

Questão 1

Shai Vaz, Heron Goulart, Alexandre Almeida, João Pedro Pedrosa, Roberto Orenstein

2023-10-05

Questão 1: S&P 500

Importar Base

```
# importar índice S&P500
ativo <- quantmod::getSymbols(
  "^GSPC",
  src = "yahoo",
  auto.assign = FALSE,
  from = '2015-09-01',
  to = '2023-09-01',
  periodicity = "daily")
```

Série em nível

Selecionando os preços de fechamento, em nível, do índice.

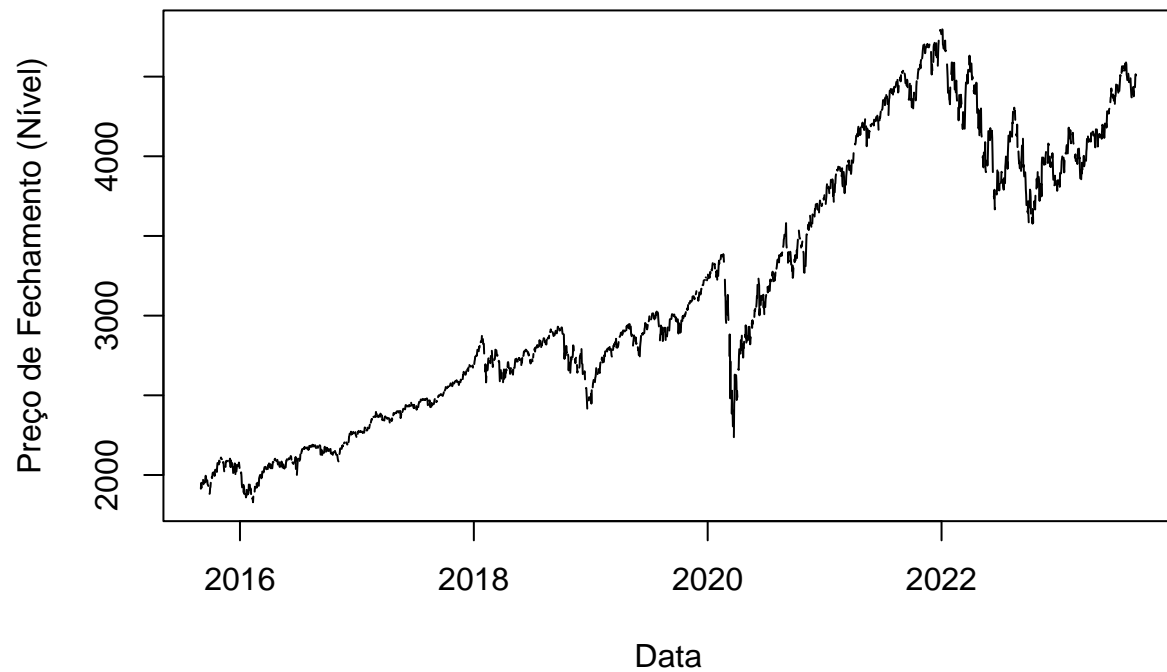
```
close <- ativo[, "GSPC.Close"]
```

Os dados estão em `xts`, mas preferiremos por simplicidade trabalhar com o formato `ts`.

```
# ts (nível)
ts_close <- ts_ts(close)
```

```
plot(ts_close,
     xlab = "Data",
     ylab = "Preço de Fechamento (Nível)",
     main = "S&P 500 Index: Série em Nível")
```

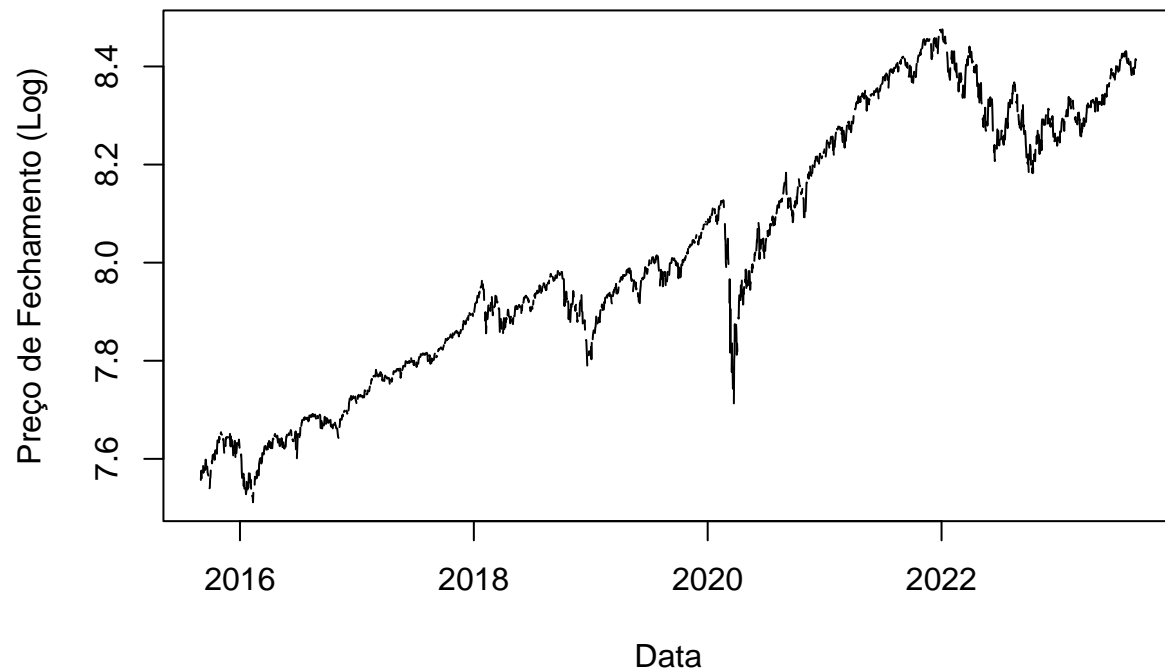
S&P 500 Index: Série em Nível



Série em Log

```
log_close <- log(close)
ts_log_close <- log(ts_close)
plot(ts_log_close,
     xlab = "Data",
     ylab = "Preço de Fechamento (Log)",
     main = "S&P 500 Index: Série em Log")
```

S&P 500 Index: Série em Log

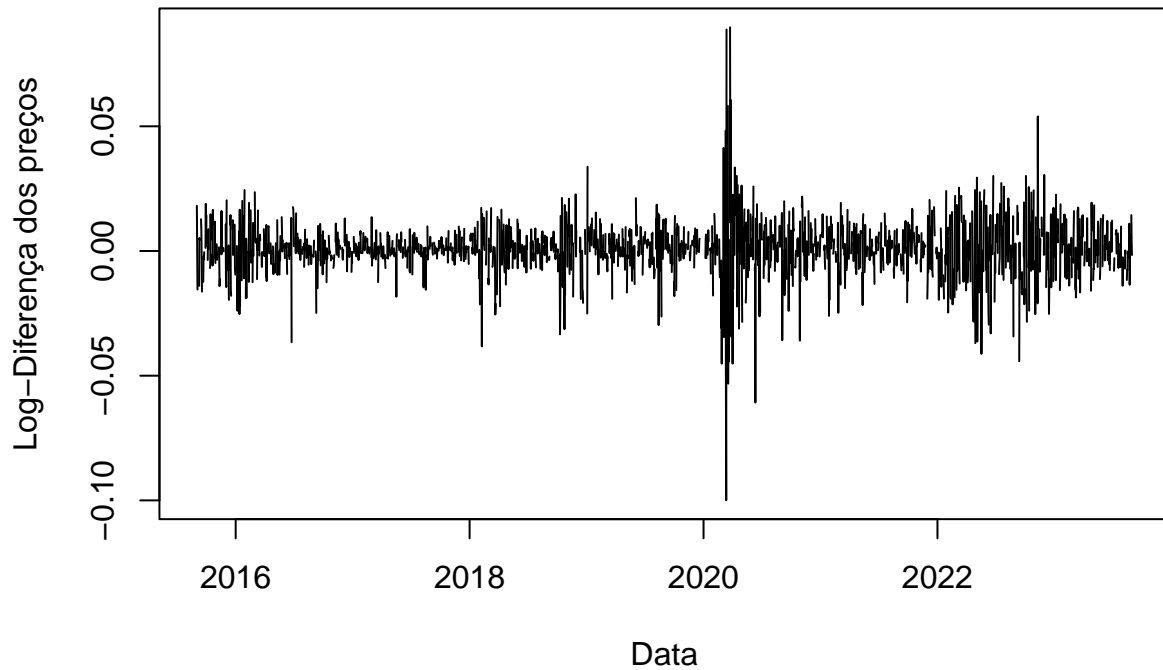


Série em Log-Dif

Tiraremos a primeira diferença da série em log.

```
dif_log_close <- diff(log_close)[-1,]  
ts_dif_log_close <- diff(ts_log_close)  
plot(ts_dif_log_close,  
     xlab = "Data",  
     ylab = "Log-Diferença dos preços",  
     main = "S&P 500 Index: Série Log-Diferença"  
)
```

S&P 500 Index: Série Log-Diferença



Teste de Phillips-Perron (PPT)

Para a série Log-Nível

```
tidy(ppt_log_close) %>%  
  kable(  
    col.names = c(  
      "Statistic: Dickey-Fuller Z (alpha)",  
      "P Value",  
      "Parameter: Truncation lag",  
      "Method",  
      "Alternative Hypothesis"  
    ),  
    caption = "Teste Philips Perron: Série Log Nível"  
  )
```

Table 1: Teste Philips Perron: Série Log Nível

Statistic: Dickey-Fuller Z (alpha)	P Value	Parameter: Truncation lag	Method	Alternative Hypothesis
-18.5791	0.0930023	8	Phillips-Perron Unit Root Test	stationary

Nesse caso, não rejeitamos a hipótese nula ao nível de significância 0.05. Há indícios portanto que a série não é estacionária e tem raiz unitária.

Para a série Log-Dif

```
ppt_dif_log_close <- pp.test(dif_log_close)
```

```
tidy(ppt_dif_log_close) %>%  
  kable(  
    col.names = c(  
      "Statistic: Dickey-Fuller Z (alpha)",  
      "P Value",  
      "Parameter: Truncation lag",  
      "Method",  
      "Alternative Hypothesis"  
    ),  
    caption = "Teste Philips Perron"  
  )
```

Table 2: Teste Philips Perron

Statistic: Dickey-Fuller Z (alpha)	P Value	Parameter: Truncation lag	Method	Alternative Hypothesis
-2404.682	0.01	8	Phillips-Perron Unit Root Test	stationary

Nesse caso, podemos rejeitar a hipótese nula e aceitar a hipótese alternativa ao nível de significância 0.05. Portanto, a série é estacionária.

Modelando a média com um ARIMA(p,d,q)

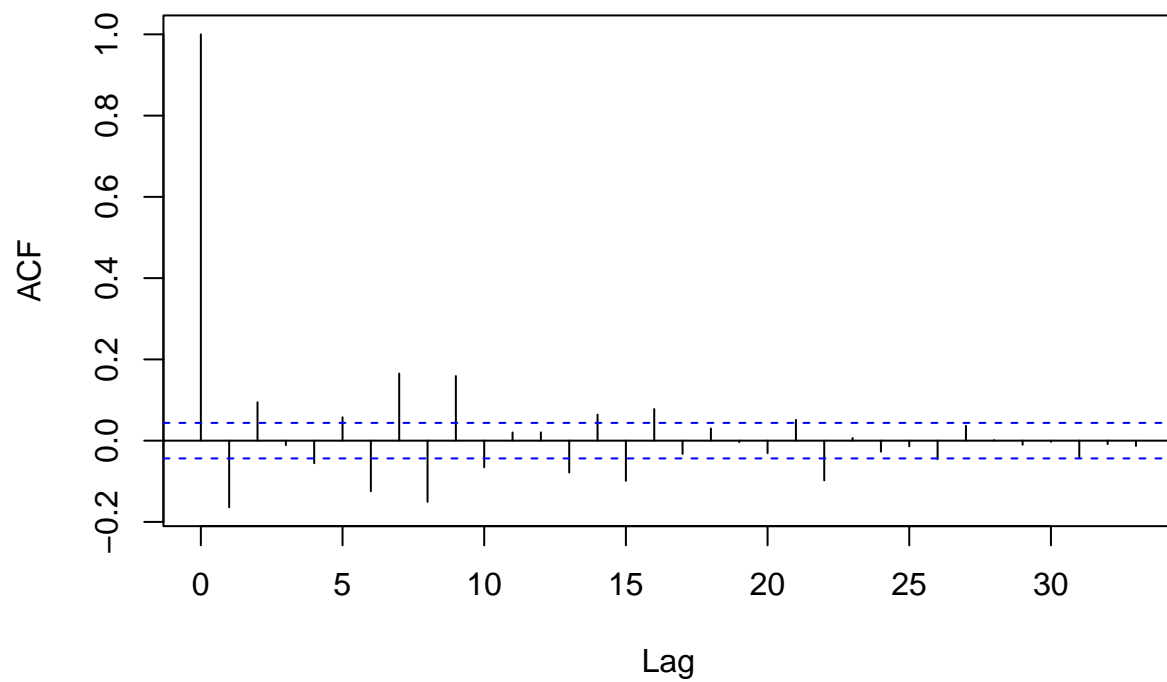
Autocorrelation Functions

Primeiramente, vamos avaliar as funções de autocorrelação e autocorrelação parcial da nossa série escolhida, a série log-dif.

```
s1 <- ts(dif_log_close)
```

```
acf(s1,  
  main = "Autocorrelation Function: S&P500 Log-Dif"  
)
```

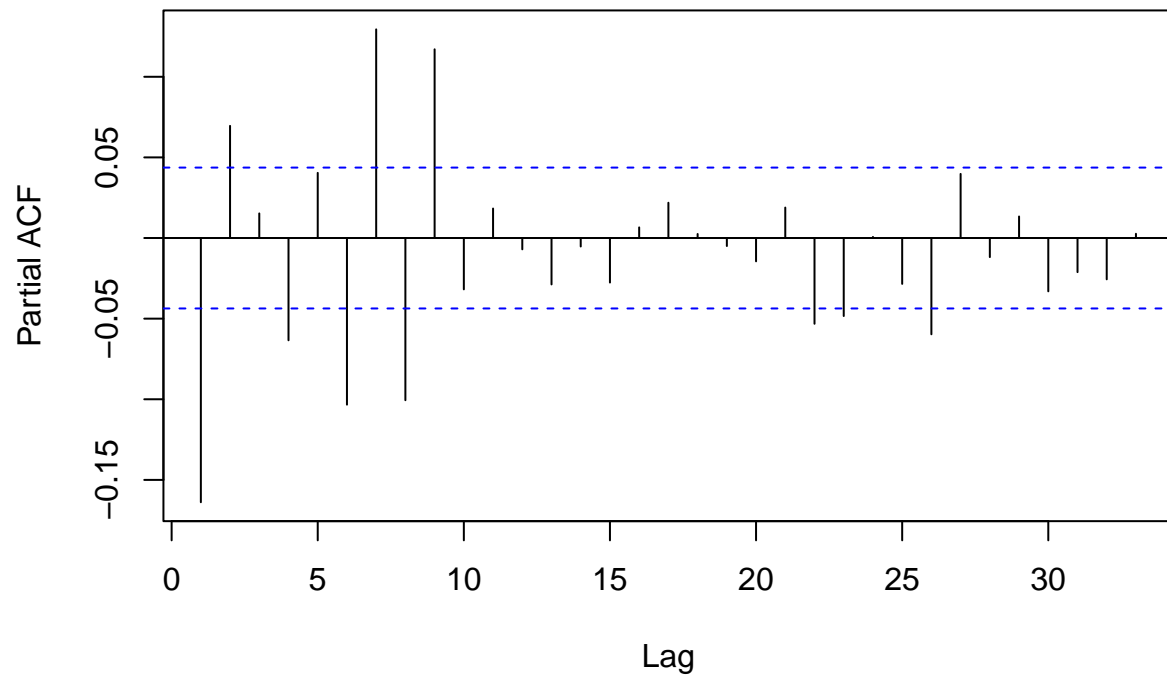
Autocorrelation Function: S&P500 Log-Dif



Temos uma série com ACF que aparece truncada na segunda defasagem, indicando um componente MA(2).

```
pacf(  
  s1,  
  main = "Partial Autocorrelation Function: S&P500 Log-Dif"  
)
```

Partial Autocorrelation Function: S&P500 Log-Dif



Temos uma série com PACF truncada na segunda defasagem, indicando um componente AR(2).

CrITÉrios de Informação

Podemos verificar a decisão através dos critérios de informação. Utilizaremos a função `auto.arima`, que, de forma automática, calcula todos os modelos dentro de limites dados, calcula seus critérios de informação e escolhe o modelo que minimiza. Aqui, escolhemos os limites de 5 lags para cada parte do modelo (AR e MA). Primeiro, minimizaremos o BIC.

```
model_auto_bic <- auto.arima(  
  max.p = 5,  
  max.q = 5,  
  s1,  
  seasonal = FALSE,  
  stepwise = FALSE,  
  approximation = FALSE,  
  ic = "bic"  
)
```

```
tidy(model_auto_bic) %>%  
  kable(  
    col.names = c(  
      "Term",  
      "Estimate",  
      "Standard Error"
```

```

),
caption = "BIC Minimizer Model"
)

```

Table 3: BIC Minimizer Model

Term	Estimate	Standard Error
ar1	-1.7401133	0.0257552
ar2	-0.8770843	0.0258495
ma1	1.6310849	0.0354612
ma2	0.7419635	0.0358254

Agora, pelo critério do AIC.

```

model_auto_aic <- auto.arima(
  max.p = 5,
  max.q = 5,
  s1,
  seasonal = FALSE,
  stepwise = FALSE,
  approximation = FALSE,
  ic = "aic"
)

```

```

tidy(model_auto_aic) %>%
  kable(
    col.names = c(
      "Term",
      "Estimate",
      "Standard Error"
    ),
    caption = "AIC Minimizer Model"
  )

```

Table 4: AIC Minimizer Model

Term	Estimate	Standard Error
ar1	-1.7390897	0.0259024
ar2	-0.8760746	0.0260392
ma1	1.6294213	0.0356304
ma2	0.7402674	0.0360618
intercept	0.0004274	0.0002373

Concluimos que, de fato, a escolha do modelo ARMA(2,2) para a média da série da Log-diferença é a ideal nesse caso, sustentado tanto pela análise das ACF e PACF quanto pelos critérios de informação.

ARMA(2,2)

Implementando o modelo escolhido ARMA(2,2).


```
m202 <- arima(s1, order = c(2, 0, 2))
```

```
stargazer(
  m202,
  header = FALSE,
  float = FALSE,
  dep.var.labels = "$\\Delta \\log (P)$"
)
```

	<i>Dependent variable:</i>
	$\Delta \log(P)$
ar1	-1.739*** (0.026)
ar2	-0.876*** (0.026)
ma1	1.629*** (0.036)
ma2	0.740*** (0.036)
intercept	0.0004* (0.0002)
Observations	2,013
Log Likelihood	6,154.515
σ^2	0.0001
Akaike Inf. Crit.	-12,297.030
<i>Note:</i> *p<0.1; **p<0.05; ***p<0.01	

Análise de Resíduos

Série dos resíduos

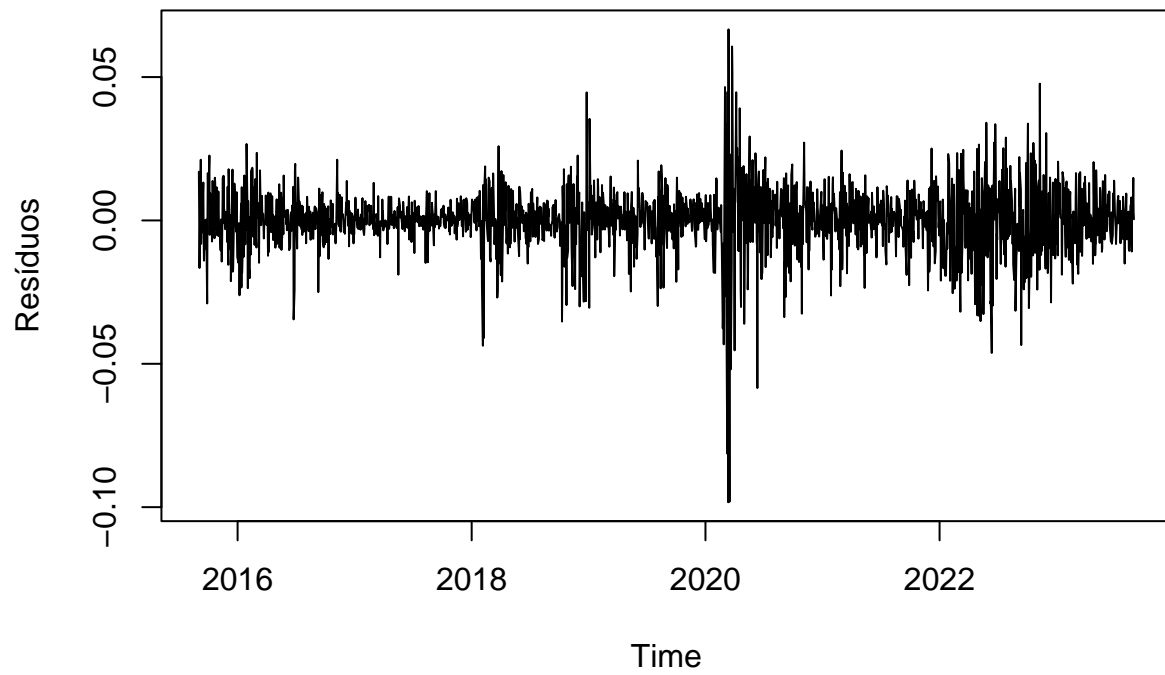
Primeiro, coletamos a série dos resíduos.

```
res_m202 <- residuals(m202)
```

Visualizando.

```
datas <- index(dif_log_close)
plot(res_m202,
  xlab = "Time",
  ylab = "Resíduos",
  x = datas,
  type = "l",
  main = "Resíduos da ARMA(2,2)")
```

Resíduos da ARMA(2,2)

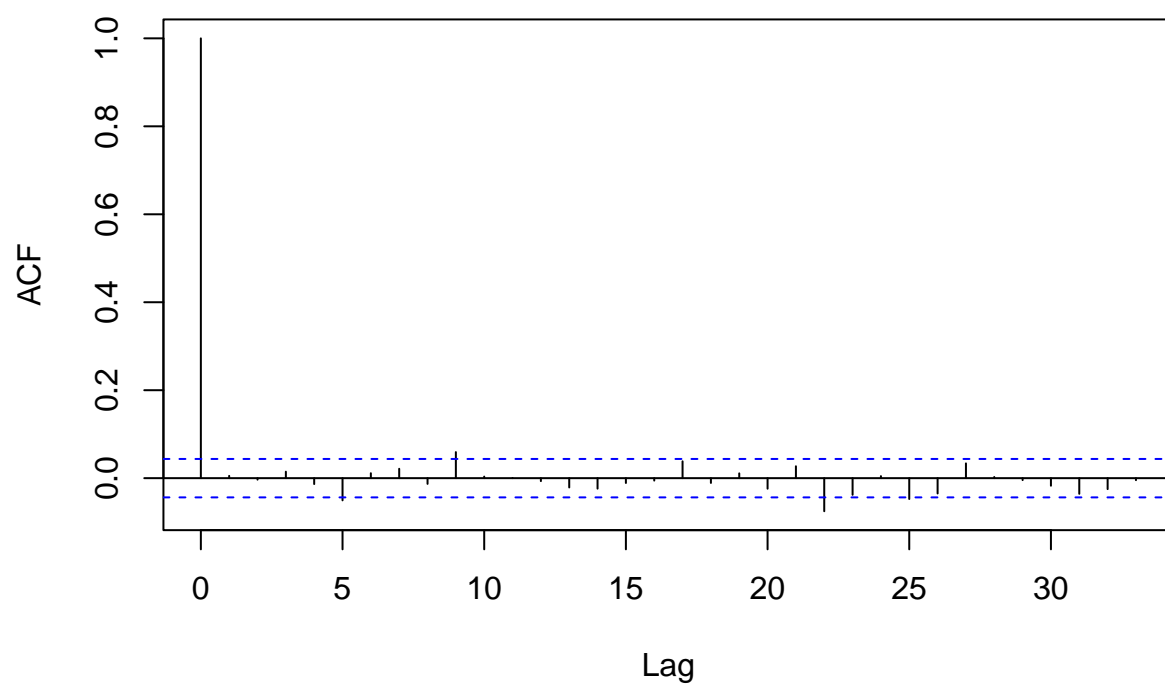


Autocorrelação dos resíduos

Quanto à ACF:

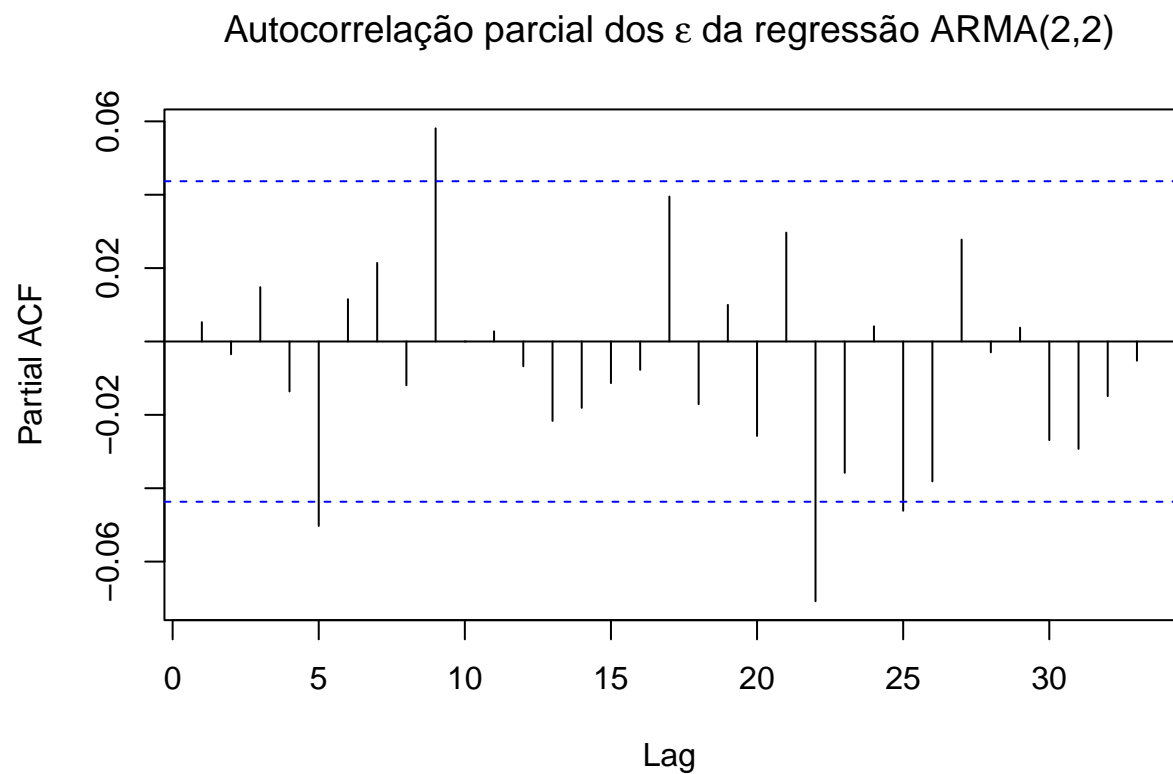
```
acf(res_m202,  
    main = TeX(r"(Autocorrelação dos  $\epsilon$  da regressão ARMA(2,2))"))
```

Autocorrelação dos ε da regressão ARMA(2,2)



Quanto à PACF:

```
pacf(res_m202,  
      main = TeX(r"(Autocorrelação parcial dos  $\varepsilon$  da regressão ARMA(2,2))"))
```



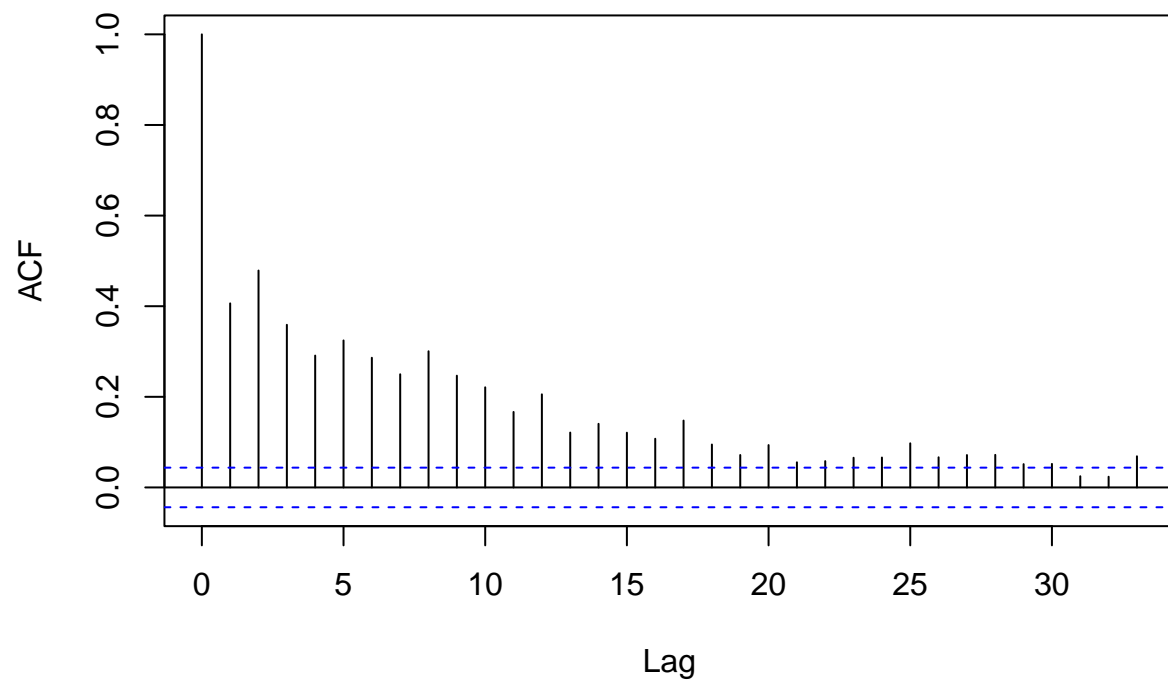
Podemos ver que, como esperado, os resíduos não exibem padrão claro de autocorrelação serial. Mas será que isso vale também para os quadrados dos resíduos, que estimam sua variância?

Autocorrelação dos quadrados dos resíduos

Quanto à ACF:

```
acf(res_m202^2,  
    main = TeX(r"(Autocorrelação dos  $\varepsilon^2$  da regressão ARMA(2,2))")  
)
```

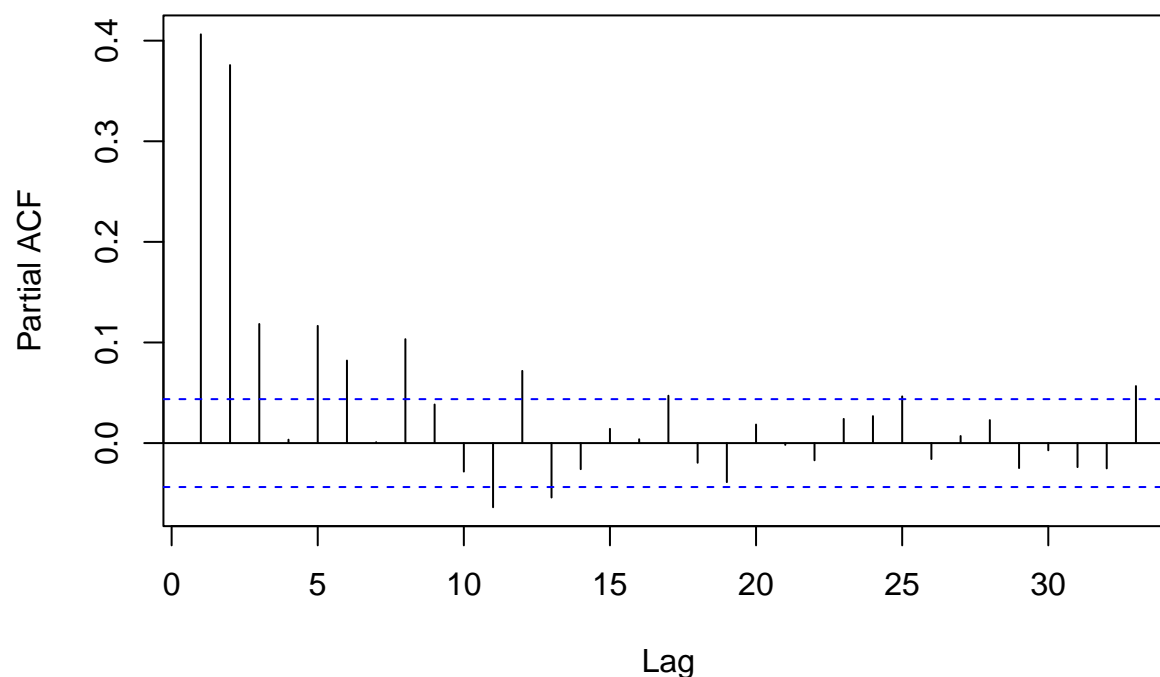
Autocorrelação dos ε^2 da regressão ARMA(2,2)



E quanto à PACF:

```
pacf(res_m202^2,  
      main = TeX(r"(Autocorrelação parcial dos  $\varepsilon^2$  da regressão ARMA(2,2))"))
```

Autocorrelação parcial dos ε^2 da regressão ARMA(2,2)



Vemos que quando analisamos os resíduos ao quadrado, encontramos a presença de autocorrelação serial, indicando que a série não é homoscedástica. Vamos realizar um teste mais robusto de autocorrelação na série.

Teste de Ljung-Box

Para a série de resíduos:

```
box_teste <- Box.test(res_m202, lag = 250, type = "Ljung-Box")
tidy(box_teste) %>%
  kable(
    digits = 5,
    col.names = c(
      "Estatística de Teste",
      "P Value",
      "Lags",
      "Method"
    )
  )
```

Estatística de Teste	P Value	Lags	Method
216.9475	0.93558	250	Box-Ljung test

Assim, não podemos rejeitar a hipótese nula de não haver autocorrelação serial nos resíduos, como era esperado olhando para as funções.

Agora, para a série de resíduos ao quadrado:

```
res2_m202 <- res_m202^2
box_teste <- Box.test(res2_m202, lag = 250, type = "Ljung-Box")
tidy(box_teste) %>%
  kable(
    digits = 5,
    col.names = c(
      "Estatística de Teste",
      "P Value",
      "Lags",
      "Method"
    )
  )
```

Estatística de Teste	P Value	Lags	Method
2751.789	0	250	Box-Ljung test

Nesse caso, ao fazer o teste de Ljung-Box encontramos um p-valor estatisticamente igual a zero. Rejeitamos portanto a hipótese nula de ausência de autocorrelação serial no quadrado dos resíduos. Concluimos que embora não haja correlação entre os resíduos, há correlação serial entre seu quadrados resíduos. Logo, faz-se necessária uma modelagem para a variância.

GARCH

Modelo

Há suporte na literatura para o uso do modelo GARCH(1,1) para modelar a volatilidade de ativos financeiros, vide (Hansen and Lunde 2005). Utilizaremos então o modelo ARMA(2,2) encontrado no passo anterior para modelar a média, e um GARCH(1,1) para modelar a variância.

```
modelo_garch <- garchFit(
  formula = ~arma(2,2) + garch(1,1),
  data = s1,
  trace = FALSE)
```

Temos o seguinte modelo:

$$\Delta \log(P_t) = c + \phi_1 \Delta \log(P_{t-1}) + \phi_2 \Delta \log(P_{t-2}) + \psi_1 \epsilon_{t-1} + \psi_2 \epsilon_{t-2} + \epsilon_t$$

$$\epsilon_t | \mathcal{I}_t \sim \mathcal{N}(0, \sigma_t^2),$$

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + \eta_t$$

```

stargazer(
  modelo_garch,
  header = FALSE,
  float = FALSE,
  dep.var.labels = "$\\Delta \\log (P)$",
  digits = 3,
  covariate.labels=c(
    "c",
    "$\\phi_1$",
    "$\\phi_2$",
    "$\\psi_1$",
    "$\\psi_2$",
    "$\\omega$",
    "$\\alpha_1$",
    "$\\beta_1$"
  ))

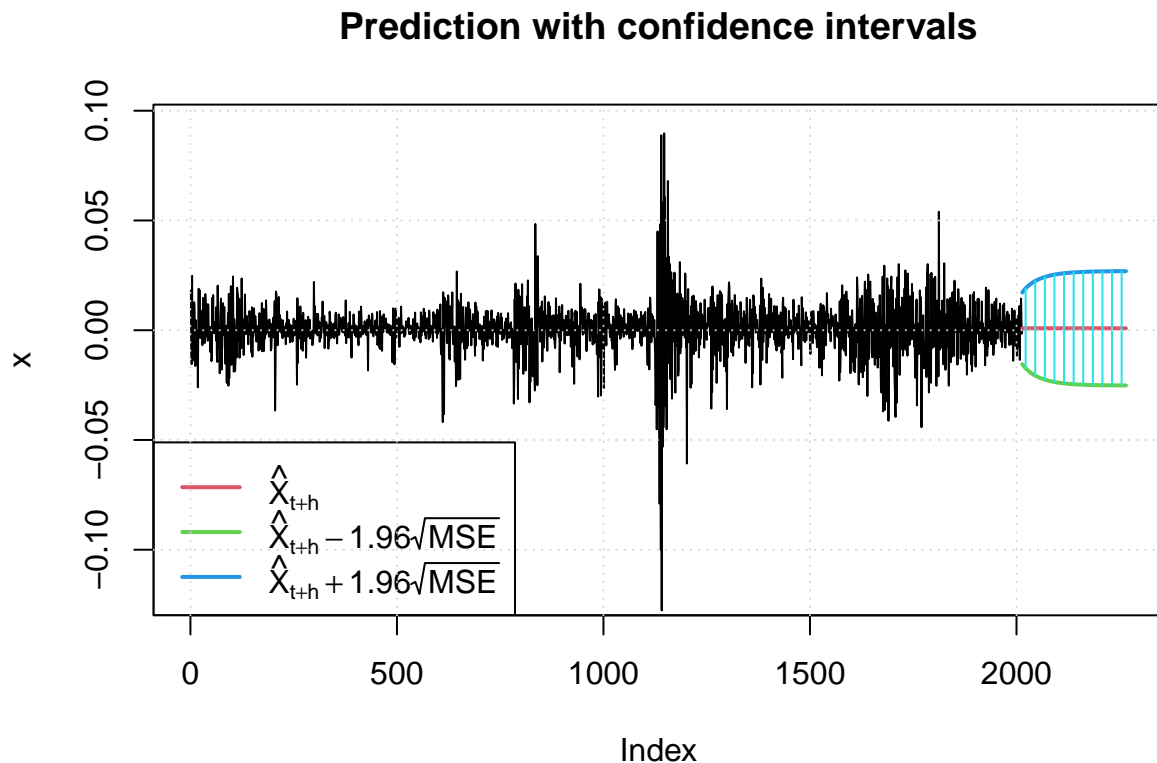
```

	<i>Dependent variable:</i>
	$\Delta \log(P)$
c	0.002*** (0.0005)
ϕ_1	-0.952*** (0.033)
ϕ_2	-0.903*** (0.034)
ψ_1	0.952*** (0.041)
ψ_2	0.902*** (0.035)
ω	0.00000*** (0.00000)
α_1	0.202*** (0.025)
β_1	0.777*** (0.023)
Observations	2,013
Log Likelihood	-6,649.232
Akaike Inf. Crit.	-6.598
Bayesian Inf. Crit.	-6.576
Note:	*p<0.1; **p<0.05; ***p<0.01

Previsão

Agora, podemos aplicar uma previsão utilizando nosso modelo levando em conta a variância condicional. Primeiro, temos uma previsão da série log-dif. Note que realizamos o forecasting com um ano-padrão de 252 dias úteis.

```
garch_predict = predict(  
  modelo_garch,  
  n.ahead = 252,  
  nx = length(s1),  
  plot = TRUE)
```

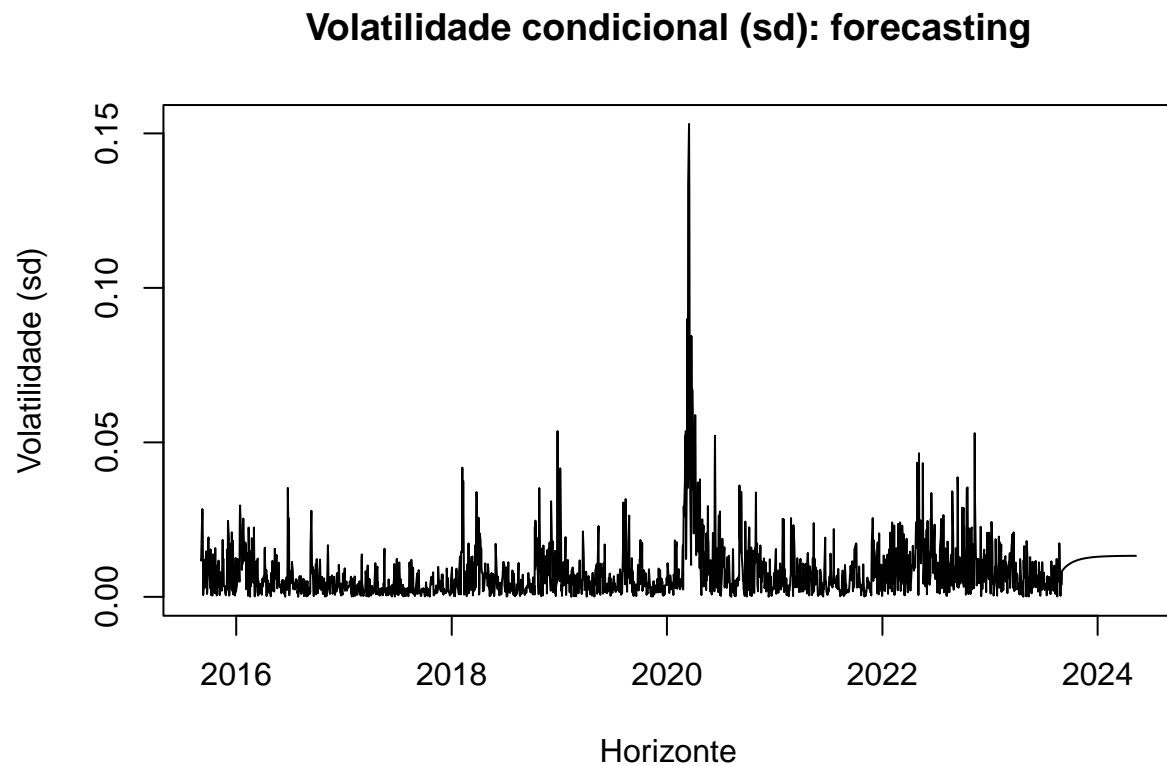


Em seguida, podemos exibir a previsão da volatilidade, representada pelo Desvio Padrão.

```
garch_predict_xts <- xts(  
  garch_predict$standardDeviation,  
  order.by = seq(as.Date("2023-09-02"), by = "days", length.out = 252))  
  
var_s1 <- rollapply(dif_log_close, width = 2, FUN = sd, fill = NA)[-1]  
  
var_predict <- rbind(var_s1, garch_predict_xts)
```

```
plot(  
  ts(var_predict),  
  type = "l",
```

```
x = index(var_predict),  
xlab = "Horizonte",  
ylab = "Volatilidade (sd)",  
main = "Volatilidade condicional (sd): forecasting"  
)
```



References

Hansen, Peter R., and Asger Lunde. 2005. "A Forecast Comparison of Volatility Models: Does Anything Beat a GARCH(1,1)?" *Journal of Applied Econometrics* 20 (7): 873–89. <https://doi.org/10.1002/jae.800>.