

homework-1

Lucas Maciel Balieiro - 800534

2025-07-11

1. Carregamento dos Pacotes e Definição dos Ativos

Primeiro, vamos carregar os pacotes necessários para a análise.

Neste chunk de código, também definimos os 4 ativos que serão analisados, conforme solicitado na primeira questão. Os ativos escolhidos foram: * **PETR4.SA**: Petrobras (Petróleo) * **VALE3.SA**: Vale (Mineração) * **ITUB4.SA**: Itaú Unibanco (Setor Financeiro) * **MGLU3.SA**: Magazine Luiza (Varejo)

```
library(BatchGetSymbols)
library(tidyverse)
library(ggthemes)
library(cowplot)
library(moments)
library(corrplot)

rm(list=ls())

tickers <- c('PETR4.SA', 'VALE3.SA', 'ITUB4.SA', 'MGLU3.SA')

first_date <- '2020-01-01'
last_date <- Sys.Date()
assets_list <- BatchGetSymbols(tickers = tickers,
                               first.date = first_date,
                               last.date = last_date,
                               type.return = "log",
                               freq.data = "daily")

assets <- assets_list[[2]]
knitr::kable(head(assets))
```

price.open	price.high	price.low	price.close	volume	price.adjusted	rf.date	ticker	ret.adjusted.prices	ret.closing.prices
30.51	30.70	30.31	30.70	37774500	9.656280	2020-01-02	PETR4.SA	NA	NA
30.88	31.24	30.45	30.45	71595600	9.577648	2020-01-03	PETR4.SA-0.0081765	-0.0081767	
30.43	30.94	29.95	30.81	81844000	9.690881	2020-01-06	PETR4.SA 0.0117532	0.0117533	
30.82	30.88	30.47	30.69	32822000	9.653134	2020-01-07	PETR4.SA-0.0039027	-0.0039024	

price.open	price.high	price.low	price.close	volume	price.adjusted	ref.date	ticker	ret.adjusted.prices	ret.closing.prices
30.69	30.77	30.24	30.50	48215600	9.593374	2020-01-08	PETR4.SA-0.0062100	-0.0062102	
30.47	30.62	30.25	30.40	36102700	9.561921	2020-01-09	PETR4.SA-0.0032840	-0.0032841	

2. Análise dos Fatos Estilizados (Questão 1)

Agora, vamos verificar os fatos estilizados para cada uma das 4 variáveis escolhidas.

```
for (ticker_atual in tickers) {

  dados_ativo <- assets %>% filter(ticker == ticker_atual)

  # --- Gráficos ---

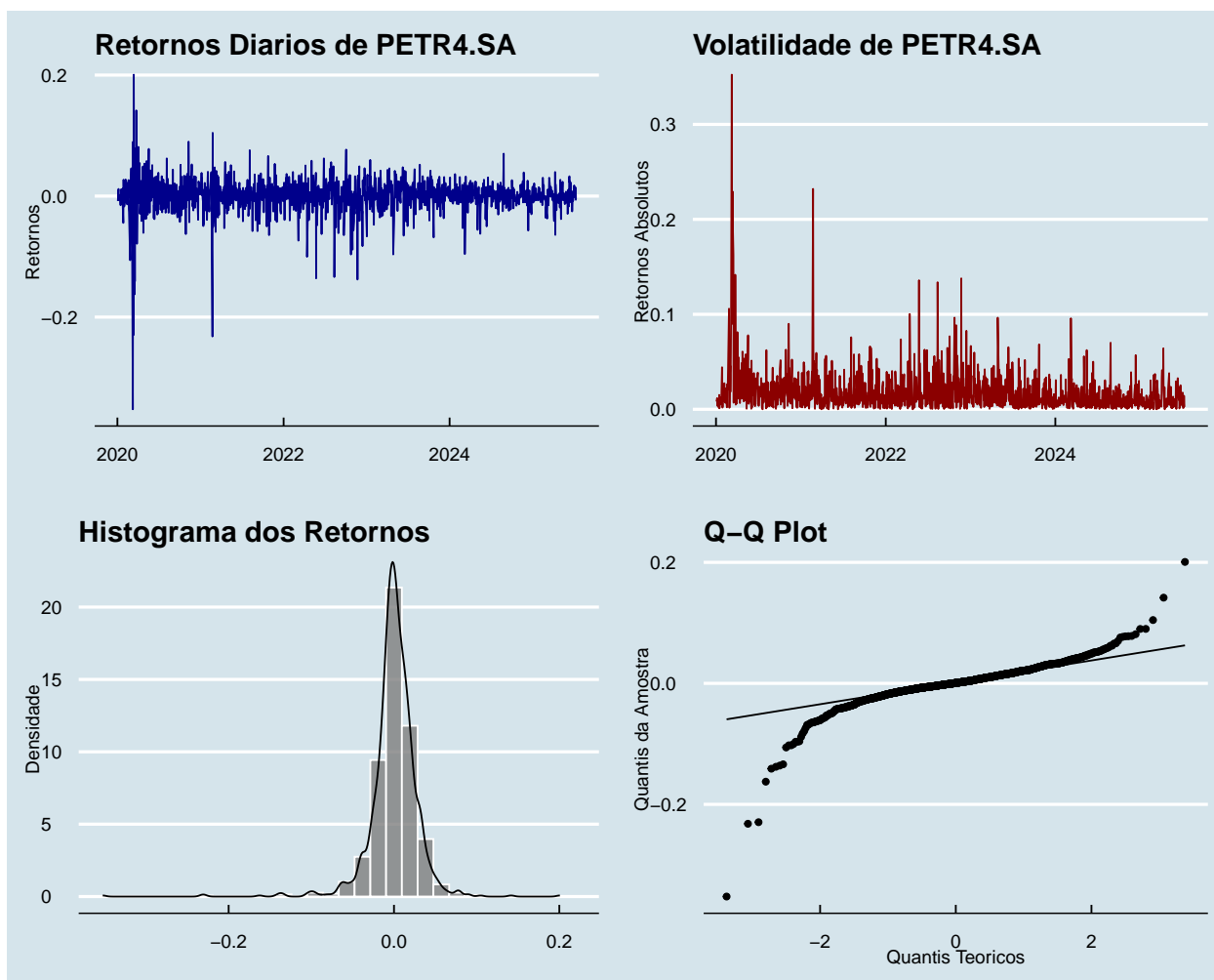
  # Gráfico dos retornos diários
  plot_returns <- ggplot(dados_ativo) + geom_line(aes(x = ref.date, y = ret.closing.prices), color = 'darkred')

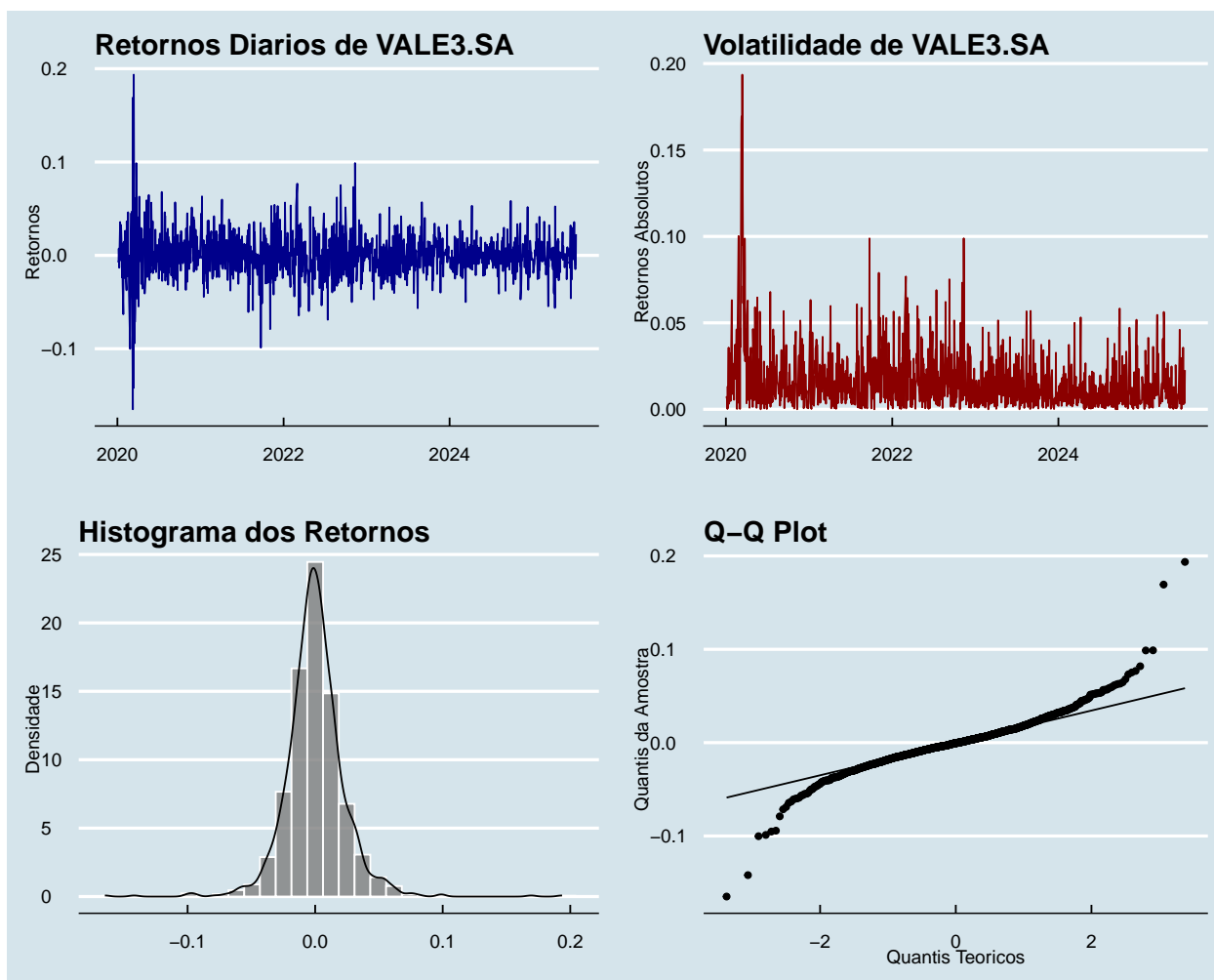
  # Gráfico da volatilidade (retornos absolutos)
  plot_volatility <- ggplot(dados_ativo) + geom_line(aes(x = ref.date, y = abs(ret.closing.prices)), color = 'darkred')

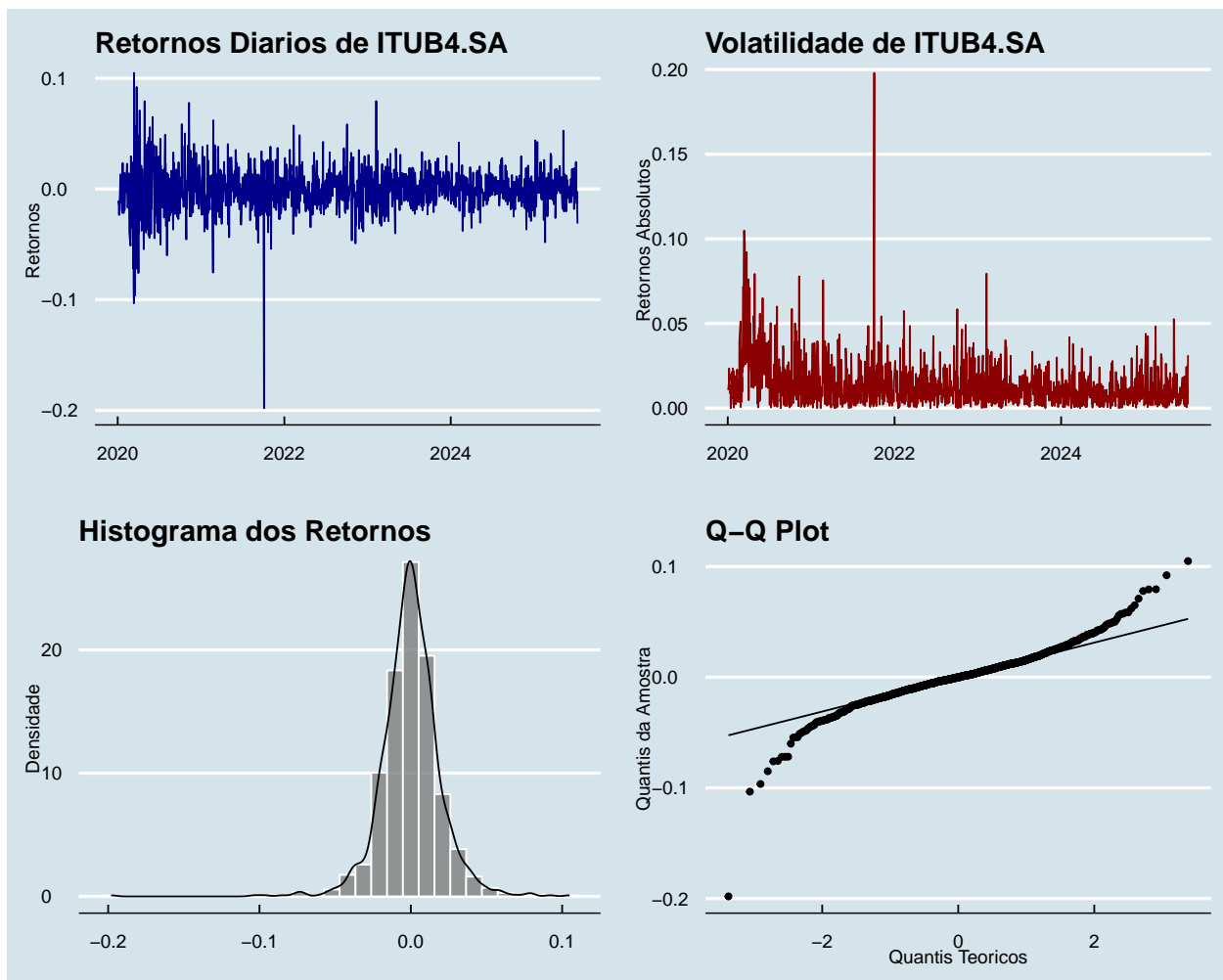
  # Histograma
  plot_histogram <- ggplot(dados_ativo) + geom_histogram(aes(x = ret.closing.prices, y = ..density..), color = 'darkred')

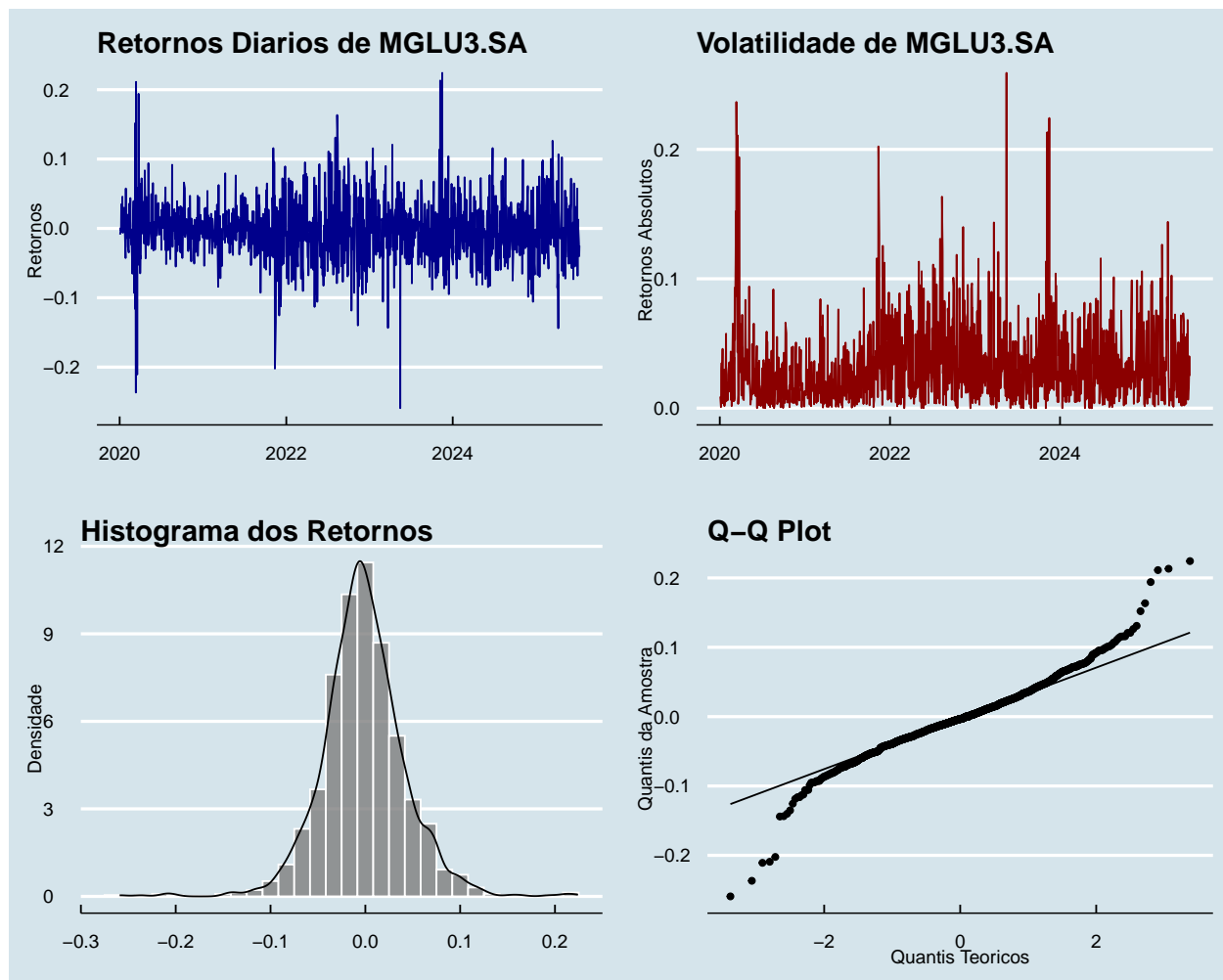
  # QQ-Plot
  plot_qq <- ggplot(dados_ativo, aes(sample = ret.closing.prices)) + stat_qq() + stat_qq_line() + labs(title = 'QQ-Plot')

  # Combinando os gráficos em um único painel
  grid_plots <- cowplot::plot_grid(plot_returns, plot_volatility, plot_histogram, plot_qq, nrow = 2)
  print(grid_plots)
}
```









3. Estatísticas Descritivas (Questão Extra 2)

Calculando as principais estatísticas descritivas para os retornos de cada um dos ativos selecionados

```
desc_stats <- assets %>%
  group_by(ticker) %>%
  summarise(
    Média = mean(ret.closing.prices, na.rm = TRUE),
    Mediana = median(ret.closing.prices, na.rm = TRUE),
    `Desvio Padrão` = sd(ret.closing.prices, na.rm = TRUE),
    Variância = var(ret.closing.prices, na.rm = TRUE),
    Assimetria = skewness(ret.closing.prices, na.rm = TRUE),
    Curtose = kurtosis(ret.closing.prices, na.rm = TRUE)
  ) %>%
  ungroup()

knitr::kable(desc_stats, caption = "Estatísticas Descritivas dos Retornos Logarítmicos")
```

Table 2: Estatísticas Descritivas dos Retornos Logarítmicos

ticker	Média	Mediana	Desvio Padrão	Variância	Assimetria	Curtose
ITUB4.SA	0.0000141	0.0000000	0.0199106	0.0003964	-0.6859280	12.767135
MGLU3.SA	-0.0019133	-0.0034187	0.0442966	0.0019622	-0.0190345	6.712002
PETR4.SA	0.0000356	0.0006859	0.0289171	0.0008362	-2.2933511	28.783304
VALE3.SA	0.0000126	-0.0005340	0.0232837	0.0005421	0.3231290	12.283590

4. Coeficiente de Correlação e Gráfico de Dispersão (Questão Extra 3)

Calculando o coeficiente de correlação entre os retornos de dois ativos (PETR4.SA e VALE3.SA) e gerando o seu respectivo gráfico de dispersão

```
# Pivotando os dados para o formato largo
assets_wide <- assets %>%
  select(ref.date, ticker, ret.closing.prices) %>%
  pivot_wider(names_from = ticker, values_from = ret.closing.prices)

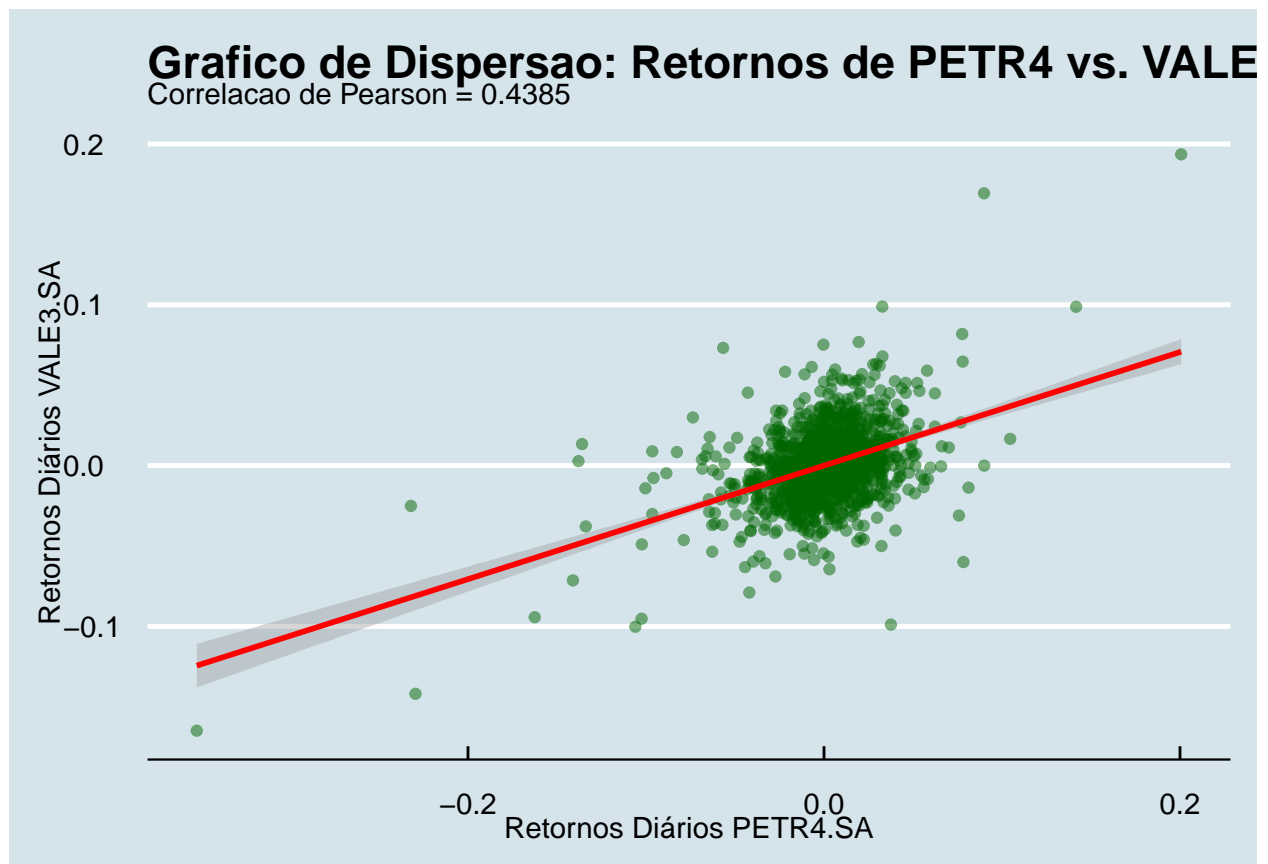
# Calculando a correlação entre PETR4 e VALE3
correlation <- cor(assets_wide$`PETR4.SA`, assets_wide$`VALE3.SA`, use = "complete.obs")

# Imprimindo o resultado
print(paste("O coeficiente de correlacao entre PETR4.SA e VALE3.SA é:", round(correlation, 4)))
```

```
## [1] "O coeficiente de correlacao entre PETR4.SA e VALE3.SA é: 0.4385"
```

```
scatter_plot <- ggplot(assets_wide, aes(x = `PETR4.SA`, y = `VALE3.SA`)) +
  geom_point(alpha = 0.5, color = "#006600") +
  geom_smooth(method = "lm", col = "red") +
  labs(
    title = "Grafico de Dispersao: Retornos de PETR4 vs. VALE3",
    subtitle = paste("Correlacao de Pearson =", round(correlation, 4)),
    x = "Retornos Diários PETR4.SA",
    y = "Retornos Diários VALE3.SA"
  ) +
  theme_economist()

print(scatter_plot)
```



5. Mapa de Calor das Correlações (Questão Extra 4)

Calculamos a matriz de correlação para todos os quatro ativos escolhidos e a visualizamos como um mapa de calor

```
# Selecionando apenas as colunas de retorno do data frame largo
returns_data <- assets_wide %>% select(all_of(tickers))

# Calculando a matriz de correlação
cor_matrix <- cor(returns_data, use = "complete.obs")
print("Matriz de Correlacao:")
```

```
## [1] "Matriz de Correlacao:"
```

```
knitr::kable(cor_matrix)
```

	PETR4.SA	VALE3.SA	ITUB4.SA	MGLU3.SA
PETR4.SA	1.0000000	0.4384517	0.4567476	0.2497458
VALE3.SA	0.4384517	1.0000000	0.3393682	0.2439033
ITUB4.SA	0.4567476	0.3393682	1.0000000	0.3286491
MGLU3.SA	0.2497458	0.2439033	0.3286491	1.0000000


```
# Criando o mapa de calor
corrplot(cor_matrix,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "black",
  tl.col = "black",
  tl.srt = 45,
  diag = FALSE,
  title = "Mapa de Calor das Correlacoes entre os Ativos",
  mar = c(0,0,1,0))
```

