

Relatorio

Captura dos dados

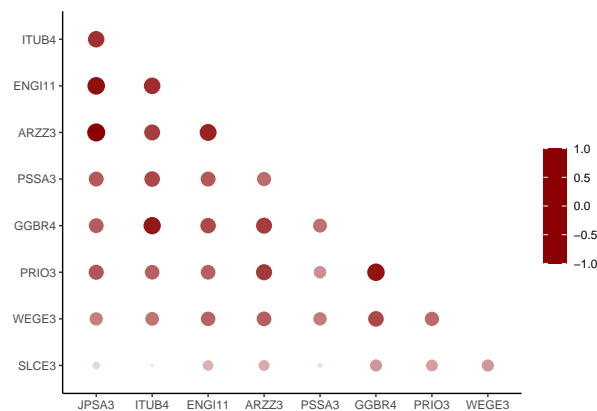
Foi feita através da biblioteca do R `quantmod`. A nossa ideia foi fazer o balanceamento da carteira recomendada (10SIM) pelo BTG do mês de novembro. O horizonte temporal foi de 01-01-2019 a 30-11-2021.

Cálculo dos retornos

Primeiro calculos os retornos diários dos dados importados.

Avaliar as correlações dos retornos

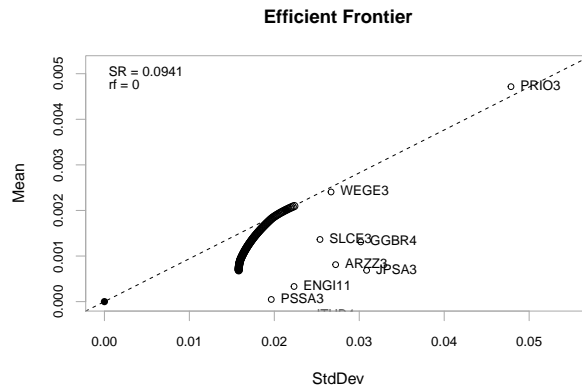
Avaliando as correlações dos retornos podemos ver que...



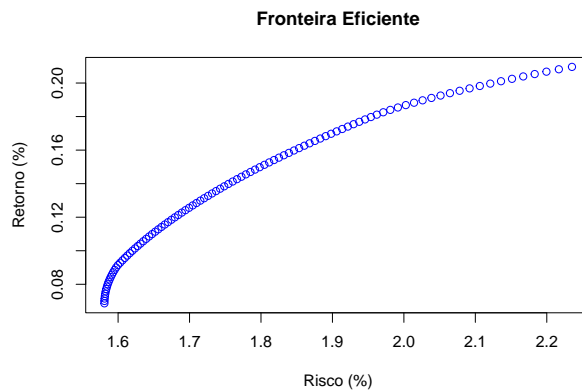
Montando a carteira

Antes de partir para a construção da carteira, dividimos nossos dados em `amostra` e `fora_amostra`, o primeiro para montarmos a carteira e o segundo para testarmos, que abrange apenas o mês de novembro de 2021.

Para montar e avaliar a carteira utilizamos dos pacotes `PortfolioAnalytics` e `PerformanceAnalytics`. Adicionamos 3 restrições a ela: totalmente investida, apenas posições compradas e pesos com mínimo de 0.01 e máximo de 0.2 do tipo `box`. Abaixo estão duas representações da fronteira eficiente considerando o argumento de `n.portfolios = 100`.



Também é possível analisar o gráfico de Risco e Retorno da seguinte maneira:



Processo de otimizacao

Para o processo de otimização, consideramos uma carteira de variância mínima, que foi montada da seguinte maneira:

```
## *****
## PortfolioAnalytics Optimization
## *****
##
## Call:
## optimize.portfolio(R = xts_amostra, portfolio = carteira_cvm,
##   optimize_method = "ROI", trace = TRUE)
##
## Optimal Weights:
## PRI03 ITUB4 JPSA3 ARZZ3 ENGI11 SLCE3 WEGE3 GGBR4 PSSA3
## 0.0100 0.1998 0.0100 0.0481 0.2000 0.2000 0.1221 0.0100 0.2000
##
## Objective Measure:
## StdDev
## 0.01581
```

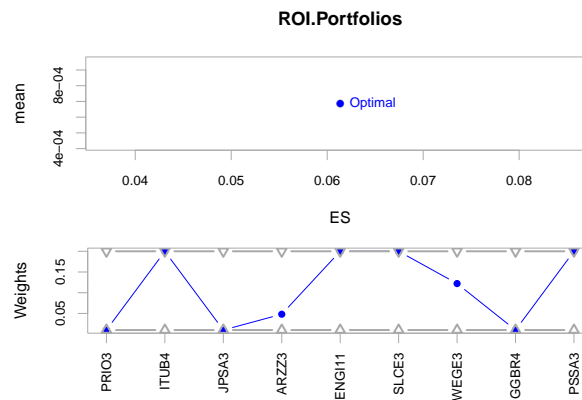
O retorno médio da carteira:

```
## [1] 0.08676089
```

Pesos em cada ativo:

```
## PRI03 ITUB4 JPSA3 ARZZ3 ENGI11 SLCE3 WEGE3 GGBR4 PSSA3
## 1.00 19.98 1.00 4.81 20.00 20.00 12.21 1.00 20.00
```

Alocações da minha carteira:

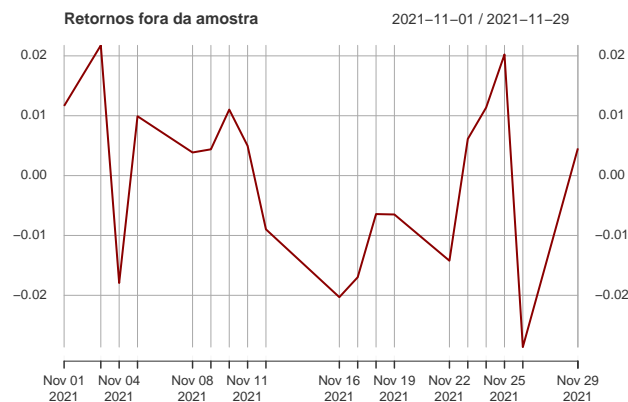


Teste fora da amostra

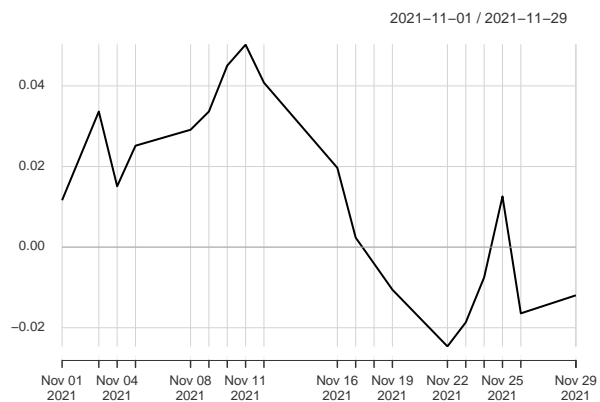
Agora que temos a carteira, iremos testar com o conjunto de dados `fora_amostra`, que definimos antes. Temos os seguintes resultados:

test_mean_return	test_sd_return	test_sharp
-0.0535808	1.431411	-0.0374322

Visualização



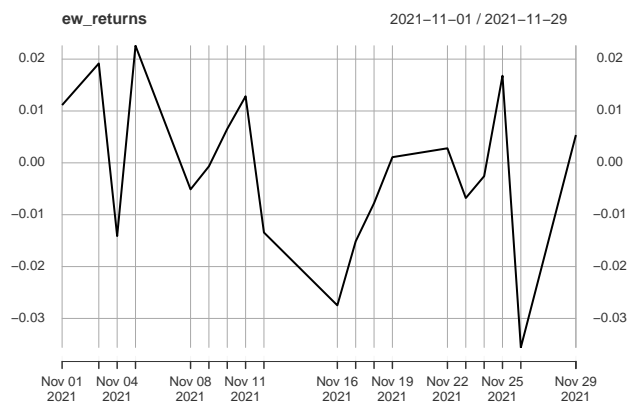
Retornos acumulados



Benchmark com pesos iguais

ew_mean_return	ew_sd_return	ew_sharp
-0.1597349	1.536623	-0.1039519

Visualização



Retornos acumulados

