

# Lista 1 - Econometria II

Gilberto Gonçalves, Henrique Fonseca, José Santos, Pedro Pereira e Tatianne Wang

2023-09-02

## Questão 2

### Importando bibliotecas

```
library('tidyverse')

## Warning: package 'tidyverse' was built under R version 4.2.3
## Warning: package 'ggplot2' was built under R version 4.2.3
## Warning: package 'tidyr' was built under R version 4.2.3
## Warning: package 'readr' was built under R version 4.2.3
## Warning: package 'purrr' was built under R version 4.2.3
## Warning: package 'forcats' was built under R version 4.2.3
```

```
library('tseries')    # Para cálculos de ACF e PACF
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

```
library('rugarch')    # Para modelagem GARCH
```

```
## Warning: package 'rugarch' was built under R version 4.2.3
```

### Lendo dados do arquivo

```
df <- read_csv("C:/Users/ggonc/OneDrive/Documentos/AMZN.csv")

## Rows: 1258 Columns: 7
## -- Column specification -----
## Delimiter: ","
## dbl  (6): Open, High, Low, Close, Adj Close, Volume
## date (1): Date
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

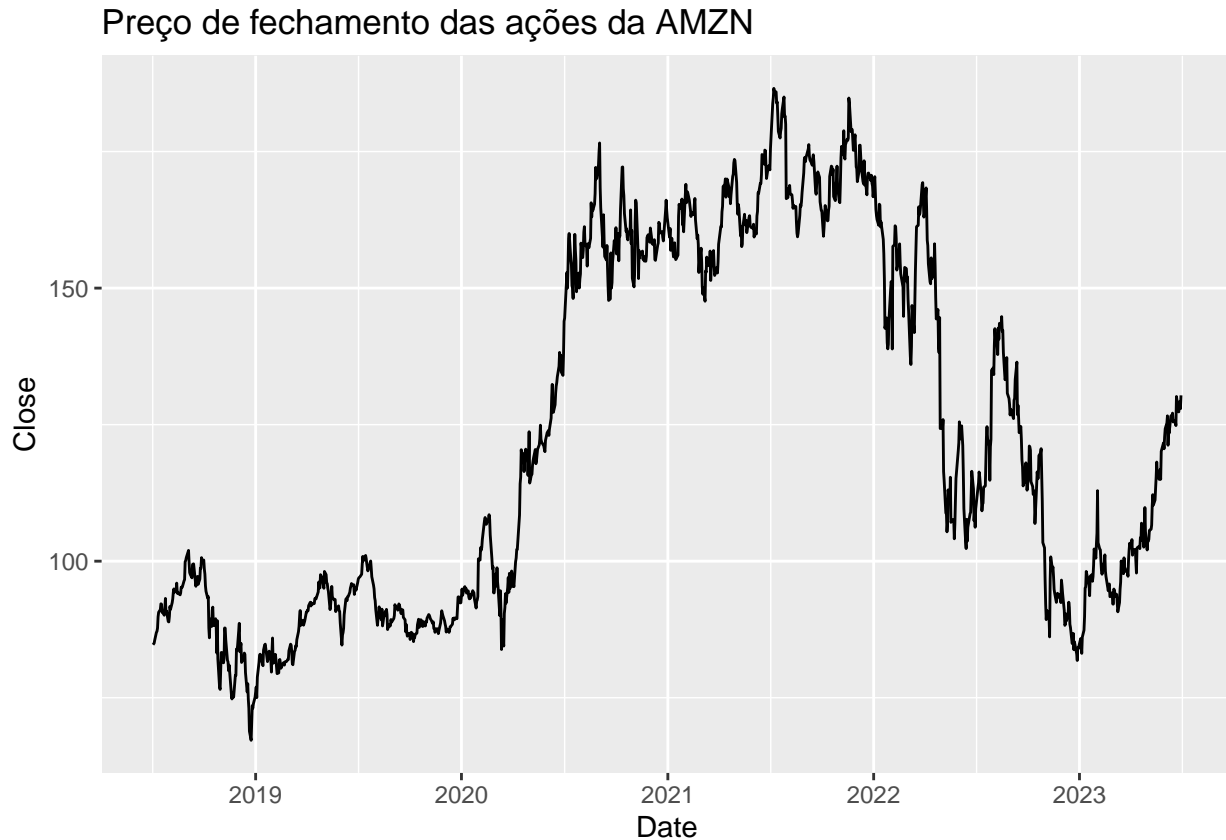
```
df$Date <- as.Date(df$Date, format = "%m/%d/%Y")
df$Returns <- (df$Close - lag(df$Close)) / lag(df$Close)
df <- na.omit(df)
df$Returns2 <- df$Returns^2
```

## Item 2.1

Vamos estimar um modelo GARCH(1,1).

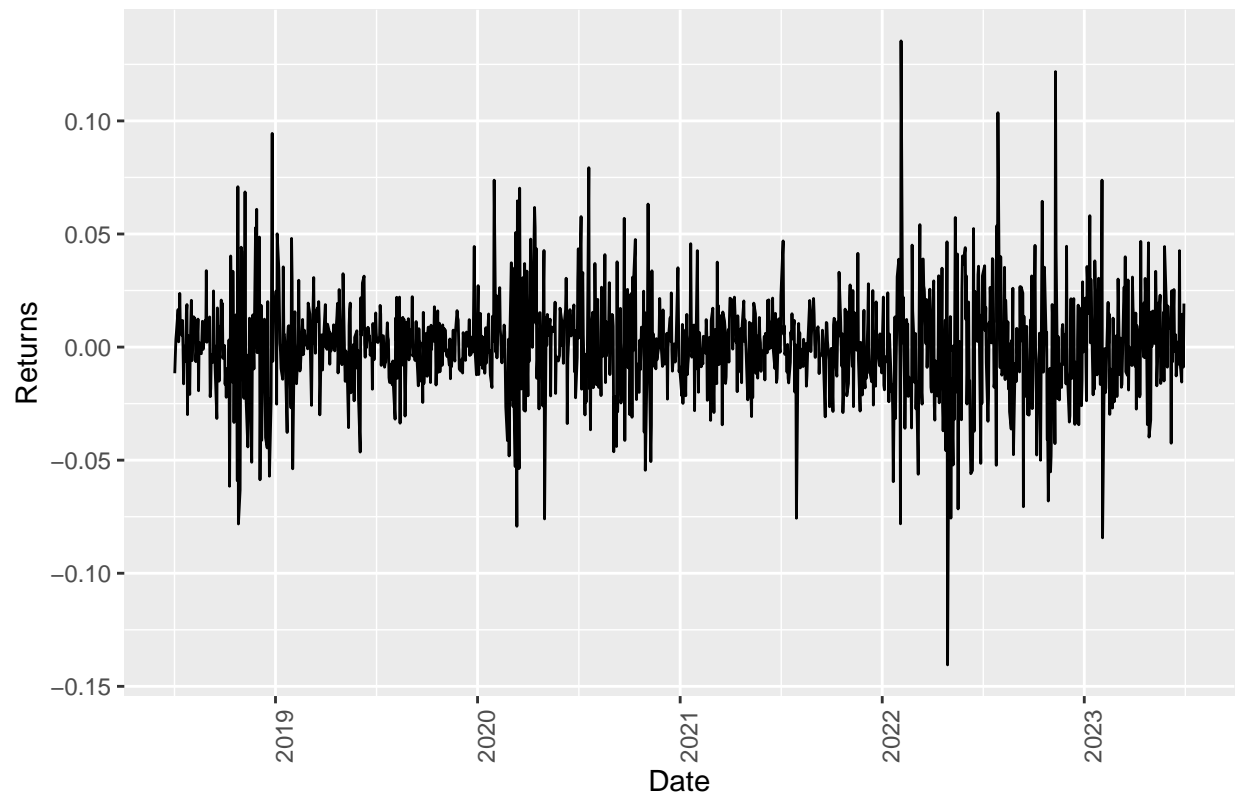
### Plotando os gráficos de preços

```
ggplot(df, aes(x = Date, y = Close)) + geom_line() +
  labs(title = "Preço de fechamento das ações da AMZN")
```



```
ggplot(df, aes(x = Date, y = Returns)) + geom_line() +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title = "Retorno diário das ações da AMZN")
```

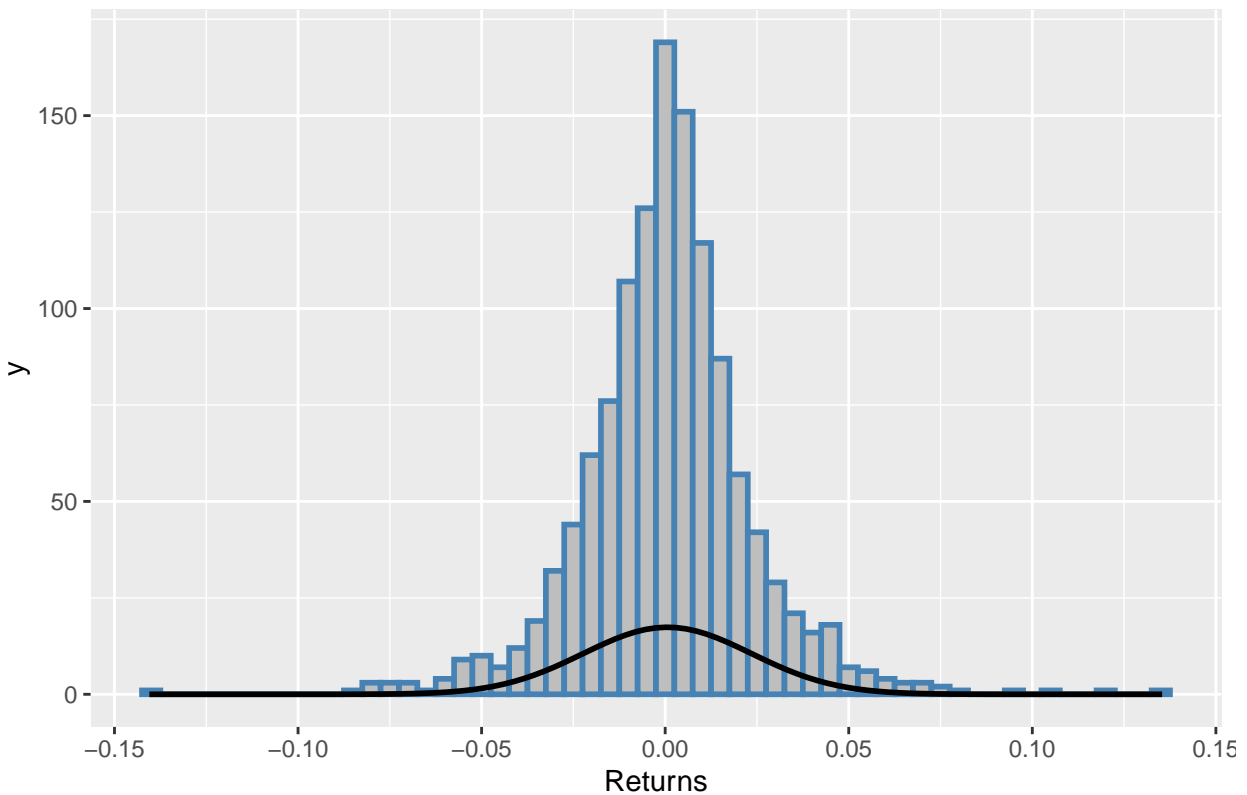
## Retorno diário das ações da AMZN



```
ggplot(df, aes(x = Returns)) +  
  geom_histogram(bins = 50, binwidth = 0.005, color="steelblue", fill="grey", size=1) +  
  stat_function(fun = dnorm, args = list(mean = mean(df$Returns, na.rm = T),  
                                         sd = sd(df$Returns, na.rm = T)), size=1) +  
  labs(title = "Distribuição dos retornos diários da AMZN")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.  
## i Please use 'linewidth' instead.  
## This warning is displayed once every 8 hours.  
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was  
## generated.
```

## Distribuição dos retornos diários da AMZN



## Estatísticas descritivas

```
summary(df$Close)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  67.20   92.66  114.30  122.82  158.10  186.57
```

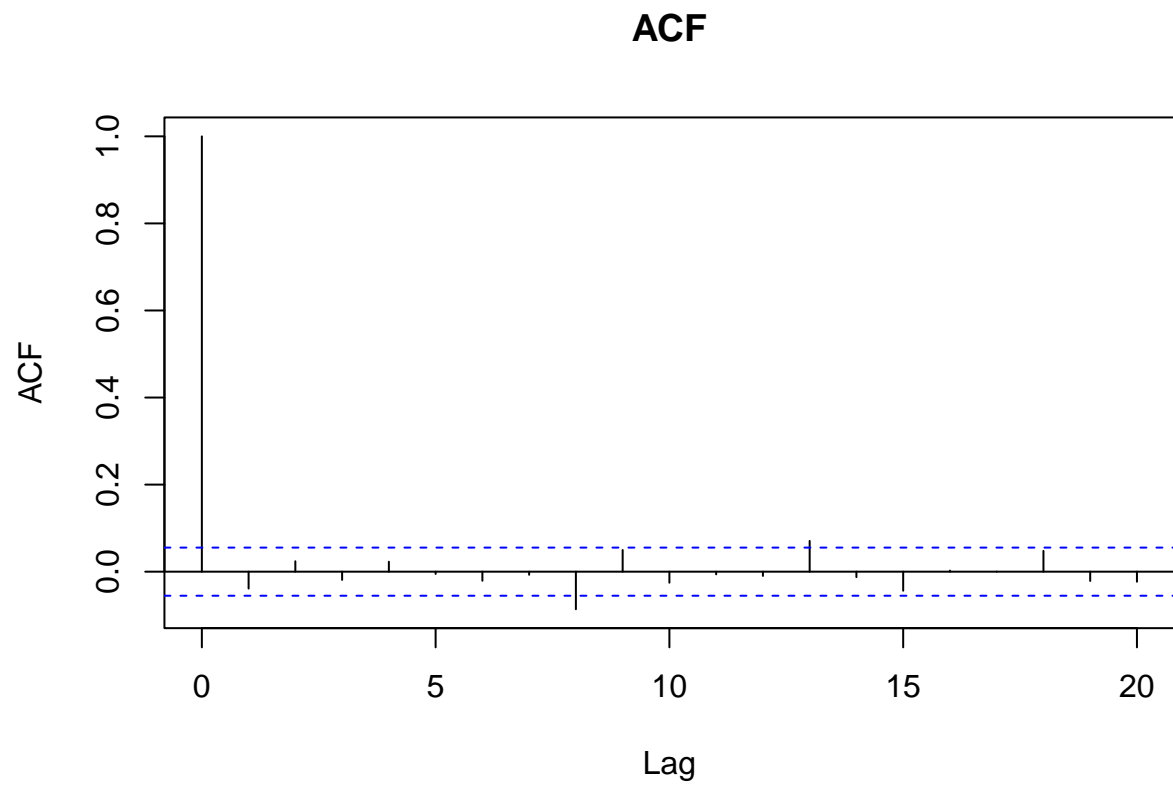
```
summary(df$Returns)
```

```
##      Min.    1st Qu.    Median      Mean   3rd Qu.     Max.
## -0.1404944 -0.0110770  0.0009821  0.0005970  0.0120764  0.1353590
```

## Funções de autocorrelação e autocorrelação parcial

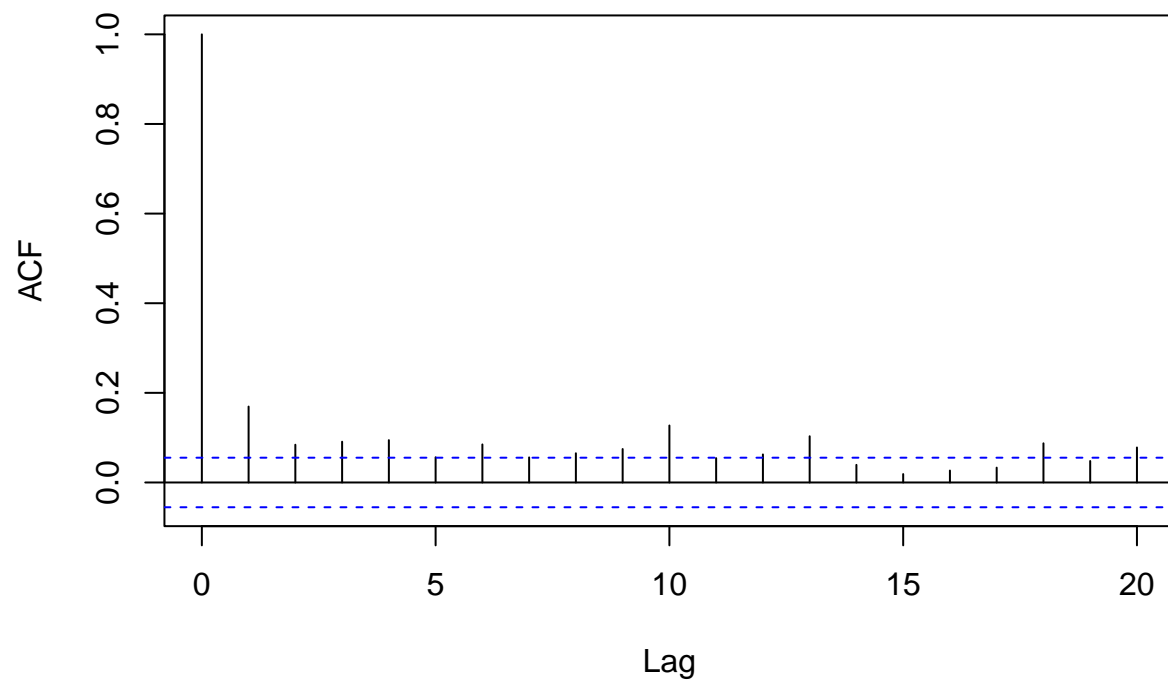
```
acf_valores <- acf(df$Returns, lag.max = 20, plot = FALSE)
acf_valores2 <- acf(df$Returns2, lag.max = 20, plot = FALSE)
pacf_valores <- pacf(df$Returns, lag.max = 20, plot = FALSE)
pacf_valores2 <- pacf(df$Returns2, lag.max = 20, plot = FALSE)

plot(acf_valores, main = "ACF") #Plota o gráfico da função de autocorrelação
```

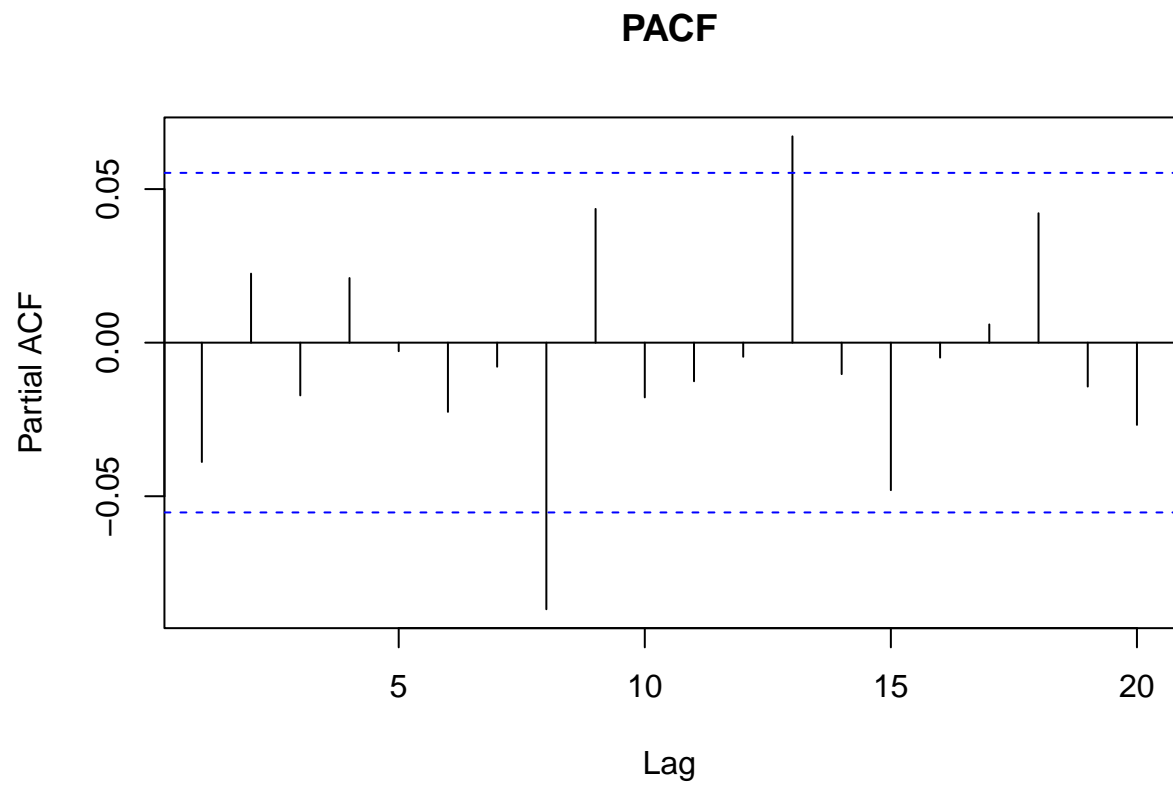


```
plot(acf_valores2, main = "ACF^2") #Plota o gráfico da função de autocor^2
```

## ACF^2

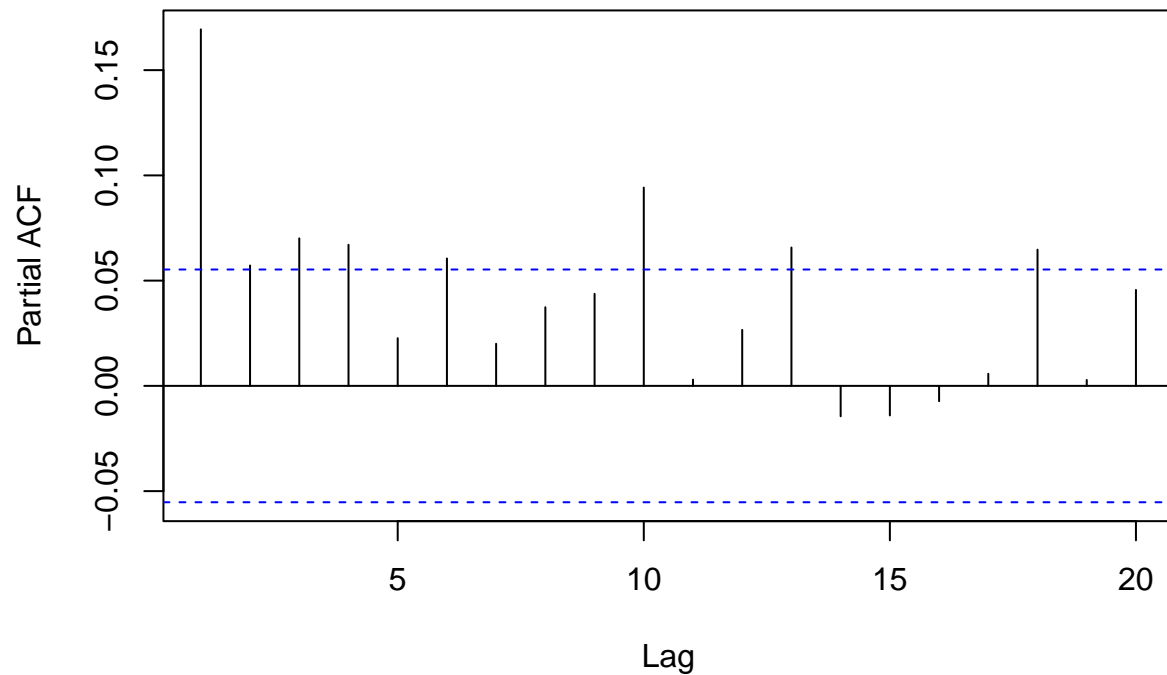


```
plot(pacf_valores, main = "PACF") #Plota o gráfico da função de autocor. parcial
```



```
plot(pacf_valores2, main = "PACF^2") #Plota o gráf. da função de autocor. parc^2
```

## PACF^2



## Estimando o modelo GARCH(1,1)

```
spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)))
modelo <- ugarchfit(data = df>Returns, spec = spec)
print(modelo)
```

```
##
## *-----*
## *          GARCH Model Fit          *
## *-----*
##
## Conditional Variance Dynamics
## -----
## GARCH Model   : sGARCH(1,1)
## Mean Model    : ARFIMA(1,0,1)
## Distribution   : norm
##
## Optimal Parameters
## -----
##      Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001155   0.000434   2.6639 0.007724
## ar1      0.890352   0.115389   7.7161 0.000000
## ma1     -0.909534   0.104593  -8.6959 0.000000
```



```

## omega    0.000017    0.000004    4.3414 0.000014
## alpha1   0.129854    0.021582    6.0167 0.000000
## beta1    0.843113    0.023789   35.4412 0.000000
##
## Robust Standard Errors:
##      Estimate Std. Error t value Pr(>|t|)
## mu      0.001155    0.000455    2.5363 0.011203
## ar1     0.890352    0.108589    8.1993 0.000000
## ma1    -0.909534    0.097873   -9.2930 0.000000
## omega   0.000017    0.000004    3.9415 0.000081
## alpha1  0.129854    0.031731    4.0923 0.000043
## beta1   0.843113    0.029041   29.0322 0.000000
##
## LogLikelihood : 3085.127
##
## Information Criteria
## -----
##
## Akaike          -4.8992
## Bayes           -4.8747
## Shibata         -4.8992
## Hannan-Quinn   -4.8900
##
## Weighted Ljung-Box Test on Standardized Residuals
## -----
##
##              statistic p-value
## Lag[1]              0.1640 0.6855
## Lag[2*(p+q)+(p+q)-1] [5] 0.7845 1.0000
## Lag[4*(p+q)+(p+q)-1] [9] 1.4597 0.9975
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----
##
##              statistic p-value
## Lag[1]              0.09401 0.7591
## Lag[2*(p+q)+(p+q)-1] [5] 0.93966 0.8732
## Lag[4*(p+q)+(p+q)-1] [9] 2.11847 0.8904
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----
##
##      Statistic Shape Scale P-Value
## ARCH Lag[3] 0.003325 0.500 2.000 0.954
## ARCH Lag[5] 0.572723 1.440 1.667 0.862
## ARCH Lag[7] 1.441454 2.315 1.543 0.833
##
## Nyblom stability test
## -----
## Joint Statistic: 3.2139
## Individual Statistics:
## mu      0.12766
## ar1     0.03042
## ma1     0.02569

```

```

## omega 0.12232
## alpha1 0.22599
## beta1 0.27778
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:      1.49 1.68 2.12
## Individual Statistic: 0.35 0.47 0.75
##
## Sign Bias Test
## -----
##          t-value  prob sig
## Sign Bias      0.03968 0.9684
## Negative Sign Bias 0.95690 0.3388
## Positive Sign Bias 0.43150 0.6662
## Joint Effect      1.30262 0.7285
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----
##  group statistic p-value(g-1)
## 1    20      41.89   1.836e-03
## 2    30      54.77   2.638e-03
## 3    40      81.84   7.151e-05
## 4    50      80.83   2.830e-03
##
##
## Elapsed time : 0.2281978

```

## Questão 2.2

### Information Criteria para o modelo GARCH(1,1)

Akaike -4.8992

Bayes -4.8747

Shibata -4.8992

Hannan-Quinn -4.8900

### Information Criteria para o modelo GARCH(4,1)

Akaike -4.8931

Bayes -4.8563

Shibata -4.8932

Hannan-Quinn -4.8793

### Information Criteria para o modelo GARCH(1,4)

Akaike -4.8948

Bayes -4.8580

Shibata -4.8949

Hannan-Quinn -4.8810

Dessa forma, a escolha de outros parâmetros resulta em critérios de informação maiores, o que mostra que o modelo GARCH(1,1) é de fato o melhor. Testamos também os modelos GARCH(4,1) e GARCH(1,4) pois há uma queda brusca no gráfico da PACF<sup>2</sup>.

### Questão 2.3

Vamos utilizar o teste estatístico de Ljung-Box com uma defasagem de 35 dias. Escolhemos um lag que fosse menor que a raiz quadrada do número de observações.

```
residuos <- residuals(modelo) # Obtenha os resíduos do modelo
resultado_teste <- Box.test(residuos, lag = 35, type = "Ljung-Box") #Teste de Ljung-Box para autocorrel.
print(resultado_teste)
```

```
##
## Box-Ljung test
##
## data:  residuos
## X-squared = 50.278, df = 35, p-value = 0.04554
```

Como chegamos a um valor p baixo:  $p = 0.048 < 0.05$ . Dessa forma, há evidências de autocorrelação serial nos resíduos. Portanto, é possível que o modelo GARCH(1,1) não esteja capturando totalmente a volatilidade presente na série temporal analisada.

### Questão 2.4

A variância condicional cresce à medida que o tempo passa, pois o futuro faz com que haja mais volatilidade em relação ao preço dos ativos.

### Previsão de 12 meses à frente da variância condicional

```
horizon <- 12
forecast_var <- ugarchforecast(modelo, n.ahead = horizon)
```

### Extraindo os valores das previsões

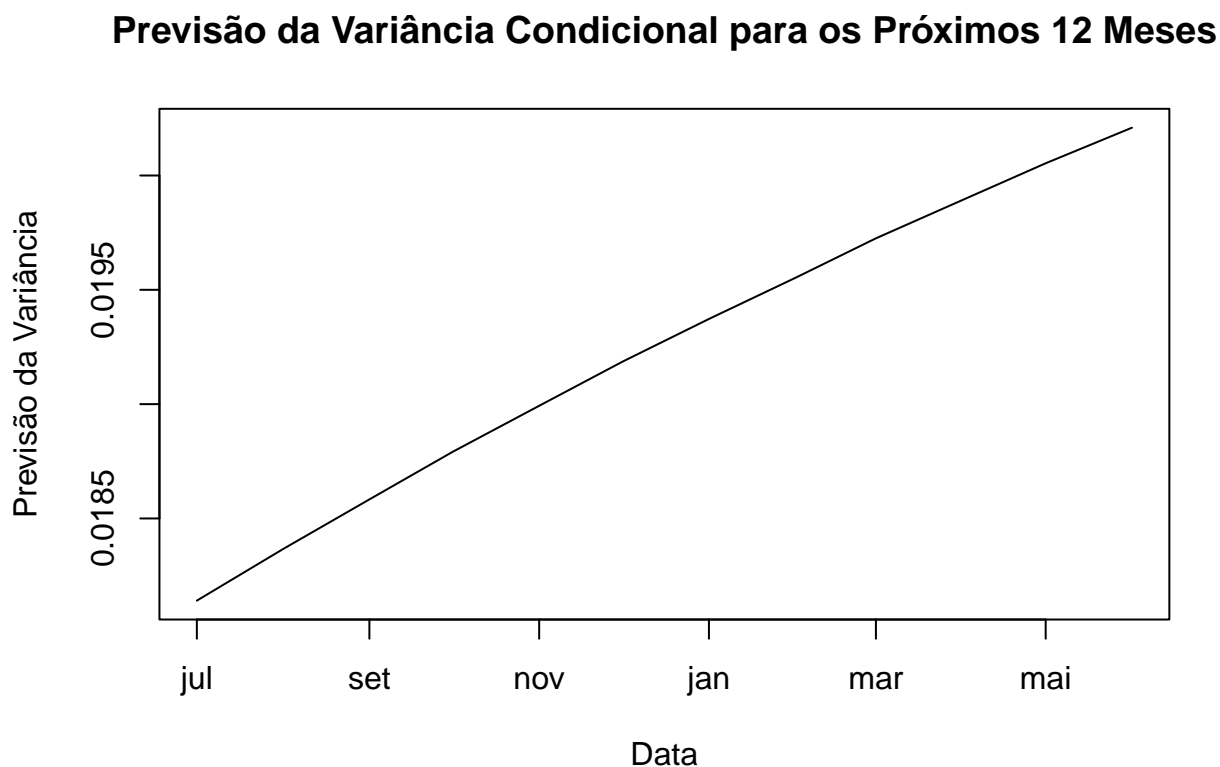
```
forecast_values <- as.numeric(sigma(forecast_var))
```

### Criando um vetor de datas para as previsões

```
forecast_dates <- seq(max(df$Date) + 1, length.out = horizon, by = "months")
```

Criando um gráfico das previsões da variância condicional

```
plot(forecast_dates, forecast_values, type = "l",  
     xlab = "Data", ylab = "Previsão da Variância",  
     main = "Previsão da Variância Condicional para os Próximos 12 Meses")
```



Questão 2.5

Especificando o modelo EGARCH

```
spec_egarch <- ugarchspec(variance.model = list(model = "eGARCH", garchOrder = c(1, 1)))
```

Ajustando o modelo EGARCH aos dados

```
modelo_egarch <- ugarchfit(data = df>Returns, spec = spec_egarch)
```

## Obtendo os critérios de informação para o modelo EGARCH

```
info_criteria_egarch <- infocriteria(modelo_egarch)
```

## Imprimindo os critérios de informação para o modelo EGARCH

```
print("Critérios de Informação para o Modelo EGARCH:")
```

```
## [1] "Critérios de Informação para o Modelo EGARCH:"
```

```
print(info_criteria_egarch)
```

```
##  
## Akaike      -4.917279  
## Bayes      -4.888675  
## Shibata    -4.917340  
## Hannan-Quinn -4.906529
```

## Comparar critérios de informação entre GARCH e EGARCH

```
info_criteria_garch <- infocriteria(modelo)  
print("Critérios de Informação para o Modelo GARCH:")
```

```
## [1] "Critérios de Informação para o Modelo GARCH:"
```

```
print(info_criteria_garch)
```

```
##  
## Akaike      -4.899168  
## Bayes      -4.874650  
## Shibata    -4.899213  
## Hannan-Quinn -4.889953
```