

# Análise de Carteiras usando o R - Parte 4

## Bibliografia – BKM, cap. 7

Claudio Lucinda

FEA-RP/USP

# Risco Não Diversificável

- A seguir, temos uma figura calculada com os retornos das 37 ações que tem bastante negócios (negociadas em 2/3 dos dias nos 14 anos entre 2004 e 2018).
- Basicamente temos os desvios-padrão dos retornos para portfólios com 1, 2, ..., 37 ações, igualmente ponderados e rebalanceados mensalmente.
- Só temos nas duas pontas porque existem limites computacionais para isso.

# Risco Não Diversificável

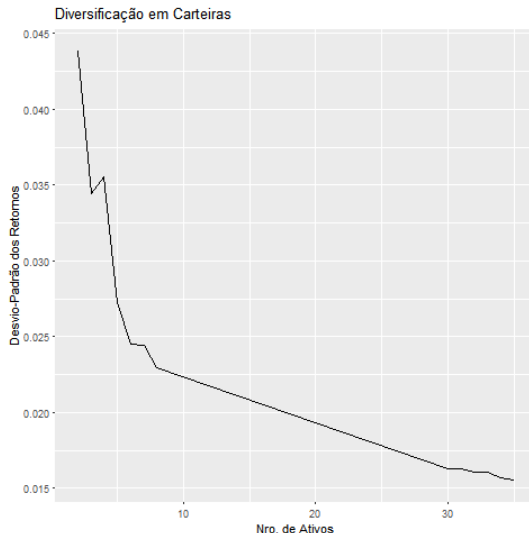


Figura 1: Carteira e Risco

# Fronteira Eficiente e o Pacote PortfolioAnalytics

# PortfolioAnalytics

- Nas aulas anteriores, nós utilizamos algumas funções do PortfolioAnalytics para alguns cálculos de carteira.
- Hoje iremos usar a funcionalidade mais legal do pacote, que é a capacidade de especificar e otimizar as carteiras de uma forma muito flexível.
- Inicialmente vamos fazer isso de uma forma mais geral para depois aplicarmos pro nosso caso.

# PortfolioAnalytics

- O processo de otimização de carteiras no PortfolioAnalytics tem quatro etapas:
  - Especificar a carteira - Quais são os ativos componentes
  - Adicionar restrições e objetivos
  - Otimizar
  - Analisar os resultados da otimização
- O código que vai ser usado aqui é o `eff_frontier.R`

# Otimizadores

- Existem quatro métodos de otimização suportados pelo PortfolioAnalytics
  - Solvers LP e QP:
    - ROI: Um pacote que centraliza solvers de LP e QP
  - Outros Métodos:
    - DEoptim: Differential Evolution Optimization
    - random: Otimização por Portfólios Aleatórios
    - GenSA: Generalized Simulated Annealing
    - pso: Particle Swarm Optimization

# Iniciando o PortfolioAnalytics

- O primeiro passo é especificar a lista de ativos que fazem parte da sua carteira:

```
port_spec<-portfolio.spec(colnames(IBOV>Returns_Final))
```

- O passo seguinte é implementar as restrições:

```
port_spec <- add.constraint(portfolio = port_spec,  
                           type = "full_investment")  
port_spec <- add.constraint(portfolio = port_spec,  
                           type = "long_only")
```



# Tipos de Restrições:

- Restrições sobre o peso:
  - `weight_sum` - Restrições sobre o peso total. Você precisa especificar as sub-opções `min_sum` e `max_sum`.
  - `full_investment` - Equivalente ao anterior, já fazendo `min_sum` e `max_sum` iguais a 1.
  - `dollar_neutral` ou `active` - igual ao anterior, só que `min_sum` e `max_sum` iguais a zero.
- `box` - Restrições sobre os limites mínimos e máximos sobre os pesos dos ativos:
  - Sub-opções: `min` e `max`
  - `long_only`: caso especial em que `min` é igual a zero e `max` é igual a um – ou seja, não há vendas a descoberto.
- `group` - Restrições que a soma dos pesos dos ativos em um grupo devem estar em uma faixa de valores (sub-opções: `group_min`, `group_max`)

## Tipos de Restrições (II):

- Restrições sobre limites de posição – `position_limit`
  - `max_pos`: Número máximo de ativos com pesos maiores que zero.
  - `max_pos_long`: Número máximo de ativos com pesos maiores que zero.
  - `max_pos_short`: Número máximo de ativos com pesos menores que zero (vendas a descoberto)
- Restrições sobre a diversificação – `diversification`
  - Diversificação entendida como a soma dos quadrados dos pesos.
- Restrição sobre o giro da carteira – `turnover`
  - Giro da carteira entendida como o volume dos negócios dividido pelo valor da carteira

## Tipos de Restrições (III):

- Restrições sobre o retorno médio – `return`
  - Sub-opção: `return_target`
- Exposição a fatores – `factor_exposure`
- Custos de transação – expressos como uma proporção do valor das negociação – `transaction_cost`, com sub-opção `ptc`.

# Objetivos

- O passo seguinte é especificar os objetivos do problema de otimização.
- Para isso, vamos usar a função `add.objective`.
- Os dois principais são objetivos de risco e retorno:
  - Os objetivos de risco são implementados como uma restrição no Lagrangiano
  - Os objetivos de retornos são implementados como a função objetivo do Lagrangiano

# Objetivos de Risco e Retorno

- O interessante deste objetivo de risco é que podemos usar todas aquelas medidas de risco para distribuição não normal que vimos anteriormente:
  - StdDev - O básico, o desvio-padrão dos retornos
  - ES - Expected Shortfall
  - VaR - *Value at Risk*
- O objetivo de retorno é uma função que vc especifica. Usualmente é o comando `mean`.

# Otimizando a carteira

- Você otimiza a carteira com o comando `optimize.portfolio`.
- Os exemplos a seguir mostram dois tipos de otimização:
  - Usando Portfólios Aleatórios
  - Usando um otimizador formal

```
opt <- optimize.portfolio(IBOV>Returns_Final,  
                          portfolio = port_spec,  
                          optimize_method = "random",  
                          search_size = 20000,  
                          trace = TRUE)  
  
opt2 <- optimize.portfolio(IBOV>Returns_Final,  
                           portfolio = port_spec,  
                           optimize_method = "ROI",  
                           trace = FALSE)
```

# Mostrando os tais dos portfólios aleatórios

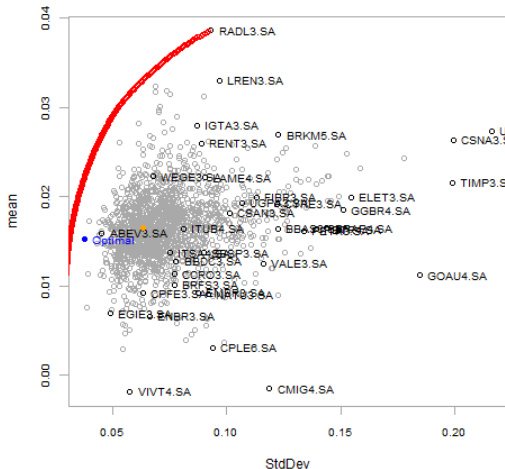


Figura 2: 20.000 Portfólios Aleatórios

## Combinando dois tipos de Fronteira Eficiente



## Combinando duas Fronteiras Eficientes

- Suponhamos que você queira combinar duas fronteiras eficientes – construídas sobre diferentes definições – em uma mesma figura para comparar como o conjunto de oportunidade de investimentos se altera.
- Podemos fazer isso facilmente com o comando `chart.EfficientFrontierOverlay`.
- Vou usar um exemplo combinando as fronteiras eficientes para dois tipos de estratégias de investimento:
  - Permitindo vendas a Descoberto (Long+Short)
  - Não permitindo vendas a Descoberto (Long Only)

## Iniciando a carteira long+short

```
port_spec2<-portfolio.spec(colnames(IBOV>Returns_Final))
port_spec2 <- add.constraint(portfolio = port_spec2,
                             type = "full_investment")
port_spec2 <- add.objective(portfolio = port_spec2,
                             type = "return",
                             name = "mean")
port_spec2 <- add.objective(portfolio = port_spec2,
                             type = "risk",
                             name = "StdDev")
```

## Fazendo o Gráfico – Código

```
portf.list<-combine.portfolios(list(port_spec, port_spec2))
legend.labels <- c("Long Only", "Long+Short")
chart.EfficientFrontierOverlay(R=IBOV>Returns_Final,
                               portfolio_list=portf.list,
                               type="mean-StdDev",
                               match.col="StdDev",
                               legend.loc="topleft",
                               legend.labels=legend.labels,
                               cex.legend=0.6,
                               labels.assets=FALSE,
                               pch.assets=18)
```

# O Gráfico

