

# P4: Teoria Moderna de Portifolio de Markowitz e VaR

Raphael Marques Franco

2025-11-27

## 1. Carregamento de Pacotes

```
# Instalar pacotes necessários (execute uma vez):
# install.packages("yfR")
# install.packages("GetBCBData")
# install.packages("moments")
# install.packages("nortest")
# install.packages("ggplot2")
# install.packages("tidyverse")
# install.packages("dplyr")
# install.packages("knitr")
# install.packages("zoo")
# install.packages("quadprog")
# install.packages("xts")
# install.packages("PerformanceAnalytics")

# Pacotes para coleta de dados financeiros
library(yfR)          # API otimizada para B3 (Brasil)
library(GetBCBData)    # Coleta de dados do Banco Central (SELIC)
library(quantmod)

# Pacotes para análise estatística
library(moments)       # Testes de assimetria e curtose
library(nortest)        # Testes de normalidade (Anderson-Darling, etc)

# Pacotes para visualização e manipulação
library(ggplot2)        # Gráficos
library(tidyverse)       # Manipulação de dados
library(dplyr)           # Manipulação de dados
library(knitr)            # Tabelas formatadas
library(zoo)              # Séries temporais (para na.locf)
library(quadprog)         # Otimização quadrática (Fronteira Eficiente)
library(xts)                # Séries temporais estendidas
library(kableExtra)        # Para formatar tabelas em PDF
```

## 2. Definição de Parâmetros

```
# Ações a serem analisadas
acoes <- c("POM03.SA", "AURE3.SA", "BBAS3.SA", "ITUB3.SA")
nomes_acoes <- c("POM03", "AURE3", "BBAS3", "ITUB3")

# Período de análise
data_inicio <- as.Date("2024-11-18")
data_fim <- as.Date("2025-11-17")

# Parâmetros para VaR
investimento <- 100000 # R$ 100.000,00
confianca <- 0.99      # 99% de confiança
janela <- 252          # 252 dias úteis (1 ano)
```

## 3. Leitura dos Dados do CSV

```
# Ler o arquivo CSV
# Importante: ajustar o separador conforme o formato do arquivo
dados_csv <- read.csv("data.csv",
                      sep = ";",                  # Separador de colunas
                      dec = ",",                  # Separador decimal
                      stringsAsFactors = FALSE)

# Remover a primeira coluna vazia se existir
if(names(dados_csv)[1] == "" || names(dados_csv)[1] == "X") {
  dados_csv <- dados_csv[, -1]
}

# Converter coluna de Data para formato Date
dados_csv$Data <- as.Date(dados_csv$Data, format = "%d/%m/%Y")

# Converter colunas numéricas (remover vírgulas e converter para numeric)
colunas_numericas <- c("POM03", "AURE3", "BBAS3", "ITUB3", "IBOV", "SELIC")

for(col in colunas_numericas) {
  if(col %in% names(dados_csv)) {
    # Substituir vírgula por ponto e converter para numérico
    dados_csv[[col]] <- as.numeric(gsub(", ", ".", dados_csv[[col]])))
  }
}

# Ordenar por data
dados_csv <- dados_csv[order(dados_csv$Data), ]

# Exibir informações sobre os dados carregados
cat("### Dados Carregados do CSV:\n\n")

## ### Dados Carregados do CSV:
```

```

cat("Total de observações:", nrow(dados_csv), "\n")

## Total de observações: 250

cat("Período:", min(dados_csv$Data), "a", max(dados_csv$Data), "\n")

## Período: 20045 a 20409

cat("Variáveis:", paste(names(dados_csv), collapse = ", "), "\n\n")

## Variáveis: Data, POMO3, AURE3, BBAS3, ITUB3, IBOV, SELIC

# Exibir primeiras linhas
kable(head(dados_csv, 10), digits = 4,
      caption = "Primeiras 10 linhas dos dados")

```

Table 1: Primeiras 10 linhas dos dados

| Data       | POMO3  | AURE3   | BBAS3   | ITUB3   | IBOV     | SELIC |
|------------|--------|---------|---------|---------|----------|-------|
| 2024-11-18 | 6.4824 | 9.7349  | 24.0307 | 24.5180 | 127768.2 | 11.15 |
| 2024-11-19 | 6.5005 | 9.8243  | 24.1615 | 24.7836 | 128197.2 | 11.15 |
| 2024-11-21 | 6.5912 | 9.6653  | 23.6101 | 24.4018 | 126922.1 | 11.15 |
| 2024-11-22 | 6.6002 | 9.8739  | 23.9279 | 24.4848 | 129125.5 | 11.15 |
| 2024-11-25 | 6.5368 | 10.3110 | 24.0681 | 24.5429 | 129036.1 | 11.15 |
| 2024-11-26 | 6.5640 | 10.4997 | 24.4522 | 24.8915 | 129922.4 | 11.15 |
| 2024-11-27 | 6.4161 | 10.0726 | 24.0236 | 24.2607 | 127668.6 | 11.15 |
| 2024-11-28 | 5.9446 | 9.7349  | 23.3187 | 23.6216 | 124610.4 | 11.15 |
| 2024-11-29 | 5.8983 | 9.7945  | 23.5949 | 23.6714 | 125667.8 | 11.15 |
| 2024-12-02 | 5.8798 | 9.4667  | 23.4044 | 23.3788 | 125235.5 | 11.15 |

## 4. Preparação dos Dados para Análise

```

# Criar objetos xts para as séries temporais
# Preços das ações
precos <- xts(dados_csv[, nomes_acoes], order.by = dados_csv$Data)
colnames(precos) <- nomes_acoes

# IBOVESPA
ibovespa <- xts(dados_csv$IBOV, order.by = dados_csv$Data)
colnames(ibovespa) <- "IBOVESPA"

# SELIC (já em % a.a. no CSV)
# Converter para taxa diária equivalente
selic_anual <- dados_csv$SELIC
selic_diaria <- (1 + selic_anual/100)^(1/252) - 1

# Criar dataframe com informações da SELIC

```

```

selic_data <- data.frame(
  ref.date = dados_csv$Data,
  value = selic_anual,
  taxa_diaria = selic_diaria
)

cat("\n### Resumo dos Dados Preparados:\n\n")

## 
## ### Resumo dos Dados Preparados:

cat("Taxa SELIC média (% a.a.):", round(mean(selic_anual, na.rm = TRUE), 2), "\n")

## Taxa SELIC média (% a.a.): 13.95

cat("Taxa SELIC atual (% a.a.):", tail(selic_anual, 1), "\n")

## Taxa SELIC atual (% a.a.): 14.9

cat("Taxa diária equivalente:",
    round(tail(selic_diaria, 1) * 100, 4), "%\n\n")

## Taxa diária equivalente: 0.0551 %

# Resumo dos preços
resumo_precos <- data.frame(
  Acao = nomes_acoes,
  Preco_Inicial = as.numeric(precos[1, ]),
  Preco_Final = as.numeric(precos[nrow(precos), ]),
  Retorno_Periodo = ((as.numeric(precos[nrow(precos), ]) /
    as.numeric(precos[1, ])) - 1) * 100
)

kable(resumo_precos, digits = 2,
      caption = "Resumo dos Preços das Ações")

```

Table 2: Resumo dos Preços das Ações

| Acao  | Preco_Inicial | Preco_Final | Retorno_Periodo |
|-------|---------------|-------------|-----------------|
| POMO3 | 6.48          | 6.33        | -2.35           |
| AURE3 | 9.73          | 11.09       | 13.92           |
| BBAS3 | 24.03         | 22.50       | -6.37           |
| ITUB3 | 24.52         | 35.64       | 45.36           |

## 4. Visualização dos Dados Coletados

### 4.1 Gráficos de Preços de Fechamento

```

# Converter xts para dataframe para ggplot2
precos_df <- data.frame(Data = index(precos), coredata(precos))

# Transformar para formato longo
precos_long <- precos_df %>%
  pivot_longer(cols = -Data, names_to = "Acao", values_to = "Preco")

# Gráfico de preços
ggplot(precos_long, aes(x = Data, y = Preco, color = Acao)) +
  geom_line(linewidth = 1) +
  facet_wrap(~Acao, scales = "free_y", ncol = 2) +
  labs(title = "Evolução dos Preços de Fechamento",
       subtitle = paste0("Período: ", data_inicio, " a ", data_fim),
       x = "Data",
       y = "Preço de Fechamento (R$)") +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 10))

```

## Evolução dos Preços de Fechamento

Período: 2024-11-18 a 2025-11-17



### 4.2 Gráfico do IBOVESPA

```
# Gráfico do IBOVESPA
ibov_df <- data.frame(Data = index(ibovespa), IBOVESPA = coredata(ibovespa))

ggplot(ibov_df, aes(x = Data, y = IBOVESPA)) +
  geom_line(color = "darkblue", linewidth = 1) +
  labs(title = "Evolução do IBOVESPA",
       x = "Data",
       y = "Pontos") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
```

## Evolução do IBOVESPA



## 5. Cálculo dos Retornos Logarítmicos

```
# Retornos logarítmicos das ações
# ln(Pt/Pt-1) - mais apropriado para análise estatística
retornos <- na.omit(diff(log(precos)))

# Retorno do IBOVESPA
retorno_ibov <- na.omit(diff(log(ibovespa)))
colnames(retorno_ibov) <- "IBOVESPA"

# Estatísticas descritivas dos retornos
estatisticas <- data.frame(
  Acao = nomes_acoes,
  Media = apply(retornos, 2, mean),
  Desvio_Padrao = apply(retornos, 2, sd),
  Minimo = apply(retornos, 2, min),
  Maximo = apply(retornos, 2, max),
  Assimetria = apply(retornos, 2, skewness),
  Curtose = apply(retornos, 2, kurtosis)
```

```

)
kable(estatisticas, digits = 4,
      caption = "Estatísticas Descritivas dos Retornos Diários")

```

Table 3: Estatísticas Descritivas dos Retornos Diários

|       | Acao  | Media   | Desvio_Padrao | Minimo  | Maximo | Assimetria | Curtose |
|-------|-------|---------|---------------|---------|--------|------------|---------|
| POMO3 | POMO3 | -0.0001 | 0.0240        | -0.1109 | 0.0720 | -0.5063    | 5.1452  |
| AURE3 | AURE3 | 0.0005  | 0.0208        | -0.0631 | 0.0730 | 0.2483     | 3.4995  |
| BBAS3 | BBAS3 | -0.0003 | 0.0176        | -0.1357 | 0.0463 | -2.0902    | 17.2573 |
| ITUB3 | ITUB3 | 0.0015  | 0.0122        | -0.0278 | 0.0470 | 0.2141     | 4.0322  |

## 5.1 Dataframe Completo: Dados Diários Consolidados

```

# Criar dataframe completo com todas as informações
# Alinhar as datas (alguns dados podem ter datas diferentes)

# Preparar dados das ações
dados_completos <- data.frame(Data = index(retornos))

# Adicionar preços e retornos de cada ação
for(i in 1:length(nomes_acoes)){
  # Preços
  col_preco <- paste0("Preco_", nomes_acoes[i])
  dados_completos[[col_preco]] <- as.numeric(precos[index(retornos), i])

  # Retornos
  col_retorno <- paste0("Retorno_", nomes_acoes[i])
  dados_completos[[col_retorno]] <- as.numeric(retornos[, i])
}

# Adicionar IBOVESPA
dados_completos$Preco_IBOVESPA <- as.numeric(ibovespa[index(retornos)])
dados_completos$Retorno_IBOVESPA <- as.numeric(retorno_ibov)

# Adicionar SELIC (fazer merge por data)
# A SELIC pode não ter dados para todos os dias, então usamos merge
selic_diaria <- selic_data[, c("ref.date", "taxa_diaria")]
colnames(selic_diaria) <- c("Data", "SELIC_Diaria")

dados_completos <- merge(dados_completos, selic_diaria,
                         by = "Data", all.x = TRUE)

# Preencher dias sem SELIC com última observação (forward fill)
dados_completos$SELIC_Diaria <- zoo::na.locf(dados_completos$SELIC_Diaria,
                                              na.rm = FALSE)

# Exibir primeiras linhas
kable(head(dados_completos, 10), digits = 6, booktabs = TRUE,

```

```

  caption = "Primeiras 10 linhas do DataFrame Consolidado") %>%
kable_styling(latex_options = c("scale_down", "striped", "HOLD_position"))

```

Table 4: Primeiras 10 linhas do DataF

| Data       | Preco_POMO3 | Retorno_POMO3 | Preco_AURE3 | Retorno_AURE3 | Preco_BBAS3 | Retorno_BBA |
|------------|-------------|---------------|-------------|---------------|-------------|-------------|
| 2024-11-19 | 6.500515    | 0.002793      | 9.824260    | 0.009142      | 24.16152    | 0.0054      |
| 2024-11-21 | 6.591178    | 0.013851      | 9.665323    | -0.016310     | 23.61006    | -0.0230     |
| 2024-11-22 | 6.600244    | 0.001375      | 9.873927    | 0.021353      | 23.92785    | 0.0133      |
| 2024-11-25 | 6.536780    | -0.009662     | 10.311002   | 0.043314      | 24.06805    | 0.0058      |
| 2024-11-26 | 6.563979    | 0.004152      | 10.499739   | 0.018139      | 24.45223    | 0.0158      |
| 2024-11-27 | 6.416058    | -0.022793     | 10.072598   | -0.041532     | 24.02357    | -0.0176     |
| 2024-11-28 | 5.944561    | -0.076327     | 9.734858    | -0.034106     | 23.31868    | -0.0297     |
| 2024-11-29 | 5.898336    | -0.007806     | 9.794459    | 0.006104      | 23.59492    | 0.0117      |
| 2024-12-02 | 5.879846    | -0.003140     | 9.466653    | -0.034041     | 23.40441    | -0.0081     |
| 2024-12-03 | 5.879846    | 0.000000      | 9.556054    | 0.009400      | 23.69018    | 0.0121      |

```

# Exibir últimas linhas
kable(tail(dados_completos, 10), digits = 6, booktabs = TRUE,
      caption = "Últimas 10 linhas do DataFrame Consolidado") %>%
kable_styling(latex_options = c("scale_down", "striped", "HOLD_position"))

```

Table 5: Últimas 10 linhas do Dat

| Data           | Preco_POMO3 | Retorno_POMO3 | Preco_AURE3 | Retorno_AURE3 | Preco_BBAS3 | Retorno_BBA |
|----------------|-------------|---------------|-------------|---------------|-------------|-------------|
| 240 2025-11-04 | 6.55        | 0.012289      | 11.11       | 0.025526      | 22.27       | -           |
| 241 2025-11-05 | 6.58        | 0.004570      | 11.60       | 0.043159      | 22.61       | -           |
| 242 2025-11-06 | 6.38        | -0.030867     | 11.67       | 0.006016      | 22.86       | -           |
| 243 2025-11-07 | 6.37        | -0.001569     | 11.33       | -0.029567     | 22.89       | -           |
| 244 2025-11-10 | 6.57        | 0.030914      | 11.55       | 0.019231      | 22.78       | -           |
| 245 2025-11-11 | 6.50        | -0.010712     | 11.53       | -0.001733     | 23.47       | -           |
| 246 2025-11-12 | 6.38        | -0.018634     | 11.45       | -0.006963     | 22.80       | -           |
| 247 2025-11-13 | 6.39        | 0.001566      | 11.28       | -0.014958     | 22.50       | -           |
| 248 2025-11-14 | 6.46        | 0.010895      | 11.17       | -0.009800     | 22.44       | -           |
| 249 2025-11-17 | 6.33        | -0.020329     | 11.09       | -0.007188     | 22.50       | -           |

```

# Salvar o dataframe completo (opcional)
# write.csv(dados_completos, "dados_completos.csv", row.names = FALSE)

```

## 5.2 Visualização dos Retornos Diários

```

# Preparar dados para visualização
retornos_df <- data.frame(Data = index(retornos), coredata(retornos))
retornos_long <- retornos_df %>%
  pivot_longer(cols = -Data, names_to = "Acao", values_to = "Retorno")

# Gráfico de retornos diários

```

```
ggplot(retornos_long, aes(x = Data, y = Retorno, color = Acao)) +
  geom_line(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  facet_wrap(~Acao, ncol = 2) +
  labs(title = "Retornos Logarítmicos Diários",
       subtitle = "Retorno = ln(Pt / Pt-1)",
       x = "Data",
       y = "Retorno") +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 10))
```

## Retornos Logarítmicos Diários

$$\text{Retorno} = \ln(P_t / P_{t-1})$$



**Interpretação:** Retornos com assimetria próxima de 0 e curtose próxima de 3 indicam distribuição mais próxima da normal. Assimetria negativa indica maior probabilidade de retornos negativos extremos. Curtose > 3 indica caudas pesadas (maior probabilidade de eventos extremos).

### 5.3 Comparação: Retornos vs IBOVESPA

```
# Combinar retornos das ações com IBOVESPA
retornos_completo <- merge(retornos, retorno_ibov)
retornos_comp_df <- data.frame(Data = index(retornos_completo),
```

```

coredata(retornos_completo))

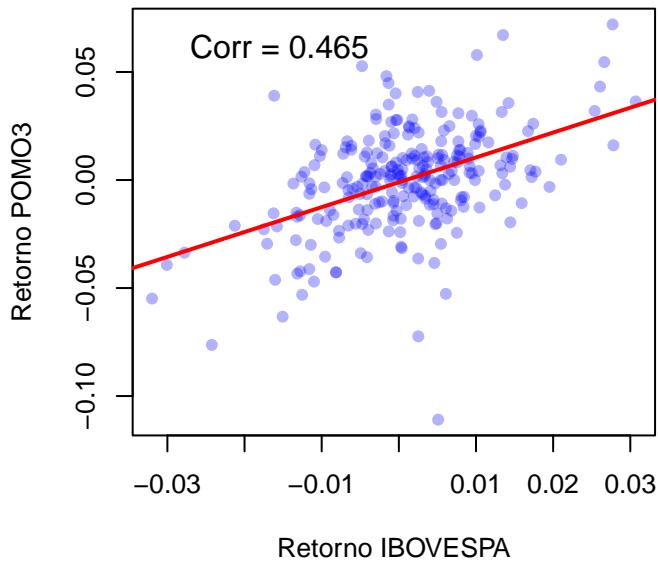
# Gráficos de dispersão: Ação vs IBOVESPA
par(mfrow = c(2, 2))
for(i in 1:length(nomes_acoes)){
  plot(retornos_comp_df$IBOVESPA, retornos_comp_df[, nomes_acoes[i]],
    main = paste(nomes_acoes[i], "vs IBOVESPA"),
    xlab = "Retorno IBOVESPA",
    ylab = paste("Retorno", nomes_acoes[i]),
    pch = 16, col = rgb(0, 0, 1, 0.3))

  # Linha de regressão
  abline(lm(retornos_comp_df[, nomes_acoes[i]] ~ retornos_comp_df$IBOVESPA),
    col = "red", lwd = 2)

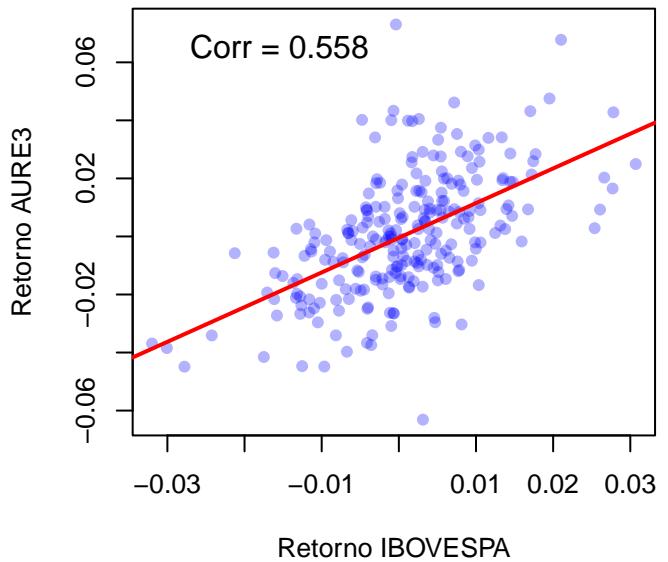
  # Calcular correlação
  cor_val <- cor(retornos_comp_df[, nomes_acoes[i]],
    retornos_comp_df$IBOVESPA)
  legend("topleft", legend = paste("Corr =", round(cor_val, 3)),
    bty = "n", cex = 1.2)
}

```

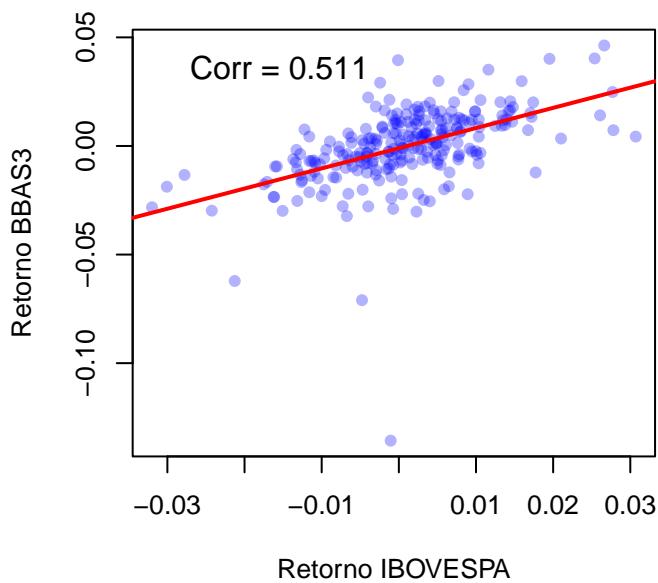
**POMO3 vs IBOVESPA**



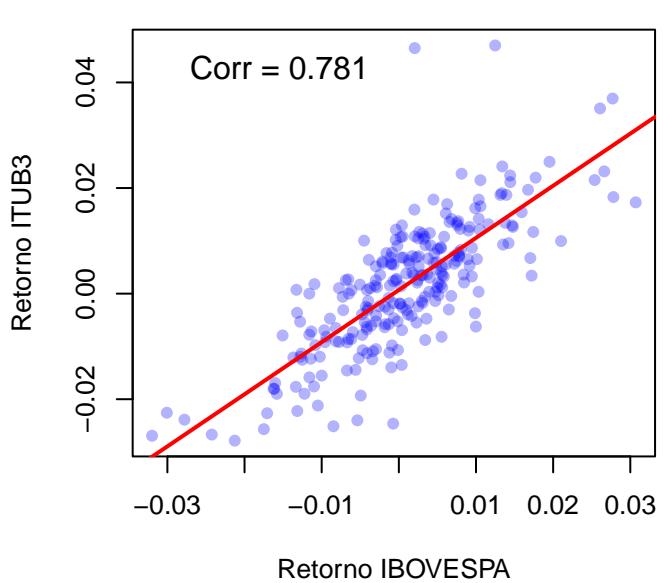
**AURE3 vs IBOVESPA**



**BBAS3 vs IBOVESPA**



**ITUB3 vs IBOVESPA**



## 6. Testes de Normalidade

```
# Função para realizar múltiplos testes de normalidade
testar_normalidade <- function(dados, nome){
  cat("\n### Testes de Normalidade para", nome, "\n\n")
```

```

# Teste de Shapiro-Wilk (mais poderoso para amostras pequenas)
# H0: dados seguem distribuição normal
# p-valor < 0.05: rejeita H0 (dados NÃO são normais)
sw <- shapiro.test(as.numeric(dados))
cat("**Shapiro-Wilk:**\n")
cat(" Estatística W =", round(sw$statistic, 4), "\n")
cat(" p-valor =", format.pval(sw$p.value, digits = 4), "\n")
cat(" Conclusão:", ifelse(sw$p.value > 0.05,
                           "Não rejeita normalidade",
                           "Rejeita normalidade"), "\n\n")

# Teste de Anderson-Darling (sensível às caudas)
ad <- ad.test(as.numeric(dados))
cat("**Anderson-Darling:**\n")
cat(" Estatística A =", round(ad$statistic, 4), "\n")
cat(" p-valor =", format.pval(ad$p.value, digits = 4), "\n")
cat(" Conclusão:", ifelse(ad$p.value > 0.05,
                           "Não rejeita normalidade",
                           "Rejeita normalidade"), "\n\n")

# Teste de Jarque-Bera (baseado em assimetria e curtose)
jb <- jarque.test(as.numeric(dados))
cat("**Jarque-Bera:**\n")
cat(" Estatística JB =", round(jb$statistic, 4), "\n")
cat(" p-valor =", format.pval(jb$p.value, digits = 4), "\n")
cat(" Conclusão:", ifelse(jb$p.value > 0.05,
                           "Não rejeita normalidade",
                           "Rejeita normalidade"), "\n\n")

return(c(SW = sw$p.value, AD = ad$p.value, JB = jb$p.value))
}

# Aplicar testes para cada ação
resultados_testes <- data.frame()
for(i in 1:ncol(retornos)){
  p_valores <- testar_normalidade(retornos[,i], nomes_acoes[i])
  resultados_testes <- rbind(resultados_testes, p_valores)
}

## 
## #### Testes de Normalidade para POMO3
##
## **Shapiro-Wilk:**
##   Estatística W = 0.9723
##   p-valor = 0.0000896
##   Conclusão: Rejeita normalidade
##
## **Anderson-Darling:**
##   Estatística A = 1.6133
##   p-valor = 0.0003722
##   Conclusão: Rejeita normalidade
##
## **Jarque-Bera:**
```

```

##  Estatística JB = 58.3806
##  p-valor = 0.000000000002103
##  Conclusão: Rejeita normalidade
##
## 
## #### Testes de Normalidade para AURE3
##
## **Shapiro-Wilk:** 
##  Estatística W = 0.9931
##  p-valor = 0.2996
##  Conclusão: Não rejeita normalidade
##
## **Anderson-Darling:** 
##  Estatística A = 0.4199
##  p-valor = 0.3233
##  Conclusão: Não rejeita normalidade
##
## **Jarque-Bera:** 
##  Estatística JB = 5.1469
##  p-valor = 0.07627
##  Conclusão: Não rejeita normalidade
##
## 
## #### Testes de Normalidade para BBAS3
##
## **Shapiro-Wilk:** 
##  Estatística W = 0.8705
##  p-valor = 0.000000000001113
##  Conclusão: Rejeita normalidade
##
## **Anderson-Darling:** 
##  Estatística A = 3.2295
##  p-valor = 0.00000004122
##  Conclusão: Rejeita normalidade
##
## **Jarque-Bera:** 
##  Estatística JB = 2290.238
##  p-valor = < 0.0000000000000022
##  Conclusão: Rejeita normalidade
##
## 
## #### Testes de Normalidade para ITUB3
##
## **Shapiro-Wilk:** 
##  Estatística W = 0.9823
##  p-valor = 0.003477
##  Conclusão: Rejeita normalidade
##
## **Anderson-Darling:** 
##  Estatística A = 0.6738
##  p-valor = 0.0776
##  Conclusão: Não rejeita normalidade
##
## **Jarque-Bera:** 

```

```

##  Estatística JB = 12.9568
##  p-valor = 0.001536
##  Conclusão: Rejeita normalidade

rownames(resultados_testes) <- nomes_acoes
kable(resultados_testes, digits = 4,
      caption = "P-valores dos Testes de Normalidade")

```

Table 6: P-valores dos Testes de Normalidade

|       | X0.0000895999492476336 | X0.000372231682844326 | X0.000000000000210276240864005 |
|-------|------------------------|-----------------------|--------------------------------|
| POMO3 | 0.0001                 | 0.0004                | 0.0000                         |
| AURE3 | 0.2996                 | 0.3233                | 0.0763                         |
| BBAS3 | 0.0000                 | 0.0000                | 0.0000                         |
| ITUB3 | 0.0035                 | 0.0776                | 0.0015                         |

#### Critérios de Interpretação:

- **p-valor > 0.05:** Não rejeitamos H<sub>0</sub>, dados podem ser considerados normais (ao nível de 5%)
- **p-valor > 0.10:** Evidência fraca contra normalidade - dados razoavelmente normais
- **p-valor entre 0.01 e 0.05:** Evidência moderada contra normalidade
- **p-valor < 0.01:** Forte evidência contra normalidade - dados claramente não normais

**Consenso:** Se pelo menos 2 dos 3 testes tiverem p-valor > 0.05, podemos considerar que a hipótese de normalidade é razoável para fins práticos.

## 6. Visualização da Distribuição

```

par(mfrow = c(2, 2))

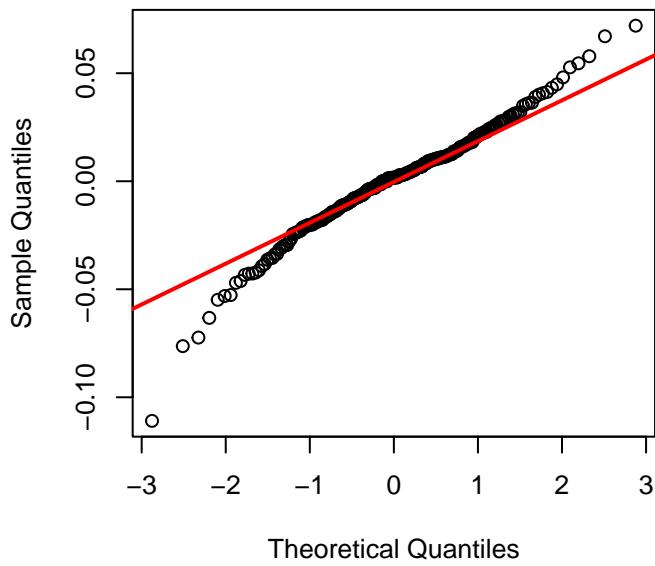
for(i in 1:ncol(retornos)){
  # Q-Q Plot (comparação com distribuição normal)
  qqnorm(retornos[,i], main = paste("Q-Q Plot -", nomes_acoes[i]))
  qqline(retornos[,i], col = "red", lwd = 2)

  # Histograma com curva normal sobreposta
  hist(retornos[,i], breaks = 30, probability = TRUE,
       main = paste("Histograma -", nomes_acoes[i]),
       xlab = "Retorno", ylab = "Densidade", col = "lightblue")

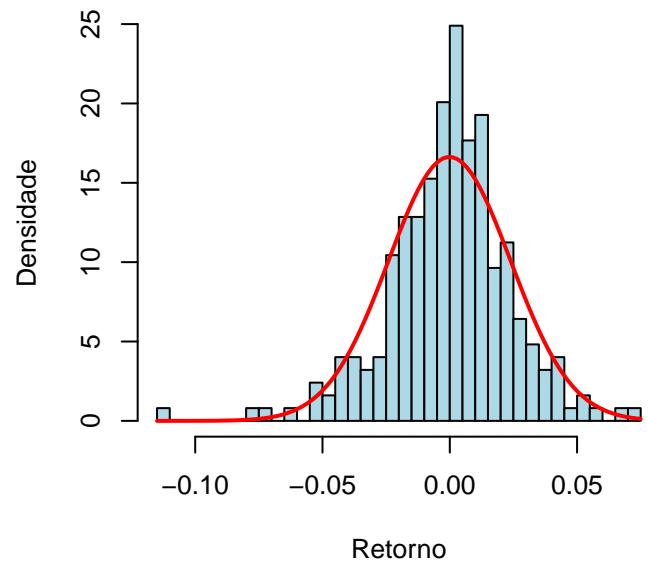
  # Curva normal teórica
  curve(dnorm(x, mean = mean(retornos[,i]), sd = sd(retornos[,i])),
         add = TRUE, col = "red", lwd = 2)
}

```

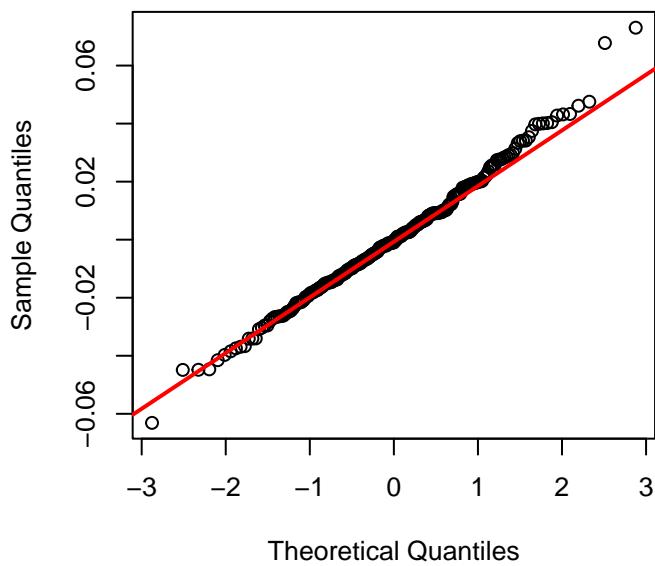
**Q-Q Plot – POMO3**



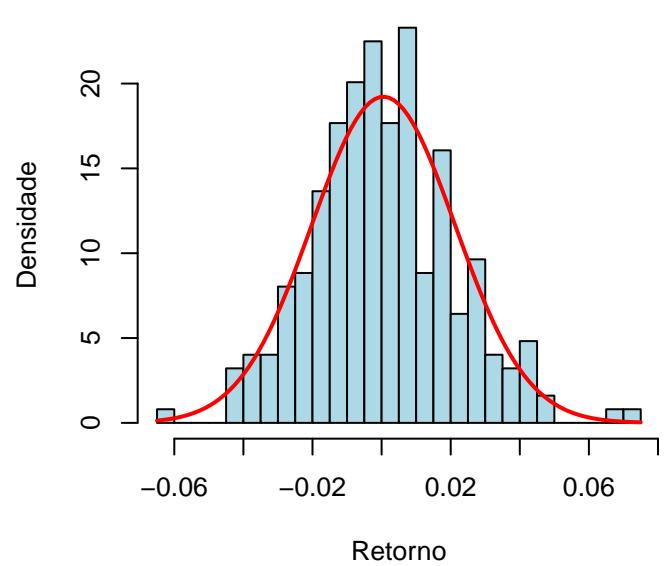
**Histograma – POMO3**

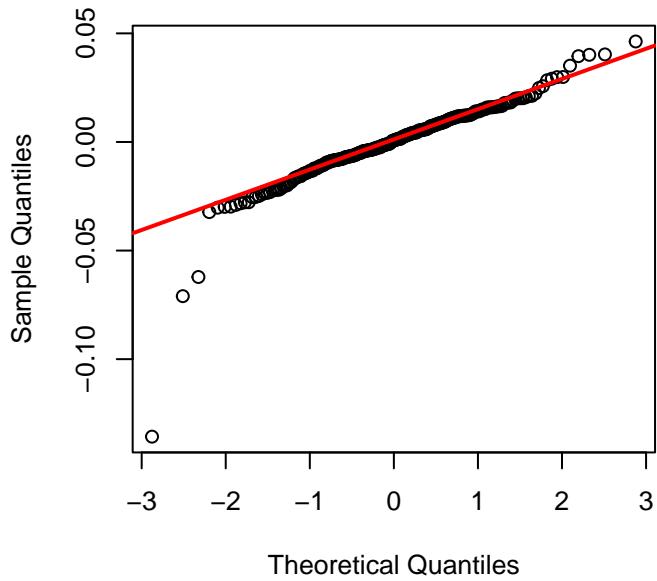
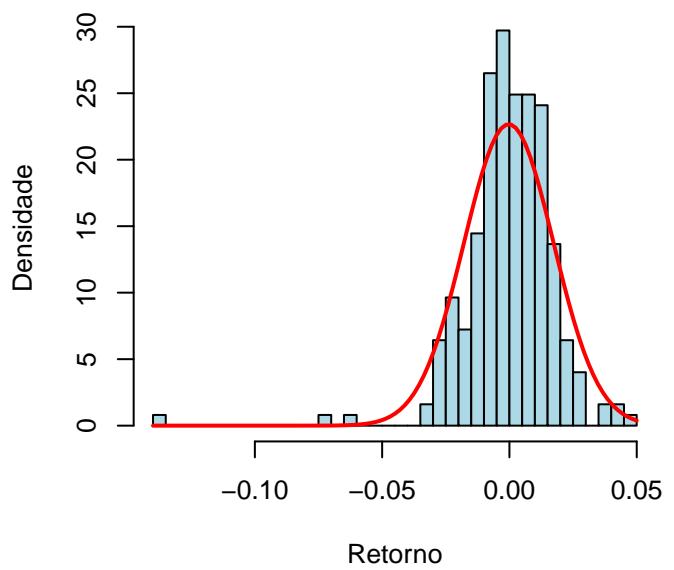
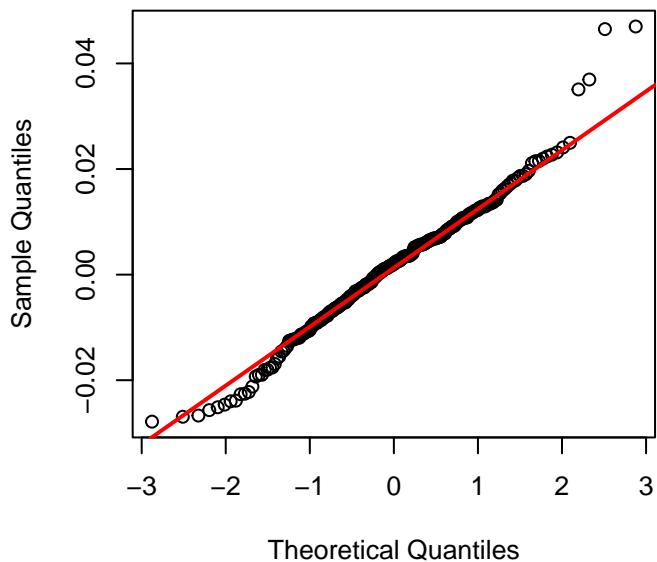
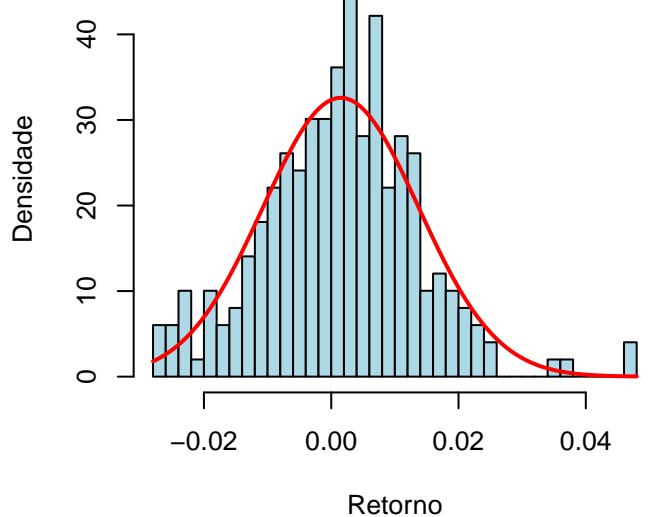


**Q-Q Plot – AURE3**



**Histograma – AURE3**



**Q-Q Plot – BBAS3****Histograma – BBAS3****Q-Q Plot – ITUB3****Histograma – ITUB3**

**Interpretação Q-Q Plot:** Pontos próximos à linha vermelha indicam normalidade. Desvios nas caudas indicam caudas mais pesadas (curtose) ou assimétricas.

## 8. Fronteira Eficiente (Markowitz)

```
# Pacote para otimização de portfólio
library(quadprog)

# Matriz de covariância dos retornos
cov_matrix <- cov(retornos)

# Vetor de retornos médios
retornos_medios <- colMeans(retornos)

# Função para calcular portfólio de variância mínima
calc_portfolio <- function(retornos_alvo, retornos_medios, cov_matrix){
  n <- length(retornos_medios)

  # Matriz D (2x covariância para função quadrática)
  Dmat <- 2 * cov_matrix

  # Vetor d (zeros para minimização de variância)
  dvec <- rep(0, n)

  # Restrições
  # 1. Soma dos pesos = 1
  # 2. Retorno esperado = retorno alvo
  # 3. Pesos >= 0 (sem venda a descoberto)
  Amat <- cbind(rep(1, n), retornos_medios, diag(n))
  bvec <- c(1, retornos_alvo, rep(0, n))

  # Resolver problema de otimização quadrática
  resultado <- tryCatch({
    solve.QP(Dmat, dvec, Amat, bvec, meq = 2)
  }, error = function(e){
    return(NULL)
  })

  if(is.null(resultado)) return(NULL)

  # Retornar pesos, retorno e risco
  pesos <- resultado$solution
  retorno <- sum(pesos * retornos_medios)
  risco <- sqrt(t(pesos) %*% cov_matrix %*% pesos)

  return(list(pesos = pesos, retorno = retorno, risco = as.numeric(risco)))
}

# Gerar fronteira eficiente
# Retornos alvo: do mínimo ao máximo possível
retorno_min <- min(retornos_medios)
retorno_max <- max(retornos_medios)
retornos_alvo <- seq(retorno_min, retorno_max, length.out = 100)

# Calcular portfólios para cada retorno alvo
fronteira <- data.frame()
```

```

for(ret_alvo in retornos_alvo){
  portfolio <- calc_portfolio(ret_alvo, retornos_medios, cov_matrix)

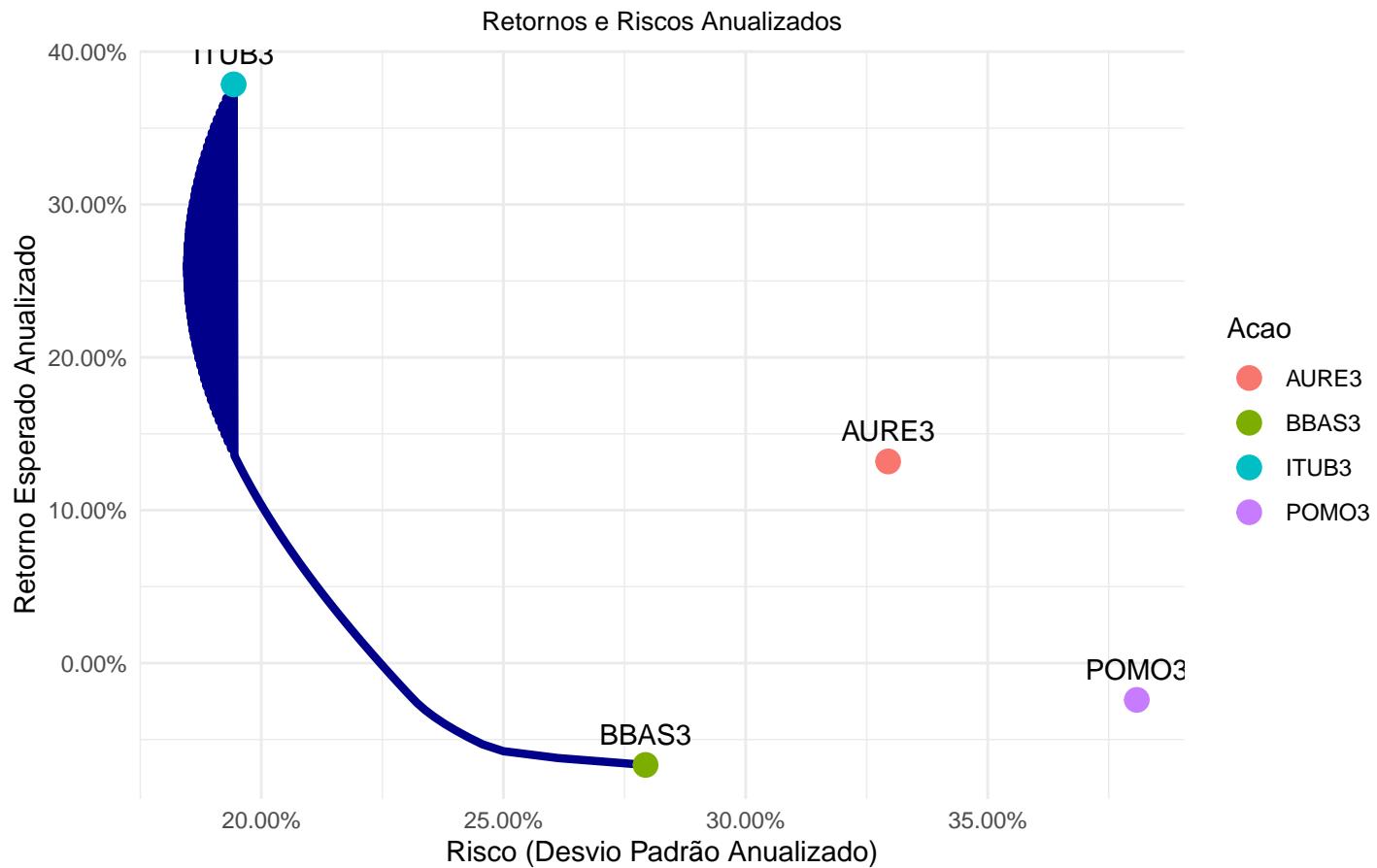
  if(!is.null(portfolio)){
    fronteira <- rbind(fronteira, data.frame(
      Retorno = portfolio$retorno * 252, # Anualizar (252 dias úteis)
      Risco = portfolio$risco * sqrt(252) # Anualizar
    ))
  }
}

# Adicionar pontos das ações individuais (anualizados)
acoes_individuais <- data.frame(
  Acao = nomes_acoes,
  Retorno = retornos_medios * 252,
  Risco = apply(retornos, 2, sd) * sqrt(252)
)

# Gráfico da Fronteira Eficiente
ggplot() +
  # Fronteira eficiente
  geom_line(data = fronteira, aes(x = Risco, y = Retorno),
            color = "darkblue", linewidth = 1.5) +
  # Ações individuais
  geom_point(data = acoes_individuais, aes(x = Risco, y = Retorno, color = Acao),
             size = 4) +
  geom_text(data = acoes_individuais, aes(x = Risco, y = Retorno, label = Acao),
            vjust = -1, size = 4) +
  # Formatação dos eixos para Porcentagem com 2 casas decimais
  scale_x_continuous(labels = scales::percent_format(accuracy = 0.01)) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 0.01)) +
  labs(title = "Fronteira Eficiente de Markowitz",
       subtitle = "Retornos e Riscos Anualizados",
       x = "Risco (Desvio Padrão Anualizado)",
       y = "Retorno Esperado Anualizado") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 10),
        legend.position = "right")

```

## Fronteira Eficiente de Markowitz



```
# Tabela com informações das ações
kable(acoes_individuais, digits = 4,
      caption = "Retorno e Risco Anualizados das Ações")
```

Table 7: Retorno e Risco Anualizados das Ações

|       | Acao  | Retorno | Risco  |
|-------|-------|---------|--------|
| POMO3 | POMO3 | -0.0241 | 0.3808 |
| AURE3 | AURE3 | 0.1319  | 0.3294 |
| BBAS3 | BBAS3 | -0.0666 | 0.2793 |
| ITUB3 | ITUB3 | 0.3786  | 0.1943 |

### Interpretação da Fronteira Eficiente:

A curva azul representa a combinação ótima de ativos que maximiza o retorno para cada nível de risco. Pontos acima da fronteira são impossíveis de alcançar com os ativos disponíveis, enquanto pontos abaixo da fronteira são ineficientes, pois existe uma combinação melhor que oferece maior retorno para o mesmo risco ou menor risco para o mesmo retorno. Investidores devem escolher portfólios ao longo da fronteira de acordo com sua tolerância ao risco, sendo que cada ponto da fronteira representa uma alocação ótima específica entre os quatro ativos analisados.

## 8.2 Fronteira Eficiente com ETF IBOVESPA

```
# Agora vamos expandir a análise incluindo o IBOVESPA como um ativo adicional
# Isso simula a possibilidade de investir em um ETF que replica o índice

# Combinar retornos das ações com o IBOVESPA
retornos_com_etf <- merge(retornos, retorno_ibov)
colnames(retornos_com_etf) <- c(nomes_acoes, "IBOVESPA")

# Recalcular matriz de covariância e retornos médios com ETF incluído
cov_matrix_etf <- cov(retornos_com_etf)
retornos_medios_etf <- colMeans(retornos_com_etf)

cat("\n### Matriz de Correlação com ETF IBOVESPA:\n")

## 
## ### Matriz de Correlação com ETF IBOVESPA:

cor_matrix_etf <- cor(retornos_com_etf)
kable(round(cor_matrix_etf, 3),
      caption = "Matriz de Correlação: Ações + ETF IBOVESPA")
```

Table 8: Matriz de Correlação: Ações + ETF IBOVESPA

|          | POMO3 | AURE3 | BBAS3 | ITUB3 | IBOVESPA |
|----------|-------|-------|-------|-------|----------|
| POMO3    | 1.000 | 0.316 | 0.211 | 0.386 | 0.465    |
| AURE3    | 0.316 | 1.000 | 0.281 | 0.440 | 0.558    |
| BBAS3    | 0.211 | 0.281 | 1.000 | 0.447 | 0.511    |
| ITUB3    | 0.386 | 0.440 | 0.447 | 1.000 | 0.781    |
| IBOVESPA | 0.465 | 0.558 | 0.511 | 0.781 | 1.000    |

```
# Função para calcular portfólio com ETF incluído
calc_portfolio_etf <- function(retornos_alvo, retornos_medios, cov_matrix){
  n <- length(retornos_medios)

  Dmat <- 2 * cov_matrix
  dvec <- rep(0, n)

  # Restrições: soma = 1, retorno = alvo, pesos >= 0
  Amat <- cbind(rep(1, n), retornos_medios, diag(n))
  bvec <- c(1, retornos_alvo, rep(0, n))

  resultado <- tryCatch({
    solve.QP(Dmat, dvec, Amat, bvec, meq = 2)
  }, error = function(e){
    return(NULL)
  })

  if(is.null(resultado)) return(NULL)
```

```

pesos <- resultado$solution
retorno <- sum(pesos * retornos_medios)
risco <- sqrt(t(pesos) %*% cov_matrix %*% pesos)

return(list(pesos = pesos, retorno = retorno, risco = as.numeric(risco)))
}

# Gerar nova fronteira eficiente com ETF
retorno_min_etf <- min(retornos_medios_etf)
retorno_max_etf <- max(retornos_medios_etf)
retornos_alvo_etf <- seq(retorno_min_etf, retorno_max_etf, length.out = 100)

fronteira_etf <- data.frame()

for(ret_alvo in retornos_alvo_etf){
  portfolio <- calc_portfolio_etf(ret_alvo, retornos_medios_etf, cov_matrix_etf)

  if(!is.null(portfolio)){
    fronteira_etf <- rbind(fronteira_etf, data.frame(
      Retorno = portfolio$retorno * 252,
      Risco = portfolio$risco * sqrt(252)
    ))
  }
}

# Adicionar ponto do ETF IBOVESPA
etf_ponto <- data.frame(
  Acao = "IBOVESPA (ETF)",
  Retorno = retornos_medios_etf["IBOVESPA"] * 252,
  Risco = sd(retorno_ibov) * sqrt(252)
)

# Combinar com ações individuais
todos_ativos <- rbind(acoes_individuais, etf_ponto)

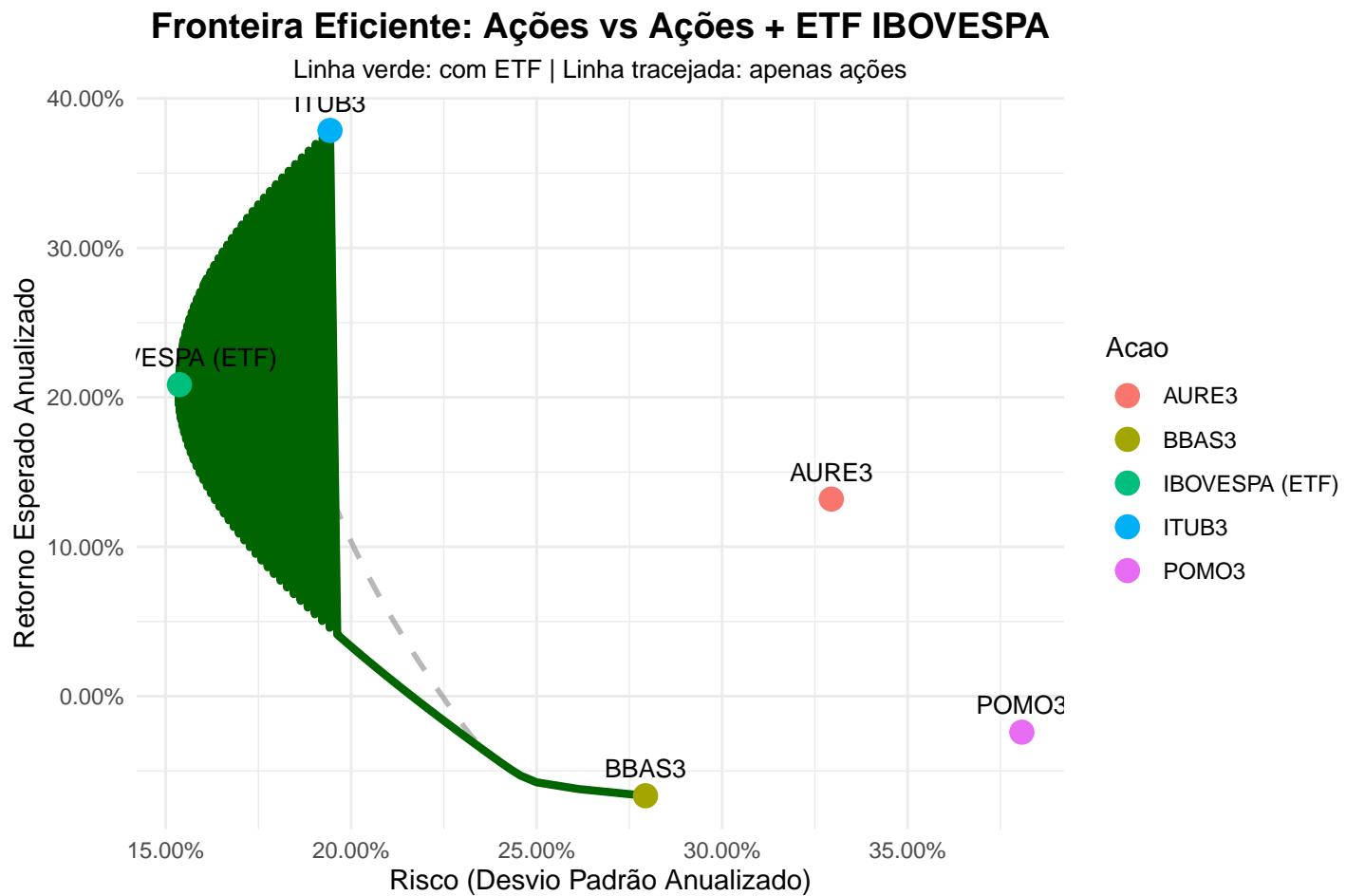
# Gráfico comparando as duas fronteiras
ggplot() +
  # Fronteira original (só ações)
  geom_line(data = fronteira, aes(x = Risco, y = Retorno),
            color = "gray60", linewidth = 1, linetype = "dashed",
            alpha = 0.7) +
  # Nova fronteira (ações + ETF)
  geom_line(data = fronteira_etf, aes(x = Risco, y = Retorno),
            color = "darkgreen", linewidth = 1.5) +
  # Todos os ativos individuais
  geom_point(data = todos_ativos, aes(x = Risco, y = Retorno, color = Acao),
             size = 4) +
  geom_text(data = todos_ativos, aes(x = Risco, y = Retorno, label = Acao),
            vjust = -1, size = 3.5, check_overlap = TRUE) +
  scale_x_continuous(labels = scales::percent_format(accuracy = 0.01)) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 0.01)) +
  labs(title = "Fronteira Eficiente: Ações vs Ações + ETF IBOVESPA",
       subtitle = "Linha verde: com ETF | Linha tracejada: apenas ações",

```

```

x = "Risco (Desvio Padrão Anualizado)",
y = "Retorno Esperado Anualizado") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
plot.subtitle = element_text(hjust = 0.5, size = 10),
legend.position = "right")

```



#### Interpretação da Diversificação com ETF:

A inclusão do ETF IBOVESPA na análise expande o conjunto de oportunidades de investimento, representado pela fronteira eficiente verde. Observe que esta nova fronteira domina a fronteira original (linha tracejada cinza) em praticamente todos os pontos, demonstrando que a adição do ETF oferece combinações superiores de risco e retorno. Isto ocorre porque o ETF IBOVESPA, sendo uma cesta diversificada de ações, apresenta características de risco e retorno que complementam as ações individuais, permitindo que o investidor alcance níveis de diversificação que seriam impossíveis usando apenas as quatro ações isoladamente.

#### 8.3 Composição Ótima: Ações + ETF

```

# Encontrar o portfólio de variância mínima com ETF
portfolio_min_var_etf <- fronteira_etf[which.min(fronteira_etf$Risco), ]

ret_alvo_min_etf <- portfolio_min_var_etf$Retorno / 252
portfolio_otimo_etf <- calc_portfolio_etf(ret_alvo_min_etf, retornos_medios_etf,
                                             cov_matrix_etf)

if(!is.null(portfolio_otimo_etf)){
  composicao_etf <- data.frame(
    Ativo = c(nomes_acoes, "IBOVESPA (ETF)"),
    Peso_Percentual = portfolio_otimo_etf$pesos * 100,
    Valor_Investido = portfolio_otimo_etf$pesos * investimento
  )

  # Filtrar apenas ativos com peso significativo (> 0.1%)
  composicao_etf_filtrada <- composicao_etf[composicao_etf$Peso_Percentual > 0.1, ]

  cat("\n### PORTFÓLIO ÓTIMO: AÇÕES + ETF IBOVESPA\n")
  kable(composicao_etf_filtrada, digits = 2, row.names = FALSE,
        caption = "Composição Ótima com ETF (Variância Mínima)")

  cat("\n**Características do Portfólio Ótimo com ETF:**\n")
  cat("- Retorno esperado anualizado:",
      round(portfolio_otimo_etf$retorno * 252 * 100, 2), "%\n")
  cat("- Risco (desvio padrão anualizado):",
      round(portfolio_otimo_etf$risco * sqrt(252) * 100, 2), "%\n")
  cat("- Sharpe Ratio (assumindo SELIC como taxa livre de risco):",
      round((portfolio_otimo_etf$retorno * 252 - mean(selic_data$value)/100) /
            (portfolio_otimo_etf$risco * sqrt(252)), 3), "\n\n")
}

## 
## ### PORTFÓLIO ÓTIMO: AÇÕES + ETF IBOVESPA
##
## 
## **Características do Portfólio Ótimo com ETF:**
## - Retorno esperado anualizado: 20.32 %
## - Risco (desvio padrão anualizado): 15.35 %
## - Sharpe Ratio (assumindo SELIC como taxa livre de risco): 0.415

```

A composição ótima que minimiza a variância mostra como o investidor deve distribuir seus recursos entre as ações individuais e o ETF IBOVESPA. Note que a otimização matemática considera não apenas o retorno esperado e o risco de cada ativo, mas também as correlações entre eles, buscando combinações que maximizem o benefício da diversificação. Ativos com peso próximo de zero ou zero indicam que, do ponto de vista da eficiência de média-variância, eles não contribuem significativamente para melhorar a relação risco-retorno do portfólio neste ponto específico da fronteira.

## 8.4 Linha de Alocação de Capital (CAL) - Portfólio com Ativo Livre de Risco

```
# A Linha de Alocação de Capital conecta o ativo livre de risco (SELIC)
# ao portfólio tangente na fronteira eficiente
# Este é o portfólio de maior Índice de Sharpe

# Taxa livre de risco (SELIC média anualizada)
taxa_livre_risco <- mean(selic_data$value) / 100

cat("\n### CONSTRUÇÃO DA LINHA DE ALOCAÇÃO DE CAPITAL\n\n")

## 
## ### CONSTRUÇÃO DA LINHA DE ALOCAÇÃO DE CAPITAL

cat("Taxa Livre de Risco (SELIC média):", round(taxa_livre_risco * 100, 2), "% a.a.\n\n")

## Taxa Livre de Risco (SELIC média): 13.95 % a.a.

# Função para calcular o Índice de Sharpe
calcular_sharpe <- function(retorno, risco, rf){
  return((retorno - rf) / risco)
}

# Encontrar o portfólio tangente (maior Sharpe Ratio)
fronteira_etf$Sharpe <- sapply(1:nrow(fronteira_etf), function(i){
  calcular_sharpe(fronteira_etf$Retorno[i], fronteira_etf$Risco[i],
                  taxa_livre_risco)
})

# Portfólio tangente
idx_tangente <- which.max(fronteira_etf$Sharpe)
portfolio_tangente <- fronteira_etf[idx_tangente, ]

cat("**Portfólio Tangente (Ótimo de Sharpe):**\n\n")

## **Portfólio Tangente (Ótimo de Sharpe):**

cat("- Retorno esperado:", round(portfolio_tangente$Retorno * 100, 2), "% a.a.\n")

## - Retorno esperado: 37.86 % a.a.

cat("- Risco:", round(portfolio_tangente$Risco * 100, 2), "% a.a.\n")

## - Risco: 19.43 % a.a.
```

```

cat("- Índice de Sharpe:", round(portfolio_tangente$Sharpe, 3), "\n\n")

## - Índice de Sharpe: 1.231

# Calcular composição do portfólio tangente
ret_alvo_tangente <- portfolio_tangente$Retorno / 252
portfolio_tangente_composicao <- calc_portfolio_etf(ret_alvo_tangente,
                                                       retornos_medios_etf,
                                                       cov_matrix_etf)

if(!is.null(portfolio_tangente_composicao)){
  composicao_tangente <- data.frame(
    Ativo = c(nomes_acoes, "IBOVESPA (ETF)"),
    Peso_Percentual = portfolio_tangente_composicao$pesos * 100,
    Valor_Investido_100k = portfolio_tangente_composicao$pesos * investimento
  )

  # Filtrar ativos com peso > 0.1%
  composicao_tangente_filtrada <- composicao_tangente[
    composicao_tangente$Peso_Percentual > 0.1,
  ]

  kable(composicao_tangente_filtrada, digits = 2, row.names = FALSE,
         caption = "Composição do Portfólio Tangente (Máximo Sharpe)")
}

```

Table 9: Composição do Portfólio Tangente (Máximo Sharpe)

| Ativo | Peso_Percentual | Valor_Investido_100k |
|-------|-----------------|----------------------|
| ITUB3 | 100             | 100000               |

```

# Criar pontos da Linha de Alocação de Capital
# CAL:  $E(R_p) = R_f + (E(R_t) - R_f) / t * p$ 
# onde  $R_t$  é o retorno do portfólio tangente

risco_cal <- seq(0, max(fronteira_etf$Risco) * 1.2, length.out = 100)
retorno_cal <- taxa_livre_risco +
  (portfolio_tangente$Retorno - taxa_livre_risco) / portfolio_tangente$Risco *
  risco_cal

linha_cal <- data.frame(Risco = risco_cal, Retorno = retorno_cal)

# Ponto do ativo livre de risco
ponto_rf <- data.frame(
  Risco = 0,
  Retorno = taxa_livre_risco,
  Label = "SELIC (Livre de Risco")
)

# Gráfico com CAL
ggplot() +

```

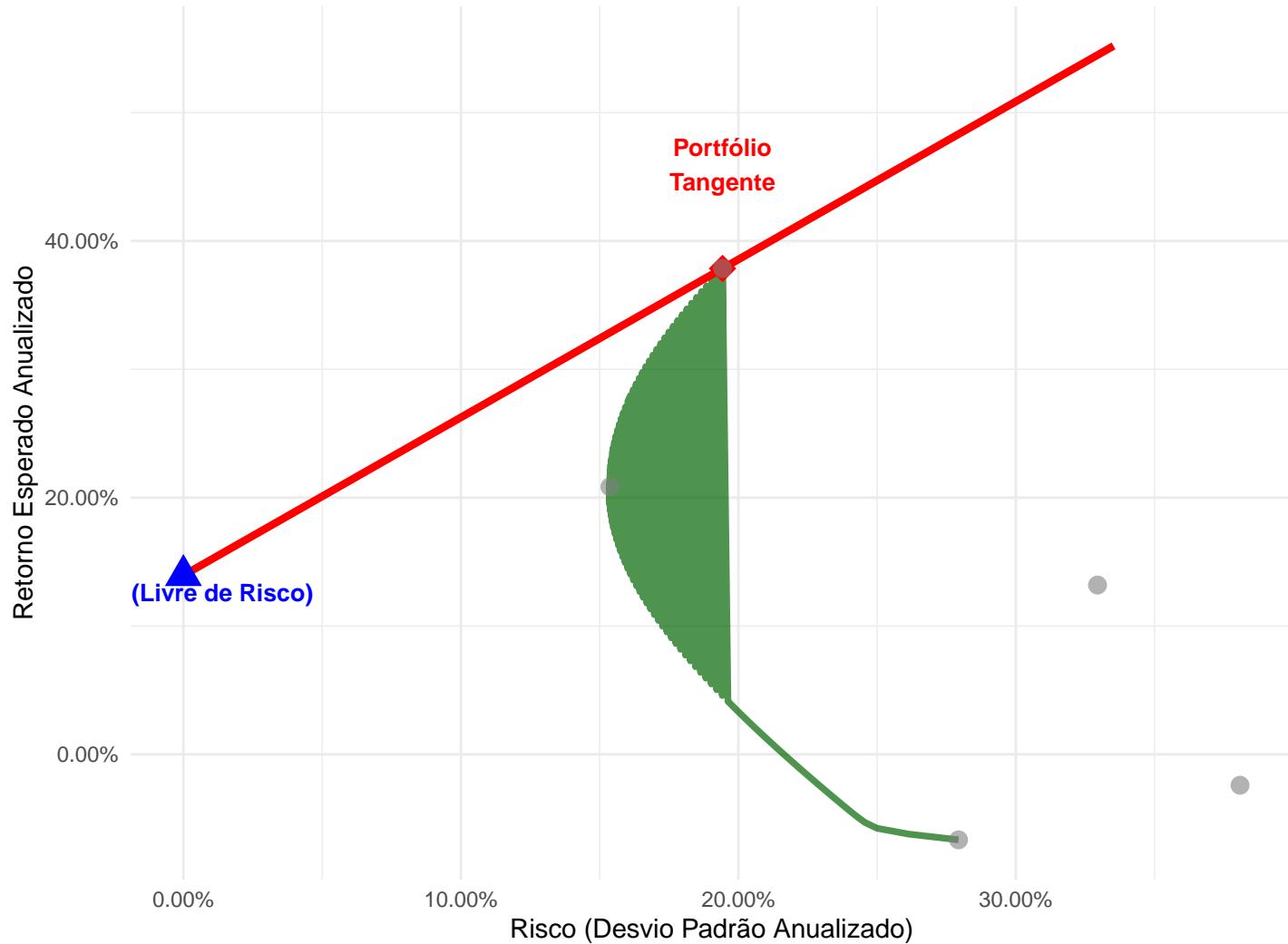
```

# Fronteira eficiente
geom_line(data = fronteira_etf, aes(x = Risco, y = Retorno),
           color = "darkgreen", linewidth = 1.3, alpha = 0.7) +
# Linha de Alocação de Capital
geom_line(data = linha_cal, aes(x = Risco, y = Retorno),
           color = "red", linewidth = 1.5) +
# Portfólio tangente
geom_point(data = portfolio_tangente, aes(x = Risco, y = Retorno),
           color = "red", size = 5, shape = 18) +
geom_text(data = portfolio_tangente, aes(x = Risco, y = Retorno),
           label = "Portfólio\nTangente", vjust = -1.5, hjust = 0.5,
           color = "red", fontface = "bold", size = 3.5) +
# Ativo livre de risco
geom_point(data = ponto_rf, aes(x = Risco, y = Retorno),
           color = "blue", size = 5, shape = 17) +
geom_text(data = ponto_rf, aes(x = Risco, y = Retorno, label = Label),
           vjust = 1.5, color = "blue", fontface = "bold", size = 3.5) +
# Ativos individuais
geom_point(data = todos_ativos, aes(x = Risco, y = Retorno),
           color = "gray50", size = 3, alpha = 0.6) +
scale_x_continuous(labels = scales::percent_format(accuracy = 0.01),
                   limits = c(0, NA)) +
scale_y_continuous(labels = scales::percent_format(accuracy = 0.01)) +
labs(title = "Linha de Alocação de Capital (CAL)",
     subtitle = "Combinações ótimas: Portfólio Tangente + SELIC",
     x = "Risco (Desvio Padrão Anualizado)",
     y = "Retorno Esperado Anualizado") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
      plot.subtitle = element_text(hjust = 0.5, size = 10))

```

## Linha de Alocação de Capital (CAL)

Combinações ótimas: Portfólio Tangente + SELIC



### Interpretação da Linha de Alocação de Capital:

A linha vermelha representa a Linha de Alocação de Capital, que mostra todas as combinações possíveis entre o ativo livre de risco (SELIC) e o portfólio tangente. Esta linha é fundamental porque domina toda a fronteira eficiente, exceto no ponto de tangência. Isso significa que, para qualquer nível de risco desejado, um investidor sempre obtém um retorno esperado maior investindo parte do capital na SELIC e parte no portfólio tangente, em vez de escolher qualquer outro ponto da fronteira eficiente.

O ponto de tangência (losango vermelho) representa o portfólio de ativos com risco que possui o maior Índice de Sharpe, ou seja, oferece a melhor recompensa por unidade de risco assumido acima da taxa livre de risco. Investidores mais conservadores devem alojar maior proporção na SELIC e menor no portfólio tangente, posicionando-se mais à esquerda da linha. Investidores mais arrojados fazem o oposto, podendo inclusive usar alavancagem para se posicionar além do portfólio tangente, mais à direita na linha.

## 8.5 Alocações Estratégicas ao Longo da CAL

```
# Vamos calcular algumas alocações específicas ao longo da CAL
# para diferentes perfis de investidores

cat("\n### ESTRATÉGIAS DE ALOCAÇÃO COM R$ 100.000,00\n\n")

## 
## ### ESTRATÉGIAS DE ALOCAÇÃO COM R$ 100.000,00

# Função para calcular alocação na CAL dado um nível de risco alvo
calcular_alocacao_cal <- function(risco_alvo, ret_tangente, risco_tangente,
                                     taxa_rf, investimento_total){

  # Peso no portfólio tangente
  w_tangente <- risco_alvo / risco_tangente

  # Peso no ativo livre de risco
  w_rf <- 1 - w_tangente

  # Retorno esperado
  retorno_esperado <- w_rf * taxa_rf + w_tangente * ret_tangente

  return(list(
    peso_rf = w_rf,
    peso_tangente = w_tangente,
    valor_rf = w_rf * investimento_total,
    valor_tangente = w_tangente * investimento_total,
    retorno_esperado = retorno_esperado,
    risco = risco_alvo
  ))
}

# Diferentes perfis de risco (em termos de desvio padrão anualizado)
perfis <- data.frame(
  Perfil = c("Conservador", "Moderado", "Arrojado", "Muito Arrojado"),
  Risco_Alvo = c(0.05, 0.10, 0.15, 0.20) # 5%, 10%, 15%, 20% de volatilidade
)

resultados_perfis <- data.frame()

for(i in 1:nrow(perfis)){
  alocacao <- calcular_alocacao_cal(
    perfis$Risco_Alvo[i],
    portfolio_tangente$Retorno,
    portfolio_tangente$Risco,
    taxa_livre_risco,
    investimento
  )

  resultados_perfis <- rbind(resultados_perfis, data.frame(
    Perfil = perfis$Perfil[i],
    Risco_Alvo = perfis$Risco_Alvo[i] * 100,
```

```

    Peso_SELIC = alocacao$peso_rf * 100,
    Peso_Portfolio_Tangente = alocacao$peso_tangente * 100,
    Valor_SELIC = alocacao$valor_rf,
    Valor_Portfolio_Tangente = alocacao$valor_tangente,
    Retorno_Esperado = alocacao$retorno Esperado * 100
  )))
}

kable(resultados_perfis, digits = 2, booktabs = TRUE, row.names = FALSE,
       caption = "Alocações Estratégicas para Diferentes Perfis de Risco")%>%
  kableExtra::kable_styling(latex_options = c("striped", "scale_down", "HOLD_position"), full_width = F

```

Table 10: Alocações Estratégicas para Diferentes Perfis de Risco

| Perfil         | Risco_Alvo | Peso_SELIC | Peso_Portfolio_Tangente | Valor_SELIC | Valor_Portfolio_Tangente |
|----------------|------------|------------|-------------------------|-------------|--------------------------|
| Conservador    | 5          | 74.27      |                         | 25.73       | 74265.44                 |
| Moderado       | 10         | 48.53      |                         | 51.47       | 48530.88                 |
| Arrojado       | 15         | 22.80      |                         | 77.20       | 22796.31                 |
| Muito Arrojado | 20         | -2.94      |                         | 102.94      | -2938.25                 |

```
cat("\n**Interpretação das Alocações:**\n\n")
```

```

## 
## **Interpretação das Alocações:** 

for(i in 1:nrow(resultados_perfis)){
  cat("**", resultados_perfis$Perfil[i], ":\n")
  cat(" - Investir R$", format(round(resultados_perfis$Valor_SELIC[i], 2),
                                big.mark = ".", decimal.mark = ","),
      "em SELIC (", round(resultados_perfis$Peso_SELIC[i], 1), "%)\n")
  cat(" - Investir R$", format(round(resultados_perfis$Valor_Portfolio_Tangente[i], 2),
                                big.mark = ".", decimal.mark = ","),
      "no Portfólio Tangente (", round(resultados_perfis$Peso_Portfolio_Tangente[i], 1), "%)\n")
  cat(" - Retorno esperado:", round(resultados_perfis$Retorno_Esperado[i], 2),
      "% a.a.\n")
  cat(" - Risco (volatilidade):", round(resultados_perfis$Risco_Alvo[i], 2),
      "% a.a.\n\n")
}

## ** Conservador :**
## - Investir R$ 74.265,44 em SELIC ( 74.3 %)
## - Investir R$ 25.734,56 no Portfólio Tangente ( 25.7 %)
## - Retorno esperado: 20.1 % a.a.
## - Risco (volatilidade): 5 % a.a.
## 
## ** Moderado :**
## - Investir R$ 48.530,88 em SELIC ( 48.5 %)
## - Investir R$ 51.469,12 no Portfólio Tangente ( 51.5 %)
## - Retorno esperado: 26.25 % a.a.
## - Risco (volatilidade): 10 % a.a.
```

```

## 
## ** Arrojado :**
##   - Investir R$ 22.796,31 em SELIC ( 22.8 %)
##   - Investir R$ 77.203,69 no Portfólio Tangente ( 77.2 %)
##   - Retorno esperado: 32.41 % a.a.
##   - Risco (volatilidade): 15 % a.a.
##
## ** Muito Arrojado :**
##   - Investir R$ -2.938,25 em SELIC ( -2.9 %)
##   - Investir R$ 102.938,2 no Portfólio Tangente ( 102.9 %)
##   - Retorno esperado: 38.56 % a.a.
##   - Risco (volatilidade): 20 % a.a.

# Detalhamento para perfil Moderado
cat("\n**Detalhamento do Perfil Moderado (R$ 100.000):**\n\n")

## 
## **Detalhamento do Perfil Moderado (R$ 100.000):**

alocacao_moderado <- calcular_alocacao_cal(0.10, portfolio_tangente$Retorno,
                                             portfolio_tangente$Risco,
                                             taxa_livre_risco, investimento)

valor_na_selic <- alocacao_moderado$valor_rf
valor_no_tangente <- alocacao_moderado$valor_tangente

cat("1. SELIC: R$", format(round(valor_na_selic, 2),
                           big.mark = ".", decimal.mark = ","), "\n\n")

## 1. SELIC: R$ 48.530,88

cat("2. Portfólio Tangente: R$", format(round(valor_no_tangente, 2),
                                         big.mark = ".", decimal.mark = ","),
    "distribuídos em:\n")

## 2. Portfólio Tangente: R$ 51.469,12 distribuídos em:

if(!is.null(portfolio_tangente_composicao)){
  for(j in 1:length(composicao_tangente_filtrada$Ativo)){
    valor_ativo <- composicao_tangente_filtrada$Peso_Percentual[j] / 100 *
      valor_no_tangente
    cat("  - ", composicao_tangente_filtrada$Ativo[j], ": R$",
        format(round(valor_ativo, 2), big.mark = ".", decimal.mark = ","),
        " (", round(composicao_tangente_filtrada$Peso_Percentual[j], 2),
        "% do portfólio tangente)\n")
  }
}

##   - ITUB3 : R$ 51.469,12  ( 100 % do portfólio tangente)

```

Esta análise final demonstra como um investidor pode implementar a estratégia ótima na prática. O portfólio tangente representa a melhor combinação de ativos com risco (ações e ETF), e a proporção entre este portfólio e a SELIC determina o perfil de risco-retorno final. O investidor conservador prioriza a proteção do capital, alocando maior parte na SELIC, enquanto o arrojado busca maximizar retornos, concentrando-se no portfólio tangente. Note que todos os perfis utilizam exatamente a mesma composição de ativos com risco (o portfólio tangente), variando apenas a proporção total investida neste portfólio versus na SELIC, demonstrando a elegância e simplicidade do modelo de Markowitz com ativo livre de risco.

## 9. Cálculo do VaR Histórico

```
# VaR Histórico: percentil (1-confiança) dos retornos históricos
# Para 99% de confiança, pegamos o 1º percentil (piores 1% dos retornos)

calcular_var <- function(retornos_acao, investimento, confianca){
  # Percentil correspondente
  percentil <- 1 - confianca

  # VaR como percentil dos retornos
  var_percentual <- quantile(retornos_acao, percentil)

  # Perda máxima esperada em reais
  perda_maxima <- -var_percentual * investimento

  return(list(
    VaR_percentual = var_percentual,
    Perda_Maxima_Reais = perda_maxima,
    Percentil = percentil * 100
  ))
}

# Calcular VaR para cada ação
resultados_var <- data.frame()

for(i in 1:ncol(retornos)){
  var_resultado <- calcular_var(retornos[,i], investimento, confianca)

  resultados_var <- rbind(resultados_var, data.frame(
    Acao = nomes_acoes[i],
    VaR_Percentual = var_resultado$VaR_percentual,
    Perda_Maxima_R = var_resultado$Perda_Maxima_Reais,
    Investimento_R = investimento,
    Perda_Percentual = (var_resultado$Perda_Maxima_Reais / investimento) * 100
  ))
}

kable(resultados_var, digits = 2,
      caption = paste0("VaR Histórico (", confianca*100, "% de confiança) - Janela de ",
                      janela, " dias úteis"))
```

Table 11: VaR Histórico (99% de confiança) - Janela de 252 dias úteis

| Acao     | VaR_Percentual | Perda_Maxima_R | Investimento_R | Perda_Percentual |
|----------|----------------|----------------|----------------|------------------|
| 1% POMO3 | -0.07          | 6798.65        | 100000         | 6.80             |
| 1% AURE3 | -0.04          | 4476.00        | 100000         | 4.48             |
| 1% BBAS3 | -0.05          | 4782.39        | 100000         | 4.78             |
| 1% ITUB3 | -0.03          | 2620.28        | 100000         | 2.62             |

## 9.1 VaR do Portfólio Ótimo (Variância Mínima)

```

# Calcular portfólio de variância mínima global
# Este é o ponto mais à esquerda da fronteira eficiente

# Encontrar o portfólio com menor risco
portfolio_min_var <- fronteira[which.min(fronteira$Risco), ]

# Para calcular VaR do portfólio, precisamos dos pesos ótimos
# Vamos recalcular para o retorno correspondente
ret_alvo_min <- portfolio_min_var$Retorno / 252 # Desanualizar

portfolio_otimo <- calc_portfolio(ret_alvo_min, retornos_medios, cov_matrix)

if(!is.null(portfolio_otimo)){
  # Calcular retornos do portfólio
  pesos_matrix <- matrix(portfolio_otimo$pesos, ncol = 1)
  retornos_portfolio <- as.matrix(retornos) %*% pesos_matrix

  # Calcular VaR do portfólio
  var_portfolio <- calcular_var(retornos_portfolio, investimento, confianca)

  # Exibir composição do portfólio
  composicao <- data.frame(
    Acao = nomes_acoes,
    Peso = portfolio_otimo$pesos * 100,
    Valor_Investido = portfolio_otimo$pesos * investimento
  )

  cat("\n### PORTFÓLIO DE VARIÂNCIA MÍNIMA\n\n")
  kable(composicao, digits = 2,
        caption = "Composição do Portfólio Ótimo (Variância Mínima)")

  cat("\n**VaR do Portfólio Ótimo:**\n")
  cat("- Retorno esperado diário:", round(portfolio_otimo$retorno * 100, 4), "%\n")
  cat("- Risco (desvio padrão):", round(portfolio_otimo$risco * 100, 4), "%\n")
  cat("- VaR (99%):", round(var_portfolio$VaR_percentual * 100, 2), "%\n")
  cat("- Perda máxima esperada: R$",
      format(round(var_portfolio$Perda_Maxima_Reais, 2),
      big.mark = ".", decimal.mark = ","), "\n\n")

  # Comparação com investimento em ações individuais
}

```

```

cat("## Comparação: O portfólio diversificado reduz o risco comparado\n")
cat("ao investimento em ações individuais, demonstrando o benefício da diversificação.\n")
}

## 
## #### PORTFÓLIO DE VARIÂNCIA MÍNIMA
##
##
## **VaR do Portfólio Ótimo:**
## - Retorno esperado diário: 0.102 %
## - Risco (desvio padrão): 1.1636 %
## - VaR (99%): -2.73 %
## - Perda máxima esperada: R$ 2.734,72
##
## **Comparação:** O portfólio diversificado reduz o risco comparado
## ao investimento em ações individuais, demonstrando o benefício da diversificação.

```

## 10. Interpretação dos Resultados

```
cat("\n## INTERPRETAÇÃO DO VALUE AT RISK (VaR)\n\n")
```

```

## 
## ## INTERPRETAÇÃO DO VALUE AT RISK (VaR)

cat("Com um investimento de R$", format(investimento, big.mark = ".", decimal.mark = ","),
    "em cada ação:\n\n")

## Com um investimento de R$ 100.000 em cada ação:

for(i in 1:nrow(resultados_var)){
  cat("### ", resultados_var$Acao[i], "\n")
  cat("- VaR (99%): ", round(resultados_var$VaR_Percentual[i] * 100, 2), "%\n")
  cat("- Perda máxima esperada: R$",
      format(round(resultados_var$Perda_Maxima_R[i], 2),
             big.mark = ".", decimal.mark = ","), "\n")
  cat("- Interpretação: Com 99% de confiança, a perda não deve exceder R$",
      format(round(resultados_var$Perda_Maxima_R[i], 2),
             big.mark = ".", decimal.mark = ","),
      "em um dia de negociação.\n")
  cat("- Percentual do investimento:",
      round(resultados_var$Perda_Percentual[i], 2), "%\n\n")
}

## #### POM03
## - VaR (99%): -6.8 %
## - Perda máxima esperada: R$ 6.798,65
## - Interpretação: Com 99% de confiança, a perda não deve exceder R$ 6.798,65 em um dia de negociação.
## - Percentual do investimento: 6.8 %
```

```

## 
## ### AURE3
## - VaR (99%): -4.48 %
## - Perda máxima esperada: R$ 4.476
## - Interpretação: Com 99% de confiança, a perda não deve exceder R$ 4.476 em um dia de negociação.
## - Percentual do investimento: 4.48 %
##
## ### BBAS3
## - VaR (99%): -4.78 %
## - Perda máxima esperada: R$ 4.782,39
## - Interpretação: Com 99% de confiança, a perda não deve exceder R$ 4.782,39 em um dia de negociação.
## - Percentual do investimento: 4.78 %
##
## ### ITUB3
## - VaR (99%): -2.62 %
## - Perda máxima esperada: R$ 2.620,28
## - Interpretação: Com 99% de confiança, a perda não deve exceder R$ 2.620,28 em um dia de negociação.
## - Percentual do investimento: 2.62 %

```

## 9. Visualização Comparativa do VaR

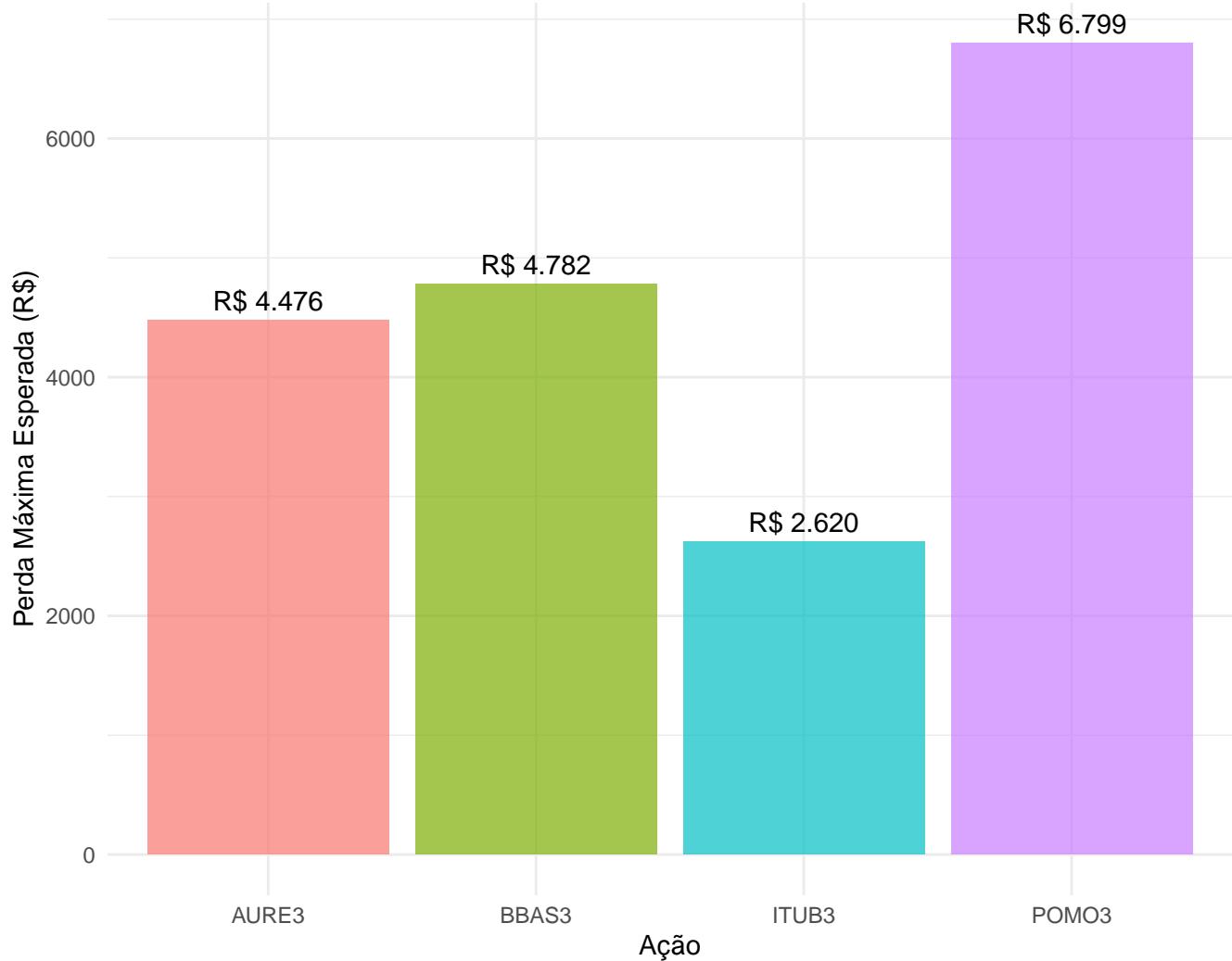
```

# Gráfico de barras comparativo
ggplot(resultados_var, aes(x = Acao, y = Perda_Maxima_R, fill = Acao)) +
  geom_bar(stat = "identity", alpha = 0.7) +
  geom_text(aes(label = paste0("R$ ", format(round(Perda_Maxima_R, 0),
                                              big.mark = "."))),
            vjust = -0.5, size = 4) +
  labs(title = "Comparação do VaR (99%) entre as Ações",
       subtitle = paste0("Perda máxima esperada com investimento de R$ ",
                        format(investimento, big.mark = ".")),
       x = "Ação",
       y = "Perda Máxima Esperada (R$)") +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 10))

```

## Comparação do VaR (99%) entre as Ações

Perda máxima esperada com investimento de R\$ 100.000



## 10. Conclusões

```
cat("\n## CONCLUSÕES PRINCIPAIS\n\n")

##
## ## CONCLUSÕES PRINCIPAIS

# Ação com maior risco
acao_maior_risco <- resultados_var$Acao[which.max(resultados_var$Perda_Maxima_R)]
maior_perda <- max(resultados_var$Perda_Maxima_R)

# Ação com menor risco
acao_menor_risco <- resultados_var$Acao[which.min(resultados_var$Perda_Maxima_R)]
```

```

menor_perda <- min(resultados_var$Perda_Maxima_R)

cat("1. **Maior Risco:** ", acao_maior_risco,
    " - Perda máxima de R$", format(round(maior_perda, 2),
                                         big.mark = ".", decimal.mark = ","), "\n\n")

## 1. **Maior Risco:** POMO3 - Perda máxima de R$ 6.798,65

cat("2. **Menor Risco:** ", acao_menor_risco,
    " - Perda máxima de R$", format(round(menor_perda, 2),
                                         big.mark = ".", decimal.mark = ","), "\n\n")

## 2. **Menor Risco:** ITUB3 - Perda máxima de R$ 2.620,28

cat("3. **Sobre a Normalidade:** Retornos financeiros geralmente apresentam:\n")

## 3. **Sobre a Normalidade:** Retornos financeiros geralmente apresentam:

cat(" - Caudas mais pesadas que a distribuição normal\n")

## - Caudas mais pesadas que a distribuição normal

cat(" - Assimetria (mais retornos negativos extremos)\n")

## - Assimetria (mais retornos negativos extremos)

cat(" - Mesmo com rejeição da normalidade, o VaR histórico é robusto\n\n")

## - Mesmo com rejeição da normalidade, o VaR histórico é robusto

cat("4. **Limitações do VaR Histórico:**\n")

## 4. **Limitações do VaR Histórico:**

cat(" - Assume que o passado se repete\n")

## - Assume que o passado se repete

cat(" - Não captura eventos extremos não observados no período\n")

## - Não captura eventos extremos não observados no período

cat(" - Não mede perdas além do percentil (tail risk)\n")

## - Não mede perdas além do percentil (tail risk)

```

---

**Nota Metodológica:** Este relatório utiliza o VaR Histórico não-paramétrico, que não assume normalidade dos retornos. Os testes de normalidade servem para verificar se métodos paramétricos (VaR Delta-Normal) seriam apropriados. Para gestão de risco conservadora, o VaR Histórico é preferível quando a normalidade é rejeitada.