FORECASTING – ASSIGNMENT 1

VISHAL PATNAIK DAMODARAPATRUNI - s3811521

17/08/2021

Introduction

The ASX data consists the monthly changes in all ordinaries (Ords) Price Index, Gold price (AUD), Crude Oil (Brent, USD/bbl) and Copper (USD/tonne) for 161 months, starting from 2004. This data is converted to Time series data.

Here, the time series data is analyzed for presence of stationary as well as the impact of components on the series data, then the respective models are fit on the series data to find the best model.

Scope

This analysis has three parts: Part 1: Checking for Non - Stationary Part 2: Impact of components on the series data Part 3: Identifying the best fit model for ASX price index

Part 1:

Identifying the trend and change in variance which makes the series stationary. Finally, performing Augumented Dicky Fuller test that says whether the series is stationary or not.

Part 2:

Using suitable decomposition method analyse the impact of individual components on the series data.

Part 3:

Finding the suitable distributed lag model among different models that best fits the ASX price index series.

Method

Using the below packages (forecast, TSA, tseries, expsmooth, funitRoots etc.) the time series data is visualized and analysed based on the stationarity and the decomposed components. Then the best distributed lag model for the ASX price index is selected.

```
library(expsmooth) # Forecasting with Exponential Smoothing. [1]
library(dplyr)
library(forecast) # Forecasting Functions for Time Series and Linear Models.
[2]
library(tseries) # Time Series Analysis and Computational Finance. [3]
```

```
library(fUnitRoots) # To analyze trends and unit roots in financial time seri
es. [4]
library(TSA) # Time Series Analysis.
library(urca) # Unit Root and Cointegration Tests. [5]
library(readr)
library(dLagM) # Distributed lag model.
library(VIF)
```

Data

The data is the monthly averages of all ordinaries (Ords) Price Index, Gold price (AUD), Crude Oil (Brent, USD/bbl) and Copper (USD/tonne). The data starts from 2004 and ends after 161 months. The dataset is in csv format and hence it is loaded using "read.csv()" function.

```
v_ASX_data <- read.csv("ASX_data.csv", header = TRUE)
head(v_ASX_data)

##   ASX.price Gold.price Crude.Oil..Brent._USD.bbl Copper_USD.tonne
## 1    2935.4      611.9                     31.29            1,650
## 2    2778.4      603.3                     32.65            1,682
## 3    2848.6      565.7                     30.34            1,656
## 4    2970.9      538.6                     25.02            1,588
## 5    2979.8      549.4                     25.81            1,651
## 6    2999.7      535.9                     27.55            1,685

# Using str() to check the type of each column.
str(v_ASX_data)

## 'data.frame':    161 obs. of  4 variables:
##  $ ASX.price                : num  2935 2778 2849 2971 2980 ...
##  $ Gold.price               : chr  "611.9" "603.3" "565.7" "538.6" ...
##  $ Crude.Oil..Brent._USD.bbl: num  31.3 32.6 30.3 25 25.8 ...
##  $ Copper_USD.tonne         : chr  "1,650" "1,682" "1,656" "1,588" ...
```

As the columns Gold.price and Copper_USD.tonne are in char format, which are supposed to be numeric. Now let us convert them into numeric format. For this let us remove "," before converting.

```
# Removing Commas
v_ASX_data$Gold.price = gsub(",","", v_ASX_data$Gold.price)
v_ASX_data$Copper_USD.tonne = gsub(",","", v_ASX_data$Copper_USD.tonne)

# Converting char to numeric
v_ASX_data$Gold.price = as.numeric(as.character(v_ASX_data$Gold.price))
v_ASX_data$Copper_USD.tonne = as.numeric(as.character(v_ASX_data$Copper_USD.t
onne))

str(v_ASX_data)

## 'data.frame':    161 obs. of  4 variables:
##  $ ASX.price                : num  2935 2778 2849 2971 2980 ...
```

```
##  $ Gold.price                 : num  612 603 566 539 549 ...
##  $ Crude.Oil..Brent._USD.bbl: num  31.3 32.6 30.3 25 25.8 ...
##  $ Copper_USD.tonne          : num  1650 1682 1656 1588 1651 ...
```

Checking Missing values.

```
colSums(is.na(v_ASX_data))

##                 ASX.price                Gold.price Crude.Oil..Brent._USD.
bbl
##                         0                         0
0
##          Copper_USD.tonne
##                         0
```

There are no missing values in the data.

Checking the class of v_ASX_data. (It should be data frame.)

```
class(v_ASX_data)

## [1] "data.frame"
```

Converting each column into different time series objects. Here, I am taking start (2004, 1) because the data is monthly and is from 2004. Also, end (2017, 5) because there are 161 observations indicating 161 months which gives 13 years and 5 months. Frequency is 12 as there are 12 months in an year.

```
v_ASX_price_TS <- ts(v_ASX_data$ASX.price, start = c(2004, 1), end = c(2017,
5), frequency = 12)
v_GOLD_price_TS <- ts(v_ASX_data$Gold.price, start = c(2004, 1), end = c(2017
, 5), frequency = 12)
v_CRUDE_price_TS <- ts(v_ASX_data$Crude.Oil..Brent._USD.bbl, start = c(2004,
1), end = c(2017, 5), frequency = 12)
v_COPPER_price_TS <- ts(v_ASX_data$Copper_USD.tonne, start = c(2004, 1), end
= c(2017, 5), frequency = 12)
```

Confirming the class of each time series object.

```
class(v_ASX_price_TS)

## [1] "ts"

class(v_GOLD_price_TS)

## [1] "ts"

class(v_CRUDE_price_TS)

## [1] "ts"

class(v_COPPER_price_TS)
```

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

Now let us visualize each time series object.

ASX PRICE

```
plot(v_ASX_price_TS, type = "b", xlab = "years", ylab = "Price index", main =
"ASX price change from 2004-1 to 2017-5 (161 months)", pch = 1)
legend("bottomright", inset = .03, title = "ASX price", legend = "ASX price s
eries", horiz = TRUE, cex = 0.8, lty = 1, box.lty = 2, box.lwd = 2, pch = 1)
```

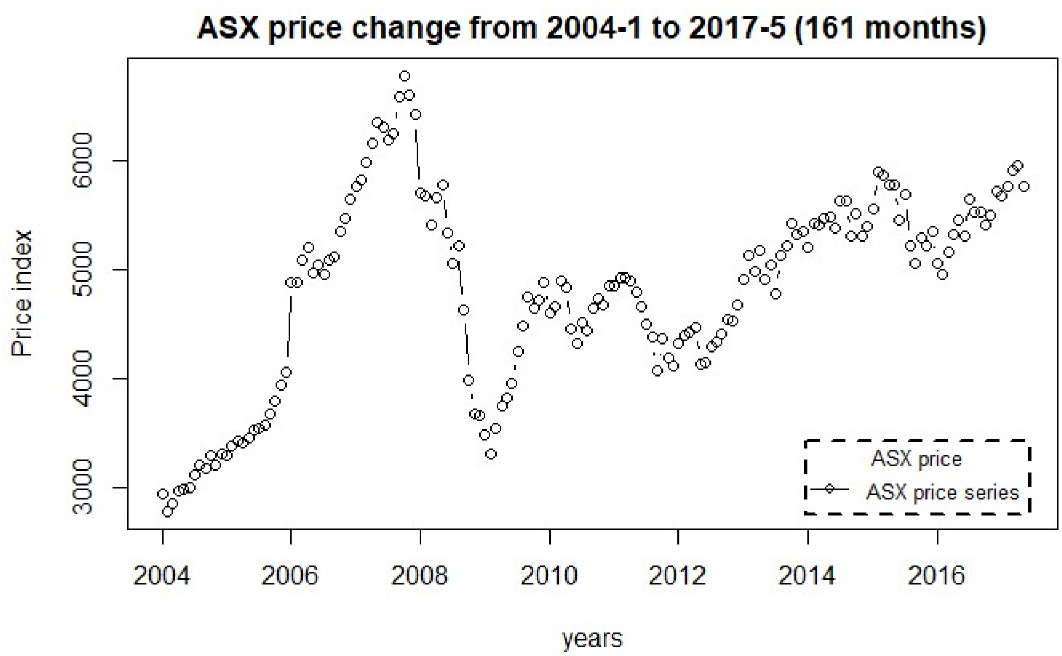

Fig 1: ASX price change - Time series plot.

```
McLeod.Li.test(y = v_ASX_price_TS, main = "McLeod-Li Test Statistics for ASX
price index")
```

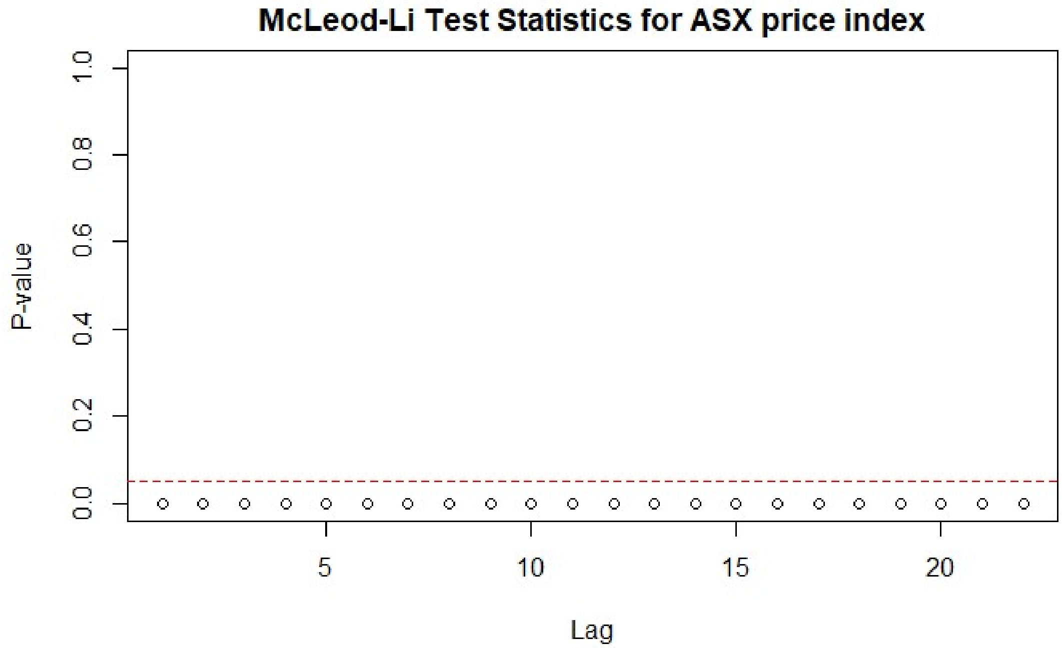

Fig 2: McLeod-Li Test Statistics for ASX price index.

Descriptive analysis

1. From fig1, we can observe an upward trend in the plot until 2017 with an intervention in the year 2008.
2. The ASX price series shows Autoregressive and moving average behaviour.
3. From fig1, we can conclude that there is no seasonality in the series.
4. From fig1 and fig2, we can see there a change in variance. Since, mean is not constant.

GOLD PRICE

```
plot(v_GOLD_price_TS, type = "b", xlab = "years", ylab = "Price index", main
= "GOLD price change from 2004-1 to 2017-5 (161 months)", pch = 1)
legend("bottomright", inset = .03, title = "GOLD price", legend = "GOLD price
series", horiz = TRUE, cex = 0.8, lty = 1, box.lty = 2, box.lwd = 2, pch = 1)
```

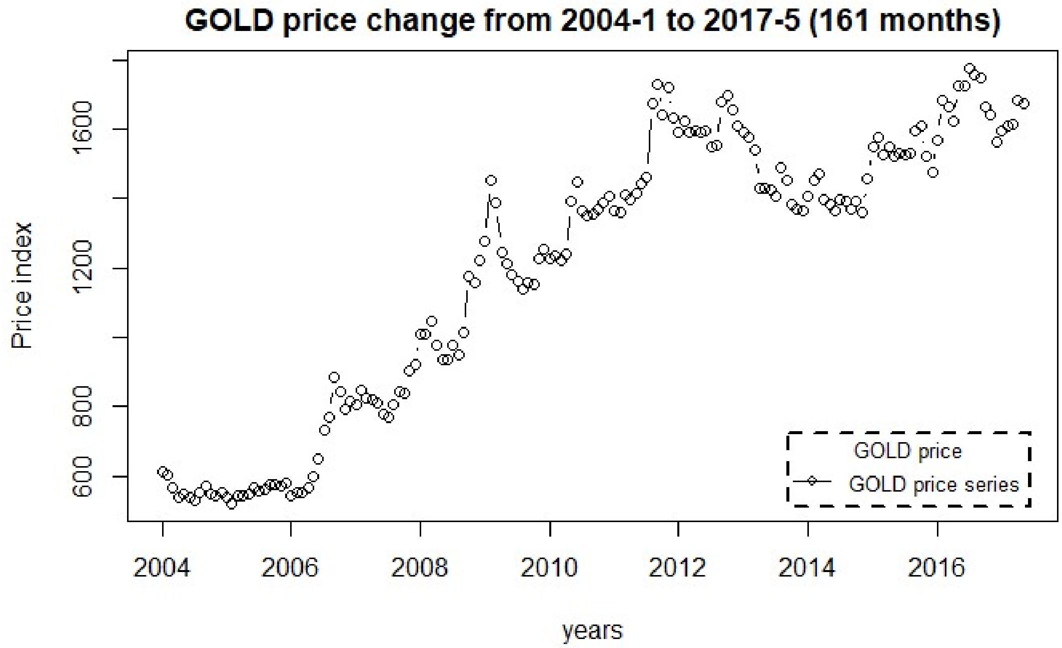

Fig 3: Gold price change - Time series plot.

```
McLeod.Li.test(y = v_GOLD_price_TS, main = "McLeod-Li Test Statistics for GOL
D price index")
```

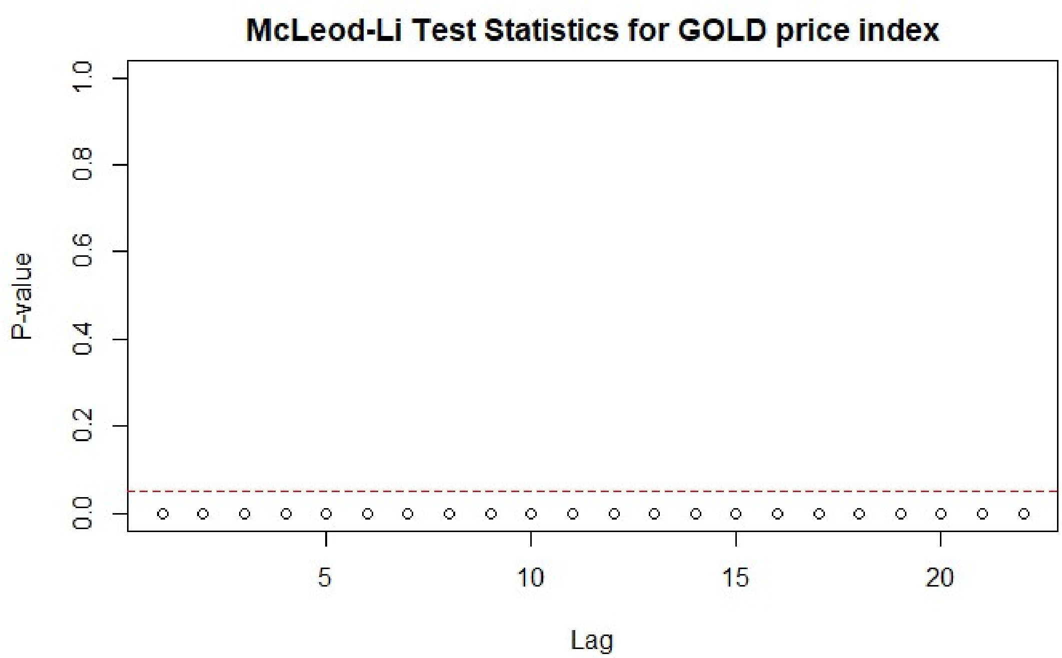

Fig 4: McLeod-Li Test Statistics for GOLD price index.

Descriptive analysis

1. From fig1, we can observe an upward trend in the plot until 2017 with no intervention in the trend.
2. The GOLD price series shows Autoregressive and moving average behaviour.
3. From fig1, we can conclude that there is no seasonality in the series.
4. From fig1 and fig2, we can see a change in variance. Since, mean is not constant.

CRUDE OIL PRICE

```
plot(v_CRUDE_price_TS, type = "b", xlab = "years", ylab = "Price index", main
= "CRUDE OIL price change from 2004-1 to 2017-5 (161 months)", pch = 1)
legend("topright", inset = .03, title = "CRUDE OIL price", legend = "CRUDE OI
L price series", horiz = TRUE, cex = 0.7, lty = 1, box.lty = 2, box.lwd = 2,
pch = 1)
```

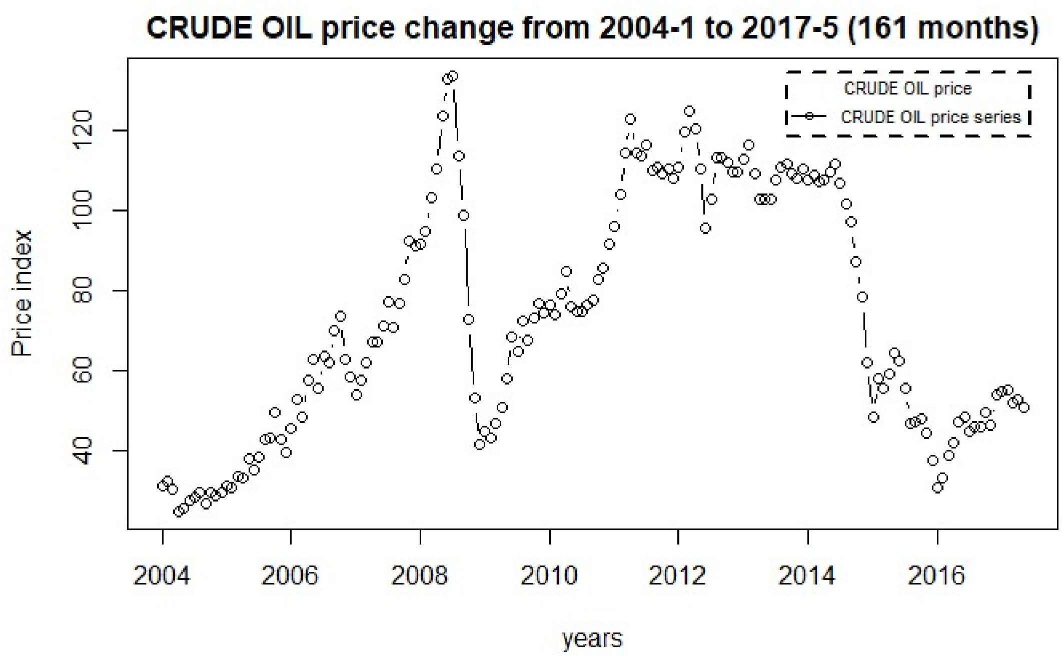

Fig 5: Crude Oil price change - Time series plot.

```
McLeod.Li.test(y = v_CRUDE_price_TS, main = "McLeod-Li Test Statistics for CR
UDE price index")
```

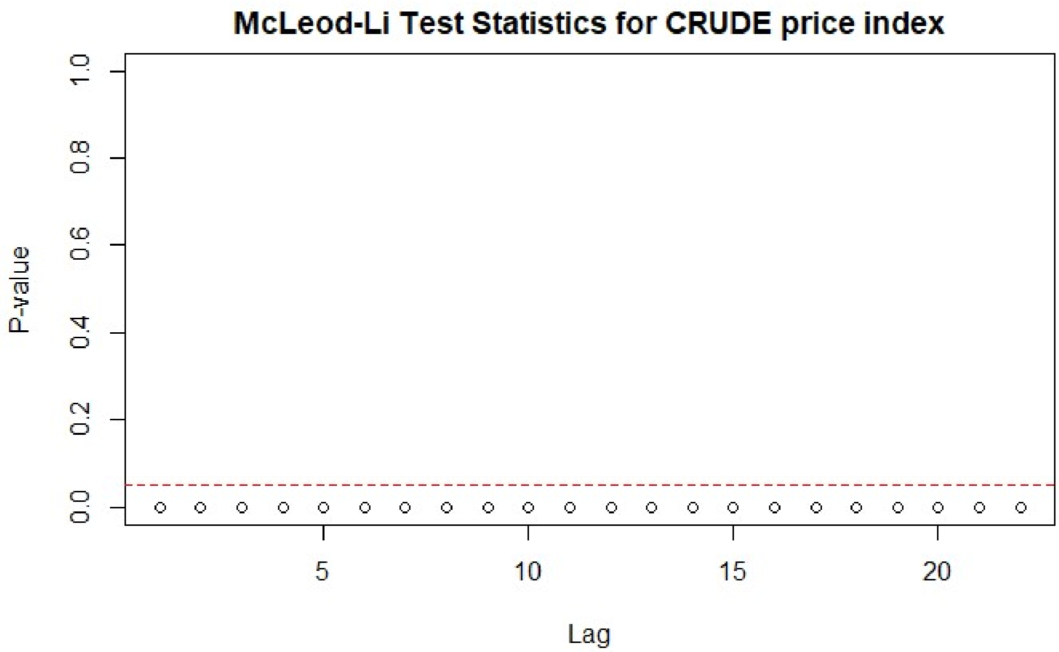

Fig 6: McLeod-Li Test Statistics for CRUDE price index.

Descriptive analysis

1. From fig1, we can observe an upward trend in the plot until 2007 with an intervention in the year 2008 and again an upward trent till 2012 which later followed a downward patern.
2. The CRUDE price series shows Autoregressive and moving average behaviour.
3. From fig1, we can conclude that there is no seasonality in the series.
4. From fig1 and fig2, we can see there is a change in variance. Since, mean is not constant.

COPPER PRICE

```
plot(v_COPPER_price_TS, type = "b", xlab = "years", ylab = "Price index", mai
n = "COPPER price change from 2004-1 to 2017-5 (161 months)", pch = 1)
legend("bottomright", inset = .03, title = "COPPER price", legend = "COPPER p
rice series", horiz = TRUE, cex = 0.8, lty = 1, box.lty = 2, box.lwd = 2, pch
= 1)
```

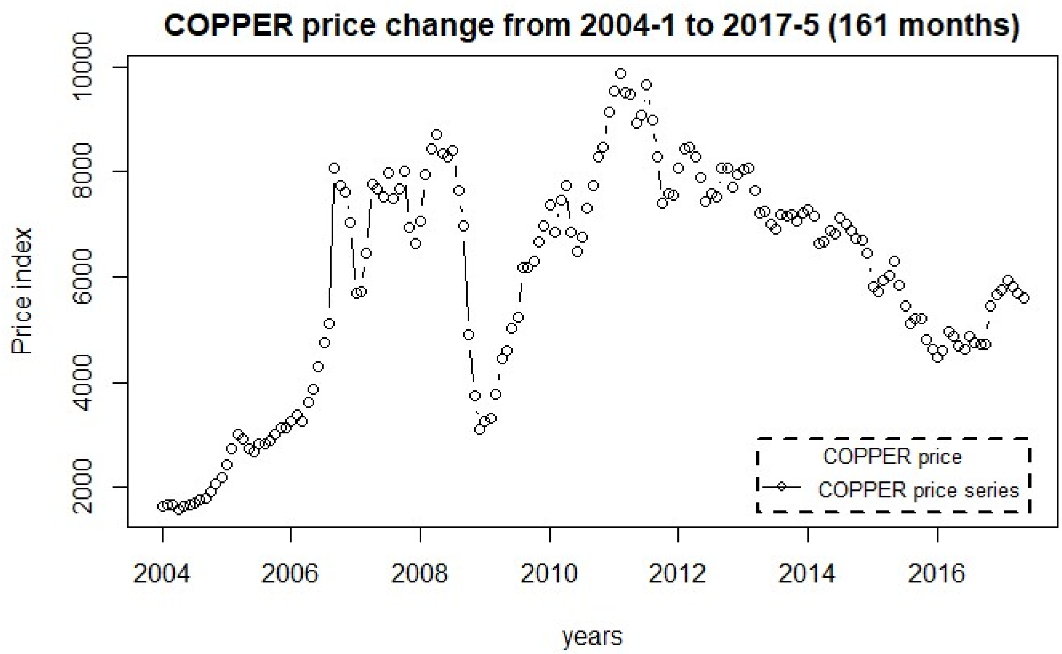

Fig 7: COPPER price change - Time series plot.

```
McLeod.Li.test(y = v_COPPER_price_TS, main = "McLeod-Li Test Statistics for C
OPPER price index")
```

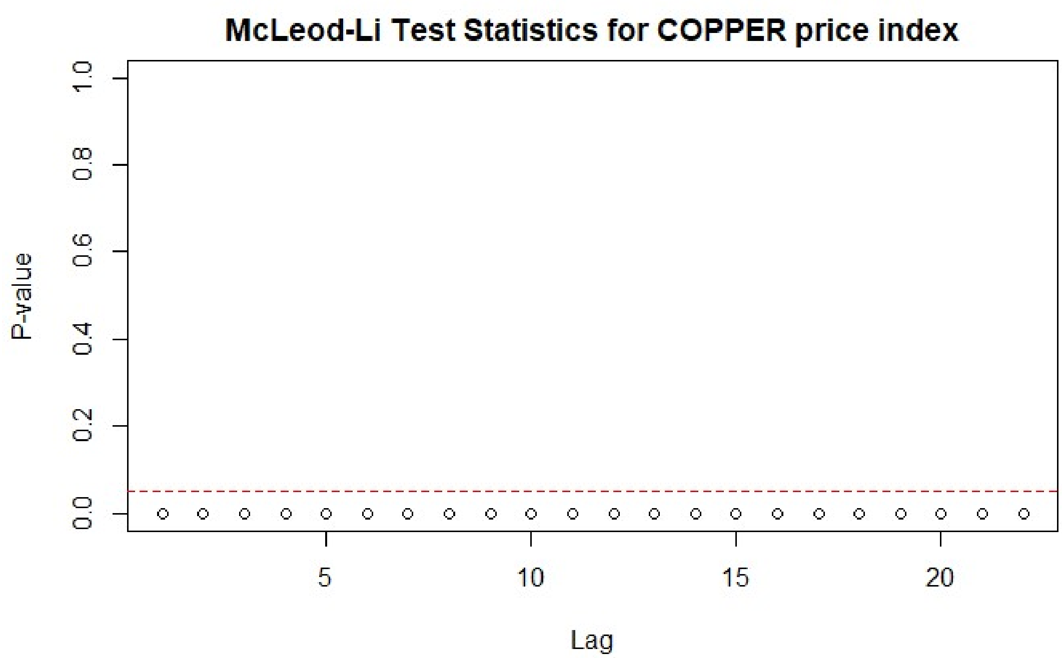

Fig 6: McLeod-Li Test Statistics for COPPER price index.

Descriptive analysis

1. From fig1, we can observe that the trend is almost like CRUDE series following an upward trend until 2008 with an intervention in the year 2009 and again an upward trend till 2011 which later followed a downward patern.
2. The COPPER price series shows Autoregressive and moving average behaviour.
3. From fig1, we can conclude that there is no seasonality in the series.
4. From fig1 and fig2, we can see there is a change in variance. Since, mean is not constant.

The existence of Non - Stationary

```
# Function to check Stationary on the series.
Stationary_Check <- function(x) {

  # Analysing trends by plotting ACF and PACF.
  par(mfrow = c(1,2))
  acf(x, main = "Price change - ACF")
  pacf(x, main = "Price change - PACF")

  # Conducting Augmented Dickey-Fuller test.
  adf.test(x)
}
```

Checking for Stationary on ASX price

```
Stationary_Check(v_ASX_price_TS)
```

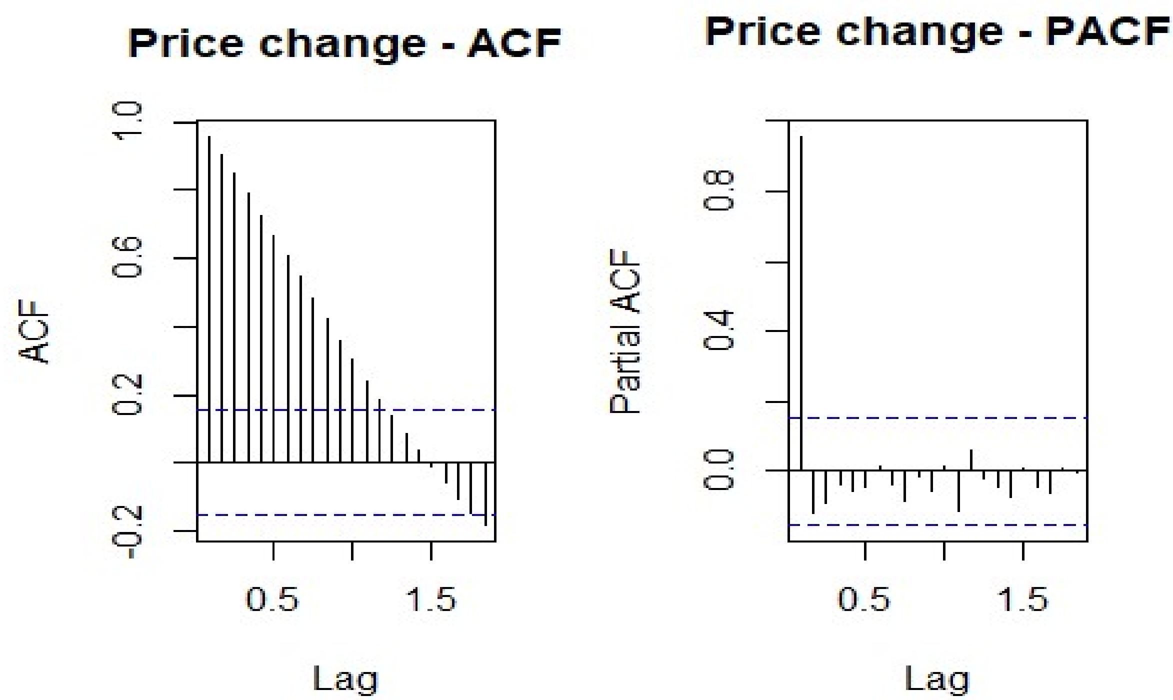

Fig 9: ASX price change - ACF and PACF

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

The decrease in the ACF plot and a high peak in the PACF plot in the beginning, suggests that there is some pattern in the ASX price trend.

Hypotheses:

H0: The data is not stationary.

HA: The data is stationary.

Interpretations:

- p - value: 0.2846 > 0.5
- p - value is greater than 0.5 and hence the test is not statistically significant. Therefore, we fail to reject Null hypothesis i.e., The data is not stationary.
- Also, as there is change in variance suggesting that the series is not stationary.
- Therefore, the ASX price series is non - stationary.

Checking for Stationary on GOLD price

```
Stationary_Check(v_GOLD_price_TS)
```

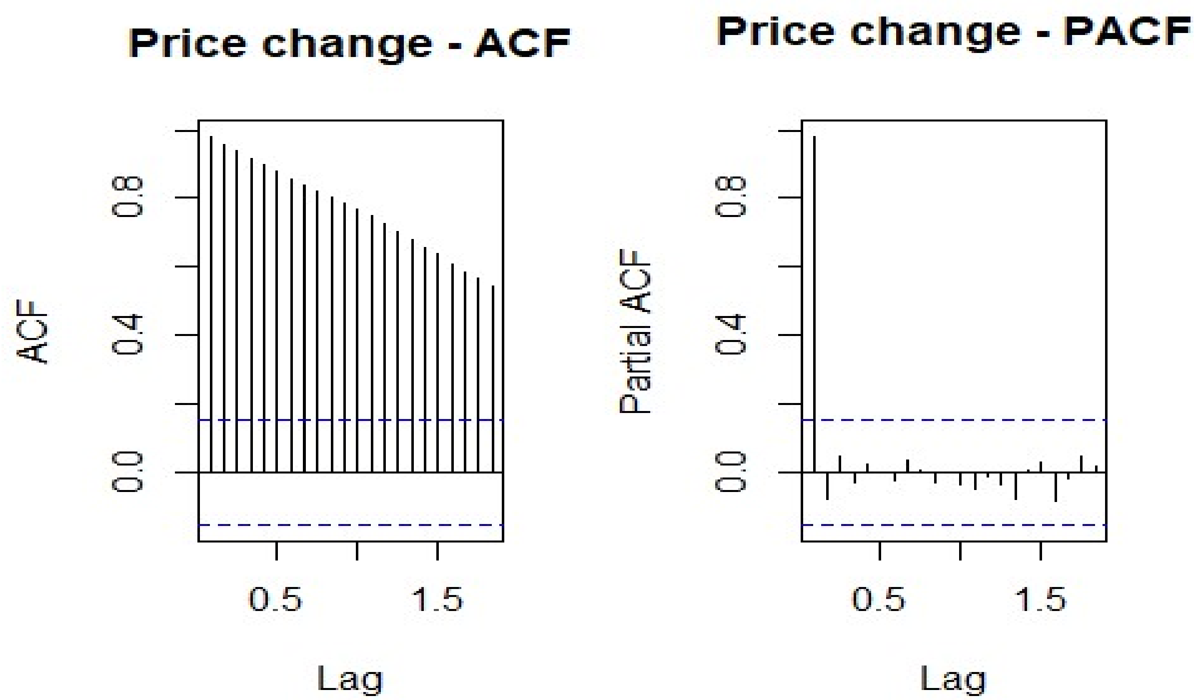

Fig 10: GOLD price change - ACF and PACF

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

The decrease in the ACF plot and a high peak in the PACF plot in the beginning, suggests that there is some pattern in the GOLD price trend.

Hypotheses:

H0: The data is not stationary.

HA: The data is stationary.

Interpretations:

- p - value: 0.6444 > 0.5
- p - value is greater than 0.5 and hence the test is not statistically significant. Therefore, we fail to reject Null hypothesis i.e., The data is not stationary.
- Also, as there is change in variance suggesting that the series is not stationary.
- Therefore, the GOLD price series is non - stationary.

Checking for Stationary on CRUDE price

```
Stationary_Check(v_CRUDE_price_TS)
```

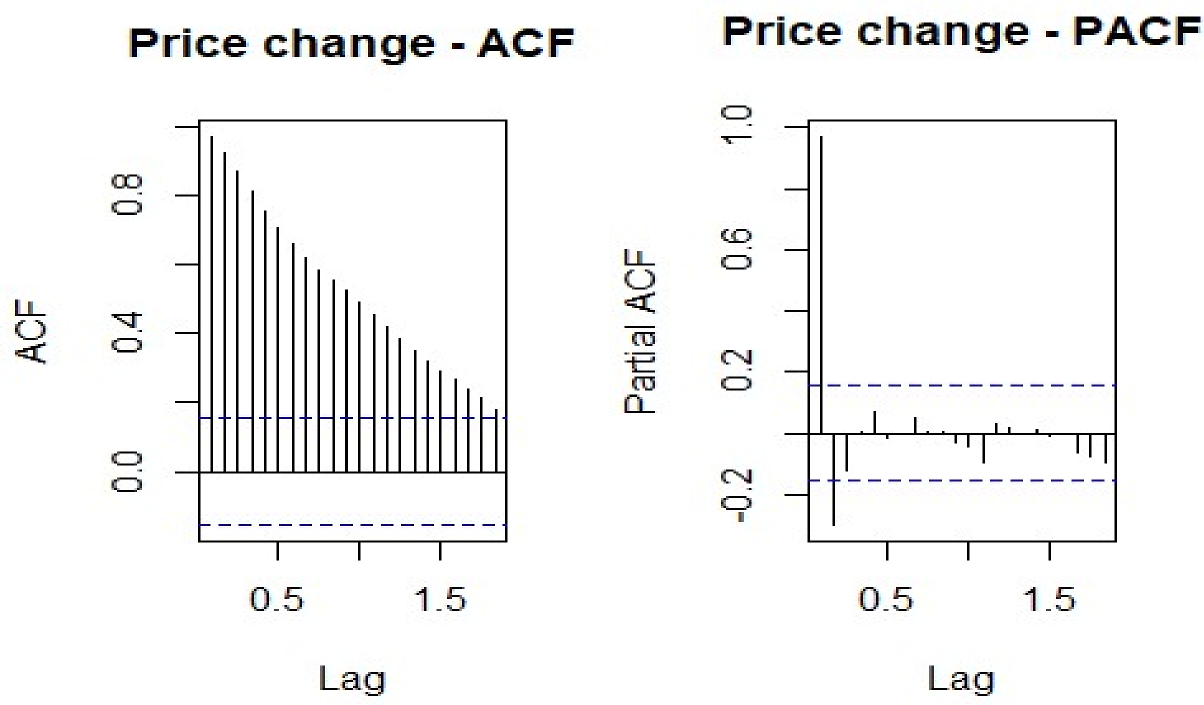

Fig 11: CRUDE price change - ACF and PACF

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

The decrease in the ACF plot and a high peak in the PACF plot in the beginning, suggests that there is some pattern in the CRUDE price trend.

Hypotheses:

H_0: The data is not stationary.

H_A: The data is stationary.

Interpretations:

- p - value: 0.6379 > 0.5
- p - value is greater than 0.5 and hence the test is not statistically significant. Therefore, we fail to reject Null hypothesis i.e., The data is not stationary.
- Also, as there is change in variance suggesting that the series is not stationary.
- Therefore, the CRUDE price series is non - stationary.

Checking for Stationary on COPPER price

```
Stationary_Check(v_COPPER_price_TS)
```

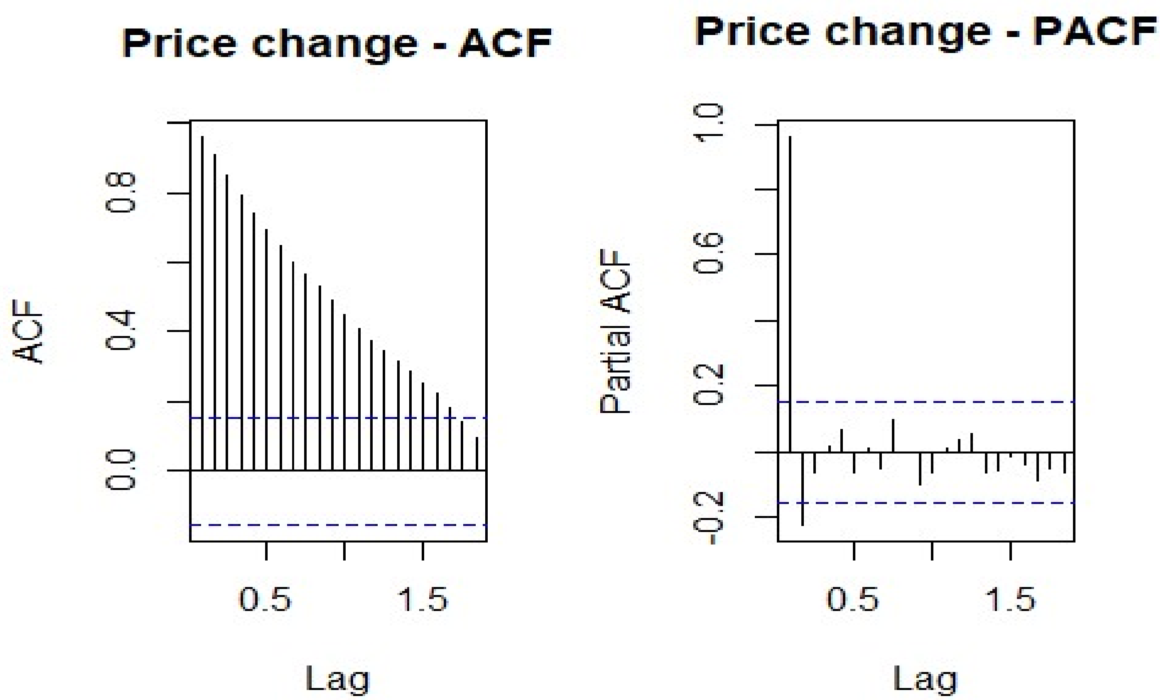

Fig 12: COPPER price change - ACF and PACF

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

The decrease in the ACF plot and a high peak in the PACF plot in the beginning, suggests that there is some pattern in the COPPER price trend.

Hypotheses:

H_0: The data is not stationary.

H_A: The data is stationary.

Interpretations:

- p - value: 0.472 > 0.5
- p - value is greater than 0.5 and hence the test is not statistically significant. Therefore, we fail to reject Null hypothesis i.e., The data is not stationary.
- Also, as there is change in variance suggesting that the series is not stationary.
- Therefore, the COPPER price series is non - stationary.

Impact of components on each time series.

The components of a series are usually,

1. Seasonality
2. Trend
3. Remainder

We should decompose the time series into the above components as we can see the impact of these components on the series data.

For this STL decomposition is used, as there is intervention in some of the series. This intervention is might be due to outliers and STL decomposition is robust in the case of outliers.

Decomposing ASX price series into components.

```
v_ASX_stl_decomp <- stl(v_ASX_price_TS, t.window = 15, s.window = "periodic", robust = TRUE)
plot(v_ASX_stl_decomp, main = "Decomposing ASX price Series into components")
```

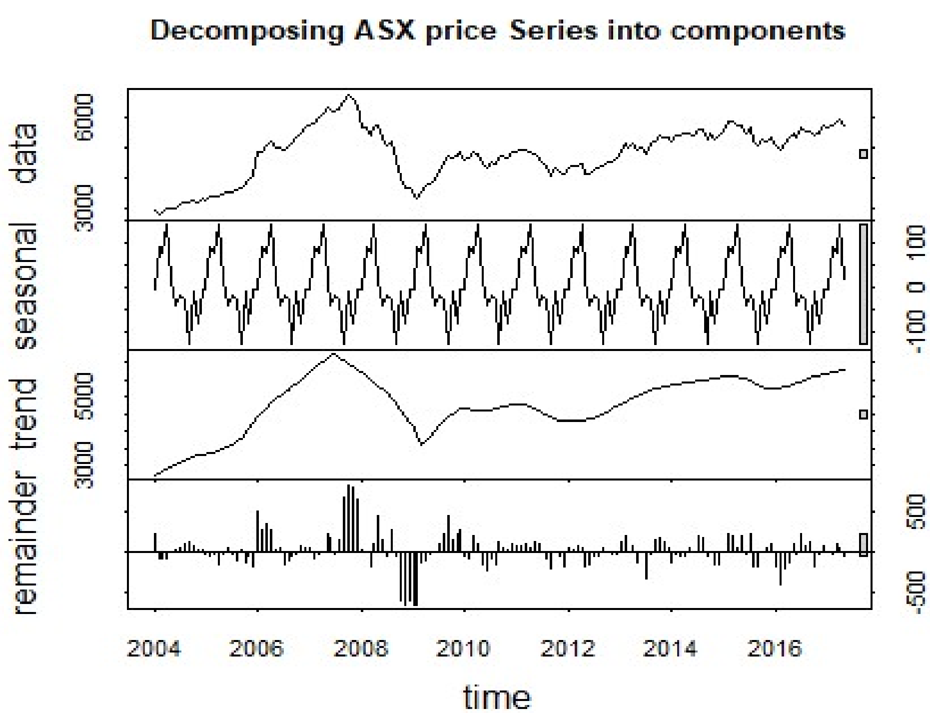

Fig 13: Decomposing ASX price series into components - stl decomposition.

1. The seasonal adjusted series is not meaningful in this case as it much deviated from the original series. Suggesting that there is no seasonality in the series as stated earlier.

2. The trend in the series data is shown exactly by the trend component.
3. Remainder component shows a high intervention point around 2008 depicting the real time global financial effect.

Decomposing GOLD price series into components.

```
v_GOLD_stl_decomp <- stl(v_GOLD_price_TS, t.window = 15, s.window = "periodic
", robust = TRUE)
plot(v_GOLD_stl_decomp, main = "Decomposing GOLD price Series into components
")
```

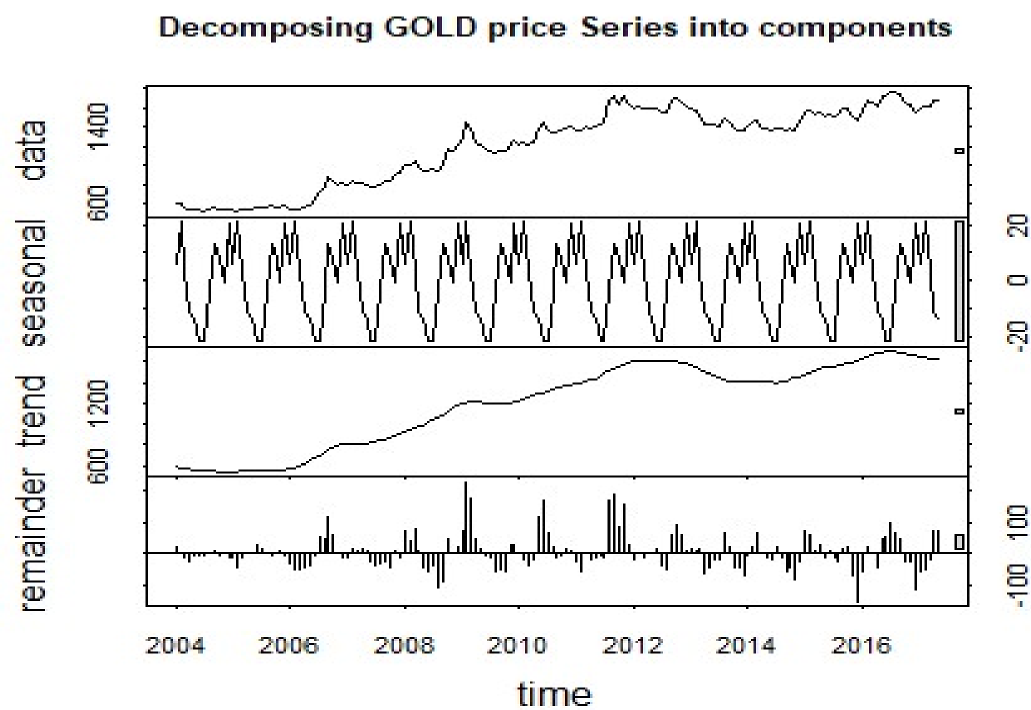

Fig 14: Decomposing GOLD price series into components - stl decomposition.

1. The seasonal adjusted series is not meaningful in this case as it much deviated from the original series. Suggesting that there is no seasonality in the series as stated earlier.
2. The trend in the series data is shown exactly by the trend component.
3. Remainder component shows a high intervention point at multiple points.

Decomposing CRUDE price series into components.

```
v_CRUDE_stl_decomp <- stl(v_CRUDE_price_TS, t.window = 15, s.window = "period
ic", robust = TRUE)
plot(v_CRUDE_stl_decomp, main = "Decomposing CRUDE price Series into componen
ts")
```

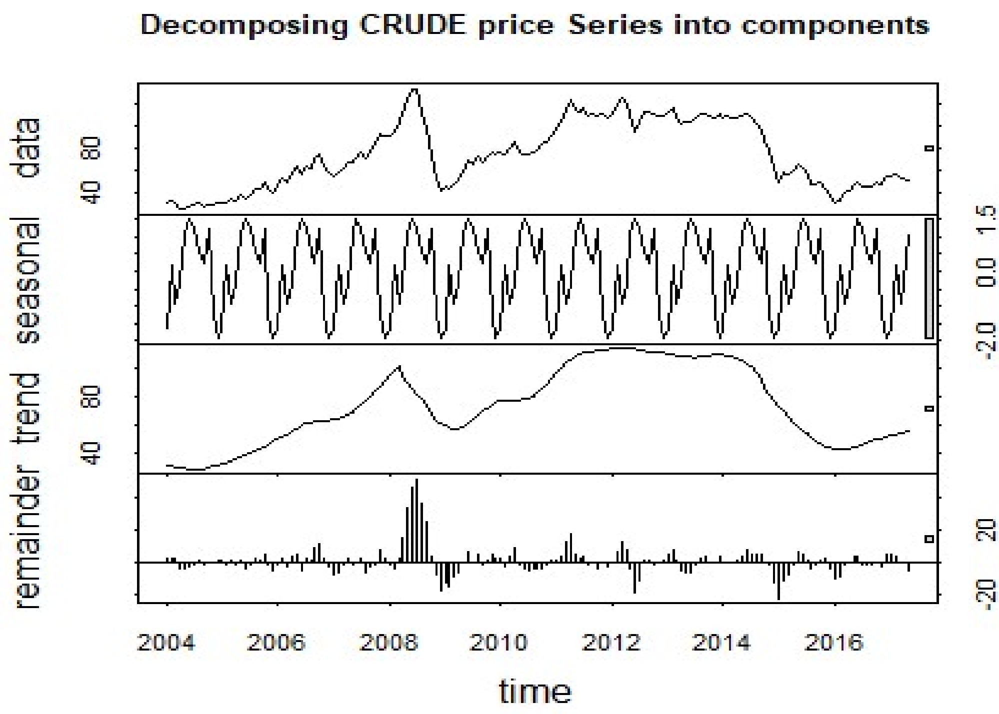

Fig 15: Decomposing CRUDE price series into components - stl decomposition.

1. The seasonal adjusted series is not meaningful in this case as it much deviated from the original series. Suggesting that there is no seasonality in the series as stated earlier.
2. The trend in the series data is shown exactly by the trend component.
3. Remainder component shows a high intervention point in 2008 depicting the real time global financial effect.

Decomposing COPPER price series into components.

```
v_COPPER_stl_decomp <- stl(v_COPPER_price_TS, t.window = 15, s.window = "peri
odic", robust = TRUE)
plot(v_COPPER_stl_decomp, main = "Decomposing COPPER price Series into compon
ents")
```

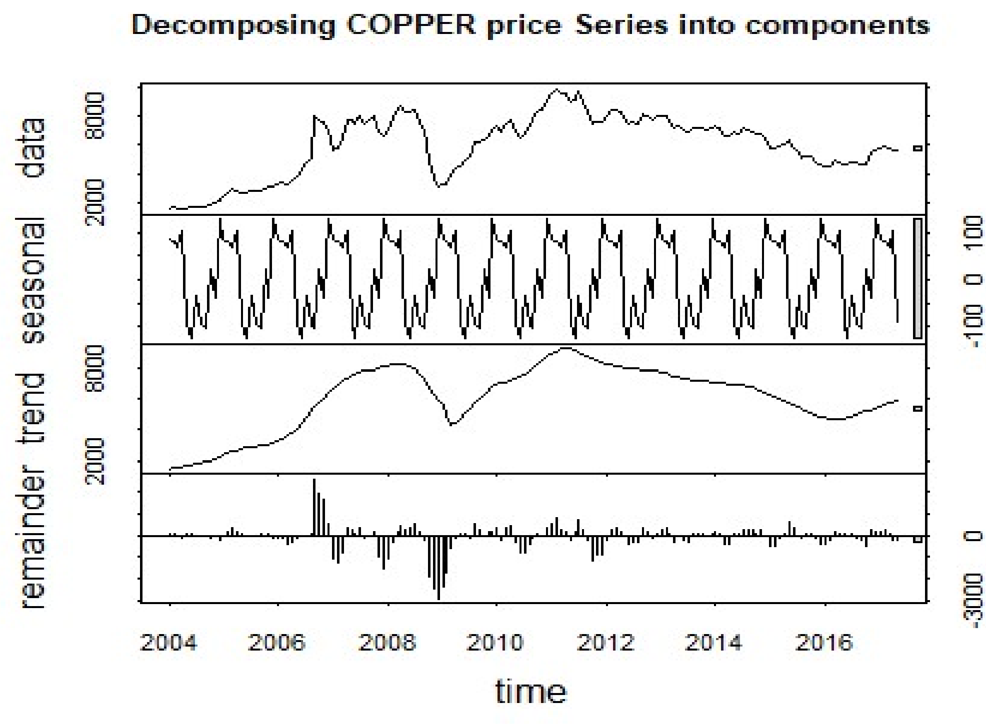

Fig 16: Decomposing COPPER price series into components - stl decomposition.

1. The seasonal adjusted series is not meaningful in this case as it much deviated from the original series. Suggesting that there is no seasonality in the series as stated earlier.
2. The trend in the series data is shown exactly by the trend component.
3. Remainder component shows a high intervention point around 2009 depicting the real time global financial effect.

Suitable distributed lag model for ASX price index

Before this let us find which variable has the highest correlation coefficient with ASX_price index.

```
# Calculating the correlation coefficient with Gold price.
cor(v_ASX_price_TS, v_GOLD_price_TS)

## [1] 0.3431908

# Calculating the correlation coefficient with CRUDE price.
cor(v_ASX_price_TS, v_CRUDE_price_TS)

## [1] 0.3290338
```

```
# Calculating the correlation coefficient with COPPER price.
cor(v_ASX_price_TS, v_COPPER_price_TS)

## [1] 0.5617864
```

As the correlation coefficient is higher w.r.t. Copper price series. let us now fit the model considering COPPER series as independent variable (x) where as ASX price series as dependent variable (y).

DLM models on ASX price index W.R.T COPPER price series.

Finite distributed lag model

```
x = v_COPPER_price_TS # Independent variable
y = v_ASX_price_TS # Dependent variable


for ( i in 1:10){
  model_1 = dlm(x = as.vector(x) , y = as.vector(y), q = i )
  cat("q = ", i, "AIC = ", AIC(model_1$model), "BIC = ", BIC(model_1$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
## q =  7 AIC =  2481.988 BIC =  2512.357
## q =  8 AIC =  2466.511 BIC =  2499.846
## q =  9 AIC =  2451.016 BIC =  2487.302
## q =  10 AIC =  2436.164 BIC =  2475.389
```

As we have the least AIC and BIC values at q = 10. Let us fit the finite distributed lag model with q = 10.

```
# Finite lag length based on AIC-BIC

finite_dlm = dlm( x = as.vector(x) , y = as.vector(y), q = 10)
summary(finite_dlm)

##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1154.09  -643.75   -11.55   596.33  1429.23
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.981e+03  2.166e+02  18.382   <2e-16 ***
## x.t          1.536e-01  1.354e-01   1.134    0.259
## x.1          1.857e-02  2.205e-01   0.084    0.933
## x.2          4.480e-02  2.220e-01   0.202    0.840
## x.3          2.830e-02  2.180e-01   0.130    0.897
## x.4          1.889e-02  2.175e-01   0.087    0.931
## x.5         -4.846e-02  2.191e-01  -0.221    0.825
## x.6          3.046e-02  2.175e-01   0.140    0.889
## x.7         -3.494e-03  2.189e-01  -0.016    0.987
## x.8         -1.349e-03  2.239e-01  -0.006    0.995
## x.9         -8.232e-02  2.222e-01  -0.371    0.712
## x.10        -1.012e-02  1.340e-01  -0.076    0.940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 737.4 on 139 degrees of freedom
## Multiple R-squared:  0.1931, Adjusted R-squared:  0.1292
## F-statistic: 3.024 on 11 and 139 DF,  p-value: 0.001201
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 2436.164 2475.389
```

Hypotheses:

H_0: The data doesn't fit the Finite distributed lag model.
H_A: The data fits the Finite distributed lag model.

Interpretations:

- F - statistic is 3.024
- R - squared is 0.1931
- Adjusted R - squared is 0.1292
- Degrees of freedom - DF are (11, 139)
- p - value (0.001201) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Finite distributed lag model.

This model suggests that there is only 19.31% of data variance. Suggesting that the model explains only 19.31% of the trend. Which implies that the model shows some trend.

Residual analysis

```
# Function for residual analysis.

res_analysis <- function(res_m) {

    par(mfrow = c(2, 2))
    # Scatter plot for model residuals
    plot(res_m, type = "b", pch = 19, col = "blue", xlab = "years", ylab = "S
tandardized Residuals", main = "Plot of Residuals over Time")
```

```
    abline(h = 0)

    # Standard distribution
    hist(res_m, xlab = 'Standardized Residuals', freq = FALSE)
    curve(dnorm(x, mean = mean(res_m), sd = sd(res_m)), col = "red", lwd = 2,
add = TRUE, yaxt = "n")

    # QQplot for model residuals
    qqnorm(res_m, col = c("blue"))
    qqline(res_m)

    # Auto-Correlation Plot
    acf(res_m, main = "ACF of Standardized Residuals",col=c("blue"))
}

res_analysis(residuals(finite_dlm$model))
```

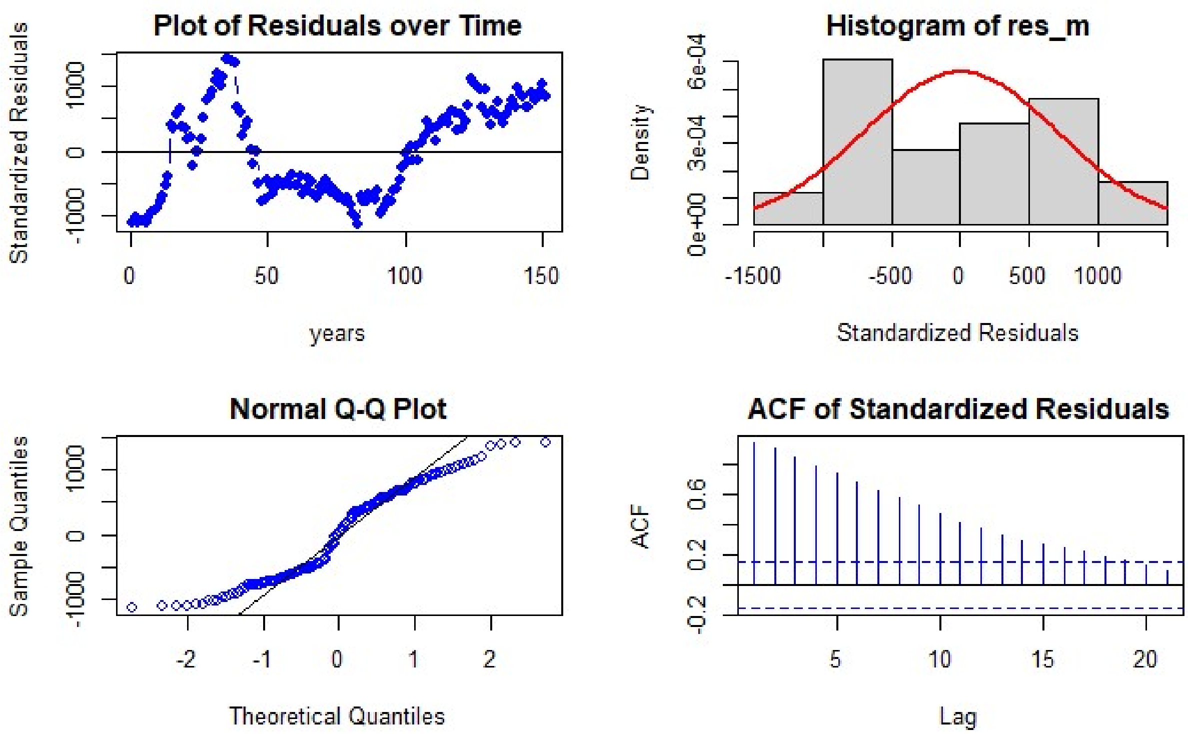

Fig 17: Residual Analysis Finite Distributed Lag Model.

Residual Analysis for Finite DLM:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.

2. From normal distribution curve, the distribution is almost symmetric and requires some transformation to make it symmetric. This suggests the non - stationary in the series.
3. The data at the tails is deviated more leaving some part on the line suggesting there is no normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.

Therefore, Further analysis is needed by adding polynomial to the lag model.

Polynomial distributed lag model

```
for (i in 1:3){
  model_2 <-  polyDlm(x = as.vector(x) , y = as.vector(y), q = i , k = i, sho
w.beta = FALSE)
  cat("q = ", i, "k = ", i, "AIC = ", AIC(model_2$model), "BIC = ", BIC(model
_2$model),"\n")
}

## q =  1 k =  1 AIC =  2574.488 BIC =  2586.789
## q =  2 k =  2 AIC =  2559.356 BIC =  2574.7
## q =  3 k =  3 AIC =  2544.155 BIC =  2562.531
```

Let us fit a polynomial model of order 3. Since least AIC and BIC scores.

```
# Ploynomial DLM

PolyDLM_model = polyDlm(x = as.vector(x), y = as.vector(y), q = 3, k = 3, sho
w.beta = TRUE)

## 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

summary(PolyDLM_model)

##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1332.00  -699.29   -97.89   621.39  1553.44
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3539.99286  185.97839  19.034   <2e-16 ***
## z.t0           0.16811    0.13081   1.285    0.201
```

```
## z.t1          -0.29701    1.04763  -0.284    0.777
## z.t2           0.21928    0.96450   0.227    0.820
## z.t3          -0.04846    0.21330  -0.227    0.821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 742.7 on 153 degrees of freedom
## Multiple R-squared:  0.2733, Adjusted R-squared:  0.2543
## F-statistic: 14.39 on 4 and 153 DF,  p-value: 5.404e-10
```

Hypotheses:

H_0: The data doesn't fit the Polynomial distributed lag model.

H_A: The data fits the Polynomial distributed lag model.

Interpretations:

- F - statistic is 14.39
- R - squared is 0.2733
- Adjusted R - squared is 0.2543
- Degrees of freedom - DF are (4, 153) p - value (0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Polynomial distributed lag model.

This model suggests that there is only 27.33% of data variance. Suggesting that the model explains only 27.33% of the trend. Which implies that the model shows some trend.

Residual analysis

```
res_analysis(residuals(PolyDLM_model$model))
```

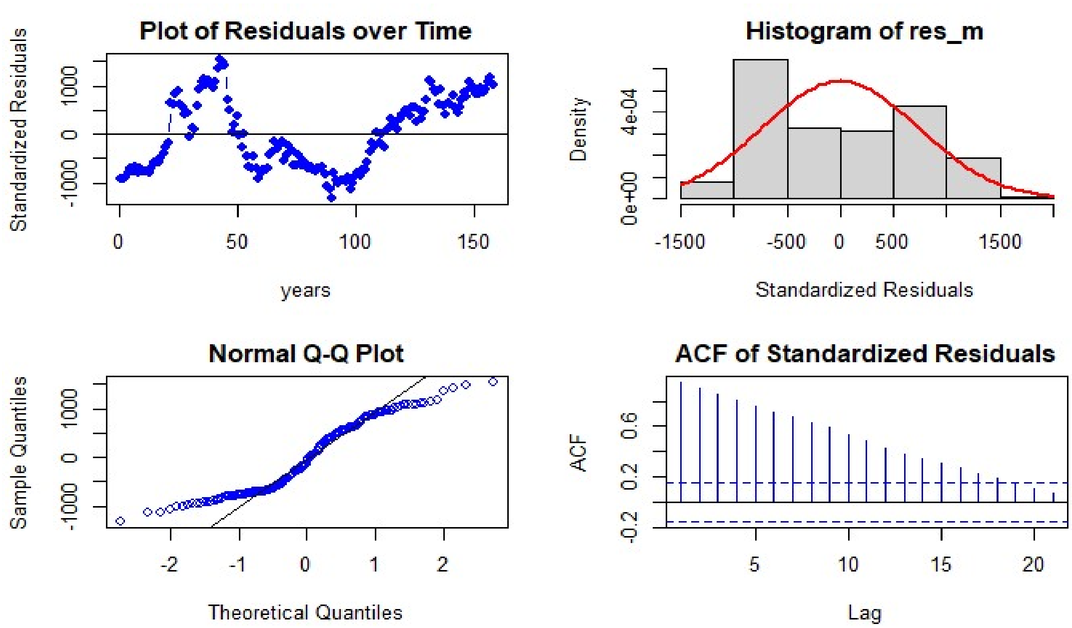

Fig 16: Residual Analysis Polynomial Distributed Lag Model.

Residual Analysis for Polynomial DLM:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric and requires some transformation to make it symmetric. This suggests the non - stationary in the series.
3. The data at the tails is deviated more leaving some part on the line suggesting there is no normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.

This analysis is not enough and we still require a better model than this. Therefore, let us fit Koyck model.

Koyck model

```
# Koyk DLM

Koyck_DLM = koyckDlm(x = as.vector(x) , y = as.vector(y))
summary(Koyck_DLM)

##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -689.64 -108.62   12.78  140.20  771.79
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 189.368812  87.644648   2.161   0.0322 *
## Y.1           0.971621   0.021895  44.376   <2e-16 ***
## X.t          -0.005864   0.009517  -0.616   0.5387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.9 on 157 degrees of freedom
## Multiple R-Squared: 0.9485,  Adjusted R-squared: 0.9479
## Wald test:  1448 on 2 and 157 DF,  p-value: < 2.2e-16
##
## Diagnostic tests:
## NULL
##
##                              alpha         beta       phi
## Geometric coefficients:  6672.885 -0.005863623 0.9716211
```

Hypotheses:

H_0: The data doesn't fit the Koyck distributed lag model.
H_A: The data fits the Koyck distributed lag model.

Interpretations:

- Walt test - statistic is 1448
- R - squared is 0.9485
- Adjusted R - squared is 0.9479
- Degrees of freedom - DF are (2, 157) p - value (0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Koyck distributed lag model.

This model suggests that there is 94.85% of data variance. Suggesting that the model explains only 94.85% of the trend. Which implies that the model performs better on the series data.

Now let us perform residual analysis.

Residual analysis

```
res_analysis(residuals(Koyck_DLM))

##           2           3           4           5           6           7
## -253.202914  -30.610848   23.082621  -86.477237  -75.025302   12.803618
##           8           9          10          11          12          13
##    5.298155 -114.678300   18.261437 -170.880048   24.533185 -103.752494
##          14          15          16          17          18          19
##    8.852679  -32.170263  -83.952475  -27.466232   -2.098674  -56.865123
##          20          21          22          23          24          25
##  -56.258419   41.535921   44.158599   92.935178   51.235104  771.792699
##          26          27          28          29          30          31
##  -32.861153  176.884448   95.956039 -253.719314   38.523936  -95.592640
##          32          33          34          35          36          37
##  104.053203   35.245022  240.940477  115.938747  189.540536  117.568097
##          38          39          40          41          42          43
##   66.356304  175.905107  205.263342  213.917574    3.463459  -86.583649
##          44          45          46          47          48          49
##   91.002511  365.530625  242.621700 -141.692507 -135.968325 -689.639612
##          50          51          52          53          54          55
##   -3.431458 -243.873540  262.447878  137.163954 -417.990992 -268.934588
##          56          57          58          59          60          61
##  161.681077 -584.660135 -677.835397 -364.478905  -80.335251 -247.606666
##          62          63          64          65          66          67
## -252.350187  161.705150  149.290705   12.444825   82.742609  255.096471
##          68          69          70          71          72          73
##  202.046722  229.415809 -110.296909   50.285608  152.562166 -293.406077
##          74          75          76          77          78          79
##   35.557299  228.407743  -64.382442 -392.363504 -153.655477  155.549369
##          80          81          82          83          84          85
##  -87.231932  180.025045   87.330525  -62.445827  167.511796    7.078568
```

```
##         86         87         88         89         90         91 
##  79.904245  11.079317 -23.496061 -108.066932 -129.399854 -159.845072 
##         92         93         94         95         96         97 
## -139.488905 -316.487992 259.891585 -197.000699 -99.959552 189.269180 
##         98         99        100        101        102        103 
##  45.284433  16.731182  31.851697 -349.789770 -26.704157 126.361650 
##        104        105        106        107        108        109 
##  25.995248  48.492013 112.049456 -32.844966 132.156466 226.632804 
##        110        111        112        113        114        115 
## 216.382610 -139.689291 182.996258 -254.784471 112.830231 -266.261379 
##        116        117        118        119        120        121 
## 338.187556  90.458999 203.539204 -100.085239  42.550965 -142.702300 
##        122        123        124        125        126        127 
## 210.564995  -9.092880  70.895949   9.292441 -85.832876 246.174124 
##        128        129        130        131        132        133 
##  12.765403 -317.254300 208.671523 -200.700159  89.282101 160.744259 
##        134        135        136        137        138        139 
## 348.671923 -23.746229 -75.816805  12.622825 -314.981263 227.827780 
##        140        141        142        143        144        145 
## -457.665890 -174.081084 214.773110 -81.539022 112.319064 -299.473074 
##        146        147        148        149        150        151 
## -127.601504 183.995301 149.606796 120.822879 -144.947561 323.454909 
##        152        153        154        155        156        157 
## -115.940495  -8.962756 -127.629175  95.901847 216.671093 -37.428029 
##        158        159        160        161 
##  92.511097 151.071498  55.297228 -174.052323
```

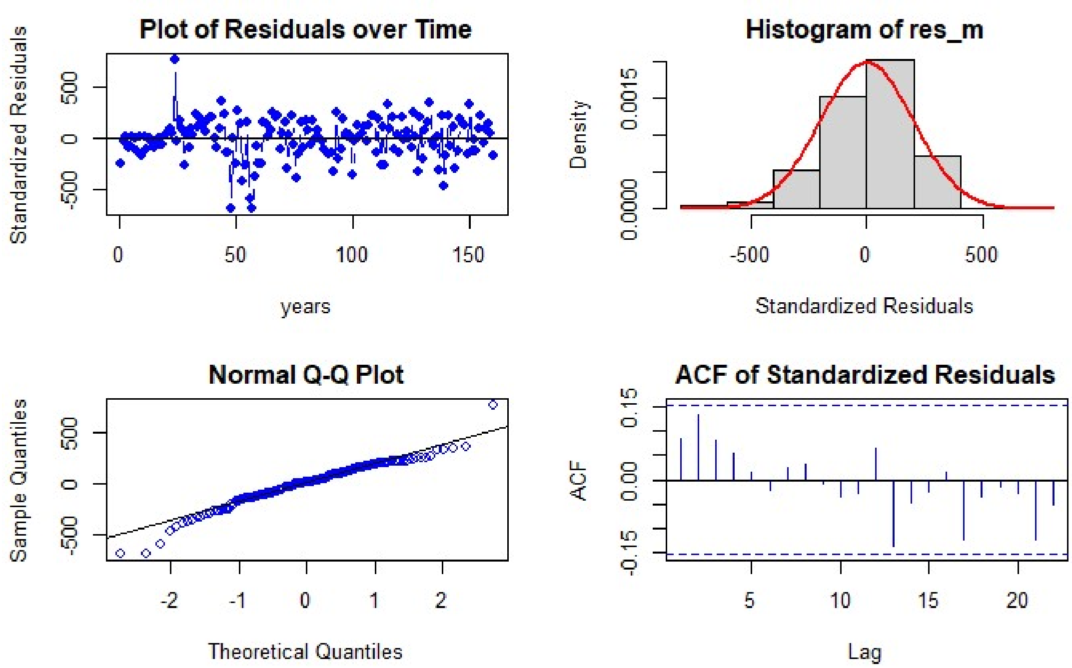

Fig 17: Residual Analysis Koyck Distributed Lag Model.

Residual Analysis for Koyck DLM:

1. The data points are below the line at both the start and end of the trend. Randomness is not seen here.
2. From normal distribution curve, the distribution is almost symmetric. This suggests a good fit with a very few data falling outside the normal curve indicating Kurtosis.
3. The data at the tails is deviated but is normal for most part of the line suggesting normality in the trend.
4. There are no significant lags in Autocorrelation plot suggesting that the stochastic component is white noise.

So far this is the best model but let us fit ardlDlm model to check whether it fits better than Koyck model or not.

Autoregressive distributed lag model

```
for (i in 1:10){
  for(j in 1:5){
    model_4 = ardlDlm(x = as.vector(x) , y = as.vector(y), p = i , q = j )
    cat("p = ", i, "q = ", j, "AIC = ", AIC(model_4$model), "BIC = ", BIC(mod
el_4$model),"\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
## p =  6 q =  1 AIC =  2084.49 BIC =  2114.924
```

```
## p =  6 q =  2 AIC =  2086.331 BIC =  2119.809
## p =  6 q =  3 AIC =  2087.163 BIC =  2123.684
## p =  6 q =  4 AIC =  2088.704 BIC =  2128.268
## p =  6 q =  5 AIC =  2090.603 BIC =  2133.211
## p =  7 q =  1 AIC =  2072.833 BIC =  2106.239
## p =  7 q =  2 AIC =  2074.698 BIC =  2111.141
## p =  7 q =  3 AIC =  2075.535 BIC =  2115.016
## p =  7 q =  4 AIC =  2077.211 BIC =  2119.729
## p =  7 q =  5 AIC =  2079.174 BIC =  2124.729
## p =  8 q =  1 AIC =  2062.338 BIC =  2098.703
## p =  8 q =  2 AIC =  2064.181 BIC =  2103.577
## p =  8 q =  3 AIC =  2065.007 BIC =  2107.433
## p =  8 q =  4 AIC =  2066.679 BIC =  2112.135
## p =  8 q =  5 AIC =  2068.654 BIC =  2117.141
## p =  9 q =  1 AIC =  2049.983 BIC =  2089.293
## p =  9 q =  2 AIC =  2051.863 BIC =  2094.197
## p =  9 q =  3 AIC =  2052.445 BIC =  2097.803
## p =  9 q =  4 AIC =  2054.13 BIC =  2102.512
## p =  9 q =  5 AIC =  2056.102 BIC =  2107.508
## p =  10 q =  1 AIC =  2034.551 BIC =  2076.793
## p =  10 q =  2 AIC =  2036.144 BIC =  2081.403
## p =  10 q =  3 AIC =  2036.502 BIC =  2084.779
## p =  10 q =  4 AIC =  2037.935 BIC =  2089.229
## p =  10 q =  5 AIC =  2039.913 BIC =  2094.224
```

p = 10 and q = 1 has the least AIC and BIC scores.

```
# ARDLM model

AR_DLM = ardlDlm(x = as.vector(x) , y = as.vector(y), p = 10 , q = 1 )
summary(AR_DLM)

##
## Time series regression with "ts" data:
## Start = 11, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data, start = 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -591.71 -104.56   -9.24  126.64  729.10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.362e+02  1.039e+02   2.273   0.0246 *
## X.t          7.806e-02  3.575e-02   2.183   0.0307 *
## X.1         -3.751e-02  5.815e-02  -0.645   0.5200
## X.2         -4.639e-04  5.855e-02  -0.008   0.9937
## X.3         -1.595e-02  5.751e-02  -0.277   0.7820
```

```
## X.4          -1.247e-02  5.736e-02  -0.217   0.8283
## X.5          -5.529e-02  5.779e-02  -0.957   0.3404
## X.6           6.729e-02  5.736e-02   1.173   0.2427
## X.7          -4.951e-03  5.772e-02  -0.086   0.9318
## X.8          -4.301e-02  5.904e-02  -0.728   0.4676
## X.9          -6.099e-02  5.859e-02  -1.041   0.2997
## X.10          7.708e-02  3.540e-02   2.178   0.0311 *
## Y.1           9.648e-01  2.237e-02  43.134   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 194.5 on 138 degrees of freedom
## Multiple R-squared:  0.9443, Adjusted R-squared:  0.9394
## F-statistic: 194.9 on 12 and 138 DF,  p-value: < 2.2e-16
```

Hypotheses:

H_0: The data doesn't fit the Autoregressive distributed lag model.

H_A: The data fits the Autoregressive distributed lag model.

Interpretations:

- F - statistic is 194.9
- R - squared is 0.9443
- Adjusted R - squared is 0.9394
- Degrees of freedom - DF are (12, 138)
- p - value (0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Autoregressive distributed lag model.

This model suggests that there is 94.43% of data variance. Suggesting that the model explains only 94.43% of the trend. Which implies that the model shows some trend.

Let us perform residual analysis on this model.

Residual analysis

```
res_analysis(residuals(AR_DLM))

## Time Series:
## Start = 11
## End = 161
## Frequency = 1
##           11           12           13           14           15
16
## -208.6351731  -27.8223986 -164.2241618  -57.9984615  -98.8140234 -130.3520
955
##           17           18           19           20           21
22
##  -49.5285799  -12.7876615  -61.9080221  -66.5659847   13.9863700   23.5101
249
##           23           24           25           26           27
28
##   93.7708791   53.3526685  729.0976415  -79.8309924  168.5590049   66.9160
```

```
186
##          29          30          31          32          33          
34
## -297.9795183  -22.5101142 -167.7885088    9.2668982 -254.7238591  101.6371
765
##          35          36          37          38          39          
40
##   -9.2423511  157.8843108  243.6849943  278.4677165  128.7998654   93.4043
167
##          41          42          43          44          45          
46
##  228.0045933  143.4354494 -124.4773632  151.3505504  325.8062664   65.5685
984
##          47          48          49          50          51          
52
##  -77.1919688   46.8233392 -591.7124400  -45.5162532 -292.0803572  149.8067
373
##          53          54          55          56          57          
58
##   83.6447179 -356.0311800 -233.2927676  127.9226637 -528.4479398 -421.7485
341
##          59          60          61          62          63          
64
##  -42.9881766  211.1268364 -121.6908035 -202.3962767  105.5347366  101.1097
936
##          65          66          67          68          69          
70
##  -70.0381043  -45.4946356    4.5551237  -22.3643169  114.6934848 -195.9261
524
##          71          72          73          74          75          
76
##  -18.5964303  121.1995838 -291.3807455   15.1157684  181.4812661  -64.1711
931
##          77          78          79          80          81          
82
## -279.1829807 -106.9523524  147.9797087  -45.7956741  167.0288196  -39.5104
778
##          83          84          85          86          87          
88
## -162.2146505  147.9106621  -59.0602381  -92.5298652  -39.4770262  -32.6986
322
##          89          90          91          92          93          
94
##    0.8672666  -72.2575714 -147.1567391 -105.7359725 -182.1073837  406.3431
483
##          95          96          97          98          99          
100
##  -81.5781285  -26.8740419  125.3551746  -56.5175164   -1.3011635  103.5702
017
```

```
##          101          102          103          104          105          
106 
## -372.7284424  -23.0985428   90.1244980   53.0840211    7.8873218  132.1393
852 
##          107          108          109          110          111          
112 
##    2.6055485  144.1225404  201.1664425  197.5100293 -155.3548309  219.4768
858 
##          113          114          115          116          117          
118 
## -188.5998957  218.3978730 -238.8652271  300.7775017  114.6240091  233.7105
258 
##          119          120          121          122          123          
124 
##  -94.2596437    1.4132487 -157.3542621  211.6735517   14.9748492   92.2186
676 
##          125          126          127          128          129          
130 
##   52.3041849  -65.9050503  219.4415603  -11.2335502 -276.3418721  252.8734
921 
##          131          132          133          134          135          
136 
## -210.6538358   99.3772381  232.4005317  414.8464034   10.7167793  -29.8875
024 
##          137          138          139          140          141          
142 
##  -17.3380471 -315.8647182  278.5153221 -373.7783938 -167.6006659  198.5416
018 
##          143          144          145          146          147          
148 
##  -63.0372814  162.2426192 -234.1890175  -74.9078850  131.4875964   89.9892
124 
##          149          150          151          152          153          
154 
##  106.1482597 -123.3199430  319.1475265 -126.0003042  -37.2455276 -149.9026
814 
##          155          156          157          158          159          
160 
##   58.2726714  210.5394054 -103.3880990   26.0480610  136.1887682  104.8440
667 
##          161 
## -180.3414982
```

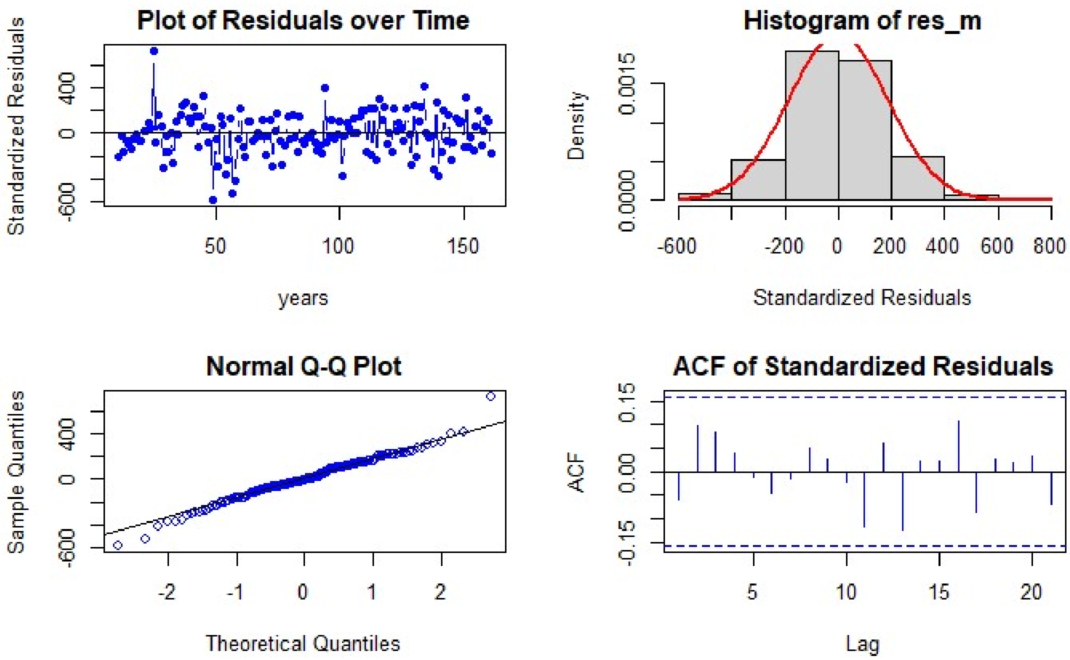

Fig 20: Residual Analysis Auto Regressive Distributed Lag Model.

Residual Analysis for Auto Regressive DLM:

1. The data points are below the line at both the start and end of the trend. Randomness is not seen here.
2. From normal distribution curve, the distribution is almost symmetric. This suggests a good fit with a very few data falling outside the normal curve indicating Kurtosis.
3. The data at the tails is deviated but is normal for most part of the line suggesting normality in the trend.
4. There are no significant lags in Autocorrelation plot suggesting that the stochastic component is white noise.

Even though Auto Regressive DLM shown better performance, Koyck model fits better with 94.85% variance.

Also, let us check with respect to Gold series. Since, it has the second highest auto correlation value.

DLM models on ASX price index W.R.T GOLD price series.

Finite distributed lag model

```
x = v_GOLD_price_TS # Independent variable
y = v_ASX_price_TS # Dependent variable


for ( i in 1:10){
```

```
  model_1 = dlm(x = as.vector(x) , y = as.vector(y), q = i )
  cat("q = ", i, "AIC = ", AIC(model_1$model), "BIC = ", BIC(model_1$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
## q =  7 AIC =  2510.535 BIC =  2540.905
## q =  8 AIC =  2493.885 BIC =  2527.22
## q =  9 AIC =  2476.983 BIC =  2513.27
## q =  10 AIC =  2460.345 BIC =  2499.57
```

As we have the least AIC and BIC values at q = 10. Let us fit the finite distributed lag model with q = 10.

```
# Finite lag length based on AIC-BIC

finite_dlm_GOLD = dlm( x = as.vector(x) , y = as.vector(y), q = 10)
summary(finite_dlm_GOLD)

##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1535.24  -575.79    20.89   480.32  1951.02
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4523.02779  225.83961  20.028   <2e-16 ***
## x.t           -0.54891    1.27022  -0.432    0.666
## x.1            0.07699    1.88146   0.041    0.967
## x.2           -0.01009    1.90952  -0.005    0.996
## x.3           -0.12278    1.92437  -0.064    0.949
## x.4           -0.30955    1.92889  -0.160    0.873
## x.5            0.47310    1.93180   0.245    0.807
## x.6            0.02590    1.94990   0.013    0.989
## x.7            0.67162    1.95391   0.344    0.732
## x.8           -0.11584    1.94844  -0.059    0.953
## x.9            0.11415    1.92690   0.059    0.953
## x.10           0.11352    1.28818   0.088    0.930
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 798.9 on 139 degrees of freedom
```

```
## Multiple R-squared:  0.05296,    Adjusted R-squared:  -0.02199 
## F-statistic: 0.7066 on 11 and 139 DF,  p-value: 0.7306
## 
## AIC and BIC values for the model:
##        AIC     BIC
## 1 2460.345 2499.57
```

Hypotheses:

H_0: The data doesn't fit the Finite distributed lag model.

H_A: The data fits the Finite distributed lag model.

Interpretations:

- F - statistic is 0.7066
- R - squared is 0.05296
- Adjusted R - squared is -0.02199
- Degrees of freedom - DF are (11, 139)
- p - value (0.7306) is > 0.05 and therefore, it is not statistically significant. Therefore, Null hypothesis cannot be rejected.
- Also, this model suggests that there is only 5.3% of data variance. Suggesting that the model explains only 5.3% of the trend. Hence, the model doesn't fit the Finite distributed lag model.

Polynomial distributed lag model

```
for (i in 1:3){
  model_2 <-  polyDlm(x = as.vector(x) , y = as.vector(y), q = i , k = i, sho
w.beta = FALSE)
  cat("q = ", i, "k = ", i, "AIC = ", AIC(model_2$model), "BIC = ", BIC(model
_2$model),"\n")
}

## q =  1 k =  1 AIC =  2613.609 BIC =  2625.91 
## q =  2 k =  2 AIC =  2596.292 BIC =  2611.637 
## q =  3 k =  3 AIC =  2579.215 BIC =  2597.59
```

Let us fit a polynomial model of order 3. Since least AIC and BIC scores.

```
# Ploynomial DLM

PolyDLM_model_GOLD = polyDlm(x = as.vector(x), y = as.vector(y), q = 3, k = 3
, show.beta = TRUE)

## 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
```

```
summary(PolyDLM_model_GOLD)

##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1579.25  -662.06   -12.23   540.91  2198.76
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4058.2998   213.4589  19.012   <2e-16 ***
## z.t0           0.1575     1.2840   0.123    0.903
## z.t1          -0.3044     9.6816  -0.031    0.975
## z.t2           0.1589     8.8853   0.018    0.986
## z.t3          -0.0102     1.9660  -0.005    0.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 829.8 on 153 degrees of freedom
## Multiple R-squared:  0.0928, Adjusted R-squared:  0.06908
## F-statistic: 3.913 on 4 and 153 DF,  p-value: 0.004707
```

Hypotheses:

H0: The data doesn't fit the Polynomial distributed lag model.
HA: The data fits the Polynomial distributed lag model.

Interpretations:

- F - statistic is 3.943
- R - squared is 0.09345
- Adjusted R - squared is 0.06975
- Degrees of freedom - DF are (4, 153)
- p - value (0.004482) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Polynomial distributed lag model.

This model suggests that there is only 9.35% of data variance. Suggesting that the model explains only 9.35% of the trend. Which implies that the model shows some trend.

Residual analysis

```
res_analysis(residuals(PolyDLM_model_GOLD$model))
```

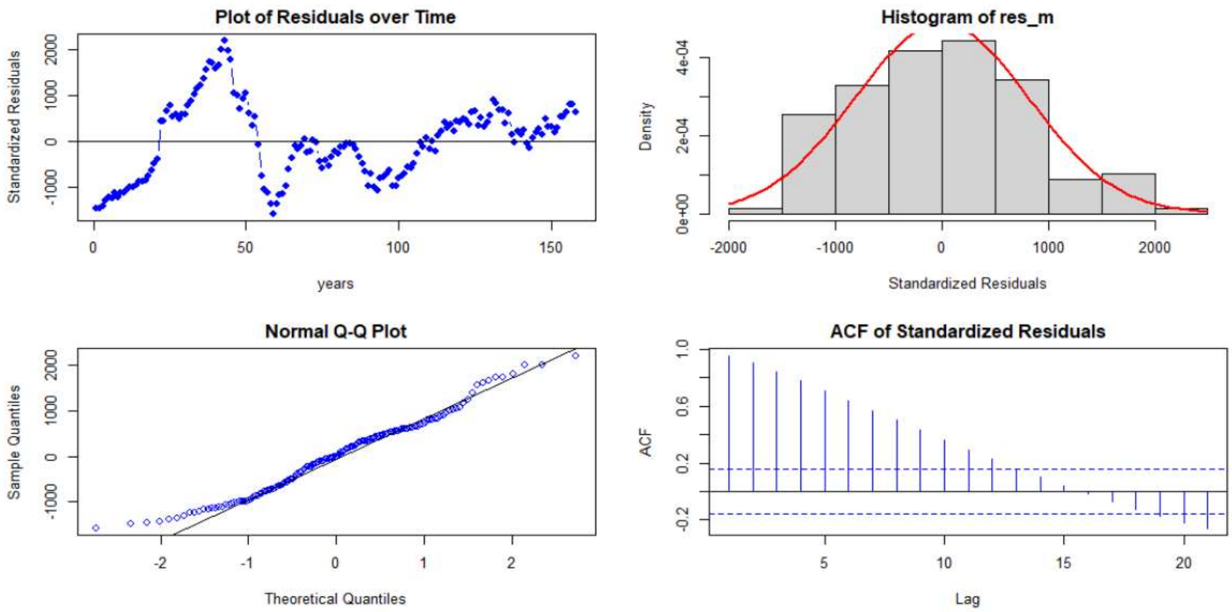

Fig 21: Residual Analysis Polynomial Distributed Lag Model.

Residual Analysis for Polynomial DLM:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric and requires some transformation to make it symmetric. This suggests the non - stationary in the series.
3. The data at the tails is deviated more leaving some part on the line suggesting there is no normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.

This analysis is not enough, and we still require a better model than this. Therefore, let us fit Koyck model.

Koyck model

```
# Koyk DLM

Koyck_DLM_GOLD = koyckDlm(x = as.vector(x) , y = as.vector(y))
summary(Koyck_DLM_GOLD)

##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -682.19 -105.44   15.86  135.04  783.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.902e+02  8.958e+01   2.123   0.0353 *
## Y.1         9.635e-01  1.909e-02  50.469   <2e-16 ***
## X.t         2.595e-03  4.304e-02   0.060   0.9520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.4 on 157 degrees of freedom
## Multiple R-Squared: 0.9488,  Adjusted R-squared: 0.9481
## Wald test:  1454 on 2 and 157 DF,  p-value: < 2.2e-16
##
## Diagnostic tests:
## NULL
##
##                              alpha        beta       phi
## Geometric coefficients:  5205.15 0.002595168 0.9634602
```

Hypotheses:

H_0: The data doesn′t fit the Koyck distributed lag model.
H_A: The data fits the Koyck distributed lag model.

Interpretations:

- Walt test - statistic is 1454
- R - squared is 0.9488
- Adjusted R - squared is 0.9481
- Degrees of freedom - DF are (2, 157)
- p - value (0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Koyck distributed lag model.

This model suggests that there is only 94.88% of data variance. Suggesting that the model explains only 94.88% of the trend. Which implies that the model performs better on the series data.

Now let us perform residual analysis.

Residual analysis

```
res_analysis(residuals(Koyck_DLM_GOLD))
```

```
##          2          3          4          5          6          7
## -241.501847  -19.941014  34.794408  -74.164805  -62.804566  25.037109
##          8          9         10         11         12         13
##   18.090550 -101.347005  30.647193 -158.368909  35.444407  -93.187885
##         14         15         16         17         18         19
##   17.335147  -24.472523  -75.470243  -17.894108   8.052784  -46.841840
```

```
##         20         21         22         23         24         25 
##  -46.281120  51.258059  54.190387 103.176531  62.675283 783.597749 
##         26         27         28         29         30         31 
##  -15.106929 195.430673 114.025923 -236.178619  51.508989 -85.056058 
##         32         33         34         35         36         37 
##  111.645019  26.326113 234.347452 112.043882 189.961258 127.353685 
##         38         39         40         41         42         43 
##   76.785963 182.495229 205.640687 216.230979   8.313365 -84.697870 
##         44         45         46         47         48         49 
##   94.609100 368.531583 246.291461 -130.325698 -124.255990 -682.191909 
##         50         51         52         53         54         55 
##   -6.949305 -250.561461 252.136641 131.076974 -422.745277 -278.070154 
##         56         57         58         59         60         61 
##  154.859912 -586.453849 -672.623401 -357.673534 -72.558393 -240.994134 
##         62         63         64         65         66         67 
## -248.074746 162.069818 148.039796  12.088043  80.582907 252.743702 
##         68         69         70         71         72         73 
##  196.729778 225.851923 -112.408332  45.021792 146.051604 -300.757456 
##         74         75         76         77         78         79 
##   28.766257 218.588235 -73.818187 -397.474250 -160.014100 146.896532 
##         80         81         82         83         84         85 
##  -97.603017 166.584126  72.188400 -77.834041 147.536738 -13.629163 
##         86         87         88         89         90         91 
##   57.186684  -8.943604 -43.328466 -124.964282 -148.048039 -183.015411 
##         92         93         94         95         96         97 
## -160.690266 -334.811584 244.666577 -211.161442 -115.196818 170.590812 
##         98         99        100        101        102        103 
##   26.052338  -2.081939  14.374857 -364.592680 -41.490375 110.797629 
##        104        105        106        107        108        109 
##   12.110461  31.296102  95.506440 -46.166498 117.311637 212.518196 
##        110        111        112        113        114        115 
##  204.189418 -147.593581 176.757270 -259.640926 107.370579 -270.138152 
##        116        117        118        119        120        121 
##  330.327618  85.711763 199.468922 -101.692046  39.250827 -146.241724 
##        122        123        124        125        126        127 
##  206.536201  -8.533981  71.409639   9.116362 -85.528604 243.937892 
##        128        129        130        131        132        133 
##   13.159571 -316.023855 207.954305 -199.494856  90.121425 165.678620 
##        134        135        136        137        138        139 
##  355.471542 -15.224607 -68.246721  18.061000 -306.857793 235.534004 
##        140        141        142        143        144        145 
## -446.159408 -167.019441 220.472049 -71.294648 123.049215 -286.974805 
##        146        147        148        149        150        151 
## -118.495668 190.185918 158.042262 131.375051 -133.007967 332.835999 
##        152        153        154        155        156        157 
## -103.123725   3.113990 -115.422781 102.945152 223.506743 -29.463322 
##        158        159        160        161 
##   98.988163 158.918386  64.963948 -163.517963
```

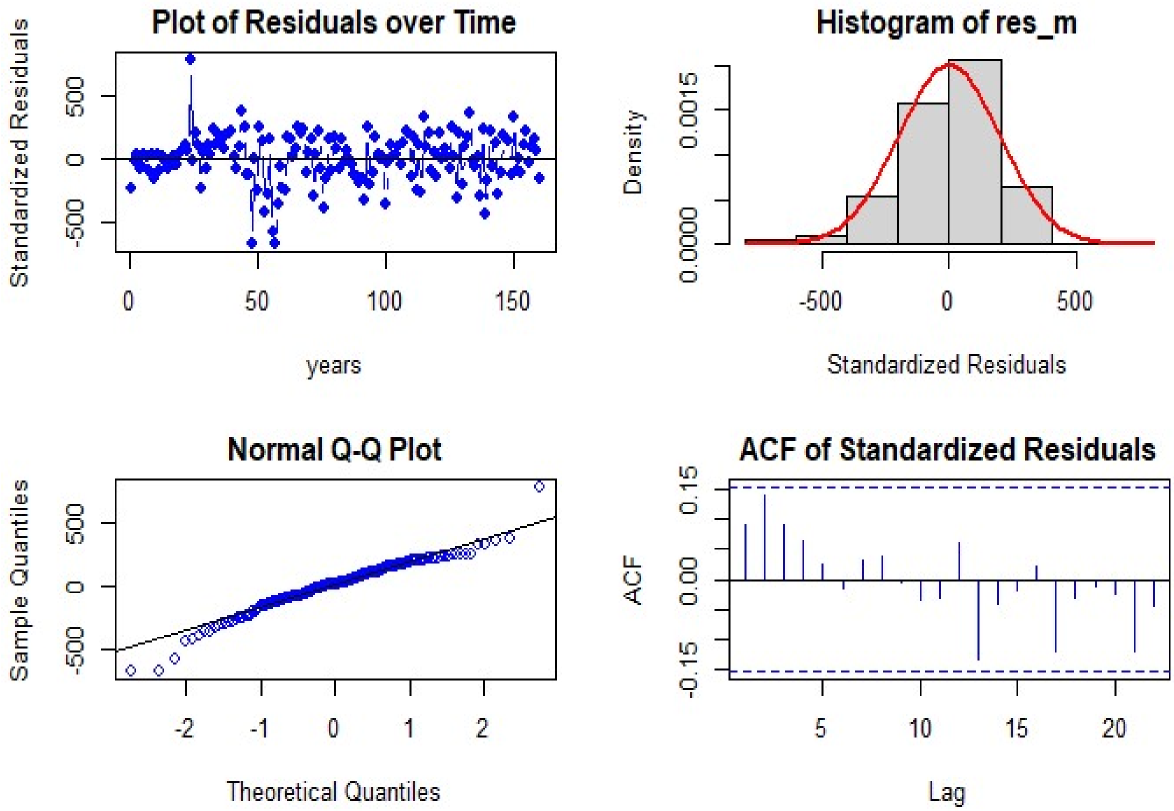

Fig 22: Residual Analysis Koyck Distributed Lag Model.

Residual Analysis for Koyck DLM:

1. The data points are below the line at both the start and end of the trend. Randomness is not seen here.
2. From normal distribution curve, the distribution is almost symmetric. This suggests a good fit with a very few data falling outside the normal curve indicating Kurtosis.
3. The data at the tails is deviated but is normal for most part of the line suggesting normality in the trend.
4. There are no significant lags in Autocorrelation plot suggesting that the stochastic component is white noise.

So far this is the best model but let us fit ardlDlm model to check whether it fits better than Koyck model or not.

Autoregressive distributed lag model

```
for (i in 1:10){
  for(j in 1:5){
    model_4 = ardlDlm(x = as.vector(x) , y = as.vector(y), p = i , q = j )
    cat("p = ", i, "q = ", j, "AIC = ", AIC(model_4$model), "BIC = ", BIC(mod
el_4$model),"\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
## p =  6 q =  1 AIC =  2083.087 BIC =  2113.521
## p =  6 q =  2 AIC =  2084.993 BIC =  2118.471
## p =  6 q =  3 AIC =  2081.777 BIC =  2118.298
## p =  6 q =  4 AIC =  2083.115 BIC =  2122.68
## p =  6 q =  5 AIC =  2084.976 BIC =  2127.584
## p =  7 q =  1 AIC =  2072.69 BIC =  2106.097
## p =  7 q =  2 AIC =  2074.588 BIC =  2111.032
## p =  7 q =  3 AIC =  2071.471 BIC =  2110.952
## p =  7 q =  4 AIC =  2072.806 BIC =  2115.324
## p =  7 q =  5 AIC =  2074.667 BIC =  2120.221
## p =  8 q =  1 AIC =  2060.657 BIC =  2097.022
## p =  8 q =  2 AIC =  2062.526 BIC =  2101.922
## p =  8 q =  3 AIC =  2059.768 BIC =  2102.194
## p =  8 q =  4 AIC =  2060.894 BIC =  2106.35
## p =  8 q =  5 AIC =  2062.836 BIC =  2111.323
## p =  9 q =  1 AIC =  2046.919 BIC =  2086.229
## p =  9 q =  2 AIC =  2048.65 BIC =  2090.985
## p =  9 q =  3 AIC =  2046.025 BIC =  2091.383
## p =  9 q =  4 AIC =  2046.982 BIC =  2095.364
## p =  9 q =  5 AIC =  2048.757 BIC =  2100.163
## p =  10 q =  1 AIC =  2036.551 BIC =  2078.793
```

```
## p =  10 q =  2 AIC =  2038.268 BIC =  2083.528
## p =  10 q =  3 AIC =  2035.644 BIC =  2083.92
## p =  10 q =  4 AIC =  2036.587 BIC =  2087.88
## p =  10 q =  5 AIC =  2038.35 BIC =  2092.661
```

p = 10 and q = 1 has the least AIC and BIC scores.

```
# ARDLM model

AR_DLM_GOLD = ardlDlm(x = as.vector(x) , y = as.vector(y), p = 10 , q = 1 )
summary(AR_DLM_GOLD)

##
## Time series regression with "ts" data:
## Start = 11, End = 161
##
## Call:
## dynlm(formula = as.formula(model.text), data = data, start = 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -577.16 -104.00    3.19  123.44  692.04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 251.32788  106.98266   2.349  0.02023 *
## X.t          -1.32788    0.31171  -4.260 3.76e-05 ***
## X.1           1.22622    0.46170   2.656  0.00884 **
## X.2           0.01481    0.46792   0.032  0.97479
## X.3          -0.10766    0.47156  -0.228  0.81975
## X.4          -0.30652    0.47267  -0.648  0.51775
## X.5           0.84867    0.47345   1.793  0.07524 .
## X.6          -0.60306    0.47801  -1.262  0.20922
## X.7           0.67531    0.47880   1.410  0.16066
## X.8          -0.99071    0.47783  -2.073  0.04000 *
## X.9           0.56889    0.47228   1.205  0.23044
## X.10         -0.02612    0.31568  -0.083  0.93418
## Y.1           0.96236    0.02063  46.656  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.8 on 138 degrees of freedom
## Multiple R-squared:  0.9435, Adjusted R-squared:  0.9386
## F-statistic: 192.2 on 12 and 138 DF,  p-value: < 2.2e-16
```

Hypotheses:

H_0: The data doesn't fit the Autoregressive distributed lag model.
H_A: The data fits the Autoregressive distributed lag model.

Interpretations:

- F - statistic is 194.9
- R - squared is 0.9435
- Adjusted R - squared is 0.9386
- Degrees of freedom - DF are (12, 138)
- p - value (0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Autoregressive distributed lag model.

This model suggests that there is only 94.35% of data variance. Suggesting that the model explains only 94.35% of the trend. Which implies that the model shows some trend.

Let us perform residual analysis on this model.

Residual analysis

```
res_analysis(residuals(AR_DLM_GOLD))

## Time Series:
## Start = 11
## End = 161
## Frequency = 1
##           11           12           13           14           15
16
## -201.2390881   -3.4773197 -144.1721026  -66.3819403  -36.3127014 -116.4198
852
##           17           18           19           20           21
22
##  -44.2162022  -17.7380897  -87.3200898  -78.1046315   38.9515009   -0.1158
940
##           23           24           25           26           27
28
##   61.8817397   45.0861630  692.0405257  -34.6406303  149.9144784   88.5401
013
##           29           30           31           32           33
34
## -240.3521831  105.7469445  -17.8727754  167.8403439  151.1742431  198.2380
282
##           35           36           37           38           39
40
##   45.7366623  189.7363529  135.1809531   30.3179711  176.4898353  151.9974
716
##           41           42           43           44           45
46
##  234.9216984  -41.5154524 -197.9216435  139.2993535  354.7192247  240.9378
974
##           47           48           49           50           51
52
##  -76.4538363  -95.4598999 -577.1628059  -29.7908302 -198.9178025  142.6129
432
##           53           54           55           56           57
58
```

```
##   117.2790747 -509.1100015 -202.0185587   13.0499053 -465.1765959 -489.9860
494
##            59           60           61           62           63
64
## -330.3216937  -43.0700463 -126.0379194    2.9485130   69.1834875  -11.6248
336
##            65           66           67           68           69
70
##  -74.2598584  165.9060949   54.5609150  140.4542129  194.3081114  -55.9271
795
##            71           72           73           74           75
76
##  141.3292931  102.7104642 -339.4529645   14.0843656  226.9475683 -109.1446
845
##            77           78           79           80           81
82
## -206.6487125 -102.7599214   47.6012016  -58.1744227  195.7745494   49.3909
563
##            83           84           85           86           87
88
## -111.0952351  142.0957447  -32.0582927  107.6291365    3.1948962  -76.1791
361
##            89           90           91           92           93
94
## -134.3995717 -103.9112833 -152.2865721  121.0861953 -238.3836291  120.5497
557
##            95           96           97           98           99
100
##  -41.1166916 -144.6827117   43.7535942   54.4191557  -68.2465758    0.9376
417
##           101          102          103          104          105
106
## -274.9011232 -124.1617334   89.5437129   -4.9617717  143.0758142  154.7200
243
##           107          108          109          110          111
112
## -142.0409544   97.1067500  221.3119690  155.3026481 -241.4058348  -37.9766
222
##           113          114          115          116          117
118
## -229.5822360   91.9271040 -345.3178975  375.3628010   57.2752472   79.5026
873
##           119          120          121          122          123
124
## -104.0840629   -5.3667868 -166.2536265  258.7687001  -23.2980844   22.2306
959
##           125          126          127          128          129
130
##   12.2676277 -150.4848076  256.7233384  -61.5443116 -330.0018822  222.5969
```

```
453
##           131          132          133          134          135          
136
## -191.9031121  141.9733325  293.2727331  384.3126877  -74.9500836   34.9134
962
##           137          138          139          140          141          
142
##  -44.0934078 -281.8841263  149.6407815 -407.9804338  -81.3726683  289.2763
956
##           143          144          145          146          147          
148
## -240.7039552   93.3776883 -173.9722038   10.7557846  125.7989355  109.8102
475
##           149          150          151          152          153          
154
##  347.2542476  -80.1461200  339.3249824 -157.9966848   50.4672704 -200.3355
514
##           155          156          157          158          159          
160
##  112.9116274    7.6170570   59.4861478   39.7518466  205.6933992   98.5656
200
##           161
##  -94.1005839
```

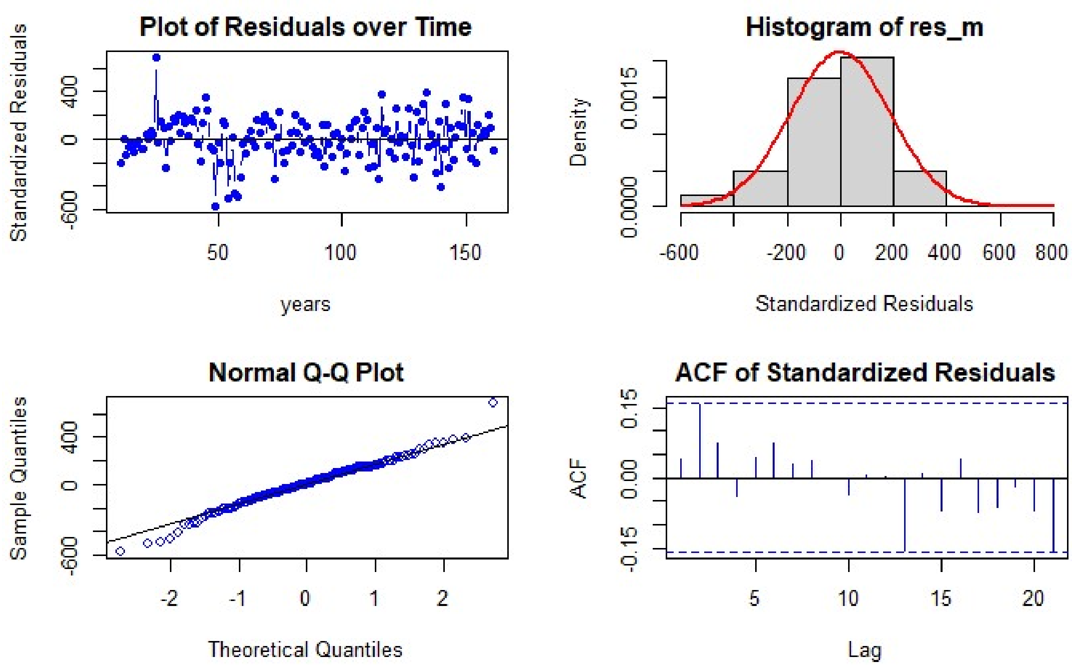

Fig 23: Residual Analysis Auto Regressive Distributed Lag Model.

Residual Analysis for Auto Regressive DLM:

1. The data points are below the line at both the start and end of the trend. Randomness is not seen here.
2. From normal distribution curve, the distribution is almost symmetric. This suggests a good fit with a very few data falling outside the normal curve indicating Kurtosis.
3. The data at the tails is deviated but is normal for most part of the line suggesting normality in the trend.
4. There are no significant lags in Autocorrelation plot suggesting that the stochastic component is white noise.

Even though Auto Regressive DLM shown better performance, Koyck model fits better with 94.88% variance.

Finally, the Koyck model with Gold got 94.88% R - Squared. Where-as, with respect to Copper it is 94.85%. But the higher Correlation coefficient in Copper making it the best model. But let us check it by finding the multi-collinearity.

```
vif(Koyck_DLM_copper$model) > 10

##   Y.1   X.t
## FALSE FALSE

vif(Koyck_DLM_GOLD$model) > 10

##   Y.1   X.t
## FALSE FALSE
```

Both the models don't suffer from multi-collinearity. But Correlation coefficient being the crucial factor Copper series should be considered on top of Gold series. Hence, Koyck model with copper will be a better model.

Overall, it is suggesting that Koyck DLM is the best fit model among all DL models.

Conclusion

Finally, we can conclude that,

1. The series data is non - stationary.
2. The components like trend, remainder and seasonality effected the stationarity of the series data.
3. The most accurate and suitable DL model is Koyck distributed lag model.

References

[1] cran, [Online]. Available: https://cran.r-project.org/web/packages/expsmooth/index.html.

[2] Cran, [Online]. Available: https://cran.r-project.org/web/packages/forecast/index.html.

[3] Cran, [Online]. Available: https://cran.r-project.org/web/packages/tseries/index.html.

[4] Cran, [Online]. Available: https://cran.r-project.org/web/packages/fUnitRoots/index.html.

[5] Cran, [Online]. Available: https://cran.r-project.org/web/packages/urca/index.html.

[6] Week 2 and 3 module tasks.

[7] My previous works for Time series Analysis in my Second year Second Sem.