Econometria Avançada

Guilherme Dias

2022-05-31

Importing the required bibs for work

```
library(tseries)
```

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

```
require(urca)
```

```
## Loading required package: urca
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(fpp2)
```

```
## ── Attaching packages ──────────────────────────────────────────── fpp2 2.4 ──
```

```
## ✔ forecast  8.16     ✔ expsmooth 2.3
## ✔ fma       2.4
```

```
##
```

```
library(vars)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following objects are masked from 'package:fma':
##
##     cement, housing, petrol
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
## Loading required package: sandwich
```

```
##
## Attaching package: 'strucchange'
```

```
## The following object is masked from 'package:stringr':
##
##     boundary
```

```
## Loading required package: lmtest
```

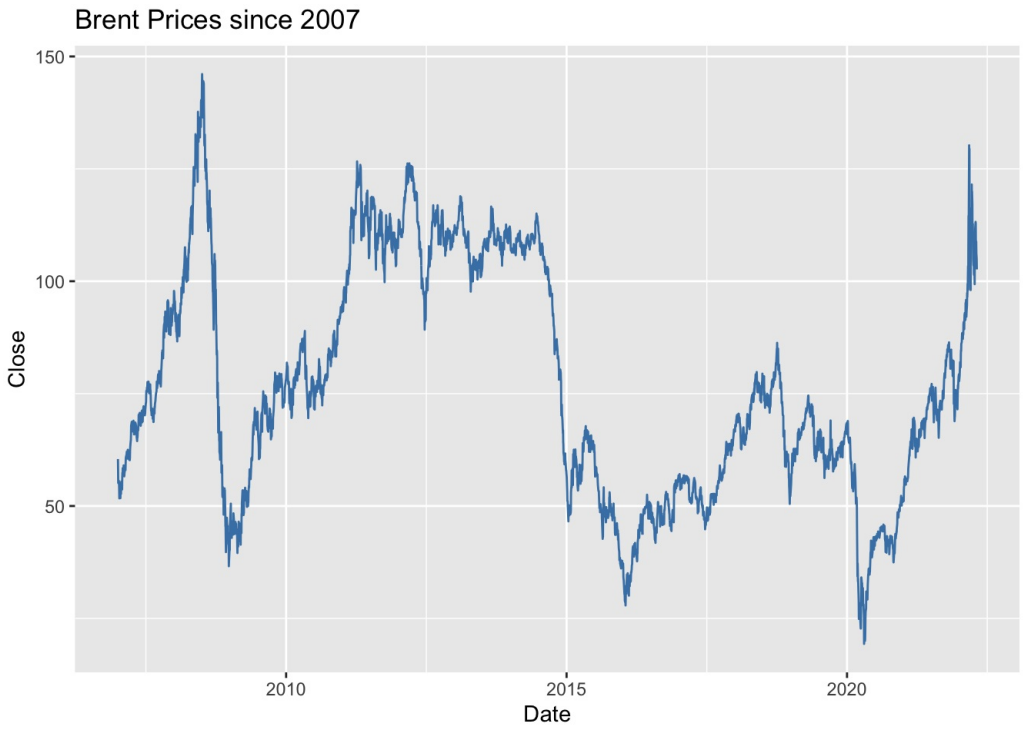
```
library(forecast)
require(lmtest)
library(readxl)
```

Importing the data

```
brent_prices <- read_xlsx("brent_prices.xlsx")
Petr4_prices <- read_xlsx("Petr4_prices.xlsx")
```

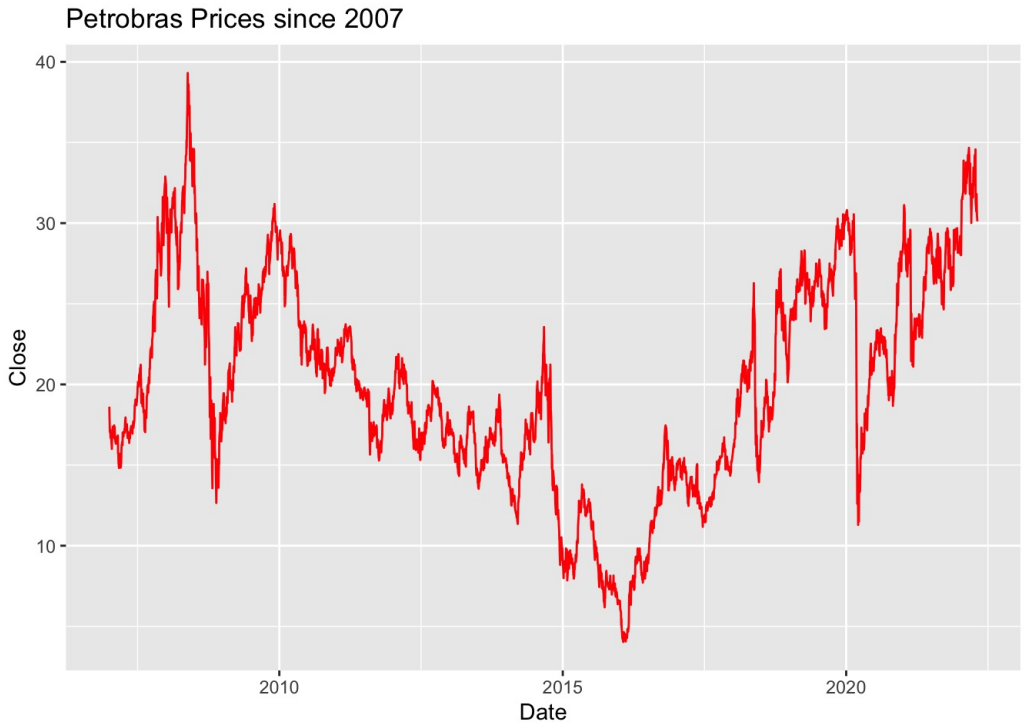
Brent Oil Prices Graph

```
ggplot(brent_prices, aes(x=Date)) +
  geom_line(aes(y = Close), color = "steelblue") +
  labs(title = 'Brent Prices since 2007')
```



Petrobrás stock prices graph

```
ggplot(Petr4_prices, aes(x=Date)) +
  geom_line(aes(y = `Close`), color = "red") +
  labs(title = 'Petrobras Prices since 2007')
```



Managing the data

```
final_data <- merge(Petr4_prices, brent_prices, by = 'Date')
names(final_data) <- c('Date','Petrobras', 'Brent Oil Prices')
normalized_final_data <- data_frame(final_data$Date,final_data[,2] / final_data[1,2],final_data[,3]/final_data[1,
3])
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
names(normalized_final_data) <- c('Date','Petrobras', 'Brent Oil Prices')
```

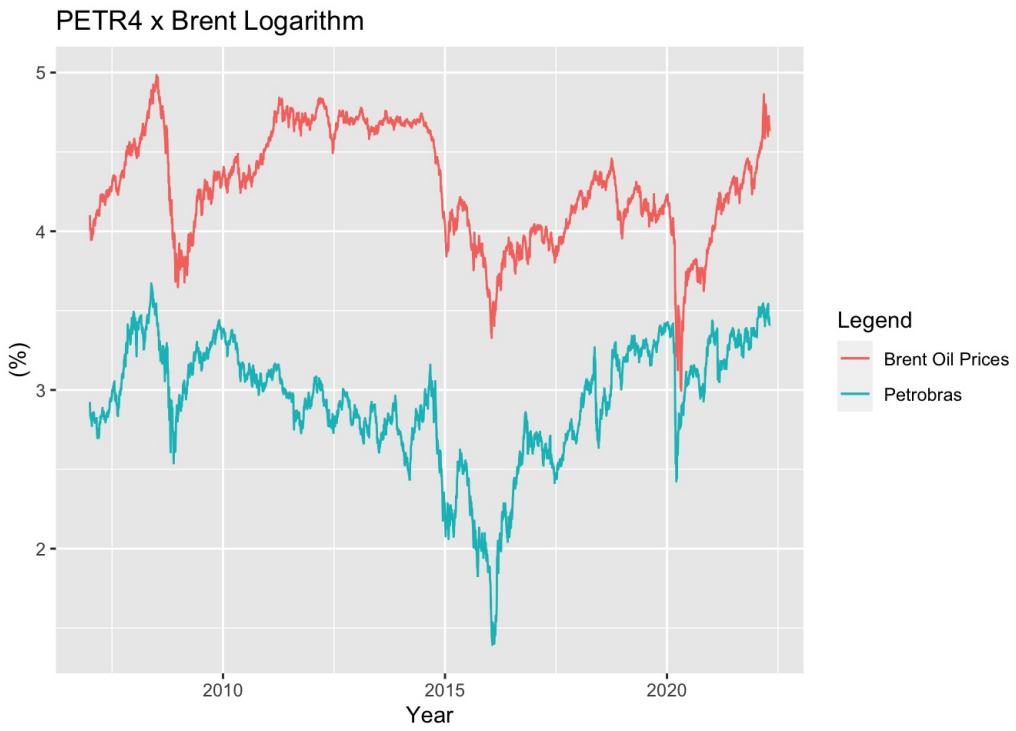
Brent x Petrobrás Graph in normalized scale

```
ggplot(normalized_final_data, aes(x = Date)) +
  geom_line(aes(y = Petrobras, color = "Petrobras")) +
  geom_line(aes(y = `Brent Oil Prices`, color = "Brent Oil Prices")) +
  labs(x = "Year",
       y = "(%)",
       color = "Legend")
```

Transforming the data to the logarithm scale and plotting Petr4 x Brent Graph

```
log_data <- log(final_data[,2:3])
log_final_data <- data_frame(final_data$Date,log(final_data[,2:3]))
log_data <- data_frame(final_data$Date,log(final_data[,2:3]))
names(log_final_data) <- c('Date','Petrobras', 'Brent Oil Prices')
names(log_data) <- c('Date','Petrobras', 'Brent Oil Prices')

ggplot(log_final_data, aes(x = Date)) +
  geom_line(aes(y = Petrobras, color = "Petrobras")) +
  geom_line(aes(y = `Brent Oil Prices`, color = "Brent Oil Prices")) +
  labs(x = "Year",
       y = "(%)",
       color = "Legend",
       title = 'PETR4 x Brent Logarithm')
```



Augmented dickey fuller test for Brent Prices

```
adf_test_brent <- ur.df(log_data$`Brent Oil Prices`,selectlags = "BIC")
summary(adf_test_brent)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.277534 -0.010297  0.000714  0.010998  0.192156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     2.063e-05  9.136e-05   0.226    0.821
## z.diff.lag -1.718e-02  1.627e-02  -1.056    0.291
##
## Residual standard error: 0.02415 on 3778 degrees of freedom
## Multiple R-squared:  0.0003075,  Adjusted R-squared:  -0.0002218
## F-statistic: 0.581 on 2 and 3778 DF,  p-value: 0.5594
##
##
## Value of test-statistic is: 0.2258
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

Augmented dickey fuller test for Brent Prices with drift

```
adf_test_brent <- ur.df(log_data$`Brent Oil Prices`,type="drift",selectlags = "BIC")
summary(adf_test_brent)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.278572 -0.010430  0.000732  0.011012  0.189506
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.009944   0.004846   2.052   0.0402 *
## z.lag.1     -0.002284   0.001127  -2.027   0.0427 *
## z.diff.lag  -0.015984   0.016269  -0.982   0.3259
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02414 on 3777 degrees of freedom
## Multiple R-squared:  0.001382,   Adjusted R-squared:  0.0008529
## F-statistic: 2.613 on 2 and 3777 DF,  p-value: 0.07345
##
##
## Value of test-statistic is: -2.027 2.1311
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

Augmented dickey fuller test for Brent Prices with trend

```
adf_test_brent <- ur.df(log_data$`Brent Oil Prices`,type="trend",selectlags = "BIC")
summary(adf_test_brent)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.278335 -0.010410  0.000744  0.011055  0.189507
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.188e-02  5.583e-03   2.128   0.0334 *
## z.lag.1     -2.616e-03  1.223e-03  -2.139   0.0325 *
## tt          -2.731e-07  3.905e-07  -0.699   0.4844
## z.diff.lag  -1.576e-02  1.627e-02  -0.969   0.3328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02414 on 3776 degrees of freedom
## Multiple R-squared:  0.001511,   Adjusted R-squared:  0.0007177
## F-statistic: 1.905 on 3 and 3776 DF,  p-value: 0.1266
##
##
## Value of test-statistic is: -2.1394 1.5836 2.2987
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

Augmented dickey fuller test for Petrobrás stock prices

```
adf_test_petr <- ur.df(log_data$Petrobras,type="none", selectlags = "BIC")
summary(adf_test_petr)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35544 -0.01446  0.00077  0.01486  0.19378
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     1.575e-06  1.638e-04   0.010   0.9923
## z.diff.lag -3.002e-02  1.626e-02  -1.846   0.0649 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02972 on 3778 degrees of freedom
## Multiple R-squared:  0.0009016,  Adjusted R-squared:  0.0003727
## F-statistic: 1.705 on 2 and 3778 DF,  p-value: 0.182
##
##
## Value of test-statistic is: 0.0096
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

Augmented dickey fuller test for Petrobrás stock prices with drift

```
adf_test_petr <- ur.df(log_data$Petrobras,type="drift", selectlags = "BIC")
summary(adf_test_petr)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35487 -0.01459  0.00065  0.01476  0.19289
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.008233   0.003760   2.189   0.0286 *
## z.lag.1     -0.002764   0.001274  -2.170   0.0301 *
## z.diff.lag  -0.028561   0.016266  -1.756   0.0792 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0297 on 3777 degrees of freedom
## Multiple R-squared:  0.002147,   Adjusted R-squared:  0.001618
## F-statistic: 4.063 on 2 and 3777 DF,  p-value: 0.01727
##
##
## Value of test-statistic is: -2.1699 2.3964
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.43 -2.86 -2.57
## phi1  6.43  4.59  3.78
```

Augmented dickey fuller test for Petrobrás stock prices with trend

```
adf_test_petr <- ur.df(log_data$Petrobras,type="trend", selectlags = "BIC")
summary(adf_test_petr)
```

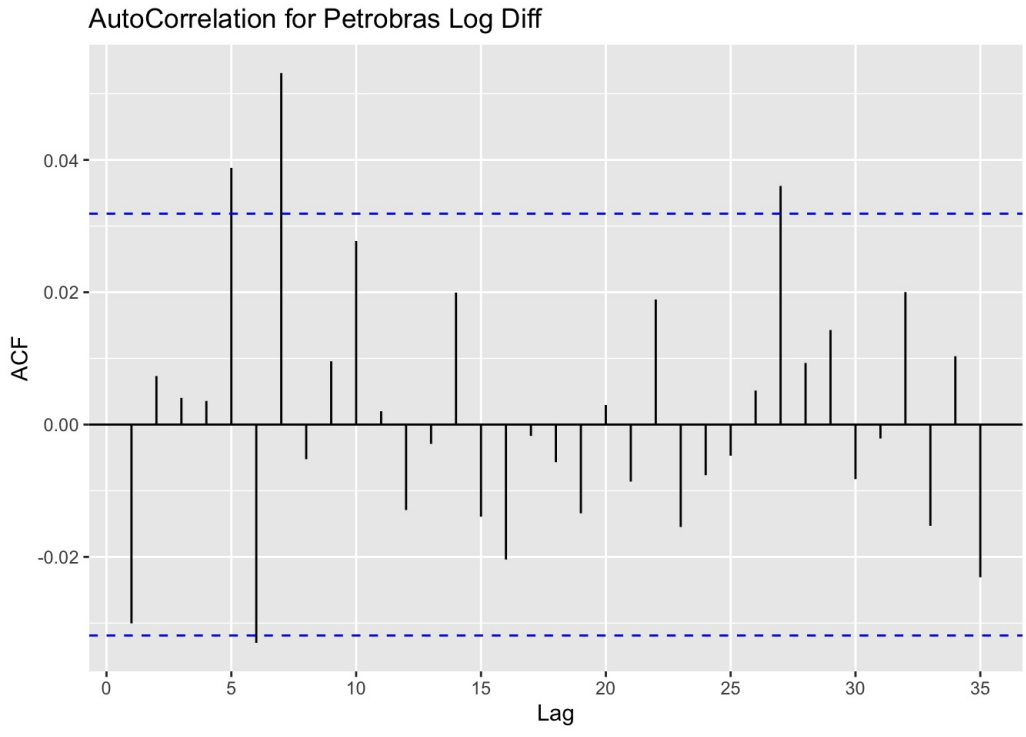
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35511 -0.01455  0.00066  0.01472  0.19265
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.882e-03  3.874e-03   2.035   0.0419 *
## z.lag.1     -2.753e-03  1.275e-03  -2.160   0.0308 *
## tt           1.674e-07  4.429e-07   0.378   0.7055
## z.diff.lag  -2.861e-02  1.627e-02  -1.759   0.0787 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02971 on 3776 degrees of freedom
## Multiple R-squared:  0.002185,   Adjusted R-squared:  0.001392
## F-statistic: 2.756 on 3 and 3776 DF,  p-value: 0.04095
##
##
## Value of test-statistic is: -2.1599 1.6449 2.425
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.96 -3.41 -3.12
## phi2  6.09  4.68  4.03
## phi3  8.27  6.25  5.34
```

Log Returns for the data

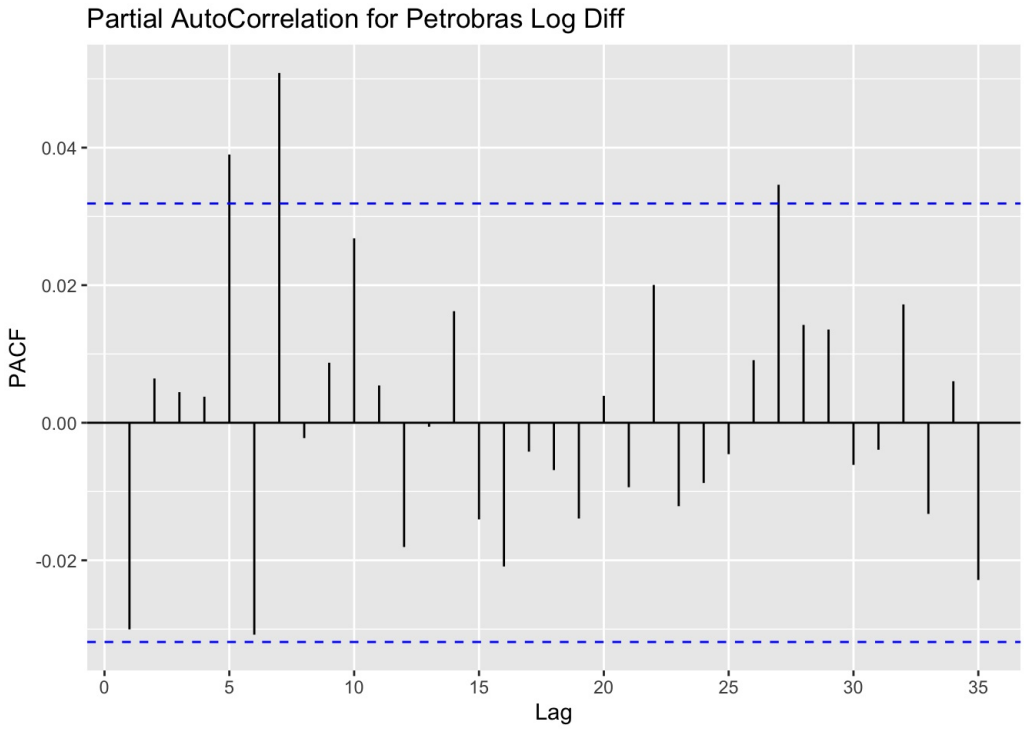
```
log_ret_petr <- diff(log_data$Petrobras)
log_ret_brent <- diff(log_data$`Brent Oil Prices`)
```

ACF, PACF and series of returns for Petrobras

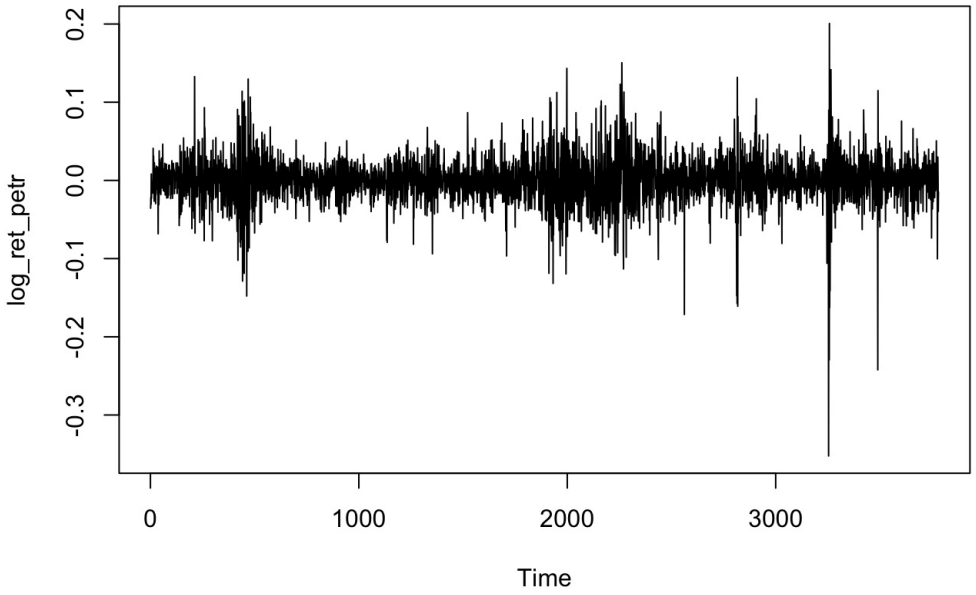
```
ggAcf(log_ret_petr) + labs(title = "AutoCorrelation for Petrobras Log Diff")
```




```
ggPacf(log_ret_petr) + labs(title = "Partial AutoCorrelation for Petrobras Log Diff")
```



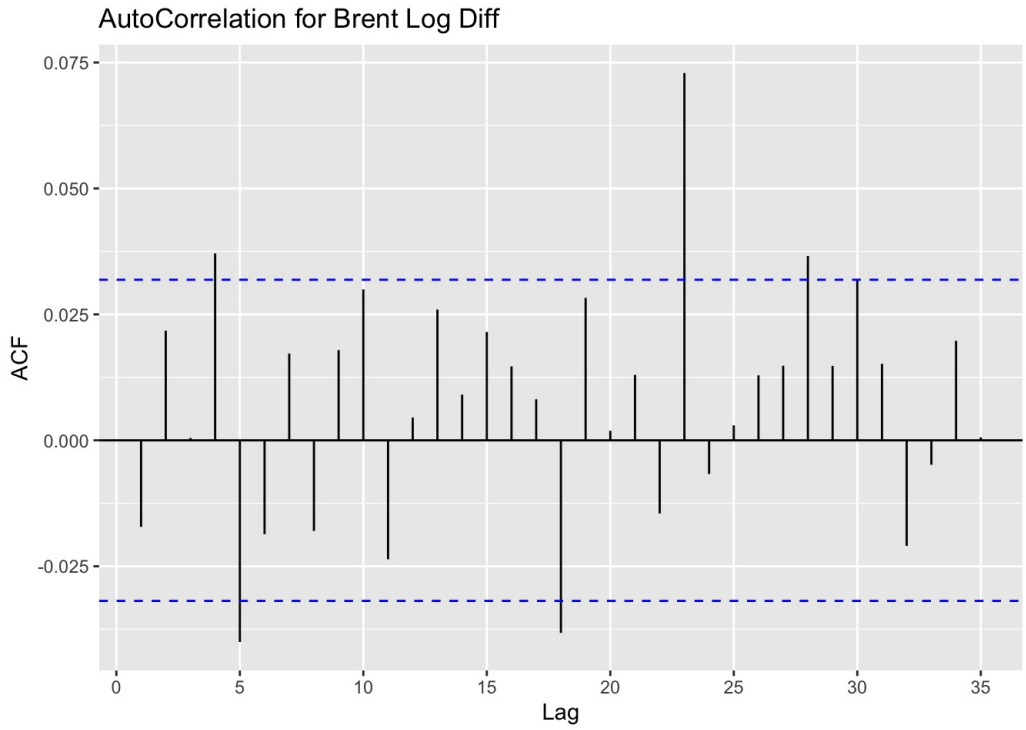

```
ts.plot(log_ret_petr) %>%
  labs(title = "Log of returns for Petrobras")
```



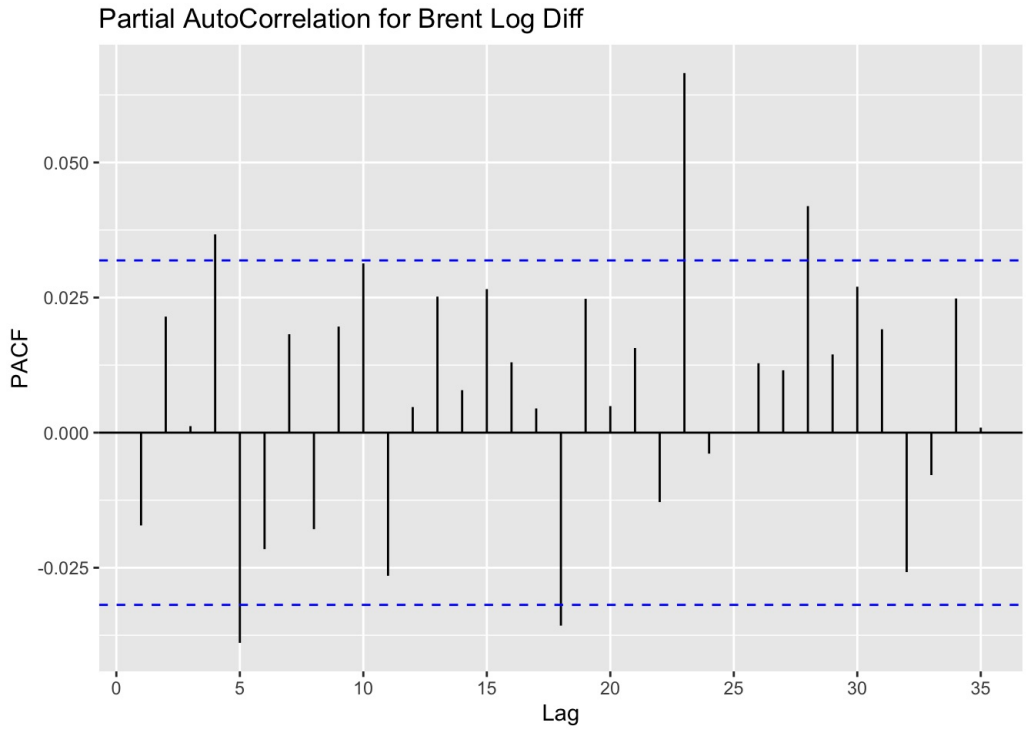

```
## [[1]]
## NULL
##
## $title
## [1] "Log of returns for Petrobras"
##
## attr(,"class")
## [1] "labels"
```

ACF, PACF and series of returns for Petrobras

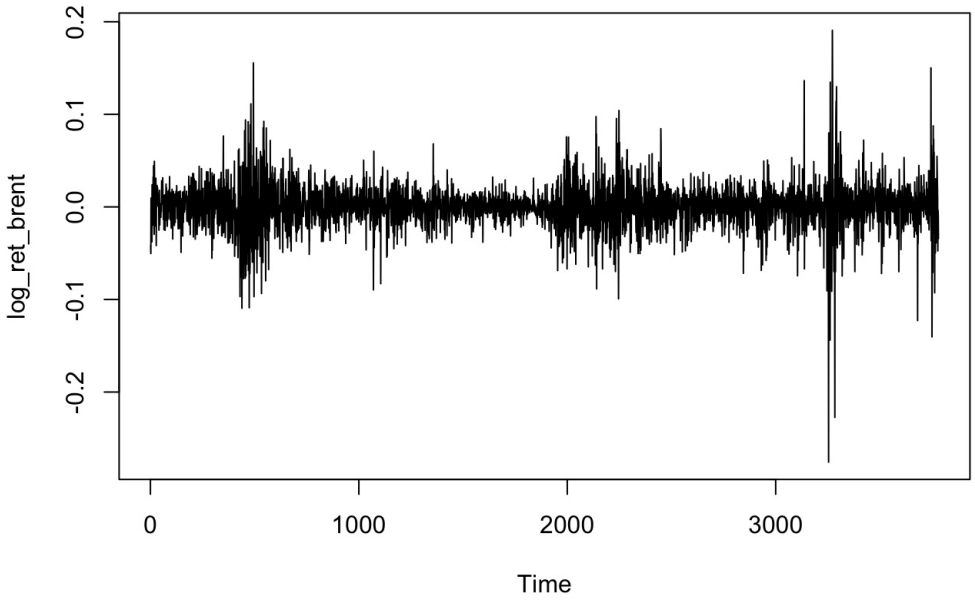
```
log_ret_brent <- diff(log_data$`Brent Oil Prices`)
ggAcf(log_ret_brent) + labs(title = "AutoCorrelation for Brent Log Diff")
```




```
ggPacf(log_ret_brent) + labs(title = "Partial AutoCorrelation for Brent Log Diff")
```




```
ts.plot(log_ret_brent) %>%
  labs(title = "Log of returns for Brent Oil Prices")
```




```
## [[1]]
## NULL
##
## $title
## [1] "Log of returns for Brent Oil Prices"
##
## attr(,"class")
## [1] "labels"
```

Specifying ARIMA model for log returns of Petrobras

```
petr_model <- Arima(log_data$Petrobras, order=c(0,0,1), include.constant = TRUE)
summary(petr_model)
```

```
## Series: log_data$Petrobras
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##       0.9473  2.9275
## s.e.  0.0036  0.0063
##
## sigma^2 = 0.04005:  log likelihood = 717.93
## AIC=-1429.85   AICc=-1429.85   BIC=-1411.14
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE     MASE
## Training set 2.344039e-05 0.2000743 0.1537669 -1.071054 5.758847 7.486993
##                   ACF1
## Training set 0.8980519
```

```
coeftest(petr_model)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ma1       0.9472680  0.0035935  263.61 < 2.2e-16 ***
## intercept 2.9274529  0.0063343  462.16 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Specifying ARIMA model for log returns of Brent

```
brent_model <- Arima(log_data$`Brent Oil Prices`, order=c(0,0,1), include.constant = TRUE)
summary(brent_model)
```

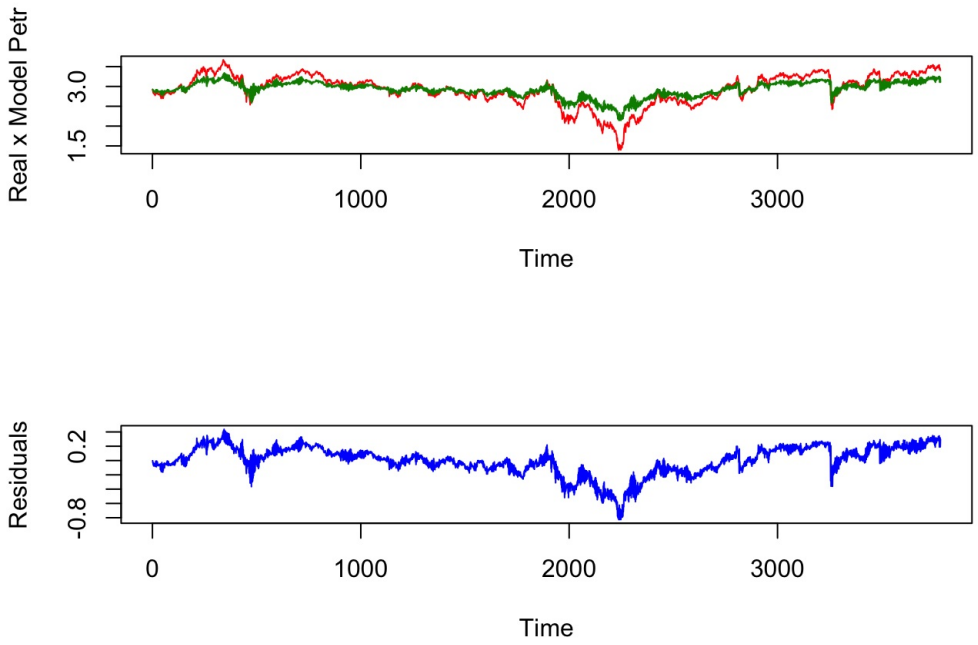
```
## Series: log_data$`Brent Oil Prices`
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##          ma1    mean
##       0.9537  4.2858
## s.e.  0.0035  0.0058
##
## sigma^2 = 0.03353:  log likelihood = 1053.93
## AIC=-2101.86   AICc=-2101.85   BIC=-2083.15
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE     MASE
## Training set 0.0001543414 0.1830627 0.1504547 -0.350834 3.578295 9.317815
##                   ACF1
## Training set 0.9008318
```

```
coeftest(brent_model)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ma1       0.9537395  0.0035273  270.39 < 2.2e-16 ***
## intercept 4.2858222  0.0058150  737.03 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

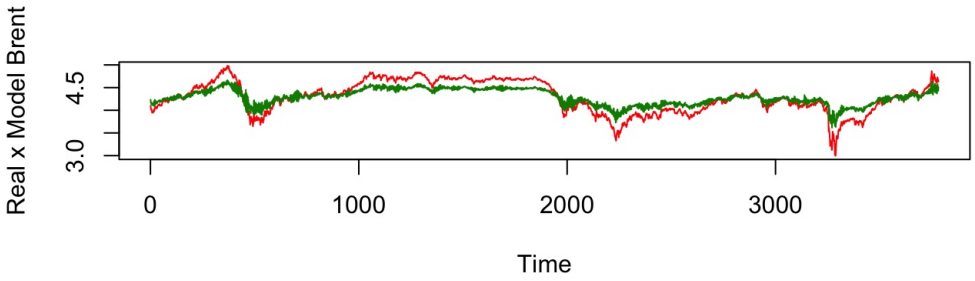
Real time series x Fitted Values for the model specified for Petrobras

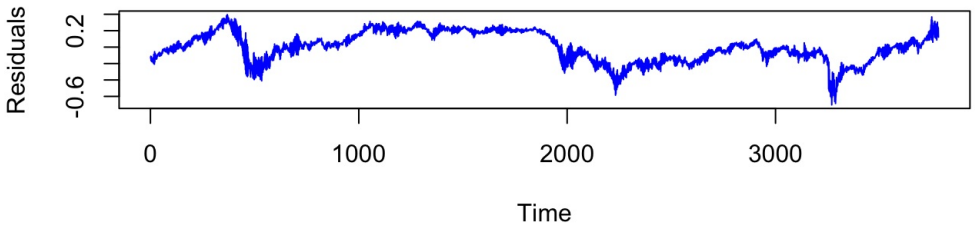
```
par(mfrow=c(2,1))
ts.plot(log_data$Petrobras, fitted(petr_model), col=c("red", "green4"), ylab = "Real x Model Petr")
ts.plot(residuals(petr_model), col="blue", ylab="Residuals")
```



Real time series x Fitted Values for the model specified for Brent

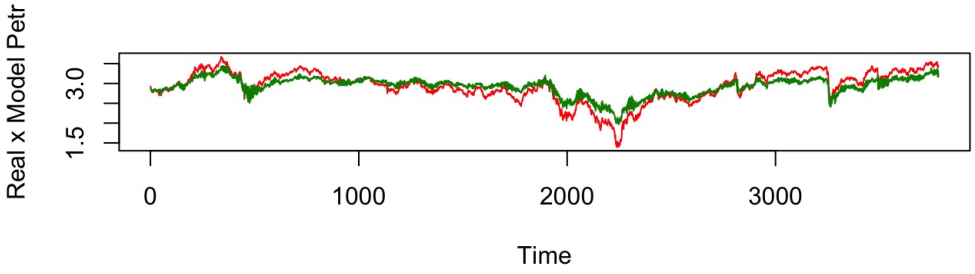
```
par(mfrow=c(2,1))
ts.plot(log_data$`Brent Oil Prices`, fitted(brent_model), col=c("red", "green4"), ylab = "Real x Model Brent")
ts.plot(residuals(brent_model), col="blue", ylab="Residuals")
```

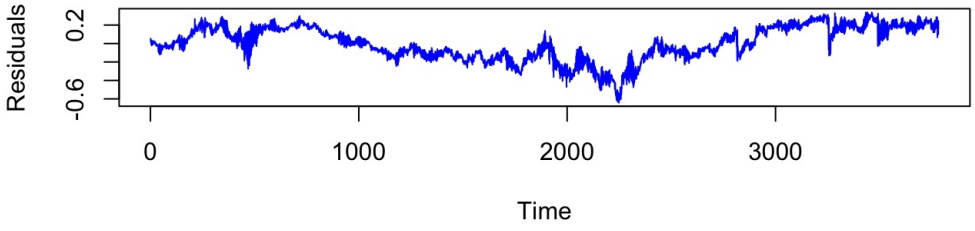




Specifying and model to predict Petrobras prices with Brent Prices

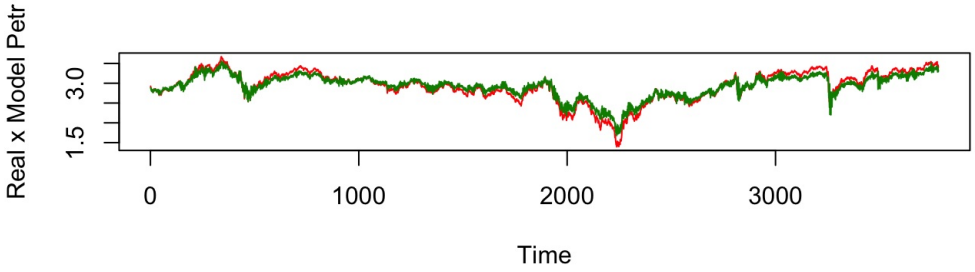
```
par(mfrow=c(2,1))
model <- Arima(log_data$Petrobras, order = c(0,0,1), xreg = log_data$`Brent Oil Prices`)
ts.plot(log_data$Petrobras, fitted(model), col=c("red", "green4"), ylab = "Real x Model Petr")
ts.plot(residuals(model), col="blue", ylab="Residuals")
```

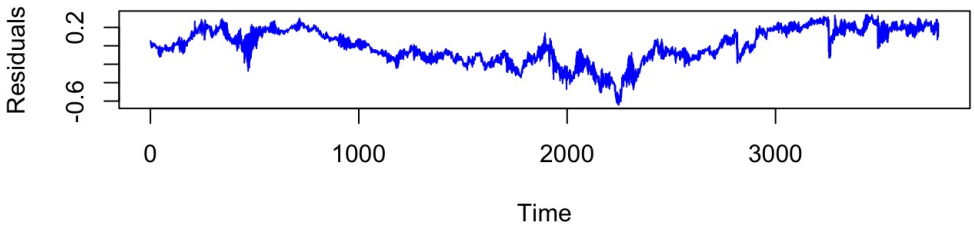




The model with 2 moving averages components seems to fit better

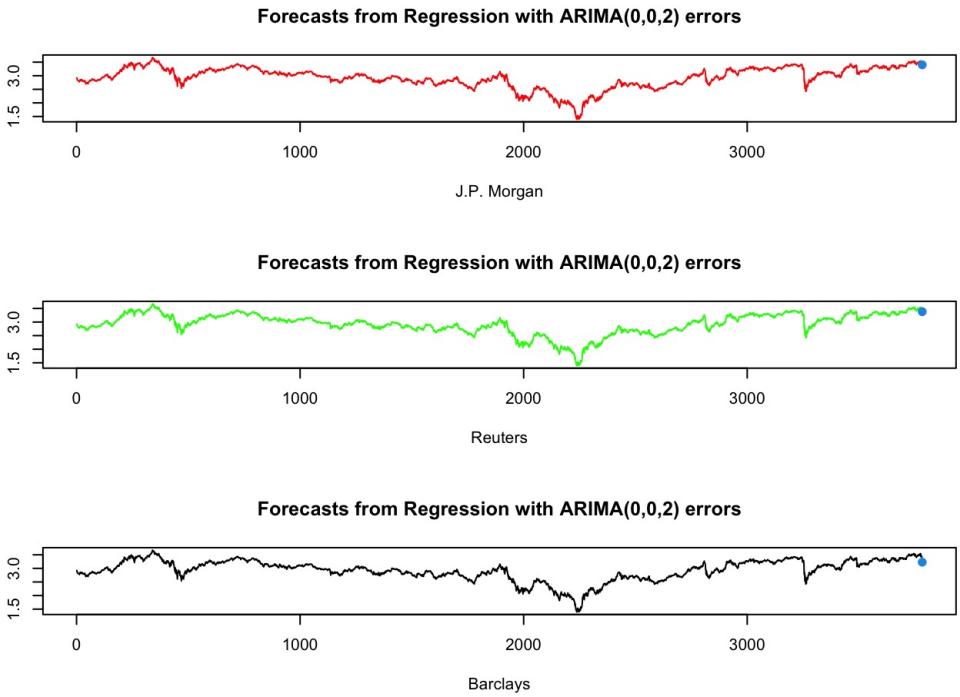
```
par(mfrow=c(2,1))
model_arimax <- Arima(log_data$Petrobras, order = c(0,0,2), xreg = log_data$`Brent Oil Prices`)
ts.plot(log_data$Petrobras, fitted(model_arimax), col=c("red", "green4"), ylab = "Real x Model Petr")
ts.plot(residuals(model), col="blue", ylab="Residuals")
```





For the forecast produce, we gonna get the J.P Morgan, Reuters and Barclays forecasts for Brent Oil prices for 2022. The information are avaible at the following links:

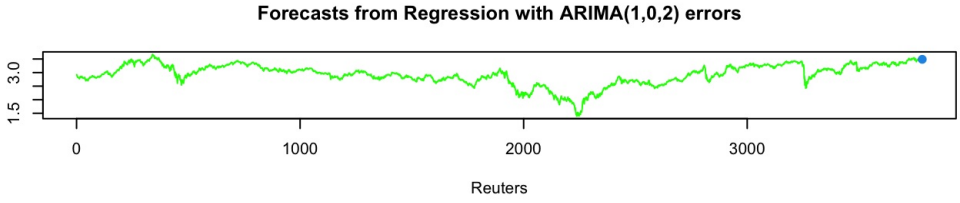
- J.P Morgan: 'https://www.thestreet.com/investing/oil-will-hit-125-a-barrel-in-2022-150-in-2023-jpmorgan (https://www.thestreet.com/investing/oil-will-hit-125-a-barrel-in-2022-150-in-2023-jpmorgan)'
- Reuters: 'https://www.reuters.com/business/energy/oil-prices-drop-demand-worries-rising-supplies-2021-11-16/ (https://www.reuters.com/business/energy/oil-prices-drop-demand-worries-rising-supplies-2021-11-16/)'
- Barclays: 'https://boereport.com/2022/01/24/barclays-hikes-2022-oil-price-view-by-5-bbl-on-depressed-inventories/ (https://boereport.com/2022/01/24/barclays-hikes-2022-oil-price-view-by-5-bbl-on-depressed-inventories/)'



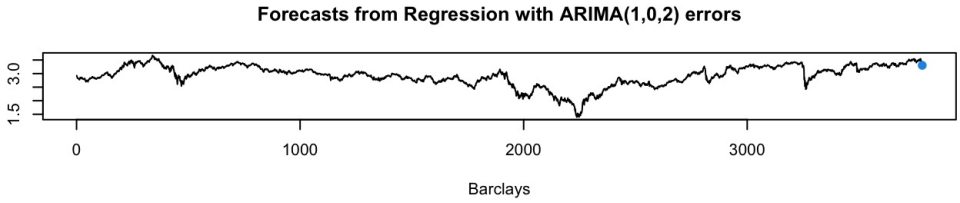
Since Brazil is going throught an election year, we decide to add a auto-regressive component, since the PETR4 stock price can pass to depend to previous values.

```
par(mfrow=c(3,1))
plot(jpm_forecast_elections, col = c("red"), xlab = "J.P. Morgan")
plot(reuters_forecast_elections, col = c("green"), xlab = 'Reuters')
plot(barclays_forecast_elections, xlab = "Barclays")
```

J.P. Morgan



Reuters



Barclays