METODOLOGIA DE BOX & JENKINGS

O objetivo da metodologia Box & Jenkins é determinar os três componentes dos modelos estruturais de séries temporais:

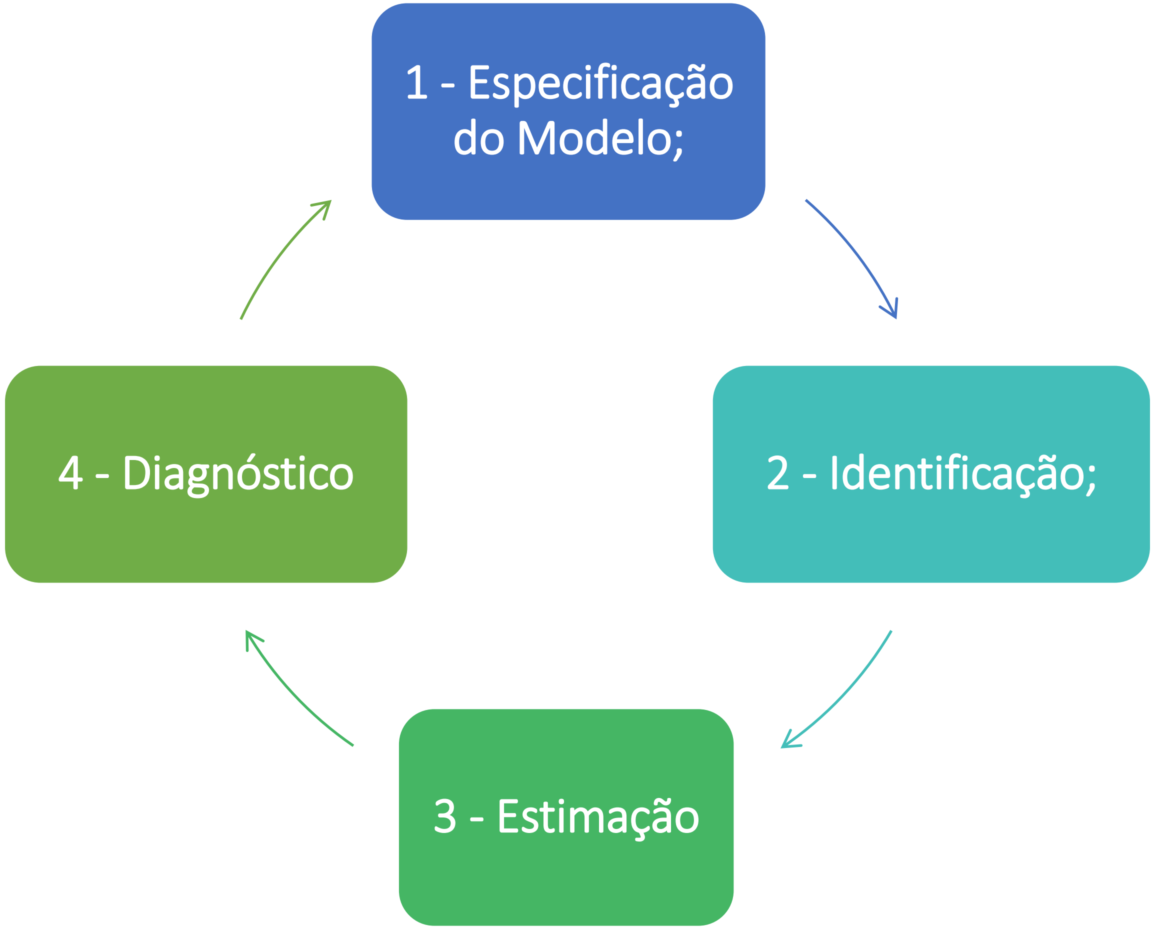
- **p** — parâmetros auto-regressivos,
- **d** — processos de diferenciação (ordem de integração)
- **q** — parâmetros de médias móveis

ARIMA (p, d, q)

Por exemplo, um ARIMA (1, 2, 0) possui um parâmetro auto-regressivo, duas diferenciações desde a série original (integrado de ordem dois) e nenhum parâmetro de médias móveis.

METODOLOGIA DE BOX AND JENKINGS

Lógica da proposta de Box & Jenkings:



1 – Especificação do Modelo

Esse passo já considera diferenciar a série de dados até deixá-la estacionária:

Primeira diferença:

$\Delta \boldsymbol{Y}_t = \boldsymbol{Y}_t - \boldsymbol{Y}_{t-1}$ -> $\mathbf{I}(1)$: Integrada de ordem 1 -> d = 1

Segunda diferença:

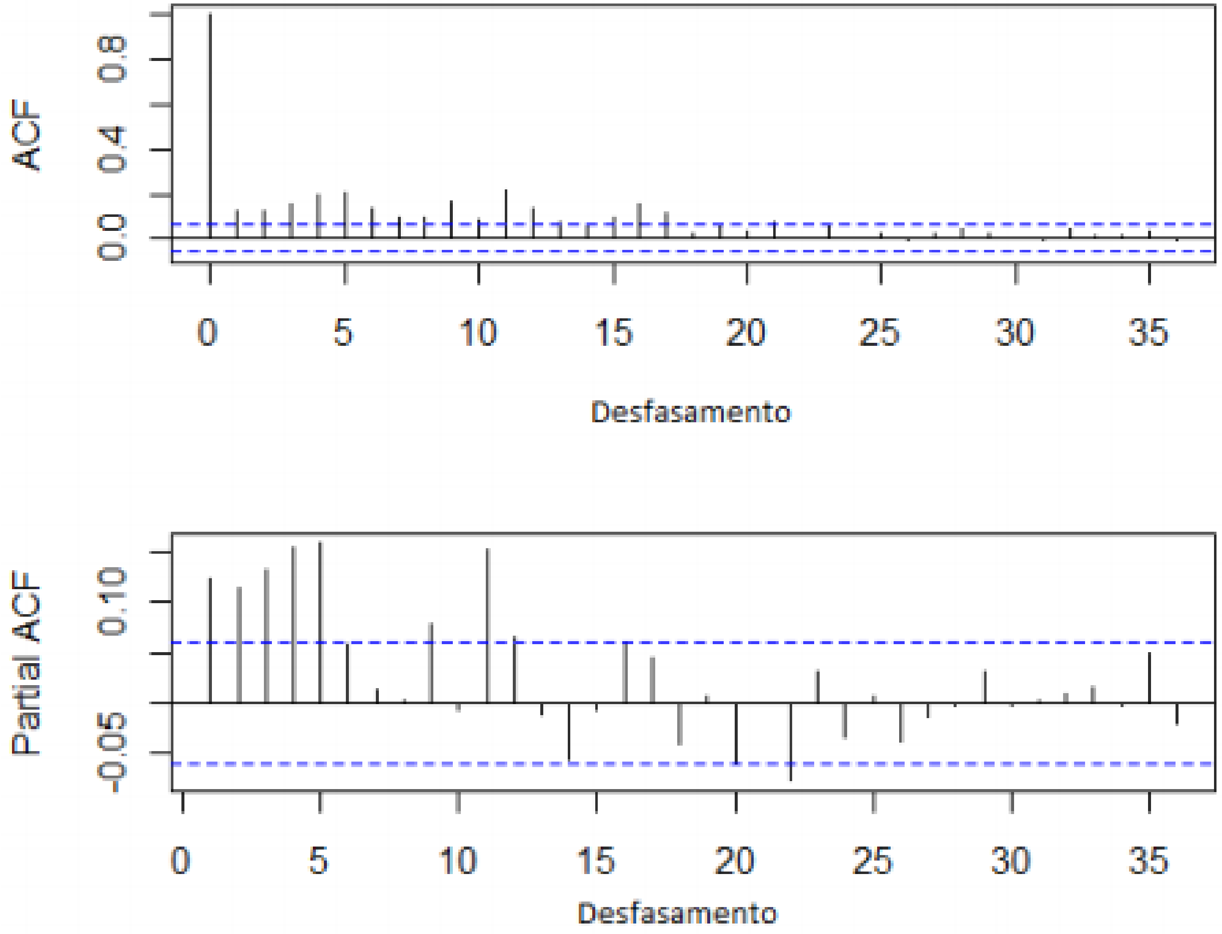
$\Delta(\Delta \boldsymbol{Y}_t) = \Delta \boldsymbol{Y}_t - \Delta \boldsymbol{Y}_{t-1}$ -> $\mathbf{I}(2)$: Integrada de ordem 2 -> d = 2

Por meio da FAC e FACP especifica-se se o modelo possui termos autorregressivos AR e de médias móveis MA.

Exemplo:

ARIMA(11, 1 , 17)

Considerando que a série foi diferenciada uma vez para se tronar estacionária



2 – Identificação do Modelo

Considerando o exemplo do ARIMA (11, 1 , 17), devemos identificar as ordens p e q para o melhor modelo.

Os modelos a considerar são:

ARIMA(1, 1 , 1)
ARIMA(1, 1 , 2)
...
ARIMA(1, 1 , 17)

ARIMA(2, 1 , 1)
ARIMA(2, 1 , 2)
...
ARIMA(2, 1 , 17)

ARIMA(3, 1 , 1)
... ...
ARIMA(17, 1 , 17)

Calcula-se os Critérios de Informação BIC e AIC para todos modelos e o modelo com menor valor indica a ordem do melhor modelo.

3 – Estimação

Identificado a ordem do melhor modelo, passamos então a estima-lo e delineamos sua forma funcional.

Por exemplo se o melhor modelo foi um ARIMA (2,1,2) sua forma funcional é:

$$\Delta Y_t = c + \emptyset_1 \Delta Y_{t-1} + \emptyset_2 \Delta Y_{t-2} + \theta_1 \Delta \varepsilon_{t-1} + \theta_2 \Delta \varepsilon_{t-2} + e$$

AR(1) **MA(1)**

AR(2) **MA(2)**

ARMA(2,1)

ARMA(2,2)

4 – Diagnóstico dos Resíduos

. Autocorrelação: Teste LM e Ljung Box

. Normalidade: Jarque-Bera, Histogramas e Kernel

. Ruído Branco: Portmanteau

. Heterocedasticidade: ARCH- LM

Teste de Heterocedasticidade ARCH -LM

O teste ARCH-LM serve para testar presença de heterocedasticidade no modelo.

Considere:

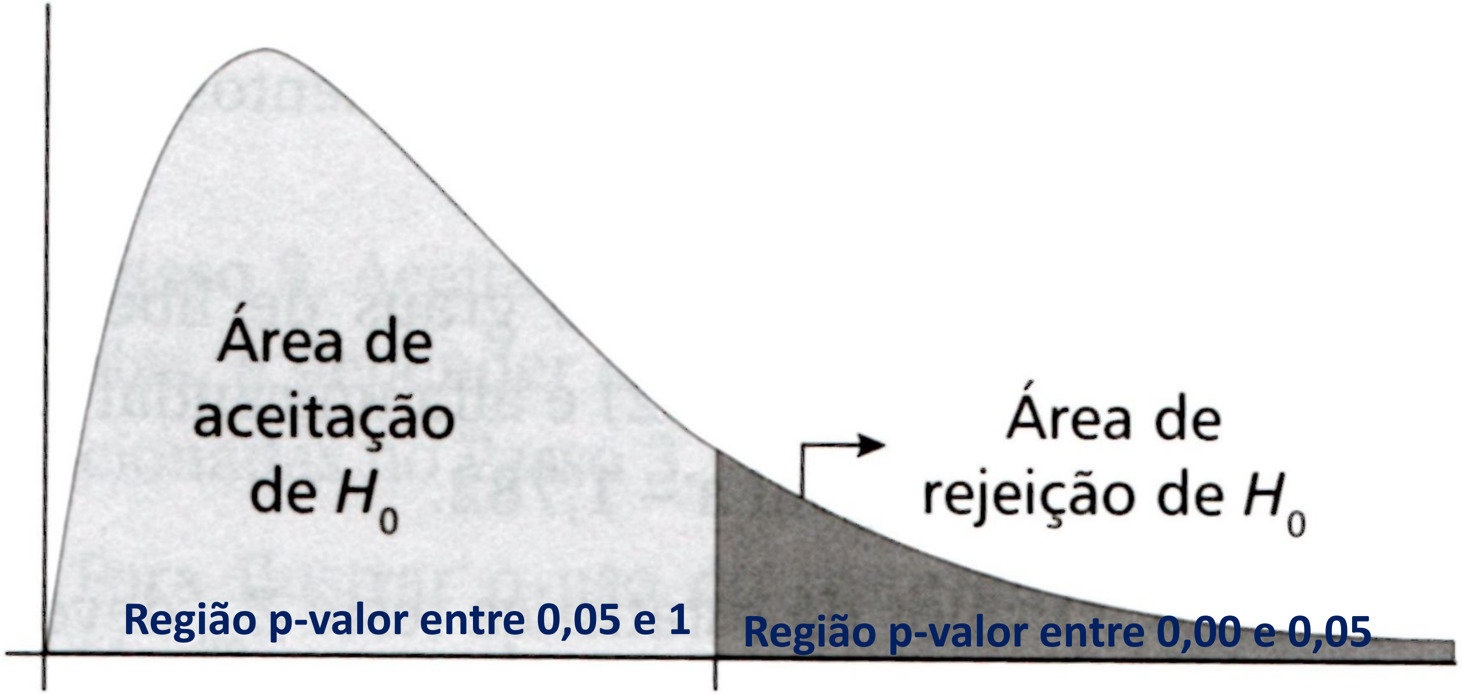
$$\hat{\varepsilon}_t = \beta_1 \widehat{\varepsilon^2}_{t-1} + \beta_2 \widehat{\varepsilon^2}_{t-2} + \ \ldots + \beta_h \widehat{\varepsilon^2}_{t-h} + \ \mu_t$$

Teste:

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_{h=} 0$$
$$H_1: \beta_1 \neq 0 \ ou \ \beta_2 \neq 0 \ldots \beta_h \neq 0$$

$$ARCH - LM_h = R^2 \xrightarrow{d} X^2_{h'}$$

H_0: não há heterocedasticidade no modelo.

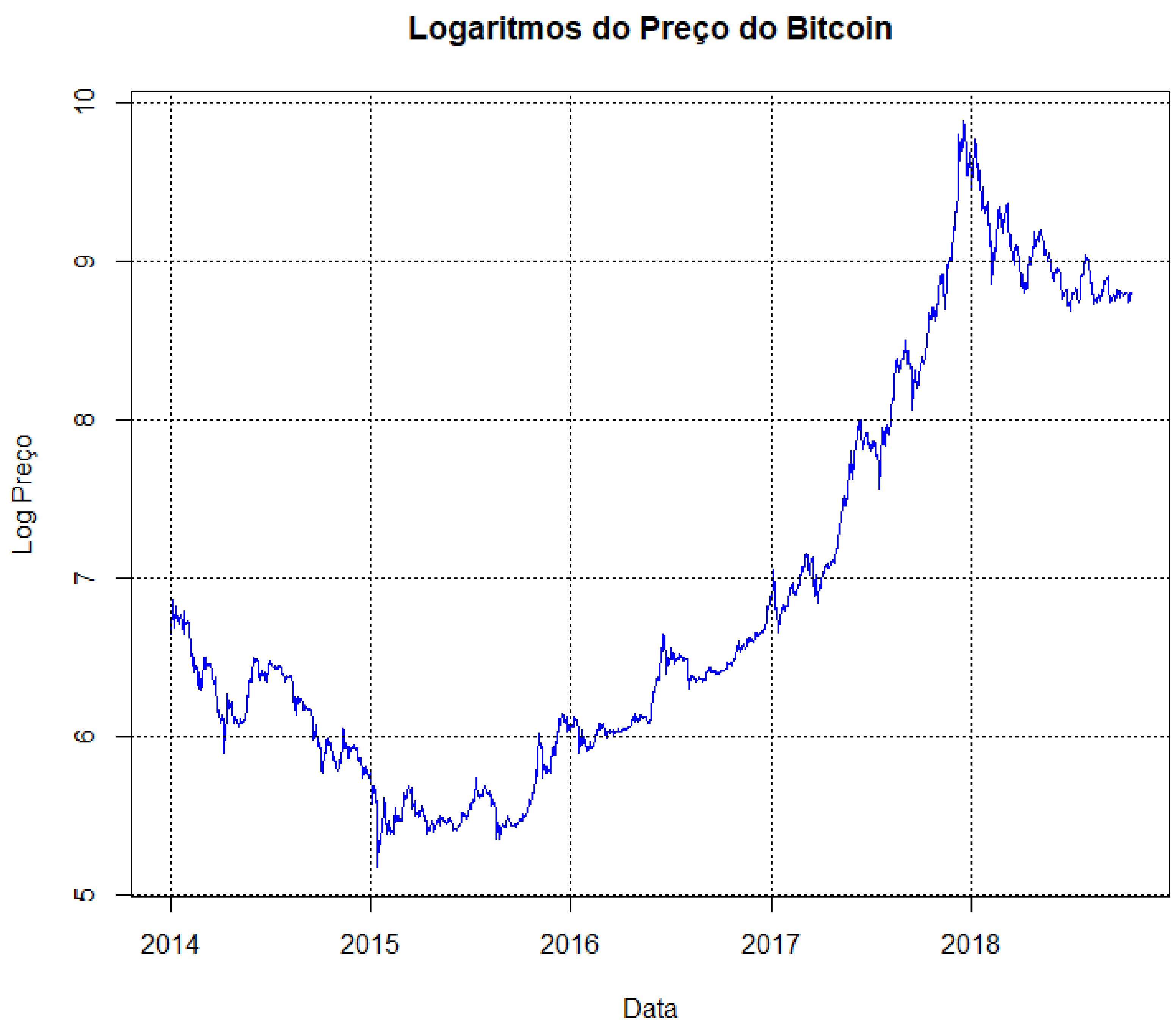


Se o valor-p > 0,05 não rejeitaremos a hipótese nula: não há heterocedasticidade

Se o valor-p < 0,05 rejeitaremos a hipótese nula: há heterocedasticidade

```r
remove.packages("readxl")
install.packages("readxl", dependencies = T)
remove.packages("aTSA")
install.packages("aTSA", dependencies = T)
remove.packages("tseries")
install.packages("tseries", dependencies = T)

library(readxl)
library(aTSA)
library(tseries)
library("urca")

BITCOIN <- na.omit(read_excel("C:/Econometria/Bitcoin.xls"))

Bitcoin <-  ts(log(BITCOIN$Close), start = 2014, frequency = 365)
```

```r
plot(Bitcoin, type="l", main="Logaritmos do Preço do Bitcoin", ylab="Log Preço", xlab="Data", col="Blue")
grid(col = "black", lty = "dotted")
```



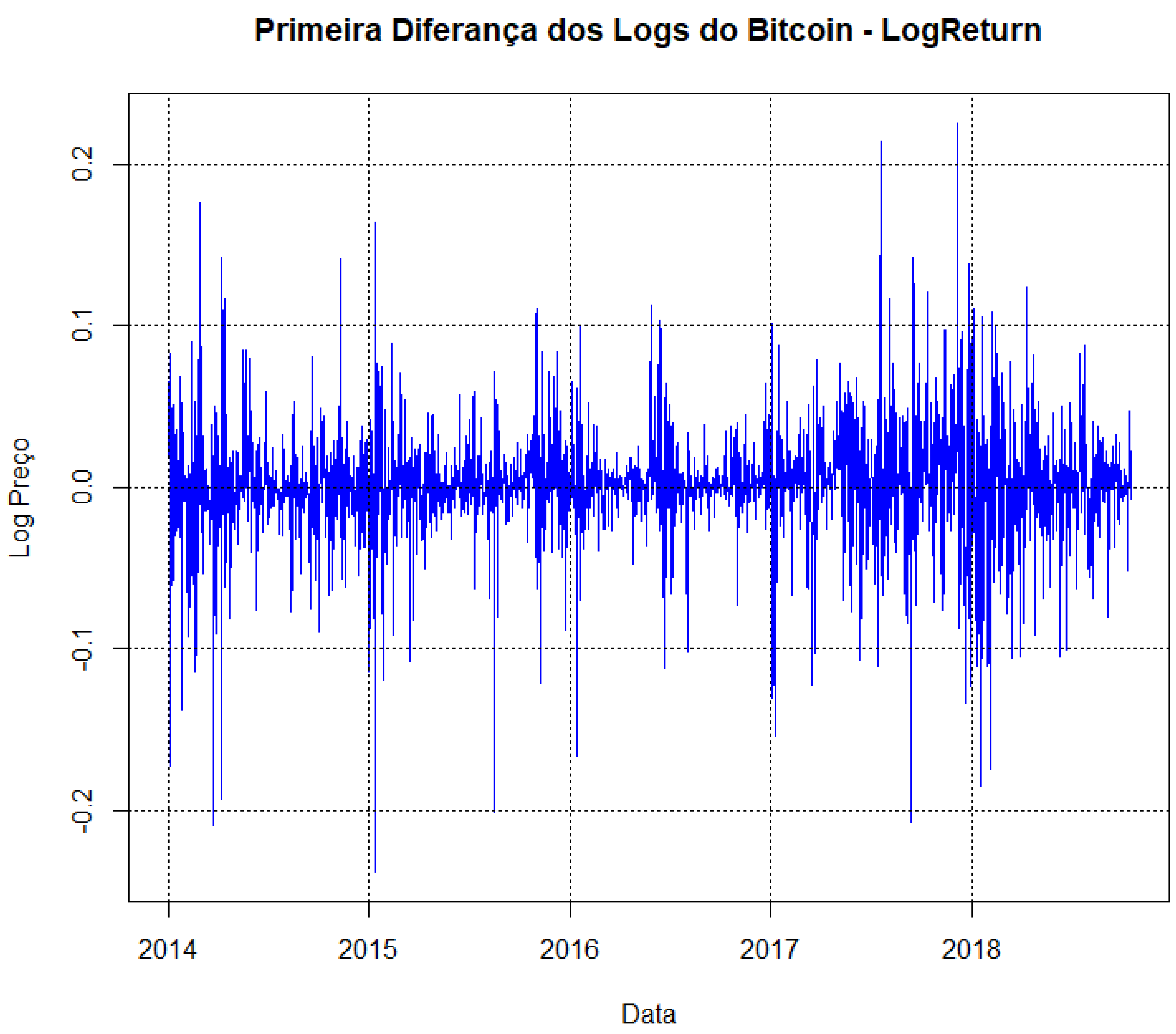
Logaritmos do Preço do Bitcoin
10
9
8
7
6
5
Log Preço
2014
2015
2016
2017
2018
Data

```r
acf(log(BITCOIN$Close),lend=2, lwd=5,col="darkblue",main= "Função Autocorrelação - FAC")
axis(1,tck = 1, col = "lightgrey", lty = "dotted")

pacf(log(BITCOIN$Close),lend=60, lwd=5,col="darkblue",main= "Função Autocorrelação Parcial - FACP")
axis(1,tck = 1, col = "lightgrey", lty = "dotted")

#Teste ADF
ur.df(Bitcoin, "none", lags = 1)

#Teste Philips-Perron
pp.test(Bitcoin)

#Teste KPSS
kpss.test(Bitcoin)
```

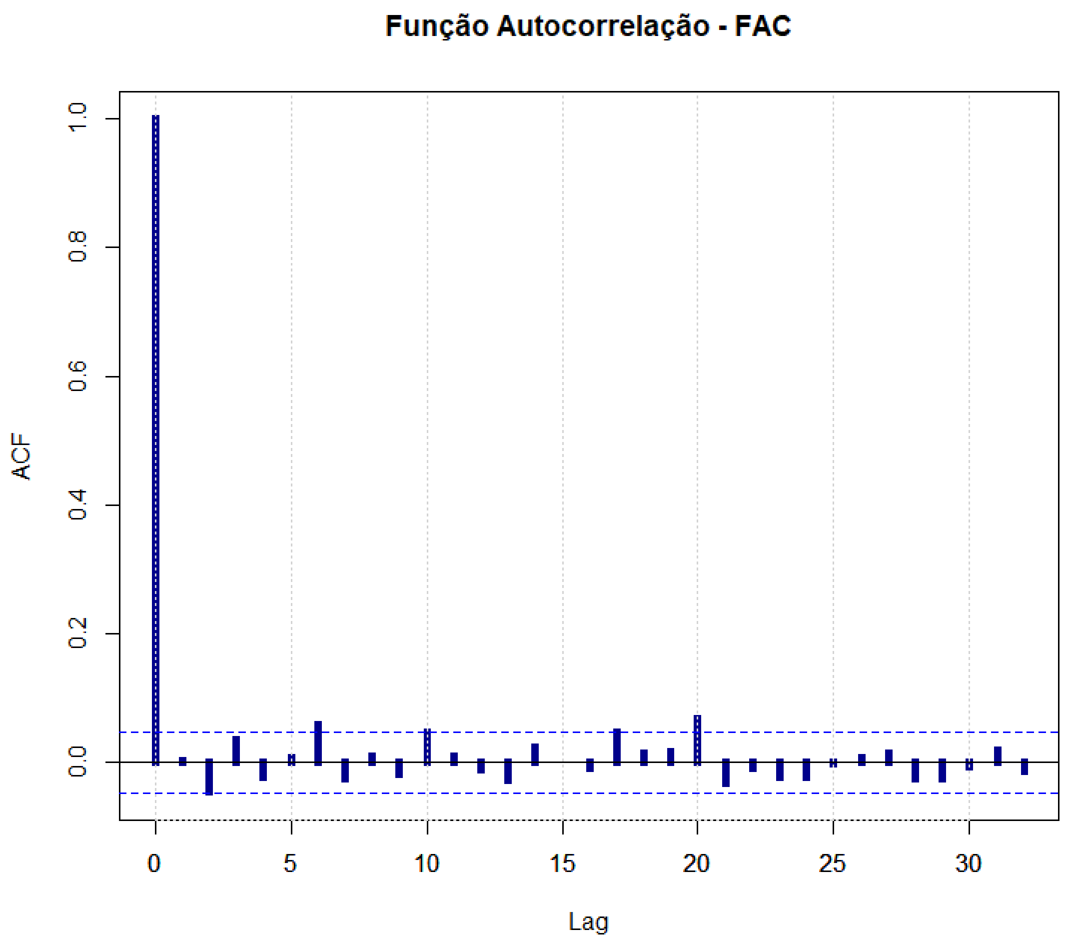
```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-0.23806 -0.01356  0.00055  0.01638  0.22302

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
z.lag.1    0.0001711  0.0001344   1.273    0.203
z.diff.lag 0.0025478  0.0239236   0.106    0.915

Residual standard error: 0.03953 on 1747 degrees of freedom
Multiple R-squared:  0.0009409,	Adjusted R-squared:  -0.0002028
F-statistic: 0.8227 on 2 and 1747 DF,  p-value: 0.4394


Value of test-statistic is: 1.2735

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

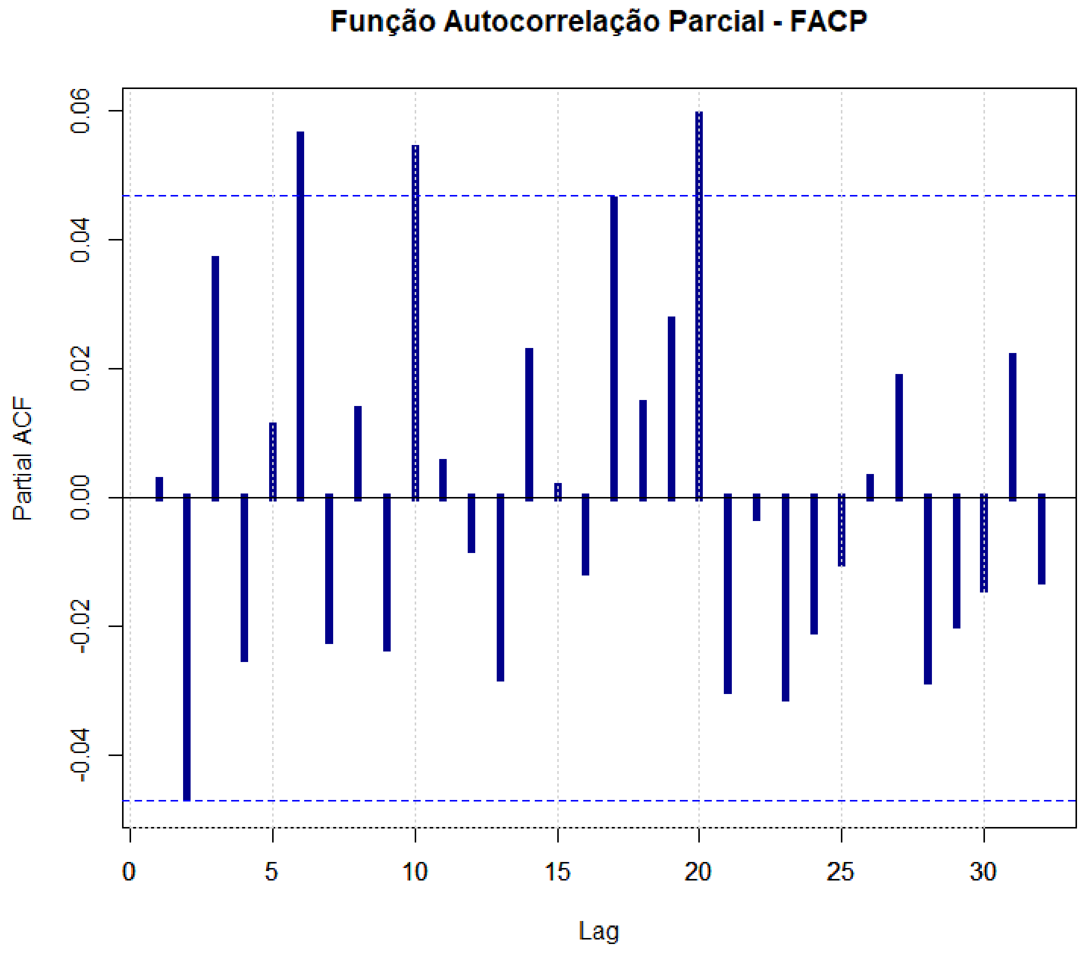
```
	Phillips-Perron Unit Root Test

data:  Bitcoin
Dickey-Fuller Z(alpha) = -4.91, Truncation lag parameter = 8, p-value = 0.8359
alternative hypothesis: stationary
```

```
	KPSS Test for Level Stationarity

data:  Bitcoin
KPSS Level = 13.825, Truncation lag parameter = 9, p-value = 0.01
```


Primeira Diferança dos Logs do Bitcoin - LogReturn
0.2
0.1
0.0
-0.1
-0.2
Log Preço
2014
2015
2016
2017
2018
Data

```
acf(lntOrdem1,lend=2, lwd=5,col="darkblue",main= "Função Autocorrelação - FAC")
axis(1,tck = 1, col = "lightgrey", lty = "dotted")

pacf(lntOrdem1,lend=60, lwd=5,col="darkblue",main= "Função Autocorrelação Parcial - FACP")
axis(1,tck = 1, col = "lightgrey", lty = "dotted")
```

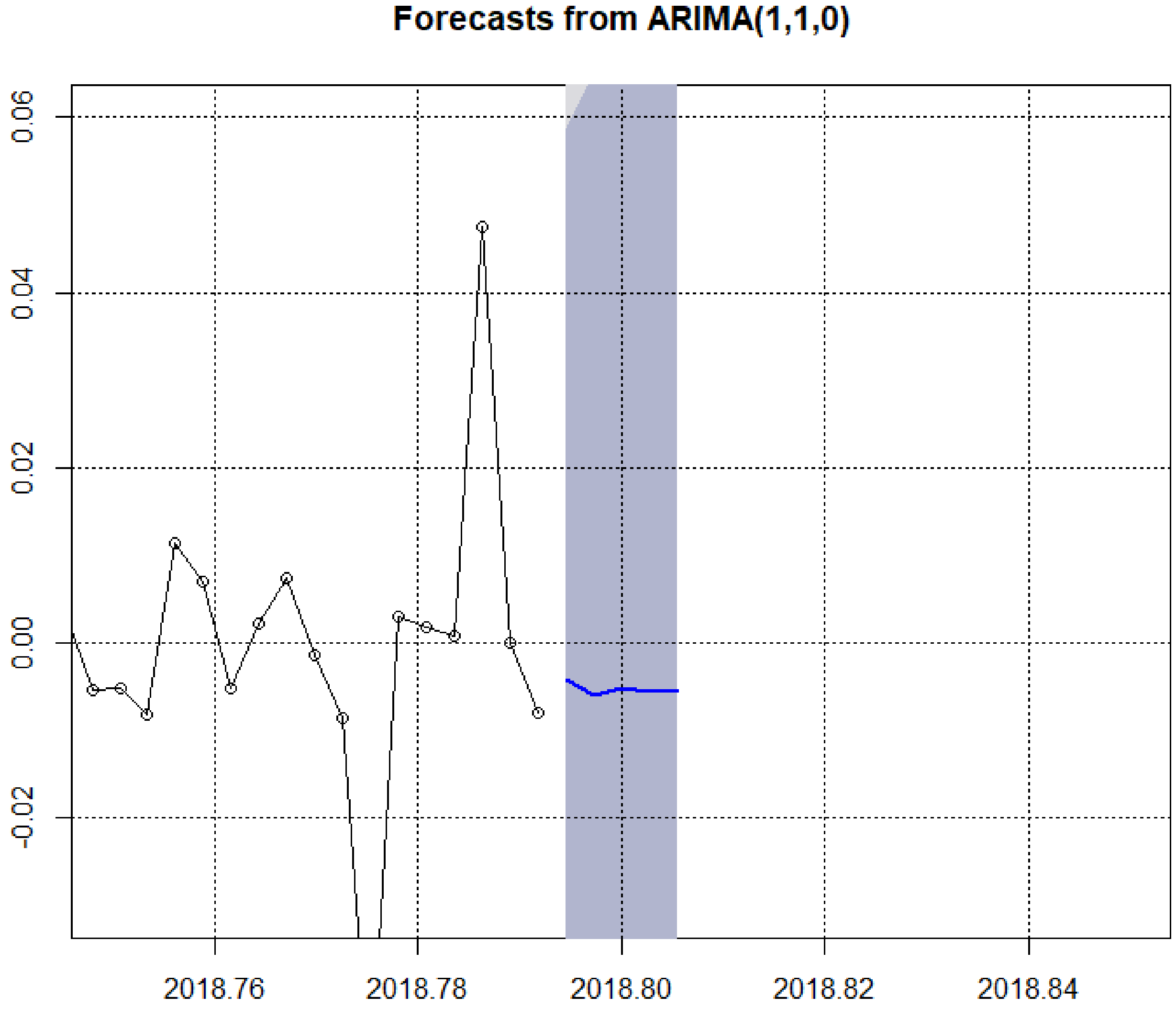




```
AR1 <- arima(IntegradaOrdem1, order = c(1,1,0))
AR1


Call:
arima(x = IntegradaOrdem1, order = c(1, 1, 0))

Coefficients:
          ar1
      -0.4753
s.e.   0.0210

sigma^2 estimated as 0.002409:  log likelihood = 2789.94,  aic = -5575.89
```

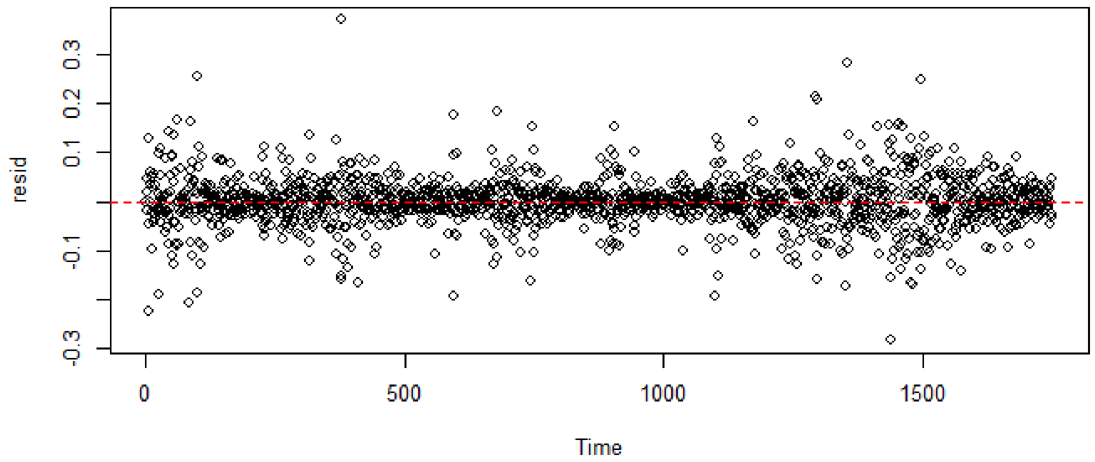
$$\Delta Y_t = -0.4753 \Delta Y_{t-1} + e$$

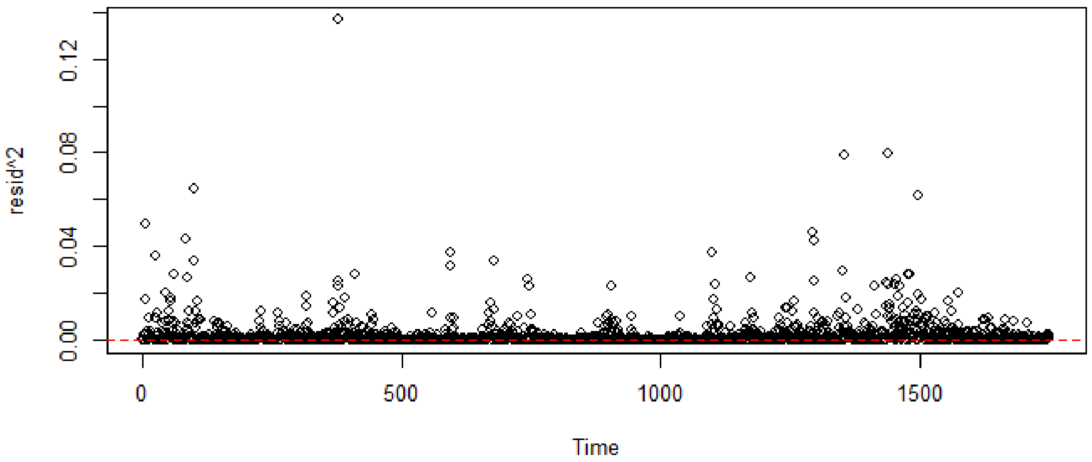
```
#Previsões

predict(AR1,15)

library(forecast)
forecast(AR1,15)

plot(forecast(AR1,5), type="o", xlim=c(2018.75,2018.85), ylim=c(-0.03,0.06))
grid(col = "black", lty = "dotted")
```



```
> predict(AR1,15)
$pred
Time Series:
Start = c(2018, 291)
End = c(2018, 305)
Frequency = 365
 [1] -0.004158340 -0.005920396 -0.005082933 -0.005480959 -0.005291787 -0.005381696 -0.005338964 -0.005359274
 [9] -0.005349621 -0.005354209 -0.005352028 -0.005353065 -0.005352572 -0.005352806 -0.005352695

$se
Time Series:
Start = c(2018, 291)
End = c(2018, 305)
Frequency = 365
 [1] 0.04908621 0.05543342 0.06656120 0.07367063 0.08117124 0.08758117 0.09375332 0.09945346 0.10488468
[10] 0.11002980 0.11495320 0.11967041 0.12421035 0.12858929 0.13282432
```

```
ARCH heteroscedasticity test for residuals
alternative: heteroscedastic

Portmanteau-Q test:
     order  PQ p.value
[1,]     4 221       0
[2,]     8 271       0
[3,]    12 291       0
[4,]    16 326       0
[5,]    20 348       0
[6,]    24 355       0
Lagrange-Multiplier test:
     order   LM p.value
[1,]     4 1570       0
[2,]     8  748       0
[3,]    12  490       0
[4,]    16  352       0
[5 arch.test(object, output = TRUE)
[6,]    24  227       0
```





```r
#Modelando a Variância

residuos <- AR1$residuals
plot(residuos, type="o", main="Residuos do AR1")
grid(col = "black", lty = "dotted")

#FAC  e FACP  dos Residuos

acf(residuos,lend=2, lwd=5,col="darkblue",main= "Função Autocorrelação - FAC")
axis(1,tck = 1, col = "lightgrey", lty = "dotted")

pacf(residuos,lend=60, lwd=5,col="darkblue",main= "Função Autocorrelação Parcial - FACP")
axis(1,tck = 1, col = "lightgrey", lty = "dotted")


GARCH202=garch(residuos,c(20,2),trace=F)
```