

```
library(BatchGetSymbols)
library(dplyr)
library(ggplot2)
library(cowplot)
library(zoo)

# Definir ticker do Ibovespa
ticker_ibovespa <- "^BVSP"

# Coletar dados históricos
ibovespa_data <- BatchGetSymbols(
  tickers = ticker_ibovespa,
  first.date = '2021-01-01',
  last.date = Sys.Date(),
  type.return = "log",
  freq.data = "daily"
)

# Extrair o dataframe do Ibovespa
ibovespa_df <- ibovespa_data$df.tickers %>%
  filter(ticker == ticker_ibovespa)
```

T3 - Modelo ARMA na prática

```
library(ggplot2)
library(cowplot)
library(zoo)
library(dplyr)
library(ggthemes) # Para o tema economist

# Função para criar gráficos
make_plots <- function(data, ticker) {
  # Filtrar por ticker
  asset_data <- data %>% filter(ticker == ticker)

  # Calcular retornos diários
  daily_returns <- asset_data %>%
    select(ref.date, ret.closing.prices)

  # Criar gráficos com cor roxa

  # Preços
  p_prices <- ggplot(asset_data, aes(x = ref.date, y = price.close)) +
    geom_line(color = 'purple') +
    labs(x = "", y = "Preço Fechamento", title = paste("Cotações Diárias de", ticker),
         subtitle = paste("Período:", min(asset_data$ref.date), "a", max(asset_data$ref.date)),
         caption = "Fonte: Yahoo Finance") +
    theme_economist()

  # Retornos Diários
  p_returns <- ggplot(daily_returns, aes(x = ref.date, y = ret.closing.prices)) +
    geom_line(color = 'purple') +
    labs(x = "", y = 'Retornos', title = paste("Retornos de", ticker),
         subtitle = paste("Período:", min(daily_returns$ref.date), "a", max(daily_returns$ref.date)),
         caption = "Fonte: Yahoo Finance") +
    theme_economist()

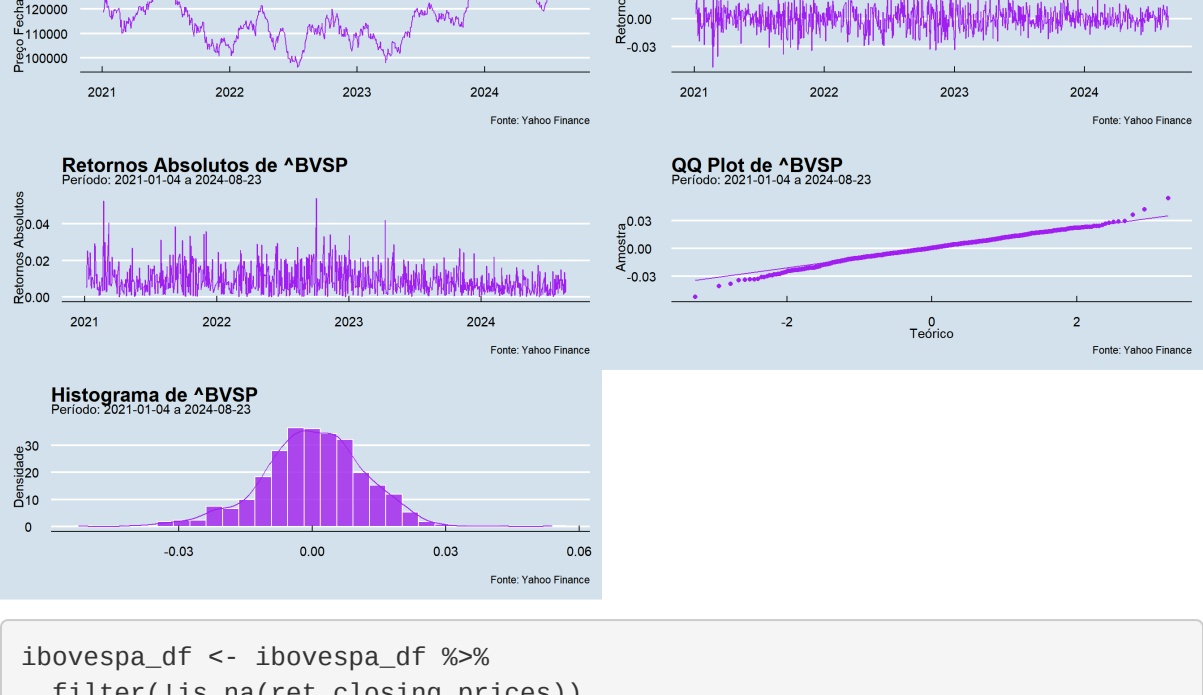
  # Retornos Absolutos
  p_volatility <- ggplot(daily_returns, aes(x = ref.date, y = abs(ret.closing.prices))) +
    geom_line(color = 'purple') +
    labs(x = "", y = 'Retornos Absolutos', title = paste("Retornos Absolutos de", ticker),
         subtitle = paste("Período:", min(daily_returns$ref.date), "a", max(daily_returns$ref.date)),
         caption = "Fonte: Yahoo Finance") +
    theme_economist()

  # QQ Plot
  p_qqplot <- ggplot(daily_returns, aes(sample = ret.closing.prices)) +
    stat_qq(color = 'purple') +
    stat_qq_line(color = 'purple') +
    labs(x = "Teórico", y = 'Amostra', title = paste("QQ Plot de", ticker),
         subtitle = paste("Período:", min(daily_returns$ref.date), "a", max(daily_returns$ref.date)),
         caption = "Fonte: Yahoo Finance") +
    theme_economist()

  # Histograma
  p_histogram <- ggplot(daily_returns) +
    geom_histogram(aes(x = ret.closing.prices, y = ..density..),
                   color = "white", fill = 'purple', linetype = "solid", alpha = 0.8) +
    geom_density(aes(x = ret.closing.prices, y = ..density..), color = 'purple') +
    labs(x = "", y = 'Densidade', title = paste("Histograma de", ticker),
         subtitle = paste("Período:", min(daily_returns$ref.date), "a", max(daily_returns$ref.date)),
         caption = "Fonte: Yahoo Finance") +
    theme_economist()

  # Organizar gráficos em uma única visualização
  plot_grid(p_prices, p_returns, p_volatility, p_qqplot, p_histogram, nrow = 3)
}

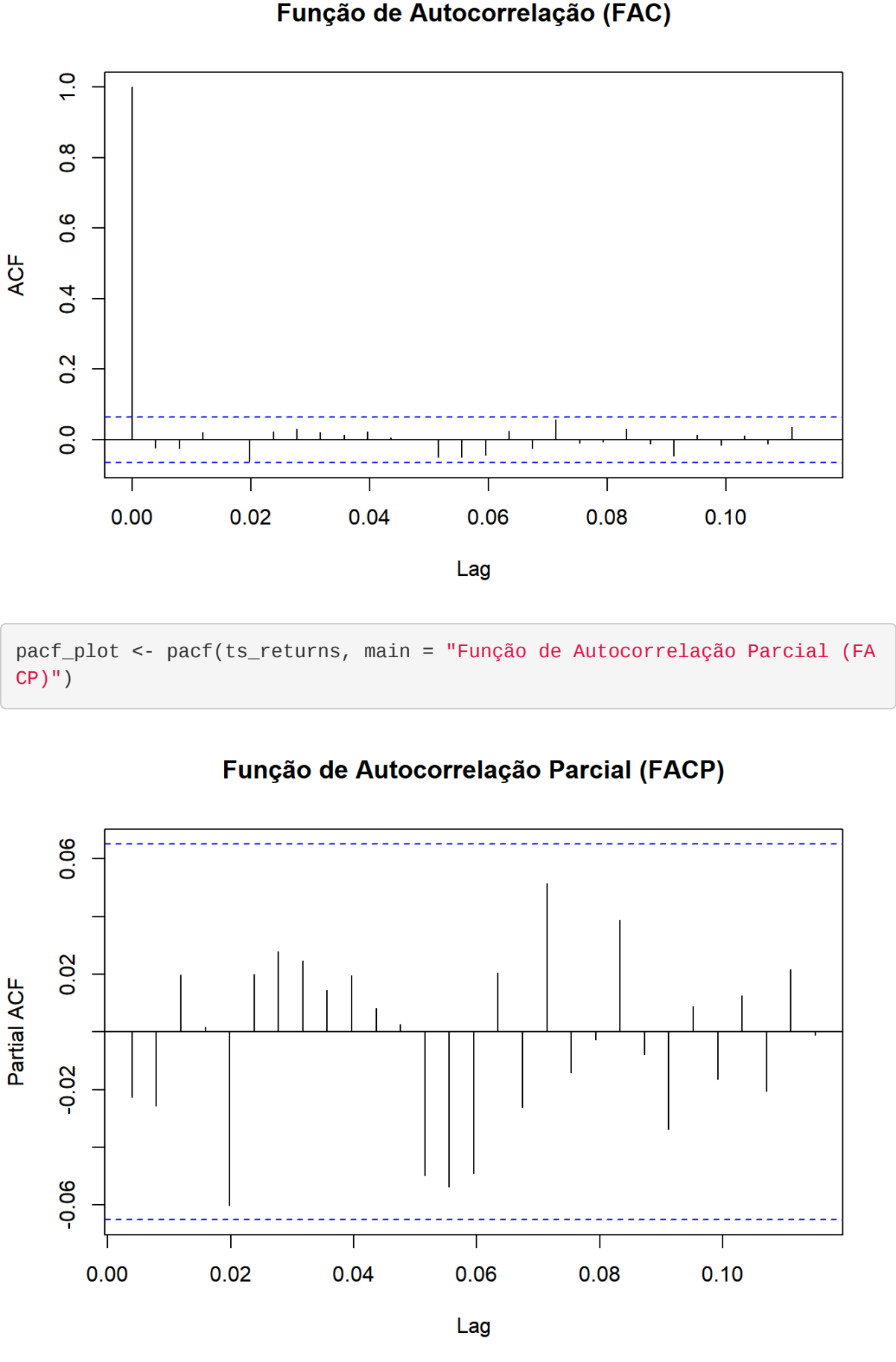
# Exemplo de chamada para gerar gráficos para o ticker específico
print(make_plots(ibovespa_df, ticker_ibovespa))
```



```
ibovespa_df <- ibovespa_df %>%
  filter(!is.na(ret.closing.prices))

# Converter para uma série temporal
ts_returns <- ts(ibovespa_df$ret.closing.prices, frequency = 252) # Assume-se 252 dias úteis por ano

# Plotar a FAC e a FACP
acf_plot <- acf(ts_returns, main = "Função de Autocorrelação (FAC)")
```



Como o PAC cai drasticamente no primeiro Lag e a FACP oscila muito, podemos assumir (por enquanto) que o melhor modelo seria um MA(1)