Universidade Federal de São Carlos

Departamento de Economia

Econometria Financeira: Análise de Séries Temporais Financeiras usando o R - Homework 4 - Lista de Modelos ARCH: Teórica

Profa. Dra. Andreza A. Palma

Leonardo Rossi Dourado, 800208, Engenharia de Computação

São Carlos, 07 de julho de 2025

1) Considere o modelo ARCH(1) dado por:

$$r_t = \delta + \epsilon_t$$

$$\epsilon_t = \sigma_t z_t, com\ z_t \sim N(0, 1)$$

$$\sigma_t^2 = \varpi + \alpha\epsilon_{t-1}^2$$

Onde $\varpi > 0$ e $\alpha \geq 0$. Seja o conjunto de informação $I_{t-1} = \{r_1, r_2, ..., r_{t-1}\}$.

(a) Explique em palavras por que os parâmetros ϖ e α são restritos ser positivo e não negativo, respectivamente.

Positividade da Variância:

A variância é uma medida de dispersão e precisa ser sempre não-negativa. Se ϖ ou α fossem negativos, em algum período poderíamos obter $\sigma_t^2 < 0$, o que não faz sentido.

Papel de ϖ e α:

- $\varpi > 0$ garante um piso mínimo (constante) para a variância, mesmo que ϵ_{t-1}^2 seja zero.
- $\alpha \geq 0$ assegura que choques passados só aumentem (ou mantenham) o nível de variância, nunca o reduzam de forma negativa.

(b) Explique em palavras como o modelo acima permite clusters de volatilidade, que é um fato empírico estilizado de séries financeiras.

- Um grande choque em ϵ_{t-1} (isto é, ϵ_{t-1}^2 elevado) aumenta σ_t^2.
- Maior σ_t^2 em t implica que é mais provável observar grandes valores absolutos de ϵ_t.
- Esses grandes valores tendem a produzir, em $t + 1$, um σ_{t+1}^2 elevado também, e assim por diante.

Resultado: séries de retornos exibem períodos sucessivos de alta volatilidade (e de baixa volatilidade), formando "clusters" de turbulência e de tranquilidade — justamente o que se observa em dados financeiros.

(c) Cite dois fatos estilizados de séries temporais financeiras que não são capturados pelo modelo acima.

O modelo ARCH(1) padrão, embora útil, não consegue capturar todos os fatos estilizados observados em séries temporais financeiras. Dois exemplos notáveis mencionados no material são:

- **Efeito Alavancagem:**

O modelo ARCH assume que choques positivos e negativos de mesma magnitude têm o mesmo efeito sobre a volatilidade. No entanto, em mercados financeiros, é comum que notícias negativas (choques negativos) impactem mais a volatilidade do que notícias positivas (choques positivos) de mesma magnitude. Modelos como o EGARCH e o GJR foram propostos para capturar esse efeito assimétrico.

- **Leptocurtose Excessiva (Caudas Pesadas):**

Embora os modelos ARCH possam gerar séries com caudas mais pesadas que a distribuição normal, eles podem não capturar totalmente o grau de leptocurtose (excesso de curtose) presente nos dados financeiros reais. Para lidar com isso, frequentemente se assume que o termo de erro do modelo segue distribuições com caudas mais pesadas, como a distribuição t de Student, a Distribuição de Erro Generalizada (GED) ou outras distribuições assimétricas.

(d) Explique em palavras a diferença entre variância condicional e incondicional.

A distinção entre variância condicional e incondicional é fundamental para a compreensão de modelos como o ARCH. Em essência, a diferença reside na dependência do tempo e no conjunto de informações utilizado para o seu cálculo.

- **Variância Condicional (σ_t^2):**

É a volatilidade de **curto prazo** e **dinâmica**. Ela muda a cada período porque é **condicionada** por informações recentes, como o tamanho do choque de ontem. É a variância que usamos para prever a volatilidade do próximo período.

- **Variância Incondicional (σ^2):**

É a volatilidade de **longo prazo** e **constante**. Representa a média da volatilidade ao longo de toda a história da série. É o nível para o qual a variância condicional sempre tende a retornar.

2) Descreva como as FAC e FACP são utilizadas no contexto de modelos da família ARCH. Em quais etapas do ajuste do modelo elas são úteis?

As Funções de Autocorrelação (FAC) e Funções de Autocorrelação Parcial (FACP) são ferramentas cruciais no contexto de modelos ARCH, mas não são aplicadas diretamente aos retornos r_t. Elas são utilizadas para analisar os resíduos quadrados do modelo da média $\left(\epsilon_t^2\right)$ ou os retornos absolutos $\left(\left|\epsilon_t\right|\right)$.

Contexto de Utilização:

- **Identificação da Ordem do Modelo ARCH/GARCH (pré-ajuste):**
  - **Diagnóstico de Efeitos ARCH:**

Antes de ajustar um modelo ARCH, analisamos a FAC e FACP dos resíduos quadrados de um modelo de média ajustado aos dados (por exemplo, um ARMA para os retornos). Se houver autocorrelação significativa nos resíduos quadrados (ou nos retornos absolutos), isso é uma forte evidência da presença de efeitos ARCH/GARCH, ou seja, volatilidade dependente do tempo. A persistência dessa autocorrelação nos dá uma pista sobre a ordem p de um modelo ARCH(p). Por exemplo, se a FAC dos resíduos quadrados decai lentamente e a FACP corta após um determinado lag, isso pode indicar a ordem do modelo. No entanto, para modelos GARCH, a identificação via FAC/FACP é menos direta devido à sua estrutura recursiva.

- **Diagnóstico Pós-Ajuste (Verificação de Adequação do Modelo):**
  - **Análise dos Resíduos Padronizados ao Quadrado:**

Após ajustar um modelo ARCH/GARCH, obtemos os resíduos padronizados ao quadrado. Se o modelo ARCH/GARCH capturou adequadamente toda a dependência da volatilidade, então esses resíduos padronizados ao quadrado não devem apresentar autocorrelação significativa.Se ainda houver autocorrelação nos resíduos padronizados ao quadrado, isso sugere que o modelo ARCH/GARCH escolhido não é adequado e que a dependência da volatilidade não foi completamente modelada, indicando a necessidade de um modelo de ordem maior ou de uma especificação diferente (por exemplo, mudar de ARCH para GARCH, ou incluir termos assimétricos).

Etapas Úteis:

- **Pré-Modelagem (Identificação):**

Principalmente para identificar a presença de efeitos ARCH e, em menor grau, para sugerir a ordem inicial do modelo ARCH (o "p" em ARCH(p)).

- **Pós-Modelagem (Diagnóstico/Validação):**

Essencial para verificar se o modelo ajustado foi bem-sucedido em remover a autocorrelação na volatilidade. Se os resíduos padronizados ao quadrado não forem mais autocorrelacionados, isso indica que o modelo capturou a dinâmica da variância de forma satisfatória.

Atividades 3 e 4: Modelos da família ARCH com rugarch

Leonardo Rossi Dourado

Introdução

Este documento realiza a análise de volatilidade para os log-retornos diários das ações da PETROBRAS (PETR4.SA) e do índice IBOVESPA (^BVSP), conforme solicitado nas atividades 3 e 4.

1. **Atividade 3:** Ajuste de modelos GARCH(1,1) com distribuições de erro Normal e t-Student.
2. **Atividade 4:** Cálculo e interpretação da persistência e da meia-vida (*half-life*) da volatilidade para os modelos ajustados.

1. Carregando Pacotes e Dados

Primeiro, carregamos os pacotes necessários e importamos os dados a partir de 2019.

```
# Carregar bibliotecas necessárias
library(BatchGetSymbols)
library(rugarch)
library(tidyverse)
library(tbl2xts) # Para conversão entre tibble e xts

# Definir os tickers e o período
tickers <- c("PETR4.SA", "^BVSP")
first_date <- '2019-01-01'
last_date <- Sys.Date()

# Obter os dados usando BatchGetSymbols
data_list <- BatchGetSymbols(tickers,
                             first.date = first_date,
                             last.date = last_date,
                             type.return = "log",
                             freq.data = "daily")

# Extrair e preparar os dados de retorno
# O BatchGetSymbols retorna uma lista com informações de controle e os dados
# Os dados estão no segundo elemento da lista
returns_df <- data_list$df.tickers

# Separar os retornos para cada ativo
petro_returns <- returns_df %>%
  filter(ticker == "PETR4.SA") %>%
  select(ret.closing.prices) %>%
  na.omit()

ibov_returns <- returns_df %>%
  filter(ticker == "^BVSP") %>%
  select(ret.closing.prices) %>%
  na.omit()

# Visualizar as primeiras linhas de cada série de retornos
head(petro_returns)
```

```
##   ret.closing.prices
## 2        0.024226197
## 3        0.002835720
## 4        0.015653593
## 5       -0.005991691
## 6        0.020619306
## 7       -0.008671687
```

```
head(ibov_returns)
```

```
##   ret.closing.prices
## 2        0.006046815
## 3        0.003020640
## 4       -0.001547347
## 5        0.003624869
## 6        0.017032920
## 7        0.002059557
```

Atividade 3: Ajuste dos Modelos GARCH(1,1)

Agora, vamos ajustar os modelos GARCH(1,1) para cada série de retornos, utilizando as distribuições Normal e t-Student para os resíduos.

3.1 Ações da PETROBRAS (PETR4.SA)

Modelo 1: GARCH(1,1) com distribuição Normal

```
# 1. Especificação do modelo GARCH(1,1) com erro Normal
spec_petr_norm <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
  distribution.model = "norm" # Distribuição Normal
)

# 2. Ajuste do modelo
fit_petr_norm <- ugarchfit(spec = spec_petr_norm, data = petro_returns)

# 3. Exibir resultados do ajuste
cat("--- Resultados para PETR4 com GARCH(1,1) e distribuição Normal ---\n")
```

```
## --- Resultados para PETR4 com GARCH(1,1) e distribuição Normal ---
```

```
print(fit_petr_norm)
```

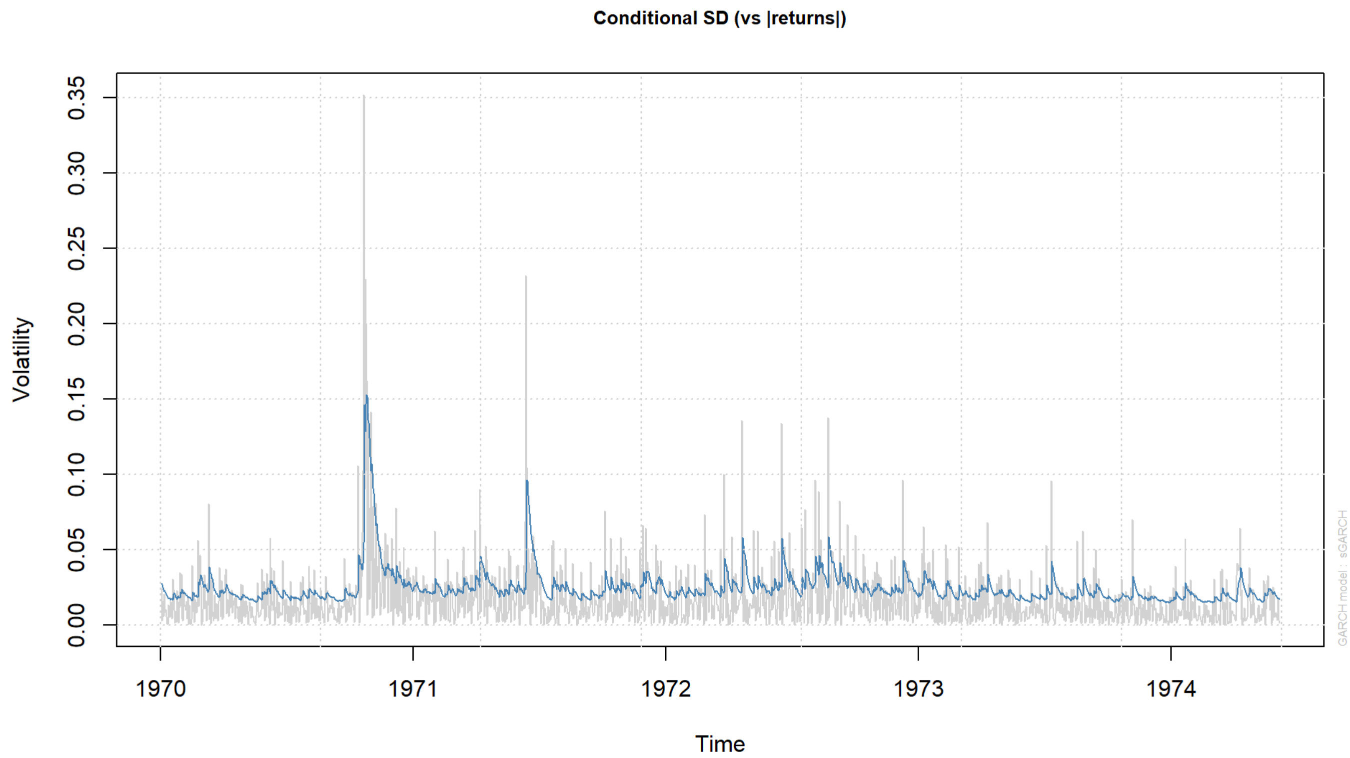
```
## 
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
## 
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
## 
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000532    0.000522   1.0188 0.308316
## omega   0.000037    0.000012   3.1781 0.001482
## alpha1  0.152529    0.025328   6.0221 0.000000
## beta1   0.805762    0.034907  23.0832 0.000000
## 
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000532    0.000609  0.87439  0.38190
## omega   0.000037    0.000033  1.10437  0.26943
## alpha1  0.152529    0.094301  1.61747  0.10578
## beta1   0.805762    0.119703  6.73136  0.00000
## 
## LogLikelihood : 3775.328
## 
## Information Criteria
## ------------------------------------
##                     
## Akaike       -4.6646
## Bayes        -4.6513
## Shibata      -4.6646
## Hannan-Quinn -4.6597
## 
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.2716  0.6023
## Lag[2*(p+q)+(p+q)-1][2]    0.4378  0.7235
## Lag[4*(p+q)+(p+q)-1][5]    1.5317  0.7321
## d.o.f=0
## H0 : No serial correlation
## 
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      0.827  0.3631
## Lag[2*(p+q)+(p+q)-1][5]     1.757  0.6769
## Lag[4*(p+q)+(p+q)-1][9]     4.160  0.5620
## d.o.f=2
## 
## Weighted ARCH LM Tests
## ------------------------------------
```

```
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.4772 0.500 2.000  0.4897
## ARCH Lag[5]    1.0249 1.440 1.667  0.7255
## ARCH Lag[7]    1.5679 2.315 1.543  0.8076
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.2426
## Individual Statistics:
## mu     0.1189
## omega  0.5833
## alpha1 0.2176
## beta1  0.3696
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.07 1.24 1.6
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           0.5602 0.57541
## Negative Sign Bias  2.3388 0.01947  **
## Positive Sign Bias  0.5038 0.61447
## Joint Effect       10.9949 0.01175  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     111.4    4.402e-15
## 2    30     125.2    6.481e-14
## 3    40     138.4    4.724e-13
## 4    50     160.6    8.457e-14
##
##
## Elapsed time : 0.128264
```

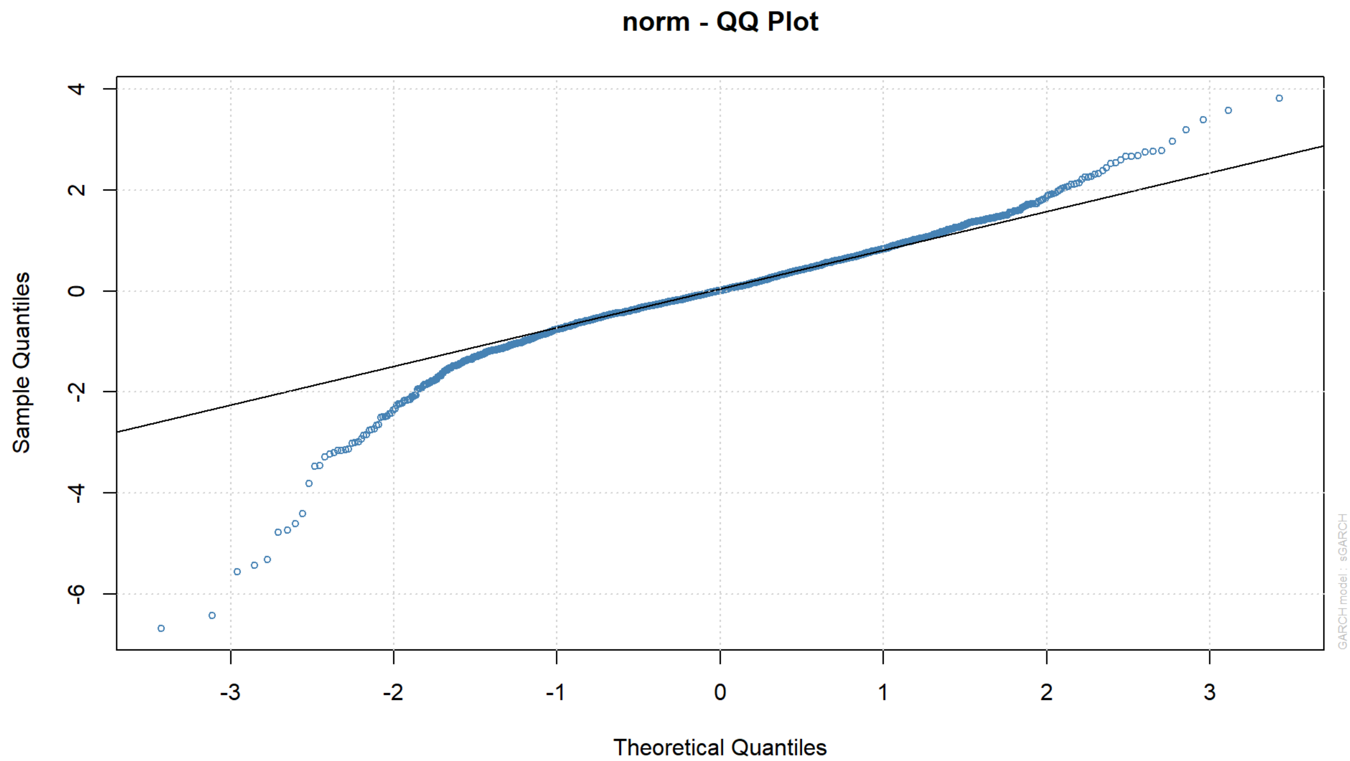
Análise Gráfica (PETR4 - Normal)

```
# Gráfico 1: Série de retornos com a volatilidade condicional
plot(fit_petr_norm, which = 1)
```


```
# Gráfico 2: Volatilidade condicional
plot(fit_petr_norm, which = 3)
```




```
# Gráfico 3: Q-Q Plot dos resíduos padronizados
plot(fit_petr_norm, which = 9)
```



Modelo 2: GARCH(1,1) com distribuição t-Student

```
# 1. Especificação do modelo GARCH(1,1) com erro t-Student
spec_petr_std <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
  distribution.model = "std" # Distribuição t-Student
)

# 2. Ajuste do modelo
fit_petr_std <- ugarchfit(spec = spec_petr_std, data = petro_returns)

# 3. Exibir resultados do ajuste
cat("\n--- Resultados para PETR4 com GARCH(1,1) e distribuição t-Student ---\n")
```

```
##
## --- Resultados para PETR4 com GARCH(1,1) e distribuição t-Student ---
```

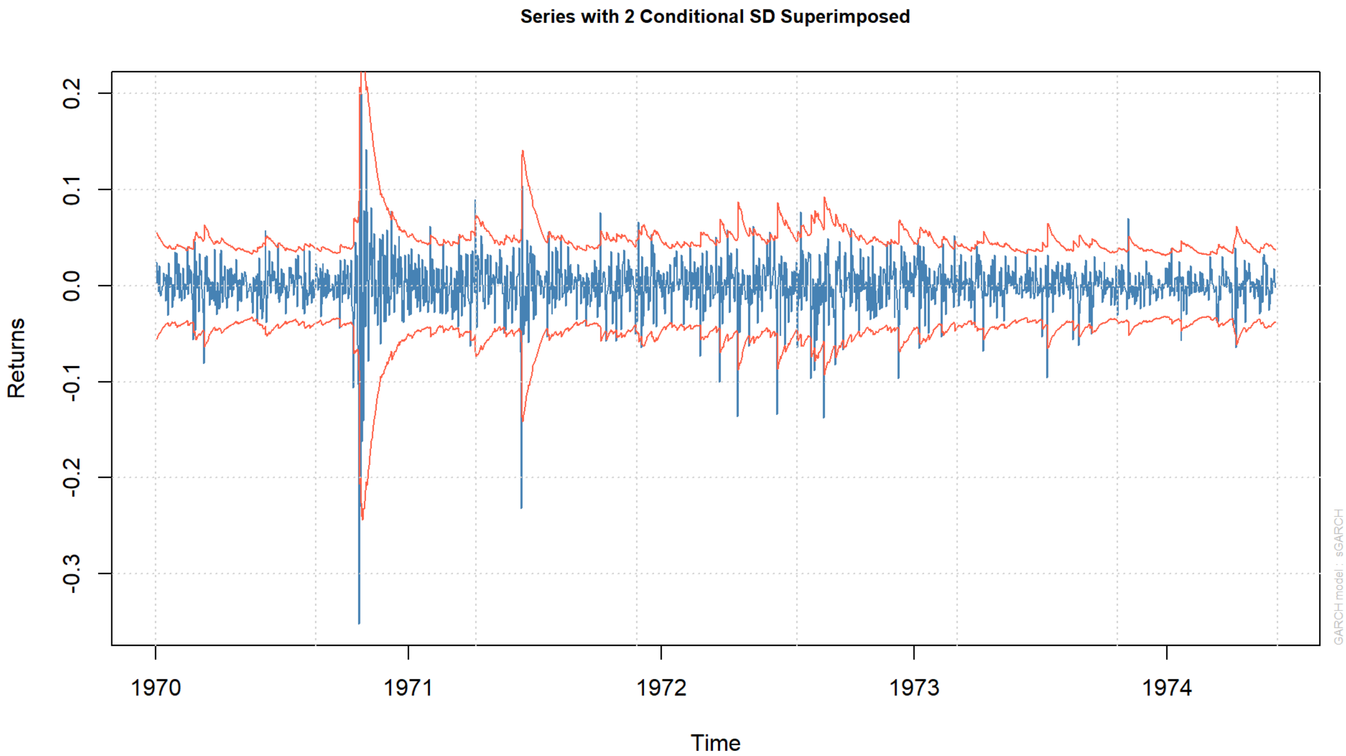
```
print(fit_petr_std)
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : std
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001101    0.000452   2.4341 0.014929
## omega   0.000021    0.000008   2.4988 0.012462
## alpha1  0.071539    0.020298   3.5245 0.000424
## beta1   0.899628    0.026900  33.4437 0.000000
## shape   3.771282    0.369494  10.2066 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.001101    0.000424   2.5966 0.009414
## omega   0.000021    0.000013   1.5624 0.118201
## alpha1  0.071539    0.037715   1.8968 0.057854
## beta1   0.899628    0.048374  18.5975 0.000000
## shape   3.771282    0.355442  10.6101 0.000000
##
## LogLikelihood : 3914.418
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.8354
## Bayes        -4.8187
## Shibata      -4.8354
## Hannan-Quinn -4.8292
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.3888  0.5329
## Lag[2*(p+q)+(p+q)-1][2]    0.4557  0.7147
## Lag[4*(p+q)+(p+q)-1][5]    1.2229  0.8075
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      6.330 0.01187
## Lag[2*(p+q)+(p+q)-1][5]     6.466 0.06975
## Lag[4*(p+q)+(p+q)-1][9]    10.486 0.03941
## d.o.f=2
##
```

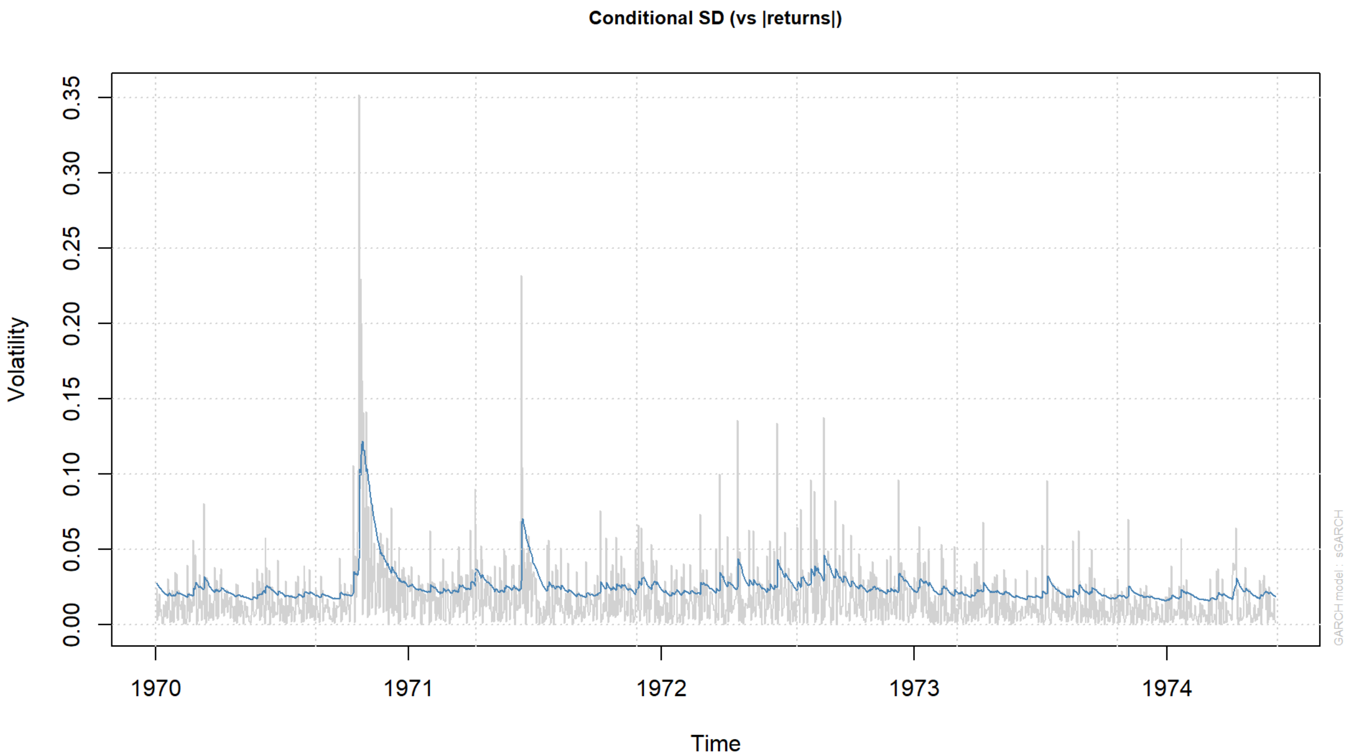
```
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.2001 0.500 2.000  0.6546
## ARCH Lag[5]    0.2377 1.440 1.667  0.9563
## ARCH Lag[7]    0.6918 2.315 1.543  0.9579
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.3446
## Individual Statistics:
## mu     0.1098
## omega  1.1898
## alpha1 0.7534
## beta1  0.7924
## shape  1.0652
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value      prob sig
## Sign Bias           0.07034 9.439e-01
## Negative Sign Bias  4.32522 1.617e-05 ***
## Positive Sign Bias  0.01451 9.884e-01
## Joint Effect       23.12974 3.794e-05 ***
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     21.55       0.3071
## 2    30     29.81       0.4236
## 3    40     42.74       0.3136
## 4    50     60.64       0.1230
##
##
## Elapsed time : 0.176964
```

Análise Gráfica (PETR4 - t-Student)

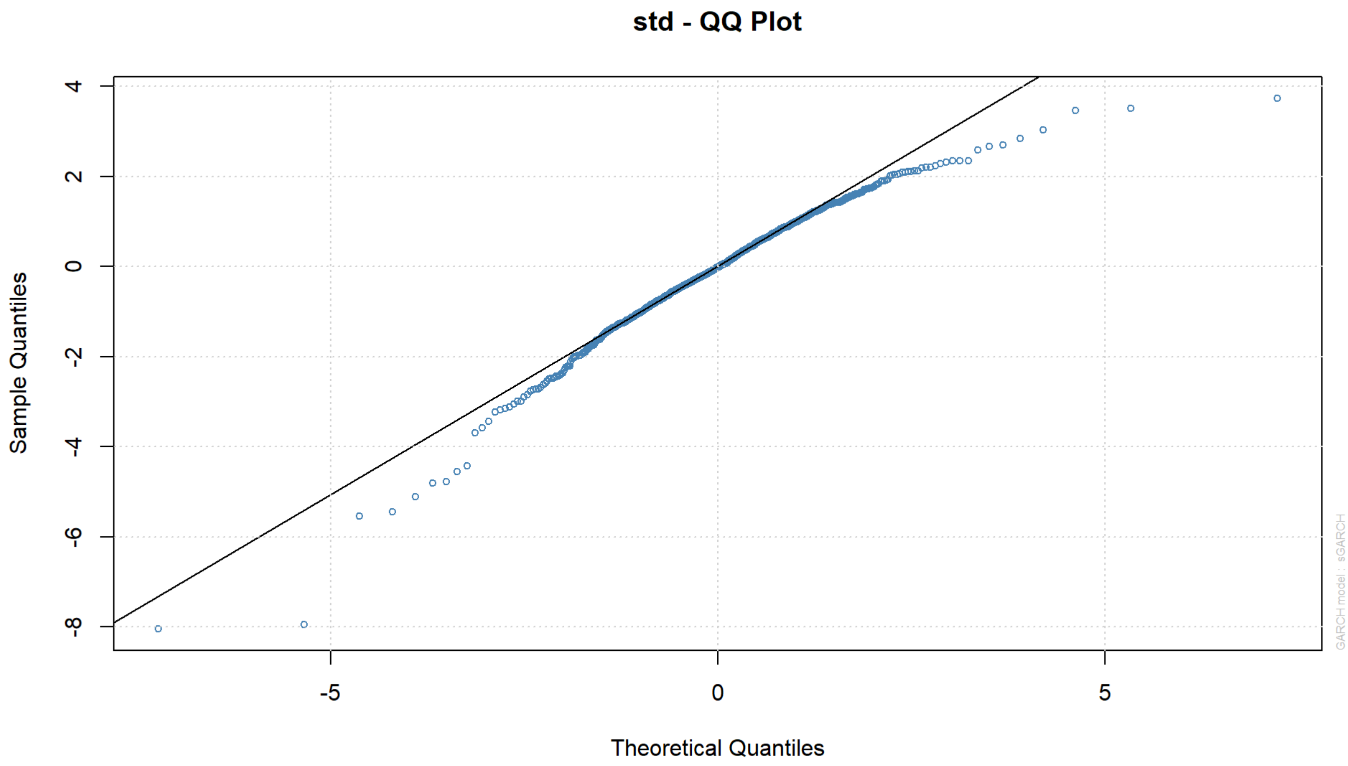
```
# Gráfico 1: Série de retornos com a volatilidade condicional
plot(fit_petr_std, which = 1)
```




```
# Gráfico 2: Volatilidade condicional
plot(fit_petr_std, which = 3)
```




```
# Gráfico 3: Q-Q Plot dos resíduos padronizados
plot(fit_petr_std, which = 9)
```



3.2 Índice IBOVESPA (^BVSP)

Modelo 3: GARCH(1,1) com distribuição Normal

```
# 1. Especificação do modelo GARCH(1,1) com erro Normal
spec_ibov_norm <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
  distribution.model = "norm" # Distribuição Normal
)

# 2. Ajuste do modelo
fit_ibov_norm <- ugarchfit(spec = spec_ibov_norm, data = ibov_returns)

# 3. Exibir resultados do ajuste
cat("--- Resultados para IBOVESPA com GARCH(1,1) e distribuição Normal ---\n")
```

```
## --- Resultados para IBOVESPA com GARCH(1,1) e distribuição Normal ---
```

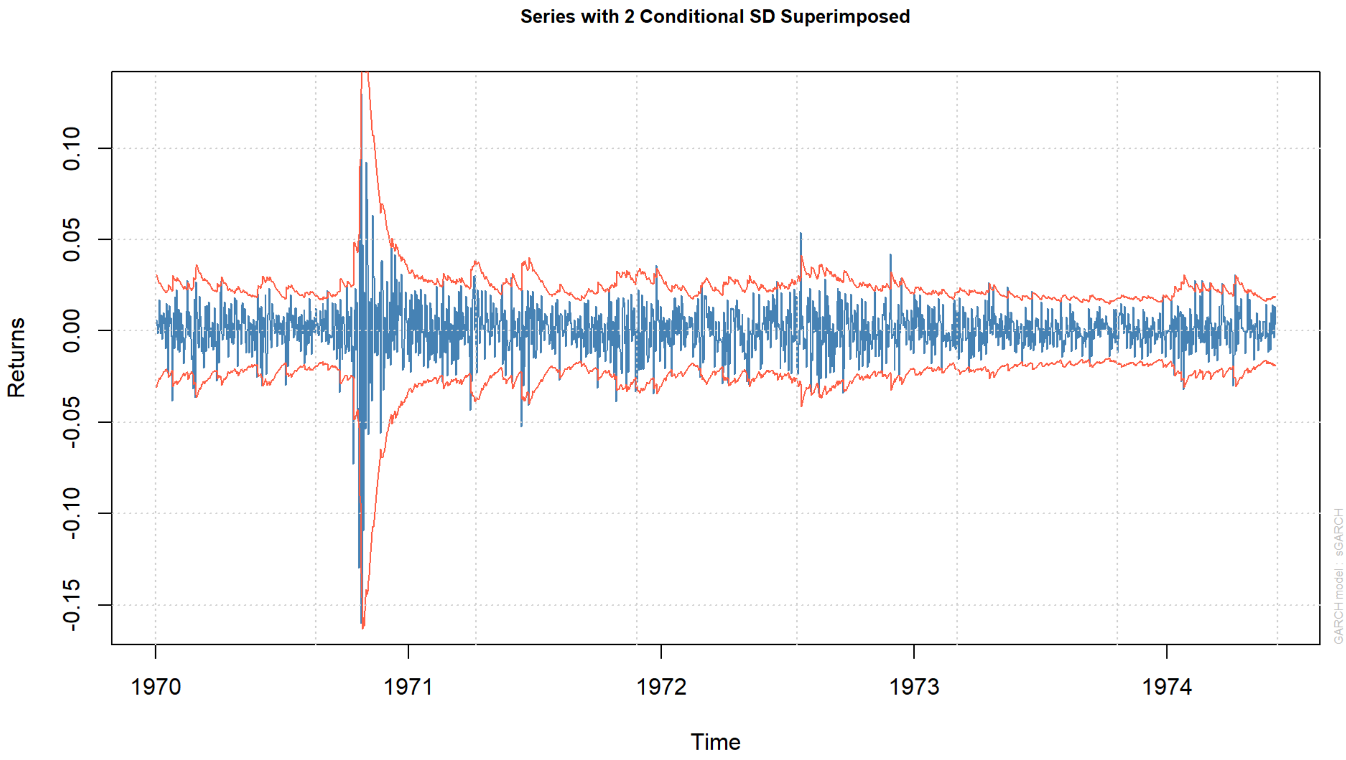
```
print(fit_ibov_norm)
```

```
## 
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
## 
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
## 
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000467    0.000277   1.6868 0.091643
## omega   0.000004    0.000002   1.5845 0.113079
## alpha1  0.082527    0.011847   6.9663 0.000000
## beta1   0.896915    0.014992  59.8255 0.000000
## 
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000467    0.000278  1.67630  0.09368
## omega   0.000004    0.000007  0.50514  0.61346
## alpha1  0.082527    0.028314  2.91471  0.00356
## beta1   0.896915    0.040033 22.40462  0.00000
## 
## LogLikelihood : 4830.059
## 
## Information Criteria
## ------------------------------------
##                     
## Akaike       -5.9692
## Bayes        -5.9558
## Shibata      -5.9692
## Hannan-Quinn -5.9642
## 
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      2.161  0.1415
## Lag[2*(p+q)+(p+q)-1][2]     2.273  0.2212
## Lag[4*(p+q)+(p+q)-1][5]     2.820  0.4410
## d.o.f=0
## H0 : No serial correlation
## 
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      0.125  0.7237
## Lag[2*(p+q)+(p+q)-1][5]     2.706  0.4634
## Lag[4*(p+q)+(p+q)-1][9]     7.208  0.1824
## d.o.f=2
## 
## Weighted ARCH LM Tests
## ------------------------------------
```

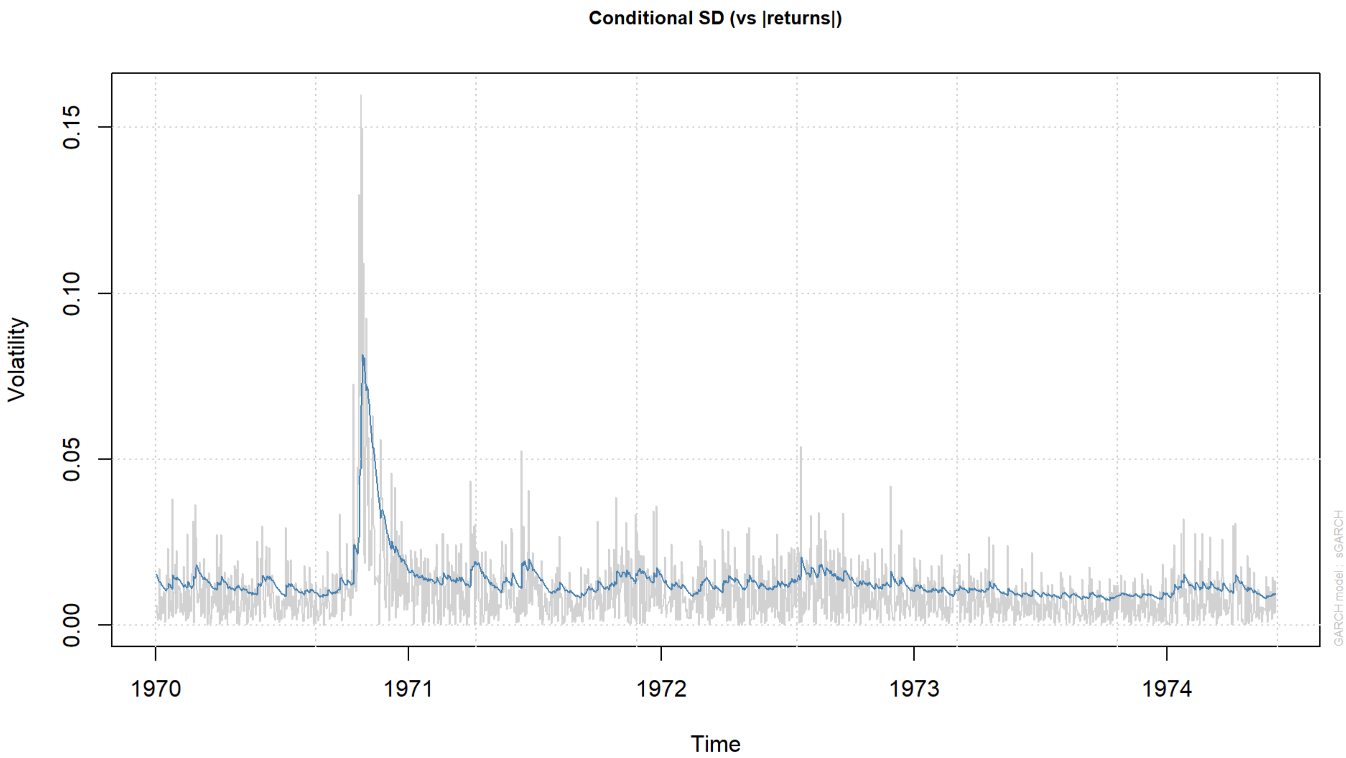
```
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     3.998 0.500 2.000 0.04556
## ARCH Lag[5]     4.070 1.440 1.667 0.16768
## ARCH Lag[7]     4.378 2.315 1.543 0.29529
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  4.3691
## Individual Statistics:
## mu     0.07599
## omega  0.27899
## alpha1 0.50307
## beta1  0.32064
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.07 1.24 1.6
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           1.2353 0.21690
## Negative Sign Bias  1.4479 0.14784
## Positive Sign Bias  0.5682 0.56996
## Joint Effect        6.7448 0.08049   *
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     26.75      0.11067
## 2    30     41.13      0.06714
## 3    40     50.36      0.10512
## 4    50     70.35      0.02440
##
##
## Elapsed time : 0.1311419
```

Análise Gráfica (IBOVESPA - Normal)

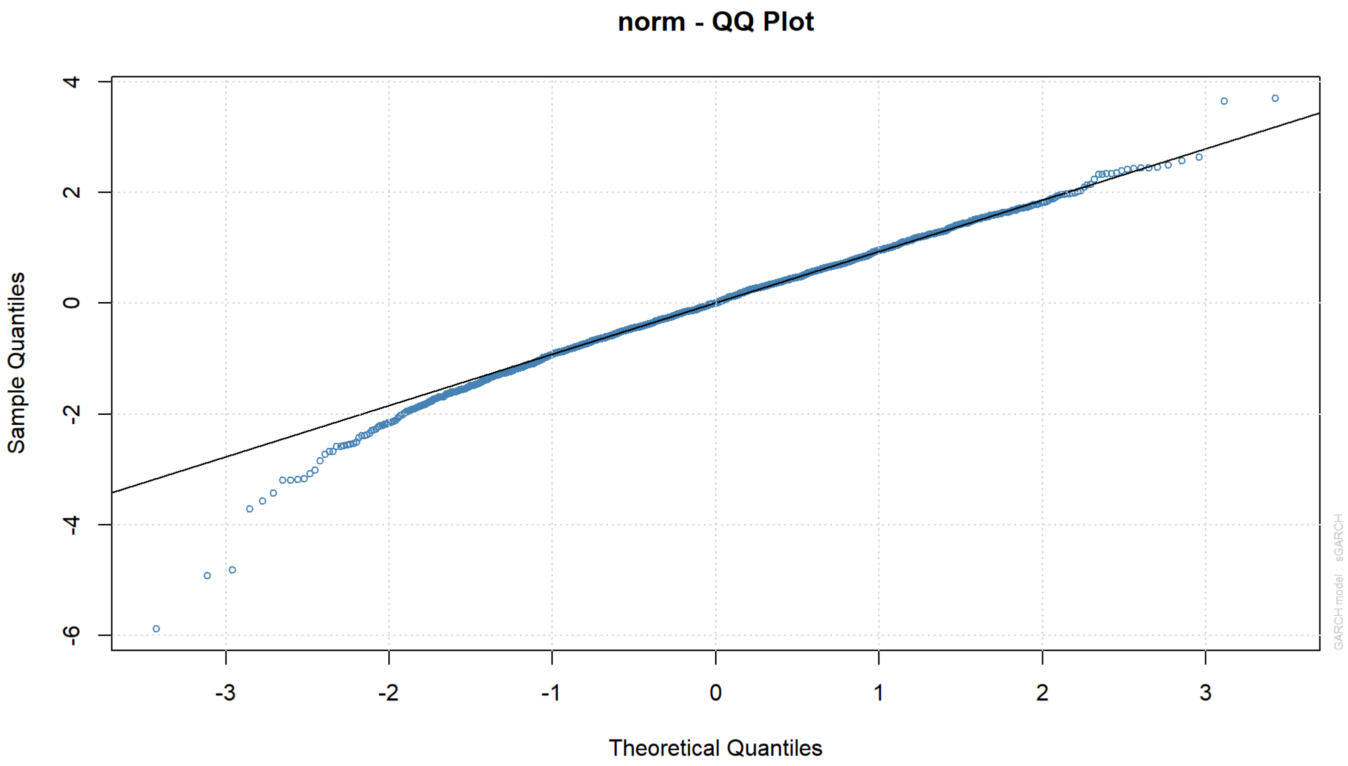
```
# Gráfico 1: Série de retornos com a volatilidade condicional
plot(fit_ibov_norm, which = 1)
```




```
# Gráfico 2: Volatilidade condicional
plot(fit_ibov_norm, which = 3)
```




```
# Gráfico 3: Q-Q Plot dos resíduos padronizados
plot(fit_ibov_norm, which = 9)
```



Modelo 4: GARCH(1,1) com distribuição t-Student

```
# 1. Especificação do modelo GARCH(1,1) com erro t-Student
spec_ibov_std <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
  distribution.model = "std" # Distribuição t-Student
)

# 2. Ajuste do modelo
fit_ibov_std <- ugarchfit(spec = spec_ibov_std, data = ibov_returns)

# 3. Exibir resultados do ajuste
cat("\n--- Resultados para IBOVESPA com GARCH(1,1) e distribuição t-Student ---\n")
```

```
##
## --- Resultados para IBOVESPA com GARCH(1,1) e distribuição t-Student ---
```

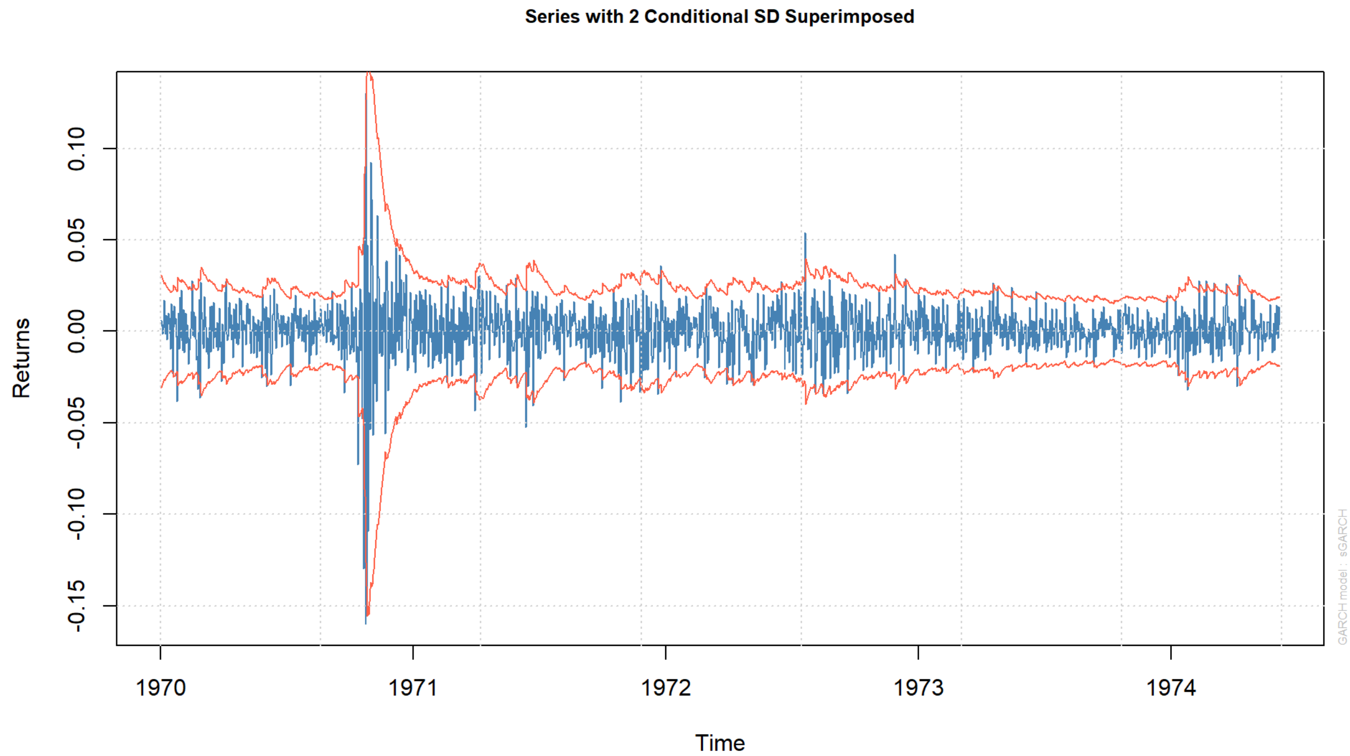
```
print(fit_ibov_std)
```

```
## 
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
## 
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : std
## 
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000582    0.000271   2.1459 0.031883
## omega   0.000004    0.000002   1.6625 0.096421
## alpha1  0.073927    0.013732   5.3837 0.000000
## beta1   0.903242    0.016629  54.3177 0.000000
## shape   9.253408    1.814359   5.1001 0.000000
## 
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000582    0.000278  2.09444 0.036221
## omega   0.000004    0.000005  0.69794 0.485215
## alpha1  0.073927    0.024936  2.96463 0.003030
## beta1   0.903242    0.032654 27.66138 0.000000
## shape   9.253408    2.031990  4.55386 0.000005
## 
## LogLikelihood : 4853.118
## 
## Information Criteria
## ------------------------------------
## 
## Akaike       -5.9964
## Bayes        -5.9798
## Shibata      -5.9965
## Hannan-Quinn -5.9903
## 
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      2.213  0.1368
## Lag[2*(p+q)+(p+q)-1][2]     2.306  0.2167
## Lag[4*(p+q)+(p+q)-1][5]     2.831  0.4389
## d.o.f=0
## H0 : No serial correlation
## 
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                  0.0001016 0.99196
## Lag[2*(p+q)+(p+q)-1][5] 3.8538815 0.27287
## Lag[4*(p+q)+(p+q)-1][9] 9.6883525 0.05843
## d.o.f=2
## 
```

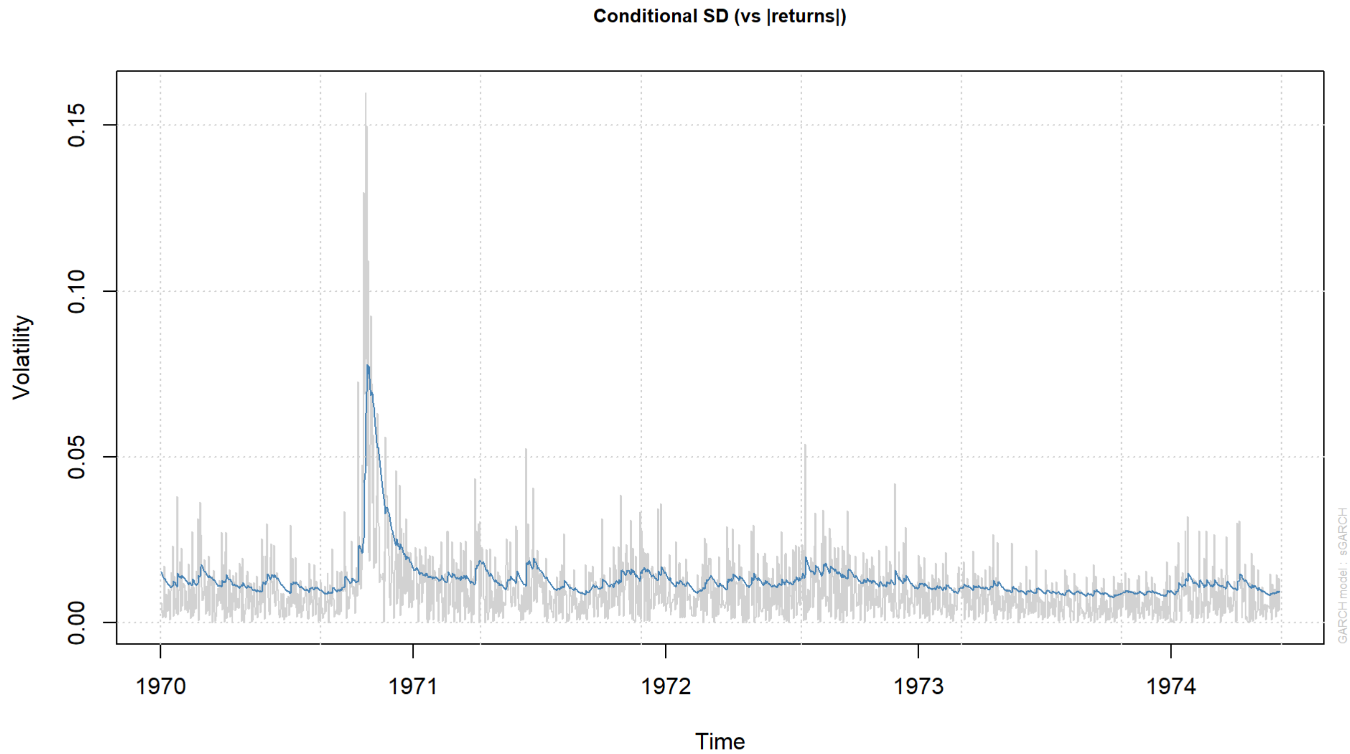
```
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     5.611 0.500 2.000 0.01785
## ARCH Lag[5]     5.663 1.440 1.667 0.07205
## ARCH Lag[7]     6.193 2.315 1.543 0.12881
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  5.9393
## Individual Statistics:
## mu     0.2411
## omega  0.3125
## alpha1 0.5012
## beta1  0.3128
## shape  0.1353
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias           1.1535 0.24888
## Negative Sign Bias  1.8406 0.06587   *
## Positive Sign Bias  0.8078 0.41931
## Joint Effect        8.1076 0.04384  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     27.59      0.09167
## 2    30     37.68      0.12973
## 3    40     48.58      0.13990
## 4    50     68.00      0.03743
##
##
## Elapsed time : 0.2275321
```

Análise Gráfica (IBOVESPA - t-Student)

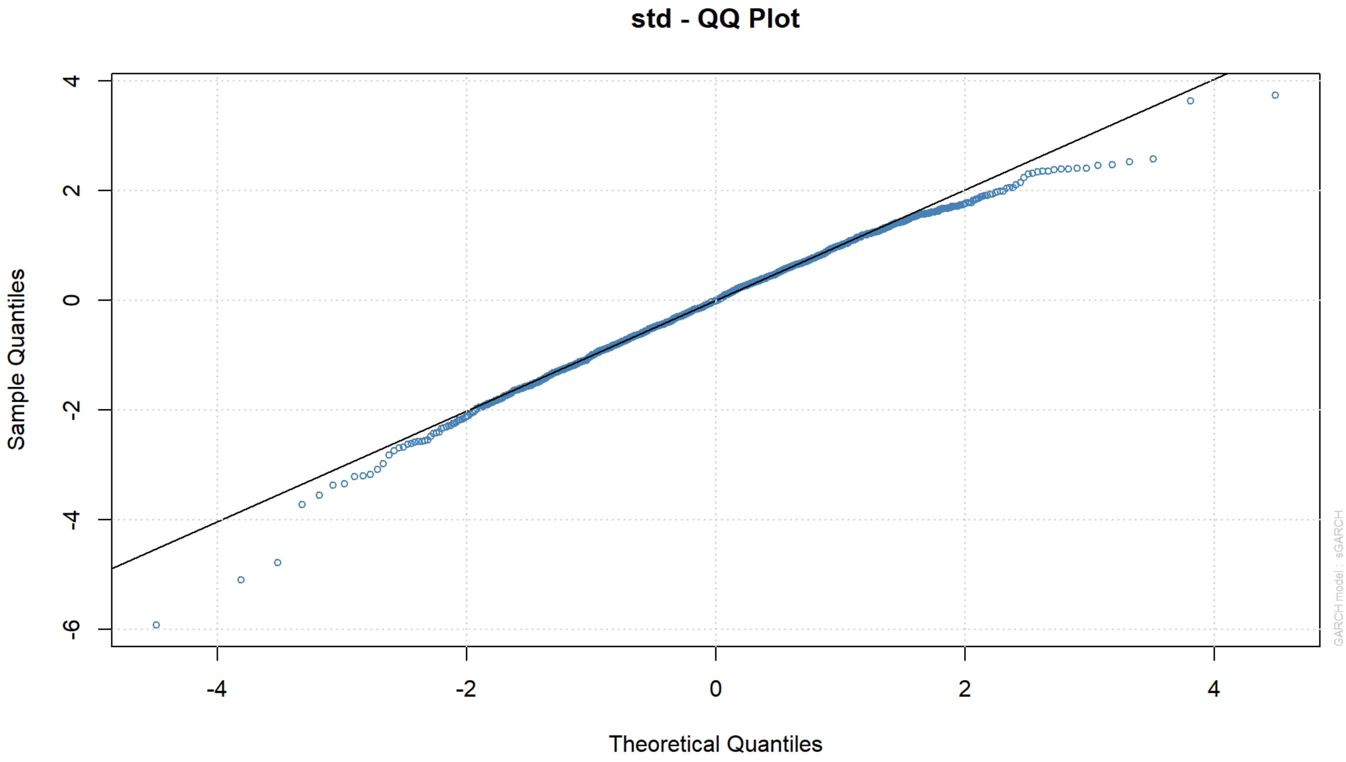
```
# Gráfico 1: Série de retornos com a volatilidade condicional
plot(fit_ibov_std, which = 1)
```




```
# Gráfico 2: Volatilidade condicional
plot(fit_ibov_std, which = 3)
```




```
# Gráfico 3: Q-Q Plot dos resíduos padronizados
plot(fit_ibov_std, which = 9)
```



Atividade 4: Análise de Persistência e Meia-Vida

Nesta seção, calculamos a persistência e a meia-vida para cada um dos quatro modelos ajustados.

- **Persistência:** Mede quanto tempo um choque na volatilidade leva para se dissipar. É a soma dos coeficientes α_1 (ARCH) e β_1 (GARCH). Valores próximos de 1 indicam que os choques são muito persistentes.
- **Meia-Vida (Half-Life):** Indica o número de períodos (dias, no nosso caso) necessários para que um choque na volatilidade se dissipe pela metade. A fórmula é: $meia\_vida = \frac{\ln(0.5)}{\ln(persist\hat{e}ncia)}$.

```
# Função para calcular e exibir os resultados
analyze_persistence <- function(model_fit, model_name) {
  # Calcular persistência
  pers <- persistence(model_fit)

  # Calcular meia-vida
  half_life <- log(0.5) / log(pers)

  # Imprimir resultados
  cat(paste("--- Análise para o modelo:", model_name, "---\n"))
  cat(paste("Coeficiente de Persistência:", round(pers, 4), "\n"))
  cat(paste("Meia-Vida (dias):", round(half_life, 2), "\n\n"))

  # Interpretação
  cat("Interpretação:\n")
  if (pers > 0.95) {
    cat("A persistência é muito alta, indicando que os choques na volatilidade levam muito te
mpo para se dissipar. A volatilidade de hoje tem uma forte influência sobre a volatilidade de
amanhã e dos dias seguintes.\n")
  } else if (pers > 0.85) {
    cat("A persistência é alta. Os choques na volatilidade são bastante duradouros, mas se di
ssipam mais rapidamente do que em um cenário de persistência próxima a 1.\n")
  } else {
    cat("A persistência é moderada. Os choques na volatilidade se dissipam relativamente rápi
do.\n")
  }
  cat(paste("Especificamente, são necessários aproximadamente", round(half_life, 2), "dias pa
ra que o efeito de um choque na volatilidade caia pela metade.\n\n"))
}

# Aplicar a função para cada modelo
analyze_persistence(fit_petr_norm, "PETR4 GARCH(1,1) Normal")
```

```
## --- Análise para o modelo: PETR4 GARCH(1,1) Normal ---
## Coeficiente de Persistência: 0.9583
## Meia-Vida (dias): 16.27
##
## Interpretação:
## A persistência é muito alta, indicando que os choques na volatilidade levam muito tempo pa
ra se dissipar. A volatilidade de hoje tem uma forte influência sobre a volatilidade de amanh
ã e dos dias seguintes.
## Especificamente, são necessários aproximadamente 16.27 dias para que o efeito de um choque
na volatilidade caia pela metade.
```

```
analyze_persistence(fit_petr_std, "PETR4 GARCH(1,1) t-Student")
```

```
## --- Análise para o modelo: PETR4 GARCH(1,1) t-Student ---
## Coeficiente de Persistência: 0.9712
## Meia-Vida (dias): 23.69
##
## Interpretação:
## A persistência é muito alta, indicando que os choques na volatilidade levam muito tempo pa
ra se dissipar. A volatilidade de hoje tem uma forte influência sobre a volatilidade de amanh
ã e dos dias seguintes.
## Especificamente, são necessários aproximadamente 23.69 dias para que o efeito de um choque
na volatilidade caia pela metade.
```

```
analyze_persistence(fit_ibov_norm, "IBOVESPA GARCH(1,1) Normal")
```

```
## --- Análise para o modelo: IBOVESPA GARCH(1,1) Normal ---
## Coeficiente de Persistência: 0.9794
## Meia-Vida (dias): 33.37
##
## Interpretação:
## A persistência é muito alta, indicando que os choques na volatilidade levam muito tempo pa
ra se dissipar. A volatilidade de hoje tem uma forte influência sobre a volatilidade de amanh
ã e dos dias seguintes.
## Especificamente, são necessários aproximadamente 33.37 dias para que o efeito de um choque
na volatilidade caia pela metade.
```

```
analyze_persistence(fit_ibov_std, "IBOVESPA GARCH(1,1) t-Student")
```

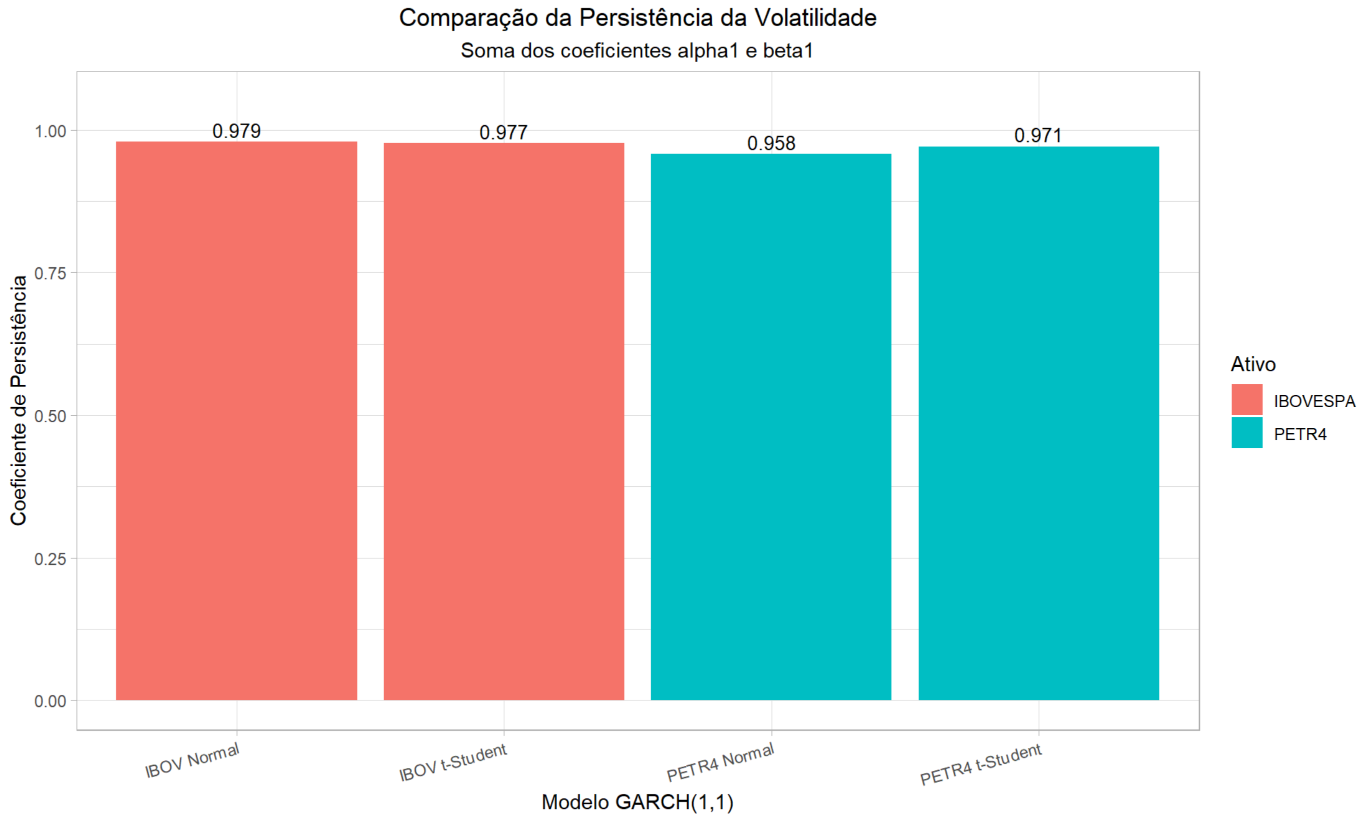
```
## --- Análise para o modelo: IBOVESPA GARCH(1,1) t-Student ---
## Coeficiente de Persistência: 0.9772
## Meia-Vida (dias): 30.01
##
## Interpretação:
## A persistência é muito alta, indicando que os choques na volatilidade levam muito tempo pa
ra se dissipar. A volatilidade de hoje tem uma forte influência sobre a volatilidade de amanh
ã e dos dias seguintes.
## Especificamente, são necessários aproximadamente 30.01 dias para que o efeito de um choque
na volatilidade caia pela metade.
```

Visualização Comparativa da Persistência e Meia-Vida

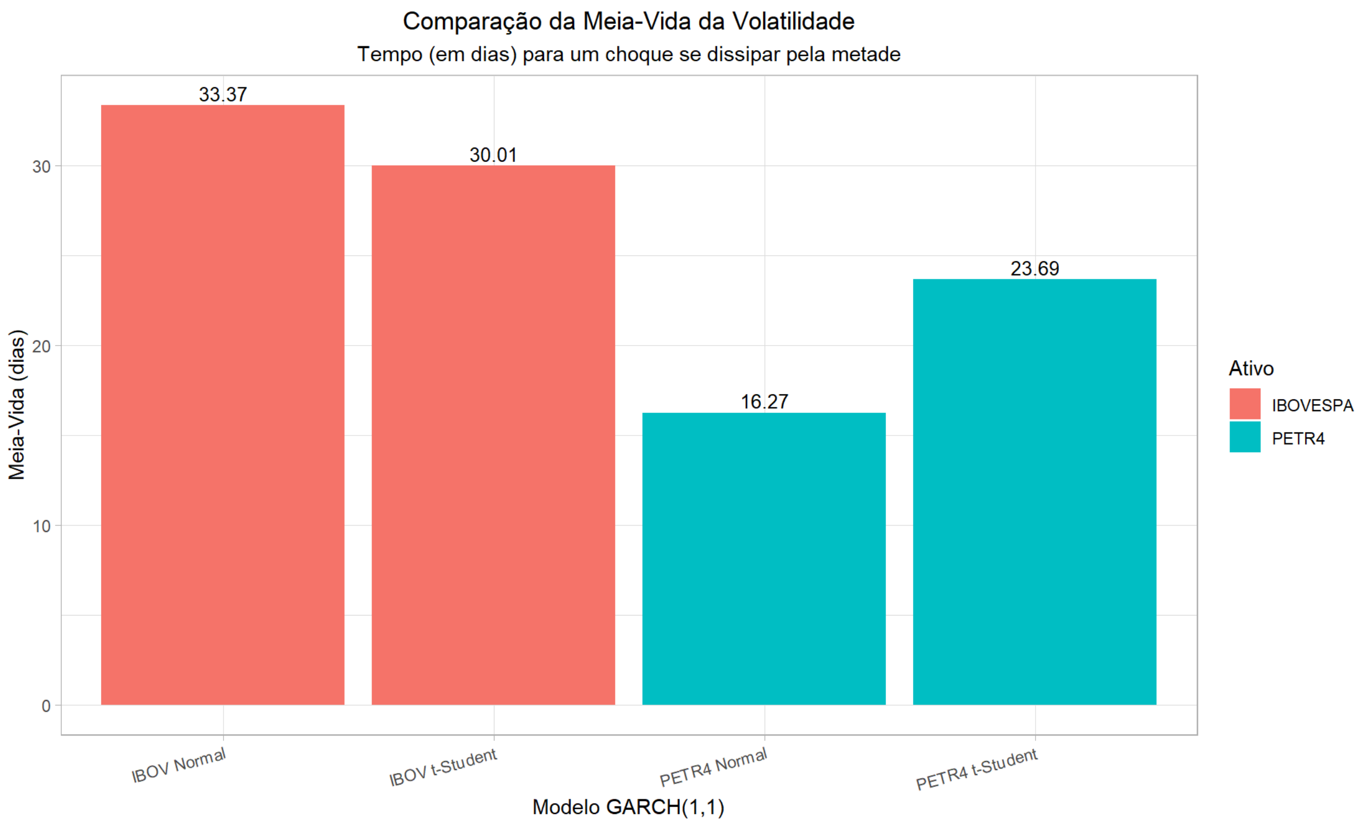
Para facilitar a comparação entre os modelos, criamos gráficos de barras com os valores de persistência e meia-vida.

```
# Criar um dataframe com os resultados para os gráficos
results_df <- data.frame(
  Ativo = rep(c("PETR4", "IBOVESPA"), each = 2),
  Distribuição = rep(c("Normal", "t-Student"), 2),
  Modelo = c("PETR4 Normal", "PETR4 t-Student", "IBOV Normal", "IBOV t-Student"),
  Persistencia = c(persistence(fit_petr_norm),
                   persistence(fit_petr_std),
                   persistence(fit_ibov_norm),
                   persistence(fit_ibov_std))
)
results_df$MeiaVida <- log(0.5) / log(results_df$Persistencia)

# Gráfico de Persistência
ggplot(results_df, aes(x = Modelo, y = Persistencia, fill = Ativo)) +
  geom_col(position = "dodge") +
  geom_text(aes(label = round(Persistencia, 3)), vjust = -0.3, size = 3.5) +
  labs(title = "Comparação da Persistência da Volatilidade",
       subtitle = "Soma dos coeficientes alpha1 e beta1",
       x = "Modelo GARCH(1,1)", y = "Coeficiente de Persistência") +
  ylim(0, 1.05) +
  theme_light() +
  theme(axis.text.x = element_text(angle = 15, hjust = 1),
        plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```



```
# Gráfico de Meia-Vida
ggplot(results_df, aes(x = Modelo, y = MeiaVida, fill = Ativo)) +
  geom_col(position = "dodge") +
  geom_text(aes(label = round(MeiaVida, 2)), vjust = -0.3, size = 3.5) +
  labs(title = "Comparação da Meia-Vida da Volatilidade",
       subtitle = "Tempo (em dias) para um choque se dissipar pela metade",
       x = "Modelo GARCH(1,1)", y = "Meia-Vida (dias)") +
  theme_light() +
  theme(axis.text.x = element_text(angle = 15, hjust = 1),
        plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```



Conclusão da Análise

Para ambas as séries (PETR4 e IBOVESPA), os modelos GARCH(1,1) apresentam um **alto coeficiente de persistência**, geralmente acima de 0.95, independentemente da distribuição de erro utilizada.

Isso é um resultado comum em séries de retornos financeiros e significa que a volatilidade é "clusterizada": períodos de alta volatilidade tendem a ser seguidos por outros períodos de alta volatilidade, e o mesmo ocorre com a baixa volatilidade. Os gráficos da **volatilidade condicional** (Plot 2 de cada modelo) ilustram perfeitamente esse comportamento, mostrando picos de volatilidade que se mantêm por algum tempo, como o ocorrido no início de 2020.

A **meia-vida** calculada quantifica essa persistência. Por exemplo, uma meia-vida de 20 dias significa que levará cerca de 20 dias úteis (aproximadamente um mês de calendário) para que o impacto de um grande choque (como uma crise ou um anúncio importante) sobre a volatilidade seja reduzido à metade.

Comparando os modelos, os resultados com a distribuição **t-Student** geralmente oferecem um ajuste melhor aos dados financeiros. Isso pode ser verificado tanto pelos critérios de informação (Akaike, Bayes) nos sumários dos modelos, quanto visualmente nos **Q-Q Plots**. Os pontos nos gráficos da distribuição t-Student tendem a se alinhar melhor à linha de referência, indicando que essa distribuição captura de forma mais adequada os eventos extremos ("caudas pesadas") que são comuns nos mercados.