

Homework5

Daniel de Souza Sobrinho Macedo RA:813524

Inicialização das bibliotecas

```
#install.packages("quantmod")
#install.packages("rugarch")

library(quantmod)
library(BatchGetSymbols)
library(tidyverse)
library(ggthemes)
library(rugarch)
library(FinTS)
library(forecast)
library(ggplot2)
```

Coleta e tratamento dos dados da VALE3 ### Recuperando retornos logarítmicos diários desde 2014

```
dados_ativos <- BatchGetSymbols('VALE3.SA',
                                first.date = '2014-01-01',
                                last.date = Sys.Date(),
                                type.return = "log",
                                freq.data = "daily")[[2]]

serie_vale <- dados_ativos %>%
  select(ret.closing.prices) %>%
  slice(-1) %>%
  pull() %>%
  ts()
```

Verificação da presença de heterocedasticidade condicional

```
ArchTest(serie_vale, lags = 10, demean = TRUE)

##
## ARCH LM-test; Null hypothesis: no ARCH effects
##
## data: serie_vale
## Chi-squared = 284.39, df = 10, p-value < 2.2e-16
```

Ajustando os modelos GARCH, EGARCH e GJR-GARCH

Função de ajuste parametrizado

```
ajustar_modelo <- function(tipo_modelo, dados, dist) {  
  especificacao <- ugarchspec(  
    variance.model = list(model = tipo_modelo, garchOrder = c(1, 1)),  
    mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),  
    distribution.model = dist  
  )  
  ugarchfit(spec = especificacao, data = dados)  
}  
  
# Ajustes dos diferentes modelos  
garch_norm <- ajustar_modelo("sGARCH", serie_vale, "norm")  
garch_t <- ajustar_modelo("sGARCH", serie_vale, "std")  
egarch_norm <- ajustar_modelo("eGARCH", serie_vale, "norm")  
egarch_t <- ajustar_modelo("eGARCH", serie_vale, "std")  
gjr_norm <- ajustar_modelo("gjrGARCH", serie_vale, "norm")  
gjr_t <- ajustar_modelo("gjrGARCH", serie_vale, "std")
```

Comparação de desempenho dos modelos

Cálculo da persistência e half-life

```
avaliar_modelo <- function(nome, modelo_ajustado) {  
  persis <- persistence(modelo_ajustado)  
  hl <- log(0.5) / log(persis)  
  ic <- infocriteria(modelo_ajustado)  
  
  tibble(  
    Modelo = nome,  
    Persistência = persis,  
    HalfLife = hl,  
    AIC = ic[1],  
    BIC = ic[2],  
    Shibata = ic[3],  
    HQ = ic[4]  
  )  
}  
  
tabela_resultados <- bind_rows(  
  avaliar_modelo("GARCH (Normal)", garch_norm),  
  avaliar_modelo("GARCH (t-Student)", garch_t),  
  avaliar_modelo("EGARCH (Normal)", egarch_norm),  
  avaliar_modelo("EGARCH (t-Student)", egarch_t),  
  avaliar_modelo("GJR (Normal)", gjr_norm),  
  avaliar_modelo("GJR (t-Student)", gjr_t)  
)  
  
tabela_resultados
```

```
## # A tibble: 6 x 7
##   Modelo                Persistência HalfLife    AIC    BIC Shibata    HQ
##   <chr>                <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 GARCH (Normal)        0.990    68.4 -4.63 -4.62 -4.63 -4.62
## 2 GARCH (t-Student)     0.989    64.7 -4.72 -4.71 -4.72 -4.72
## 3 EGARCH (Normal)       0.987    52.9 -4.63 -4.62 -4.63 -4.63
## 4 EGARCH (t-Student)    0.990    67.4 -4.72 -4.71 -4.72 -4.72
## 5 GJR (Normal)          0.990    67.3 -4.63 -4.62 -4.63 -4.62
## 6 GJR (t-Student)       0.989    65.4 -4.72 -4.71 -4.72 -4.71
```

Visualização dos resíduos dos modelos ajustados

Organizando dados dos resíduos para plotagem

```
residuos_df <- data.frame(
  Tempo = dados_ativos$ref.date[-1],
  GARCH_N = residuals(garch_norm),
  GARCH_T = residuals(garch_t),
  EGARCH_N = residuals(egarch_norm),
  EGARCH_T = residuals(egarch_t),
  GJR_N = residuals(gjr_norm),
  GJR_T = residuals(gjr_t)
)

plotar_residuos <- function(base, coluna, titulo) {
  ggplot(base, aes(x = Tempo, y = !!sym(coluna))) +
    geom_line(color = "#990000", alpha = 0.7) +
    labs(title = titulo, x = "Data", y = "Resíduos") +
    theme_classic()
}

# Criando os gráficos para cada modelo
library(cowplot)

##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggthemes':
##
##   theme_map

## The following object is masked from 'package:lubridate':
##
##   stamp

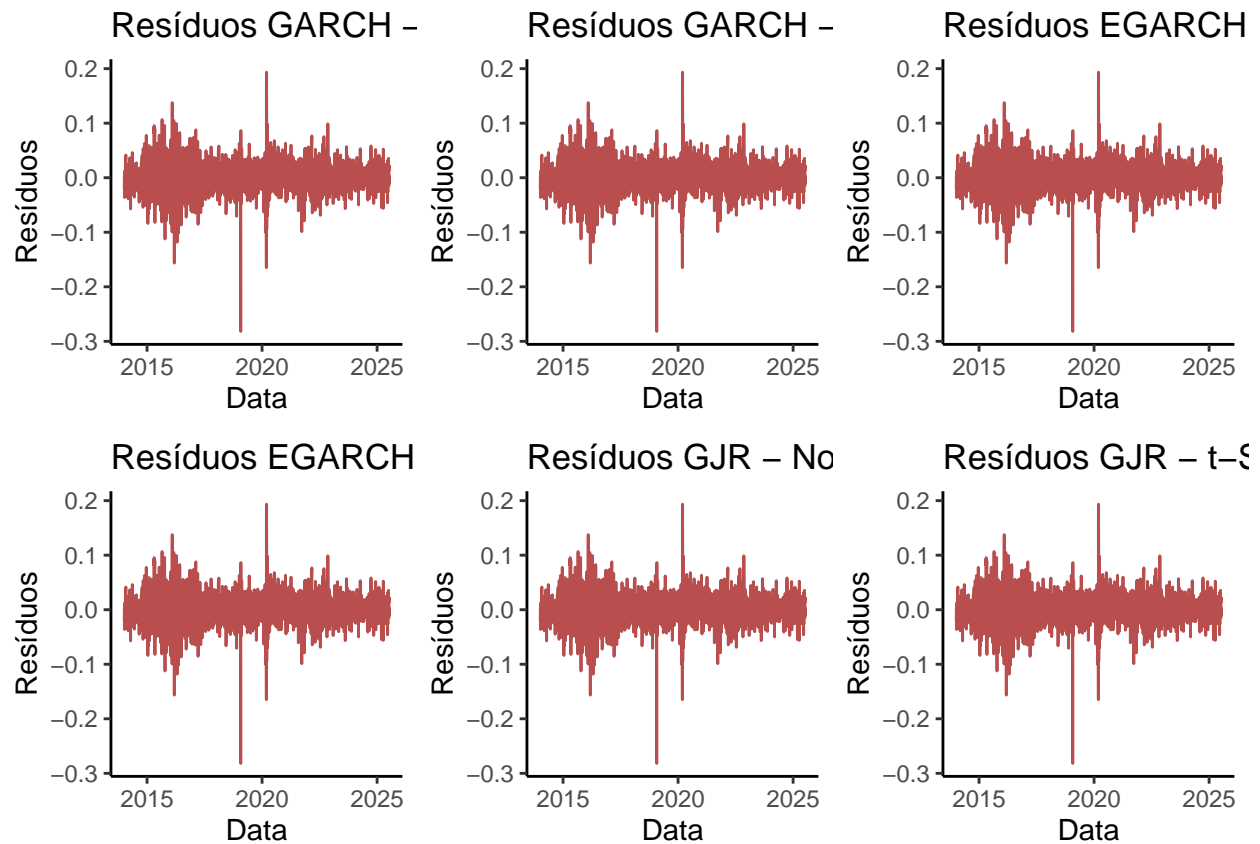
grid_residuos <- plot_grid(
  plotar_residuos(residuos_df, "GARCH_N", "Resíduos GARCH - Normal"),
  plotar_residuos(residuos_df, "GARCH_T", "Resíduos GARCH - t-Student"),
  plotar_residuos(residuos_df, "EGARCH_N", "Resíduos EGARCH - Normal"),
  plotar_residuos(residuos_df, "EGARCH_T", "Resíduos EGARCH - t-Student"),
  plotar_residuos(residuos_df, "GJR_N", "Resíduos GJR - Normal"),
```

```

plotar_residuos(residuos_df, "GJR_T", "Resíduos GJR - t-Student"),
nrow = 2
)

grid_residuos

```



Observação: Podemos verificar visualmente quais modelos apresentam maior estabilidade dos resíduos ao longo do tempo.

Previsão de volatilidade futura (Forecast)

Gerando previsões de 10 passos à frente

```

f10_garch_n <- ugarchforecast(garch_norm, n.ahead = 10)
f10_garch_t <- ugarchforecast(garch_t, n.ahead = 10)
f10_egarch_n <- ugarchforecast(egarch_norm, n.ahead = 10)
f10_egarch_t <- ugarchforecast(egarch_t, n.ahead = 10)
f10_gjr_n <- ugarchforecast(gjr_norm, n.ahead = 10)
f10_gjr_t <- ugarchforecast(gjr_t, n.ahead = 10)

```

```

sigmas_df <- tibble(
  Horizonte = rep(1:10, 6),
  Sigma = c(
    sigma(f10_garch_n),
    sigma(f10_garch_t),
    sigma(f10_egarch_n),
    sigma(f10_egarch_t),
    sigma(f10_gjr_n),
    sigma(f10_gjr_t)
  ),
  Modelo = rep(
    c("GARCH - Normal", "GARCH - t-Student",
      "EGARCH - Normal", "EGARCH - t-Student",
      "GJR - Normal", "GJR - t-Student"),
    each = 10
  )
)

ggplot(sigmas_df, aes(x = Horizonte, y = Sigma, color = Modelo)) +
  geom_line(size = 1.2) +
  scale_color_brewer(palette = "Dark2") +
  labs(title = "Volatilidade Prevista ( ) para os Modelos",
       x = "Passos à Frente",
       y = "Sigma (Volatilidade Estimada)",
       color = "Modelo") +
  theme_minimal(base_size = 13) +
  theme(legend.position = "bottom")

```

Visualização das volatilidades previstas (sigmas)

```

## 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.

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Volatilidade Prevista ( ) para os Modelos' in
## 'mbcsToSbcs': dot substituted for <cf>

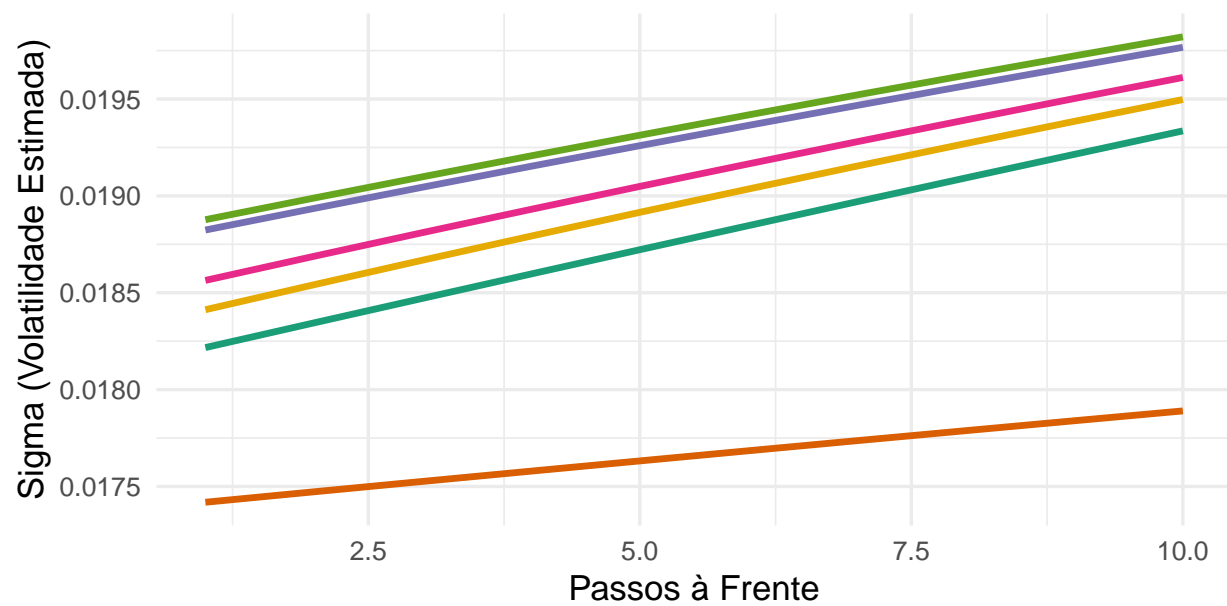
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Volatilidade Prevista ( ) para os Modelos' in
## 'mbcsToSbcs': dot substituted for <83>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Volatilidade Prevista ( ) para os Modelos' in
## 'mbcsToSbcs': dot substituted for <cf>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Volatilidade Prevista ( ) para os Modelos' in
## 'mbcsToSbcs': dot substituted for <83>

```

Volatilidade Prevista (..) para os Modelos



Modelo

- EGARCH - Normal
- EGARCH - t-Student
- GARCH - Normal
- GARCH - t-Student
- GJR - Normal
- GJR - t-Student