GARCH em Ações

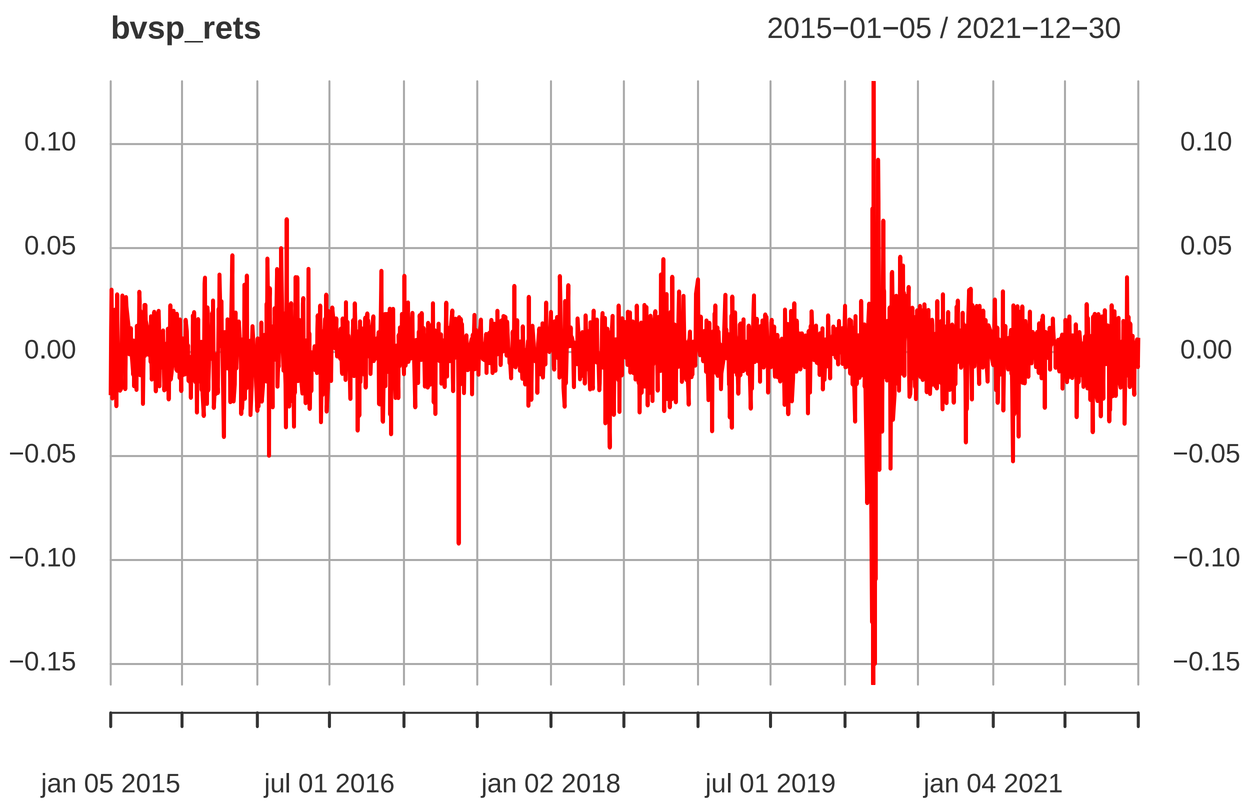
Wilson Freitas

```r
library(fGarch)
library(xts)
library(quantmod)
library(purrr)
library(readr)
library(stringr)
library(dplyr)
library(ggplot2)
library(forcats)
```

```r
bvsp <- getSymbols("^BVSP",
  auto.assign = FALSE,
  from = "2015-01-01", to = "2021-12-31"
) |> Ad()

bvsp_rets <- log(bvsp) |>
  diff() |>
  na.omit()
```

```r
plot(bvsp_rets, col = "red")
```




```r
(bvsp_rets) |> Box.test(lag = 10)
```

```
##
##  Box-Pierce test
##
## data:  (bvsp_rets)
## X-squared = 84.183, df = 10, p-value = 7.561e-14
```

```r
(bvsp_rets) |> Box.test(lag = 15)
```

```
##
##  Box-Pierce test
##
## data:  (bvsp_rets)
## X-squared = 95.435, df = 15, p-value = 9.504e-14
```

```r
(bvsp_rets) |> Box.test(lag = 20)
```

```
##
##  Box-Pierce test
##
## data:  (bvsp_rets)
## X-squared = 99.425, df = 20, p-value = 1.596e-12
```

```r
(bvsp_rets ** 2) |> Box.test(lag = 10)
```

```
##
##  Box-Pierce test
##
## data:  (bvsp_rets^2)
## X-squared = 2100.3, df = 10, p-value < 2.2e-16
```

```r
(bvsp_rets ** 2) |> Box.test(lag = 15)
```

```
##
##  Box-Pierce test
##
## data:  (bvsp_rets^2)
## X-squared = 2214, df = 15, p-value < 2.2e-16
```

```r
(bvsp_rets ** 2) |> Box.test(lag = 20)
```

```
##
##  Box-Pierce test
##
## data:  (bvsp_rets^2)
## X-squared = 2236, df = 20, p-value < 2.2e-16
```

Modelo GARCH

$$r_t = \sqrt{h_t} e_t$$

onde e_t é uma variável aleatória IID. Aqui vamos utilizar a distribuição Normal, mas podemos utilizar outras distribuições como t-Student, por exemplo.

h_t é o processo da variância e possui componente autoregressiva e dependência de r_t^2.

$$h_t = \omega + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{i=1}^{q} \beta_i h_{t-i}$$

GARCH(1,1)

Vamos fazer o ajuste da série de retornos do IBOVESPA para o GARCH(1,1)

$$h_t = \omega + \alpha_1 r_{t-1}^2 + \beta_1 h_{t-1}$$

```r
mod <- garchFit(~ garch(1, 1), data = bvsp_rets, trace = FALSE)
```

```r
summary(mod)
```

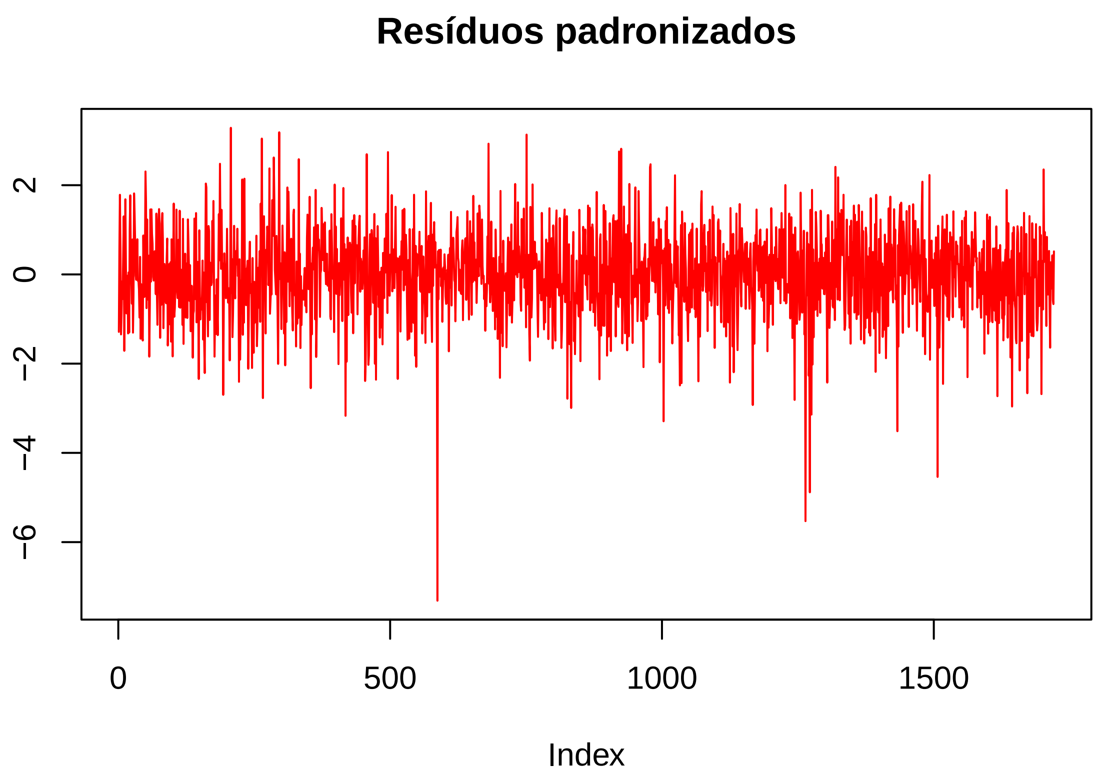
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = bvsp_rets, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x00000000253c0280>
##  [data = bvsp_rets]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 7.2747e-04  1.2087e-05  9.3993e-02  8.5250e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     7.275e-04   3.232e-04    2.251 0.024398 *
## omega  1.209e-05   3.139e-06    3.851 0.000118 ***
## alpha1 9.399e-02   1.583e-02    5.938 2.88e-09 ***
## beta1  8.525e-01   2.450e-02   34.791  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  4865.291    normalized:  2.827014
##
## Description:
##  Sat Mar 26 16:44:12 2022 by user: wilso
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  595.5008  0
##  Shapiro-Wilk Test  R    W      0.9795552 5.845924e-15
##  Ljung-Box Test     R    Q(10)  8.083516  0.6206794
##  Ljung-Box Test     R    Q(15)  12.58065  0.6346531
##  Ljung-Box Test     R    Q(20)  13.83581  0.8387222
##  Ljung-Box Test     R^2  Q(10)  6.110542  0.805892
##  Ljung-Box Test     R^2  Q(15)  8.209119  0.9151253
##  Ljung-Box Test     R^2  Q(20)  11.10563  0.9434371
##  LM Arch Test       R    TR^2   7.455796  0.8260831
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -5.649379 -5.636711 -5.649390 -5.644692
```

Os resíduos

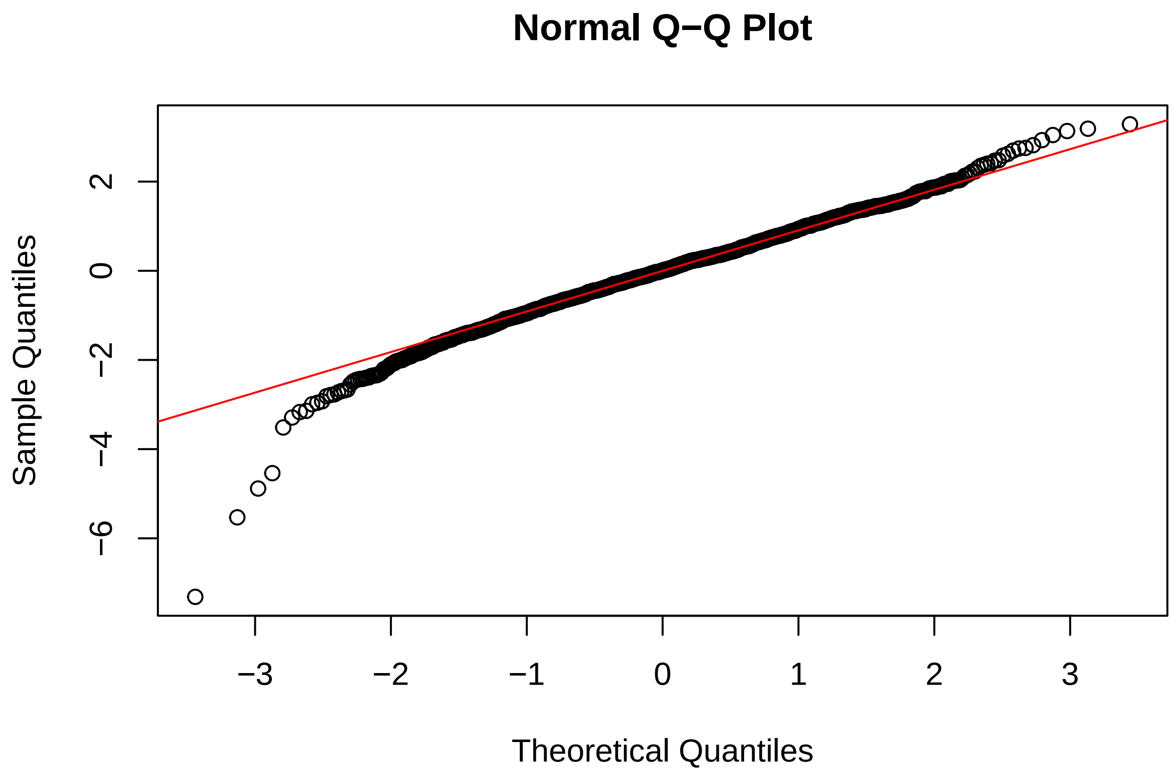
Os resíduos são padronizados, pois:

$$e_t = \frac{r_t}{\sqrt{h_t}}$$

```
plot(residuals(mod, standardize = TRUE),
  type = "l", col = "red",
  main = "Resíduos padronizados", ylab = ""
)
```




```
residuals(mod, standardize = TRUE) |> qqnorm()
residuals(mod, standardize = TRUE) |> qqline(col = "red")
```



Fatos Estilizados do GARCH

Fatos estilizados são *coisas* que vêm de graça com o GARCH.

Curtose

Bollerslev (1986) e Teräsvirta (1999) demostram que a curtose de um modelo GARCH(p,q) é superior a 3, curtose da distribuição Normal.

Reversão a Média

É possível reescrever o GARCH como um ARMA, dessa maneira o processo de volatilidade é estacionário, assim a volatilidade evolui em torno de um valor médio.

Volatility Clusters

O GARCH captura muito bem as propriedades autoregressivas da série. Por este motivo, os resíduos de um GARCH bem ajustado não apresentam autocorrelação significativa e nem a série de quadrados dos resíduos. Isso acontece pelo parâmetro $beta_1$ (para o GARCH(1,1), do nosso exemplo) apresentar um valor de 0.85, que dá um grande peso para h_{t-1} que gera uma grande contribuição para h_t. Assim, grandes variâncias produzem novas grandes variâncias, e o mesmo acontece com pequenas variâncias.

IBOVESPA

Vamos calcular o GARCH para todas as ações que compõem o IBOVESPA.

A composição da carteira do IBOVESPA pode ser obtido no site da B3.

```
symbols <- read_delim("IBOVDia_21-03-22.csv",
  skip = 1,
  delim = ";",
  locale = locale(encoding = "latin1"),
) |>
  filter(!is.na(`Ação`)) |>
  pull(`Código`) |>
  paste0(".SA")
```

Pegar dados 3 anos de dados

```
series <- map(symbols, function(x) {
  x <- getSymbols(x,
    auto.assign = FALSE,
    from = "2019-01-01",
    to = "2021-12-31"
  )
  Ad(x)
})

series <- set_names(series, symbols)
```

Calculando os parâmetros dos modelos

```
models <- map(symbols, function(x) {
  data <- series[[x]]
  rets <- log(data) |>
    diff() |>
    na.omit()
  garchFit(data = rets, trace = FALSE)
})
models <- set_names(models, symbols)
```

```
params <- map_dfr(symbols, function(x) {
  mod <- models[[x]]
  params <- coef(mod)
  sv <- sqrt(var(mod@data, na.rm = FALSE) * 252) |> as.numeric()
  v0 <- sum(params[-1] * c(1, tail(mod@data, 1)^2, tail(mod@h.t, 1))) * 252
  tibble(
    symbol = x,
    length = length(mod@data),
    omega = params["omega"],
    alpha1 = params["alpha1"],
    beta1 = params["beta1"],
    check = alpha1 + beta1 < 1,
    instant_volatility = 100 * sqrt(v0),
    sample_volatility = 100 * sv
```

```
  )
})
```

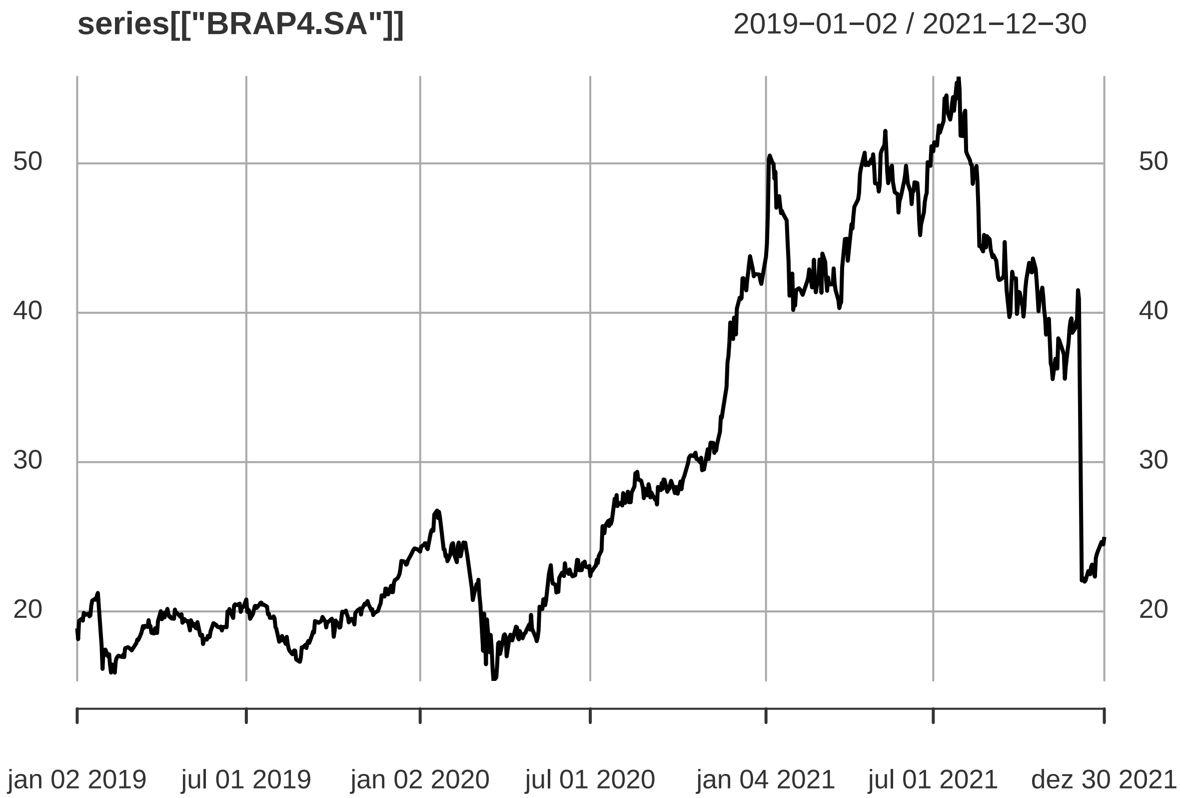
```r
params
```

```
## # A tibble: 91 x 8
##    symbol   length   omega alpha1  beta1 check instant_volatil~ sample_volatili~
##    <chr>     <int>   <dbl>  <dbl>  <dbl> <lgl>            <dbl>            <dbl>
##  1 RRRP3.SA    277 6.87e-4 0.189  0.145  TRUE              46.4             50.9
##  2 ALPA4.SA    742 1.18e-4 0.137  0.698  TRUE              36.6             47.5
##  3 ABEV3.SA    742 1.05e-4 0.209  0.579  TRUE              25.3             35.5
##  4 AMER3.SA    742 5.39e-5 0.0743 0.888  TRUE              57.4             63.2
##  5 ASAI3.SA    209 1.45e-5 0.0728 0.887  TRUE              33.9             30.0
##  6 AZUL4.SA    742 4.08e-5 0.139  0.846  TRUE              67.1             77.2
##  7 B3SA3.SA    742 3.10e-5 0.0904 0.863  TRUE              33.7             43.8
##  8 BIDI11.~    603 1.10e-3 0.481  0.0622 TRUE              54.6             73.5
##  9 BPAN4.SA    742 2.00e-4 0.359  0.589  TRUE              45.7             75.2
## 10 BBSE3.SA    742 1.69e-5 0.0893 0.853  TRUE              20.4             30.5
## # ... with 81 more rows
```

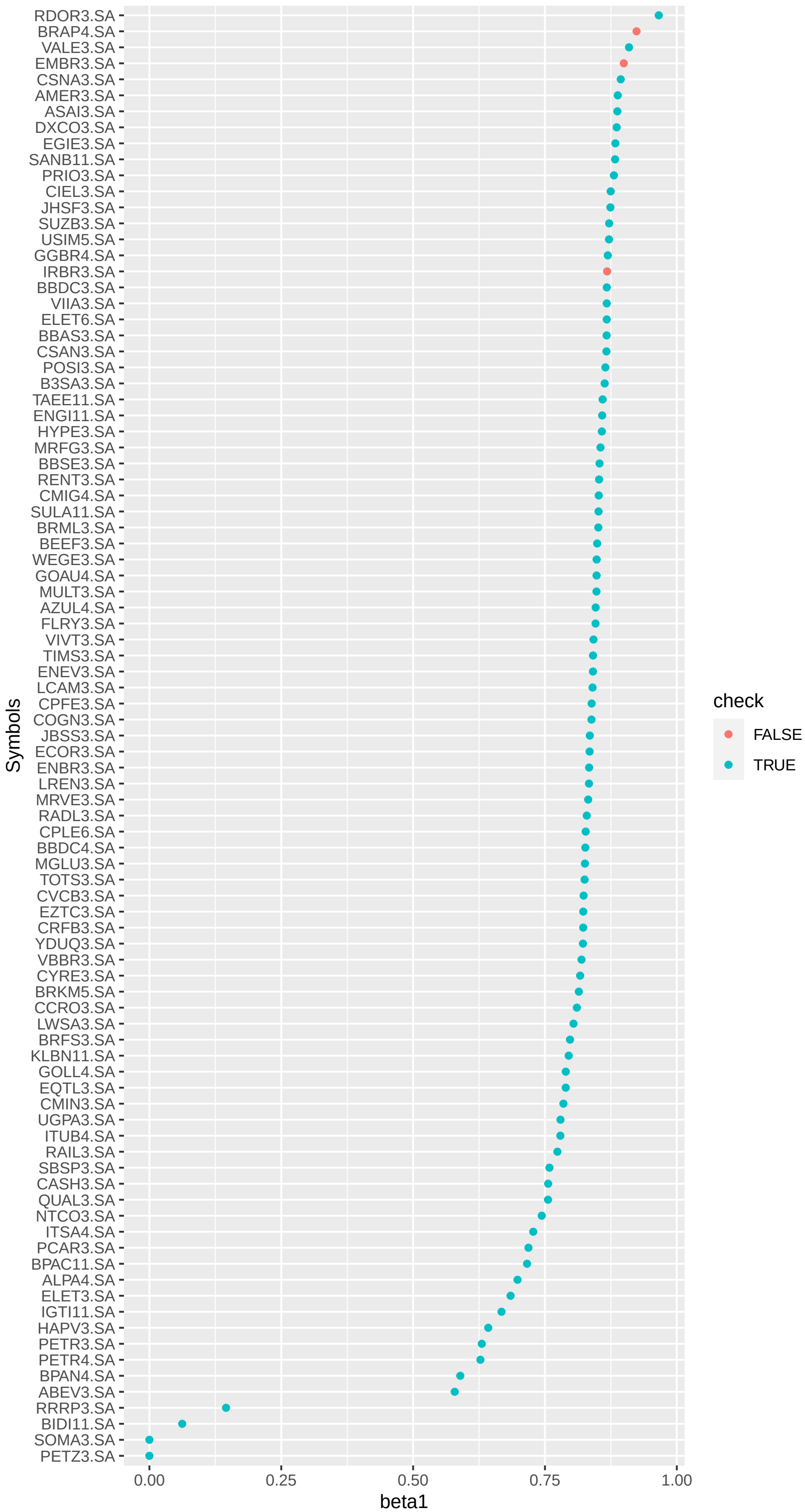
```r
params |> filter(!check)
```

```
## # A tibble: 3 x 8
##   symbol   length     omega alpha1 beta1 check instant_volatil~ sample_volatili~
##   <chr>     <int>     <dbl>  <dbl> <dbl> <lgl>            <dbl>            <dbl>
## 1 BRAP4.SA    742   1.24e-9  0.143 0.924 FALSE            195.              55.8
## 2 EMBR3.SA    742   7.30e-6  0.112 0.899 FALSE             70.1             57.6
## 3 IRBR3.SA    742   1.79e-5  0.132 0.868 FALSE             42.4             69.5
```

```r
plot(series[["BRAP4.SA"]])
```




```r
params |>
  ggplot(aes(y = fct_reorder(symbol, beta1), x = beta1, colour = check)) +
  geom_point() +
  labs(y = "Symbols")
```


RDOR3.SA
BRAP4.SA
VALE3.SA
EMBR3.SA
CSNA3.SA
AMER3.SA
ASAI3.SA
DXCO3.SA
EGIE3.SA
SANB11.SA
PRIO3.SA
CIEL3.SA
JHSF3.SA
SUZB3.SA
USIM5.SA
GGBR4.SA
IRBR3.SA
BBDC3.SA
VIIA3.SA
ELET6.SA
BBAS3.SA
CSAN3.SA
POSI3.SA
B3SA3.SA
TAEE11.SA
ENGI11.SA
HYPE3.SA
MRFG3.SA
BBSE3.SA
RENT3.SA
CMIG4.SA
SULA11.SA
BRML3.SA
BEEF3.SA
WEGE3.SA
GOAU4.SA
MULT3.SA
AZUL4.SA
FLRY3.SA
VIVT3.SA
TIMS3.SA
ENEV3.SA
LCAM3.SA
CPFE3.SA
COGN3.SA
JBSS3.SA
ECOR3.SA
ENBR3.SA
LREN3.SA
MRVE3.SA
RADL3.SA
CPLE6.SA
BBDC4.SA
MGLU3.SA
TOTS3.SA
CVCB3.SA
EZTC3.SA
CRFB3.SA
YDUQ3.SA
VBBR3.SA
CYRE3.SA
BRKM5.SA
CCRO3.SA
LWSA3.SA
BRFS3.SA
KLBN11.SA
GOLL4.SA
EQTL3.SA
CMIN3.SA
UGPA3.SA
ITUB4.SA
RAIL3.SA
SBSP3.SA
CASH3.SA
QUAL3.SA
NTCO3.SA
ITSA4.SA
PCAR3.SA
BPAC11.SA
ALPA4.SA
ELET3.SA
IGTI11.SA
HAPV3.SA
PETR3.SA
PETR4.SA
BPAN4.SA
ABEV3.SA
RRRP3.SA
BIDI11.SA
SOMA3.SA
PETZ3.SA
Symbols
0.00
0.25
0.50
0.75
1.00
beta1
check
FALSE
TRUE

```r
params |> filter(beta1 < 0.5)
```

```
## # A tibble: 4 x 8
##   symbol   length   omega alpha1   beta1 check instant_volatil~ sample_volatili~
##   <chr>     <int>   <dbl>  <dbl>   <dbl> <lgl>            <dbl>            <dbl>
## 1 RRRP3.SA    277 6.87e-4  0.189 1.45e-1 TRUE              46.4             50.9
## 2 BIDI11.~    603 1.10e-3  0.481 6.22e-2 TRUE              54.6             73.5
## 3 SOMA3.SA    349 7.01e-4  0.113 1   e-8 TRUE              42.9             44.7
## 4 PETZ3.SA    318 5.89e-4  0.208 1   e-8 TRUE              38.6             43.4
```

Volatilidade de Longo Prazo

A variância incondicional é dada por:

$$\mathrm{Var}\, r_t = \frac{\omega}{1 - \alpha_1 - \beta_1}$$

```r
params <- params |>
  mutate(
    lt_variance = omega / (1 - alpha1 - beta1),
    lt_volatility = 100 * sqrt(lt_variance * 252)
  ) |>
  select(-lt_variance)
```

```r
params
```

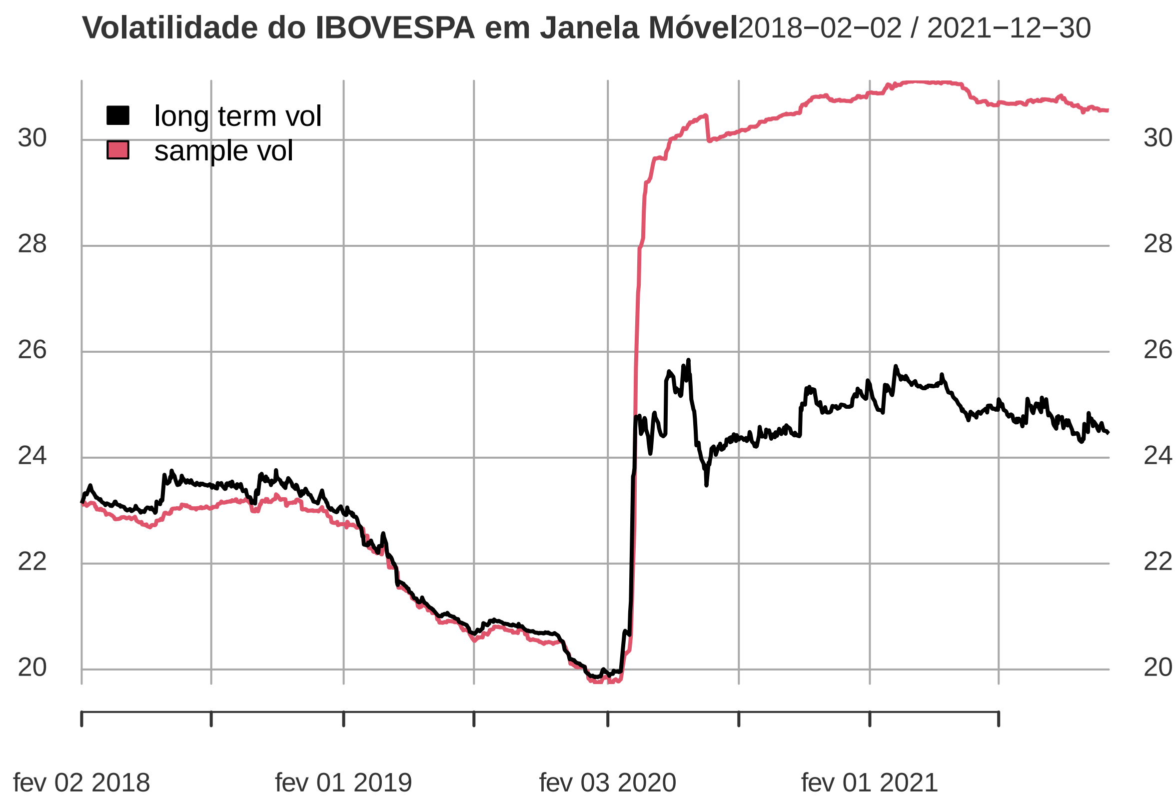
```
## # A tibble: 91 x 9
##    symbol   length   omega alpha1  beta1 check instant_volatil~ sample_volatili~
##    <chr>     <int>   <dbl>  <dbl>  <dbl> <lgl>            <dbl>            <dbl>
##  1 RRRP3.SA    277 6.87e-4 0.189  0.145  TRUE              46.4             50.9
##  2 ALPA4.SA    742 1.18e-4 0.137  0.698  TRUE              36.6             47.5
##  3 ABEV3.SA    742 1.05e-4 0.209  0.579  TRUE              25.3             35.5
##  4 AMER3.SA    742 5.39e-5 0.0743 0.888  TRUE              57.4             63.2
##  5 ASAI3.SA    209 1.45e-5 0.0728 0.887  TRUE              33.9             30.0
##  6 AZUL4.SA    742 4.08e-5 0.139  0.846  TRUE              67.1             77.2
##  7 B3SA3.SA    742 3.10e-5 0.0904 0.863  TRUE              33.7             43.8
##  8 BIDI11.~    603 1.10e-3 0.481  0.0622 TRUE              54.6             73.5
##  9 BPAN4.SA    742 2.00e-4 0.359  0.589  TRUE              45.7             75.2
## 10 BBSE3.SA    742 1.69e-5 0.0893 0.853  TRUE              20.4             30.5
## # ... with 81 more rows, and 1 more variable: lt_volatility <dbl>
```

```r
lt_vols <- rollapply(bvsp_rets, 756, function(x) {
  mod <- garchFit(~ garch(1, 1), data = x, trace = FALSE)
  params <- coef(mod)
  lt_variance <- params["omega"] / (1 - params["alpha1"] - params["beta1"])
  100 * sqrt(lt_variance * 252)
}, align = "right")
```

```r
sample_vols <- rollapply(bvsp_rets, 756, function(x) {
  v <- sqrt(var(x, na.rm = FALSE) * 252) |> as.numeric()
  100 * v
}, align = "right")
```

```r
vols <- merge(lt_vols, sample_vols)
colnames(vols) <- c("long term vol", "sample vol")
```

```r
plot(vols |> na.omit(),
  legend.loc = "topleft",
  main = "Volatilidade do IBOVESPA em Janela Móvel"
)
```



Estrutura a Termo de Volatilidade

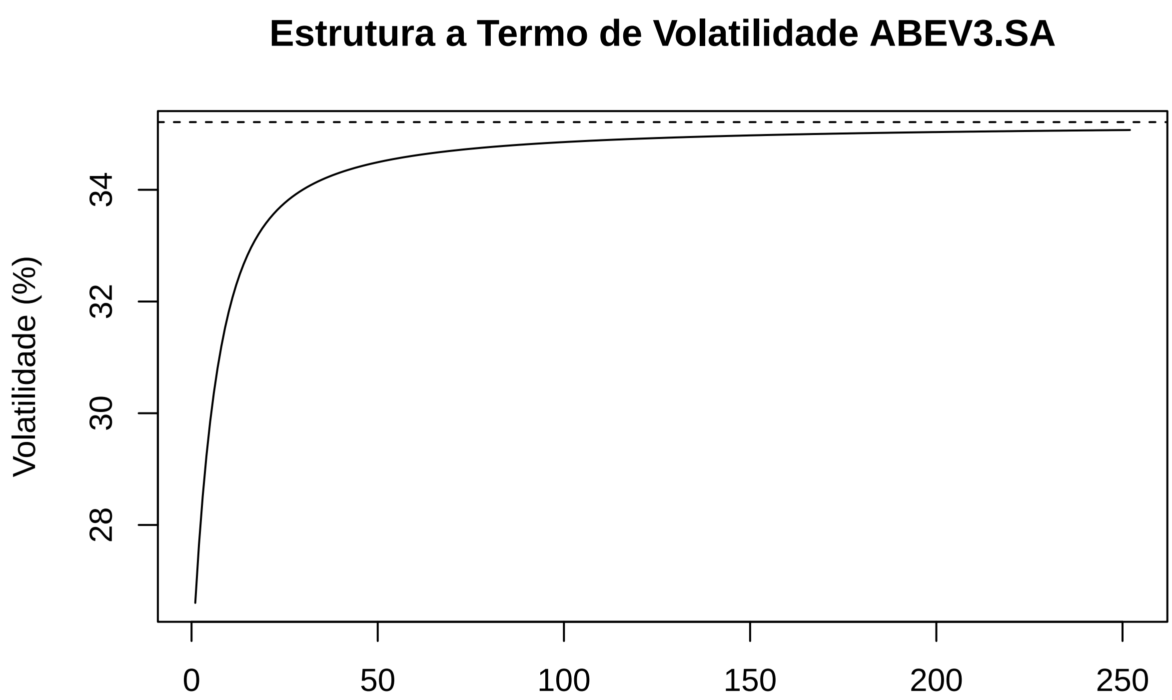
$$a = \ln \frac{1}{\alpha_1 + \beta_1}$$

$$h_T = 252 \left(V_L + \frac{1 - e^{-aT}}{aT} (V(0) - V_L) \right)$$

```r
vts <- function(t, params) {
  a <- log(1 / (params$alpha1 + params$beta1))
  V_L <- ((params$lt_volatility / 100)**2) / 252
  V_0 <- ((params$instant_volatility / 100)**2) / 252
  100 * sqrt(252 * (V_L + (V_0 - V_L) * (1 - exp(-a * t)) / (a * t)))
}
```

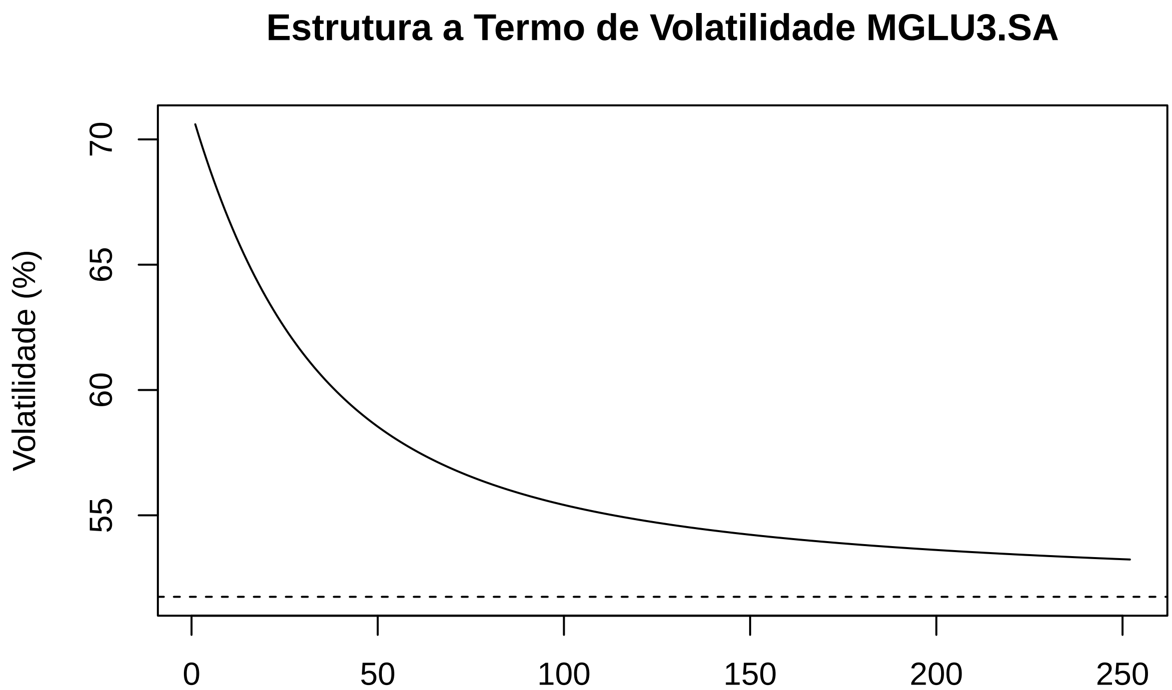
```r
stock_symbol <- "ABEV3.SA"
symbol_params <- params |> filter(symbol == stock_symbol)

t <- 1:252
x <- vts(t, symbol_params)
plot(t, x,
  type = "l", main = paste("Estrutura a Termo de Volatilidade", stock_symbol),
  ylab = "Volatilidade (%)", xlab = ""
)
abline(h = symbol_params$lt_volatility, lty = "dashed")
```




```r
stock_symbol <- "MGLU3.SA"
symbol_params <- params |> filter(symbol == stock_symbol)

t <- 1:252
x <- vts(t, symbol_params)
plot(t, x,
  type = "l", main = paste("Estrutura a Termo de Volatilidade", stock_symbol),
  ylab = "Volatilidade (%)", xlab = "",
  ylim = c(min(symbol_params$lt_volatility, x), max(symbol_params$lt_volatility, x))
)
abline(h = symbol_params$lt_volatility, lty = "dashed")
```


Estrutura a Termo de Volatilidade MGLU3.SA
Volatilidade (%)
70
65
60
55
0
50
100
150
200
250