

GARCH em Ações

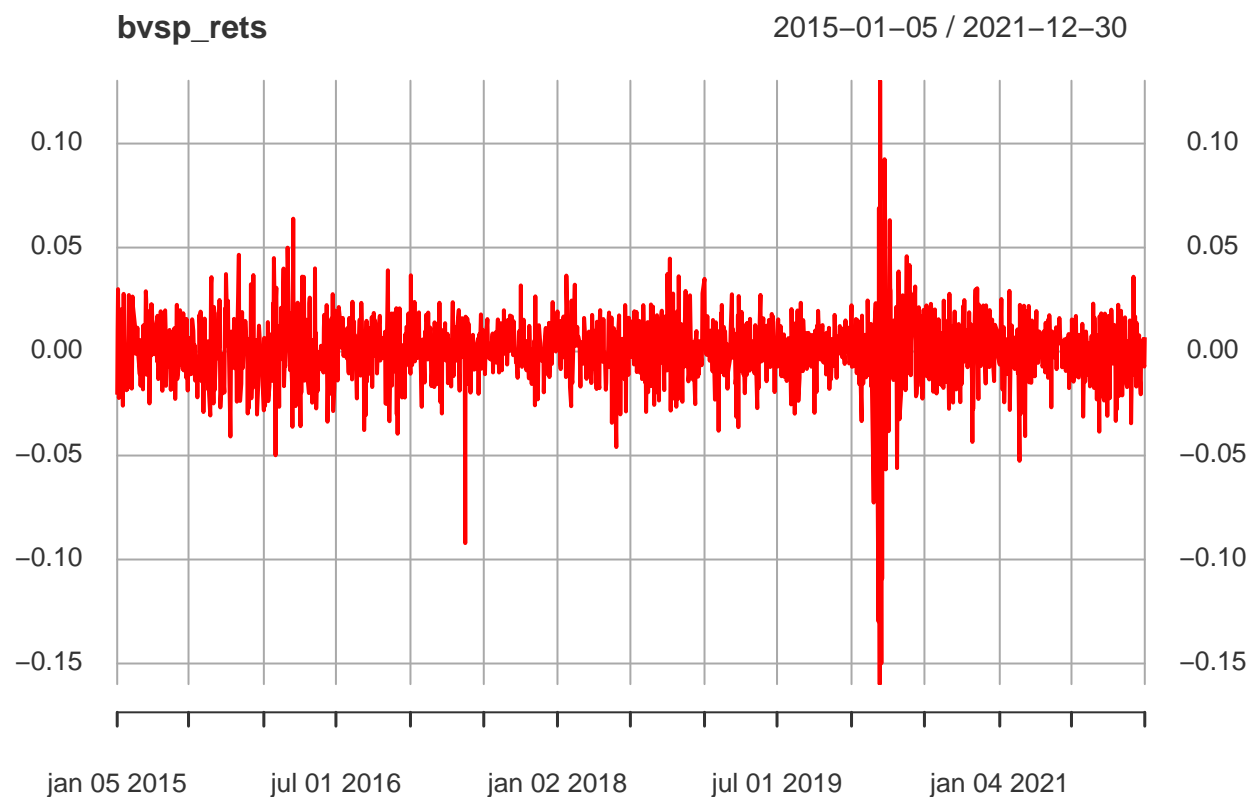
Wilson Freitas

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

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

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

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



Modelo GARCH

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

e

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

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

```
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: 0x0000023e6f355320>
```

```
## [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:
## Wed Mar 23 06:05:48 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
```

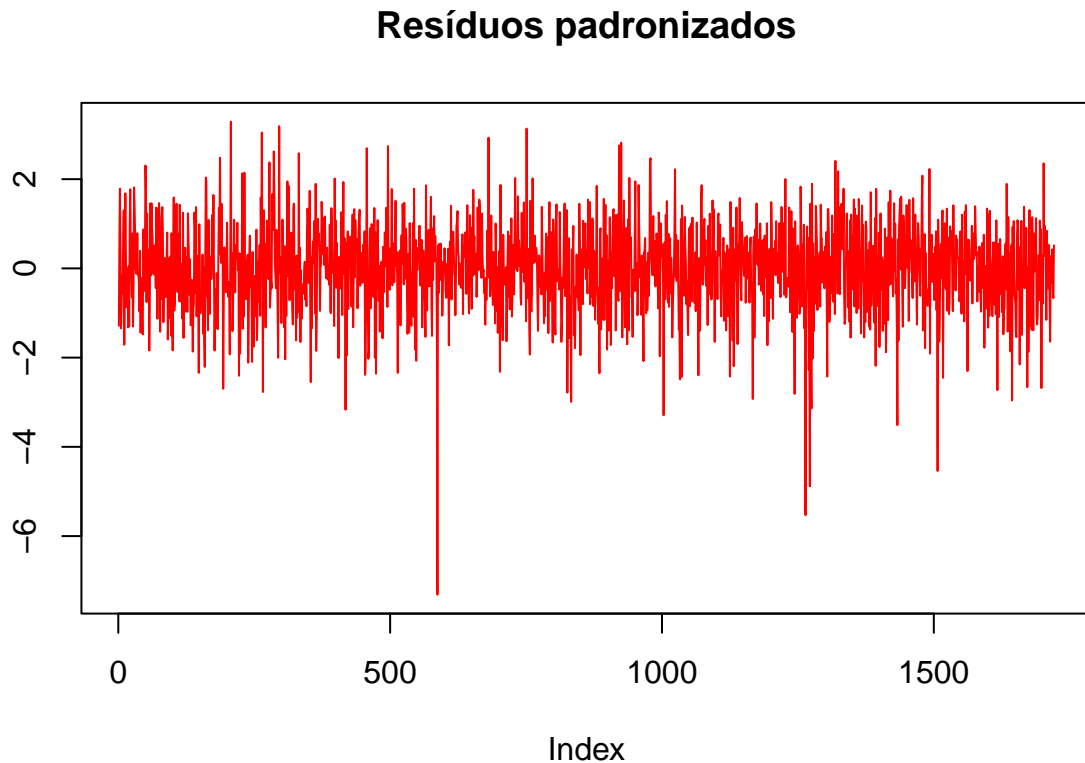
Os resíduos

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 = ""
)
```



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
  )
})

```

params

```

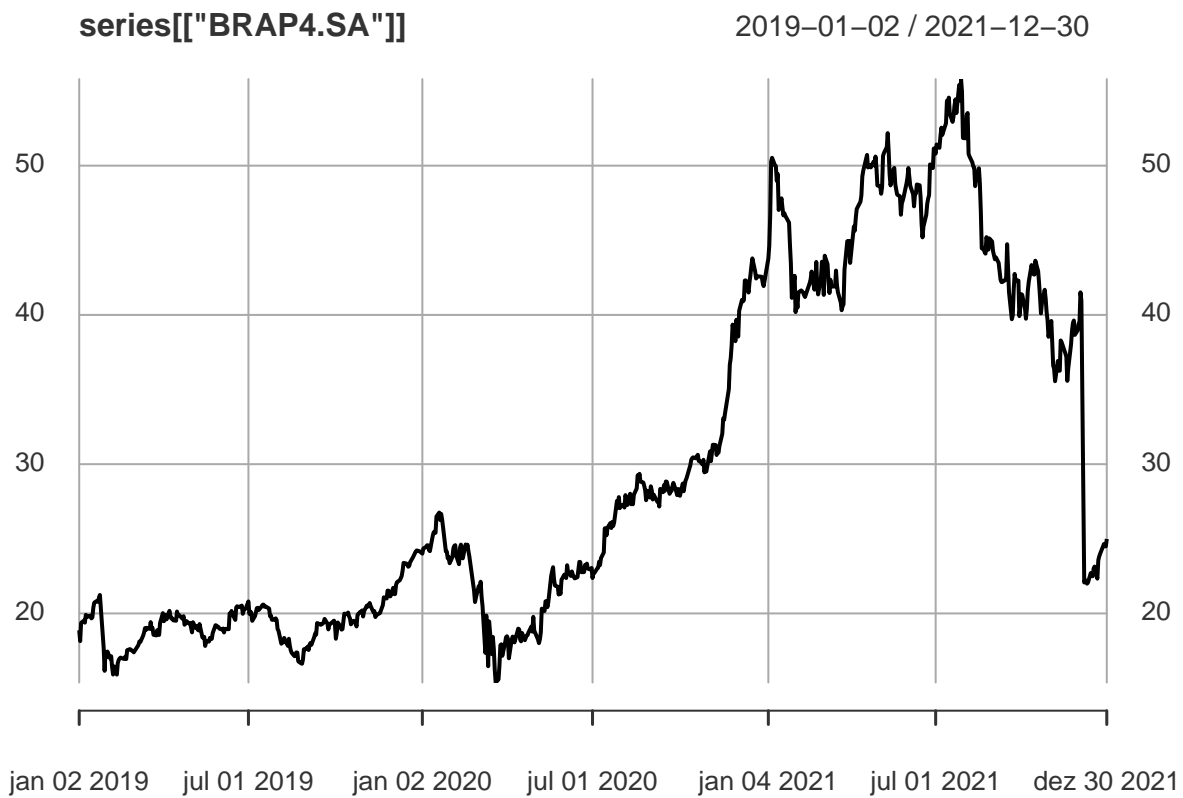
## # A tibble: 91 x 8
##   symbol    length    omega alpha1  beta1 check instant_volatility sample_volatility
##   <chr>      <int>    <dbl> <dbl>  <dbl> <lgl>          <dbl>          <dbl>
## 1 RRRP3.SA    277 0.000687 0.189 0.145 TRUE          46.4          50.9
## 2 ALPA4.SA    742 0.000118 0.137 0.698 TRUE          36.6          47.5
## 3 ABEV3.SA    742 0.000105 0.209 0.579 TRUE          25.3          35.5
## 4 AMER3.SA    742 0.0000539 0.0743 0.888 TRUE          57.4          63.2
## 5 ASAI3.SA    209 0.0000145 0.0728 0.887 TRUE          33.9          30.0
## 6 AZUL4.SA    742 0.0000408 0.139 0.846 TRUE          67.1          77.2
## 7 B3SA3.SA    742 0.0000310 0.0904 0.863 TRUE          33.7          43.8
## 8 BIDI11.SA   603 0.00110 0.481 0.0622 TRUE          54.6          73.5
## 9 BPAN4.SA    742 0.000200 0.359 0.589 TRUE          45.7          75.2
## 10 BBSE3.SA   742 0.0000169 0.0893 0.853 TRUE          20.4          30.5
## # ... with 81 more rows

```

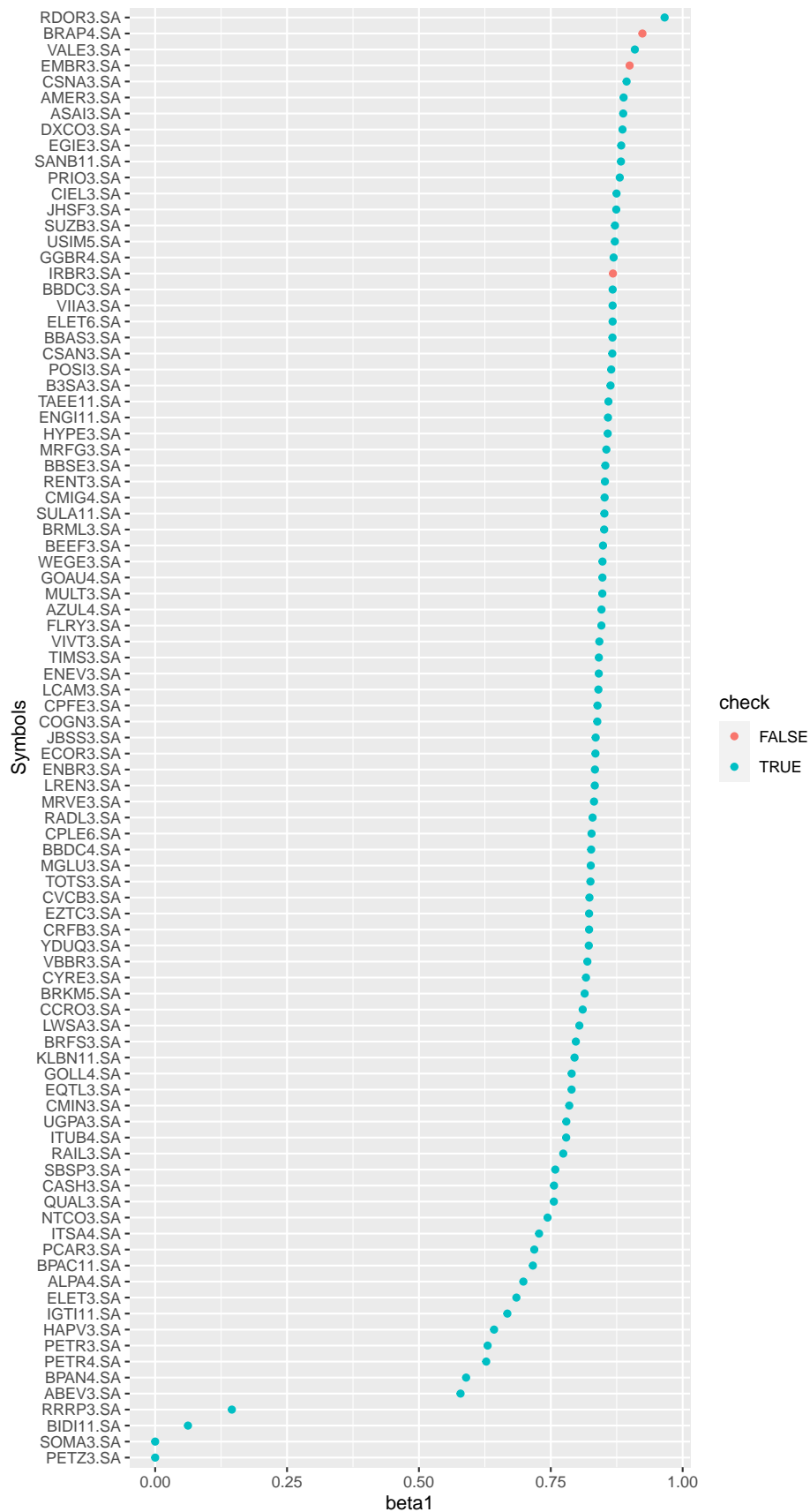
```
params |> filter(!check)
```

```
## # A tibble: 3 x 8
##   symbol    length      omega alpha1 beta1 check instant_volatility sample_volatility
##   <chr>      <int>      <dbl> <dbl> <dbl> <lgl>          <dbl>          <dbl>
## 1 BRAP4.SA    742 0.00000000124  0.143 0.924 FALSE         195.          55.8
## 2 EMBR3.SA    742 0.00000730    0.112 0.899 FALSE         70.1          57.6
## 3 IRBR3.SA    742 0.0000179    0.132 0.868 FALSE         42.4          69.5
```

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



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



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

```
## # A tibble: 4 x 8
##   symbol    length    omega alpha1    beta1 check instant_volatility sample_volatility
##   <chr>      <int>    <dbl> <dbl>    <dbl> <lg1>      <dbl>          <dbl>
## 1 RRRP3.SA    277 0.000687 0.189 0.145    TRUE      46.4          50.9
## 2 BIDI11.SA   603 0.00110 0.481 0.0622    TRUE      54.6          73.5
## 3 SOMA3.SA    349 0.000701 0.113 0.00000001 TRUE      42.9          44.7
## 4 PETZ3.SA    318 0.000589 0.208 0.00000001 TRUE      38.6          43.4
```

Volatilidade de Longo Prazo

A variância incondicional é dada por:

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

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

```
params
```

```
## # A tibble: 91 x 9
##   symbol    length    omega alpha1    beta1 check instant_volatility sample_volatility lt_volatility
##   <chr>      <int>    <dbl> <dbl>    <dbl> <lg1>      <dbl>          <dbl>      <dbl>
## 1 RRRP3.SA    277 0.000687 0.189 0.145    TRUE      46.4          50.9        51.0
## 2 ALPA4.SA    742 0.000118 0.137 0.698    TRUE      36.6          47.5        42.4
## 3 ABEV3.SA    742 0.000105 0.209 0.579    TRUE      25.3          35.5        35.2
## 4 AMER3.SA    742 0.0000539 0.0743 0.888    TRUE      57.4          63.2        60.0
## 5 ASAI3.SA    209 0.0000145 0.0728 0.887    TRUE      33.9          30.0        30.3
## 6 AZUL4.SA    742 0.0000408 0.139 0.846    TRUE      67.1          77.2        83.7
## 7 B3SA3.SA    742 0.0000310 0.0904 0.863    TRUE      33.7          43.8        41.1
## 8 BIDI11.SA   603 0.00110 0.481 0.0622    TRUE      54.6          73.5        77.7
## 9 BPAN4.SA    742 0.000200 0.359 0.589    TRUE      45.7          75.2        99.1
## 10 BBSE3.SA   742 0.0000169 0.0893 0.853    TRUE      20.4          30.5        27.3
## # ... with 81 more rows
```

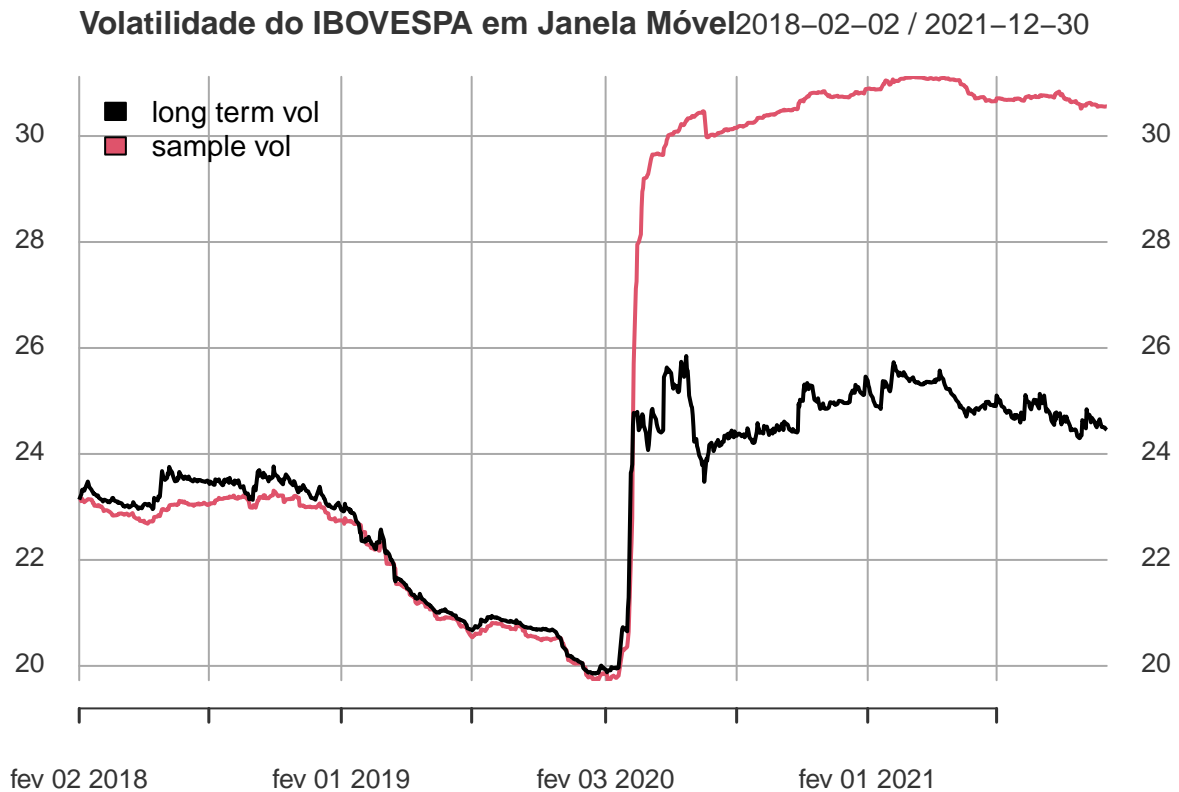
```
lt_vols <- rollapply(bvsp_rets, 756, function(x) {
  mod <- garchFit(~ garch(1, 1), data = x, trace = FALSE)
  params <- coef(mod)
  lt_variance <- params["omega"] / (1 - params["alpha1"] - params["beta1"])
  100 * sqrt(lt_variance * 252)
}, align = "right")
```

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



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

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



Estrutura a Termo de Volatilidade