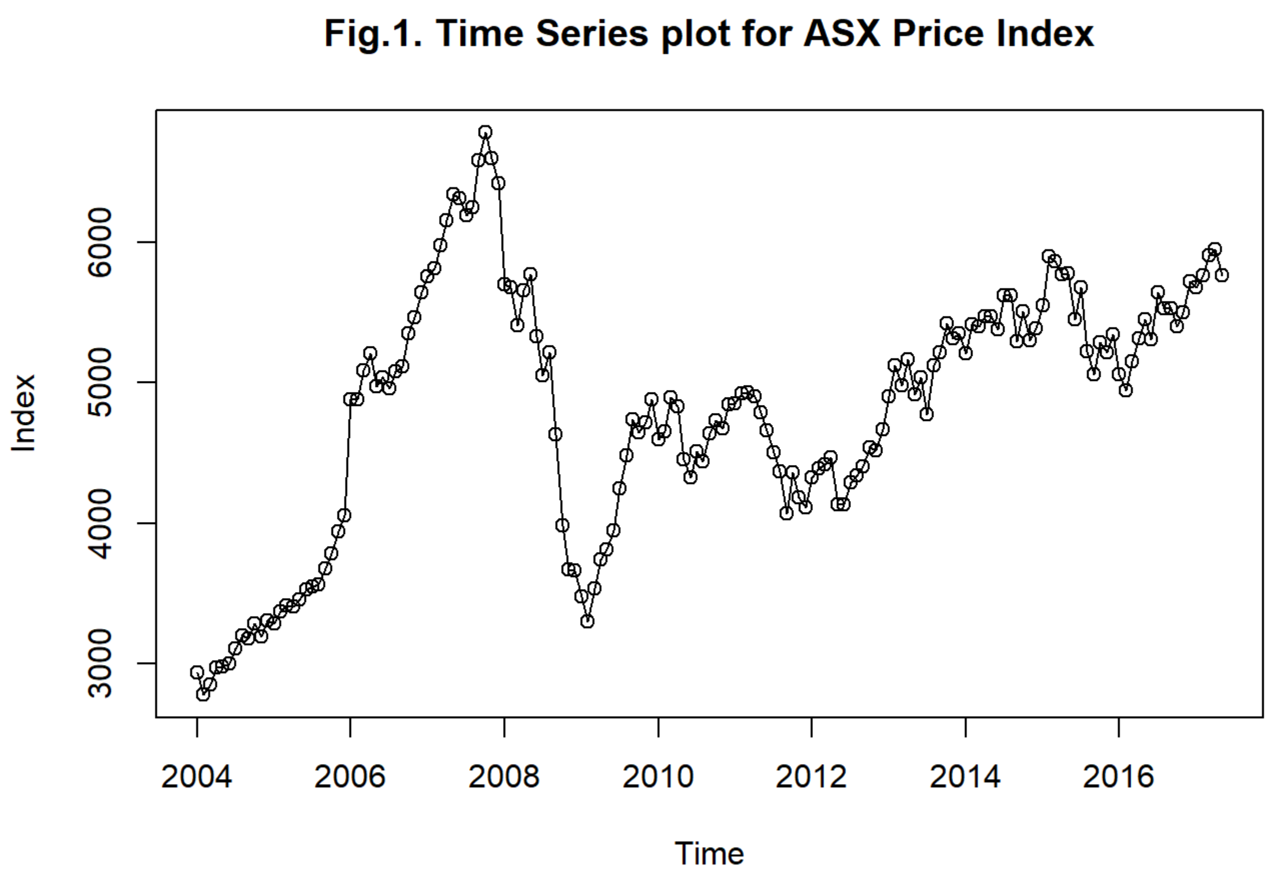
```
## spc_tbl_ [161 × 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ASX price                : num [1:161] 2935 2778 2849 2971 2980 ...
##  $ Gold price               : num [1:161] 612 603 566 539 549 ...
##  $ Crude Oil (Brent)_USD/bbl: num [1:161] 31.3 32.6 30.3 25 25.8 ...
##  $ Copper_USD/tonne         : num [1:161] 1650 1682 1656 1588 1651 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `ASX price` = col_double(),
##   ..   `Gold price` = col_number(),
##   ..   `Crude Oil (Brent)_USD/bbl` = col_double(),
##   ..   `Copper_USD/tonne` = col_number()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The dataframe a1_data is to be converted into time series for further analysis.

```
asx=ts(a1_data$`ASX price`,start=c(2004,1),frequency=12)
gold=ts(a1_data$`Gold price`,start=c(2004,1),frequency=12)
crude_oil=ts(a1_data$`Crude Oil (Brent)_USD/bbl`,start=c(2004,1),frequency=12)
copper=ts(a1_data$`Copper_USD/tonne`,start=c(2004,1),frequency=12)
data.ts=ts(a1_data[,1:4],start=c(2004,1),frequency=12)
```

Time Series visualisation for all the variables

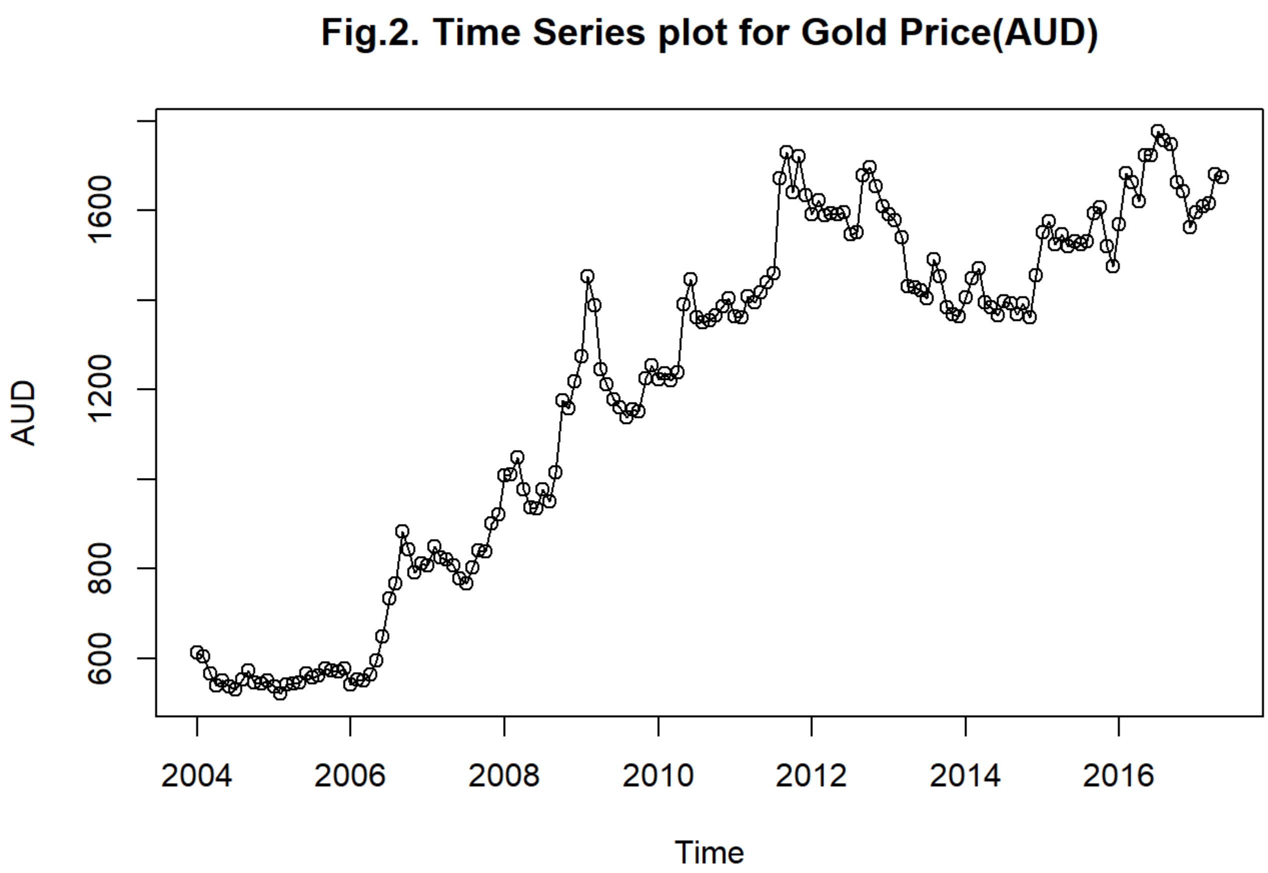
```
plot(asx, main="Fig.1. Time Series plot for ASX Price Index", ylab="Index",type='o')
```



Based on Fig.1. the following points can be infered:

1. Trend: The time series plot indicates a long-term upward trend which indicates that the ASX Price Index generally increased over the period with some declines in between. There is a strong upward trend from 2004 to early 2008, followed by a sharp decline which is likely to be associated with the global financial crisis.
2. Seasonality: The plot does not indicate any obvious seasonality over the period as the plot appears to be irregular and not tied to specific times of the year.
3. Variance: The fluctuations appear to be irregular and do not seem to repeat on a yearly basis which indicates high variance in the data.
4. Change Point: The most significant change point in the time series occured around 2008 which is likely to be caused by the global financial crisis.
5. The absence of constant mean and an overall trend suggests that the series is likely non-stationary. Further analysis using ACF and PACF plots with statistical tests like the ADF test would be necessary to confirm this observation.

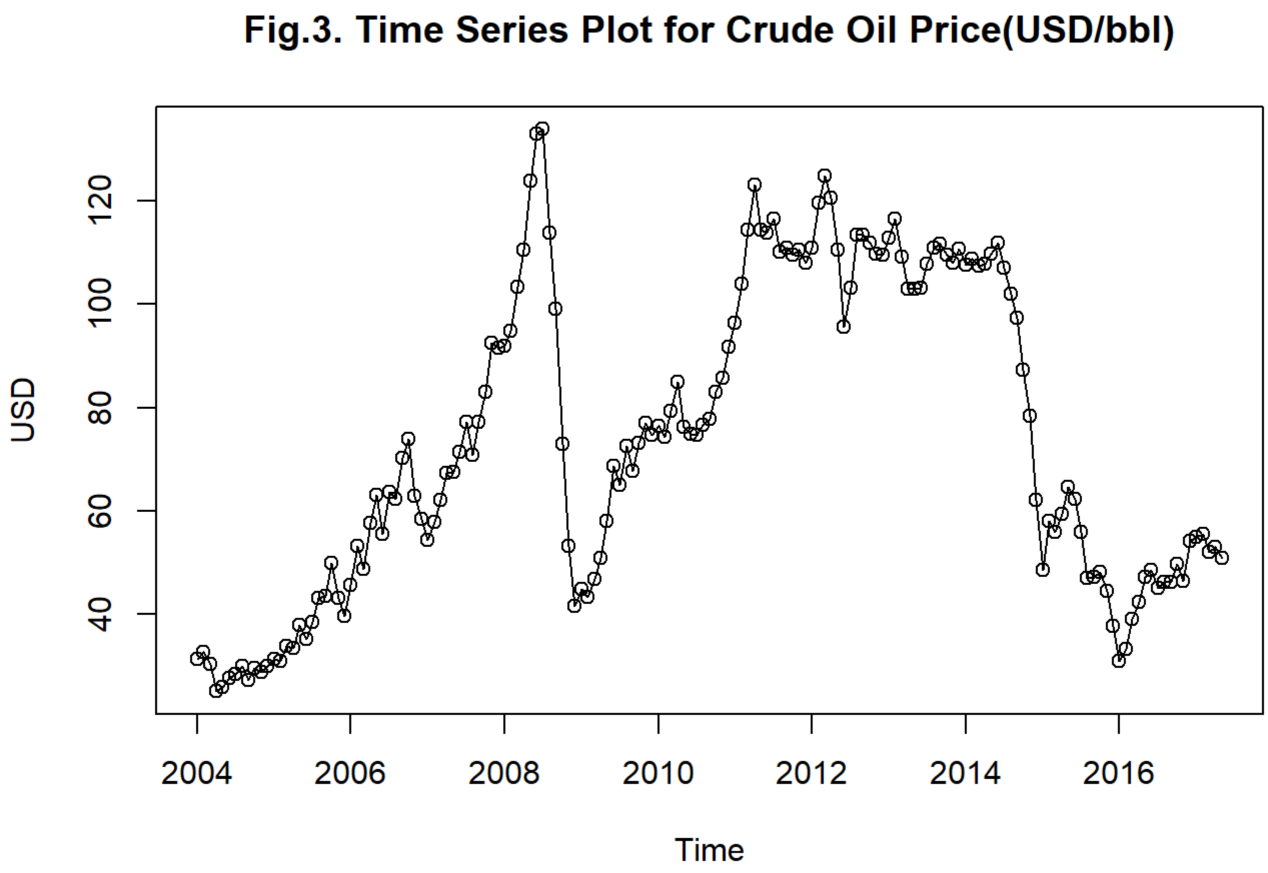
```
plot(gold, main='Fig.2. Time Series plot for Gold Price(AUD)',ylab='AUD',type='o')
```



Based on Fig.2. the following points can be infered:

1. Trend: The time series plot for Gold indicates a long-term upward trend which indicates that the gold price increased over the entire period. However, this trend is not smooth with certain jumps and drops which can be observed.
2. Seasonality: The fluctuations seem to be irregular which indicates the absence of a seasonal pattern for the time series.
3. Variance: The time series suggests that the variance is not constant and this volatility seems to increase as the gold price increases. The magnitude for these fluctuations are much larger in the later years.
4. Change Points: There is no significant changepoint in this time series. The time series is likely to be non-stationary as there is no constant mean and there is an upward trend. Further analysis using the ACF, PACF and ADF tests would confirm if the series is non-stationary.

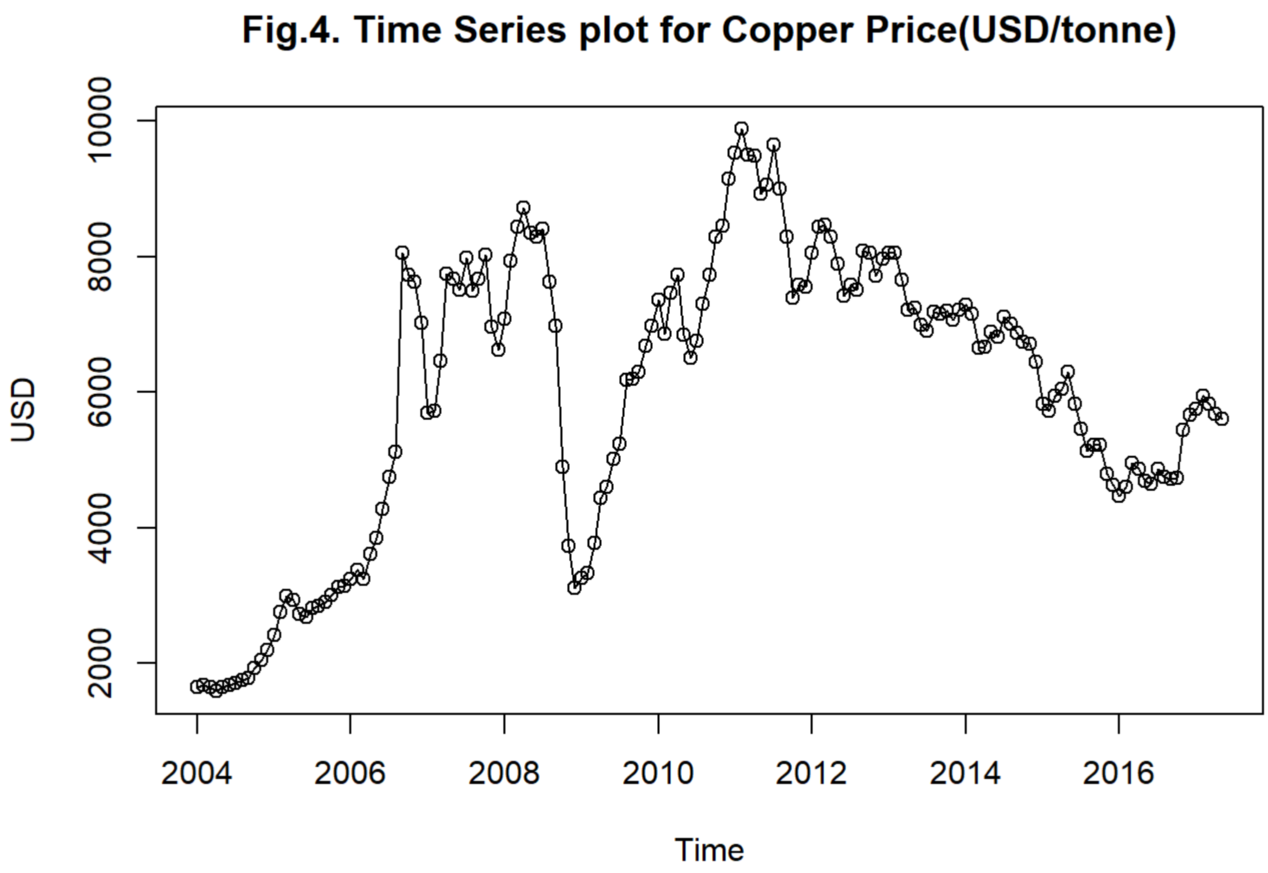
```
plot(crude_oil,main='Fig.3. Time Series Plot for Crude Oil Price(USD/bbl)',ylab="USD",type
='o')
```



Based on Fig.3. the following points can be infered:

1. Trend: This particular time series exhibits a mix of upward trend and downward trend. The upward trend appears to be from 2004 to mid 2008 which is followed by a sharp decline.
2. Seasonality: The large fluctuations are irregular and does not appear to follow a regular pattern over the period which does not indicate any seasonality.
3. Variance: As mentioned earlier, the large fluctuations indicate that the variance of the data is clearly not constant.
4. Change Point: From the above time series plots, there appear to be 2 significant change points, the first around 2008 where the price peaked and then showed a sharp decline. The second is around 2014 where the price shows a sharp decline once again.
5. This time series too shows absence of a constant mean and a constant variance which indicates that the series is non-stationary and this can be confirmed by the further analysis using ACF, PACF, and ADF tests.

```
plot(copper,main='Fig.4. Time Series plot for Copper Price(USD/tonne)',ylab='USD',type='o')
```

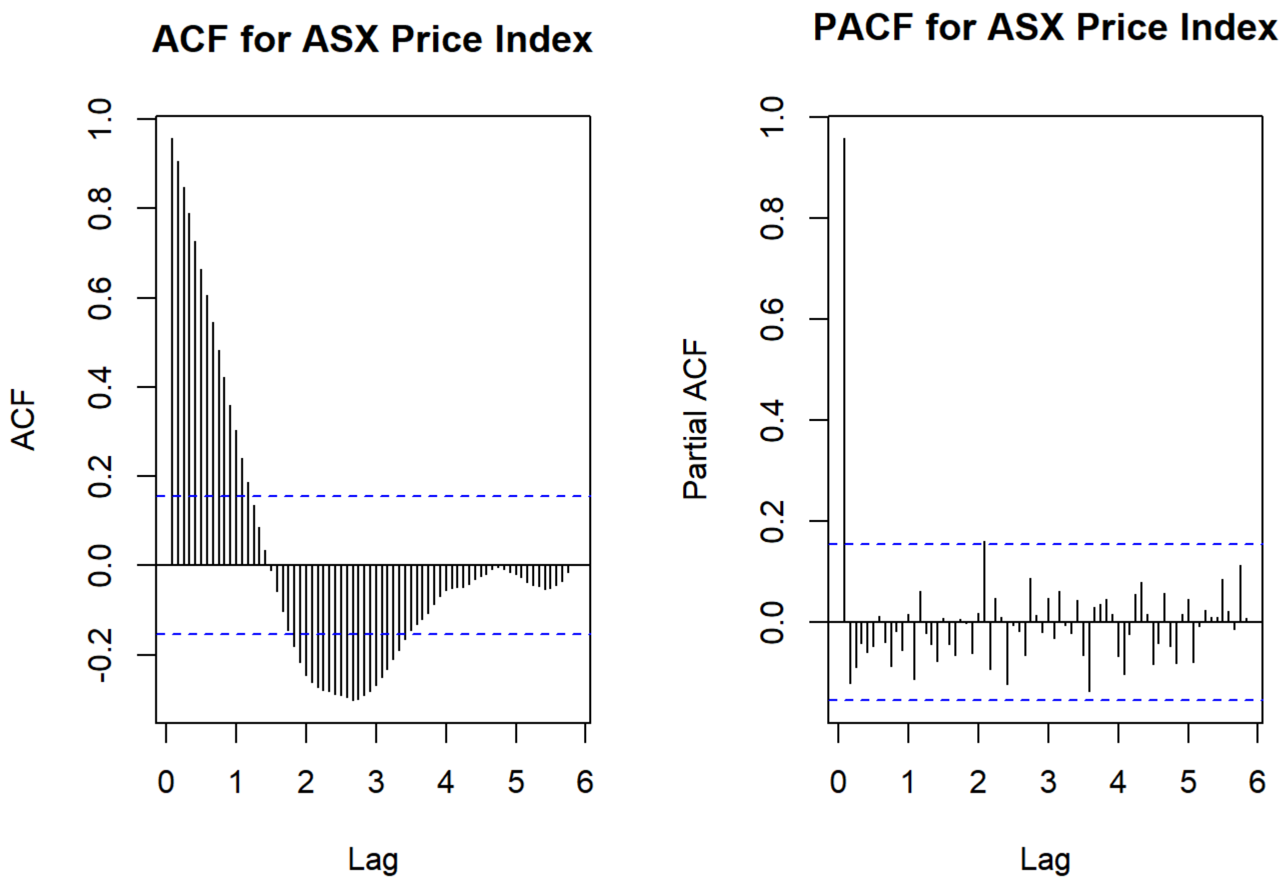


Based on Fig.4. the following points can be inferred:

1. Trend: The time series plot for copper price shows a clear non-linear trend. There is an upward trend from 2004 to 2008, which is followed by a sharp drop.
2. Seasonality: The movements and large fluctuations appear to be irregular and they don't seem to follow a regular pattern which indicates that there is no seasonality present.
3. Variance: It is clear from the above plot that the variance is not constant, the fluctuations are significant and of a large magnitude.
4. Change Point: The most prominent change point is around 2008 which results in a sharp drop of the copper price.
5. This time series too appears to be non-stationary as there is no constant mean and this can be confirmed by further analysis using ACF, PACF, and ADF tests.

Existence of non stationarity in the dataset

```
#ACF and PACF plot for ASX Price Index
par(mfrow=c(1,2))
acf(asx,lag.max=70,main='ACF for ASX Price Index')
pacf(asx,lag.max=70,main='PACF for ASX Price Index')
```



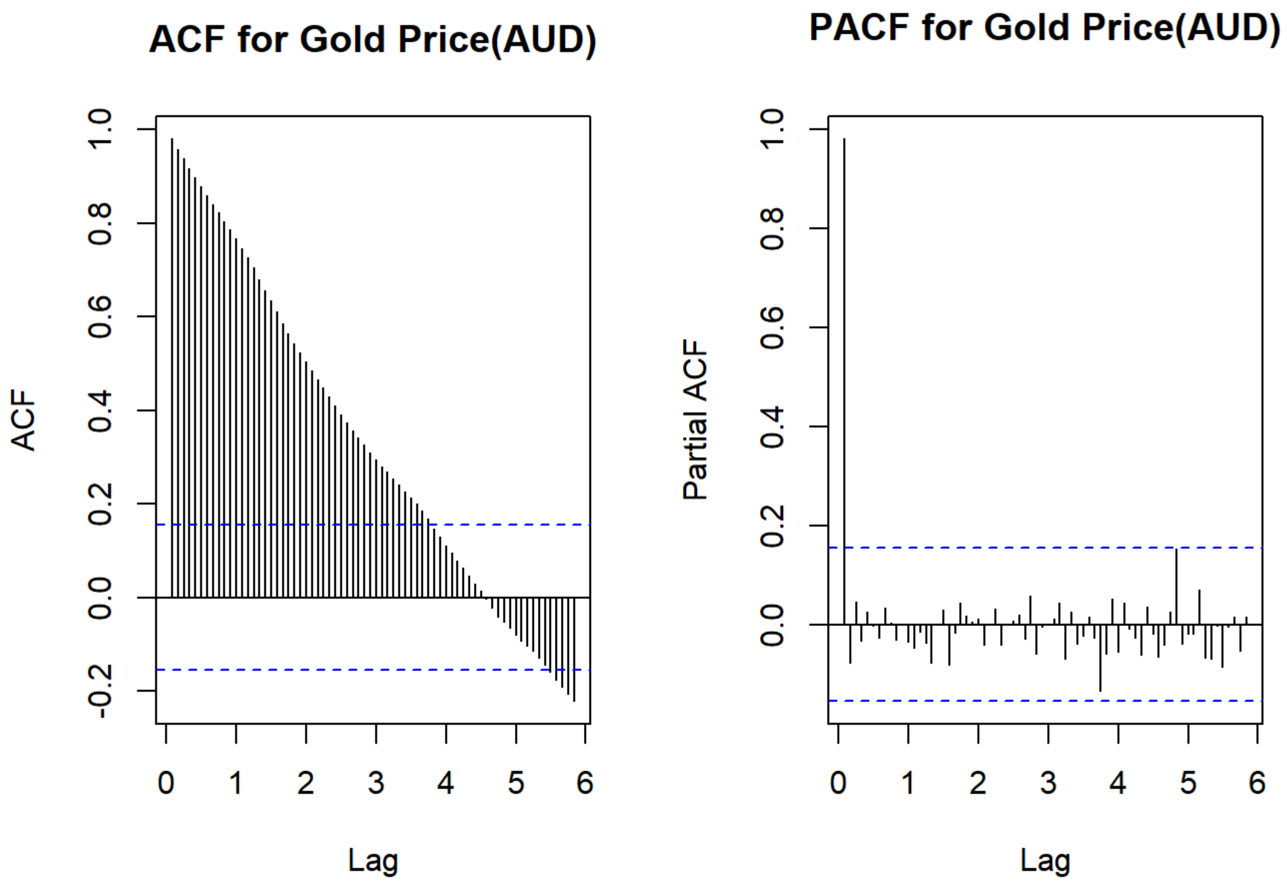
- The slow decay in the above ACF plot clearly indicates that the series in non stationary, and it is also a classic sign of a strong trend in the data. There is no evidence of seasonality present in the data.
- In the PACF plot, the presence of a very large significant spike at lag 1, followed by a sharp drop-off to insignificant values indicates that the series is non stationary.

```
adf.test(asx)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  asx
## Dickey-Fuller = -2.6995, Lag order = 5, p-value = 0.2846
## alternative hypothesis: stationary
```

- The p-value of 0.2846 from the ADF test is >0.05, this fails to reject the null hypothesis of non-stationarity.
- Based on the observations from the ACF, PACF plots and the ADF test it is confirmed that non stationarity exists in the time series.

```
#ACF and PACF plot for Gold Price
par(mfrow=c(1,2))
acf(gold,lag.max=70,main='ACF for Gold Price(AUD)')
pacf(gold,lag.max=70,main='PACF for Gold Price(AUD)')
```



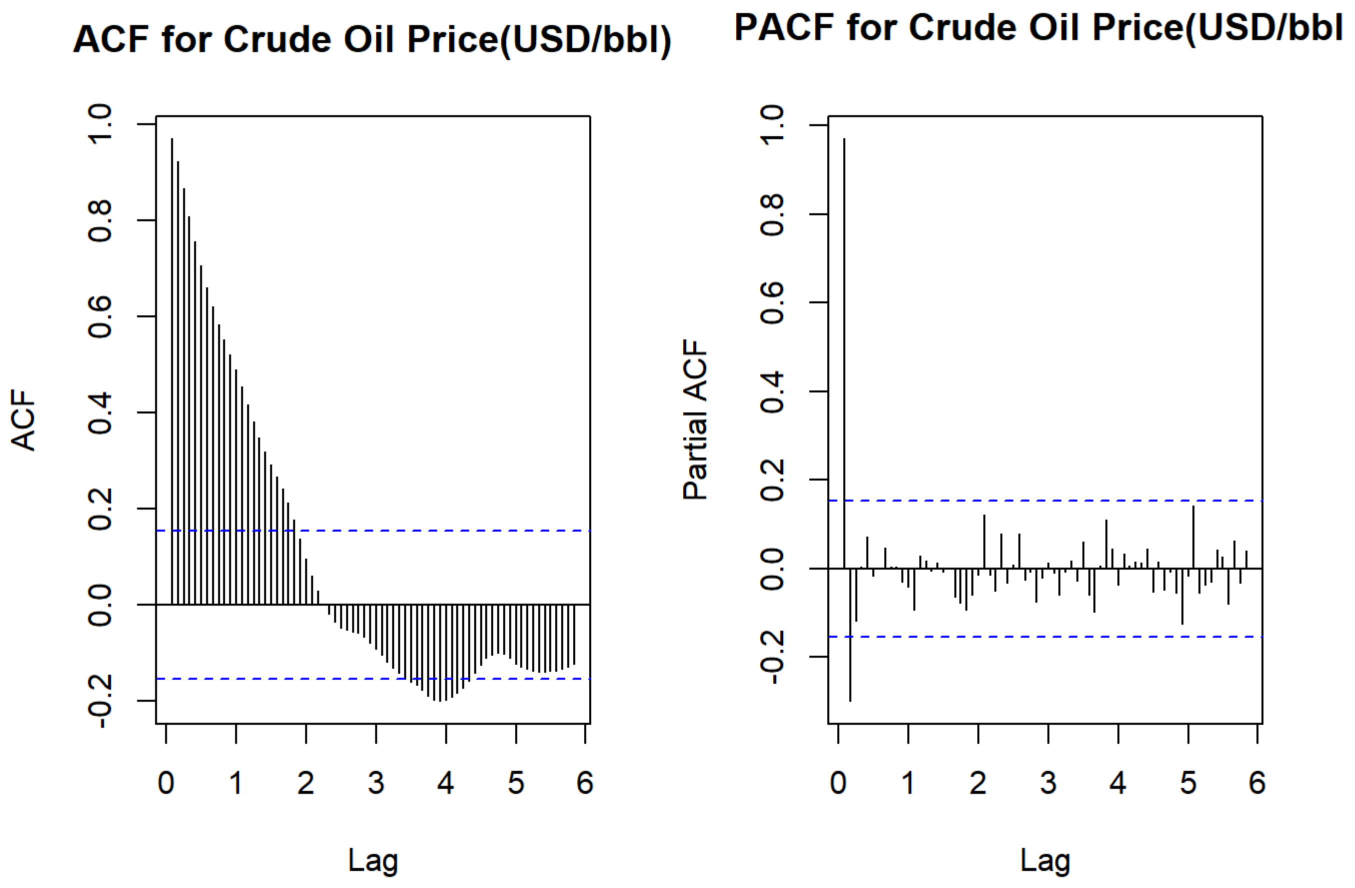
- The gradual decay in the ACF plot for Gold price suggests that the series is non-stationary and it also highlights the presence of a strong trend in the data. As the ACF plot does not exhibit a repeating pattern there is no seasonality present in the data.
- As the PACF plot begins with a large spike at lag 1, which is followed by insignificant values indicates the non-stationary behaviour of the data.

```
adf.test(gold)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gold
## Dickey-Fuller = -1.8369, Lag order = 5, p-value = 0.6444
## alternative hypothesis: stationary
```

- Based on the ADF test, the p-value of 0.6444(>0.05) fails to reject the null hypothesis of non-stationarity and this confirms that the time series is non-stationary.

```
#ACF and PACF plot for Crude Oil Price
par(mfrow=c(1,2))
acf(crude_oil,lag.max=70,main='ACF for Crude Oil Price(USD/bbl)')
pacf(crude_oil,lag.max=70,main='PACF for Crude Oil Price(USD/bbl)')
```



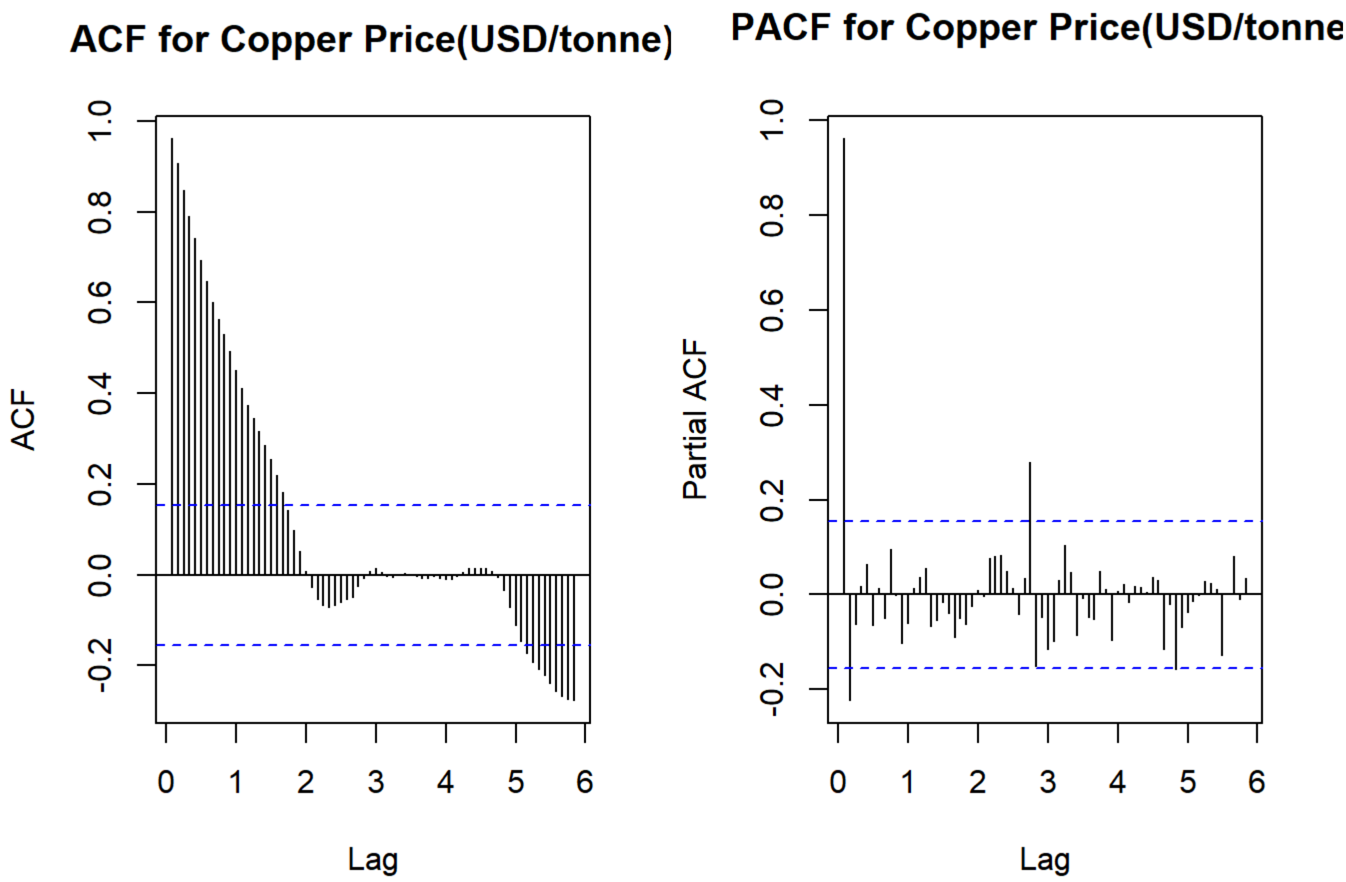
- This ACF plot for Crude Oil Price shows a decaying pattern suggesting the existence of trend. In addition to that, no seasonality was observed in the ACF plot.
- Like previous PACF plots, this plot for Crude Oil Price also shows significant spike at lag which is again followed by insignificant values. This overall behaviour indicates that the time series is non-stationary.

```
adf.test(crude_oil)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  crude_oil
## Dickey-Fuller = -1.8523, Lag order = 5, p-value = 0.6379
## alternative hypothesis: stationary
```

- After performing the ADF test, the p-value turns out to be 0.6379 which is <0.05, it fails to reject the null hypothesis non-stationarity and this confirms that the time series is non-stationary.

```
#ACF and PACF plot for Copper Price
par(mfrow=c(1,2))
acf(copper,lag.max=70,main='ACF for Copper Price(USD/tonne)')
pacf(copper,lag.max=70,main='PACF for Copper Price(USD/tonne)')
```



- The ACF plot shows a slow, gradual decay with no seasonality in the data.
- There is a significant spike at the start of the PACF plot which is followed by a sharp drop-off to low insignificant values. This evidence is sufficient to conclude that the time series is non-stationary.

```
adf.test(copper)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  copper
## Dickey-Fuller = -2.2502, Lag order = 5, p-value = 0.472
## alternative hypothesis: stationary
```

- Based on the above ADF Test, the p-value of 0.472 greater than 0.05 which means we fail to reject null hypothesis. It provides strong statistical evidence that the series in non-stationary.

```
#Box-Cox lambda for asx price index
BC_lambda_asx=BoxCox.lambda(asx)
print(paste('Lambda for ASX',BC_lambda_asx))
```

```
## [1] "Lambda for ASX 1.99992424816297"
```

```
#Box-Cox lambda for gold price
BC_lambda_gold=BoxCox.lambda(gold)
print(paste('Lambda for Gold',BC_lambda_gold))
```

```
## [1] "Lambda for Gold 0.976694988165088"
```

```
#Box-Cox lambda for crude oil price
BC_lambda_crude=BoxCox.lambda(crude_oil)
print(paste('Lambda for Crude Oil',BC_lambda_crude))
```

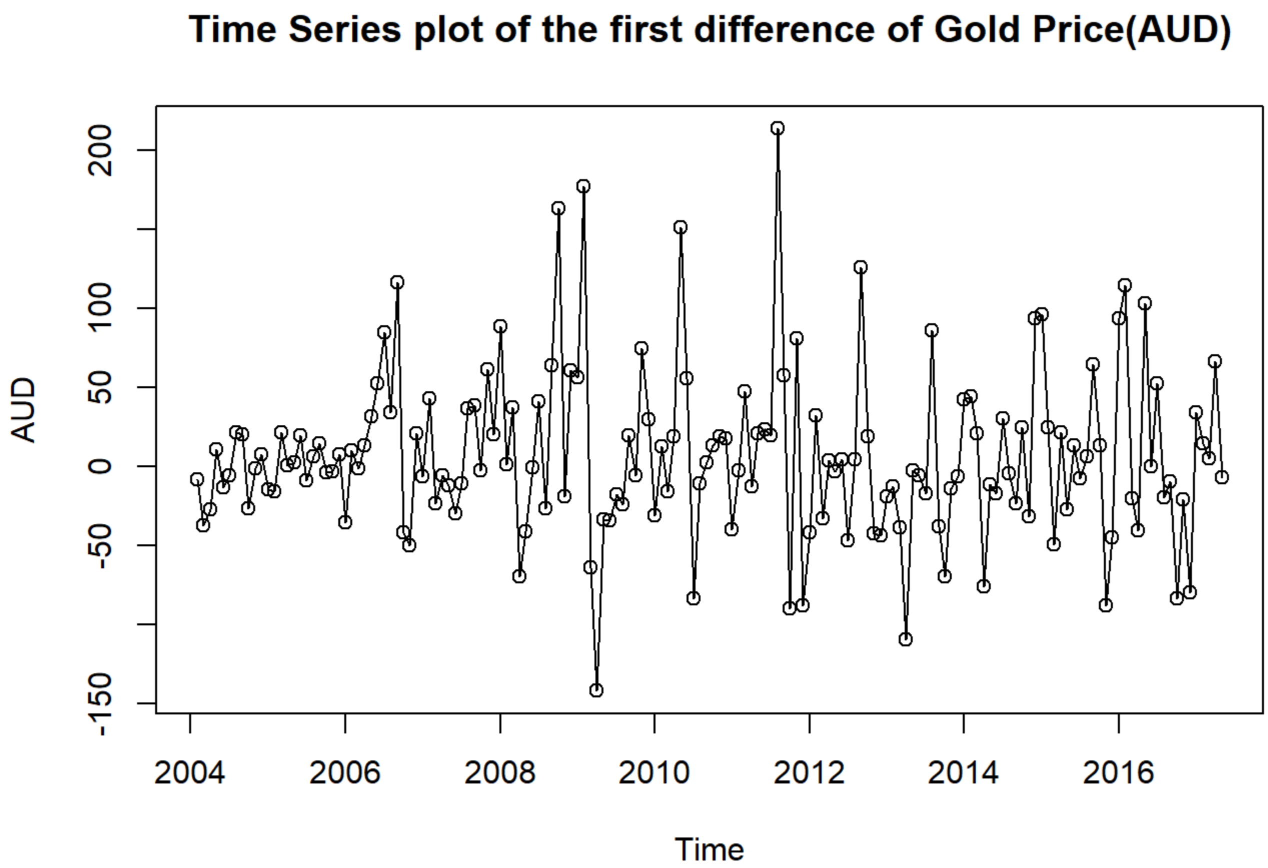
```
## [1] "Lambda for Crude Oil -0.830413646585157"
```

```
#Box-Cox lambda for copper price
BC_lambda_copper=BoxCox.lambda(copper)
print(paste('Lambda for Copper',BC_lambda_copper))
```

```
## [1] "Lambda for Copper 0.933678345677826"
```

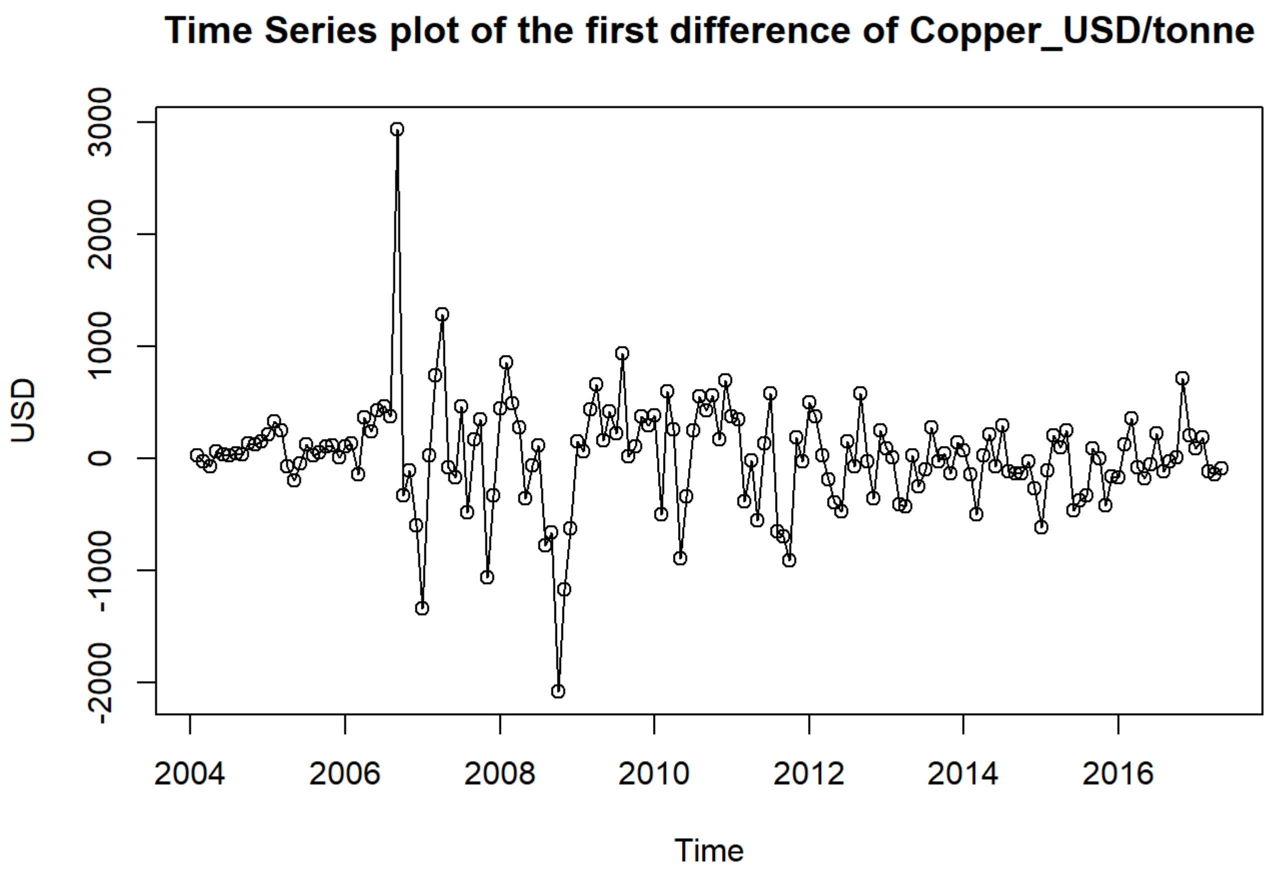
The lambda value for Gold and Copper is close to 1, which indicates that those 2 time series won't be needing any transformation. So we will just applying differencing to those 2 series and then we will proceed with the box-cox transformation and differencing for ASX Price Index and Crude Oil.

```
#first order differencing for Gold Price
gold.diff = diff(gold,differences = 1)
plot(gold.diff,ylab='AUD',
xlab='Time',main = "Time Series plot of the first difference of Gold Price(AUD)",type='o')
```



After performing differencing on the time series for gold, the plot indicates that the series now fluctuates around a constant mean. This clearly indicates that the trend is removed and the series is stationary.

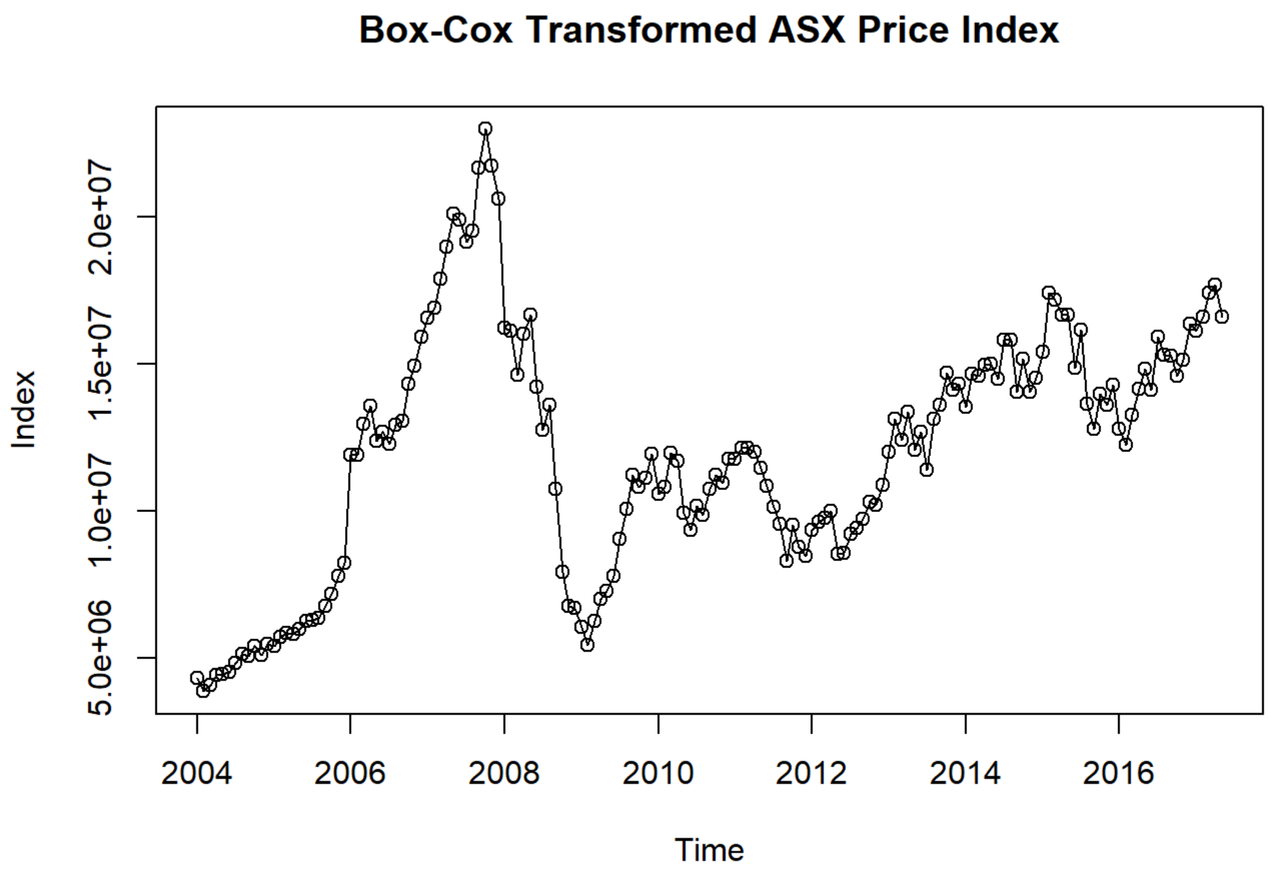
```
#first order differencing for Copper.
copper.diff = diff(copper,differences = 1)
plot(copper.diff,ylab='USD',
xlab='Time',main = "Time Series plot of the first difference of Copper_USD/tonne",type='o')
```



The above plot illustrates the differenced version of the time series for copper. It can be obeserved that the trend has be removed and now the series fluctuates around a constant mean. This is a key indicator of a stationary series. Summing it up, we can say that the trend is removed and the series is stationary.

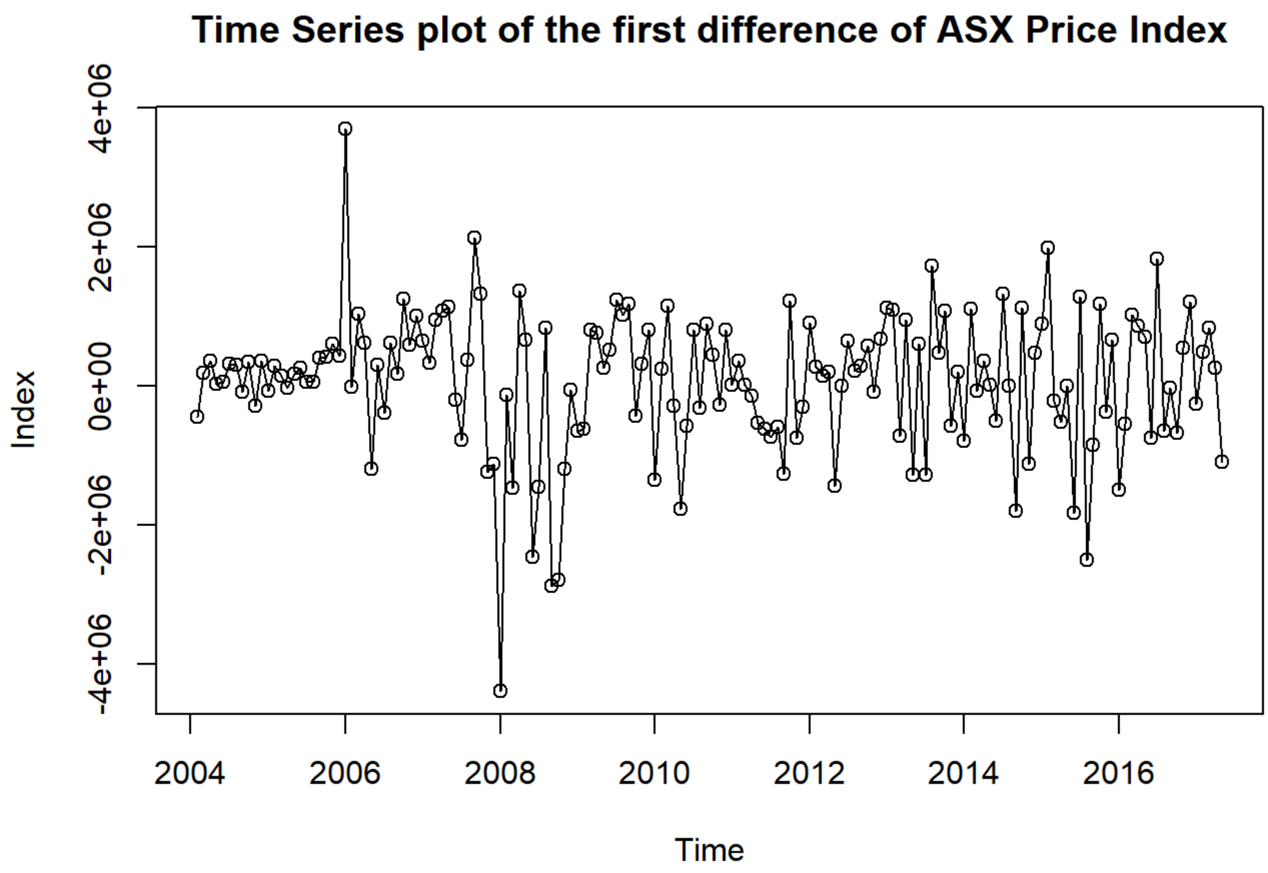
For ASX Price Index

```
#Box-Cox transformation for ASX
lambda=1.999924
BC.asx=((asx^(lambda))-1)/lambda
plot(BC.asx,ylab='Index',xlab='Time',type='o', main="Box-Cox Transformed ASX Price Index")
```



Even after applying Box-Cox transformation, the high variance in the data is still present along with the upward trend. Therefore, I am applying 1st order differencing to remove the trend and making the series stationary.

```
#first order differencing to the Box-Cox transformed data
asx.diff = diff(BC.asx,differences = 1)
plot(asx.diff,ylab='Index',
xlab='Time',main = "Time Series plot of the first difference of ASX Price Index",type='o')
```



After appplying 1st order differencing, the time series appears to fluctuate around a constant mean, which highly indicates that stationarity is acheived. However, I would still conduct an ADF test to confirm that the series is stationary.

```
adf.test(asx.diff)
```

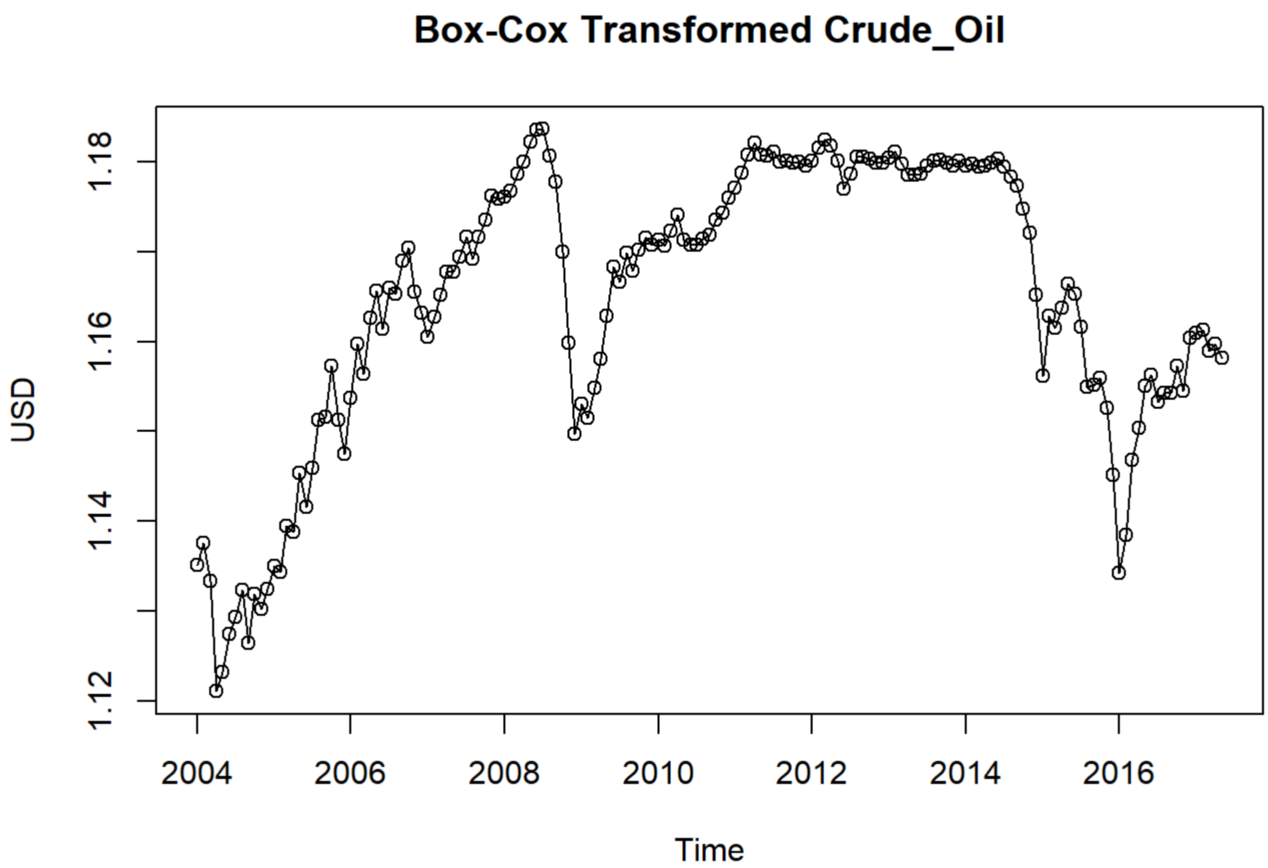
```
## Warning in adf.test(asx.diff): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  asx.diff
## Dickey-Fuller = -4.4343, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Based on the above ADF test results, the p-value is 0.01 which is less than 0.05 and therefore, it confirms that the time series is now stationary.

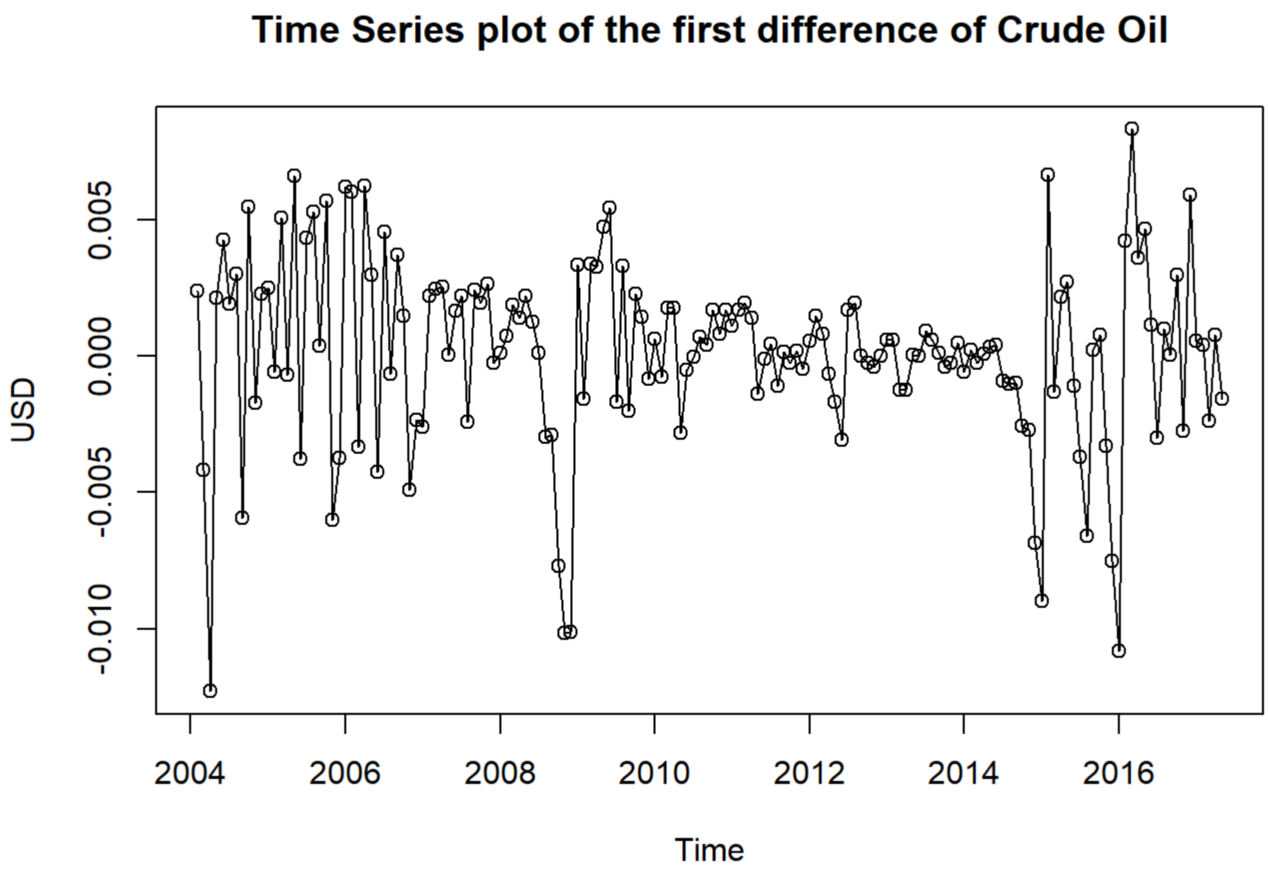
For Crude Oil Price (USD/bbl)

```
lambda=-0.8304136
BC.crude=((crude_oil^(lambda))-1)/lambda
plot(BC.crude,ylab='USD',xlab='Time',type='o', main="Box-Cox Transformed Crude_Oil")
```



Even after applying the Box-Cox transformation it is evident that the series still exhibits a clear trend and high volatility. The transformation was not sufficient to make the series stationary. Therefore, I would be applying first-order differencing to stabilize the mean and make the series stationary.

```
#first order differencing to the Box-Cox transformed data for Crude Oil
crude.diff = diff(BC.crude,differences = 1)
plot(crude.diff,ylab='USD',
xlab='Time',main = "Time Series plot of the first difference of Crude Oil",type='o')
```



Upon applying 1st order differencing, it can be observed that the trend has be removed and the differenced series fluctuates around a constant mean which clearly indicates that the trend has been eliminated and the series is now stationary.

```
adf.test(crude.diff)
```

```
## Warning in adf.test(crude.diff): p-value smaller than printed p-value
```

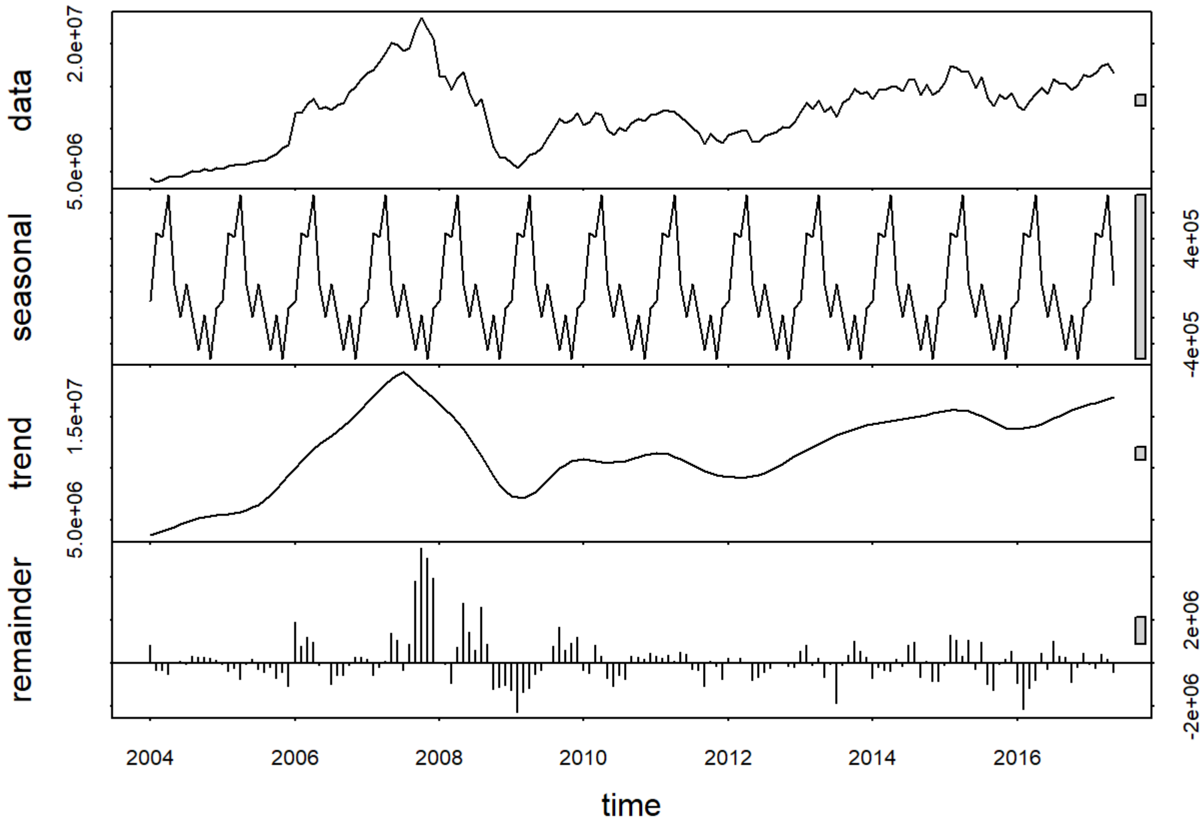
```
##
##  Augmented Dickey-Fuller Test
##
## data:  crude.diff
## Dickey-Fuller = -5.5931, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

From the above ADF result it can be seen that the p-value is 0.01 which is less than 0.05 and this indicates that the series is stationary.

Components of a Time Series Data

STL Decomposition for ASX Price Index

```
fit.asx=stl(BC.asx,t.window=15,s.window='periodic',robust=TRUE)
plot(fit.asx)
```

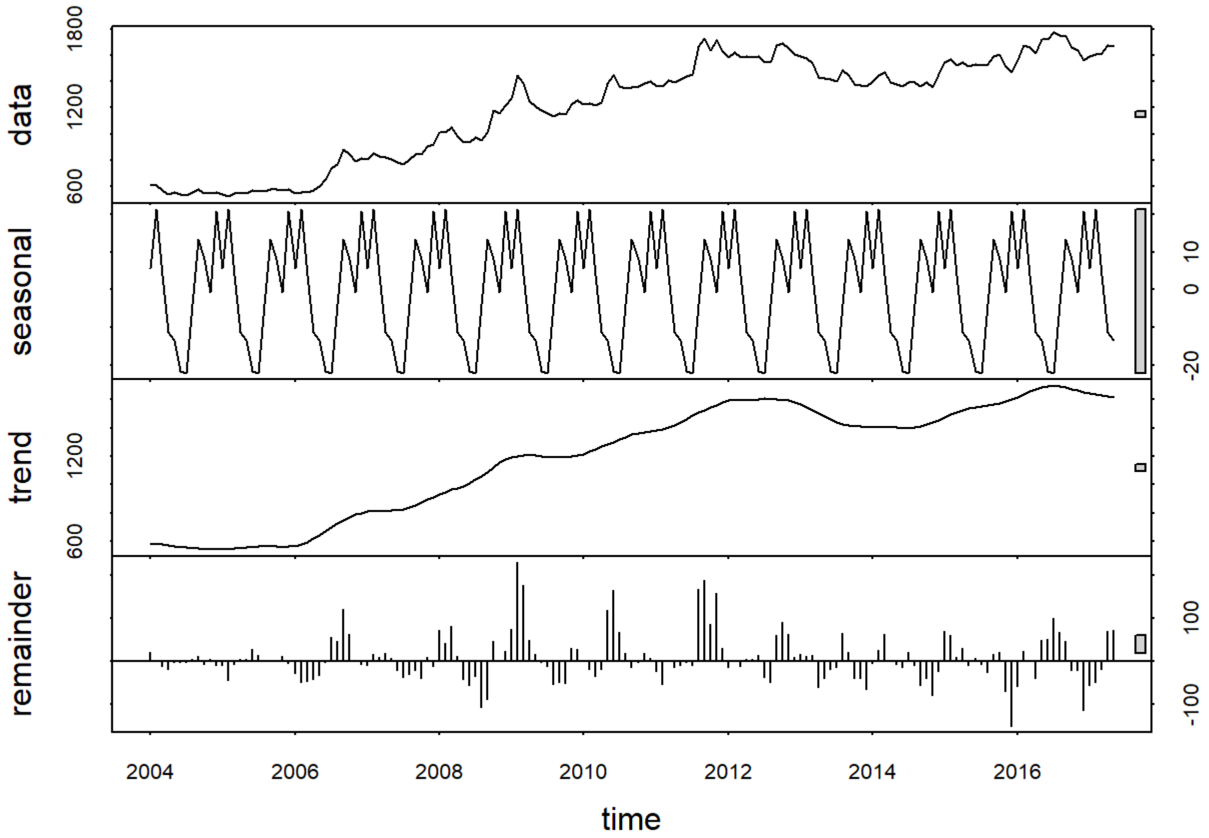


From the above plot, it can be inferrred that:

- The trend present in the time series is clearly visible in the trend component of the time series and it is a long-term upward trend which a significant decline during the year 2008.
- A significant change point occured around the year 2008 and it can be observed that both the trend component and the remainder component showed a significant peak.

STL Decomposition for Gold Price

```
fit.gold=stl(gold,t.window = 15,s.window = 'periodic',robust=TRUE)
plot(fit.gold)
```

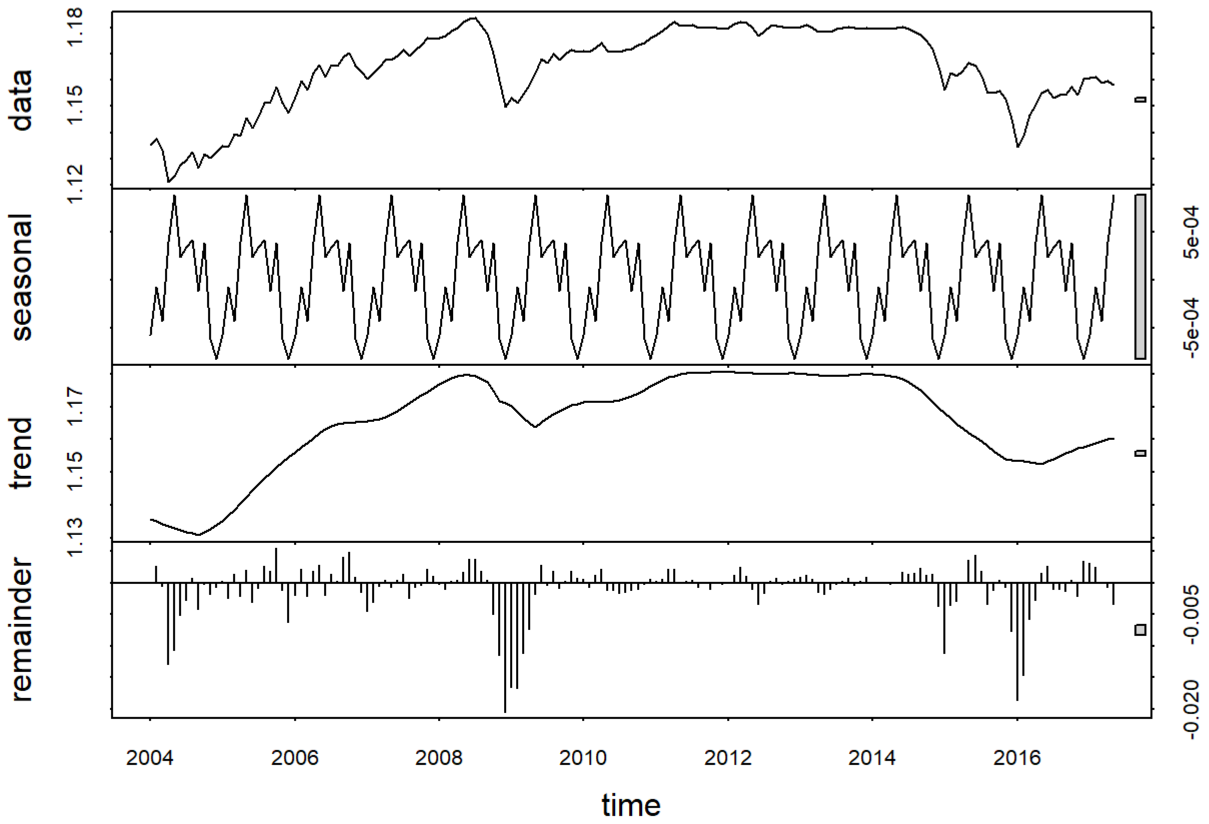


From the above plot it can be inferred that:

- Based on the trend component, it is clear that the series exhibits a long-term upward trend.

STL Decomposition for Crude Oil

```
fit.crude=stl(BC.crude,t.window = 15,s.window = 'periodic',robust=TRUE)
plot(fit.crude)
```

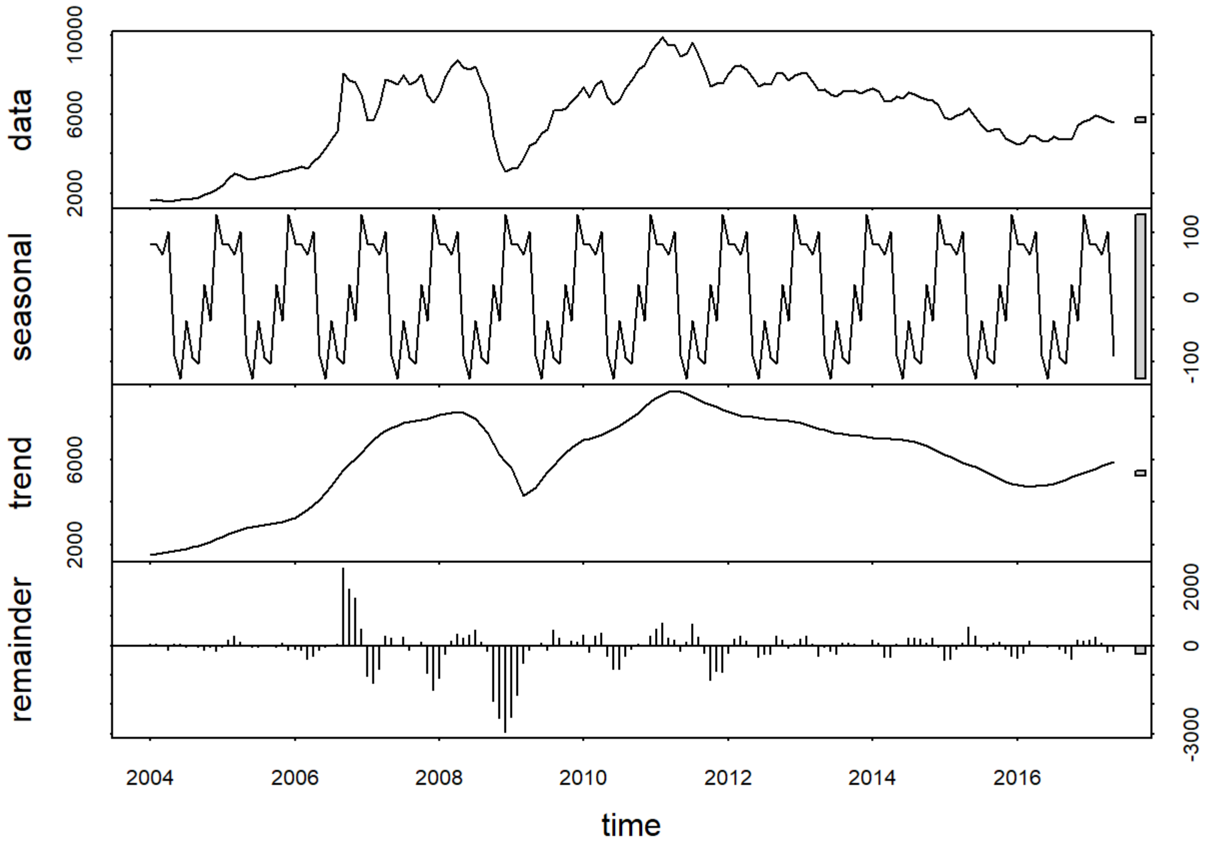


Based on the above plot it can be inferred that:

- According to the trend component, there is a long term non-linear movement in the series where there is an upward trend from 2004 to 2008 which is followed by a relative stability before a downward trend from 2014.
- Two significant change points can be observed during the year 2008 and 2016, where the trend component and the remainder component indicates large spikes.

STL Decomposition for Copper

```
fit.copper=stl(copper,t.window = 15,s.window = 'periodic',robust=TRUE)
plot(fit.copper)
```

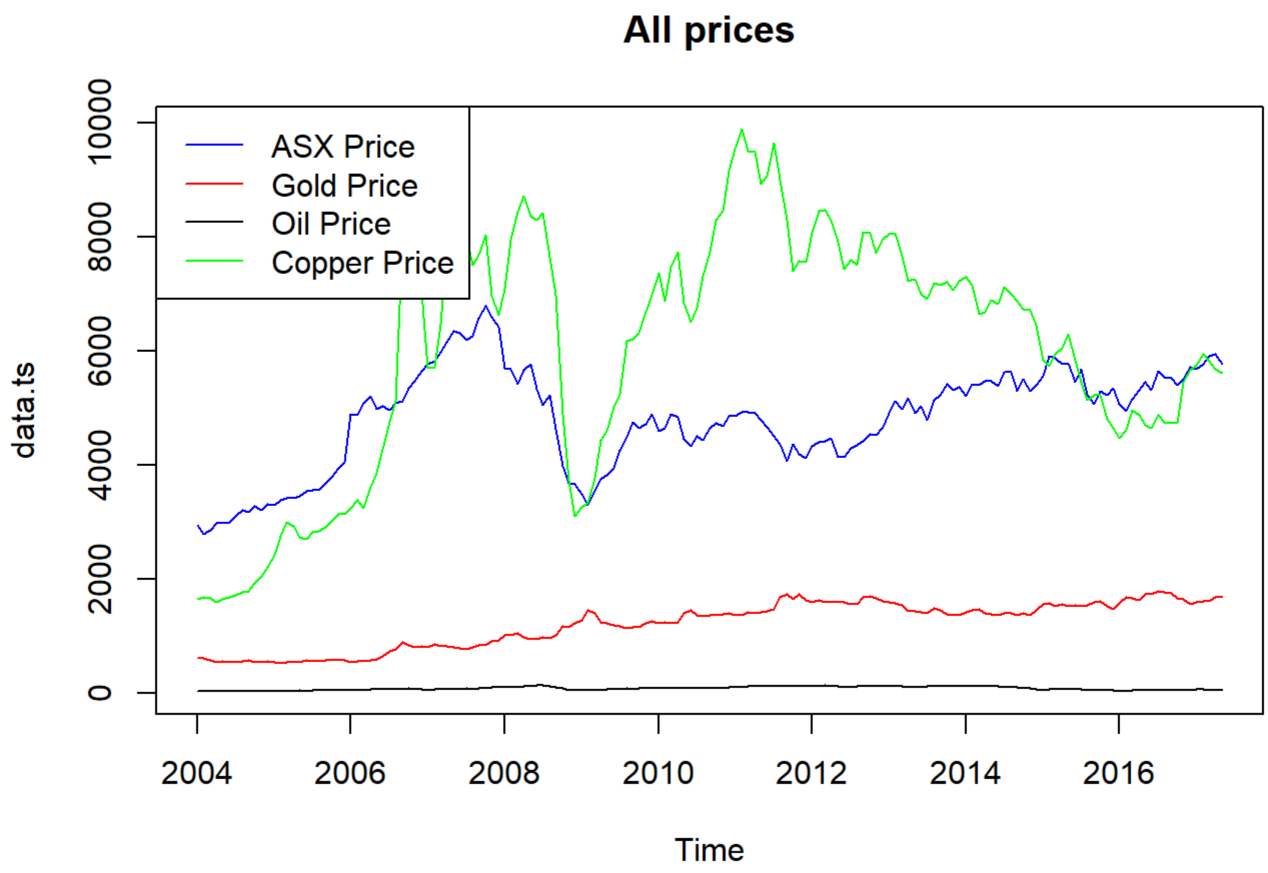


Based on the above plot, the following points can be inferred:

- The trend component captures the long-term trend which points out the sharp decline around the year 2009 which is followed by a recovery curve.
- A significant change point is observed during the year 2009 during which the trend component and the remainder component shows large spikes.

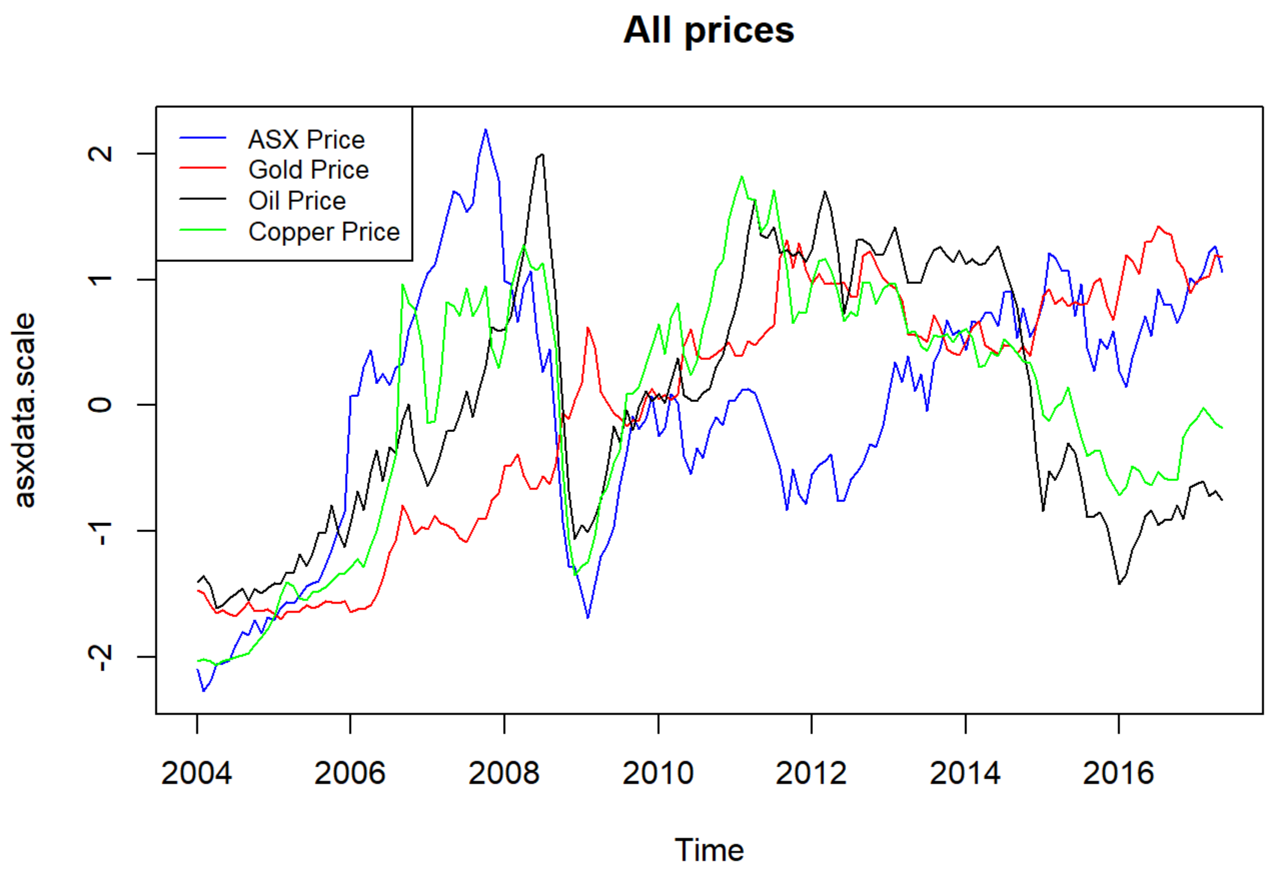
Modelling

```
plot(data.ts, plot.type="s", col=c("blue", "red", "black", "green"), main = "All prices")
legend('topleft',
       legend = c("ASX Price", "Gold Price", "Oil Price", "Copper Price"),
       col = c("blue", "red", "black", "green"),
       lty = 1,)
```



The components are to be scaled in the next chunk to draw correlation between them.

```
asxdata.scale = scale(data.ts)
plot(asxdata.scale, plot.type="s", col=c("blue", "red", "black", "green"), main = "All price
s")
legend("topleft",
       legend=c("ASX Price", "Gold Price", "Oil Price", "Copper Price"),
       col=c("blue", "red", "black", "green"),
       lty=1,
       cex=0.8)
```



From the above plot, it is clearly visible that the Copper Price and the Crude Price track each other closely which clearly indicates that those two are high correlated. In addition to that, the ASX Price Index and Copper also have a high degree of correlation between them. In order to confirm this, let's inspect the correlation matrix between the variables.

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

```
##                           ASX price Gold price Crude Oil (Brent)_USD/bbl
## ASX price                 1.0000000  0.3431908                 0.3290338
## Gold price                0.3431908  1.0000000                 0.4366382
## Crude Oil (Brent)_USD/bbl 0.3290338  0.4366382                 1.0000000
## Copper_USD/tonne          0.5617864  0.5364213                 0.8664296
##                           Copper_USD/tonne
## ASX price                        0.5617864
## Gold price                       0.5364213
## Crude Oil (Brent)_USD/bbl        0.8664296
## Copper_USD/tonne                 1.0000000
```

According to the above correlation matrix,it is confirmed that the correlation between Crude Oil Price and Copper Price is on the high positive side. It is also confirmed that there is a high correlation between ASX Price and Copper Price. We will proceed with the relation of each attribute with ASX Price Index so that we can find the most accurate and suitable distributed lag model for the ASX Price Index.

Modelling for ASX Price Index with Gold

DLM Model Fitting for ASX Price VS Gold Price

```
for (i in 1:6){
  GAmodel1.1=dlm(x=as.vector(gold),y=as.vector(asx),q=i)
  cat("q = ",i, "AIC = ", AIC(GAmodel1.1$model), "BIC = ",BIC(GAmodel1.1$model),"\n")
}
```

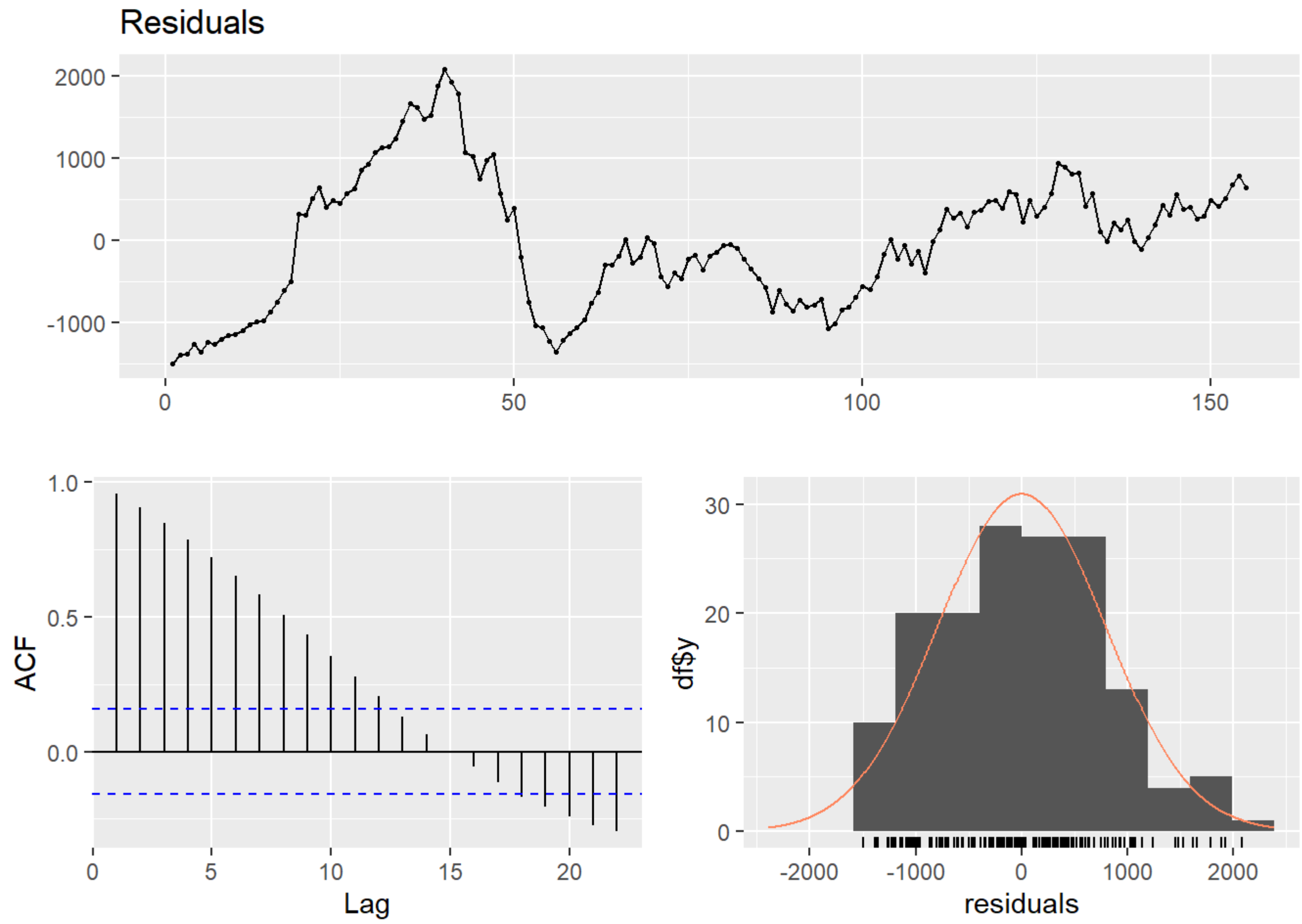
```
## q =  1 AIC =  2613.609 BIC =  2625.91
## q =  2 AIC =  2596.292 BIC =  2611.637
## q =  3 AIC =  2579.215 BIC =  2597.59
## q =  4 AIC =  2562.296 BIC =  2583.69
## q =  5 AIC =  2544.887 BIC =  2569.286
## q =  6 AIC =  2527.575 BIC =  2554.966
```

We will proceed with the q value of 6 as it exhibits the smallest AIC and BIC values.

```
GAmodel1=dlm(x=as.vector(gold),y=as.vector(asx),q=6)
summary(GAmodel1)
```

```
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1498.74  -617.99    -5.33   492.71  2084.31
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4250.19119  217.51364  19.540   <2e-16 ***
## x.t           -0.19930    1.26861  -0.157    0.875
## x.1           -0.04654    1.90563  -0.024    0.981
## x.2            0.07110    1.94292   0.037    0.971
## x.3           -0.05378    1.95473  -0.028    0.978
## x.4           -0.11744    1.94739  -0.060    0.952
## x.5            0.27967    1.91454   0.146    0.884
## x.6            0.61039    1.26350   0.483    0.630
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 814.8 on 147 degrees of freedom
## Multiple R-squared:  0.07473,    Adjusted R-squared:  0.03067
## F-statistic: 1.696 on 7 and 147 DF,  p-value: 0.1142
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2527.575 2554.966
```

```
checkresiduals(GAmodel1$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 143.09, df = 11, p-value < 2.2e-16
```

Polynomial DLM Model Fitting for ASX Price and Gold Price

```
q_val=1:5
k_val=1:3
for (i in q_val){
  for (j in k_val){
    if (j<=i){
      GAmodel2.2=polyDlm(x=as.vector(gold),y=as.vector(asx),q=i,k=j,show.beta=TRUE)
      cat("q = ",i,"k= ",j, "AIC = ", AIC(GAmodel2.2$model), "BIC = ",BIC(GAmodel2.2$mode
l),"\n")
    }
  }
}
```

```
## 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
## q =  1 k=  1 AIC =  2613.609 BIC =  2625.91
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0    0.162     0.8450   0.192 8.48e-01
## beta.1    0.232     0.0555   4.190 4.73e-05
## beta.2    0.302     0.8420   0.359 7.20e-01
## q =  2 k=  1 AIC =  2594.354 BIC =  2606.629
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    0.396       1.28   0.310   0.757
## beta.1   -0.233       1.90  -0.122   0.903
## beta.2    0.536       1.27   0.421   0.675
## q =  2 k=  2 AIC =  2596.292 BIC =  2611.637
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.0402      0.615  0.0654   0.948
## beta.1   0.1230      0.210  0.5880   0.557
## beta.2   0.2060      0.206  1.0000   0.319
## beta.3   0.2900      0.611  0.4740   0.636
## q =  3 k=  1 AIC =  2575.233 BIC =  2587.484
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1540      1.050  0.1470   0.883
## beta.1   0.0107      0.864  0.0124   0.990
## beta.2   0.0937      0.864  0.1080   0.914
## beta.3   0.4030      1.040  0.3870   0.700
## q =  3 k=  2 AIC =  2577.215 BIC =  2592.528
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error  t value P(>|t|)
## beta.0  0.15800       1.28 0.123000   0.903
## beta.1  0.00179       1.92 0.000929   0.999
## beta.2  0.10300       1.93 0.053200   0.958
## beta.3  0.39900       1.28 0.311000   0.756
## q =  3 k=  3 AIC =  2579.215 BIC =  2597.59
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0  -0.0665     0.4720  -0.141 0.888000
## beta.1   0.0290     0.2390   0.121 0.904000
## beta.2   0.1240     0.0333   3.740 0.000262
## beta.3   0.2200     0.2350   0.935 0.351000
## beta.4   0.3150     0.4690   0.673 0.502000
## q =  4 k=  1 AIC =  2556.384 BIC =  2568.609
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.0456      0.873  0.0522   0.958
## beta.1  -0.0262      0.433 -0.0604   0.952
## beta.2   0.0134      0.727  0.0185   0.985
## beta.3   0.1640      0.433  0.3790   0.705
## beta.4   0.4270      0.866  0.4930   0.623
## q =  4 k=  2 AIC =  2558.36 BIC =  2573.642
## Estimates and t-tests for beta coefficients:
```

```
##        Estimate Std. Error t value P(>|t|)
## beta.0  -0.0721      1.160 -0.0623   0.950
## beta.1   0.1600      1.270  0.1260   0.900
## beta.2   0.0134      0.729  0.0184   0.985
## beta.3  -0.0221      1.270 -0.0174   0.986
## beta.4   0.5440      1.150  0.4730   0.637
## q =  4 k=  3 AIC =  2560.335 BIC =  2578.673
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0  -0.1630     0.3780  -0.431  0.6670
## beta.1  -0.0589     0.2290  -0.257  0.7970
## beta.2   0.0451     0.0818   0.552  0.5820
## beta.3   0.1490     0.0781   1.910  0.0581
## beta.4   0.2530     0.2250   1.130  0.2620
## beta.5   0.3570     0.3740   0.955  0.3410
## q =  5 k=  1 AIC =  2537.143 BIC =  2549.342
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error  t value P(>|t|)
## beta.0  0.02830      0.732  0.03860   0.969
## beta.1 -0.09560      0.259 -0.36900   0.712
## beta.2 -0.10600      0.501 -0.21100   0.833
## beta.3 -0.00212      0.502 -0.00423   0.997
## beta.4  0.21500      0.257  0.83700   0.404
## beta.5  0.54700      0.725  0.75400   0.452
## q =  5 k=  2 AIC =  2539.047 BIC =  2554.296
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0  -0.1960      1.040 -0.1880   0.851
## beta.1   0.1510      0.852  0.1770   0.860
## beta.2   0.0400      0.695  0.0576   0.954
## beta.3  -0.1480      0.695 -0.2130   0.832
## beta.4  -0.0312      0.852 -0.0367   0.971
## beta.5   0.7700      1.030  0.7440   0.458
## q =  5 k=  3 AIC =  2540.952 BIC =  2559.251
```

The lowest AIC and BIC values are observed at q=5 and k=1, therefore I will proceed with those two values for Polynomial DLM modelling.

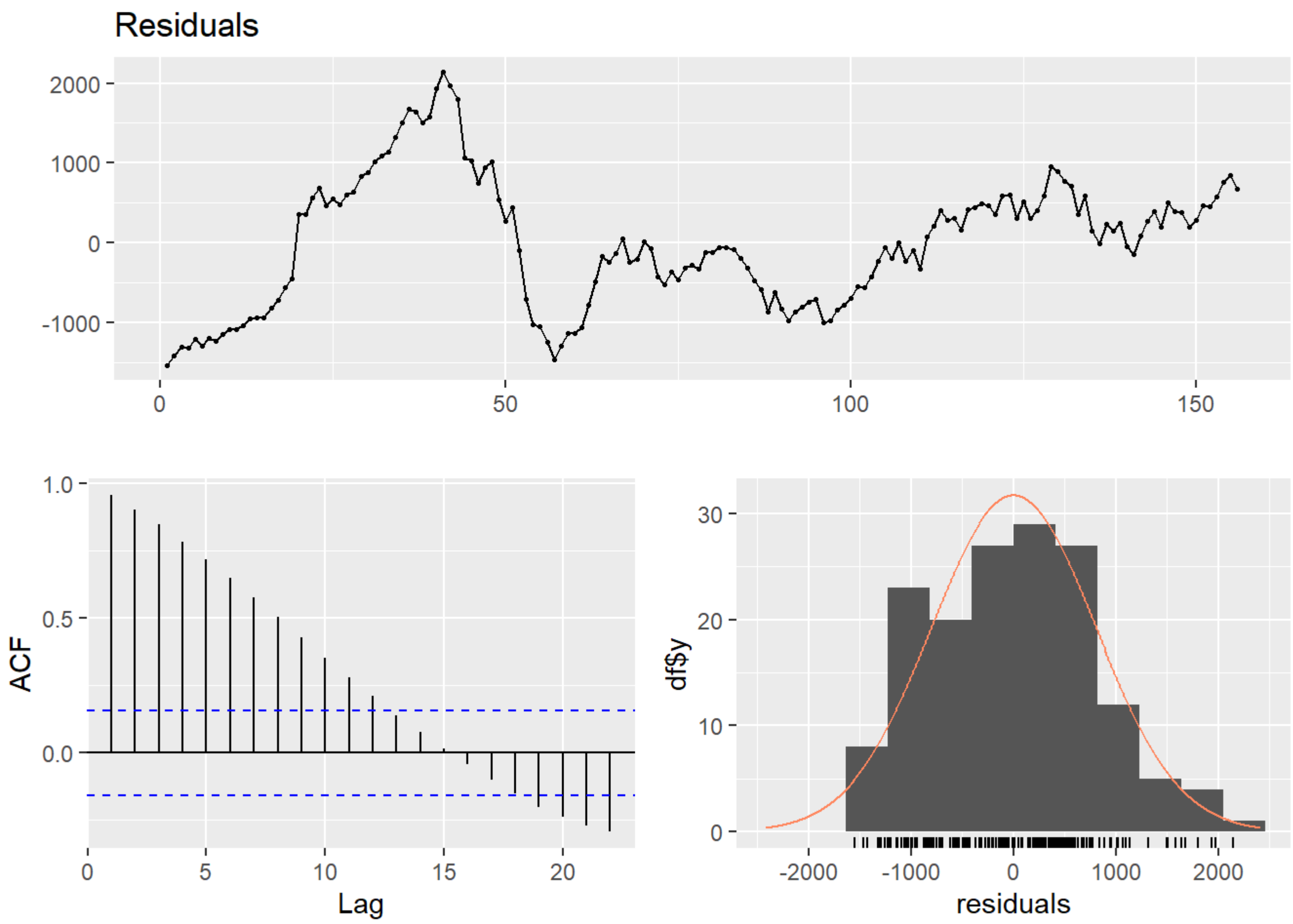
```
GAmodel2=polyDlm(x=as.vector(gold),y=as.vector(asx),q=5,k=1,show.beta=TRUE)
```

```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0  -0.1630     0.3780  -0.431  0.6670
## beta.1  -0.0589     0.2290  -0.257  0.7970
## beta.2   0.0451     0.0818   0.552  0.5820
## beta.3   0.1490     0.0781   1.910  0.0581
## beta.4   0.2530     0.2250   1.130  0.2620
## beta.5   0.3570     0.3740   0.955  0.3410
```

```
summary(GAmodel2)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1545.46  -637.78    -5.36   510.38  2144.26
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4184.8214   212.5718  19.687   <2e-16 ***
## z.t0          -0.1629     0.3781  -0.431    0.667
## z.t1           0.1040     0.1500   0.693    0.489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 810 on 153 degrees of freedom
## Multiple R-squared:  0.07926,    Adjusted R-squared:  0.06723
## F-statistic: 6.586 on 2 and 153 DF,  p-value: 0.001805
```

```
checkresiduals(GAmodel2$model)
```



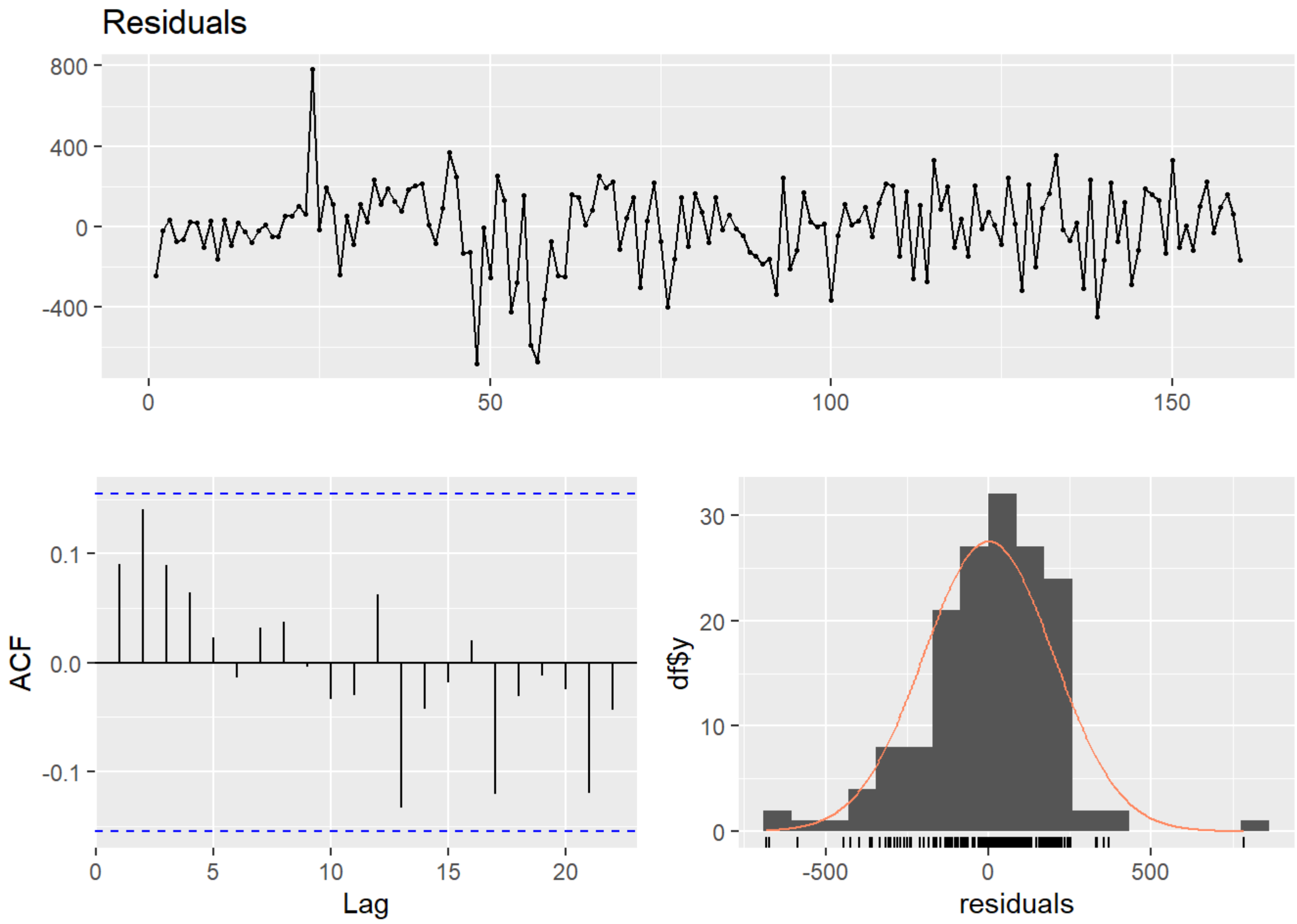

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 143.27, df = 10, p-value < 2.2e-16
```

KoyckDLM Model Fitting for ASX Price and Gold Price

```
GAmodel3 = koyckDlm(x=as.vector(gold), y=as.vector(asx))
summary(GAmodel3,diagnostics=TRUE)
```

```
##
## 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:
##                  df1 df2  statistic       p-value
## Weak instruments   1 157 8006.63657 1.266826e-136
## Wu-Hausman         1 156   18.06854  3.655601e-05
##
##                            alpha        beta       phi
## Geometric coefficients:  5205.15 0.002595168 0.9634602
```

```
checkresiduals(GAmodel3$model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 7.3793, df = 10, p-value = 0.6892
##
## Model df: 0.   Total lags used: 10
```

AutoRegressive DLM Model Fitting for ASX Price and Gold Price

```
for(i in 1:5){
  for(j in 1:5){
    GAmodel4.1 = ardlDlm(x = as.vector(gold), y = as.vector(asx), p = i, q = j)
    cat("p = ", i, "q = ", j, "AIC = ", AIC(GAmodel4.1$model), "BIC = ", BIC(GAmodel4.1$mode
l), "\n")
  }
}
```

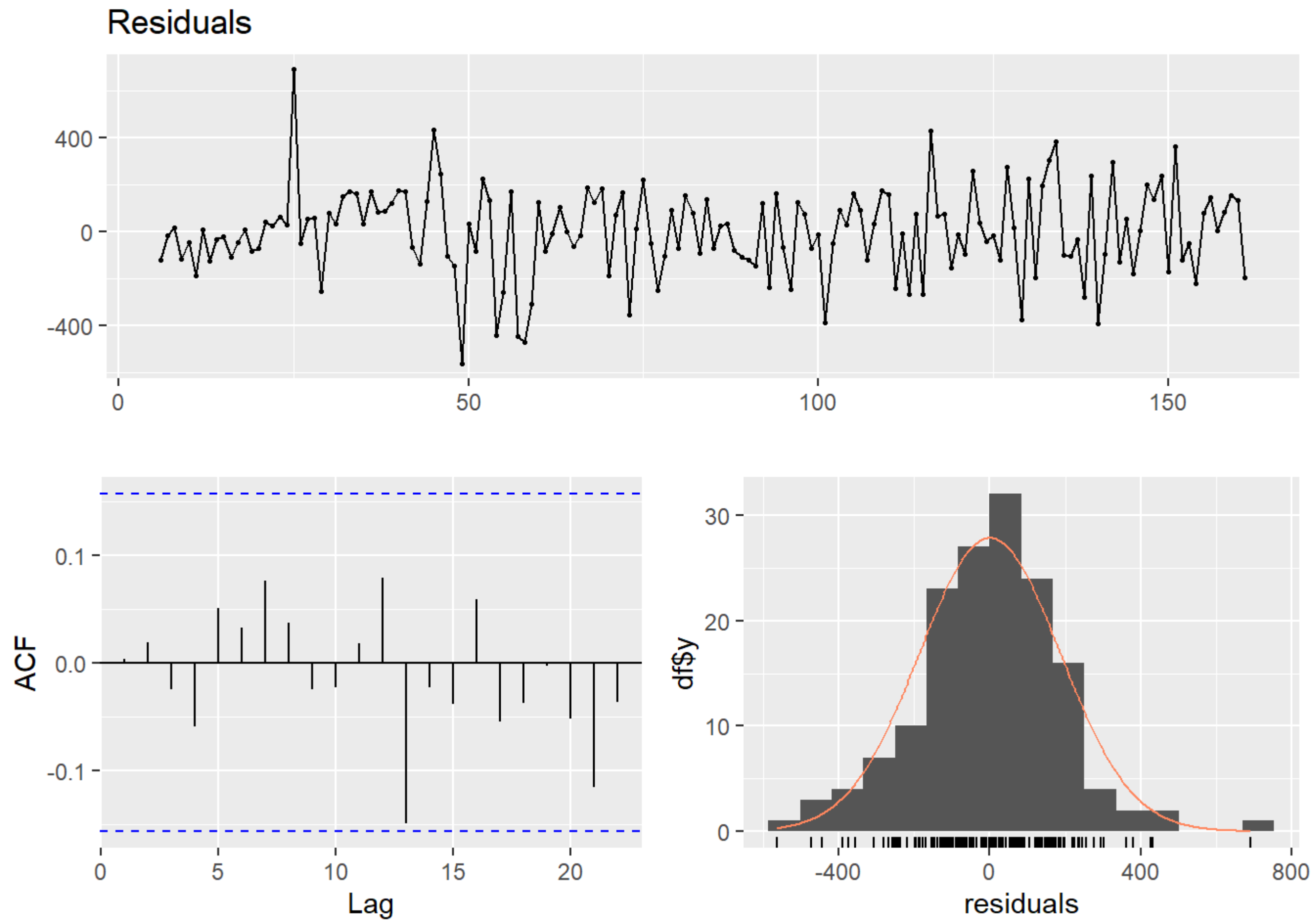
```
## p =  1 q =  1 AIC =  2140.897 BIC =  2156.273
## p =  1 q =  2 AIC =  2128.524 BIC =  2146.938
## p =  1 q =  3 AIC =  2113.99 BIC =  2135.428
## p =  1 q =  4 AIC =  2102.754 BIC =  2127.204
## p =  1 q =  5 AIC =  2092.194 BIC =  2119.643
## p =  2 q =  1 AIC =  2128.627 BIC =  2147.04
## p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

The lowest AIC and BIC values are observed at p=1 and q=5, therefore I will proceed with those two values for Autoregressive DLM modelling.

```
GAmodel4 = ardlDlm(x = as.vector(gold), y = as.vector(asx), p = 1, q = 5)
summary(GAmodel4)
```

```
##
## Time series regression with "ts" data:
## Start = 6, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data, start = 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -564.02 -106.74    8.99  126.34  691.74
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 234.72597   92.97346   2.525 0.012635 *
## X.t          -1.21331    0.30480  -3.981 0.000107 ***
## X.1           1.21097    0.30013   4.035 8.73e-05 ***
## Y.1           0.96620    0.07927  12.189  < 2e-16 ***
## Y.2           0.13687    0.11316   1.210 0.228368
## Y.3          -0.07572    0.11193  -0.676 0.499816
## Y.4          -0.04931    0.11174  -0.441 0.659640
## Y.5          -0.02130    0.07871  -0.271 0.787092
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191.6 on 148 degrees of freedom
## Multiple R-squared:  0.9502, Adjusted R-squared:  0.9478
## F-statistic: 403.1 on 7 and 148 DF,  p-value: < 2.2e-16
```

```
checkresiduals(GAmodel4$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 5.7145, df = 11, p-value = 0.8917
```

Based on the above results after modelling, it can be concluded that the best distributed lag model for ASX Price Index and Gold Price is the AutoRegressive Lag model i.e. GAmodel4 which has p=1 and q=5 as it has lowest AIC and BIC values among all i.e. AIC=2092.194 and BIC=2119.643. In addition to that it has a very high adjusted R-square value as well along with no significant lags in the ACF plot for the residuals.

Modelling for ASX Price Index with Crude Oil

DLM Model Fitting for ASX Price VS Crude Oil

```
#CA--> CrudeOil and ASX
for (i in 1:6){
  CAmodel1.1=dlm(x=as.vector(crude_oil),y=as.vector(asx),q=i)
  cat("q = ",i, "AIC = ", AIC(CAmodel1.1$model), "BIC = ",BIC(CAmodel1.1$model),"\n")
}
```

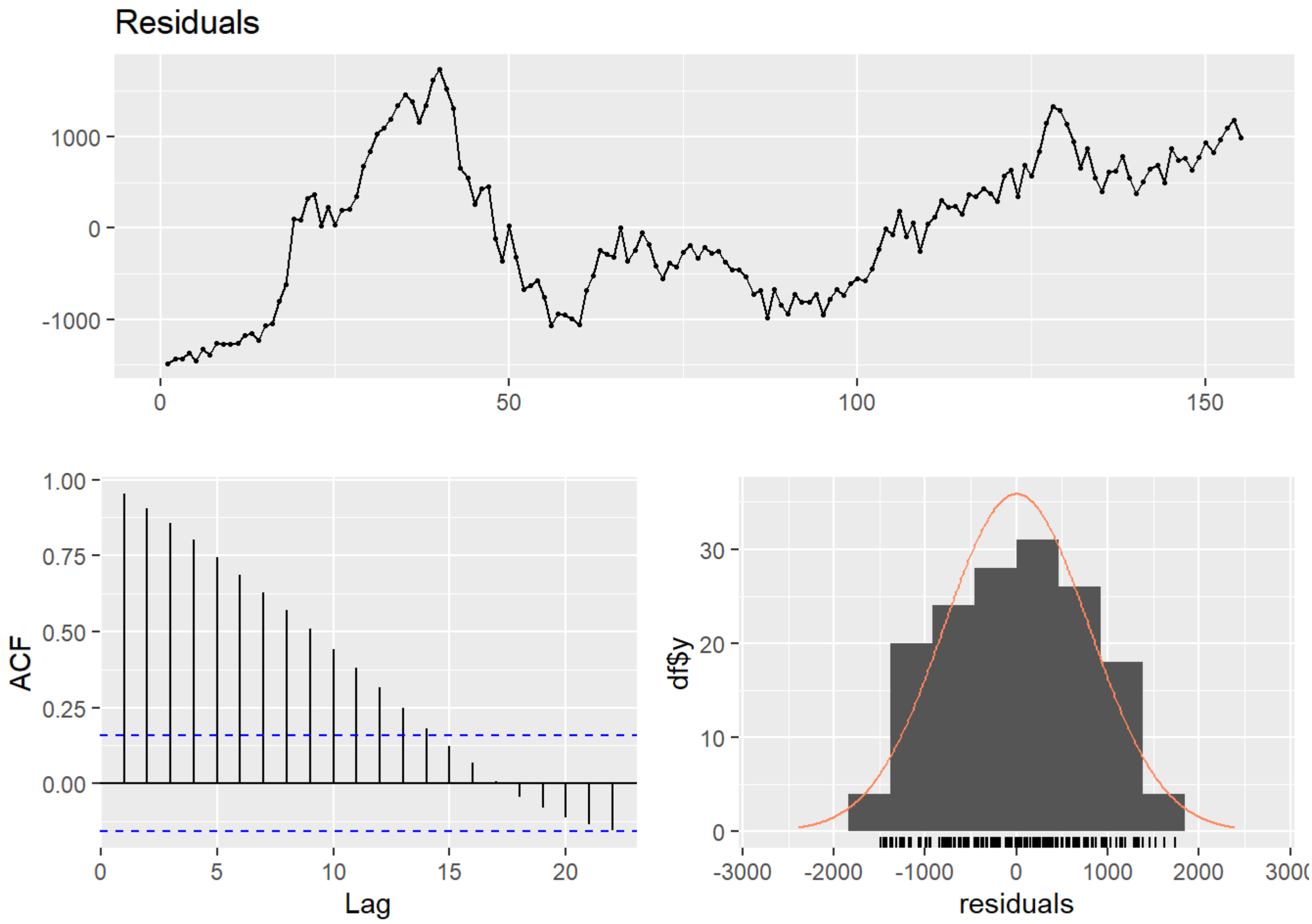
```
## q =  1 AIC =  2614.698 BIC =  2626.998
## q =  2 AIC =  2596.715 BIC =  2612.059
## q =  3 AIC =  2579.101 BIC =  2597.477
## q =  4 AIC =  2561.888 BIC =  2583.281
## q =  5 AIC =  2544.936 BIC =  2569.335
## q =  6 AIC =  2527.701 BIC =  2555.091
```

We will proceed with the q value of 6 as it exhibits the smallest AIC and BIC values.

```
CAmodel1=dlm(x=as.vector(crude_oil),y=as.vector(asx),q=6)
summary(CAmodel1)
```

```
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1489.01  -649.84    20.52   631.35  1742.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4465.209    189.119  23.611   <2e-16 ***
## x.t           11.039     11.476   0.962    0.338
## x.1            1.035     19.234   0.054    0.957
## x.2           -3.096     19.421  -0.159    0.874
## x.3            4.826     19.469   0.248    0.805
## x.4           -5.191     19.433  -0.267    0.790
## x.5            1.270     19.231   0.066    0.947
## x.6           -4.356     11.386  -0.383    0.703
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 815.1 on 147 degrees of freedom
## Multiple R-squared:  0.07398,    Adjusted R-squared:  0.02988
## F-statistic: 1.678 on 7 and 147 DF,  p-value: 0.1187
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2527.701 2555.091
```

```
checkresiduals(CAmodel1$model)
```




```
##
## 	Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 142.79, df = 11, p-value < 2.2e-16
```

Polynomial DLM Model Fitting for ASX Price and Crude Oil

```
q_val=1:5
k_val=1:3
for (i in q_val){
  for (j in k_val){
    if (j<=i){
      CAmodel2.2=polyDlm(x=as.vector(crude_oil),y=as.vector(asx),q=i,k=j,show.beta=TRUE)
      cat("q = ",i,"k= ",j, "AIC = ", AIC(CAmodel2.2$model), "BIC = ",BIC(CAmodel2.2$mode
l),"\n")
    }
  }
}
```

```
## 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
## q =  1 k=  1 AIC =  2614.698 BIC =  2626.998
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0    13.90      6.310    2.20 0.029300
## beta.1     2.82      0.744    3.79 0.000215
## beta.2    -8.23      6.270   -1.31 0.191000
## q =  2 k=  1 AIC =  2594.715 BIC =  2606.991
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    13.70       11.6   1.180   0.239
## beta.1     3.18       19.1   0.167   0.868
## beta.2    -8.41       11.5  -0.730   0.467
## q =  2 k=  2 AIC =  2596.715 BIC =  2612.059
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    10.80       4.17   2.580 0.01090
## beta.1     4.86       1.50   3.250 0.00142
## beta.2    -1.03       1.47  -0.699 0.48500
## beta.3    -6.92       4.14  -1.670 0.09680
## q =  3 k=  1 AIC =  2575.112 BIC =  2587.362
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0    11.20       9.40   1.190   0.236
## beta.1     4.44       8.33   0.533   0.595
## beta.2    -1.45       8.34  -0.174   0.862
## beta.3    -6.48       9.35  -0.693   0.489
## q =  3 k=  2 AIC =  2577.109 BIC =  2592.422
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error  t value P(>|t|)
## beta.0    11.80       11.5  1.02000   0.309
## beta.1     2.90       19.3  0.15000   0.881
## beta.2     0.09       19.3  0.00466   0.996
## beta.3    -7.07       11.5 -0.61500   0.539
## q =  3 k=  3 AIC =  2579.101 BIC =  2597.477
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0     8.59      3.020    2.84 0.00505
## beta.1     4.98      1.570    3.18 0.00179
## beta.2     1.38      0.449    3.06 0.00259
## beta.3    -2.23      1.540   -1.45 0.14800
## beta.4    -5.84      2.990   -1.96 0.05230
## q =  4 k=  1 AIC =  2556.008 BIC =  2568.233
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0     8.67       7.22   1.200   0.232
## beta.1     4.95       3.48   1.420   0.158
## beta.2     1.30       6.35   0.205   0.838
## beta.3    -2.27       3.49  -0.649   0.517
## beta.4    -5.77       7.16  -0.806   0.421
## q =  4 k=  2 AIC =  2558.007 BIC =  2573.289
## Estimates and t-tests for beta coefficients:
```

```
##        Estimate Std. Error t value P(>|t|)
## beta.0   11.100      10.70  1.0400   0.300
## beta.1    0.816      13.80  0.0592   0.953
## beta.2    1.290       6.37  0.2030   0.839
## beta.3    1.860      13.80  0.1350   0.893
## beta.4   -8.170      10.60 -0.7740   0.440
## q =  4 k=  3 AIC =  2559.908 BIC =  2578.246
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0    6.850      2.330   2.940 0.003820
## beta.1    4.510      1.440   3.140 0.002030
## beta.2    2.180      0.604   3.600 0.000428
## beta.3   -0.159      0.579  -0.275 0.784000
## beta.4   -2.490      1.400  -1.780 0.077700
## beta.5   -4.830      2.300  -2.100 0.037200
## q =  5 k=  1 AIC =  2537.158 BIC =  2549.357
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0     8.28       5.56   1.490   0.139
## beta.1     4.26       1.69   2.510   0.013
## beta.2     1.08       3.91   0.275   0.783
## beta.3    -1.26       3.92  -0.322   0.748
## beta.4    -2.76       1.68  -1.640   0.104
## beta.5    -3.41       5.51  -0.619   0.537
## q =  5 k=  2 AIC =  2539.075 BIC =  2554.324
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error  t value P(>|t|)
## beta.0  10.1000       9.45  1.07000   0.286
## beta.1   2.1500       8.91  0.24200   0.809
## beta.2  -0.1620       6.47 -0.02500   0.980
## beta.3  -0.0286       6.45 -0.00443   0.996
## beta.4  -0.6490       8.91 -0.07290   0.942
## beta.5  -5.2300       9.35 -0.55900   0.577
## q =  5 k=  3 AIC =  2541.015 BIC =  2559.314
```

The lowest AIC and BIC values are observed at q=5 and k=1, therefore I will proceed with those two values for Polynomial DLM modelling.

```
CAmodel2=polyDlm(x=as.vector(crude_oil),y=as.vector(asx),q=5,k=1,show.beta=TRUE)
```

```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0    6.850      2.330   2.940 0.003820
## beta.1    4.510      1.440   3.140 0.002030
## beta.2    2.180      0.604   3.600 0.000428
## beta.3   -0.159      0.579  -0.275 0.784000
## beta.4   -2.490      1.400  -1.780 0.077700
## beta.5   -4.830      2.300  -2.100 0.037200
```

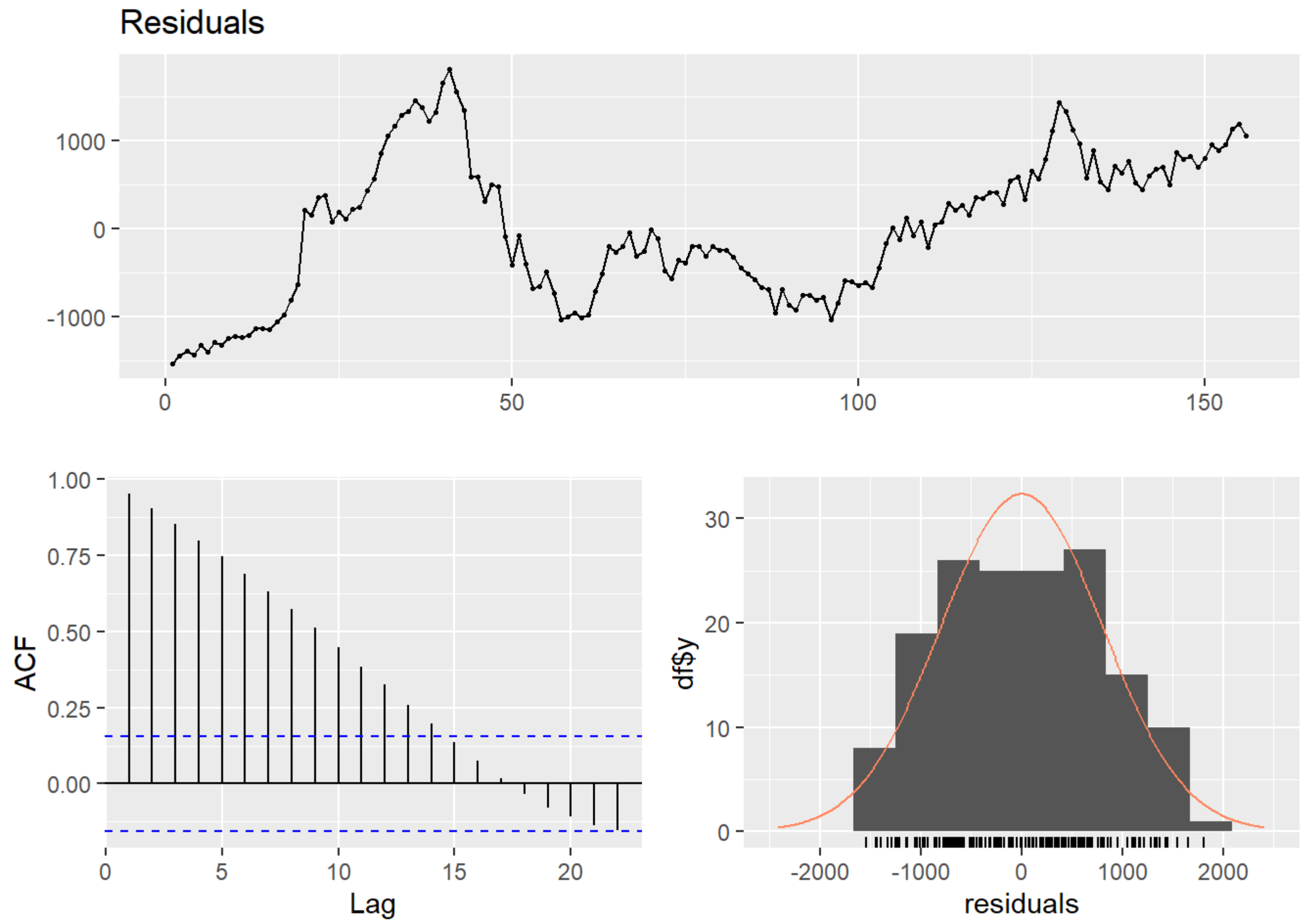
```
summary(CAmodel2)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1538.77  -659.37    -3.62   589.53  1808.12
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4416.2938   180.6143  24.452  < 2e-16 ***
## z.t0           6.8487     2.3310   2.938  0.00381 **
## z.t1          -2.3359     0.9135  -2.557  0.01153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 810 on 153 degrees of freedom
## Multiple R-squared:  0.07917,    Adjusted R-squared:  0.06714
## F-statistic: 6.578 on 2 and 153 DF,  p-value: 0.001818
```

```
summary(CAmodel2)
```


```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1538.77  -659.37    -3.62   589.53  1808.12
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4416.2938   180.6143  24.452  < 2e-16 ***
## z.t0           6.8487     2.3310   2.938  0.00381 **
## z.t1          -2.3359     0.9135  -2.557  0.01153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 810 on 153 degrees of freedom
## Multiple R-squared:  0.07917,    Adjusted R-squared:  0.06714
## F-statistic: 6.578 on 2 and 153 DF,  p-value: 0.001818
```


```
checkresiduals(CAmodel2$model)
```



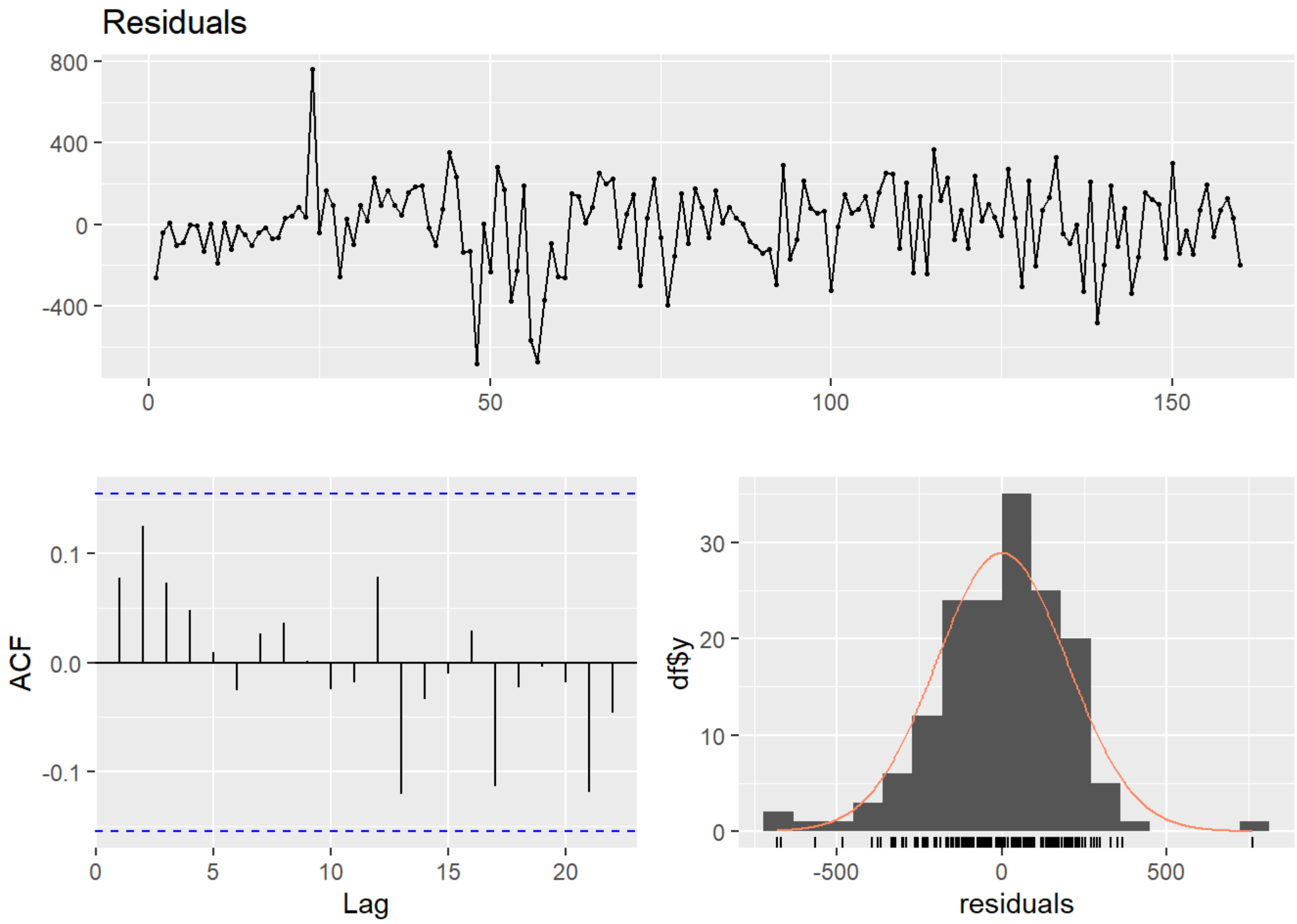

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 143.45, df = 10, p-value < 2.2e-16
```

KoyckDLM Model Fitting for ASX Price and Crude Oil

```
CAmodel3 = koyckDlm(x=as.vector(crude_oil), y=as.vector(asx))
summary(CAmodel3,diagnostics=TRUE)
```

```
##
## 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:
##                  df1 df2 statistic       p-value
## Weak instruments   1 157 3018.4519 1.997814e-104
## Wu-Hausman         1 156   11.1979  1.026113e-03
##
##                            alpha       beta       phi
## Geometric coefficients:  8522.034 -0.9990694 0.9753703
```

```
checkresiduals(CAmodel3$model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 5.4432, df = 10, p-value = 0.8597
##
## Model df: 0.   Total lags used: 10
```

AutoRegressive DLM Model Fitting for ASX Price and Crude Oil

```
for(i in 1:5){
  for(j in 1:5){
    CAmodel4.1 = ardlDlm(x = as.vector(crude_oil), y = as.vector(asx), p = i, q = j)
    cat("p = ", i, "q = ", j, "AIC = ", AIC(CAmodel4.1$model), "BIC = ", BIC(CAmodel4.1$mode
l), "\n")
  }
}
```

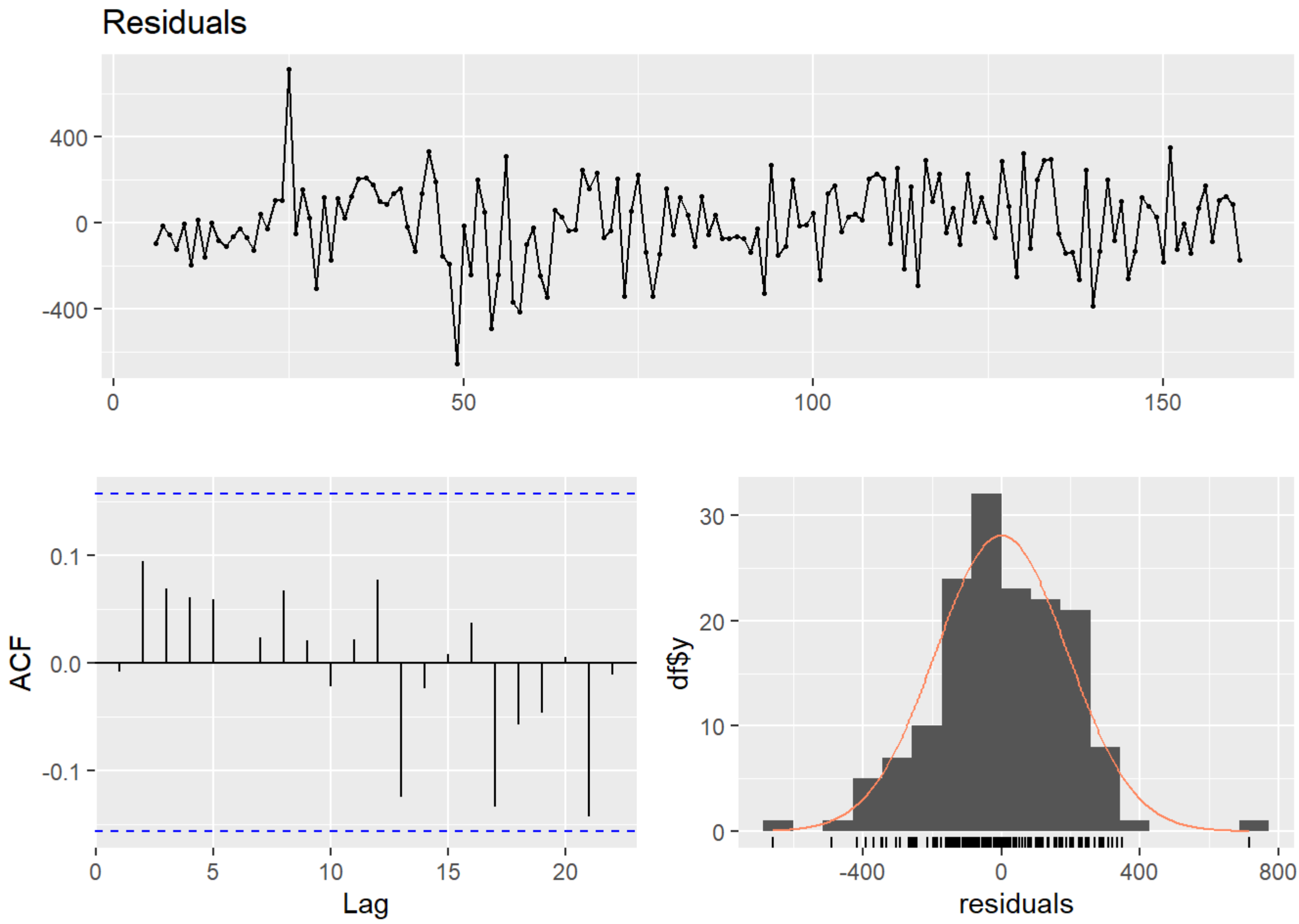
```
## p =  1 q =  1 AIC =  2146.524 BIC =  2161.9
## p =  1 q =  2 AIC =  2134.107 BIC =  2152.521
## p =  1 q =  3 AIC =  2121.07 BIC =  2142.508
## p =  1 q =  4 AIC =  2109.4 BIC =  2133.85
## p =  1 q =  5 AIC =  2098.335 BIC =  2125.784
## p =  2 q =  1 AIC =  2132.312 BIC =  2150.726
## p =  2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

The lowest AIC and BIC values are observed at p=5 and q=1, therefore I will proceed with those two values for Autoregressive DLM modelling.

```
CAmodel4 = ardlDlm(x = as.vector(crude_oil), y = as.vector(asx), p = 5, q = 1)
summary(CAmodel4)
```

```
##
## Time series regression with "ts" data:
## Start = 6, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data, start = 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -657.99 -121.05   -4.38  123.84  714.56
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 239.54102   94.36245   2.539   0.0122 *
## X.t           5.68288    2.73874   2.075   0.0397 *
## X.1          -3.32354    4.57294  -0.727   0.4685
## X.2          -3.29907    4.62372  -0.714   0.4767
## X.3           3.05569    4.63039   0.660   0.5103
## X.4          -8.00745    4.57207  -1.751   0.0820 .
## X.5           5.08983    2.72010   1.871   0.0633 .
## Y.1           0.96648    0.01934  49.983   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 194.6 on 148 degrees of freedom
## Multiple R-squared:  0.9486, Adjusted R-squared:  0.9461
## F-statistic:   390 on 7 and 148 DF,  p-value: < 2.2e-16
```

```
checkresiduals(CAmodel4$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 5.18, df = 11, p-value = 0.9221
```

Based on the above results after modelling, it can be concluded that the best distributed lag model for ASX Price Index and Crude Oil is the AutoRegressive Lag model i.e. CAmodel4 which has p=5 and q=1 as it has lowest AIC and BIC values among all i.e. AIC=2097.076 and BIC=2124.525. In addition to that it has a very high adjusted R-square value as well along with no significant lags in the ACF plot for the residuals

Modelling for ASX Price Index with Copper Price

DLM Model Fitting for ASX Price VS Copper

```
for (i in 1:6){
  CuAmodel1.1=dlm(x=as.vector(copper),y=as.vector(asx),q=i)
  cat("q = ",i, "AIC = ", AIC(CuAmodel1.1$model), "BIC = ",BIC(CuAmodel1.1$model),"\n")
}
```

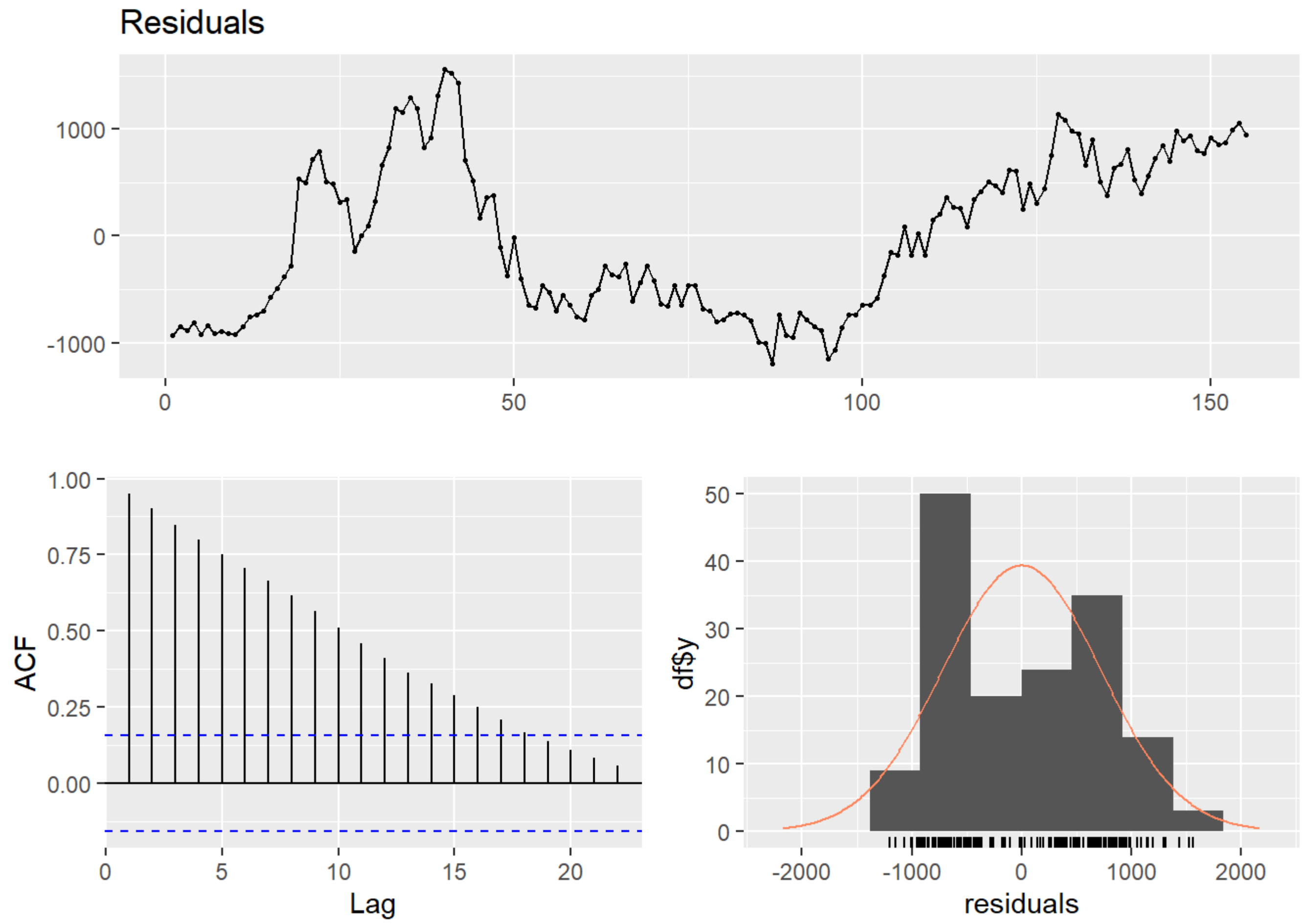
```
## q =  1 AIC =  2574.488 BIC =  2586.789
## q =  2 AIC =  2559.356 BIC =  2574.7
## q =  3 AIC =  2544.155 BIC =  2562.531
## q =  4 AIC =  2528.895 BIC =  2550.289
## q =  5 AIC =  2513.265 BIC =  2537.664
## q =  6 AIC =  2497.775 BIC =  2525.166
```

We will proceed with the q value of 6 as it exhibits the smallest AIC and BIC values.

```
CuAmodel1=dlm(x=as.vector(copper),y=as.vector(asx),q=6)
summary(CuAmodel1)
```

```
##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1200.0  -691.5  -106.3   631.6  1564.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.722e+03  1.980e+02  18.799   <2e-16 ***
## x.t          1.582e-01  1.325e-01   1.195    0.234
## x.1          4.879e-02  2.127e-01   0.229    0.819
## x.2          1.864e-02  2.128e-01   0.088    0.930
## x.3          3.026e-02  2.129e-01   0.142    0.887
## x.4          2.216e-02  2.130e-01   0.104    0.917
## x.5          2.903e-03  2.127e-01   0.014    0.989
## x.6         -9.299e-02  1.304e-01  -0.713    0.477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 740.1 on 147 degrees of freedom
## Multiple R-squared:  0.2366, Adjusted R-squared:  0.2002
## F-statistic: 6.507 on 7 and 147 DF,  p-value: 1.086e-06
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2497.775 2525.166
```

```
checkresiduals(CuAmodel1$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 142, df = 11, p-value < 2.2e-16
```

Polynomial DLM Model Fitting for ASX Price VS Copper

```
q_val=1:5
k_val=1:3
for (i in q_val){
  for (j in k_val){
    if (j<=i){
      CuAmodel2.2=polyDlm(x=as.vector(copper),y=as.vector(asx),q=i,k=j,show.beta=TRUE)
      cat("q = ",i,"k= ",j, "AIC = ", AIC(CuAmodel2.2$model), "BIC = ",BIC(CuAmodel2.2$mode
l),"\n")
    }
  }
}
```

```
## 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
## q =  1 k=  1 AIC =  2574.488 BIC =  2586.789
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0   0.1670    0.07680   2.170 3.14e-02
## beta.1   0.0746    0.00943   7.910 4.27e-13
## beta.2  -0.0175    0.07540  -0.232 8.17e-01
## q =  2 k=  1 AIC =  2557.367 BIC =  2569.643
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0  0.17800      0.131  1.3600   0.176
## beta.1  0.05290      0.207  0.2550   0.799
## beta.2 -0.00654      0.129 -0.0507   0.960
## q =  2 k=  2 AIC =  2559.356 BIC =  2574.7
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0   0.1500     0.0525   2.860 4.88e-03
## beta.1   0.0859     0.0191   4.490 1.37e-05
## beta.2   0.0219     0.0180   1.210 2.26e-01
## beta.3  -0.0422     0.0513  -0.822 4.12e-01
## q =  3 k=  1 AIC =  2540.208 BIC =  2552.459
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1510     0.1070   1.410   0.161
## beta.1   0.0848     0.0924   0.918   0.360
## beta.2   0.0207     0.0926   0.224   0.823
## beta.3  -0.0410     0.1060  -0.389   0.698
## q =  3 k=  2 AIC =  2542.208 BIC =  2557.521
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1680      0.131   1.290   0.201
## beta.1   0.0419      0.210   0.199   0.842
## beta.2   0.0636      0.210   0.302   0.763
## beta.3  -0.0578      0.129  -0.448   0.655
## q =  3 k=  3 AIC =  2544.155 BIC =  2562.531
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0  0.13600     0.0391   3.480 6.55e-04
## beta.1  0.08860     0.0204   4.340 2.59e-05
## beta.2  0.04140     0.0058   7.130 3.67e-11
## beta.3 -0.00586     0.0193  -0.303 7.63e-01
## beta.4 -0.05310     0.0379  -1.400 1.64e-01
## q =  4 k=  1 AIC =  2523.1 BIC =  2535.325
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1300     0.0841  1.5500  0.1240
## beta.1   0.0913     0.0406  2.2500  0.0259
## beta.2   0.0469     0.0720  0.6520  0.5150
## beta.3  -0.0031     0.0406 -0.0764  0.9390
## beta.4  -0.0587     0.0825 -0.7120  0.4770
## q =  4 k=  2 AIC =  2525.094 BIC =  2540.375
## Estimates and t-tests for beta coefficients:
```

```
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1680     0.1220   1.390   0.168
## beta.1   0.0285     0.1490   0.190   0.849
## beta.2   0.0468     0.0722   0.649   0.517
## beta.3   0.0598     0.1490   0.400   0.690
## beta.4  -0.0965     0.1200  -0.807   0.421
## q =  4 k=  3 AIC =  2526.896 BIC =  2545.234
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0   0.1230    0.03080    4.00 9.86e-05
## beta.1   0.0871    0.01910    4.56 1.05e-05
## beta.2   0.0510    0.00814    6.26 3.72e-09
## beta.3   0.0148    0.00730    2.02 4.47e-02
## beta.4  -0.0214    0.01810   -1.18 2.38e-01
## beta.5  -0.0576    0.02980   -1.94 5.49e-02
## q =  5 k=  1 AIC =  2505.646 BIC =  2517.846
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0   0.1200     0.0664   1.810 7.17e-02
## beta.1   0.0877     0.0217   4.040 8.62e-05
## beta.2   0.0532     0.0455   1.170 2.44e-01
## beta.3   0.0170     0.0456   0.373 7.10e-01
## beta.4  -0.0209     0.0212  -0.986 3.26e-01
## beta.5  -0.0605     0.0648  -0.932 3.53e-01
## q =  5 k=  2 AIC =  2507.644 BIC =  2522.893
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value P(>|t|)
## beta.0   0.1720     0.1090   1.580   0.116
## beta.1   0.0305     0.0975   0.313   0.755
## beta.2   0.0193     0.0724   0.267   0.790
## beta.3   0.0507     0.0723   0.701   0.484
## beta.4   0.0365     0.0976   0.374   0.709
## beta.5  -0.1120     0.1070  -1.040   0.299
## q =  5 k=  3 AIC =  2509.27 BIC =  2527.57
```

The lowest AIC and BIC values are observed at q=5 and k=1, therefore I will proceed with those two values for Polynomial DLM modelling.

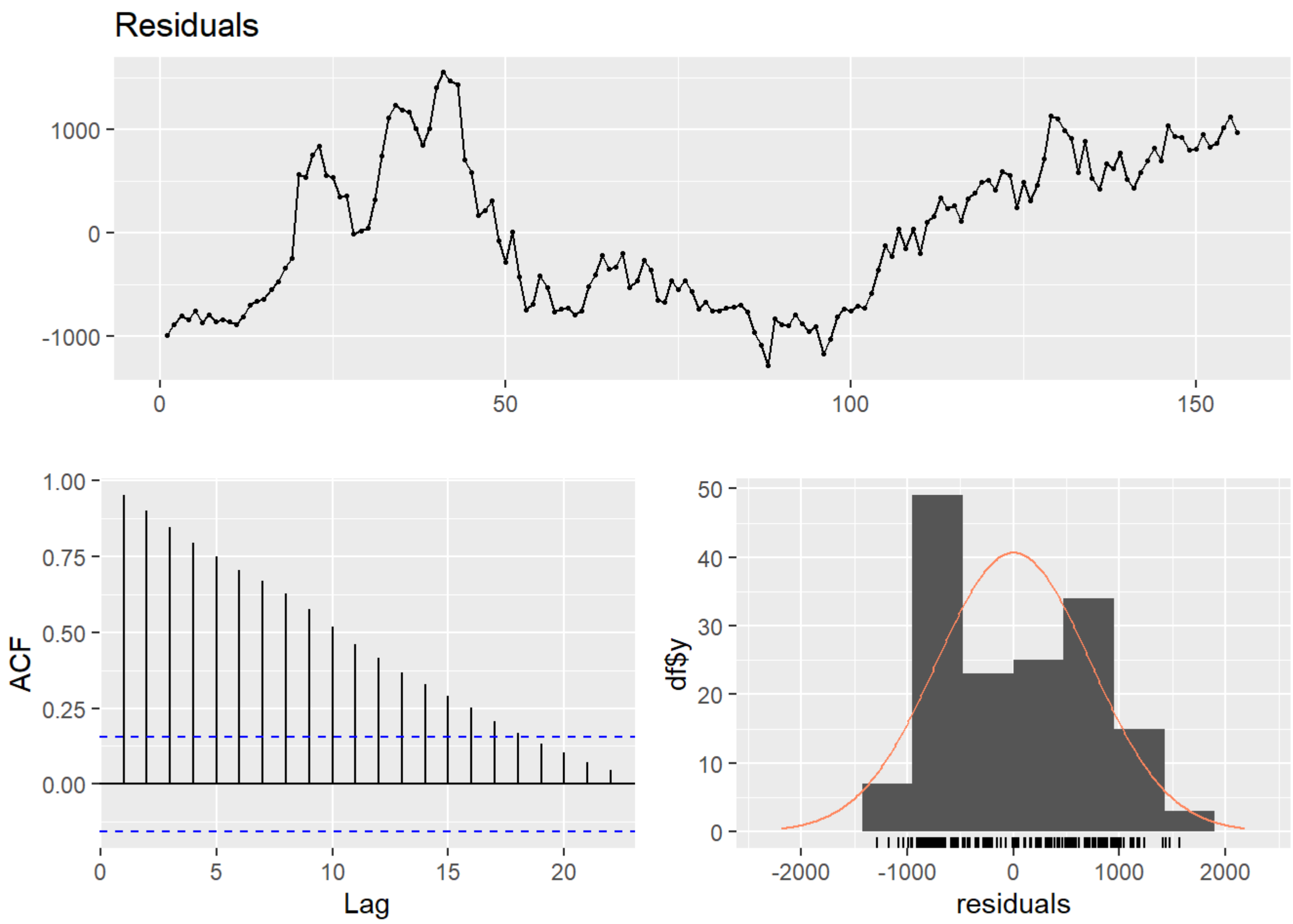
```
CuAmodel2=polyDlm(x=as.vector(copper),y=as.vector(asx),q=5,k=1,show.beta=TRUE)
```

```
## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0   0.1230    0.03080    4.00 9.86e-05
## beta.1   0.0871    0.01910    4.56 1.05e-05
## beta.2   0.0510    0.00814    6.26 3.72e-09
## beta.3   0.0148    0.00730    2.02 4.47e-02
## beta.4  -0.0214    0.01810   -1.18 2.38e-01
## beta.5  -0.0576    0.02980   -1.94 5.49e-02
```

```
summary(CuAmodel2)
```

```
##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1288.17  -723.90   -37.72   590.24  1562.18
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3662.09797  188.19249  19.459  < 2e-16 ***
## z.t0           0.12333    0.03083   4.001 9.81e-05 ***
## z.t1          -0.03618    0.01196  -3.025  0.00291 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 732.2 on 153 degrees of freedom
## Multiple R-squared:  0.2476, Adjusted R-squared:  0.2378
## F-statistic: 25.17 on 2 and 153 DF,  p-value: 3.536e-10
```

```
checkresiduals(CuAmodel2$model)
```



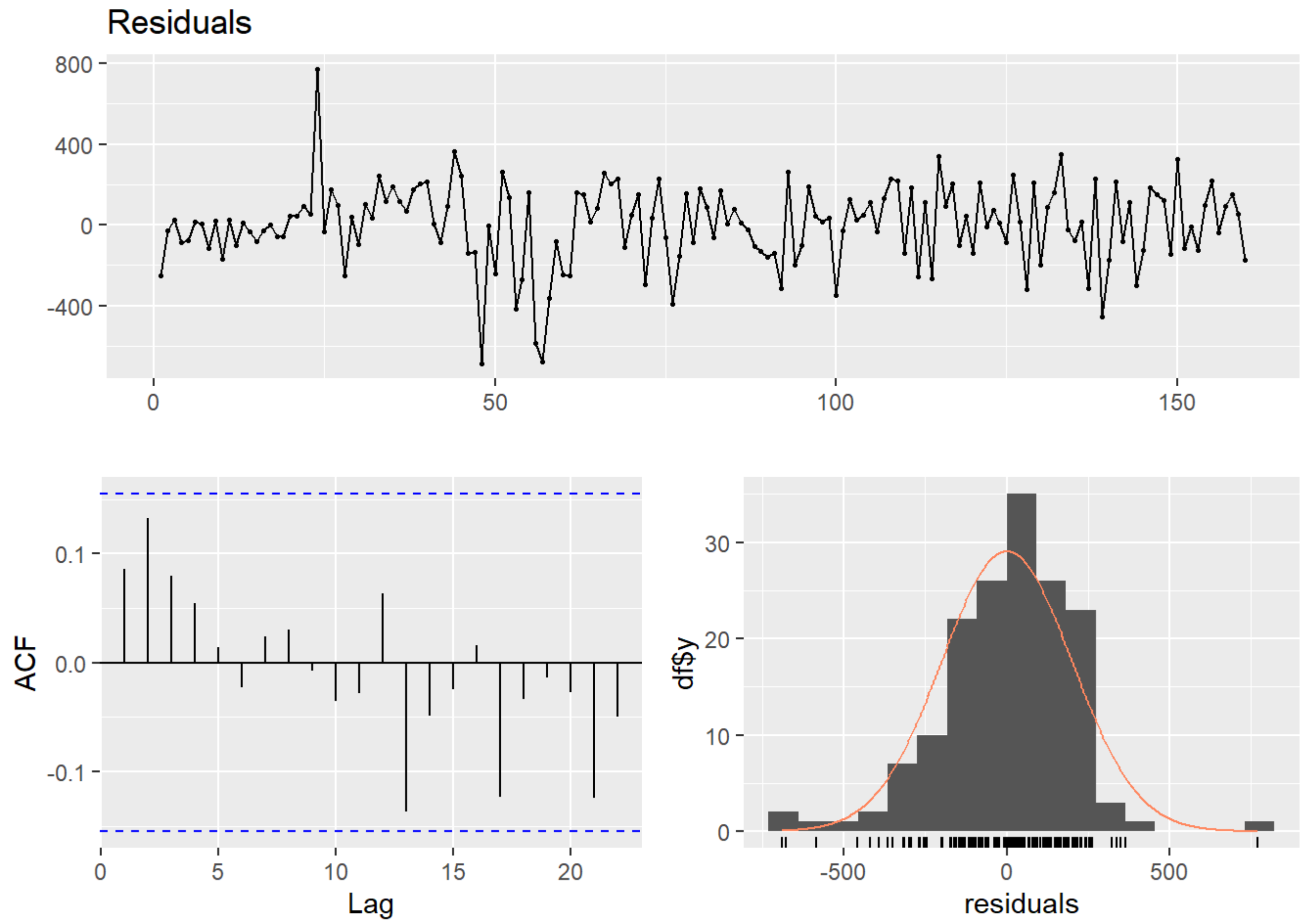

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 143.6, df = 10, p-value < 2.2e-16
```

KoyckDLM Model Fitting for ASX Price VS Copper

```
CuAmodel3=koyckDlm(x=as.vector(copper),y=as.vector(asx))
summary(CuAmodel3,diagnostics=TRUE)
```

```
##
## 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:
##                  df1 df2  statistic      p-value
## Weak instruments   1 157 1966.86799 1.043205e-90
## Wu-Hausman         1 156   10.97528 1.147725e-03
##
##                             alpha         beta       phi
## Geometric coefficients:  6672.885 -0.005863623 0.9716211
```

```
checkresiduals(CuAmodel3$model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 6.2327, df = 10, p-value = 0.7953
##
## Model df: 0.   Total lags used: 10
```

AutoRegressive DLM Model Fitting for ASX Price VS Copper

```
for(i in 1:5){
  for(j in 1:5){
    CuAmodel4.1 = ardlDlm(x = as.vector(copper), y = as.vector(asx), p = i, q = j)
    cat("p = ", i, "q = ", j, "AIC = ", AIC(CuAmodel4.1$model), "BIC = ", BIC(CuAmodel4.1$mod
el), "\n")
  }
}
```

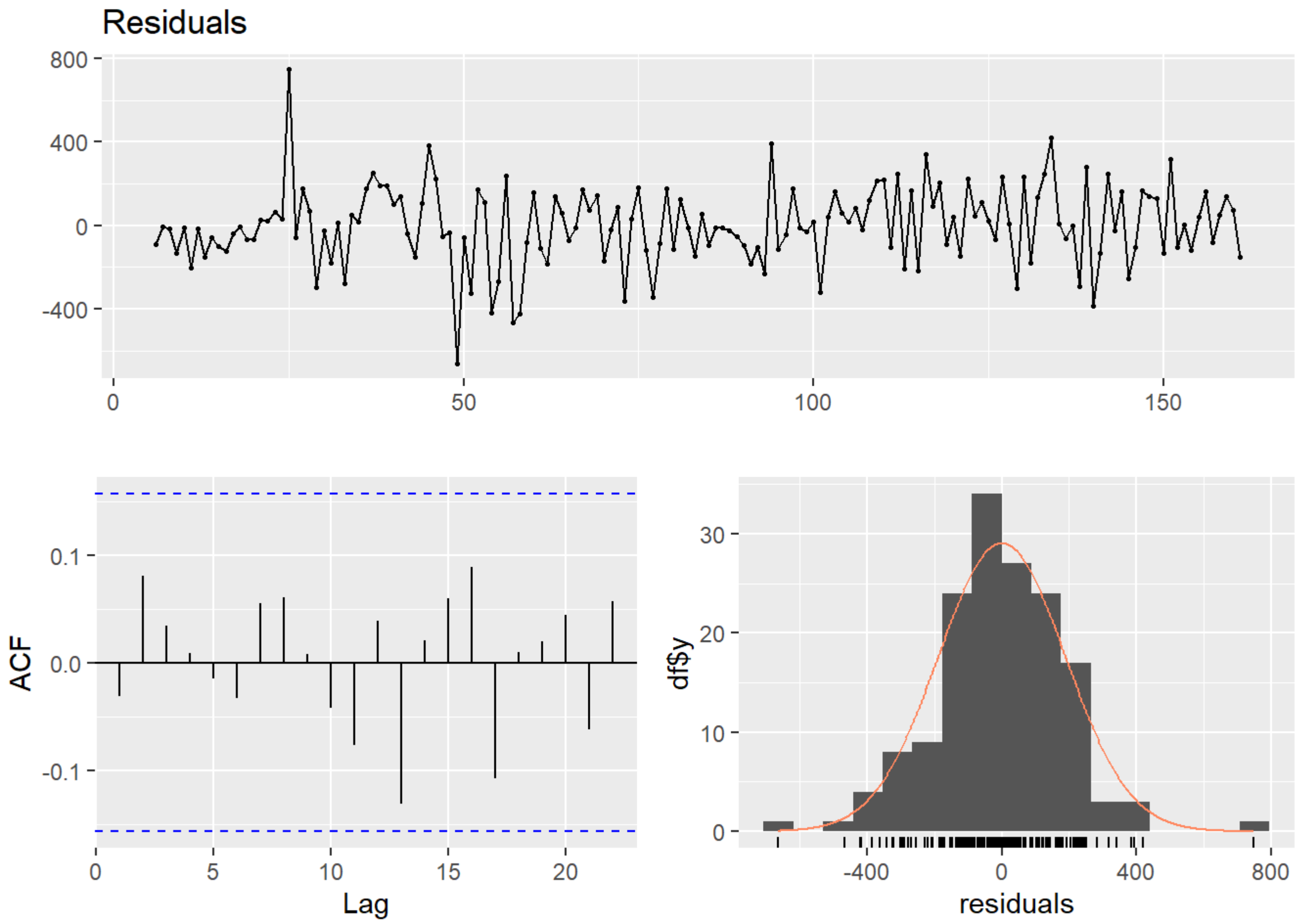
```
## p =  1 q =  1 AIC =  2147.741 BIC =  2163.116
## p =  1 q =  2 AIC =  2135.4 BIC =  2153.813
## p =  1 q =  3 AIC =  2121.12 BIC =  2142.558
## p =  1 q =  4 AIC =  2109.759 BIC =  2134.209
## p =  1 q =  5 AIC =  2099.056 BIC =  2126.505
## p =  2 q =  1 AIC =  2130.043 BIC =  2148.456
## p =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

The lowest AIC and BIC values are observed at p=5 and q=1, therefore I will proceed with those two values for Autoregressive DLM modelling.

```
CuAmodel4 = ardlDlm(x=as.vector(copper), y=as.vector(asx), p=5, q=1)
summary(CuAmodel4)
```

```
##
## Time series regression with "ts" data:
## Start = 6, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data, start = 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -663.01 -104.28   -6.52  134.04  748.83
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 223.07851   91.31917   2.443   0.0157 *
## X.t           0.08180    0.03445   2.374   0.0189 *
## X.1          -0.02382    0.05497  -0.433   0.6654
## X.2          -0.02426    0.05554  -0.437   0.6629
## X.3          -0.01059    0.05554  -0.191   0.8490
## X.4          -0.01423    0.05498  -0.259   0.7962
## X.5          -0.01564    0.03396  -0.461   0.6457
## Y.1           0.96496    0.02137  45.160   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 193.4 on 148 degrees of freedom
## Multiple R-squared:  0.9492, Adjusted R-squared:  0.9468
## F-statistic: 395.2 on 7 and 148 DF,  p-value: < 2.2e-16
```

```
checkresiduals(CuAmodel4$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 11
##
## data:  Residuals
## LM test = 5.0114, df = 11, p-value = 0.9306
```

Based on the above results after modelling, it can be concluded that the best distributed lag model for ASX Price Index and Copper Price is the AutoRegressive Lag model i.e. CuAmodel4 which has p=5 and q=1 as it has lowest AIC and BIC values among all i.e. AIC=2095.118 and BIC=2122.566. In addition to that it has a very high adjusted R-square value as well along with no significant lags in the ACF plot for the residuals.

Best Distributed Lag Model

```
vif(GAmodel4$model)
```

```
##       X.t L(X.t, 1) L(y.t, 1) L(y.t, 2) L(y.t, 3) L(y.t, 4) L(y.t, 5)
##  60.01512  58.78005  19.12905  39.78823  39.87314  40.82652  20.70864
```

```
vif(CAmodel4$model)
```

```
##       X.t L(X.t, 1) L(X.t, 2) L(X.t, 3) L(X.t, 4) L(X.t, 5) L(y.t, 1)
## 26.641389 75.289551 78.102900 79.148667 77.934884 27.873442  1.103211
```

```
vif(CuAmodel4$model)
```

```
##       X.t L(X.t, 1) L(X.t, 2) L(X.t, 3) L(X.t, 4) L(X.t, 5) L(y.t, 1)
## 19.978853 52.439298 55.194347 56.783339 57.189818 22.396512  1.364232
```

On comparing the models for the relations between ASX Price Index and other variables it was observed that the Autoregressive Distributed Lag Model was the best performer for all the three with lowest AIC and BIC values. Between these 3, the Autoregressive Lag Model for the ASX Price Index and Gold Price is the best performer with the lowest AIC value of 2092.194 and BIC value of 2119.643. Inspite of being the best performer, upon running VIF checks it was observed that there is a high presence of multicollinearity in the model which can be seen in the above R chunks.

Conclusion:

The main task of this report was to analyze the provided time series and to run visual checks to decide the presence of non-stationarity in the dataset. Upon visual inspection, it was observed that the series were non-stationary which were then confirmed using the ACF, PACF plots and the ADF test. In order to remove the non-stationarity, Box-Cox transformation and 1st order differencing was performed to address the issue of high variance and absence of constant mean.

While modelling the ASX Price Index with the other three variables, it was observed that the Autoregressive Distributed Lag Model was the best performer among all the 3 relations. The ARDLM model with ASX Price Index and Gold Price was the best distributed lag model with p=1 and q=5 which had an adjusted R-square value of 0.9478. However, upon running VIF checks on this model it was observed that there is a high presence of multicollinearity which may affect its accuracy.