

PETR4

Luan Rezende Eduardo

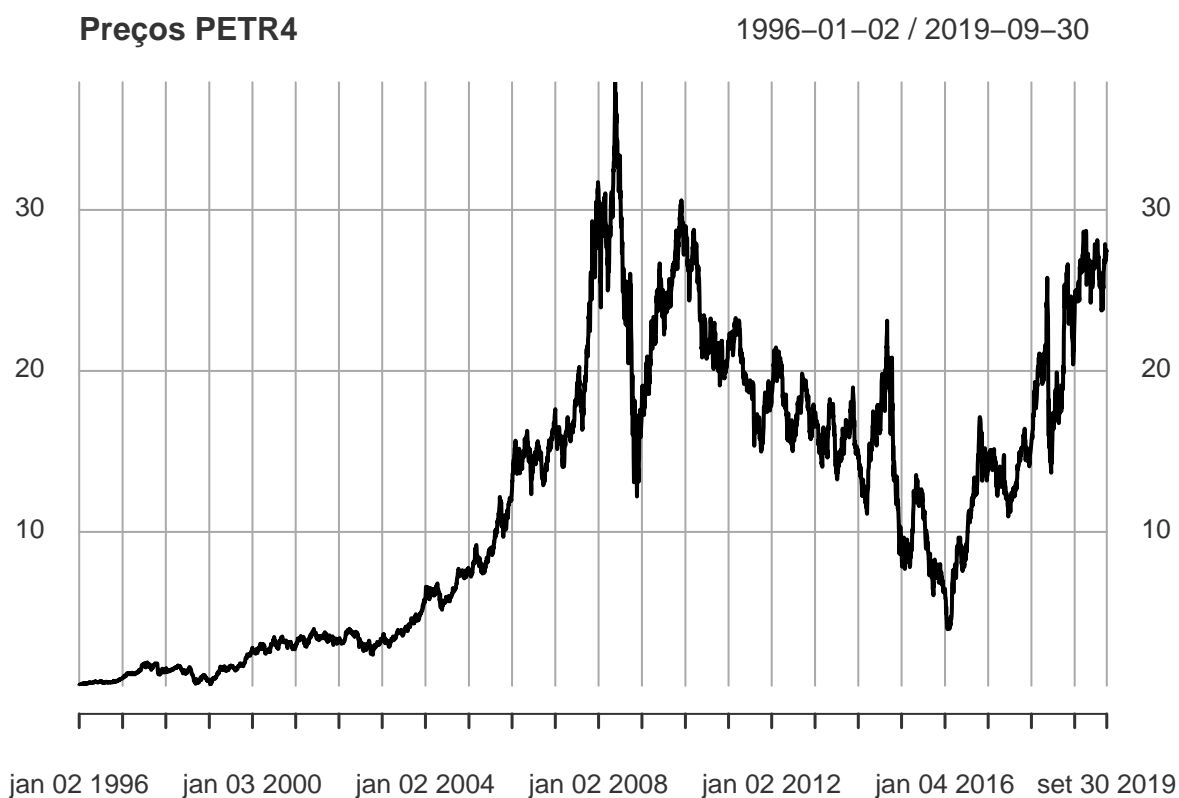
Introdução

Visa-se com este trabalho analisar os dados da série de retornos aproximada obtida a partir do log da diferença da série de preços do ativo financeiro PETR4, modelando a sua média condicional através de um modelo ARIMA e sua volatilidade condicional através de um modelo GARCH, de forma a obter predições sobre o seu comportamento fora da amostra.

A sigla PETR4 se refere as ações preferenciais da estatal brasileira Petrobras S.A. e os códigos utilizados podem ser encontradas no link https://github.com/luanreduardo/econometrics_stocks.

Análise Exploratória

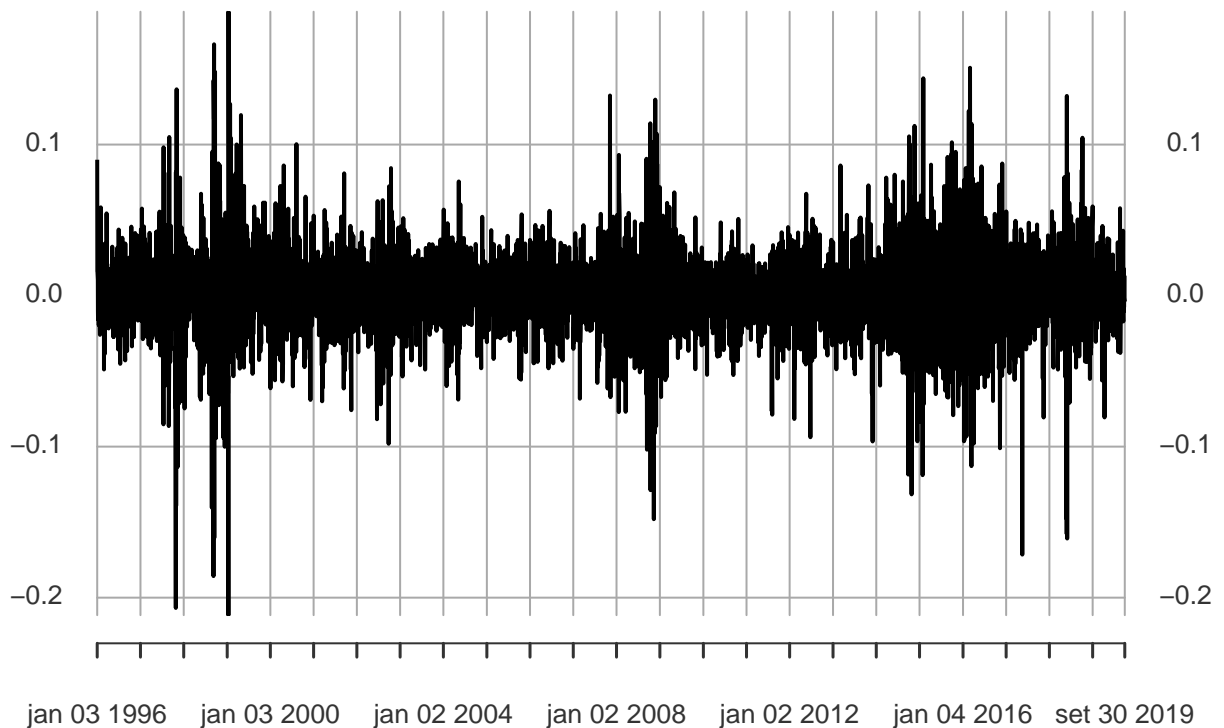
Utiliza-se a base de dados da Economática contendo os preços de fechamento do ativo PETR4 entre os anos de 1996 e 2019, convertidos em uma série temporal do tipo xts. Após a remoção dos dias de não-atividade do mercado tem-se os preços conforme o seguinte gráfico.



Posteriormente, aplica-se a diferença do logaritmo para obter uma série aproximada de retornos para este ativo, com o objetivo de obter uma série estacionária, uma vez que esta é condição necessária para a modelagem. Obtém-se, então, a série abaixo.

Retornos PETR4

1996-01-03 / 2019-09-30

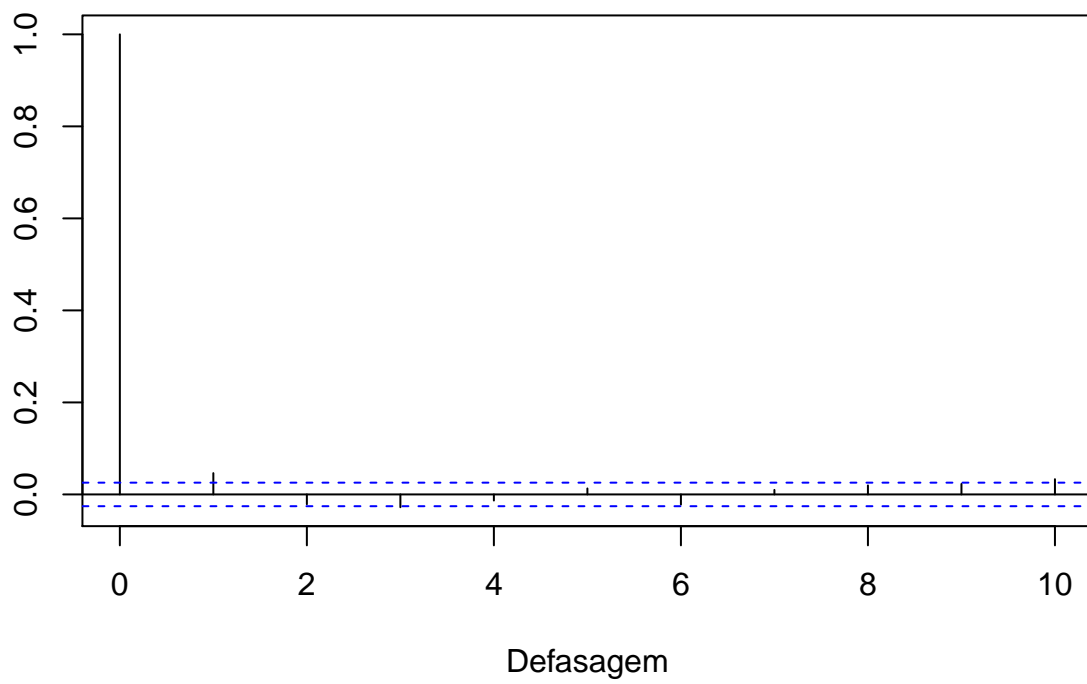


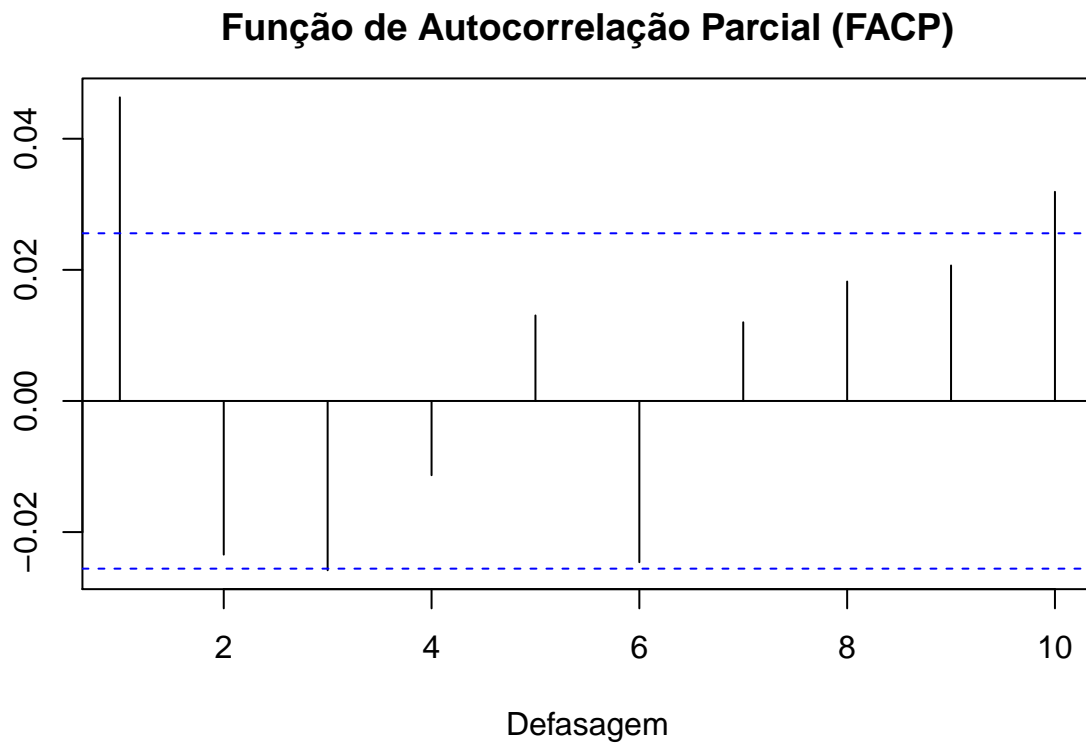
A partir dos dados aparentemente estacionários da série de retornos, testamos a estacionariedade através de um teste de Dickey-Fuller aumentado, com o número de defasagens sugerido pela função como a raiz cúbica do tamanho da amostra menos 1. Nesse teste a H_0 aponta para um passeio aleatório, o presente resultado rejeita H_0 indicando a estacionariedade da série.

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: petr4_returns  
## Dickey-Fuller = -16.601, Lag order = 18, p-value = 0.01  
## alternative hypothesis: stationary
```

Por último, calcula-se as funções de autocorrelação e autocorrelação parcial para se verificar se os seus decaimentos indicam a estacionariedade, os resultados abaixo tem rápido decaimento e corroboram o indicado no teste de Dickey-Fuller.

Função de Autocorrelação (FAC)





Modelagem do ARIMA

Para determinar as ordens máximas p , d e q utiliza-se os resultados obtidos na análise exploratória. Como a série dos retornos, utilizada a partir daqui, já apresenta estacionariedade não há necessidade de diferenciação adicional, logo $d = 0$. Já os gráficos da FAC e da FACP indicam, respectivamente, um MA(1) e um AR(1) máximos, dessa forma testaremos todos os modelos entre ARIMA(0,0,0) e ARIMA(1,0,1).

##	spec	log_likelihood	qt_parameters	size	AIC	BIC
## 1	ARIMA000	12733.20	2	5872	-25462.39	-25449.04
## 2	ARIMA100	12739.51	3	5872	-25473.02	-25452.98
## 3	ARIMA001	12739.79	3	5872	-25473.57	-25453.54
## 4	ARIMA101	12740.32	4	5872	-25472.64	-25445.93

Os resultados indicam que modelo de menor AIC e BIC é o ARIMA(0,0,1), ou MA(1), com três parâmetros. Esse, portanto, é o escolhido para os próximos passos.

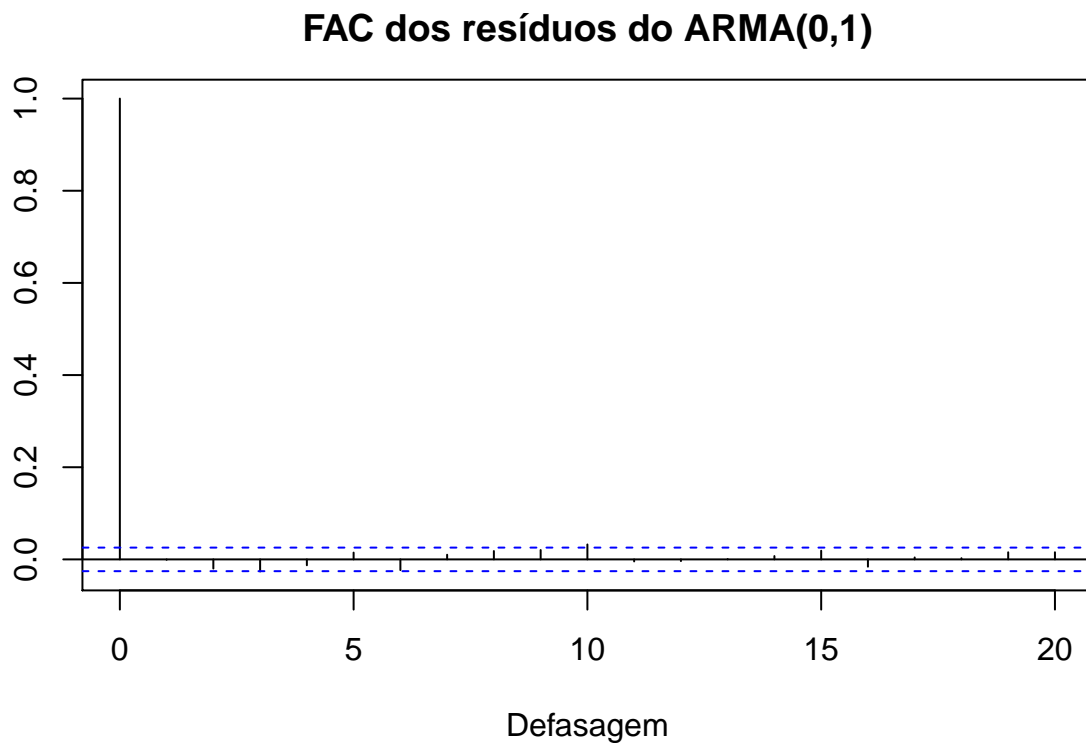
```
##
## Call:
## arima(x = petr4_returns, order = c(0, 0, 1), method = "ML")
##
## Coefficients:
##          ma1  intercept
##          0.0484      7e-04
## s.e.    0.0133      4e-04
##
## sigma^2 estimated as 0.0007639:  log likelihood = 12739.79,  aic = -25473.57
```

Análise dos Resíduos para o MA(1)

Aqui é feita uma análise dos resíduos para verificar se existem indicativos de padrões não esperados, uma vez que esses devem se comportar como um ruído branco. Para isso, inicia-se com um teste de Ljung-Box para testar se há indicativo de autocorrelação serial dos resíduos, onde o objetivo é aceitar a hipótese de que os resíduos são i.i.d, H_0 . Tem-se um p-valor < 0.05 , evidência contra H_0 .

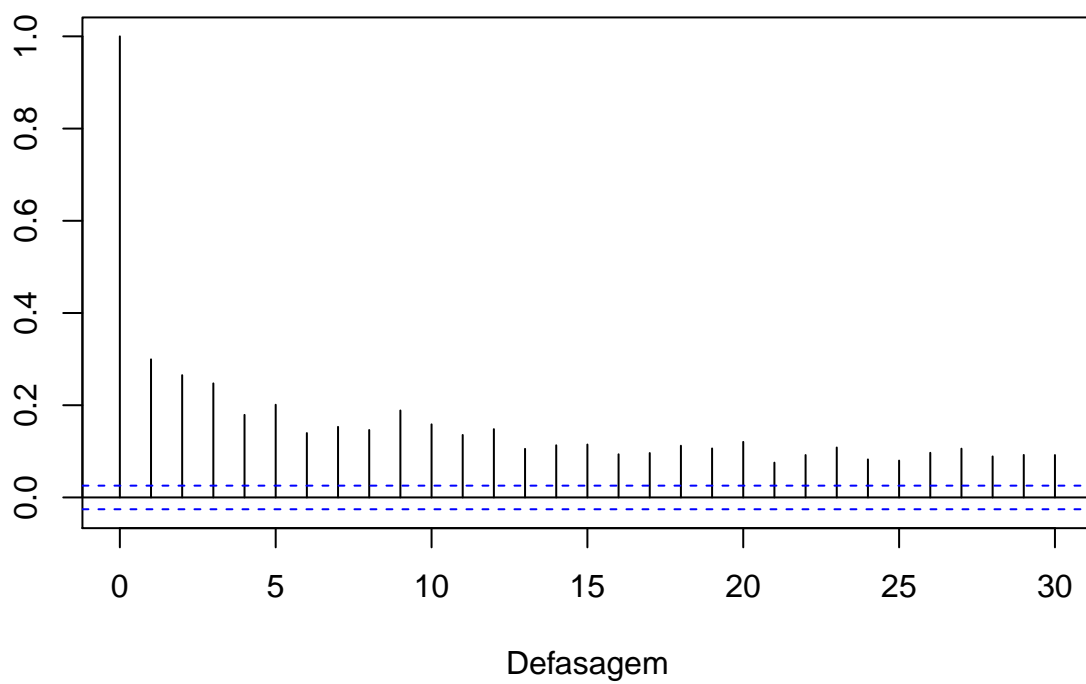
```
##  
## Box-Ljung test  
##  
## data: cond_average$residuals  
## X-squared = 85.189, df = 63, p-value = 0.03281
```

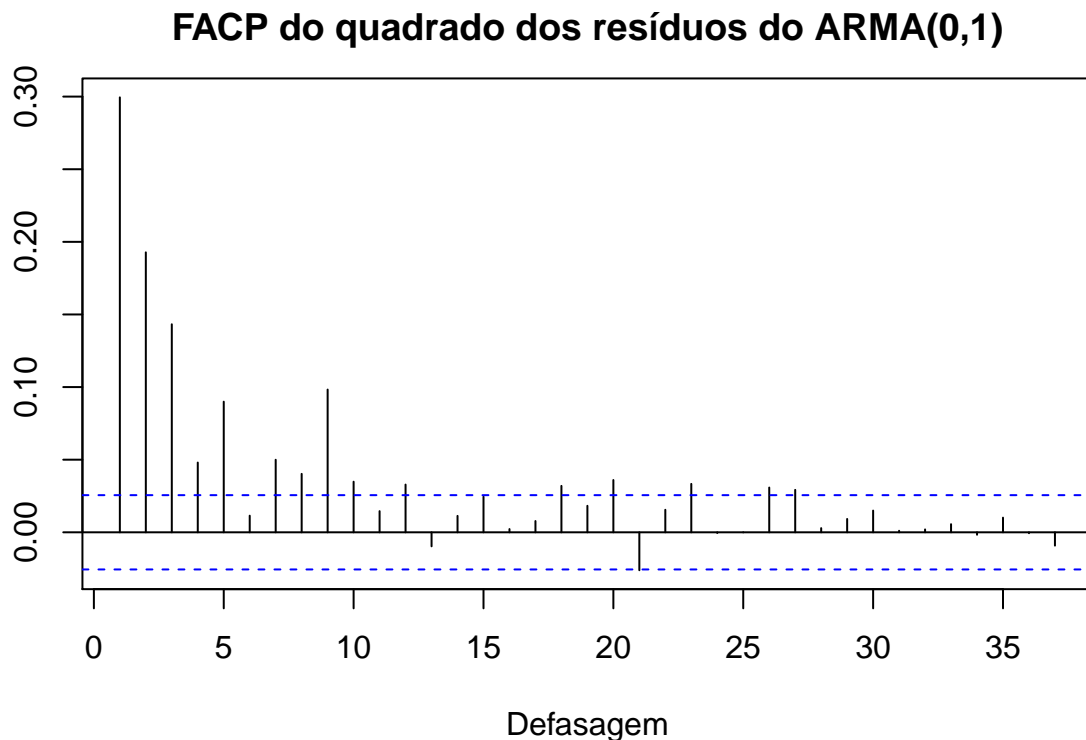
Utiliza-se a FAC dos resíduos, indica a ausência de uma autocorrelação.



Já a FAC dos resíduos ao quadrado indica a existência de uma heterocedasticidade condicional nos dados, o que corrobora um fato estilizado do mercado financeiro, que apresenta clusters de volatilidade e onde o módulo de $t - 1$ afeta o valor de t .

FAC do quadrado dos resíduos do ARMA(0,1)





A não realização dos testes de Shapiro e Jarque-Bera para normalidade da distribuição dos resíduos é justificado pela pouca confiabilidade de seus resultados para grande amostras, uma vez que pequenas distorções na distribuição já causam a rejeição da hipótese de normalidade adotada para o modelo.

Modelo ARMA-GARCH

A presença de heterocedasticidade condicional evidenciada pelas funções de autocorrelação e autorrelação parcial torna necessária a adição de uma ferramenta de modelagem da variância condicional da série temporal. Portanto, busca-se a partir daqui modelar um ARIMA-GARCH para os dados.

As funções de FAC e FACP dos resíduos ao quadrados indicam as ordens m e n máximas do modelo de variância condicional GARCH, uma vez que o FAC não entram na banda de aceitação em defasagens muito altas de cientes de que altas ordens não são usuais, opta-se por $m = 5$ e $n = 5$ conforme FACP. Os resultados de cada composição de ordens segue abaixo, a distribuição adotada foi a t assimétrica.

##	spec	ln_likelihood	param_number	sample_size	aic
## 1	~arma(0,1)-garch10	-13357.82	5	5872	-4.547963
## 2	~arma(0,1)-garch20	-13469.02	6	5872	-4.585498
## 3	~arma(0,1)-garch30	-13525.34	7	5872	-4.604339
## 4	~arma(0,1)-garch40	-13557.91	8	5872	-4.615092
## 5	~arma(0,1)-garch50	-13582.15	9	5872	-4.623008
## 6	~arma(0,1)-garch11	-13663.62	6	5872	-4.651779
## 7	~arma(0,1)-garch21	-13663.58	7	5872	-4.651424
## 8	~arma(0,1)-garch31	-13663.53	8	5872	-4.651067
## 9	~arma(0,1)-garch41	-13663.50	9	5872	-4.650714
## 10	~arma(0,1)-garch51	-13663.48	10	5872	-4.650367

## 11	~arma(0,1)-garch12	-13663.55	7	5872	-4.651412
## 12	~arma(0,1)-garch22	-13663.97	8	5872	-4.651215
## 13	~arma(0,1)-garch32	-13663.91	9	5872	-4.650855
## 14	~arma(0,1)-garch42	-13663.87	10	5872	-4.650499
## 15	~arma(0,1)-garch52	-13663.84	11	5872	-4.650149
## 16	~arma(0,1)-garch13	-13663.78	8	5872	-4.651152
## 17	~arma(0,1)-garch23	-13665.34	9	5872	-4.651341
## 18	~arma(0,1)-garch33	-13665.34	10	5872	-4.651000
## 19	~arma(0,1)-garch43	-13665.29	11	5872	-4.650643
## 20	~arma(0,1)-garch53	-13665.25	12	5872	-4.650290
## 21	~arma(0,1)-garch14	-13664.50	9	5872	-4.651057
## 22	~arma(0,1)-garch24	-13666.43	10	5872	-4.651374
## 23	~arma(0,1)-garch34	-13666.85	11	5872	-4.651175
## 24	~arma(0,1)-garch44	-13667.54	12	5872	-4.651071
## 25	~arma(0,1)-garch54	-13667.43	13	5872	-4.650691
## 26	~arma(0,1)-garch15	-13664.95	10	5872	-4.650868
## 27	~arma(0,1)-garch25	-13666.63	11	5872	-4.651101
## 28	~arma(0,1)-garch35	-13666.76	12	5872	-4.650805
## 29	~arma(0,1)-garch45	-13667.45	13	5872	-4.650699
## 30	~arma(0,1)-garch55	-13666.47	14	5872	-4.650026
##	bic				
## 1	-4.542276				
## 2	-4.578674				
## 3	-4.596379				
## 4	-4.605994				
## 5	-4.612773				
## 6	-4.644955				
## 7	-4.643463				
## 8	-4.641969				
## 9	-4.640479				
## 10	-4.638994				
## 11	-4.643452				
## 12	-4.642117				
## 13	-4.640619				
## 14	-4.639127				
## 15	-4.637639				
## 16	-4.642054				
## 17	-4.641106				
## 18	-4.639628				
## 19	-4.638134				
## 20	-4.636643				
## 21	-4.640822				
## 22	-4.640002				
## 23	-4.638665				
## 24	-4.637424				
## 25	-4.635907				
## 26	-4.639495				
## 27	-4.638591				
## 28	-4.637158				
## 29	-4.635914				
## 30	-4.634104				

Os resultados de obtidos indicam menores AIC e BIC no modelo ARMA(0,1)-GARCH(1,1), definindo o modelo utilizado no restante desse trabalho.


```

##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : sGARCH(1,1)
## Mean Model    : ARFIMA(0,0,1)
## Distribution   : sstd
##
## Optimal Parameters
## -----
##      Estimate   Std. Error   t value   Pr(>|t|)
## mu      0.001196    0.000293    4.0845  0.000044
## ma1     0.043826    0.013606    3.2212  0.001277
## omega    0.000015    0.000003    5.0656  0.000000
## alpha1   0.095911    0.010185    9.4167  0.000000
## beta1    0.884728    0.008183   108.1185 0.000000
## skew     0.980684    0.018097   54.1898 0.000000
## shape    7.449034    0.717432   10.3829 0.000000
##
## Robust Standard Errors:
##      Estimate   Std. Error   t value   Pr(>|t|)
## mu      0.001196    0.000301    3.9729  0.000071
## ma1     0.043826    0.013858    3.1624  0.001565
## omega    0.000015    0.000005    3.0661  0.002169
## alpha1   0.095911    0.016591    5.7810  0.000000
## beta1    0.884728    0.007175   123.3108 0.000000
## skew     0.980684    0.018583   52.7731 0.000000
## shape    7.449034    0.889329    8.3760  0.000000
##
## LogLikelihood : 13658.74
##
## Information Criteria
## -----
##
## Akaike          -4.6498
## Bayes           -4.6418
## Shibata         -4.6498
## Hannan-Quinn   -4.6470
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##                                statistic p-value
## Lag[1]                                1.390  0.2385
## Lag[2*(p+q)+(p+q)-1] [2]          1.605  0.3810
## Lag[4*(p+q)+(p+q)-1] [5]          2.353  0.6087
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##                                statistic p-value

```

```

## Lag[1] 0.01029 0.91922
## Lag[2*(p+q)+(p+q)-1][5] 9.08645 0.01574
## Lag[4*(p+q)+(p+q)-1][9] 11.42840 0.02441
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##          Statistic Shape Scale P-Value
## ARCH Lag[3] 0.7213 0.500 2.000 0.3957
## ARCH Lag[5] 1.2422 1.440 1.667 0.6627
## ARCH Lag[7] 1.8402 2.315 1.543 0.7511
##
## Nyblom stability test
## -----
## Joint Statistic: 22.7542
## Individual Statistics:
## mu 0.57547
## ma1 3.81021
## omega 2.46960
## alpha1 0.24960
## beta1 0.25663
## skew 0.15682
## shape 0.09256
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic: 1.69 1.9 2.35
## Individual Statistic: 0.35 0.47 0.75
##
## Sign Bias Test
## -----
##          t-value      prob sig
## Sign Bias 0.06651 0.9469752
## Negative Sign Bias 2.30475 0.0212151 **
## Positive Sign Bias 1.66932 0.0951075 *
## Joint Effect 16.46468 0.0009105 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
## group statistic p-value(g-1)
## 1 20 13.14 0.8315
## 2 30 21.26 0.8494
## 3 40 28.11 0.9021
## 4 50 33.94 0.9498
##
##
## Elapsed time : 1.497141

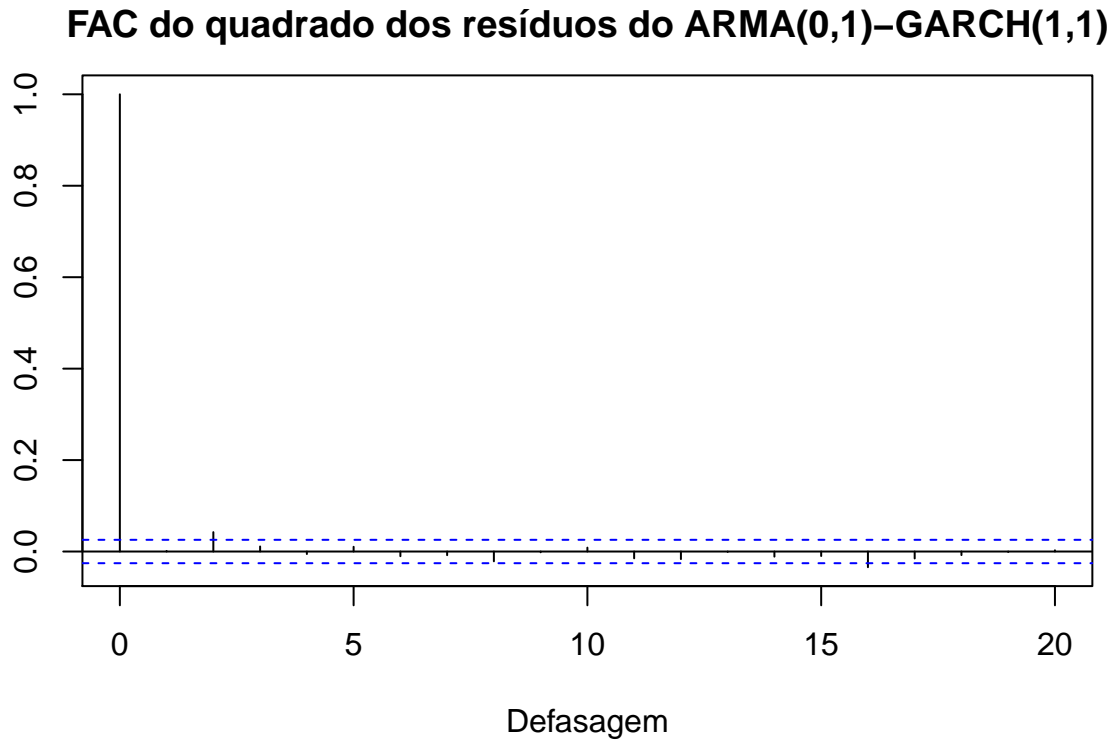
```

Análise dos Resíduos MA(1)-GARCH(1,1)

Uma análise similar a feita acima é realizada para testar a presença de autocorrelação serial ou heterocedasticidade condicional dos resíduos, agora no novo modelo ARMA-GARCH. Inicia-se com os resultados do teste de Dickey-Fuller aumentado, que sugere aceitar que os resíduos são i.i.d, ou seja, H_0 .

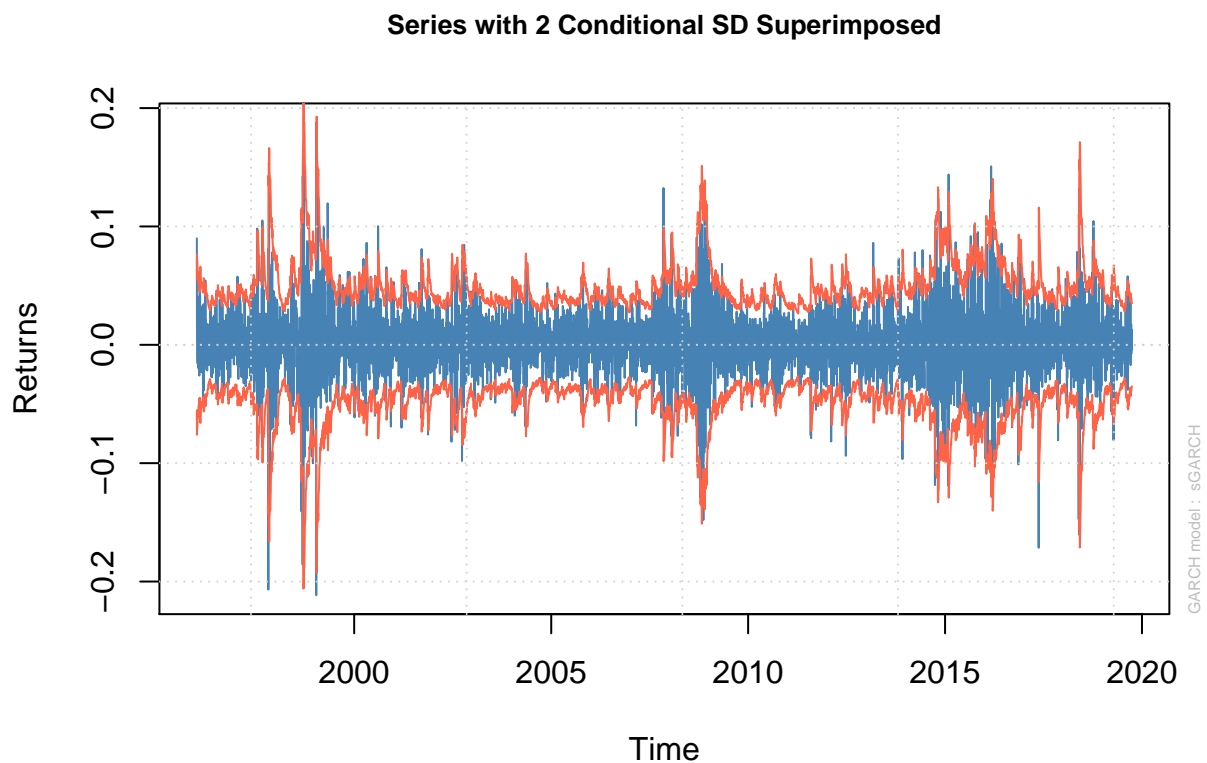
```
##  
## Box-Ljung test  
##  
## data: fGarch::residuals(chosen_arma_garch, standardize = TRUE)  
## X-squared = 60.811, df = 63, p-value = 0.5548
```

Abaixo, o gráfico da função de autocorrelação dos resíduos padronizados ao quadrado também sugere a ausência de heterocedasticidade.



O modelo proposto parece capturar os padrões dos dados e a análise dos resíduos mostra que eles apresentam um bom comportamento

O resultado do ARMA(0,1)-GARCH(1,1) é expresso no gráfico abaixo



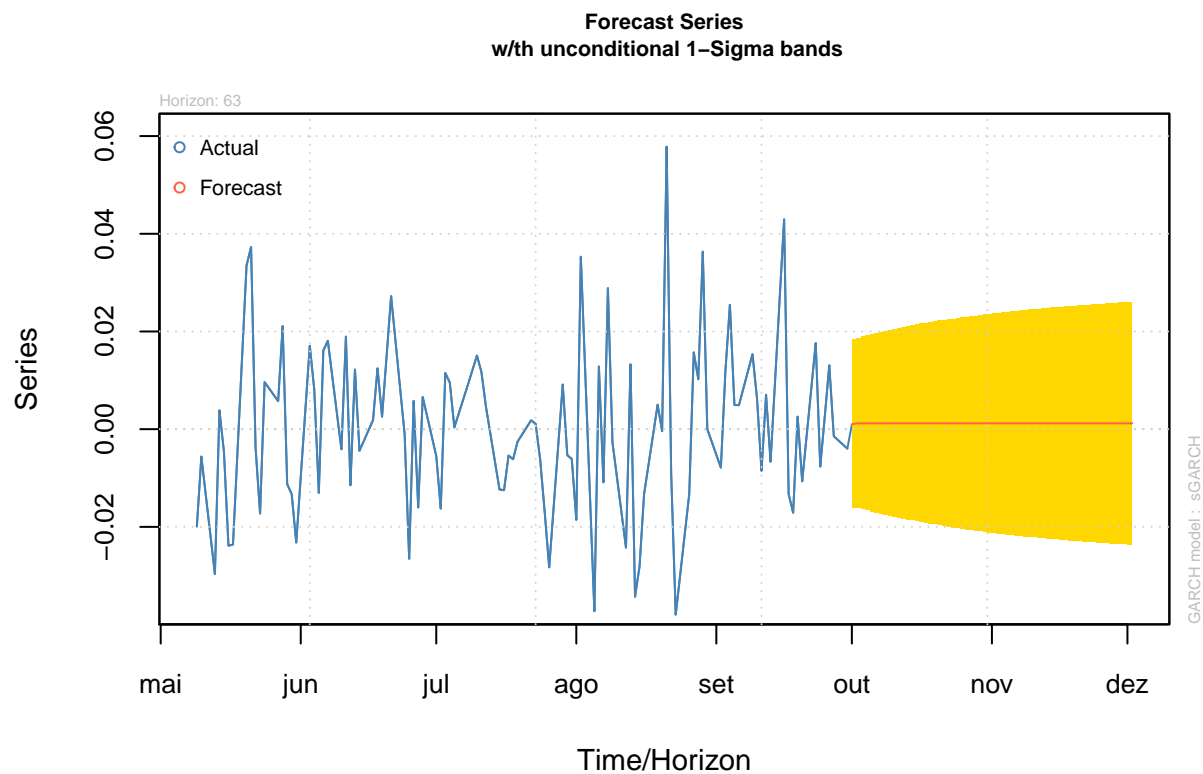
Predições e acurácia

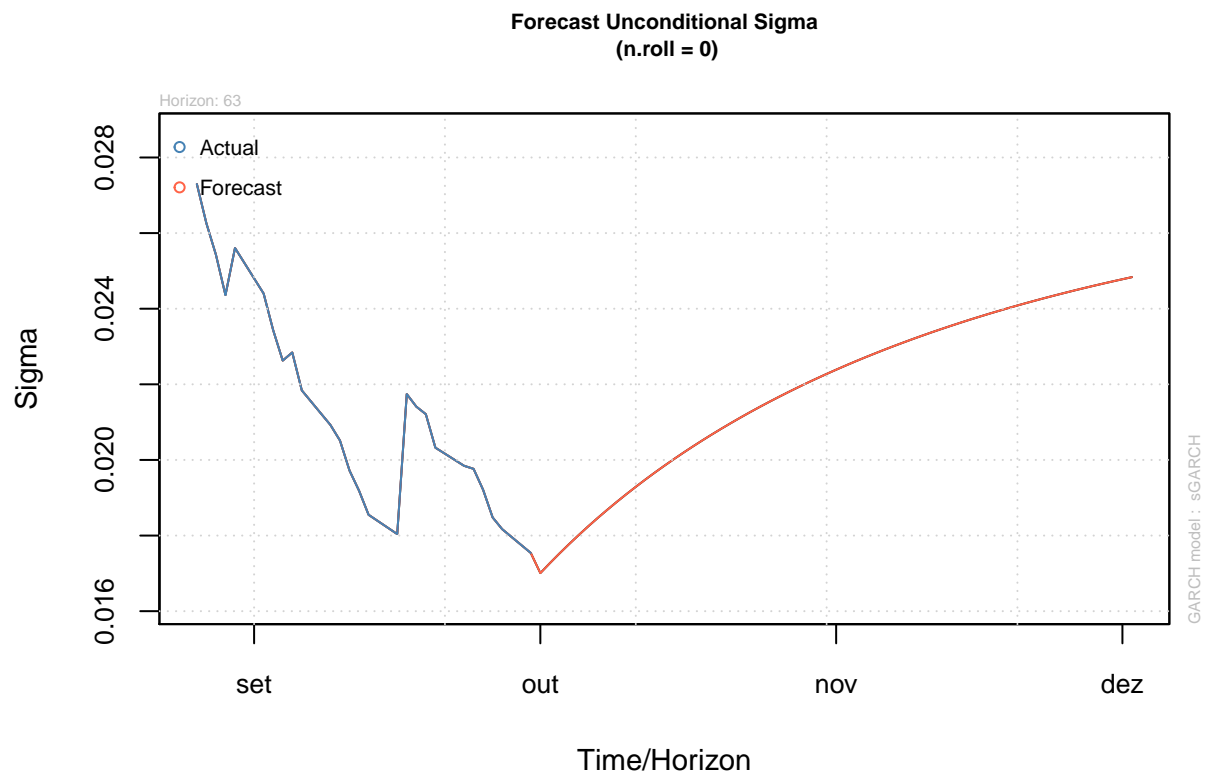
Por fim, uma previsão de 63 períodos é calculada com os seguintes valores de média e variância para cada período $T + i$ a partir de 30 de setembro de 2019.

```
##
## *-----*
## *      GARCH Model Forecast      *
## *-----*
## Model: sGARCH
## Horizon: 63
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=2019-09-30]:
##      Series  Sigma
## T+1  0.0009746 0.01701
## T+2  0.0011955 0.01727
## T+3  0.0011955 0.01753
## T+4  0.0011955 0.01777
## T+5  0.0011955 0.01801
## T+6  0.0011955 0.01824
## T+7  0.0011955 0.01846
## T+8  0.0011955 0.01868
## T+9  0.0011955 0.01889
```

T+10 0.0011955 0.01909
T+11 0.0011955 0.01929
T+12 0.0011955 0.01948
T+13 0.0011955 0.01967
T+14 0.0011955 0.01985
T+15 0.0011955 0.02002
T+16 0.0011955 0.02019
T+17 0.0011955 0.02036
T+18 0.0011955 0.02052
T+19 0.0011955 0.02068
T+20 0.0011955 0.02083
T+21 0.0011955 0.02098
T+22 0.0011955 0.02112
T+23 0.0011955 0.02127
T+24 0.0011955 0.02140
T+25 0.0011955 0.02154
T+26 0.0011955 0.02167
T+27 0.0011955 0.02180
T+28 0.0011955 0.02192
T+29 0.0011955 0.02204
T+30 0.0011955 0.02216
T+31 0.0011955 0.02227
T+32 0.0011955 0.02239
T+33 0.0011955 0.02250
T+34 0.0011955 0.02260
T+35 0.0011955 0.02271
T+36 0.0011955 0.02281
T+37 0.0011955 0.02291
T+38 0.0011955 0.02301
T+39 0.0011955 0.02310
T+40 0.0011955 0.02319
T+41 0.0011955 0.02329
T+42 0.0011955 0.02337
T+43 0.0011955 0.02346
T+44 0.0011955 0.02354
T+45 0.0011955 0.02363
T+46 0.0011955 0.02371
T+47 0.0011955 0.02379
T+48 0.0011955 0.02386
T+49 0.0011955 0.02394
T+50 0.0011955 0.02401
T+51 0.0011955 0.02409
T+52 0.0011955 0.02416
T+53 0.0011955 0.02422
T+54 0.0011955 0.02429
T+55 0.0011955 0.02436
T+56 0.0011955 0.02442
T+57 0.0011955 0.02448
T+58 0.0011955 0.02455
T+59 0.0011955 0.02461
T+60 0.0011955 0.02467
T+61 0.0011955 0.02472
T+62 0.0011955 0.02478
T+63 0.0011955 0.02484

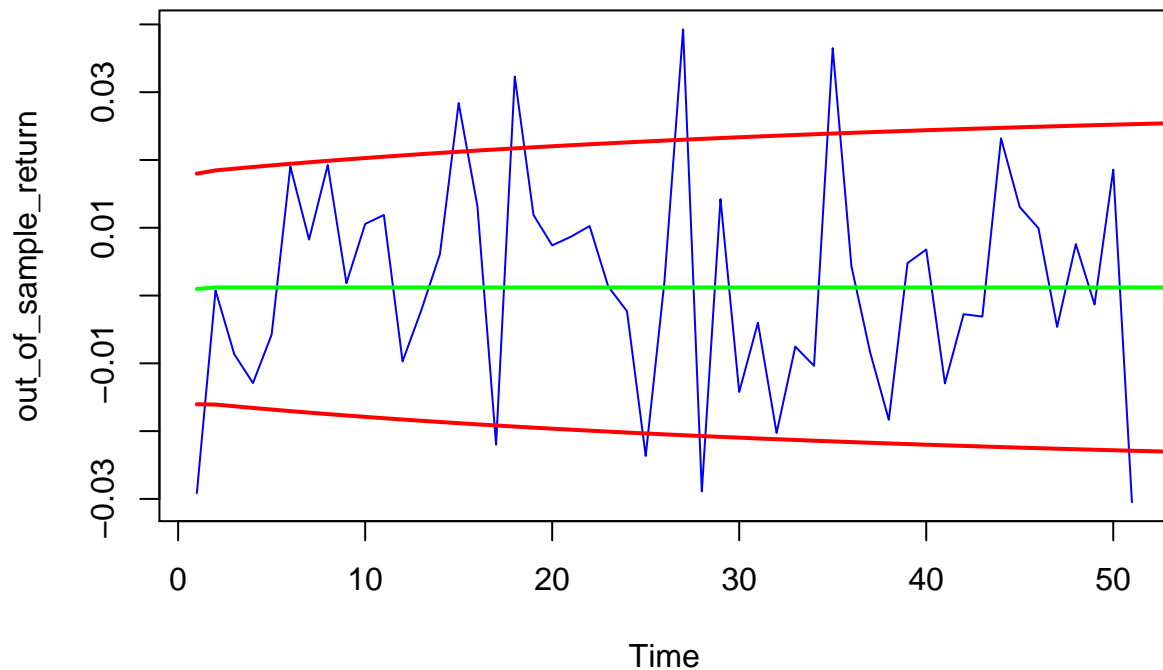
A partir do qual plotam-se os gráficos contendo os dados in-sample desde maio até o final de setembro, com a previsão dentro da banda de 1 sigma e também a previsão para o sigma incondicional, respectivamente.





Um gráfico com os as previsões de média (em verde) e uma banda de dois sigma (média + sigma, média - sigma; em vermelho) juntamente com os dados de retornos dos meses de outubro, novembro e dezembro é plotado.

63 períodos fora da amostra



Conclui-se com o teste de acurácia, onde a média da série de erros (ME) se apresenta baixa, assim como a média das raízes dos erros quadráticos (RMSE) e do valor absoluto do erro (MAE) também apresentam valores baixos. A média da série de erros percentual (MPE) e de erro absoluto percentual (MAPE) tem valores próximos ao 100%, indicando que o modelo tem pouco poder preditivo. Este resultado condiz com os fatos estilizados do mercado onde qualquer padrão tende a se dissipar pela ação de arbitragem dos agentes.

```
##               ME      RMSE      MAE      MPE      MAPE
## Test set 0.0005338995 0.01607075 0.01273942 96.87605 99.22019
```

```
#packages
library(readxl)
library(xts)
library(zoo)
library(PerformanceAnalytics)

#loading data into a dataframe
closing_prices <- readxl::read_xlsx('data/acoesBR.xlsx', na = "-", )

#creating the index
date_string <- as.Date(closing_prices$Data)

#creating the time series
petr4_xts <- xts::xts(closing_prices[, 393], order.by = date_string)
```



```

colnames(petr4_xts) <- "PETR4"

#excluding NAs
petr4_xts <- na.omit(petr4_xts)

#sub-setting to 1996 up to september/2019
petr4_xts <- petr4_xts['1996/2019-09']

#calculating log returns
petr4_returns <- PerformanceAnalytics::Return.calculate(petr4_xts, method = "log")
petr4_returns <- na.omit(petr4_returns)

saveRDS(petr4_xts, file = "data/petr4.rds")
saveRDS(petr4_returns, file = "data/p4_returns.rds")

#####

#exploratory analysis

library(xts)
library(fUnitRoots)
library(stats)
library(tseries)

#reading data
petr4_xts <- readRDS('data/petr4.rds')
petr4_returns <- readRDS('data/p4_returns.rds')

#plotting the timeseries of prices and returns
plot.xts(petr4_xts$PETR4, main = "Preços PETR4")
plot.xts(petr4_returns, main = "Retornos PETR4")

#testing stationarity, H0 points to random walk
unit_root <- tseries::adf.test(petr4_returns)

#auto-correlation and partial ac functions
acf_petr4_ret <- stats::acf(petr4_returns, na.action = na.pass, plot = FALSE, lag.max = 10)
pacf_petr4_ret <- stats::pacf(petr4_returns, na.action = na.pass, plot = FALSE, lag.max = 10)

par(mfrow = c(2,1))
plot(acf_petr4_ret, main = "", ylab = "", xlab = "Defasagem")
title("Função de Autocorrelação (FAC)", adj = 0.5, line = 1)
plot(pacf_petr4_ret, main = "", ylab = "", xlab = "Defasagem")
title("Função de Autocorrelação Parcial (FACP)", adj = 0.5, line = 1)
par(mfrow = c(1,1))

#####

#ARIMA modelling

library(stats)

```

```

#loading data
petr4_returns <- readRDS('data/p4_returns.rds')

#possibilities equal to or smaller than the last lag of ACF/PACF
feasible_set <- expand.grid(ar = 0:1, diff = 0, ma = 0:1)

#defining a list for the results
model <- list()

#estimating parameters through maximum likelihood
for (i in 1:nrow(feasible_set)) {
  model[[i]] <- stats::arima(petr4_returns, order = unlist(feasible_set[i, 1:3]), method = "ML")
}

#greater value through log of ML
log_likelihood <- list()
for (i in 1:length(model)) {
  log_likelihood[[i]] <- model[[i]]$loglik
}

#AIC
aicarma <- list()
for (i in 1:length(model)) {
  aicarma[[i]] <- stats::AIC(model[[i]])
}

#BIC
bicarma <- list()
for (i in 1:length(model)) {
  bicarma[[i]] <- stats::BIC(model[[i]])
}

#number of parameters
qt_parameters <- list()
for (i in 1:length(model)) {
  qt_parameters[[i]] <- length(model[[i]]$coef) + 1 # + error's variance
}

#table of results
spec <- paste0("ARIMA",feasible_set$ar, feasible_set$diff, feasible_set$ma)
size <- rep(length(petr4_returns), length(model))
results <- data.frame(spec, log_likelihood = unlist(log_likelihood),
                      qt_parameters = unlist(qt_parameters),
                      size, AIC = unlist(aicarma),
                      BIC = unlist(bicarma), stringsAsFactors = FALSE)

print(results)

#chosen arima model
cond_average <- arima(petr4_returns, order = c(0,0,1), method = "ML")

#####

```

```

#residuals analysis

library(stats)
library(tseries)
library(lmtest)

#autocorrelation function for the residuals
acf_residuals <- stats::acf(cond_average$residuals, na.action = na.pass, plot = FALSE, lag.max = 20)
plot(acf_residuals, main = "", ylab = "", xlab = "Defasagem")
title("FAC dos resíduos do ARMA(0,1)", adj = 0.5, line = 1)

#Ljung-Box
box_test_arma <- stats::Box.test(cond_average$residuals, lag = 63, type = "Ljung-Box") #check lags
print(box_test_arma)

#autocorrelation function, and partial acf, for the squared residuals
acf_residuals_square <- acf(cond_average$residuals^2,
                             na.action = na.pass, plot = FALSE, lag.max = 30)
plot(acf_residuals_square, main = "", ylab = "", xlab = "Defasagem")
title("FAC do quadrado dos resíduos do ARMA(0,1)", adj = 0.5, line = 1)
pacf_residuals_square <- stats::pacf(cond_average$residuals^2,
                                     plot = FALSE, na.action = na.pass, max.lag = 30)
plot(pacf_residuals_square, main = "", ylab = "", xlab = "Defasagem")
title("FACP do quadrado dos resíduos do ARMA(0,1)", adj = 0.5, line = 1)

#tests for the distribution of the residuals, H0: follows a normal dist.
#shapiro_test <- stats::shapiro.test(na.remove(cond_average$residuals))
#jarque_bera <- tseries::jarque.bera.test(na.remove(cond_average$residuals))
#print(jarque_bera)

#####

#Garch modelling

library(fGarch)
library(xts)
library(stats)
library(rugarch)

#all possible m, n for garch modelling
pars_arma_garch <- expand.grid(m = 1:5, n = 0:5)

#list for results
arma_garch_model <- list()

#arma specification defined on previous codes
arma_set <- "~arma(0,1)"

# probability distribution for the residuals (norm, std, snorm, sstd)
arma_residuals_dist <- "sstd"

#skew and assymetry definitions

```

```

include.skew = FALSE
include.shape = TRUE

#parameters estimation through maximum likelihood
for (i in 1:nrow(pars_arma_garch)) {
  arma_garch_model[[i]] <- fGarch::garchFit(as.formula(paste0(arma_set,"+", "garch(",pars_arma_garch[i,1],
  data = petr4_returns, trace = FALSE, cond.dist = arma_resid,
  include.skew = include.skew, include.shape = include.shape,
})

# function maximum value
log_likelihood_arma_garch <- list()
for (i in 1:length(arma_garch_model)) {
  log_likelihood_arma_garch[[i]] <- arma_garch_model[[i]]@fit$llh
}

# AIC
aicarma_garch <- list()
for (i in 1:length(arma_garch_model)) {
  aicarma_garch[[i]] <- arma_garch_model[[i]]@fit$ics[1]
}

# BIC
bicarma_garch <- list()
for (i in 1:length(arma_garch_model)) {
  bicarma_garch[[i]] <- arma_garch_model[[i]]@fit$ics[2]
}

# number of estimated parameters
arma_garch_param_number <- list()
for (i in 1:length(arma_garch_model)) {
  arma_garch_param_number[[i]] <- length(arma_garch_model[[i]]@fit$coef)
}

# table of results
spec <- paste0(arma_set,"-", "garch",pars_arma_garch$m,pars_arma_garch$n)
sample_size <- rep(length(petr4_returns), length(arma_garch_model))
results_arma_garch <- data.frame(spec, ln_likelihood = unlist(log_likelihood_arma_garch),
  param_number = unlist(arma_garch_param_number),
  sample_size, aic = unlist(aicarma_garch), bic = unlist(bicarma_garch),
  stringsAsFactors = FALSE, row.names = NULL)

print(results_arma_garch)

#using rugarch modelling package for it provides greater possibilities of analysis
ugarch_version_spec <- rugarch::ugarchspec(variance.model = list(garchOrder = c(1,1)),
  mean.model = list(armaOrder = c(0,1)), distribution.model = "sstd")
chosen_arma_garch <- rugarch::ugarchfit(spec = ugarch_version_spec, data = petr4_returns)
print(chosen_arma_garch)

#####

# ARMA-GARCH residual analysis

```

```

library(fGarch)
library(stats)

# squared residuals acf
acf_residuals_square_arma_garch <- acf(fGarch::residuals(chosen_arma_garch, standardize = TRUE)^2,
                                       na.action = na.pass, plot = FALSE, lag.max = 20)
plot(acf_residuals_square_arma_garch, main = "", ylab = "", xlab = "Defasagem")
title("FAC do quadrado dos resíduos do ARMA(0,1)-GARCH(1,1)", adj = 0.5, line = 1)

#Ljung-Box
box_test_garch <- stats::Box.test(fGarch::residuals(chosen_arma_garch, standardize = TRUE),
                                lag = 63, type = "Ljung-Box") #check lags
print(box_test_garch)

#####

#Results and accuracy

library(stats)
library(fGarch)
library(rugarch)
library(forecast)
library(quantmod)
library(PerformanceAnalytics)
library(xts)

#plotting selected arma-garch model
fGarch::plot(chosen_arma_garch, which = 'all')
fGarch::plot(chosen_arma_garch, which = 1)

#out-of-sample data from quantmod
out_of_sample_prices <- quantmod::getSymbols(Symbols = "PETR4.SA", src = "yahoo", from = '2019-10-01')
out_of_sample_return <- na.omit(PerformanceAnalytics::Return.calculate(PETR4.SA$PETR4.SA.Close, method = "log"))

#plotting out-of-sample return set
plot.xts(out_of_sample_return, main = "Dados fora da amostra - PETR4")

#forecasting 3 months out of sample
arma_garch_forecast <- rugarch::ugarchforecast(chosen_arma_garch, n.ahead = 63)
print(arma_garch_forecast)
fGarch::plot(arma_garch_forecast, which = 1)
fGarch::plot(arma_garch_forecast, which = 3)

#plotting forecast against out-of-sample data
mean_arma_garch <- arma_garch_forecast@forecast$seriesFor
upper_bound_garch <- arma_garch_forecast@forecast$seriesFor + arma_garch_forecast@forecast$sigmaFor
lower_bound_garch <- arma_garch_forecast@forecast$seriesFor - arma_garch_forecast@forecast$sigmaFor

ylim <- c(min(out_of_sample_return), max(out_of_sample_return))
plot.ts(out_of_sample_return, col = "blue", ylim = ylim, main = '63 períodos fora da amostra')
lines(mean_arma_garch, col = "green", ylim = ylim, lwd = 2)
lines(upper_bound_garch, col = "red", ylim = ylim, lwd = 2)

```

```
lines(lower_bound_garch, col = "red", ylim = ylim, lwd = 2)

#accuracy
accuracy_test_set <- forecast::accuracy(as.ts(arima_garch_forecast@forecast$seriesFor), out_of_sample_r
```

Códigos