MATH1318 Time Series Analysis / MATH2204 Time Series and Forecasting Final Project Report

Declaration of contributions:

No	Name of Team Member	Contribution to the project
1	Duy Phong Thach	1
2		
3		
4		
5		
6		
	Sum:	must be 1

RMIT University,
School of Science
2022

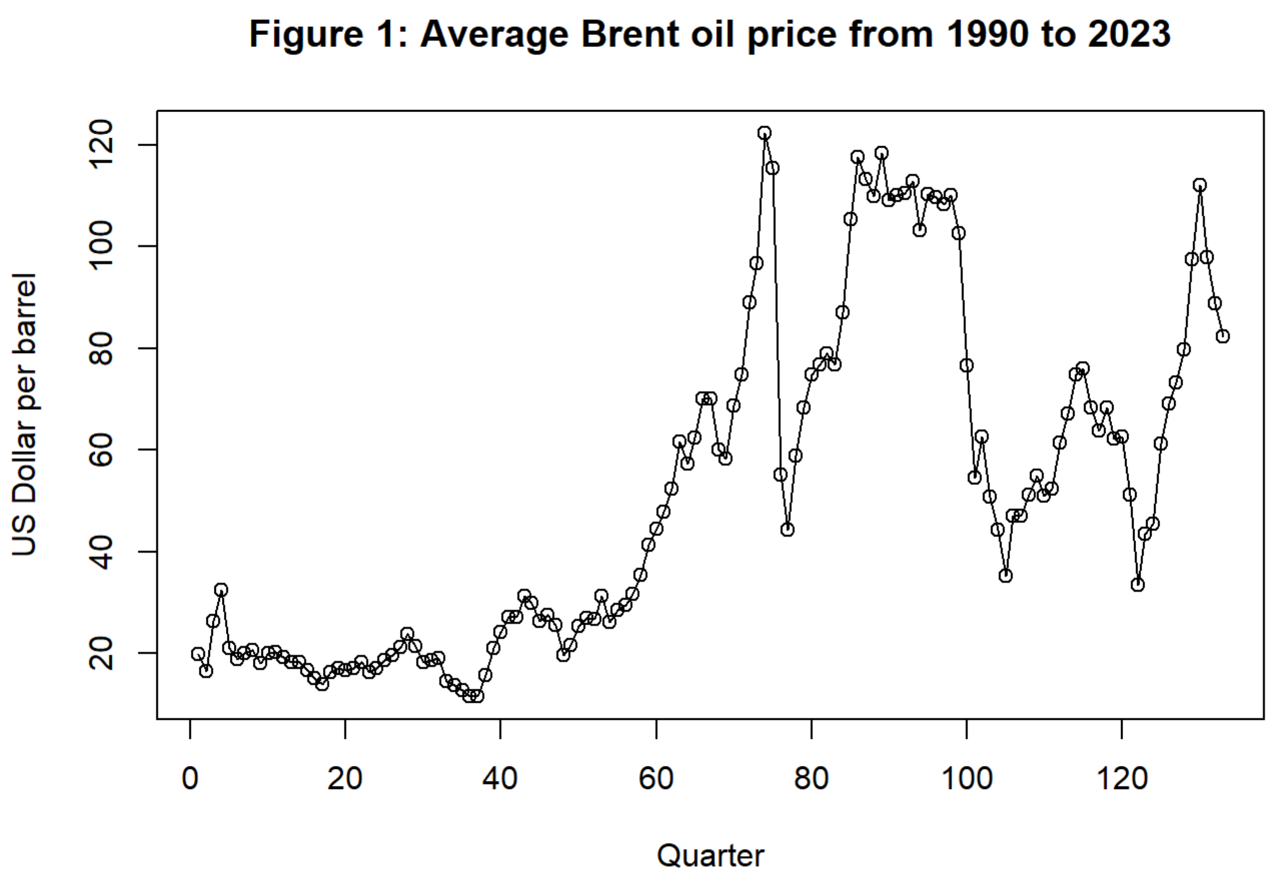
```
library(TSA)
library(tseries)
library(dplyr)
library(FSAdata)
library(lmtest)
library(forecast)
library(zoo)
library(MASS)
library(ggplot2)
```

I.Introduction:

One of the main benchmarks for oil pricing, the price of Brent crude, has seen a noticeable upward trend recently. This upward trend has generated more interest in and speculative thinking about oil prices. Then what is the accurate prediction for oil price in the next 10 quarters. The data of oil price is collected in quarter from FRED: https://fred.stlouisfed.org/series/POILBREUSDQ (https://fred.stlouisfed.org/series/POILBREUSDQ)

1.Decriptive Analysis:

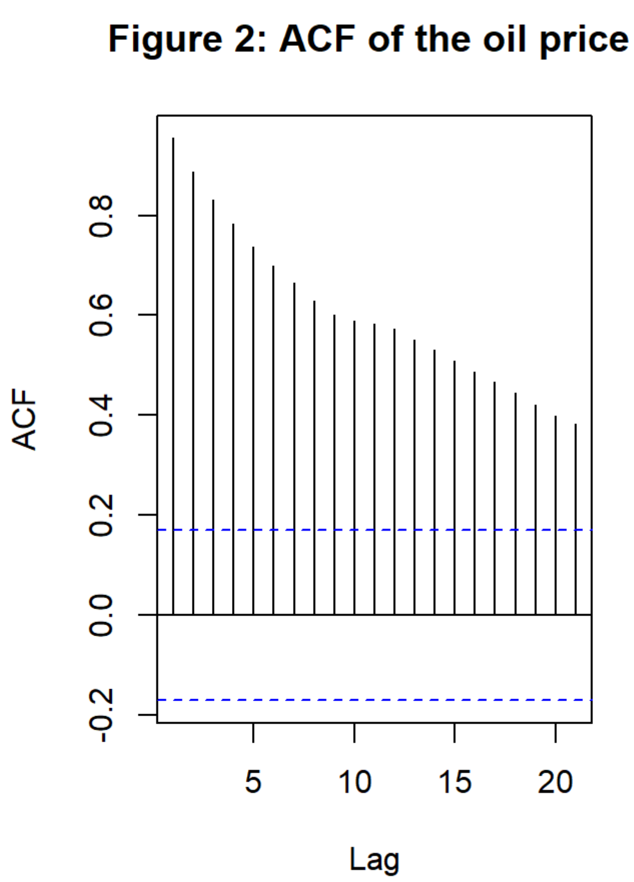
```
plot(Oil_prices.TS, xlab = "Quarter", ylab = "US Dollar per barrel", type = "o",  main = "Fig
ure 1: Average Brent oil price from 1990 to 2023")
```

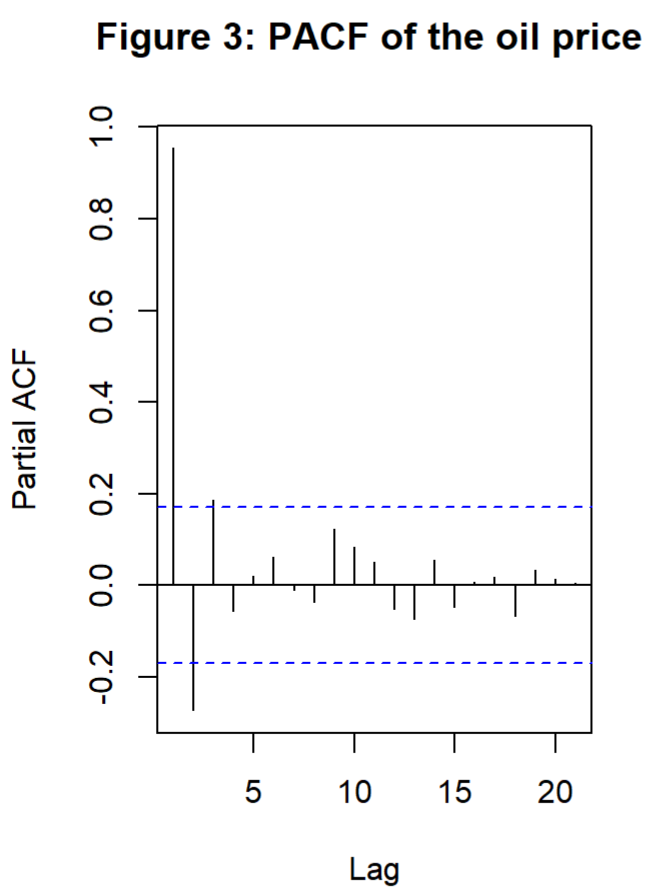


- Overall, it is a linear increasing trend over the period with the quadratic increasing trend can be observed from 1990 to 2008 (quarter 1 to quarter 74).
- The seasonality is not observed until 2008, but the pattern is unclear.
- Overtime, there is increasing variance until 2008, then slowing down in recent quarters.

- The plot shows AR behaviors as it is a frequent consecutive increase or decrease over time.
- Change point from 2008 as from the quadratic trend to seasonality.

```
par(mfrow=c(1,2))
acf(Oil_prices.TS, main = "Figure 2: ACF of the oil price ")

pacf(Oil_prices.TS, main = "Figure 3: PACF of the oil price ")
```






```
par(mfrow=c(1,1))
summary(Oil_prices.TS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   11.50   20.93   46.98   51.62   74.74  122.22
```

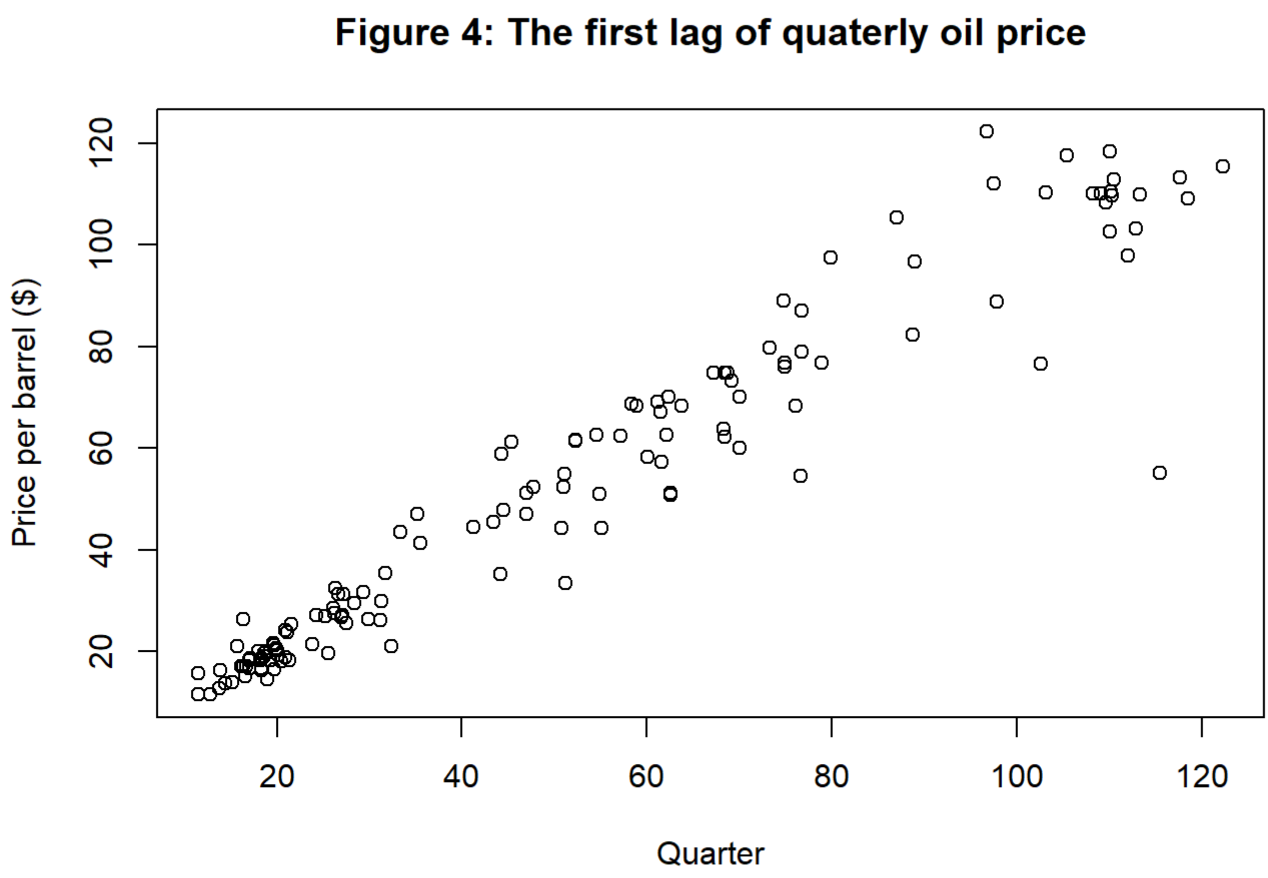
- The slow decreasing pattern of ACF plot (Figure 2) and very high first lag at PACF plot (Figure 3) indicates the trend is detected in time series data making it non-stationary.
- The summary shows the huge difference between minimum and maximum.The median is roughly 10% lower than the mean (46.97 and 51.62). Additionally, the difference of first quartile and min is smaller than the figure of third quartile and max imply the left skewness of the data. Implying the high price of oil most of the time.

2.Lag

```
original.dataset = Oil_prices.TS
data.first.lag = zlag(Oil_prices.TS)
index = 2:length(data.first.lag)
cor(original.dataset[index],data.first.lag[index])
```

```
## [1] 0.9610726
```

```
plot(y = Oil_prices.TS,x = zlag(Oil_prices.TS),ylab='Price per barrel ($)', xlab='Quarter', m
ain = "Figure 4: The first lag of quaterly oil price")
```



- The two variables have a strong positive linear relationship, as indicated by the correlation value of 0.9610726. The number is close to 1, indicating a strong correlation between the data and the initial lag. This significant positive correlation shows that tends to increase along with data as it increases, and vice versa.

II.Fitting

```
t <- time(Oil_prices.TS)
t
```

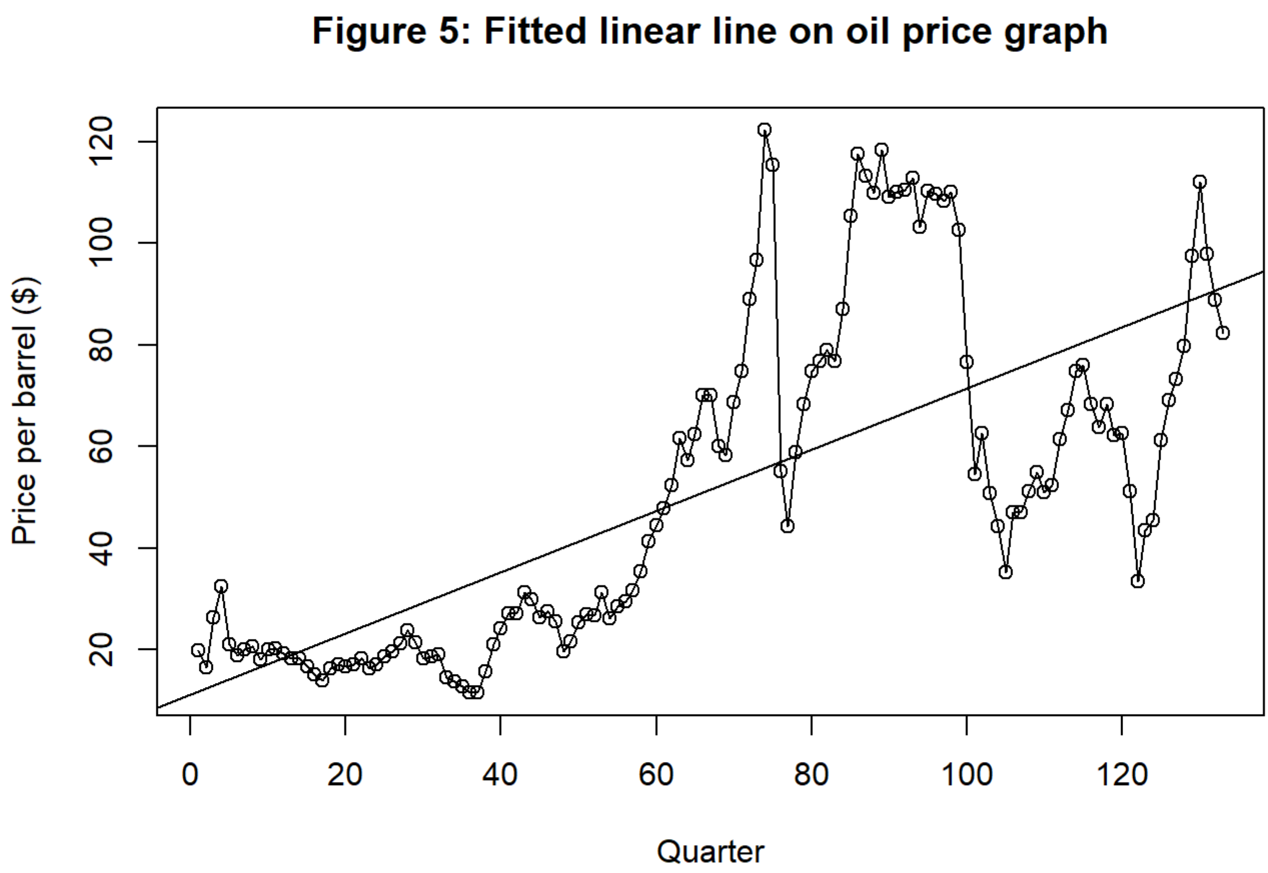
```
## Time Series:
## Start = 1
## End = 133
## Frequency = 1
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133
```

1.Linear

```
Linear_model <- lm(Oil_prices.TS ~ t)
summary(Linear_model)
```

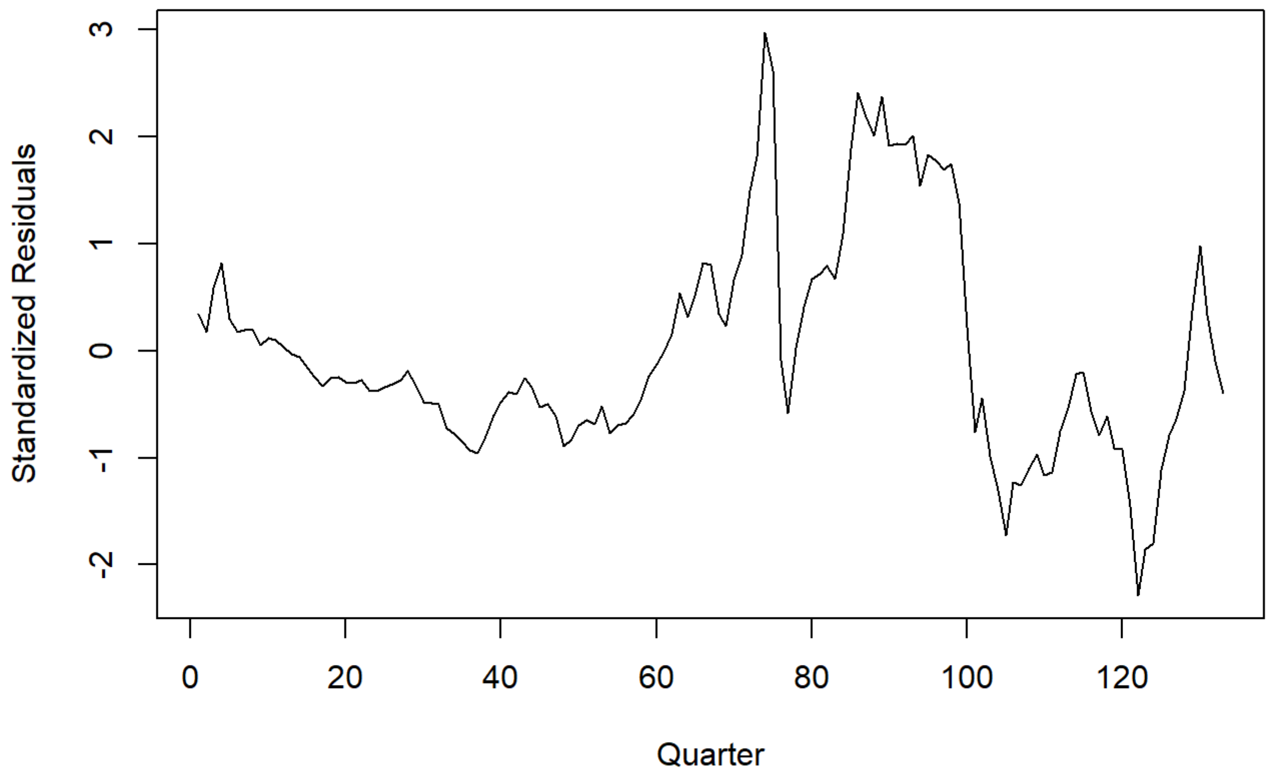
```
## 
## Call:
## lm(formula = Oil_prices.TS ~ t)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -51.411 -15.026  -5.871   9.477  66.374 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 11.22359    4.02892   2.786  0.00613 ** 
## t            0.60298    0.05217  11.557  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 23.1 on 131 degrees of freedom
## Multiple R-squared:  0.5049, Adjusted R-squared:  0.5011 
## F-statistic: 133.6 on 1 and 131 DF,  p-value: < 2.2e-16
```

```
plot(Oil_prices.TS,ylab='Price per barrel ($)', xlab='Quarter',type='o',
     main = "Figure 5: Fitted linear line on oil price graph")
abline(Linear_model)
```



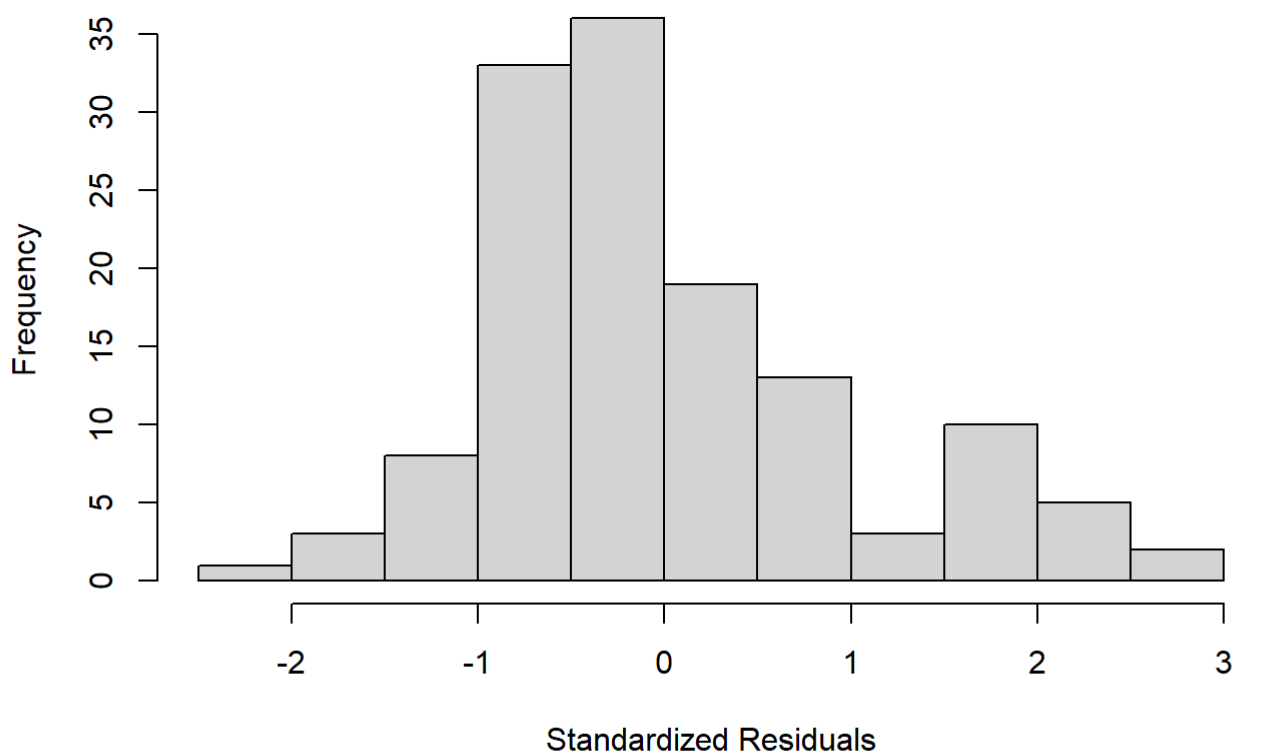

```
res.Linear_model = rstudent(Linear_model)
par(mfrow=c(1,1))
plot(y = res.Linear_model, x = as.vector(t),xlab = 'Quarter', ylab='Standardized Residuals',t
ype='l',main = "Figure 6: Standardised residuals from linear model.")
```

Figure 6: Standardised residuals from linear model.

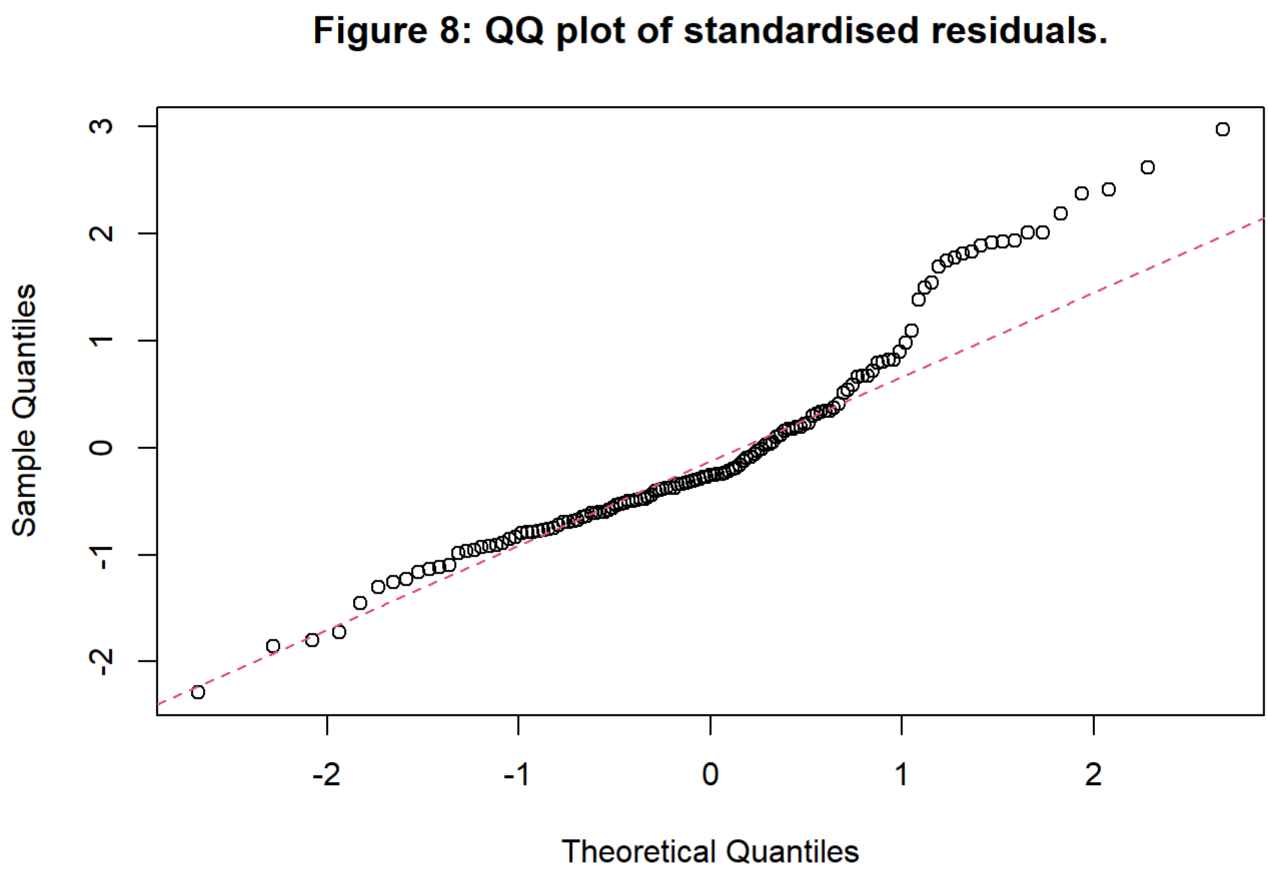



```
hist(res.Linear_model,xlab='Standardized Residuals', main = "Figure 7: Histogram of standardi
sed residuals.")
```

Figure 7: Histogram of standardised residuals.



```
qqnorm(y=res.Linear_model, main = "Figure 8: QQ plot of standardised residuals.")
qqline(y=res.Linear_model, col = 2, lwd = 1, lty = 2)
```

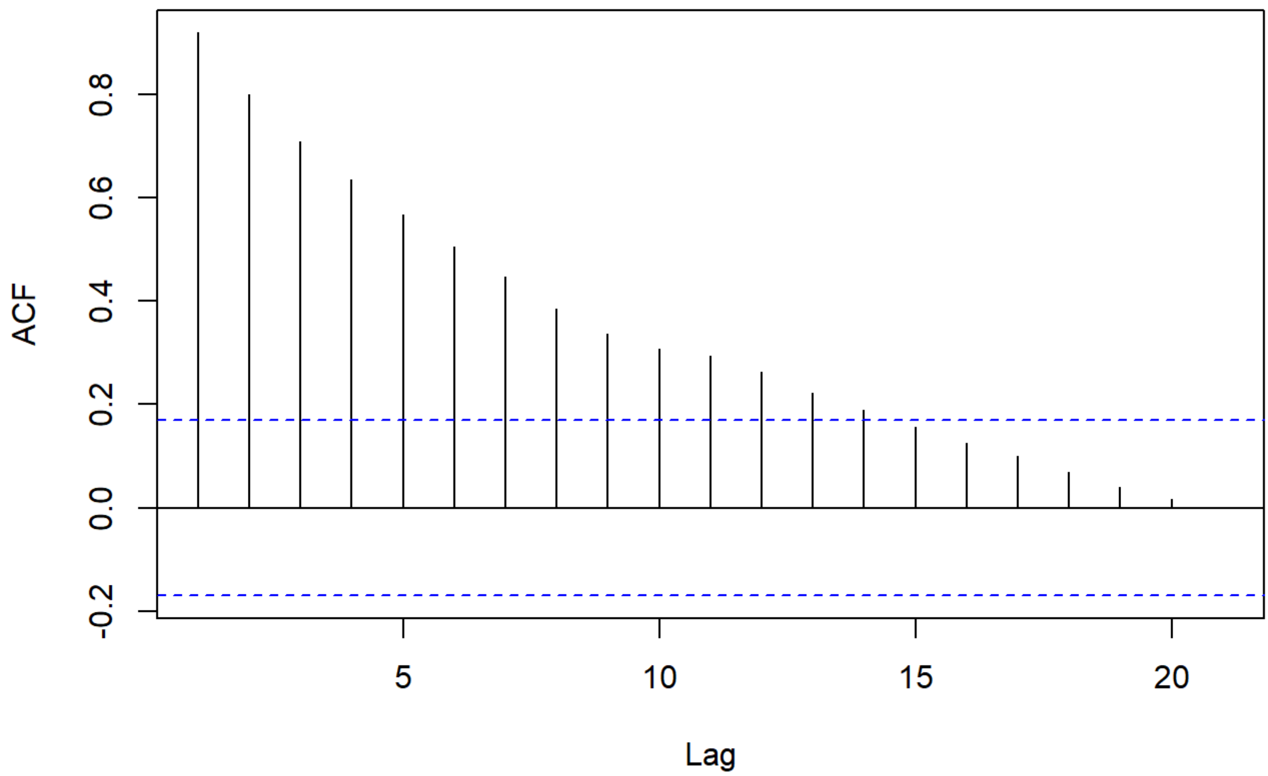



```
shapiro.test(res.Linear_model)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.Linear_model
## W = 0.93542, p-value = 8.095e-06
```

```
acf(res.Linear_model, main = "Figure 9: ACF of standardized residuals.")
```

Figure 9: ACF of standardized residuals.



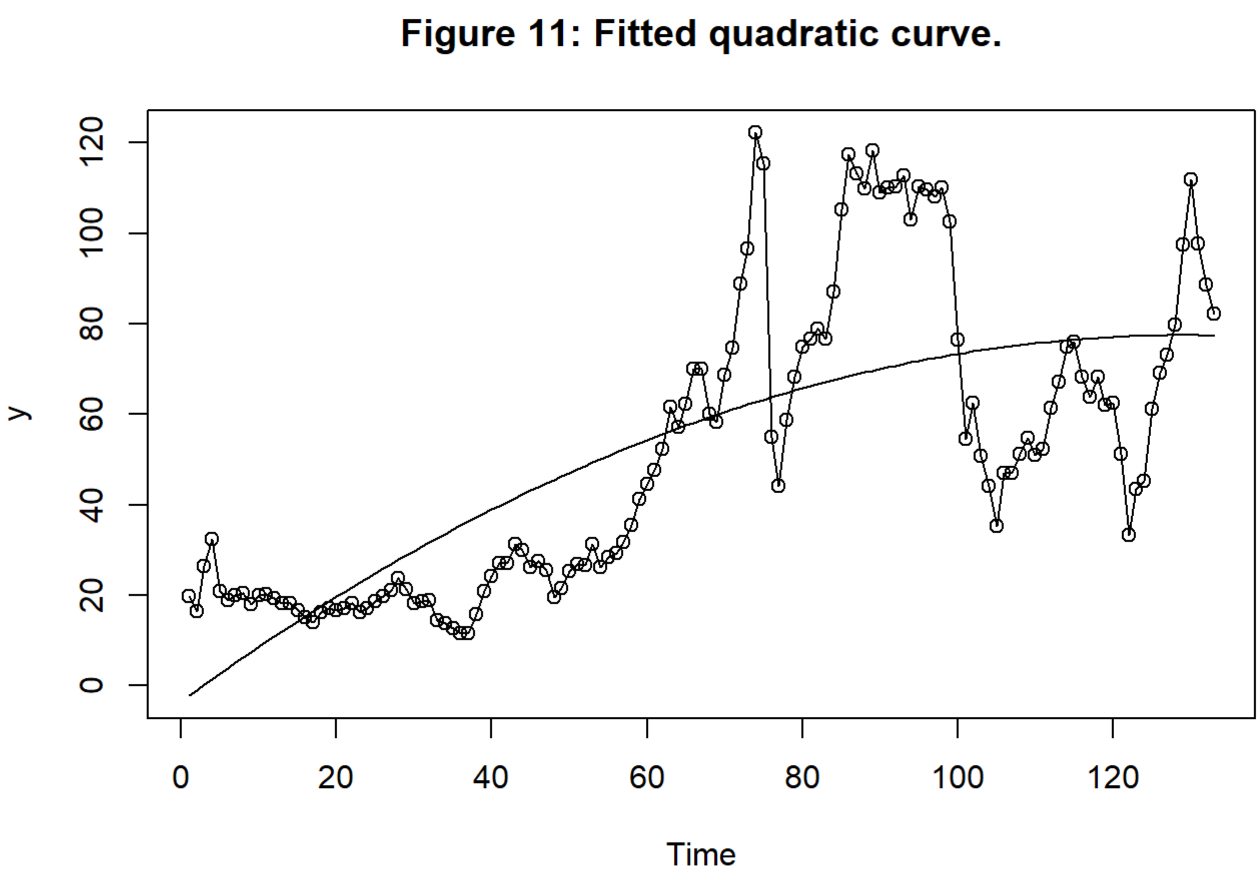
- With very small p-value, time is the significant variable of the model. Meanwhile, the model also significant with small p-value. The R-squared of the model is 0.5011 which mean time can explain 50% of oil price. About the normality of residuals, the plot show no trend with random movement (Figure 6). The histogram show the right skewness of the residuals as the residuals concentrates from -1 to 0 (Figure 7). The QQ plots from Figure 8 shows that the residuals does not follow normal distribution as the residuals in the right tail is far from straight line. The Figure 9 of ACF shows that majority of columns stand outside the confidence interval zone which is not normal distribution. Lastly, the Shapiro-Wilk test indicates that residual is not normaly distributed with p-value < 0.05.

2.Quadratic:

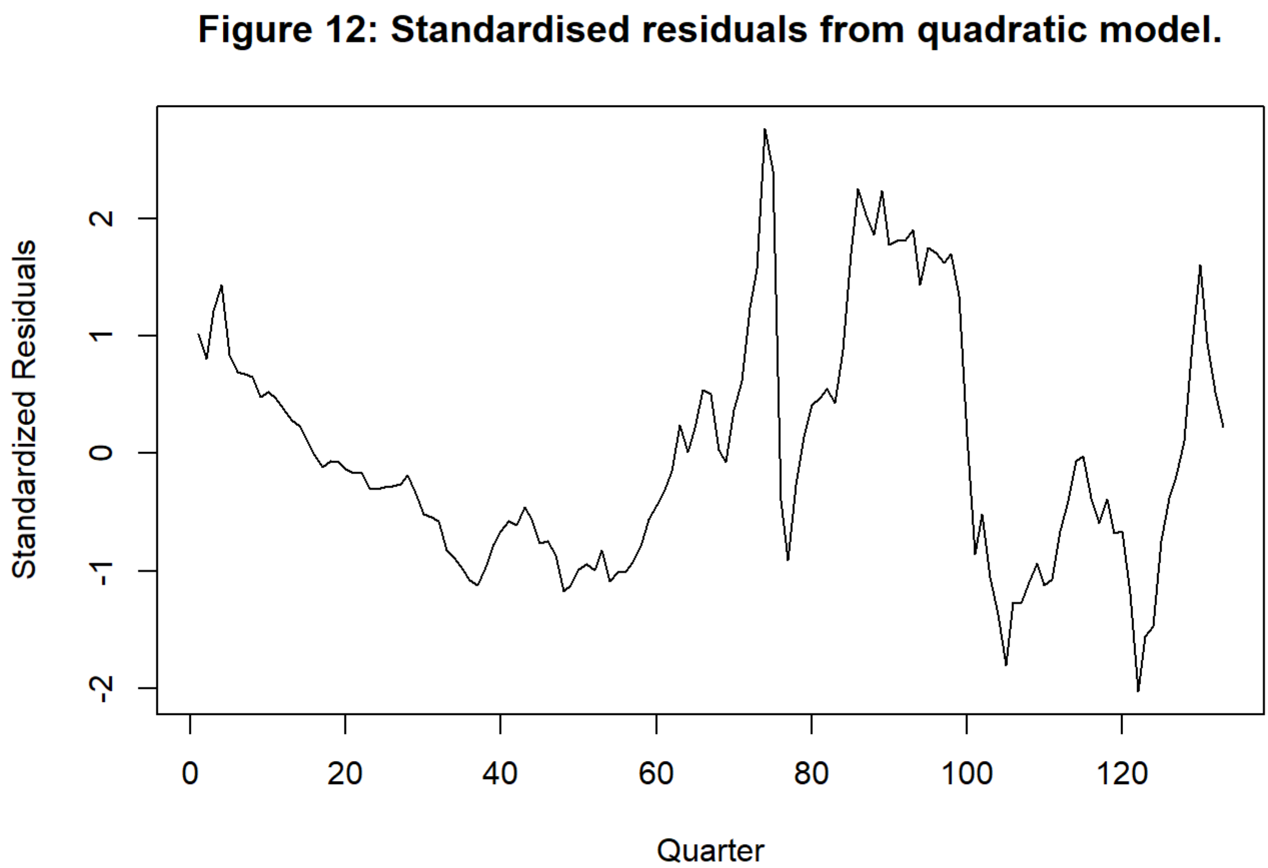
```
t_squared = t^2
Quadratic_model = lm(Oil_prices.TS~ t + t_squared)
summary(Quadratic_model)
```

```
##
## Call:
## lm(formula = Oil_prices.TS ~ t + t_squared)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.875 -17.166  -4.075  11.850  59.451
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.425253   5.880418  -0.582  0.56125
## t            1.254044   0.202596   6.190  7.3e-09 ***
## t_squared   -0.004859   0.001465  -3.317  0.00118 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.27 on 130 degrees of freedom
## Multiple R-squared:  0.5435, Adjusted R-squared:  0.5365
## F-statistic: 77.39 on 2 and 130 DF,  p-value: < 2.2e-16
```

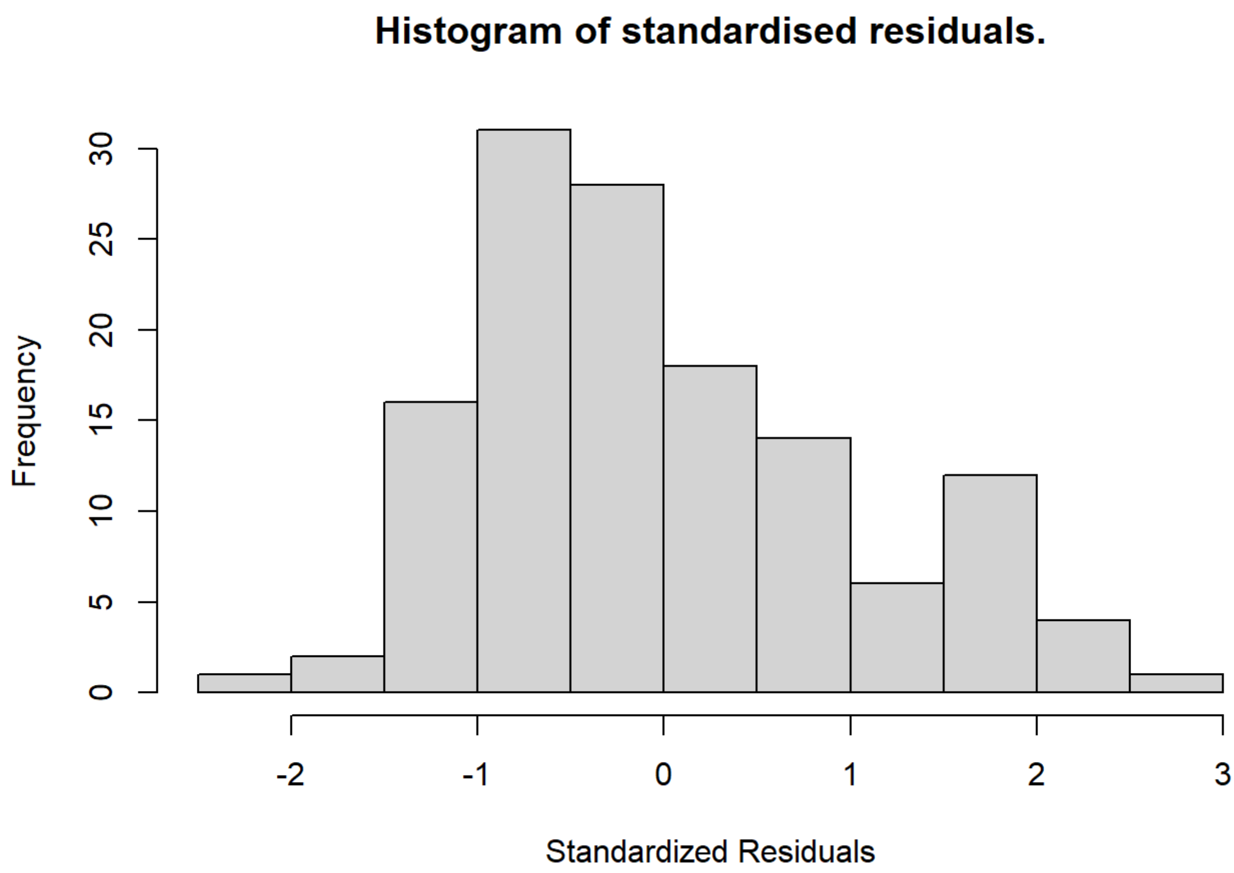
```
plot(ts(fitted(Quadratic_model)), ylab='y', main = "Figure 11: Fitted quadratic curve.",
     ylim = c(min(c(fitted(Quadratic_model), as.vector(Oil_prices.TS))) ,
              max(c(fitted(Quadratic_model), as.vector(Oil_prices.TS)))
     ) )
lines(as.vector(Oil_prices.TS),type="o")
```



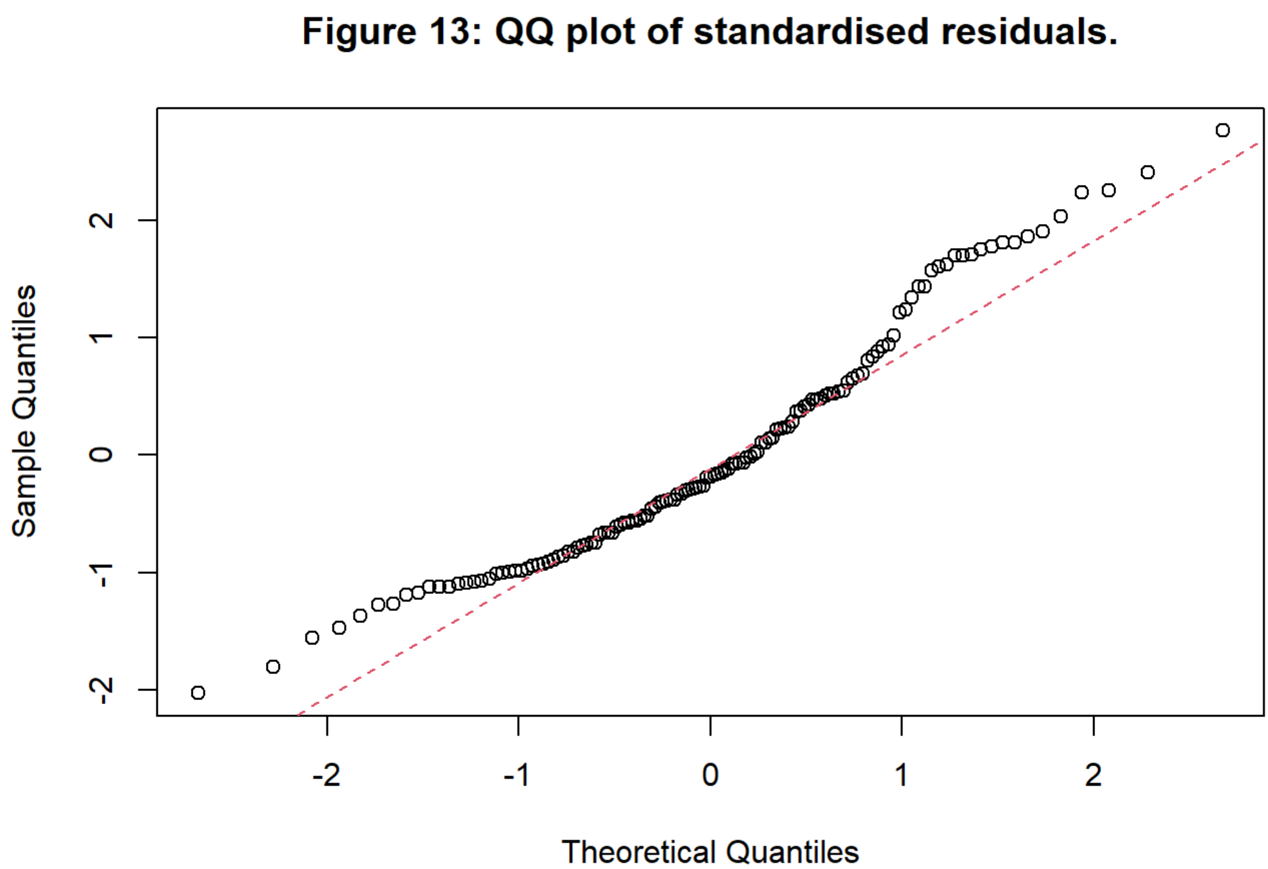
```
res.Quadratic_model = rstudent(Quadratic_model)
plot(y = res.Quadratic_model, x = as.vector(t),xlab = 'Quarter', ylab='Standardized Residual
s',type='l',main = "Figure 12: Standardised residuals from quadratic model.")
```




```
hist(res.Quadratic_model,xlab='Standardized Residuals', main = "Histogram of standardised res
iduals.")
```



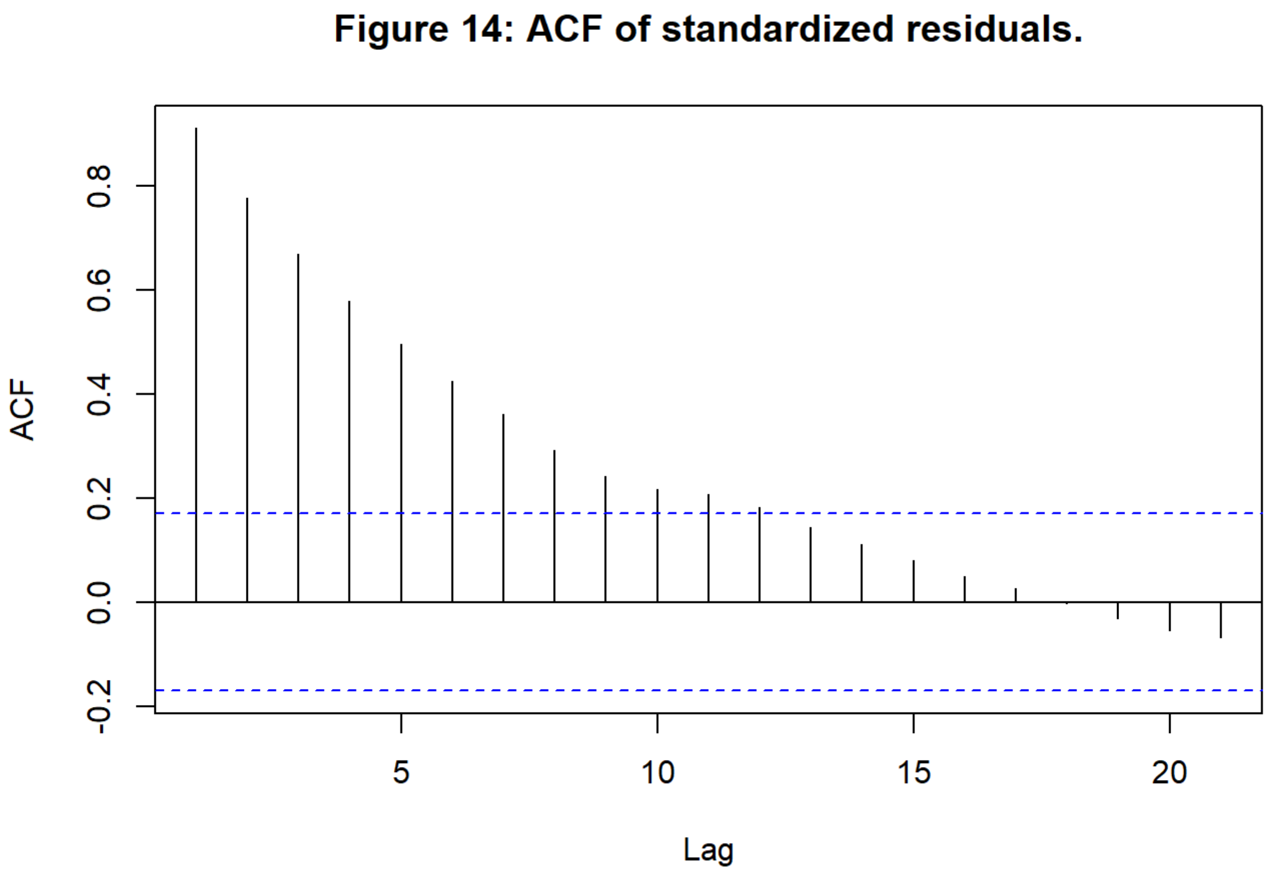

```
qqnorm(y=res.Quadratic_model, main = "Figure 13: QQ plot of standardised residuals.")
qqline(y=res.Quadratic_model, col = 2, lwd = 1, lty = 2)
```



```
shapiro.test(res.Quadratic_model)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.Quadratic_model
## W = 0.95257, p-value = 0.0001492
```

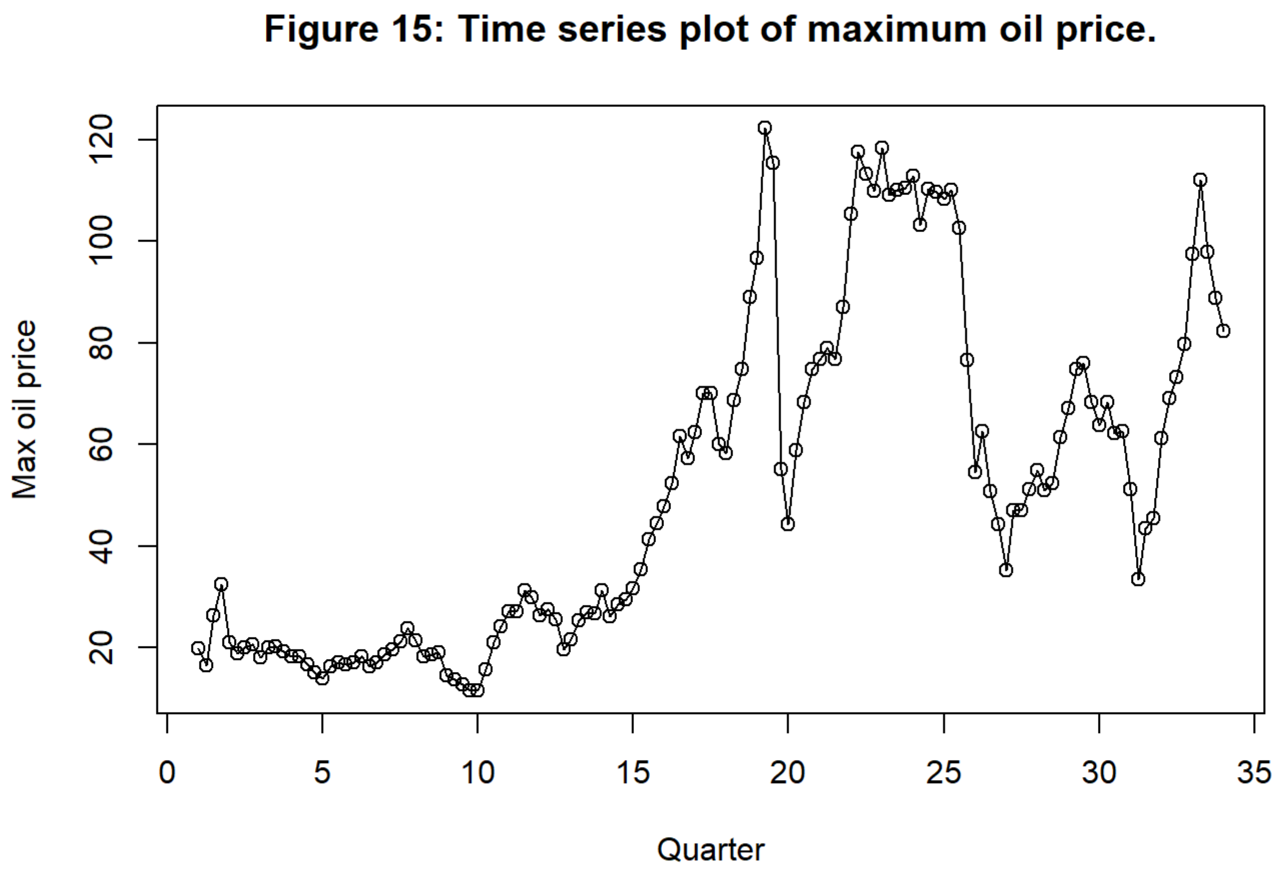
```
acf(res.Quadratic_model, main = "Figure 14: ACF of standardized residuals.")
```



- Time and time squared are the model's significant variables, with a very modest p-value. The model, nevertheless, is significant and has a low p-value. The model's R-squared value is 0.5365, which indicates that it can account for 53.65% of the oil price. Regarding the plot of residuals, the figure (Figure 11) shows no trend over time. In Figure 12, the histogram depicts the residuals' right skewness as they concentrate from -1.5 to 0. Because the residuals in the two tails are not in the straight line, the QQ plots demonstrate that the residuals do not follow a normal distribution (Figure 13). Figure 14 of the ACF demonstrates that the distribution of columns is not normal because several of them are outside the confidence range. The Shapiro-Wilk test also reveals the residuals is not normal distributed with p-value < 0.05.

3.Seasonal:

```
Oil_price_yearly <- ts(Oil_prices$'Brent Crude Oil price', frequency = 4)
plot(Oil_price_yearly, ylab='Max oil price', xlab='Quarter', type='o',
     main = "Figure 15: Time series plot of maximum oil price.")
```

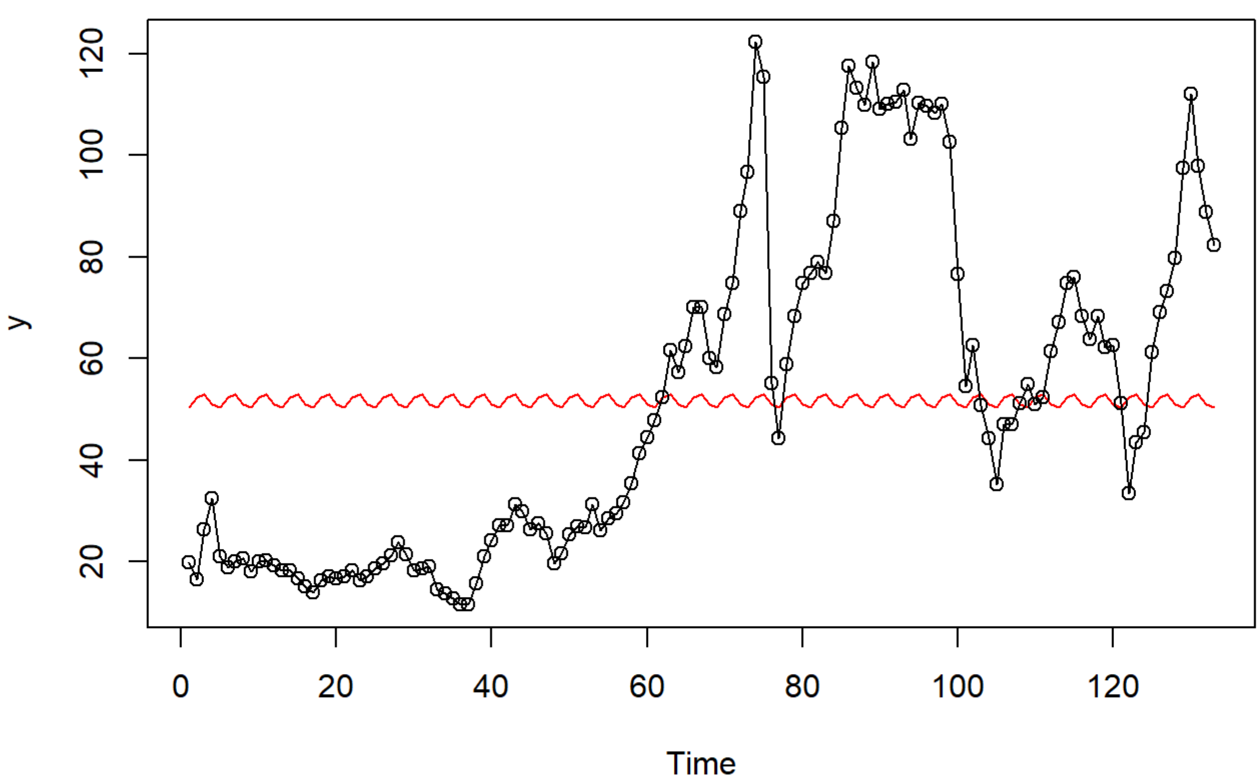



```
year.=season(Oil_price_yearly)
Seasonal_model=lm(Oil_price_yearly ~ year. -1)
summary(Seasonal_model)
```

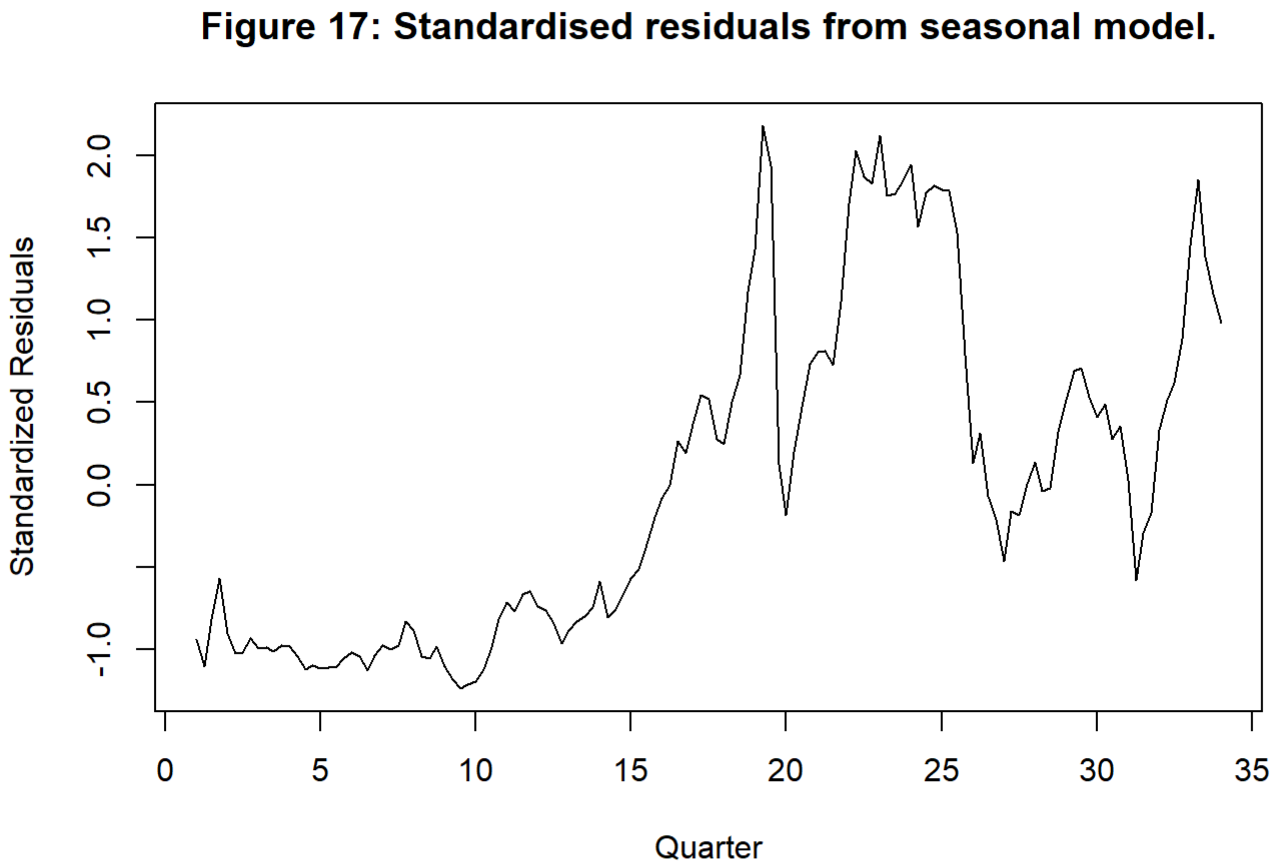
```
##
## Call:
## lm(formula = Oil_price_yearly ~ year. - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.235 -30.571  -5.592  21.737  69.913
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## year.1Q   50.290      5.671   8.869 5.20e-15 ***
## year.2Q   52.306      5.756   9.087 1.53e-15 ***
## year.3Q   53.002      5.756   9.208 7.76e-16 ***
## year.4Q   50.937      5.756   8.850 5.79e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.06 on 129 degrees of freedom
## Multiple R-squared:  0.7154, Adjusted R-squared:  0.7066
## F-statistic: 81.09 on 4 and 129 DF,  p-value: < 2.2e-16
```

```
plot(ts(fitted(Seasonal_model)), ylab='y', main = "Figure 16: Fitted seaonal model to annual
oil price series.",
     ylim = c(min(c(fitted(Seasonal_model), as.vector(Oil_price_yearly))) ,
              max(c(fitted(Seasonal_model), as.vector(Oil_price_yearly)))), col = "red" )
lines(as.vector(Oil_price_yearly),type="o")
```

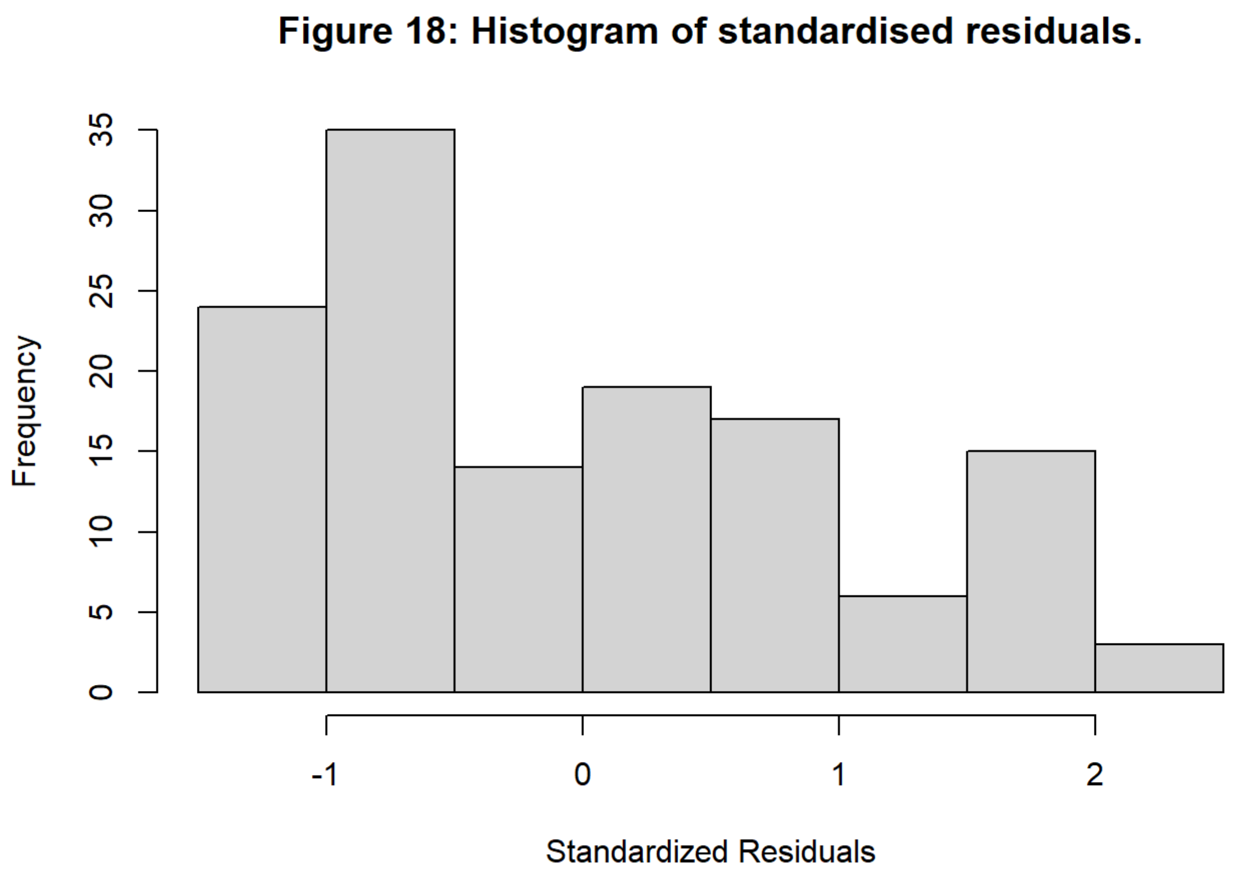
Figure 16: Fitted seaonal model to annual oil price series.



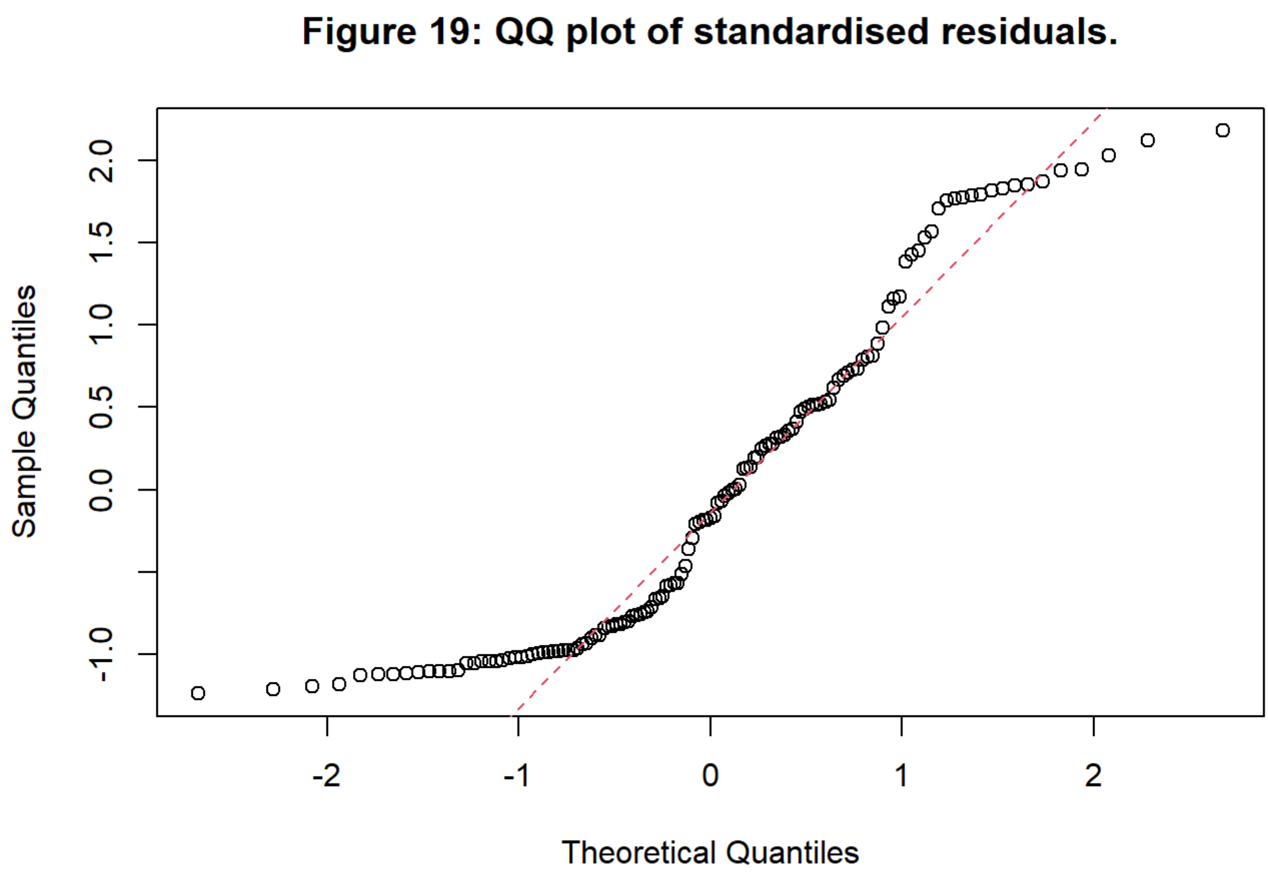
```
res.Seasonal_model = rstudent(Seasonal_model)
plot(y = res.Seasonal_model, x = as.vector(time(Oil_price_yearly)),xlab = 'Quarter', ylab='St
andardized Residuals',type='l',main = "Figure 17: Standardised residuals from seasonal mode
l.")
```




```
hist(res.Seasonal_model,xlab='Standardized Residuals', main = "Figure 18: Histogram of standa
rdised residuals.")
```



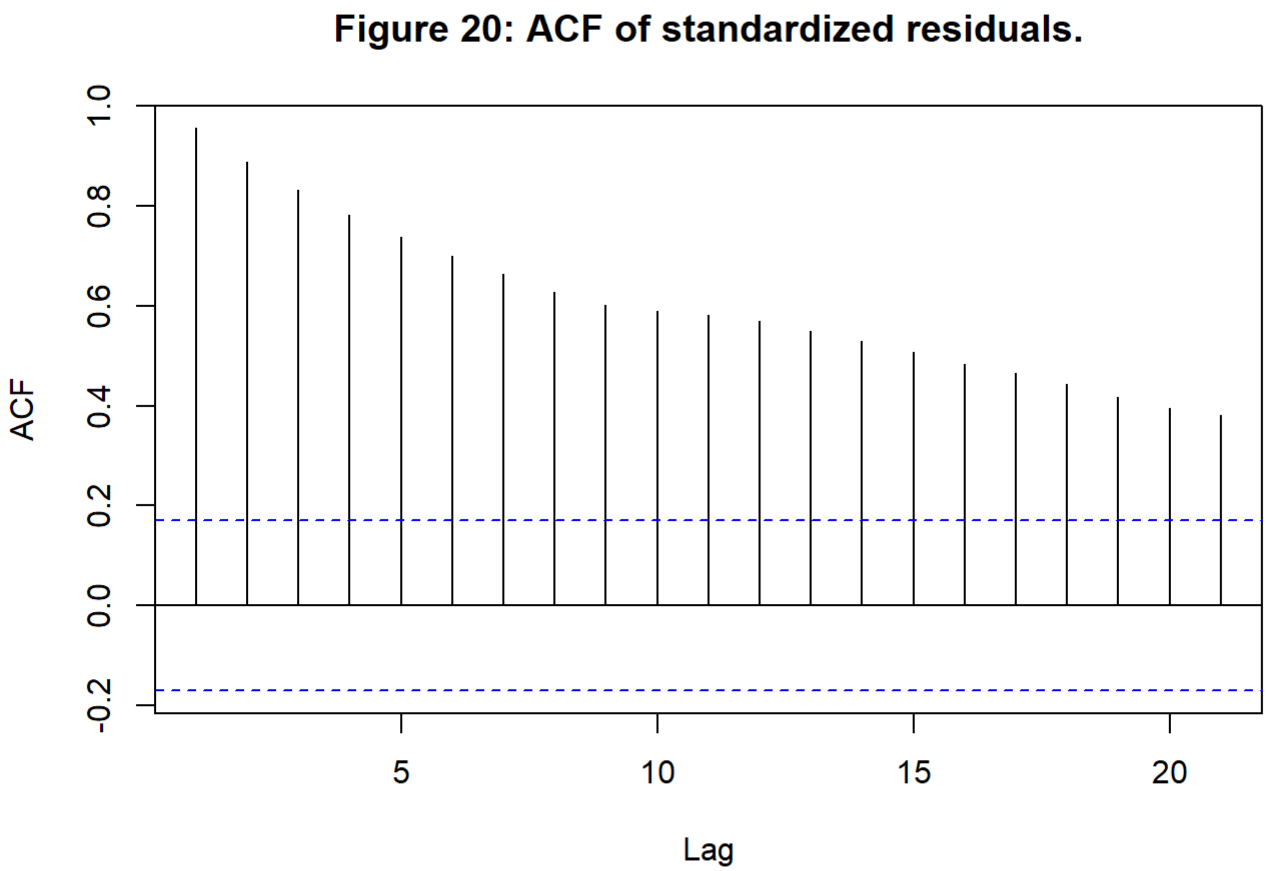

```
qqnorm(y=res.Seasonal_model, main = "Figure 19: QQ plot of standardised residuals.")
qqline(y=res.Seasonal_model, col = 2, lwd = 1, lty = 2)
```



```
shapiro.test(res.Seasonal_model)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.Seasonal_model
## W = 0.8975, p-value = 4.334e-08
```

```
acf(res.Seasonal_model, main = "Figure 20: ACF of standardized residuals.")
```



- All of the model's variables have very low p-values and are all statistically significant. Despite this, the model is significant and has a small p-value. The model's R-squared value is 0.7066, meaning it can explain 70.66% of the price of oil. In terms of the residuals' normality, the figure (Figure 17) exhibits random movement and no obvious trend. The histogram in Figure 18 shows the residuals slightly right skewness. The QQ plots show that the residuals do not follow a normal distribution since the residuals in the two tails are not lied in the straight line (Figure 19).As every column is outside of the confidence interval, Figure 20 of the ACF shows that the distribution of columns is not normal. Additionally, the Shapiro-Wilk test demonstrates that the residuals is not normal distributed with p-value < 0.05

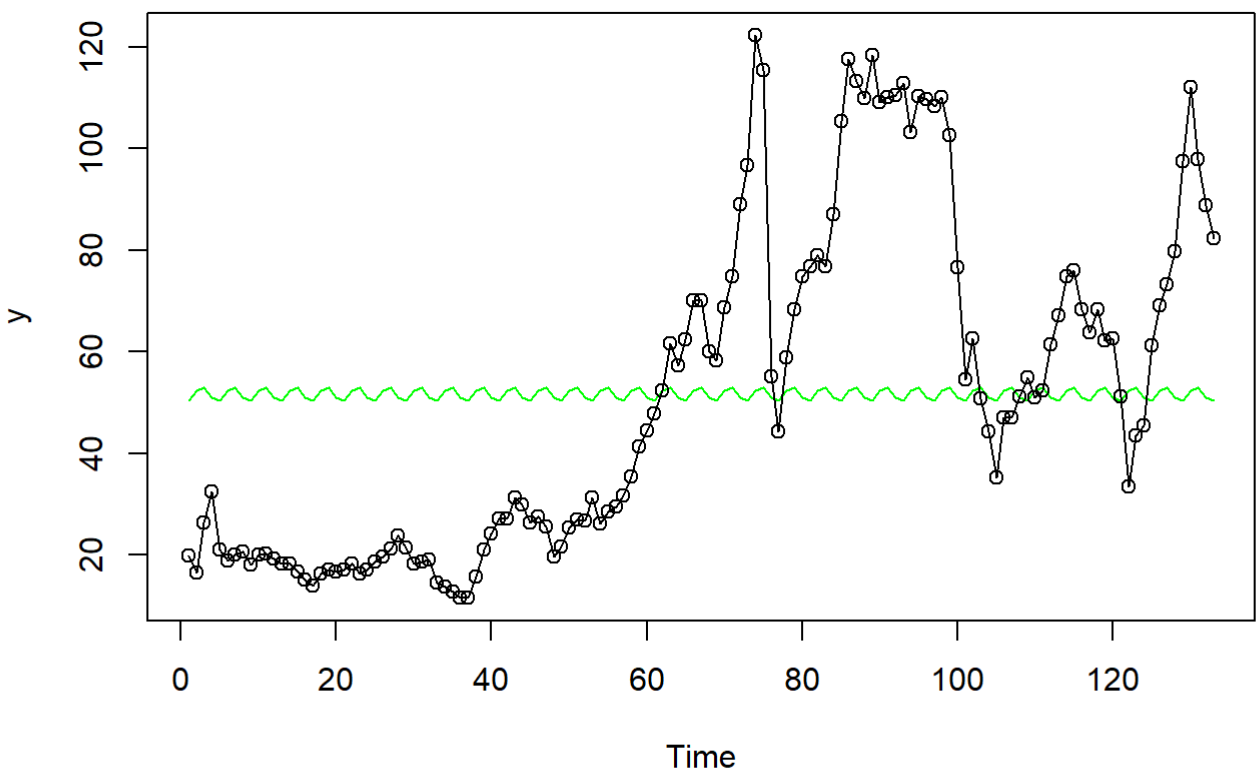
4.Cosine:

```
har. <- harmonic(Oil_price_yearly, 1)
data <- data.frame(Oil_price_yearly,har.)
Cosine_model <- lm(Oil_price_yearly ~ cos.2.pi.t. + sin.2.pi.t. , data = data)
summary(Cosine_model)
```

```
##
## Call:
## lm(formula = Oil_price_yearly ~ cos.2.pi.t. + sin.2.pi.t., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.222 -30.558  -5.605  21.750  69.900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  51.6338     2.8562  18.078   <2e-16 ***
## cos.2.pi.t.  -1.3561     4.0242  -0.337    0.737
## sin.2.pi.t.   0.6845     4.0543   0.169    0.866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.94 on 130 degrees of freedom
## Multiple R-squared:  0.001092,   Adjusted R-squared:  -0.01428
## F-statistic: 0.07104 on 2 and 130 DF,  p-value: 0.9315
```

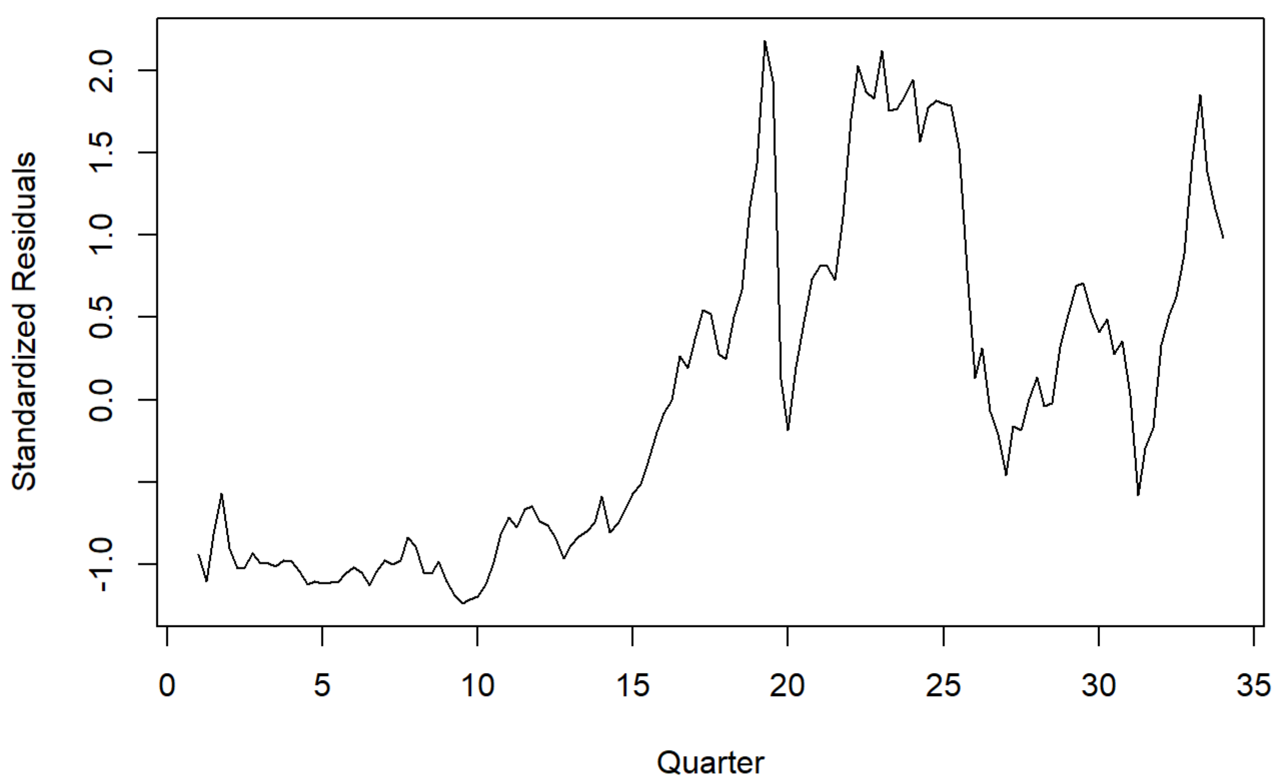
```
plot(ts(fitted(Cosine_model)), ylab='y', main = "Figure 21: Fitted cosine wave annual max oil
price series.",
     ylim = c(min(c(fitted(Cosine_model), as.vector(Oil_price_yearly))) ,
              max(c(fitted(Cosine_model), as.vector(Oil_price_yearly)))
     ), col = "green" )
lines(as.vector(Oil_price_yearly),type="o")
```

Figure 21: Fitted cosine wave annual max oil price series.

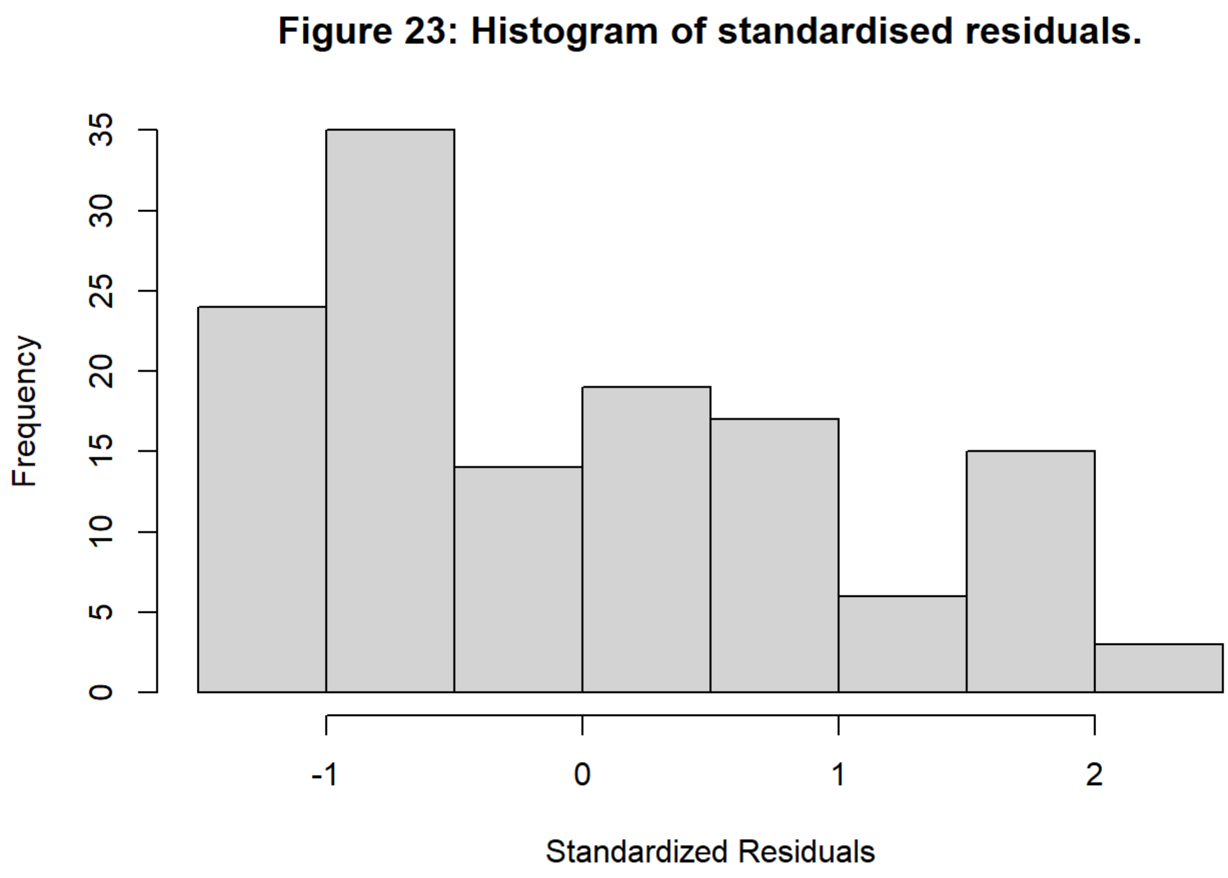


```
res.Cosine_model = rstudent(Cosine_model)
plot(y = res.Cosine_model, x = as.vector(time(Oil_price_yearly)),xlab = 'Quarter', ylab='Standardized Residuals',type='l',main = "Figure 22: Standardised residuals from seasonal model.")
```

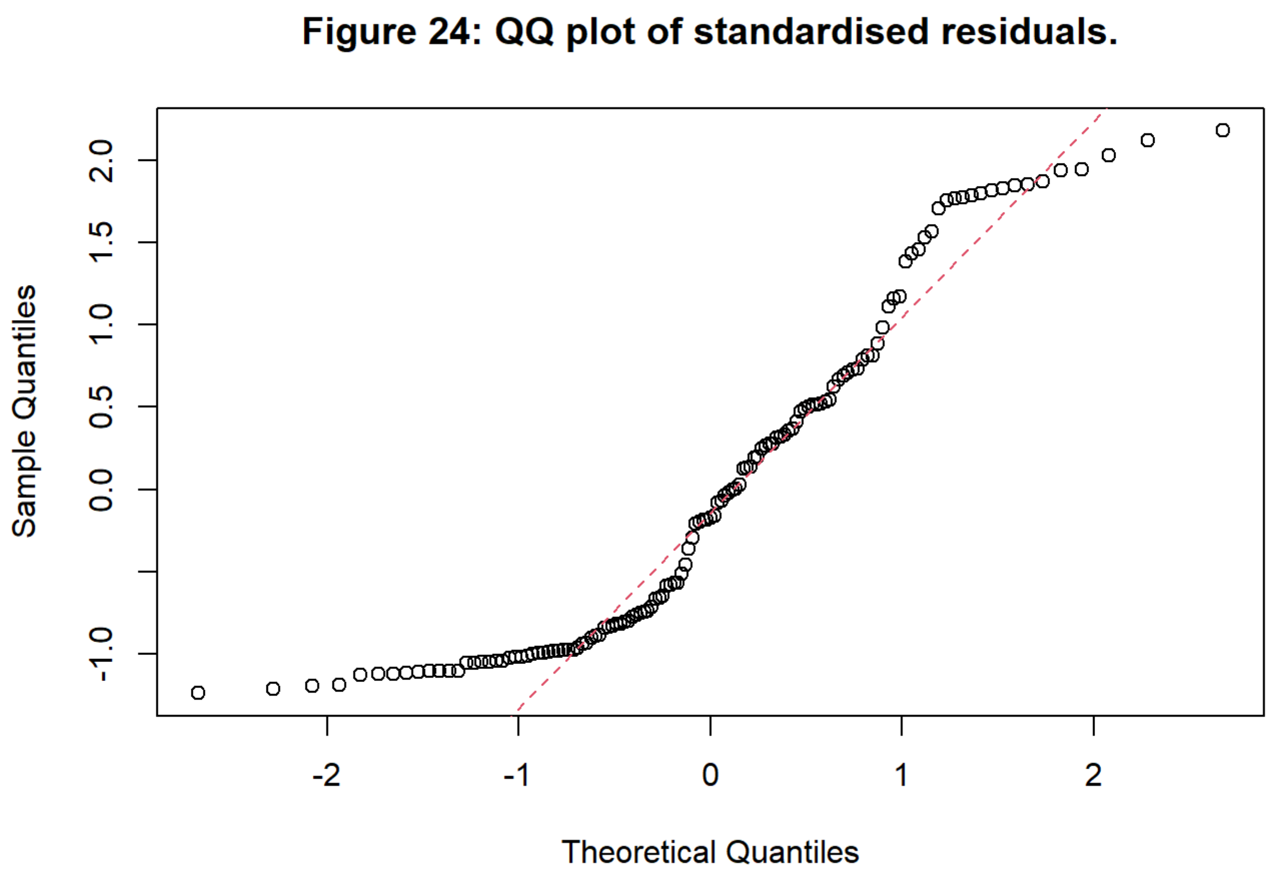
Figure 22: Standardised residuals from seasonal model.




```
hist(res.Cosine_model,xlab='Standardized Residuals', main = "Figure 23: Histogram of standardised residuals.")
```



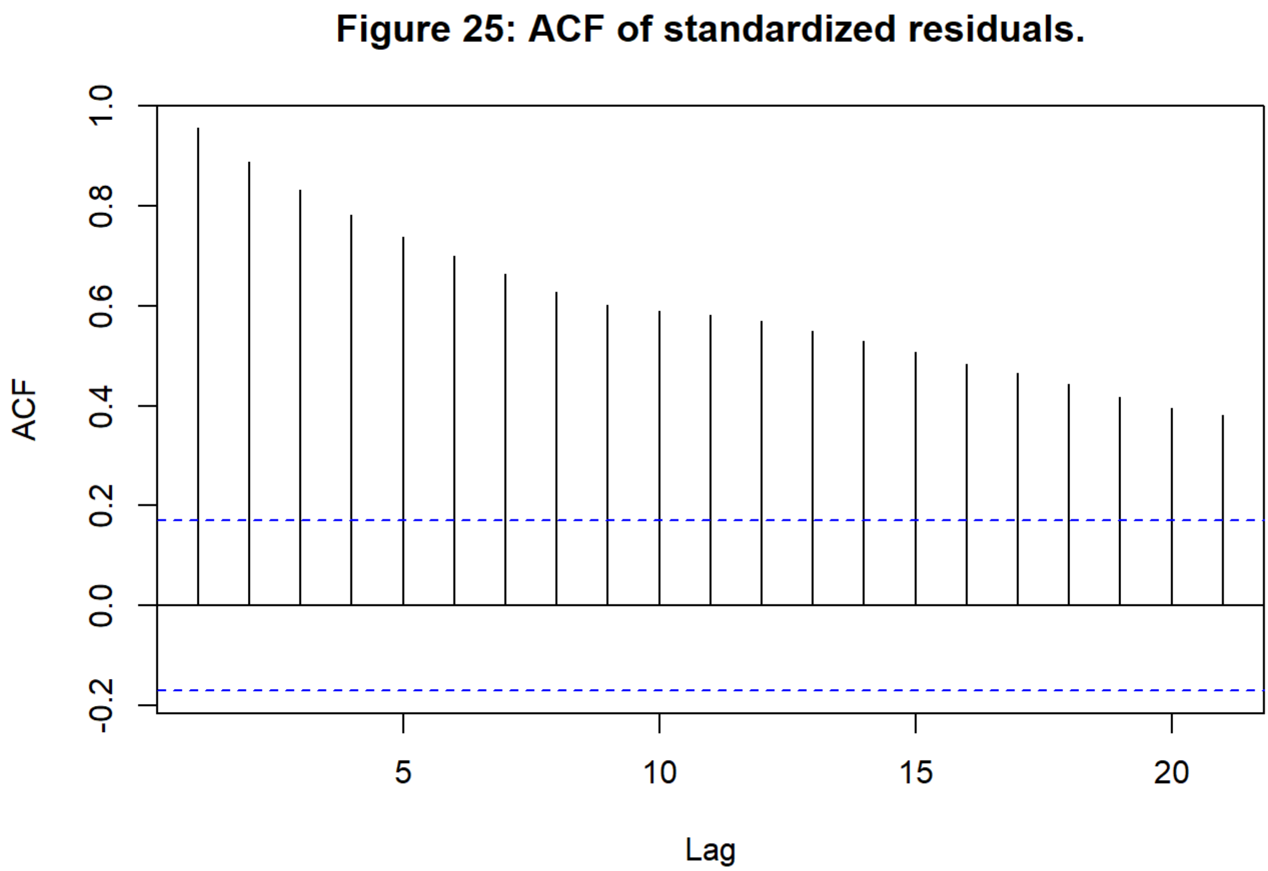

```
qqnorm(y=res.Cosine_model, main = "Figure 24: QQ plot of standardised residuals.")
qqline(y=res.Cosine_model, col = 2, lwd = 1, lty = 2)
```



```
shapiro.test(res.Cosine_model)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.Cosine_model
## W = 0.89748, p-value = 4.326e-08
```

```
acf(res.Cosine_model, main = "Figure 25: ACF of standardized residuals.")
```



- All of the model's variables are statistically insignificant as having p-values > 0.05. Despite this, the model has a high p-value and is insignificant. The model's R-squared score of -0.014 shows that using these variables to explain oil price worst. The residuals is seen in Figure 22 as growing in trend. The histogram show slightly right skewness (Figure 23). Since the residuals in the two tails are not aligned in a straight line, the QQ plots demonstrate that the residuals do not follow a normal distribution (Figure 24).The ACF's Figure 25 illustrates that the distribution of columns is not normal because every column is outside of the confidence range. The Shapiro-Wilk test additionally reveals that the residuals is not normal distributed.

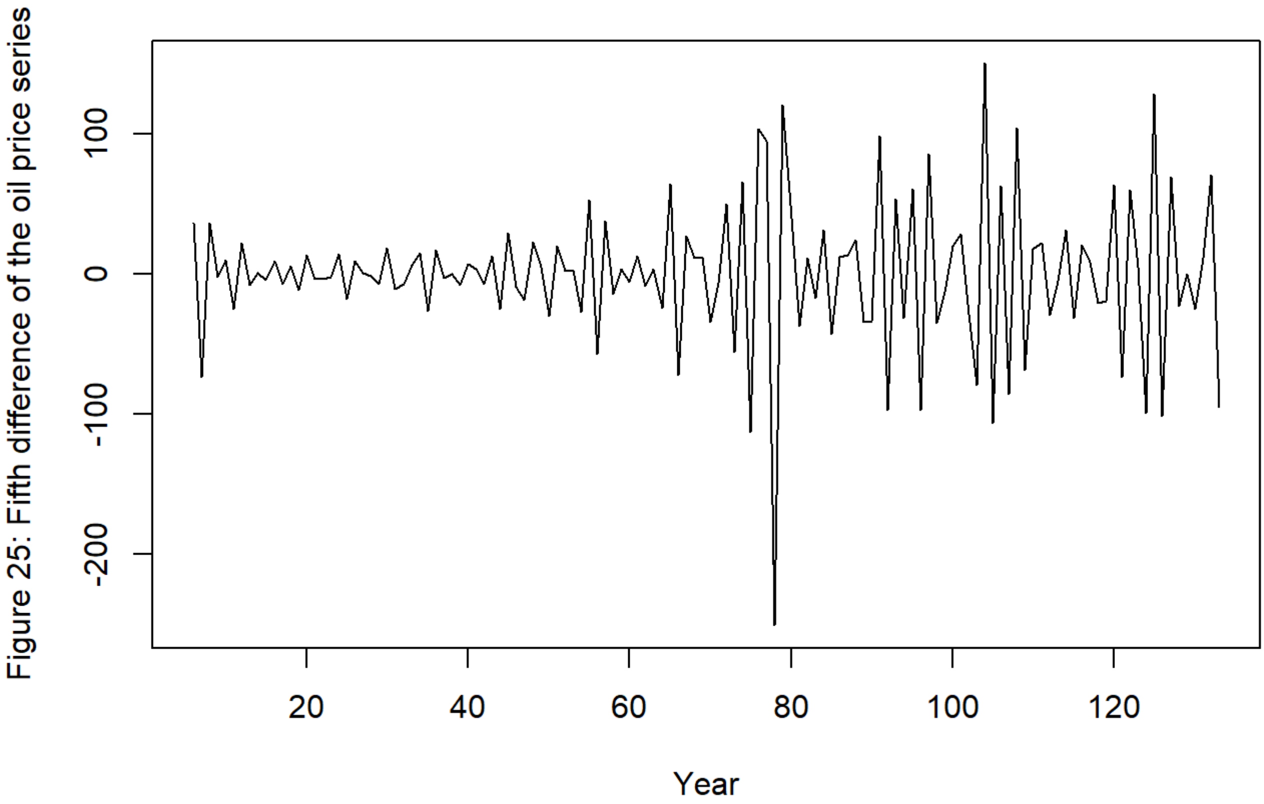
III.ARIMA

```
BoxCoxSearch = function(y, lambda=seq(-3,3,0.01),
                        m= c("sf", "sw","ad" ,"cvm", "pt", "lt", "jb"), plotit = T, verbose =
T){
  N = length(m)
  BC.y = array(NA,N)
  BC.lam = array(NA,N)
  for (i in 1:N){
    if (m[i] == "sf"){
      wrt = "Shapiro-Francia Test"
    } else if (m[i] == "sw"){
      wrt = "Shapiro-Wilk  Test"
    } else if (m[i] == "ad"){
      wrt = "Anderson-Darling Test"
    } else if (m[i] == "cvm"){
      wrt = "Cramer-von Mises Test"
    } else if (m[i] == "pt"){
      wrt = "Pearson Chi-square Test"
    } else if (m[i] == "lt"){
      wrt = "Lilliefors Test"
    } else if (m[i] == "jb"){
      wrt = "Jarque-Bera Test"
    }

    print(paste0("------------- ",wrt," -------------"))
    out = tryCatch({boxcoxnc(y, method = m[i], lam = lambda, lambda2 = NULL, plot = plotit, a
lpha = 0.05, verbose = verbose)
                   BC.lam[i] = as.numeric(out$lambda.hat)},
                   error = function(e) print("No results for this test!"))

  }
  return(list(lambda = BC.lam,p.value = BC.y))
}

find_lambda <- BoxCoxSearch(y = Oil_prices.TS, lambda = seq(-3, 3, 0.01), m = c("sf", "sw",
"ad", "cvm", "pt", "lt", "jb"), plotit = TRUE, verbose = TRUE)
```

```
## [1] "------------- Shapiro-Francia Test -------------"
## [1] "No results for this test!"
## [1] "------------- Shapiro-Wilk  Test -------------"
## [1] "No results for this test!"
## [1] "------------- Anderson-Darling Test -------------"
## [1] "No results for this test!"
## [1] "------------- Cramer-von Mises Test -------------"
## [1] "No results for this test!"
## [1] "------------- Pearson Chi-square Test -------------"
## [1] "No results for this test!"
## [1] "------------- Lilliefors Test -------------"
## [1] "No results for this test!"
## [1] "------------- Jarque-Bera Test -------------"
## [1] "No results for this test!"
```

- After using this function to find the suitable lambda for the time series. The result shows that there is no suitable lamba for the dataset. Then the Box-Cox tranformation will not be used.

```
Oil_prices.TS.Diff <- diff(Oil_prices.TS, differences = 5)
plot(Oil_prices.TS.Diff, xlab = "Year", ylab = "Figure 25: Fifth difference of the oil price series")
```




```
adf.test(Oil_prices.TS.Diff, alternative = c("stationary"))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Oil_prices.TS.Diff
## Dickey-Fuller = -12.583, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(Oil_prices.TS.Diff)
```

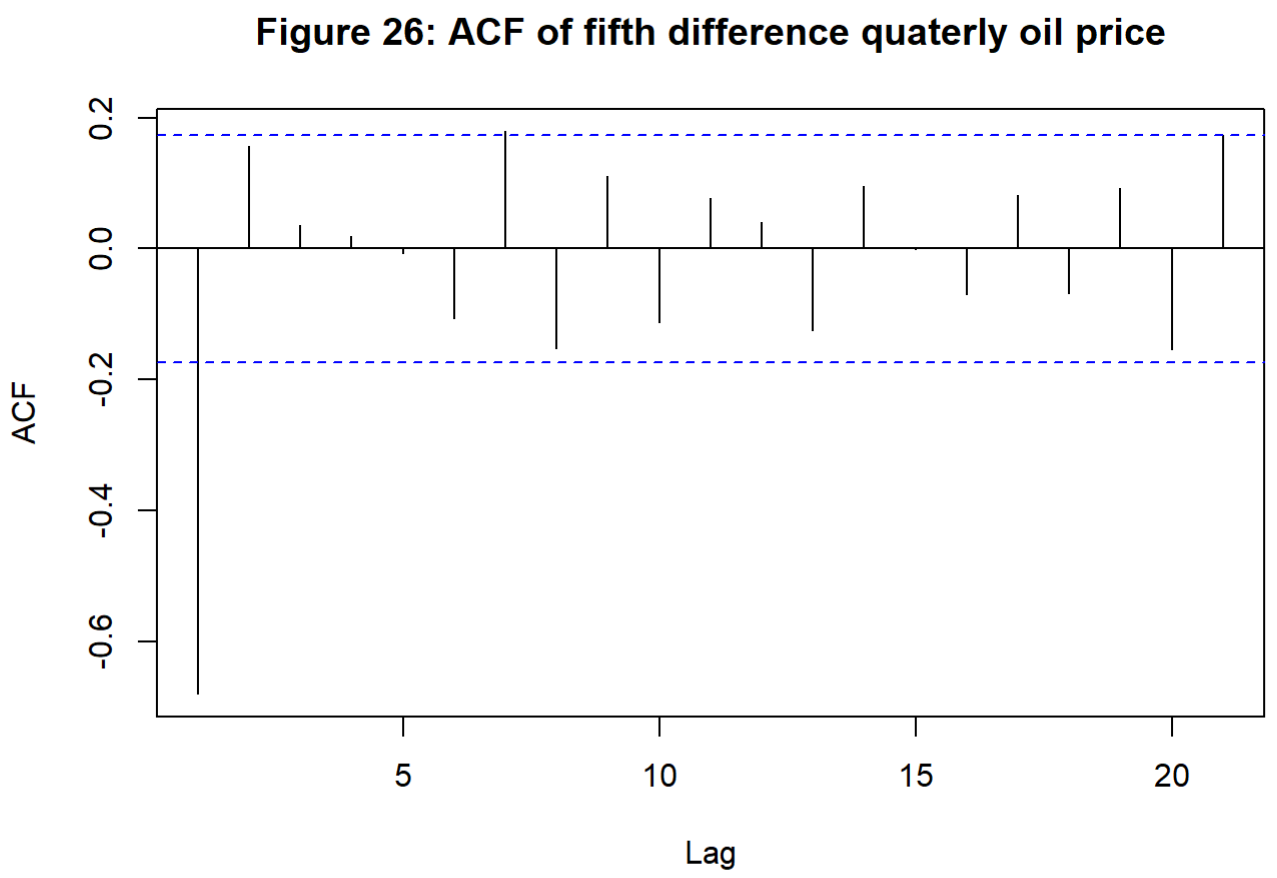
```
##
##  Phillips-Perron Unit Root Test
##
## data:  Oil_prices.TS.Diff
## Dickey-Fuller Z(alpha) = -188.09, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(Oil_prices.TS.Diff)
```

```
## 
##  KPSS Test for Level Stationarity
## 
## data:  Oil_prices.TS.Diff
## KPSS Level = 0.026689, Truncation lag parameter = 4, p-value = 0.1
```

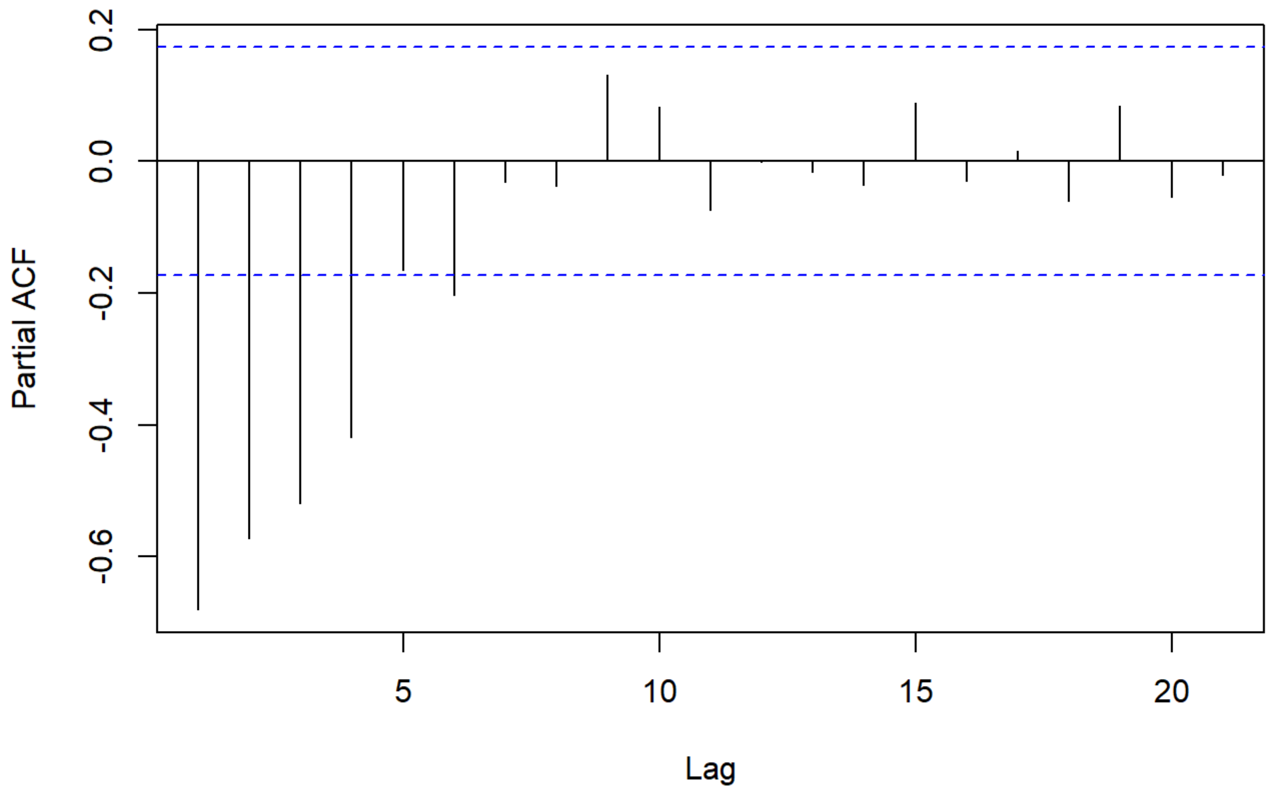
- The p-value of ADF test 0.01 which is smaller than 0.05. The p-value of pp test is 0.01 which is smaller than 0.05. The kpss test has the p-value of 0.1 which is greater than 0.05. The result of these test indicate that the data is stationary.

```
acf(Oil_prices.TS.Diff, main = "Figure 26: ACF of fifth difference quaterly oil price")
```




```
pacf(Oil_prices.TS.Diff, main = "Figure 27: PACF of fifth difference quaterly oil price")
```

Figure 27: PACF of fifth difference quaterly oil price

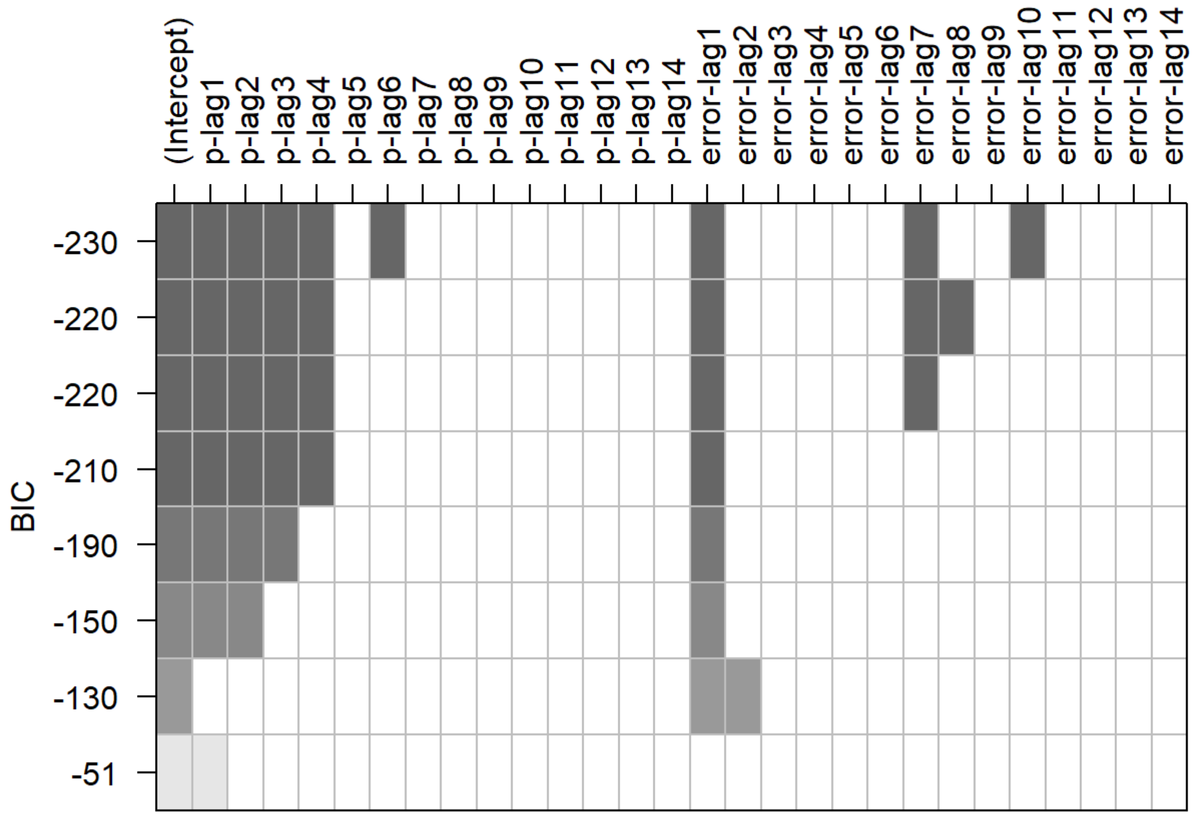


- From the ACF graph (Figure 26), the possible set for p could be (3,4,5,6), when the possible set for q could be (5,6) from PACF graph (Figure 27). Then the possible set of ARIMA is: ARIMA(3, 5, 5), ARIMA(3, 5, 6), ARIMA(4, 5, 5), ARIMA(4, 5, 6), ARIMA(5, 5, 5), ARIMA(5, 5, 6), ARIMA(6, 5, 5), ARIMA(6, 5, 6), ARIMA(3, 5, 0), ARIMA(3, 5, 1), ARIMA(4, 5, 0), ARIMA(4, 5, 1), ARIMA(1, 5, 1), ARIMA(2, 5, 1).

```
eacf(Oil_prices.TS.Diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o o o o o  o  o  o
## 1 x x o o o o o o o o o  o  x  o
## 2 x x x o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x x o o x x o o o o o  o  o  o
## 5 x x x o o x x o o o o  o  o  o
## 6 x x x o o x x x o o o  o  o  o
## 7 x x x o o x x o o x o  o  o  o
```

```
res = armasubsets(y= Oil_prices.TS.Diff , nar=14 , nma=14, y.name='p', ar.method='ols')
plot(res)
```



- From eacf the top left method recommend the possible set of p is (2,3) with the possible set of q is (0,1). The the ARIMA set is: ARIMA(2, 5, 0), ARIMA(2, 5, 1), ARIMA(3, 5, 0), ARIMA(3, 5, 1). From the BIC table, the possible set for p is (1,2) and the possible set for q is (1) then ARIMA(1,5,1), ARIMA(2,5,1) are added to the list. Overall, with the repetition of ARIMA(2,5,1) ,there is the list of ARIMA: ARIMA(3, 5, 5), ARIMA(3, 5, 6), ARIMA(4, 5, 5), ARIMA(4, 5, 6), ARIMA(5, 5, 5), ARIMA(5, 5, 6), ARIMA(6, 5, 5), ARIMA(6, 5, 6), ARIMA(3, 5, 0), ARIMA(3, 5, 1), ARIMA(4, 5, 0), ARIMA(4, 5, 1), ARIMA(1, 5, 1), ARIMA(2, 5, 1), ARIMA(3, 5, 0), ARIMA(3, 5, 1), ARIMA(1,5,1), ARIMA(2,5,1)

IV.Fitted ARIMA model:

```
fit_arima_models <- function(data, order_list, methods = c("ML", "CSS", "CSS-ML")) {
  model_results <- list()

  for (i in seq_along(order_list)) {
    order <- order_list[[i]]
    model_number <- paste0("model.", paste(order, collapse = ""))

    for (method in methods) {
      model <- arima(data, order = order, method = method)
      coefs <- coeftest(model)
      model_results[[paste0(model_number, ".", method)]] <- list(model = model, coefs = coef
s, method = method)
    }
  }

  return(model_results)
}

order_list <- list(
  c(3, 5, 5),
  c(3, 5, 6),
  c(4, 5, 5),
  c(4, 5, 6),
  c(5, 5, 5),
  c(5, 5, 6),
  c(6, 5, 5),
  c(6, 5, 6),
  c(2, 5, 0),
  c(2, 5, 1),
  c(3, 5, 0),
  c(3, 5, 1),
  c(1, 5, 1)
)

model_results <- fit_arima_models(Oil_prices.TS, order_list)
```

1.Coeficient

```
coefs.355 <- model_results[["model.355.ML"]]$coefs
coefs.355_css <- model_results[["model.355.CSS"]]$coefs
coefs.355_css_ml <- model_results[["model.355.CSS-ML"]]$coefs

coefs.355
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -1.223536   0.032762 -37.3459  < 2e-16 ***
## ar2 -0.797485   0.039893 -19.9907  < 2e-16 ***
## ar3 -0.312959        NaN      NaN      NaN
## ma1 -1.981223   0.025111 -78.8989  < 2e-16 ***
## ma2  0.330716   0.029525  11.2010  < 2e-16 ***
## ma3  1.009185   0.035066  28.7792  < 2e-16 ***
## ma4 -0.039961   0.020255  -1.9729  0.04851 *
## ma5 -0.318613   0.020367 -15.6432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.355_css
```

```
##
## z test of coefficients:
##
##        Estimate  Std. Error   z value  Pr(>|z|)
## ar1 -0.71592897         NaN       NaN       NaN
## ar2 -0.48298644         NaN       NaN       NaN
## ar3 -0.36696256  0.00038575 -951.2961 < 2.2e-16 ***
## ma1 -2.46737389  0.04845695  -50.9189 < 2.2e-16 ***
## ma2  1.43210513  0.21694255    6.6013 4.075e-11 ***
## ma3  0.68684263  0.37577473    1.8278   0.06758 .
## ma4 -0.77749891  0.30251335   -2.5701   0.01017 *
## ma5  0.12584147  0.09560292    1.3163   0.18808
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.355_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.715932   0.045338 -15.7909 < 2.2e-16 ***
## ar2 -0.482987   0.070611  -6.8401 7.911e-12 ***
## ar3 -0.366960        NaN      NaN       NaN
## ma1 -2.460202   0.063469 -38.7621 < 2.2e-16 ***
## ma2  1.421907   0.067489  21.0687 < 2.2e-16 ***
## ma3  0.686706   0.045587  15.0636 < 2.2e-16 ***
## ma4 -0.772798   0.037489 -20.6139 < 2.2e-16 ***
## ma5  0.125065   0.028614   4.3707 1.238e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The result of the ML and CSS is slightly different, then CSS-ML is used and the result shows that every variables is significant except ar3.

```
coefs.356 <- model_results[["model.356.ML"]]$coefs
coefs.356_css <- model_results[["model.356.CSS"]]$coefs
coefs.356_css_ml <- model_results[["model.356.CSS-ML"]]$coefs
coefs.356
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error   z value  Pr(>|z|)
## ar1 -0.9935307  0.0105493  -94.1795 < 2.2e-16 ***
## ar2 -0.5026276        NaN       NaN       NaN
## ar3 -0.1607708  0.0261443   -6.1494 7.779e-10 ***
## ma1 -2.3225600  0.0041481 -559.9125 < 2.2e-16 ***
## ma2  0.8914186        NaN       NaN       NaN
## ma3  1.1440224        NaN       NaN       NaN
## ma4 -0.4268227  0.0305916  -13.9523 < 2.2e-16 ***
## ma5 -0.5067893  0.0074867  -67.6921 < 2.2e-16 ***
## ma6  0.2209406  0.0100617   21.9586 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.356_css
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -1.052380        NaN     NaN      NaN
## ar2 -0.754445        NaN     NaN      NaN
## ar3 -0.394264        NaN     NaN      NaN
## ma1 -1.942223        NaN     NaN      NaN
## ma2  0.424921        NaN     NaN      NaN
## ma3  0.831714        NaN     NaN      NaN
## ma4 -0.136135        NaN     NaN      NaN
## ma5 -0.098438        NaN     NaN      NaN
## ma6 -0.082953        NaN     NaN      NaN
```

```
coefs.356_css_ml
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.9892628  0.0324525 -30.4834 < 2.2e-16 ***
## ar2 -0.4946141  0.0568660  -8.6979 < 2.2e-16 ***
## ar3 -0.1374649  0.0304944  -4.5079 6.548e-06 ***
## ma1 -2.3264799  0.0308511 -75.4099 < 2.2e-16 ***
## ma2  0.8633846  0.0444284  19.4332 < 2.2e-16 ***
## ma3  1.1988723  0.0212159  56.5082 < 2.2e-16 ***
## ma4 -0.4045798  0.0203125 -19.9177 < 2.2e-16 ***
## ma5 -0.5855573        NaN      NaN       NaN
## ma6  0.2544248  0.0079942  31.8263 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The result of the ML and CSS is different, then CSS-ML is used and the result shows that every variables is significant except ma5.

```
coefs.455 <- model_results[["model.455.ML"]]$coefs
coefs.455_css <- model_results[["model.455.CSS"]]$coefs
coefs.455_css_ml <- model_results[["model.455.CSS-ML"]]$coefs
coefs.455
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.441482   0.462183  -3.1189  0.001816 **
## ar2 -1.192577   0.726721  -1.6410  0.100789
## ar3 -0.696322   0.537193  -1.2962  0.194899
## ar4 -0.289875   0.240314  -1.2062  0.227726
## ma1 -1.878787   0.034917 -53.8069 < 2.2e-16 ***
## ma2  0.124992        NaN      NaN       NaN
## ma3  0.985526   0.073778  13.3581 < 2.2e-16 ***
## ma4  0.202249   0.014520  13.9289 < 2.2e-16 ***
## ma5 -0.433936   0.049044  -8.8478 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.455_css
```

```
##
## z test of coefficients:
##
##        Estimate  Std. Error    z value  Pr(>|z|)
## ar1 -1.26905564  0.00101015 -1256.3047 < 2.2e-16 ***
## ar2 -0.90452100  0.00021828 -4143.9331 < 2.2e-16 ***
## ar3 -0.61078307  0.00020037 -3048.3370 < 2.2e-16 ***
## ar4 -0.26428306  0.00065555  -403.1491 < 2.2e-16 ***
## ma1 -2.02539147  0.02667946   -75.9158 < 2.2e-16 ***
## ma2  0.45000076  0.07476046     6.0192 1.752e-09 ***
## ma3  0.89157167  0.07208836    12.3678 < 2.2e-16 ***
## ma4  0.00636868         NaN        NaN       NaN
## ma5 -0.31810875         NaN        NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.455_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -1.311710        NaN      NaN      NaN
## ar2 -1.071025        NaN      NaN      NaN
## ar3 -0.669069        NaN      NaN      NaN
## ar4 -0.280089        NaN      NaN      NaN
## ma1 -1.987339   0.035428 -56.0945   <2e-16 ***
## ma2  0.367908   0.035409  10.3903   <2e-16 ***
## ma3  0.920733        NaN      NaN      NaN
## ma4  0.044428   0.043863   1.0129   0.3111
## ma5 -0.345686   0.019187 -18.0168   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The result of the ML and CSS is different, then CSS-ML is used and the result shows that only ma1, ma2 and ma5 is significant.

```
coefs.456 <- model_results[["model.456.ML"]]$coefs
coefs.456_css <- model_results[["model.456.CSS"]]$coefs
coefs.456_css_ml <- model_results[["model.456.CSS-ML"]]$coefs
coefs.456
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.271382        NaN     NaN       NaN
## ar2 -0.723293        NaN     NaN       NaN
## ar3 -0.393844        NaN     NaN       NaN
## ar4 -0.295236        NaN     NaN       NaN
## ma1 -2.087619        NaN     NaN       NaN
## ma2  0.236326        NaN     NaN       NaN
## ma3  1.617286        NaN     NaN       NaN
## ma4 -0.267563   0.020254 -13.210 < 2.2e-16 ***
## ma5 -0.800704        NaN     NaN       NaN
## ma6  0.302413   0.010256  29.486 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.456_css
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error   z value  Pr(>|z|)
## ar1 -1.2743757  0.0193620  -65.8184 < 2.2e-16 ***
## ar2 -0.9702423  0.0021420 -452.9614 < 2.2e-16 ***
## ar3 -0.6587521  0.0016261 -405.1017 < 2.2e-16 ***
## ar4 -0.3191363        NaN       NaN       NaN
## ma1 -1.9947074        NaN       NaN       NaN
## ma2  0.4060128  0.0040490  100.2761 < 2.2e-16 ***
## ma3  0.9323322  0.1259303    7.4036 1.326e-13 ***
## ma4 -0.1356338  0.1857380   -0.7302   0.46524
## ma5 -0.0641310  0.1509637   -0.4248   0.67097
## ma6 -0.1450792  0.0597762   -2.4270   0.01522 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.456_css_ml
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error   z value  Pr(>|z|)
## ar1 -1.1868682  0.0294976  -40.2360 < 2.2e-16 ***
## ar2 -0.4492156  0.0556541   -8.0716 6.940e-16 ***
## ar3 -0.1589007  0.0655361   -2.4246   0.01532 *
## ar4 -0.2333688  0.0416671   -5.6008 2.134e-08 ***
## ma1 -2.2661532        NaN       NaN       NaN
## ma2  0.3234630        NaN       NaN       NaN
## ma3  2.0908316        NaN       NaN       NaN
## ma4 -0.4841782  0.0043921 -110.2376 < 2.2e-16 ***
## ma5 -1.2598268        NaN       NaN       NaN
## ma6  0.5958969        NaN       NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The result of the ML and CSS is different, then CSS-ML is used and the result shows that most ar variables is significant and most ma is insignificant.

```
coefs.555 <- model_results[["model.555.ML"]]$coefs
coefs.555_css <- model_results[["model.555.CSS"]]$coefs
coefs.555_css_ml <- model_results[["model.555.CSS-ML"]]$coefs
coefs.555
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.6533284        NaN      NaN       NaN
## ar2 -1.5398732        NaN      NaN       NaN
## ar3 -1.0790264  0.0107749 -100.142 < 2.2e-16 ***
## ar4 -0.7740895  0.0074007 -104.597 < 2.2e-16 ***
## ar5 -0.5687665  0.0020563 -276.592 < 2.2e-16 ***
## ma1 -1.9030957  0.0045329 -419.837 < 2.2e-16 ***
## ma2  0.2100810  0.0051761   40.587 < 2.2e-16 ***
## ma3  0.9045411  0.0031815  284.313 < 2.2e-16 ***
## ma4  0.1924772  0.0059543   32.325 < 2.2e-16 ***
## ma5 -0.4038716  0.0046005  -87.788 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.555_css
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.346656   0.100564 -13.3911 < 2.2e-16 ***
## ar2 -1.094428   0.033793 -32.3859 < 2.2e-16 ***
## ar3 -0.573034   0.116564  -4.9161  8.83e-07 ***
## ar4 -0.112818   0.088388  -1.2764    0.2018
## ar5  0.041993        NaN      NaN       NaN
## ma1 -1.961226        NaN      NaN       NaN
## ma2  0.348896        NaN      NaN       NaN
## ma3  0.839563        NaN      NaN       NaN
## ma4  0.165083        NaN      NaN       NaN
## ma5 -0.391728   0.042285  -9.2640 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.555_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.359774   0.103837 -13.0952 < 2.2e-16 ***
## ar2 -1.183527   0.184946  -6.3993 1.561e-10 ***
## ar3 -0.772894   0.100597  -7.6831 1.553e-14 ***
## ar4 -0.387591        NaN      NaN       NaN
## ar5 -0.076223   0.046384  -1.6433    0.1003
## ma1 -1.962219   0.040864 -48.0186 < 2.2e-16 ***
## ma2  0.347383        NaN      NaN       NaN
## ma3  0.825320   0.076120  10.8424 < 2.2e-16 ***
## ma4  0.195030   0.039037   4.9961 5.851e-07 ***
## ma5 -0.405510   0.011968 -33.8831 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The result of the ML show that all variable is significant when most of variable in CSS is NaN, then CSS-ML is used and the result shows that most variables is significant.

```
coefs.556 <- model_results[["model.556.ML"]]$coefs
coefs.556_css <- model_results[["model.556.CSS"]]$coefs
coefs.556_css_ml <- model_results[["model.556.CSS-ML"]]$coefs
coefs.556
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.3098172  0.0491650 -26.6413 < 2.2e-16 ***
## ar2 -1.0184239  0.0735131 -13.8536 < 2.2e-16 ***
## ar3 -0.7192924  0.0583959 -12.3175 < 2.2e-16 ***
## ar4 -0.3742537  0.0682119  -5.4866 4.097e-08 ***
## ar5 -0.0759118  0.0202553  -3.7477 0.0001784 ***
## ma1 -2.0311322  0.0223796 -90.7583 < 2.2e-16 ***
## ma2  0.3767728        NaN      NaN       NaN
## ma3  1.1242253        NaN      NaN       NaN
## ma4 -0.2141753        NaN      NaN       NaN
## ma5 -0.2658488        NaN      NaN       NaN
## ma6  0.0101889  0.0081961   1.2431 0.2138164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.556_css
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.3077369  0.0925505 -14.1300 < 2.2e-16 ***
## ar2 -1.1088180  0.1750915  -6.3328 2.408e-10 ***
## ar3 -0.6311432        NaN      NaN       NaN
## ar4 -0.1552067        NaN      NaN       NaN
## ar5 -0.0084312  0.1741572  -0.0484    0.9614
## ma1 -1.9379384  0.0286805 -67.5699 < 2.2e-16 ***
## ma2  0.4117880        NaN      NaN       NaN
## ma3  0.6511022  0.4190479   1.5538    0.1202
## ma4  0.1025123  1.0258273   0.0999    0.9204
## ma5 -0.0615176  0.8064074  -0.0763    0.9392
## ma6 -0.1660367  0.2239127  -0.7415    0.4584
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.556_css_ml
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error   z value  Pr(>|z|)
## ar1 -1.3778040  0.0799176  -17.2403 < 2.2e-16 ***
## ar2 -1.2892199  0.1117786  -11.5337 < 2.2e-16 ***
## ar3 -0.8608119  0.0699047  -12.3141 < 2.2e-16 ***
## ar4 -0.4109327  0.0158816  -25.8748 < 2.2e-16 ***
## ar5 -0.1116446        NaN       NaN       NaN
## ma1 -1.9567380  0.0051894 -377.0614 < 2.2e-16 ***
## ma2  0.3878587        NaN       NaN       NaN
## ma3  0.7209037  0.0212632   33.9038 < 2.2e-16 ***
## ma4  0.1949739  0.0260083    7.4966 6.549e-14 ***
## ma5 -0.2842018  0.0221881  -12.8087 < 2.2e-16 ***
## ma6 -0.0627482  0.0037984  -16.5194 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The Ml and CSS show different significant variable, but the CSS-ML shows most variables are significant.

```
coefs.655 <- model_results[["model.655.ML"]]$coefs
coefs.655_css <- model_results[["model.655.CSS"]]$coefs
coefs.655_css_ml <- model_results[["model.655.CSS-ML"]]$coefs
coefs.655
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.310861        NaN     NaN       NaN
## ar2 -0.999345        NaN     NaN       NaN
## ar3 -0.897462        NaN     NaN       NaN
## ar4 -0.588551        NaN     NaN       NaN
## ar5 -0.391033        NaN     NaN       NaN
## ar6 -0.245013   0.020957 -11.691 < 2.2e-16 ***
## ma1 -2.054326   0.022340 -91.958 < 2.2e-16 ***
## ma2  0.292034        NaN     NaN       NaN
## ma3  1.519441        NaN     NaN       NaN
## ma4 -0.671614        NaN     NaN       NaN
## ma5 -0.085520        NaN     NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.655_css
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.3213840  0.0089669 -147.363 < 2.2e-16 ***
## ar2 -1.1900217        NaN      NaN       NaN
## ar3 -0.8642391  0.0469105  -18.423 < 2.2e-16 ***
## ar4 -0.5541677        NaN      NaN       NaN
## ar5 -0.3608164        NaN      NaN       NaN
## ar6 -0.2703308  0.0175990  -15.361 < 2.2e-16 ***
## ma1 -2.0358621        NaN      NaN       NaN
## ma2  0.4565065        NaN      NaN       NaN
## ma3  0.8842220        NaN      NaN       NaN
## ma4  0.0651627        NaN      NaN       NaN
## ma5 -0.3695750        NaN      NaN       NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.655_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.367792   0.061353 -22.2937 < 2.2e-16 ***
## ar2 -1.255195   0.089943 -13.9555 < 2.2e-16 ***
## ar3 -0.987723   0.082542 -11.9663 < 2.2e-16 ***
## ar4 -0.711300   0.095198  -7.4718 7.910e-14 ***
## ar5 -0.433788   0.076117  -5.6989 1.206e-08 ***
## ar6 -0.253339   0.055743  -4.5447 5.500e-06 ***
## ma1 -1.998380   0.069748 -28.6516 < 2.2e-16 ***
## ma2  0.390598   0.012112  32.2498 < 2.2e-16 ***
## ma3  0.891310   0.087111  10.2319 < 2.2e-16 ***
## ma4  0.070067   0.060696   1.1544 0.2483424
## ma5 -0.353578   0.092097  -3.8392 0.0001234 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The Ml and CSS show most variable is NaN, but the CSS-ML shows most variables are significant.

```
coefs.656 <- model_results[["model.656.ML"]]$coefs
coefs.656_css <- model_results[["model.656.CSS"]]$coefs
coefs.656_css_ml <- model_results[["model.656.CSS-ML"]]$coefs
coefs.656
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error   z value  Pr(>|z|)
## ar1 -1.336320   0.017240  -77.5146 < 2.2e-16 ***
## ar2 -1.145873   0.046712  -24.5305 < 2.2e-16 ***
## ar3 -0.906705   0.071526  -12.6765 < 2.2e-16 ***
## ar4 -0.653103   0.064357  -10.1481 < 2.2e-16 ***
## ar5 -0.427414        NaN       NaN       NaN
## ar6 -0.251143        NaN       NaN       NaN
## ma1 -1.990331   0.015307 -130.0314 < 2.2e-16 ***
## ma2  0.293865   0.025226   11.6493 < 2.2e-16 ***
## ma3  1.123220   0.023557   47.6810 < 2.2e-16 ***
## ma4 -0.099344   0.013308   -7.4653 8.312e-14 ***
## ma5 -0.363832   0.036604   -9.9398 < 2.2e-16 ***
## ma6  0.036559   0.024370    1.5002    0.1336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.656_css
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error    z value Pr(>|z|)
## ar1 -1.3609932  0.0005916 -2300.5166  < 2e-16 ***
## ar2 -1.2755752  0.1022688   -12.4728  < 2e-16 ***
## ar3 -1.0294142  0.0345238   -29.8175  < 2e-16 ***
## ar4 -0.6816172        NaN        NaN      NaN
## ar5 -0.4400167        NaN        NaN      NaN
## ar6 -0.2883002  0.0266992   -10.7981  < 2e-16 ***
## ma1 -1.9669095  0.0219364   -89.6640  < 2e-16 ***
## ma2  0.3998843  0.1571742     2.5442  0.01095 *
## ma3  0.8648120  0.3939442     2.1953  0.02814 *
## ma4 -0.0911877  0.4364028    -0.2090  0.83448
## ma5 -0.1540098  0.1845572    -0.8345  0.40401
## ma6 -0.0525911        NaN        NaN      NaN
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.656_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.360993   0.053385 -25.4940 < 2.2e-16 ***
## ar2 -1.275575   0.073386 -17.3818 < 2.2e-16 ***
## ar3 -1.029415   0.059793 -17.2163 < 2.2e-16 ***
## ar4 -0.681618   0.080466  -8.4708 < 2.2e-16 ***
## ar5 -0.440017   0.068132  -6.4583 1.059e-10 ***
## ar6 -0.288300   0.045391  -6.3515 2.133e-10 ***
## ma1 -1.966666   0.039983 -49.1874 < 2.2e-16 ***
## ma2  0.399824   0.054546   7.3300 2.302e-13 ***
## ma3  0.864812   0.028284  30.5761 < 2.2e-16 ***
## ma4 -0.091056        NaN      NaN       NaN
## ma5 -0.153896   0.024626  -6.2494 4.121e-10 ***
## ma6 -0.052505   0.032391  -1.6210     0.105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The Ml and CSS show some variable are NaN, but the CSS-ML shows most variables are significant.

```
coefs.250 <- model_results[["model.250.ML"]]$coefs
coefs.250_css <- model_results[["model.250.CSS"]]$coefs
coefs.250_css_ml <- model_results[["model.250.CSS-ML"]]$coefs
coefs.250
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.103958   0.063873 -17.284 < 2.2e-16 ***
## ar2 -0.599414   0.016544 -36.231 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.250_css
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.098450   0.071232 -15.421 < 2.2e-16 ***
## ar2 -0.594477   0.071615  -8.301 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.250_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.097207   0.021394 -51.287 < 2.2e-16 ***
## ar2 -0.592466   0.025315 -23.404 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- All test show that all variables is significant

```
coefs.251 <- model_results[["model.251.ML"]]$coefs
coefs.251_css <- model_results[["model.251.CSS"]]$coefs
coefs.251_css_ml <- model_results[["model.251.CSS-ML"]]$coefs
coefs.251
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.9108699  0.0201501  -45.204 < 2.2e-16 ***
## ar2 -0.5424421        NaN      NaN       NaN
## ma1 -0.9995303  0.0086969 -114.929 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.251_css
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.7690197  0.0011655 -659.808 < 2.2e-16 ***
## ar2 -0.4616848  0.0200757  -22.997 < 2.2e-16 ***
## ma1 -1.0536832  0.0053202 -198.052 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.251_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.908284   0.017267 -52.603 < 2.2e-16 ***
## ar2 -0.540379        NaN     NaN       NaN
## ma1 -0.998341   0.014741 -67.728 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The ar2 is insignificant in CSS-ML test

```
coefs.350 <- model_results[["model.350.ML"]]$coefs
coefs.350_css <- model_results[["model.350.CSS"]]$coefs
coefs.350_css_ml <- model_results[["model.350.CSS-ML"]]$coefs
coefs.350
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -1.42468        NaN     NaN      NaN
## ar2 -1.20099        NaN     NaN      NaN
## ar3 -0.55427        NaN     NaN      NaN
```

```
coefs.350_css
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.431656   0.073129 -19.5773 < 2.2e-16 ***
## ar2 -1.210911   0.099933 -12.1173 < 2.2e-16 ***
## ar3 -0.563614   0.073787  -7.6384 2.199e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.350_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -1.431654   0.191556  -7.4738 7.789e-14 ***
## ar2 -1.210904   0.211633  -5.7217 1.055e-08 ***
## ar3 -0.563609   0.015984 -35.2599 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The ML method shows that all variables are NaN while the CSS shows all variables are significant. Then CSS-ML support that all variables are significant

```
coefs.351 <- model_results[["model.351.ML"]]$coefs
coefs.351_css <- model_results[["model.351.CSS"]]$coefs
coefs.351_css_ml <- model_results[["model.351.CSS-ML"]]$coefs
coefs.351
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.181362        NaN     NaN       NaN
## ar2 -1.007812        NaN     NaN       NaN
## ar3 -0.499446        NaN     NaN       NaN
## ma1 -0.999734   0.026736 -37.393 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.351_css
```

```
##
## z test of coefficients:
##
##        Estimate  Std. Error  z value  Pr(>|z|)
## ar1 -1.08183786  0.00023198 -4663.57 < 2.2e-16 ***
## ar2 -0.77947217  0.00020821 -3743.65 < 2.2e-16 ***
## ar3 -0.29514381  0.00173506  -170.11 < 2.2e-16 ***
## ma1 -1.07192621  0.00089621 -1196.07 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.351_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -1.185092        NaN     NaN       NaN
## ar2 -1.010492        NaN     NaN       NaN
## ar3 -0.502496        NaN     NaN       NaN
## ma1 -0.999663   0.015832 -63.141 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The ML method shows that all variables are NaN while the CSS shows all variables are significant. Then CSS-ML support that all variables are NaN.

```
coefs.151 <- model_results[["model.151.ML"]]$coefs
coefs.151_css <- model_results[["model.151.CSS"]]$coefs
coefs.151_css_ml <- model_results[["model.151.CSS-ML"]]$coefs
coefs.151
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.598322   0.019054 -31.401 < 2.2e-16 ***
## ma1 -0.999857   0.019491 -51.297 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.151_css
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.597436   0.074814  -7.9856 1.398e-15 ***
## ma1 -0.893094   0.029694 -30.0763 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefs.151_css_ml
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.589593   0.017780 -33.160 < 2.2e-16 ***
## ma1 -0.999888   0.015009 -66.618 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- All test suggest that all variables is significant

2.Scoring

```
AIC(coefs.355,coefs.356,coefs.455,coefs.456,coefs.555,coefs.556,
     coefs.655,coefs.656,coefs.250,coefs.251,coefs.350,coefs.351,coefs.151)
```

```
##           df      AIC
## coefs.355  9 1015.862
## coefs.356 10 1008.328
## coefs.455 10 1007.636
## coefs.456 11 1004.814
## coefs.555 11 1055.712
## coefs.556 12 1010.093
## coefs.655 12 1001.946
## coefs.656 13 1004.766
## coefs.250  3 1251.631
## coefs.251  4 1149.720
## coefs.350  4 1206.665
## coefs.351  5 1115.734
## coefs.151  3 1190.602
```

```
BIC(coefs.355,coefs.356,coefs.455,coefs.456,coefs.555,coefs.556,
     coefs.655,coefs.656,coefs.250,coefs.251,coefs.350,coefs.351,coefs.151)
```

```
##           df      BIC
## coefs.355  9 1041.531
## coefs.356 10 1036.848
## coefs.455 10 1036.157
## coefs.456 11 1036.186
## coefs.555 11 1087.085
## coefs.556 12 1044.317
## coefs.655 12 1036.171
## coefs.656 13 1041.843
## coefs.250  3 1260.187
## coefs.251  4 1161.129
## coefs.350  4 1218.073
## coefs.351  5 1129.994
## coefs.151  3 1199.159
```

```
sort.score <- function(x, score = c("bic", "aic")){
  if (score == "aic"){
    x[with(x, order(AIC)),]
  } else if (score == "bic") {
    x[with(x, order(BIC)),]
  } else {
    warning('score = "x" only accepts valid arguments ("aic","bic")')
  }
}

sort.score(AIC(coefs.355,coefs.356,coefs.455,coefs.456,coefs.555,coefs.556,
     coefs.655,coefs.656,coefs.250,coefs.251,coefs.350,coefs.351,coefs.151), score = "aic")
```

```
##            df      AIC
## coefs.655 12 1001.946
## coefs.656 13 1004.766
## coefs.456 11 1004.814
## coefs.455 10 1007.636
## coefs.356 10 1008.328
## coefs.556 12 1010.093
## coefs.355  9 1015.862
## coefs.555 11 1055.712
## coefs.351  5 1115.734
## coefs.251  4 1149.720
## coefs.151  3 1190.602
## coefs.350  4 1206.665
## coefs.250  3 1251.631
```

```
sort.score(BIC(coefs.355,coefs.356,coefs.455,coefs.456,coefs.555,coefs.556,
     coefs.655,coefs.656,coefs.250,coefs.251,coefs.350,coefs.351,coefs.151), score = "bic")
```

```
##            df      BIC
## coefs.455 10 1036.157
## coefs.655 12 1036.171
## coefs.456 11 1036.186
## coefs.356 10 1036.848
## coefs.355  9 1041.531
## coefs.656 13 1041.843
## coefs.556 12 1044.317
## coefs.555 11 1087.085
## coefs.351  5 1129.994
## coefs.251  4 1161.129
## coefs.151  3 1199.159
## coefs.350  4 1218.073
## coefs.250  3 1260.187
```

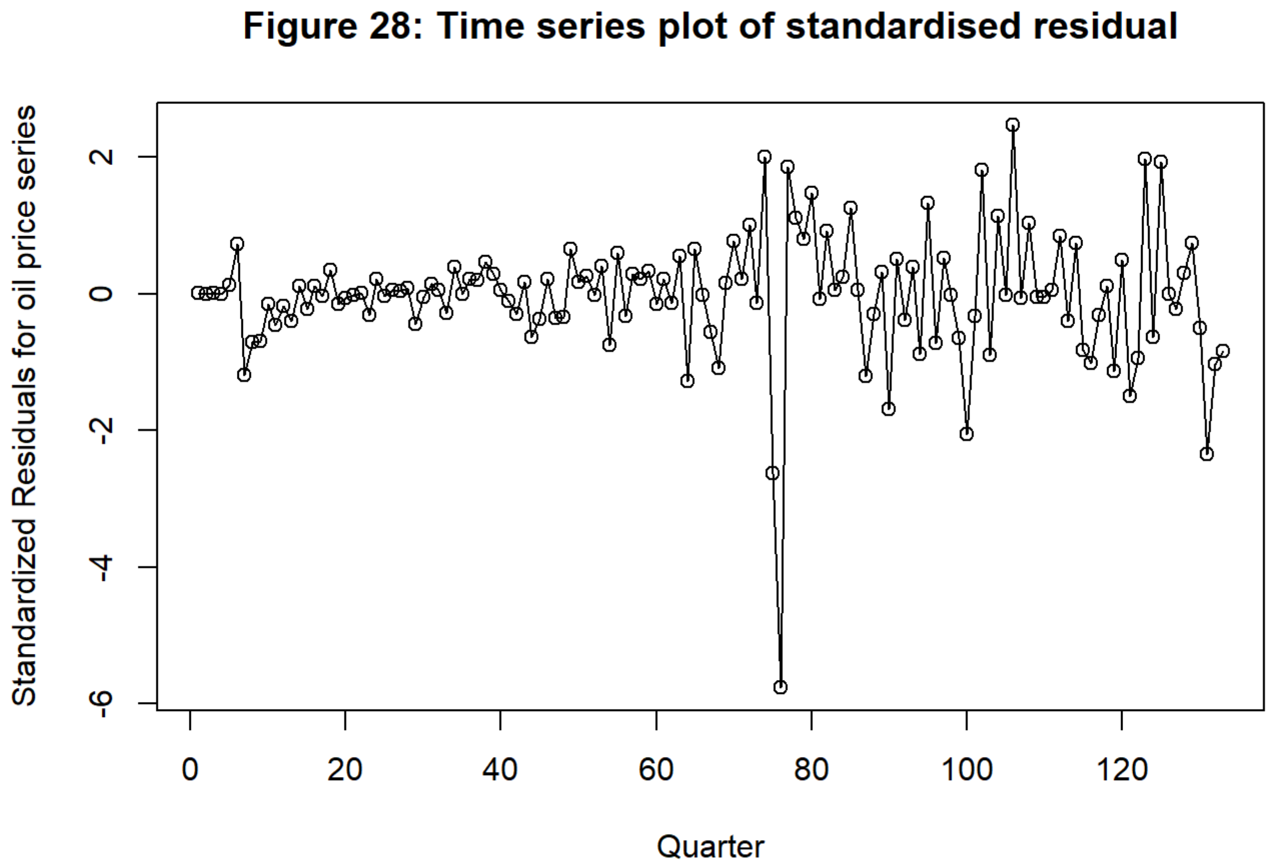
- From the AIC and BIC score, the ARIMA(4,5,5) and ARIMA (6,5,5) have the highest score.

```r
calculate_accuracy <- function(p, d, q) {
  model <- Arima(Oil_prices.TS, order=c(p, d, q), method='ML')
  accuracy_vals <- accuracy(model)[1:7]
  return(accuracy_vals)
}

order_values <- list(
  c(3, 5, 5),
  c(3, 5, 6),
  c(4, 5, 5),
  c(4, 5, 6),
  c(5, 5, 5),
  c(5, 5, 6),
  c(6, 5, 5),
  c(6, 5, 6),
  c(2, 5, 0),
  c(2, 5, 1),
  c(3, 5, 0),
  c(3, 5, 1),
  c(1, 5, 1)
)

df.Smodels <- data.frame(matrix(NA, nrow = length(order_values), ncol = 7))
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE", "MASE", "ACF1")
rownames(df.Smodels) <- c(
  "ARIMA(3,5,5)", "ARIMA(3,5,6)", "ARIMA(4,5,5)", "ARIMA(4,5,6)",
  "ARIMA(5,5,5)", "ARIMA(5,5,6)", "ARIMA(6,5,5)", "ARIMA(6,5,6)",
  "ARIMA(2,5,0)","ARIMA(2,5,1)", "ARIMA(3,5,0)", "ARIMA(3,5,1)",
  "ARIMA(1,5,1)"
)

for (i in seq_along(order_values)) {
  order <- order_values[[i]]
  accuracy_vals <- calculate_accuracy(order[1], order[2], order[3])
  df.Smodels[i, ] <- accuracy_vals
}

df.Smodels
```

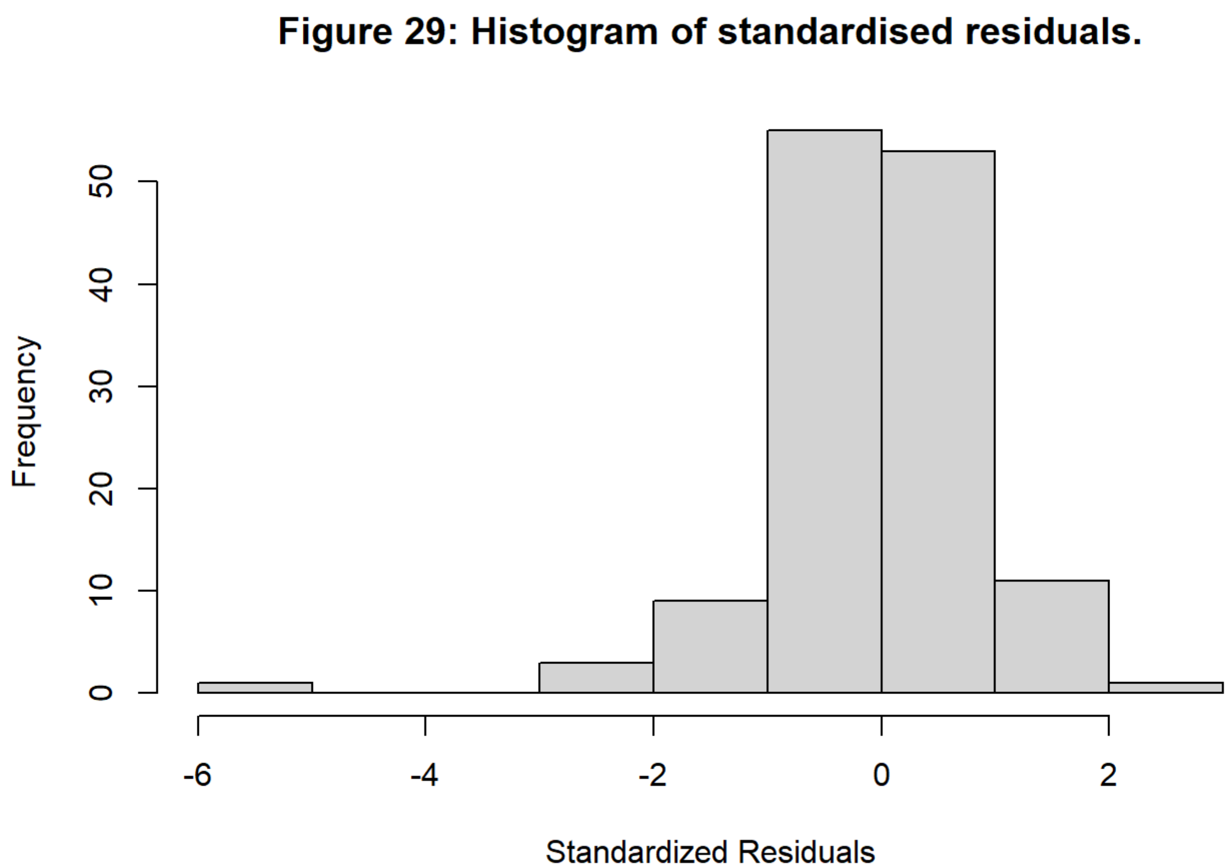
```
##                        ME      RMSE       MAE         MPE     MAPE     MASE
## ARIMA(3,5,5) -0.43239449 10.378157  6.489478 -0.8830866 13.55274 1.128253
## ARIMA(3,5,6) -0.41050046  9.969228  6.287695 -0.6873986 13.16448 1.093171
## ARIMA(4,5,5) -0.36963256  9.873129  6.365109 -0.6107049 13.35152 1.106630
## ARIMA(4,5,6) -0.41751769  9.690003  6.124332 -0.8553681 13.10067 1.064769
## ARIMA(5,5,5) -0.28759360 11.516486  7.273997 -0.6999009 14.90938 1.264648
## ARIMA(5,5,6) -0.40042834  9.779435  6.211411 -0.7591101 13.09420 1.079908
## ARIMA(6,5,5) -0.34790061  9.388846  6.030124 -0.6018272 13.02143 1.048390
## ARIMA(6,5,6) -0.36561828  9.479107  6.015892 -0.6746306 12.89362 1.045916
## ARIMA(2,5,0) -0.05307052 30.145302 19.260245  0.2563949 41.57364 3.348563
## ARIMA(2,5,1) -0.58110217 19.612249 12.902636 -2.2497083 27.32089 2.243237
## ARIMA(3,5,0) -0.12081059 24.986437 15.545490 -0.2508913 31.73346 2.702720
## ARIMA(3,5,1) -0.51035439 16.927239 10.705009 -1.7331172 21.76283 1.861160
## ARIMA(1,5,1) -0.81556114 23.323055 14.396583 -2.9912836 31.59350 2.502972
##                     ACF1
## ARIMA(3,5,5) -0.08709690
## ARIMA(3,5,6) -0.04112305
## ARIMA(4,5,5) -0.01207303
## ARIMA(4,5,6) -0.01753476
## ARIMA(5,5,5) -0.09126565
## ARIMA(5,5,6) -0.02754314
## ARIMA(6,5,5) -0.04099118
## ARIMA(6,5,6) -0.06524697
## ARIMA(2,5,0) -0.32593235
## ARIMA(2,5,1) -0.26436561
## ARIMA(3,5,0) -0.28159607
## ARIMA(3,5,1) -0.20109921
## ARIMA(1,5,1) -0.30722606
```

-

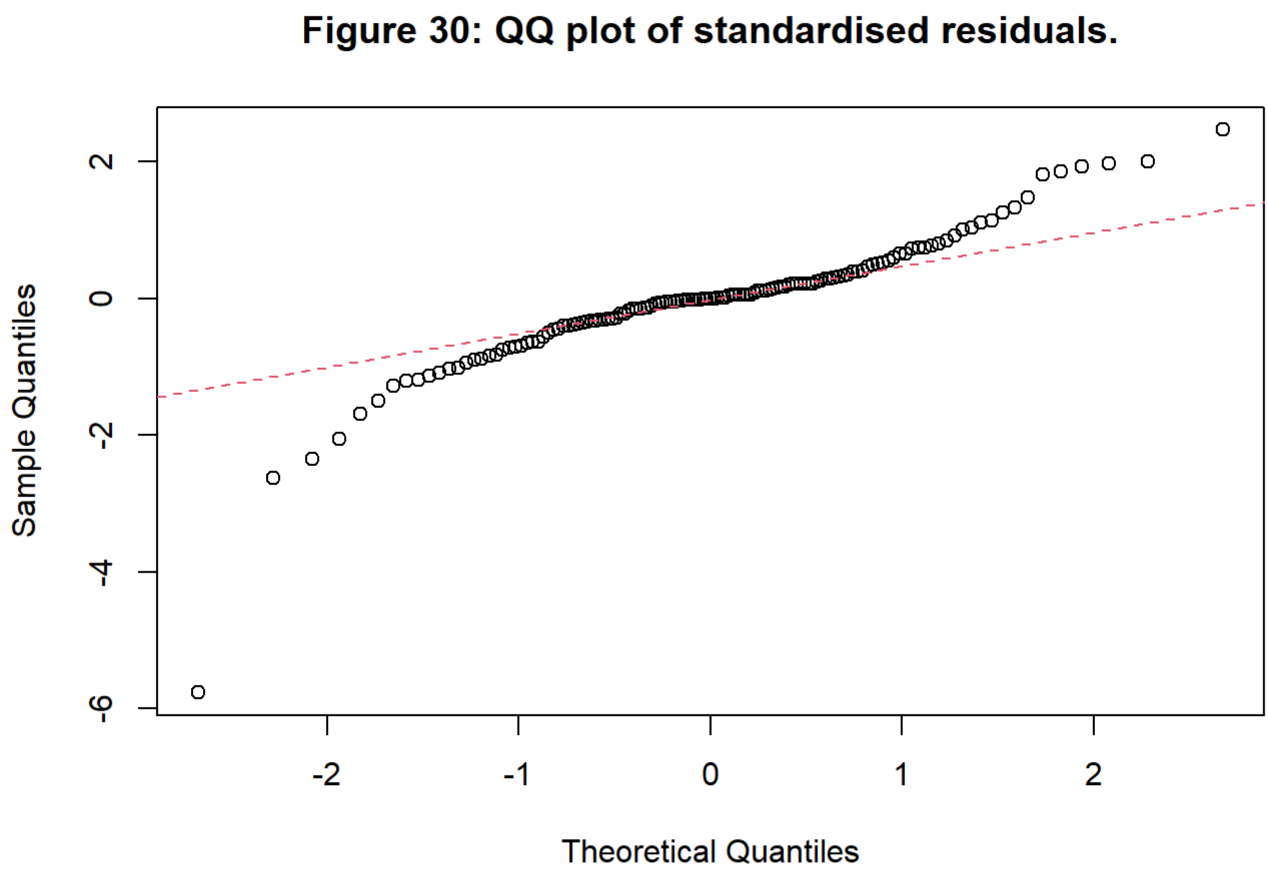
```
model.655 <- Arima(Oil_prices.TS, order = c(6, 5, 5), method = 'CSS-ML')
model.655Res = rstandard(model.655)
plot(model.655Res, xlab='Quarter',
     ylab='Standardized Residuals for oil price series',type='o', main = "Figure 28: Time ser
ies plot of standardised residual")
```




```
hist(model.655Res, xlab='Standardized Residuals',type='o',
     main = "Figure 29: Histogram of standardised residuals.")
```



```
qqnorm(model.655Res, main = "Figure 30: QQ plot of standardised residuals.")
qqline(model.655Res, col = 2, lwd = 1, lty = 2)
```

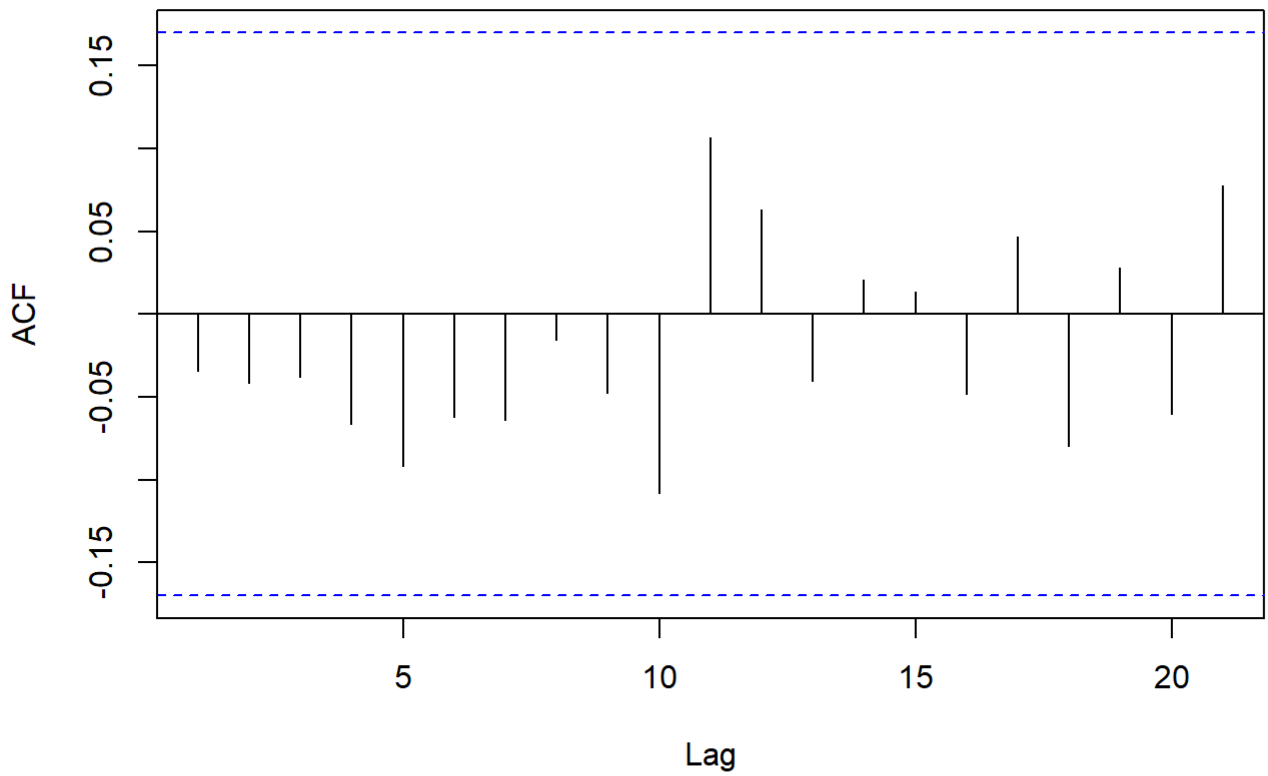



```
shapiro.test(model.655Res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model.655Res
## W = 0.86517, p-value = 1.185e-09
```

```
acf(model.655Res, main = "Figure 31: ACF plot of standardised residuals." )
```

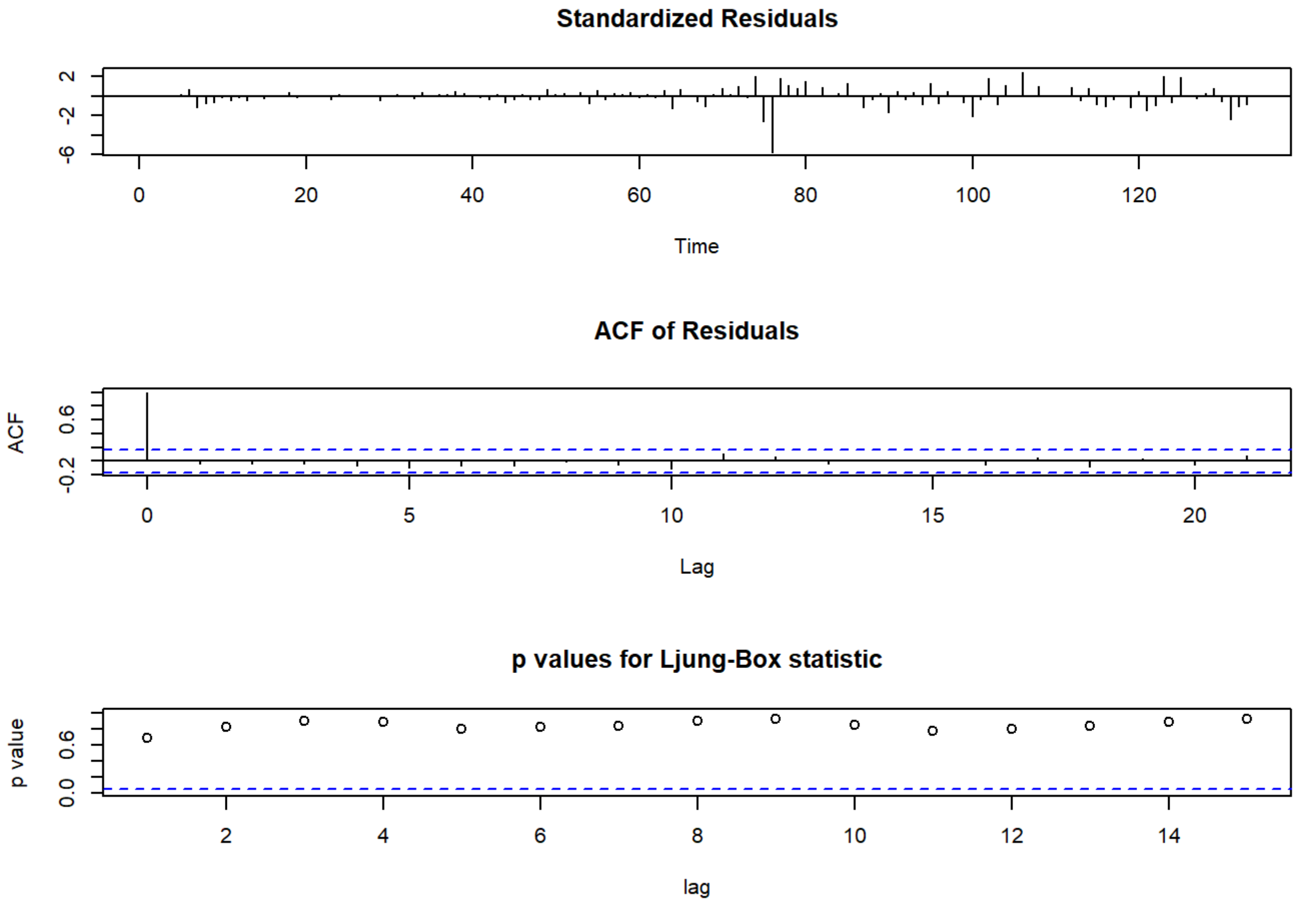
Figure 31: ACF plot of standardised residuals.




```
Box.test(model.655Res, type = "Ljung-Box")
```

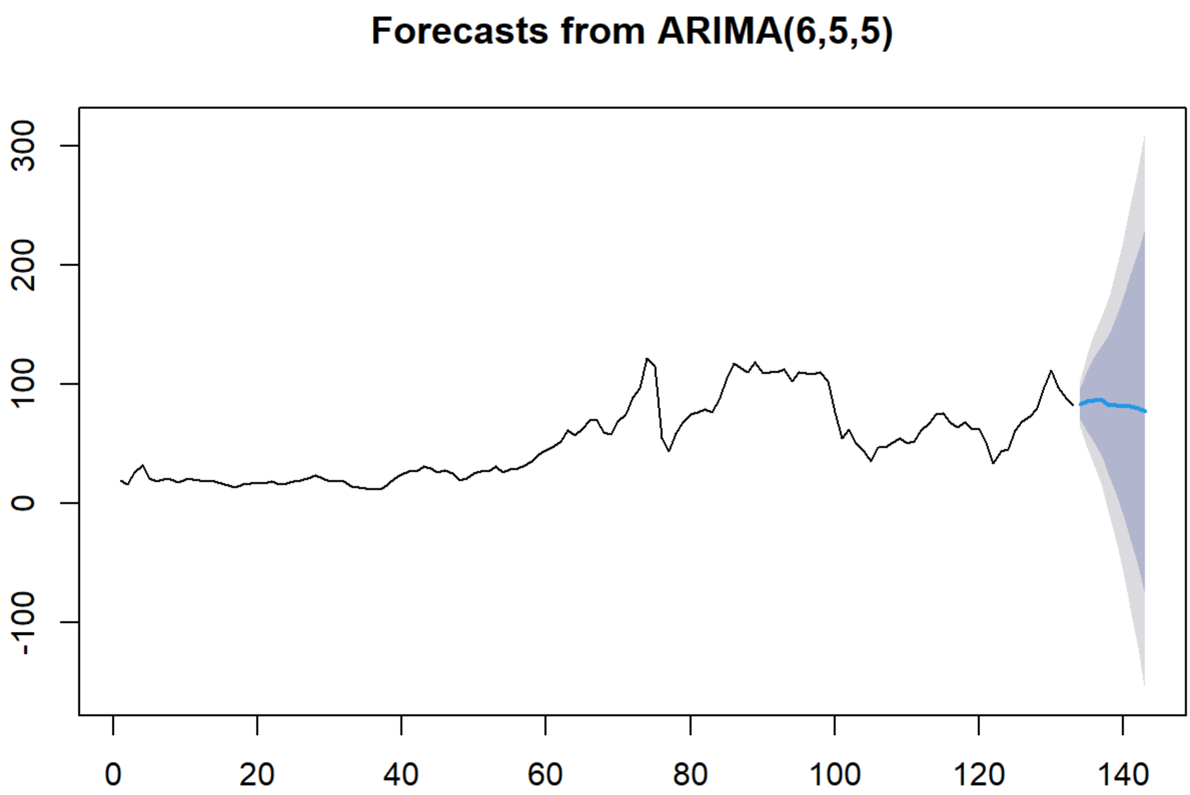
```
##
##  Box-Ljung test
##
## data:  model.655Res
## X-squared = 0.15857, df = 1, p-value = 0.6905
```

```
tsdiag(model.655,gof=15,omit.initial=F)
```



- The figure 28 shows the graph for standardised residuals, majority of dot is close to 0 or in range from -2 to 2 with exception from the quarter of 2008 and the recent quarter. The graph also show the changing variance as after the drop in 2008 the variance become wider. The histogram (Figure 29) show a symmetric shape from -2 to 2 are which is normal distributed. The qqplot (Figure 30) shows that the residuals in two tail is not in the line imply the residuals is not normally distributed. The ACF plot (Figure 29) suggest that residuals are normally distributed as all the bar is included in confidence interval zone. The Ljung-box result suggest that there is no serial correlation in residuals. Lastly, the Shapiro-Wilk concludes that the residuals is not normally distributed.
- Overall, the the model specification in previous part shows that all method resulted in not normally distributed. In the ARIMA, some methods support that the residuals is normally distributed but other method against it. In conclusion, ARIMA model is more proper to predict the oil price.

```
model.655Afrc = forecast::forecast(model.655, h = 10)
plot(model.655Afrc)
```




```
model.655Afrc
```

```
##     Point Forecast      Lo 80     Hi 80      Lo 95    Hi 95
## 134       83.59700  70.574083  96.61992   63.680168 103.5138
## 135       85.76090  60.577658 110.94413   47.246457 124.2753
## 136       87.13208  51.229217 123.03494   32.223391 142.0408
## 137       86.42406  39.534238 133.31387   14.712267 158.1358
## 138       82.42839  22.798195 142.05859   -8.768125 173.6249
## 139       82.61712   8.279863 156.95438  -31.071905 196.3062
## 140       81.71794  -9.160247 172.59613  -57.268253 220.7041
## 141       81.39232 -28.909732 191.69438  -87.300113 250.0848
## 142       79.71834 -51.654908 211.09160 -121.199707 280.6364
## 143       77.24739 -77.239329 231.73411 -159.019659 313.5144
```

- The ARIMA predicts that over the next 10 quarter the oil price will slowly decrease in price

V. Conclusion:

- The oil price is complicated and it is hard to predict precisely and the time factor seem not the only important factor affect oil price. As the data also shows that the oil prices had been affected hugely by economic events at 2008. Then the prediction based on the ARIMA(6,5,6) could be not the right method to predict the future oil price as some powerful tests detect non-normality in residuals. In conclusion, based on the given materials, this is the best model to predict the oil price.