Untitled25

July 15, 2023

1 Fronteira eficiente com múltiplos ativos

Iniciaremos trabalhando um exemplo hipotético. Suponha que um investidor pretenda compor sua carteir com seis ativos, denominados ativo 1, ativo 2, ativo 3, ativo 4, ativo 5 e ativo 6. As rentabilidades esperadas desses ativos são, respectivamente, $\overline{r}_1$, $\overline{r}_2$ e $\overline{r}_3$, $\overline{r}_4$, $\overline{r}_5$ e $\overline{r}_6$.

Suponha que nosso investidor invista as frações w_1, w_2, w_3, w_4, w_5, w_6, com $w_1 + w_2 + w_3 + w_3 + w_3 + w_4 = 1$, de sua riqueza nos ativos 1, 2, 3, 4, 5, e 6, respectivamente. A rentabilidade esperada desse investidor, $\overline{R}$ será dada pela média ponderada das rentabilidades esperadas dos ativos:

$$\overline{R} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} \overline{r}_1 \\ \overline{r}_2 \\ \overline{r}_3 \\ \overline{r}_4 \\ \overline{r}_5 \\ \overline{r}_6 \end{bmatrix} = \mathbf{w}^\top \mathbf{r}$$

Notando por $\sigma_{i,j}$ a covariância do ativo i em relação ao ativo j, a variância da rentabilidade dessa carteira de investimento, será

$$\sigma_R^2 = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \sigma_{1,3} & \sigma_{1,4} & \sigma_{1,5} & \sigma_{1,6} \\ \sigma_{2,1} & \sigma_{2,2} & \sigma_{2,3} & \sigma_{2,4} & \sigma_{2,5} & \sigma_{2,6} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma_{3,3} & \sigma_{3,4} & \sigma_{3,5} & \sigma_{3,6} \\ \sigma_{4,1} & \sigma_{4,2} & \sigma_{4,3} & \sigma_{4,4} & \sigma_{4,5} & \sigma_{4,6} \\ \sigma_{5,1} & \sigma_{5,2} & \sigma_{5,3} & \sigma_{5,4} & \sigma_{5,5} & \sigma_{5,6} \\ \sigma_{6,1} & \sigma_{6,2} & \sigma_{6,3} & \sigma_{6,4} & \sigma_{6,5} & \sigma_{6,6} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix} = \mathbf{w}^\top \Omega \mathbf{w}$$

Asuma que

$$\mathbf{r} = \begin{bmatrix} 0.03 \\ 0.10 \\ 0.20 \\ 0.08 \\ 0.08 \\ 0.03 \end{bmatrix}$$

e

$$\Omega = \begin{bmatrix} 1.8755 & 0.9880 & 0.8370 & -2.2961 & 1.2364 & 0.9995 \\ 0.9880 & 1.6900 & 2.8234 & -0.5599 & 1.4165 & 0.4817 \\ 0.8370 & 2.8234 & 6.2874 & 0.6477 & 2.4204 & 0.3091 \\ -2.2961 & -0.5599 & 0.6477 & 4.1814 & -1.0229 & -1.3037 \\ 1.2364 & 1.4165 & 2.4204 & -1.0229 & 1.6463 & 0.6313 \\ 0.9995 & 0.4817 & 0.3091 & -1.3037 & 0.6313 & 0.8978 \end{bmatrix}$$

Vamos simular quais serão os desvios padrões e a rentabilidade esperada da carteira de investimentos, assumindo diversos valores para os pesos w_i de cada um dos ativos nessa carteira.

[1]:
```
re <- c(0.03, 0.1, 0.2, 0.08, 0.08, 0.03)
```

[2]:
```
mcov <- matrix(c(
                  1.8755,   0.9880, 0.8370, -2.2961,  1.2364,  0.9995,
                  0.9880,   1.6900, 2.8234, -0.5599,  1.4165,  0.4817,
                  0.8370,   2.8234, 6.2874,  0.6477,  2.4204,  0.3091,
                  -2.2961, -0.5599, 0.6477,  4.1814, -1.0229, -1.3037,
                  1.2364,   1.4165, 2.4204, -1.0229,  1.6463,  0.6313,
                  0.9995,   0.4817, 0.3091, -1.3037,  0.6313,  0.8978
                 ),
              nrow=6, byrow=TRUE)
```

[3]:
```
n <- length(re) # número de ativos
```

Vamos simular várias combinações possíveis de w_1, w_2, w_3, w_4, w_5 e w_6. Primeiro, criarei uma variável para armazenar o número de simulações

[4]:
```
n_simul  <- 1e6 # n_simul é o número de simulações: um milhã
```

Agora, vamos criar um vetor com n×n_simul (n_simul para cada ativo) valores entre -1 e 3

[5]:
```
ws <- runif(n * n_simul, -1, 3) # valores antre -1 e 3 escolhidos␣
 ↪aleatoreamente a partir de uma distribuição uniforme
```

Depois, criamos uma matriz de n colunas com os dados desse vetor. Cada linha dessa matriz deverá representar uma possível combinação de pesos.

[6]:
```
pesos <- matrix(ws, nrow=n_simul, ncol=n)
```

Os pesos em cada linha não somam 1:

[7]:
```
head(rowSums(pesos))
```

1. 4.6616495763883 2. 5.53133753500879 3. 10.4802599130198 4. 7.03322399780154 5. 9.41391689330339 6. 8.80513434112072

Para corrigir isso, dividimos, para cada linha, cada um de seus componentes pela soma dos componentes

[8]:
```
spesos <- rowSums(pesos)
pesos <- pesos / spesos
```

Agora, os pesos estão corretos:

[9]:
```
head(rowSums(pesos))
```

1. 1 2. 1 3. 1 4. 1 5. 1 6. 1

Agora, vamos calcular as rentabilidades esperadas da carteira de ativos, considerando os pesos em cada linha da matriz pesos, ou seja, vamos calcular a rentabilidade esperada de cada carteira de ativos simulada.

[10]:
```
re_simul <- pesos %*% re # usamos o operador %*% para fazer uma multiplicação␣
 ↪matricial
```

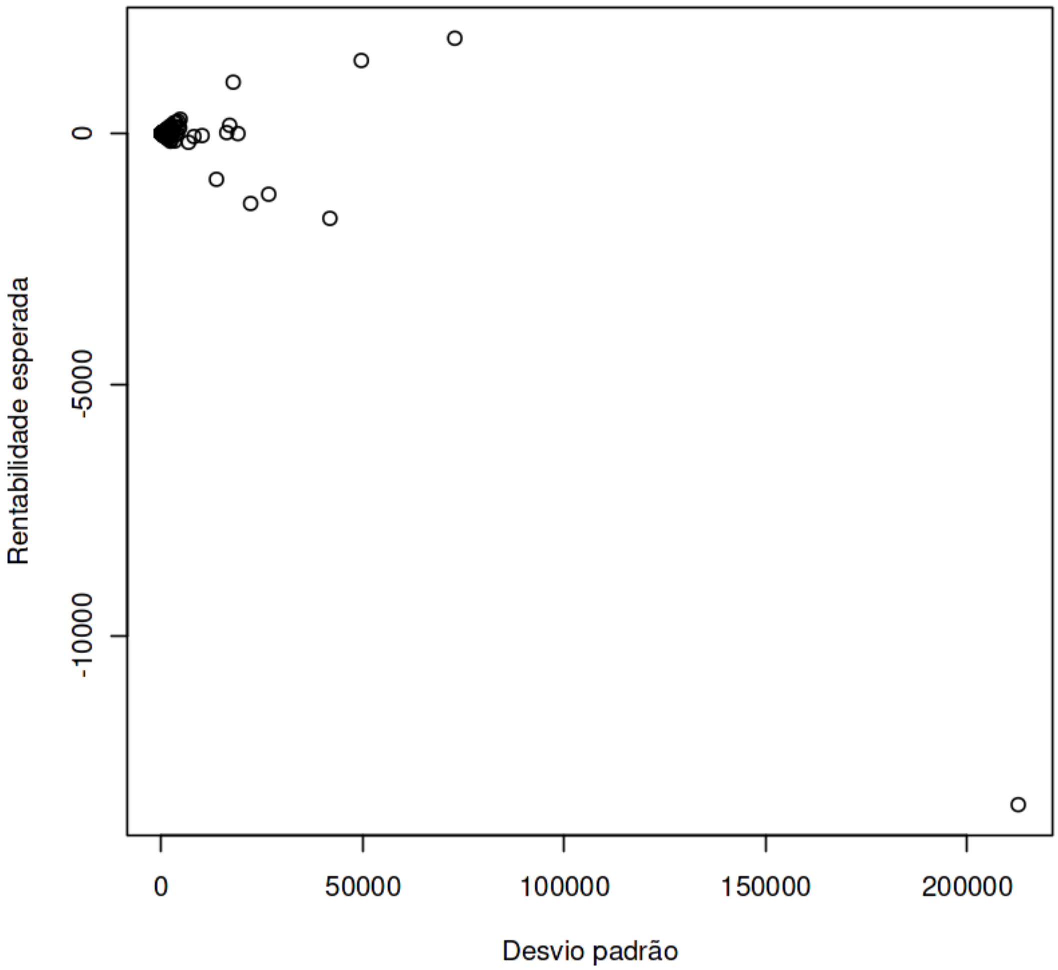
Calculamos também os devios padrões em cada linha. Para isso, primeiro definimos uma função que determina a variância da carteira dado um vetor de pesos $w = (w_1, w_2, w_3, w_4, w_5, w_6)$. Depois, aplicamos essa função a cada linha da matriz de pesos.

[11]:
```
clc_dp <- function(w){
    return(sqrt(w %*% mcov %*% w)[1])
}
# o R interpreta o vetor w à esquerda da expressão como um vetor linha e o
# vetor w à direita como um vetor coluna.
# O resultado da fórmula sqrt(w %*% mcov %*% w) é uma matriz de um único␣
 ↪elemento,
# ou seja, uma matriz 1 por 1.
# Como queremos que esse resultado seja tratado como um objeto sem dimensões,
# usamos o índice [1] que instrui o R a retornar o primeiro (e único)␣
 ↪componente
# da matriz.
```

[12]:
```
dp_simul <- apply(pesos, MARGIN=1, FUN=clc_dp)
```

Nossas simulações estão prontas. Vamos visualizar os resultados constuindo um gráfico de dispersão relacionando o desvio padrão e a rentabilidade esperada das diversas carteiras de ativos.

[13]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada")
```

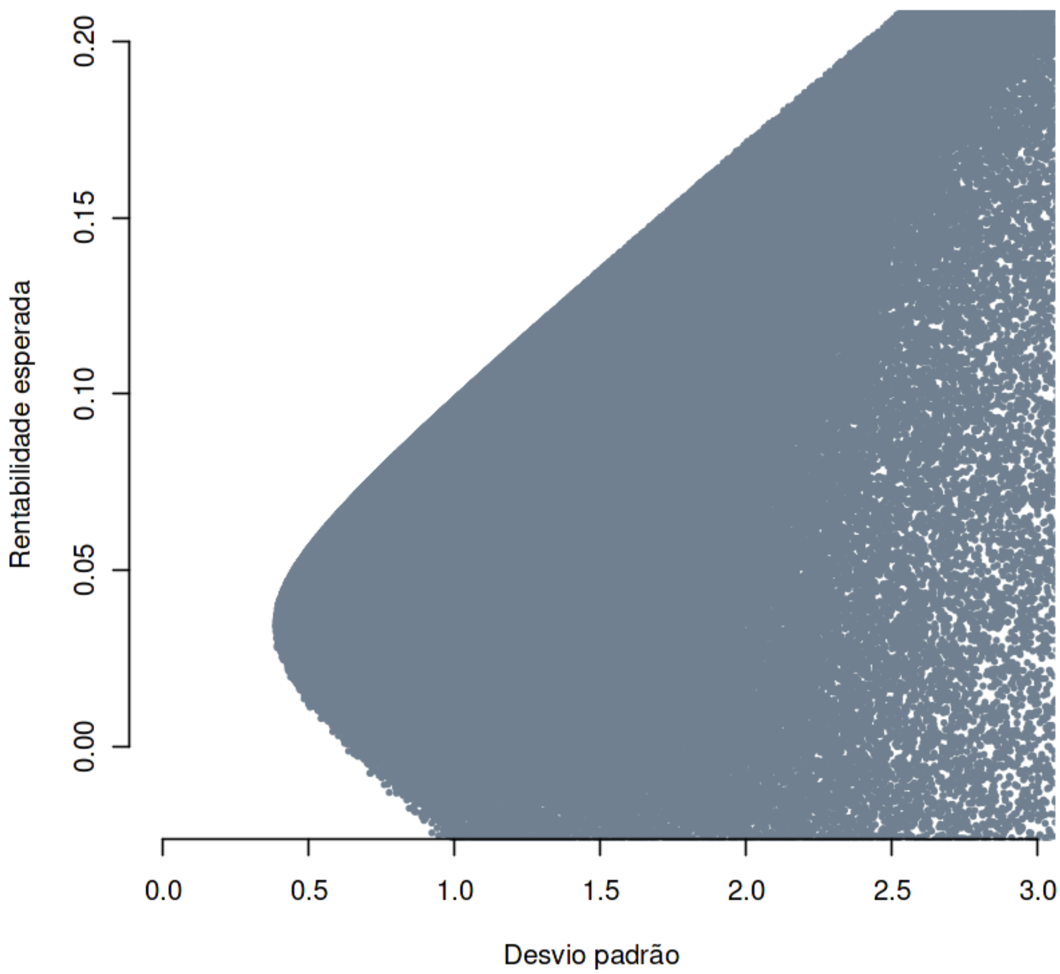


O R está reservando muito espaço no gráfico para poder exibir alguns poucos pontos que estão muito afastados da maioria dos outros. Vamos instruir ao R para construir o gráfico escolhendo um valor máximo no eixo horizontal correspondente ao percentil 98 dos desvios padrões obtidos nas simulações e um valor mínimo e um valor máximo para o eixo vertical correspondentes aos percentis 2% e 98% das rentabilidades esperadas obtidas nas simulações. Como isso, poderemo observa a área do gráfico na qual há maior densidade de pontos com mais detalhe.

[14]:
```
xmax <- quantile(dp_simul, 0.96)
ymin <- quantile(re_simul, 0.02)
ymax <- quantile(re_simul, 0.98)
```

[15]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3,
```

```
    col='slategray', bty='n')
```



2 Determinando a carteira de variância mínima

Vimos que a fórmula para calcular a carteira de variância mínima é

$$z_m = A_m^{-1} b_m.$$

Na qual

$$z_m = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ \lambda \end{bmatrix},$$

$ $ é um multiplicador de Lagrange,

$$b_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

e

$$A_m = \begin{bmatrix} 2\sigma_{11} & 2\sigma_{12} & 2\sigma_{13} & 2\sigma_{14} & 2\sigma_{15} & 2\sigma_{16} & 1 \\ 2\sigma_{21} & 2\sigma_{22} & 2\sigma_{23} & 2\sigma_{24} & 2\sigma_{25} & 2\sigma_{26} & 1 \\ 2\sigma_{31} & 2\sigma_{32} & 2\sigma_{33} & 2\sigma_{34} & 2\sigma_{35} & 2\sigma_{36} & 1 \\ 2\sigma_{41} & 2\sigma_{42} & 2\sigma_{43} & 2\sigma_{44} & 2\sigma_{45} & 2\sigma_{46} & 1 \\ 2\sigma_{51} & 2\sigma_{52} & 2\sigma_{53} & 2\sigma_{54} & 2\sigma_{55} & 2\sigma_{56} & 1 \\ 2\sigma_{61} & 2\sigma_{62} & 2\sigma_{63} & 2\sigma_{64} & 2\sigma_{65} & 2\sigma_{66} & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

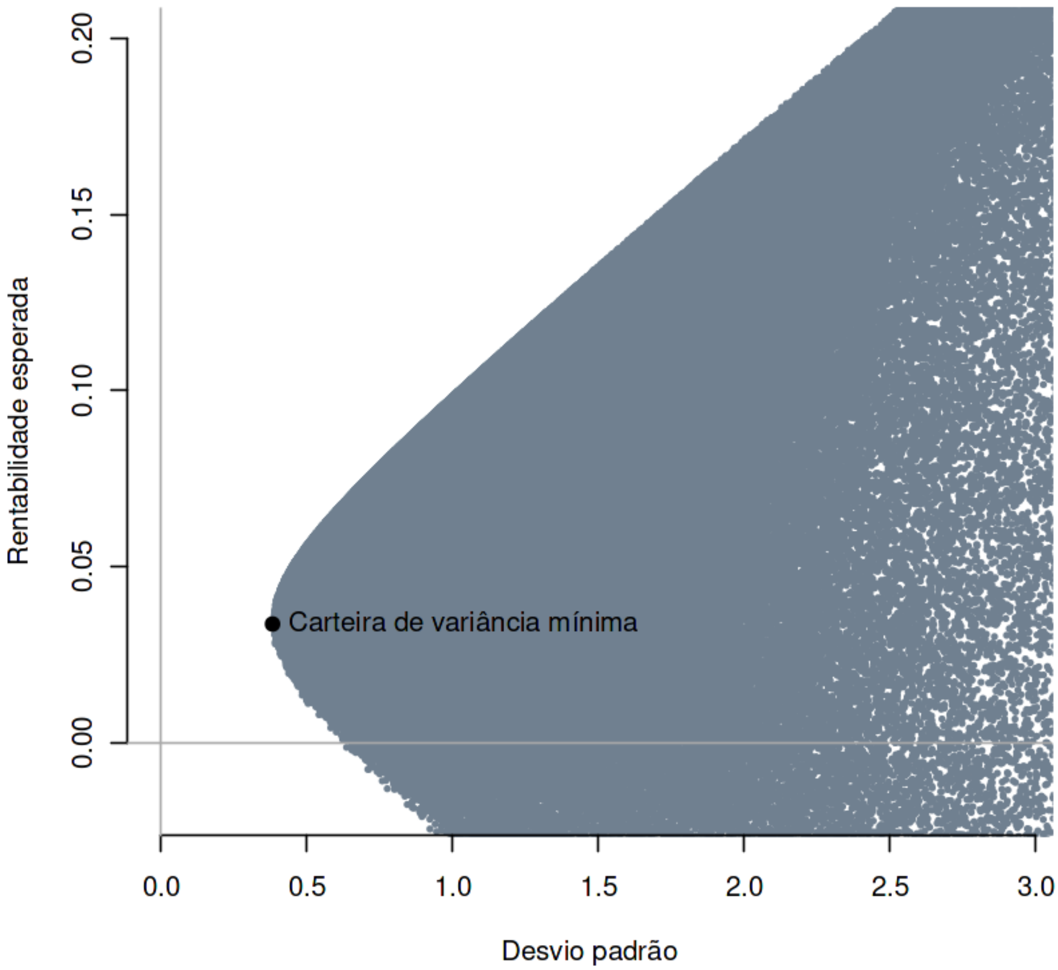
[16]:
```
Am <- rbind(cbind(2 * mcov, rep(1, n)), c(rep(1, n), 0))
bm <- c(rep(0, n), 1)
zm <- solve(Am, bm)
```

Os n primeiros componentes do vetor $\mathbf{z}_m$ constituem o vetor de pesos da carteira de variância mínima. Criamos o vetor `wvmin` com esses componentes e usamos esse vetor para calcular a rentabilidade esperada (`rvmin`) e o desvio padrão (`dpvmin`) da carteira de variância mínima:

[17]:
```
wvmin <- zm[1:n]
rvmin <- sum(wvmin * re)
dpvmin <- clc_dp(wvmin)
```

No gráfico abaixo, marcamos o ponto correspondente à carteira de variância mínima.

[18]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3,
     col="slategray", bty='n')
points(dpvmin, rvmin, pch=19)
text(dpvmin, rvmin, "Carteira de variância mínima", pos=4)
abline(h=0, v=0, col='darkgrey')
```



3 Determinando a fronteira de variância mínima

Para determinar a fronteira de variância mínima, podemos achar diversos pontos sobre essa fronteira usando a fórmula

$$z = A^{-1}b$$

na qual,

$$z = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ \lambda_1 \\ \lambda_2 \end{bmatrix},$$

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \overline{R} \\ 1 \end{bmatrix}$$

e

$$A = \begin{bmatrix} 2\sigma_{1,1} & 2\sigma_{1,2} & 2\sigma_{1,3} & 2\sigma_{1,4} & 2\sigma_{1,5} & 2\sigma_{1,6} & \overline{r}_1 & 1 \\ 2\sigma_{2,1} & 2\sigma_{2,2} & 2\sigma_{2,3} & 2\sigma_{2,4} & 2\sigma_{2,5} & 2\sigma_{2,6} & \overline{r}_2 & 1 \\ 2\sigma_{3,1} & 2\sigma_{3,2} & 2\sigma_{3,3} & 2\sigma_{3,4} & 2\sigma_{3,5} & 2\sigma_{3,6} & \overline{r}_3 & 1 \\ 2\sigma_{4,1} & 2\sigma_{5,2} & 2\sigma_{4,3} & 2\sigma_{4,4} & 2\sigma_{5,5} & 2\sigma_{4,6} & \overline{r}_4 & 1 \\ 2\sigma_{5,1} & 2\sigma_{6,2} & 2\sigma_{5,3} & 2\sigma_{5,4} & 2\sigma_{6,5} & 2\sigma_{5,6} & \overline{r}_5 & 1 \\ 2\sigma_{6,1} & 2\sigma_{7,2} & 2\sigma_{6,3} & 2\sigma_{6,4} & 2\sigma_{7,5} & 2\sigma_{6,6} & \overline{r}_6 & 1 \\ \overline{r}_1 & \overline{r}_3 & \overline{r}_3 & \overline{r}_4 & \overline{r}_5 & \overline{r}_6 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Calculamos, então os pesos ótimos para diversos valores de $\overline{R}$ e, com base nesses pesos, calculamos o desvio padrão

$$\sigma_R = \sum_{i=1}^{6}\sum_{j=1}^{6} w_i w_j \sigma ij$$

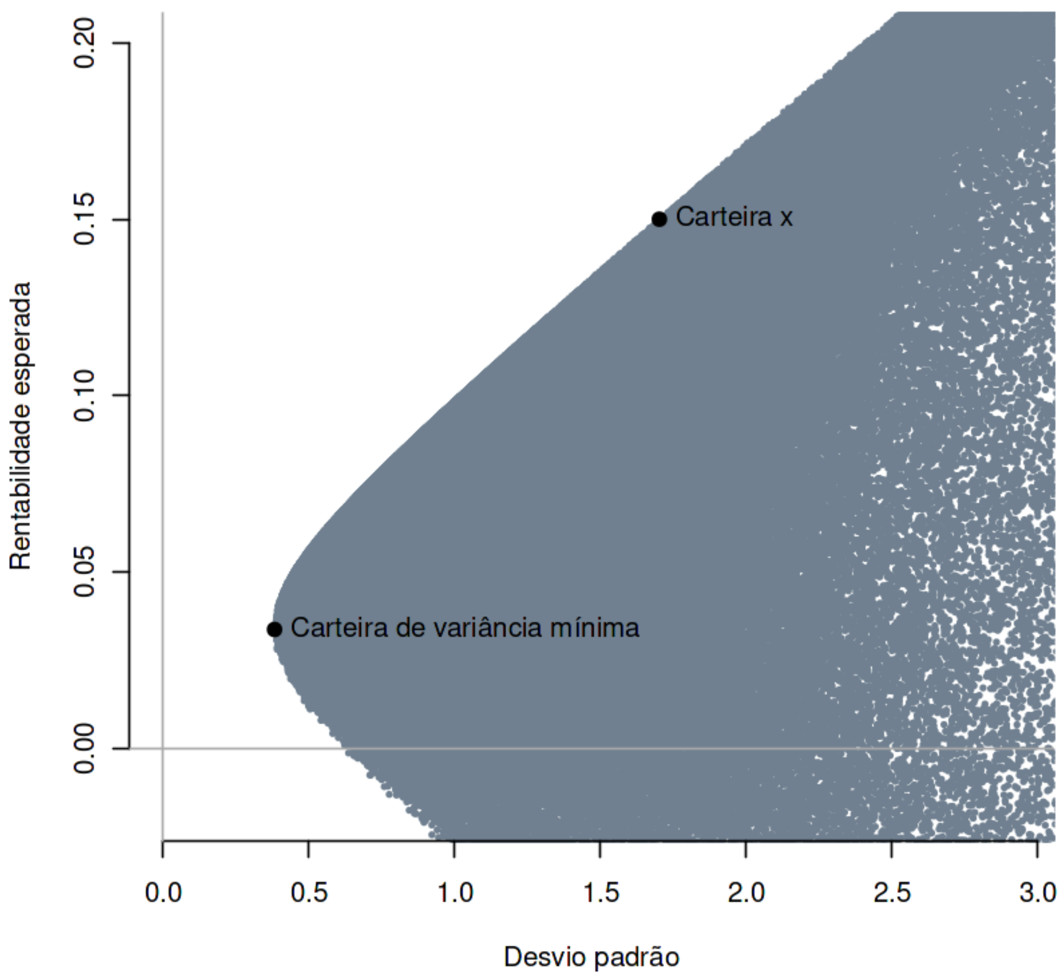
[19]:
```
A <- rbind(cbind(2 * mcov, re, rep(1, n)), c(re, 0, 0), c(rep(1, n), 0, 0))
```

Para ilustrar, calculemos a carteira de variância mínima associada à rentabilidade esperada de 15%.

[20]:
```
rrq  <- 0.15
b <- c(rep(0, n), rrq, 1)
z <- solve(A, b)
wx <- z[1:n]
rx  <- sum(wx*re)
dpx <- clc_dp(wx)
```

Vamos nos certificar que a carteira que encontramos realmente encontra-se sobre a FVM

[21]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3,
     col="slategray", bty='n')
points(c(dpvmin, dpx), c(rvmin, rx), pch=19)
text(c(dpvmin, dpx), c(rvmin, rx), c("Carteira de variância mínima", "Carteira␣
 ↪x"), pos=4)
abline(h=0, v=0, col='darkgrey')
```



Agora, vamos construir as carteiras de variância mínima condicionadas a diversas rentabilidades esperadas, calcular seus desvios padrões, e usá-las para desenhar a FVM. Primeiramente, construímos um vetor com 500 rentabilidades esperadas.

[22]:
```
rfvm <- seq(ymin, ymax, length.out=500)
```

Para calcular as carteiras de variância mínima associadas a cada uma dessas rentabilidades, definimos uma função (`cart_var_min`) que tem por argumento o valor da rentabilidade esperada e retorna a carteira de variância mínima condicionada a essa rentabilidade.

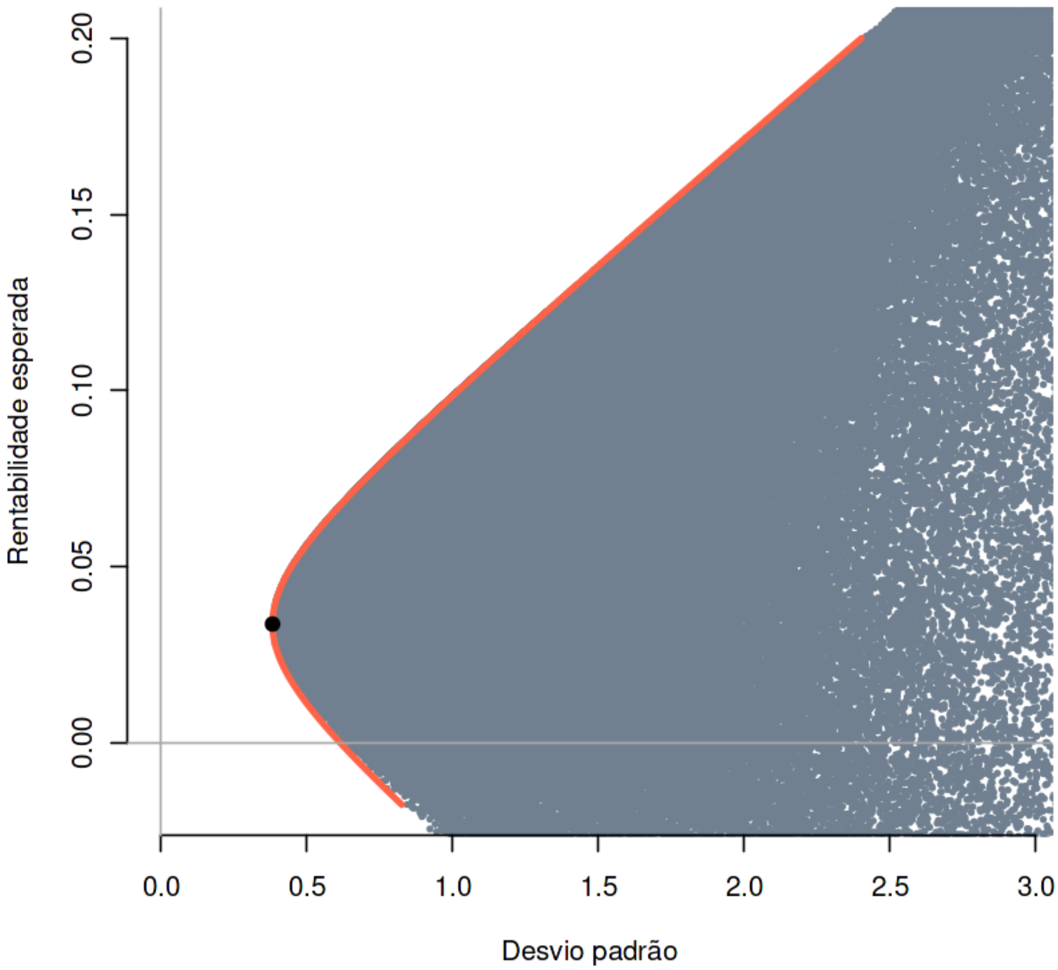
[23]:
```
cart_var_min <- function(r){
    b <- c(rep(0, n), r, 1)
    zb <- solve(A, b)
    return(zb[1:n])
}
```

Agora aplicamos essa função a cada componente do vetor `rfvm` para obtermos a matriz, que denominaremos `wfvm`, na qual cada coluna representa a carteira de variância mínima associada à respectiva rentabilidade no vetor `rfvm`.

[24]:
```
wfvm  <- sapply(rfvm, cart_var_min)
dsvpdsb <- apply(wfvm, 2, clc_dp)
```

Podemos agora usar a função lines do R (que plota os pontos unindo-os com segmentos de reta) para desenhar a fronteira de variância mínima.

[25]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3, col='slategray',␣
 ↪bty='n')
lines(dsvpdsb, rfvm, col='tomato', lw=3)
points(dpvmin, rvmin, pch=19)
abline(h=0, v=0, col='darkgrey')
```



Se $\mathbf{w}_m$ e $\mathbf{w}_a$ são duas carteiras sobre a fronteira de variância mínima, então, para qualquer $\alpha \in \mathbb{R}$, a carteira

$$\mathbf{w}_\alpha = \alpha \mathbf{w}_m + (1-\alpha)\mathbf{w}_a$$

também está sobre a fronteira de variância mínima. Vamos usar esse resultado para fazer uma outra estimativa da fronteira de variância mínima. Primeiramente, escolhemos duas carteiras quaisquer na matriz `wfvm` de carteiras de variância mínima condicionada que já calculamos:

[26]:
```
idxm <- 0.4 * length(rfvm)
idxa <- 0.05 * length(rfvm)
```

Agora criamos as variáveis `rm`, `dpm`, `wm`, `ra`, `dpa`, `wa` e `covam` que armazenarão os valores das rentabilidades esperadas, dos desvios padrões, os pesos dos ativos e as covariâncias das duas carteiras

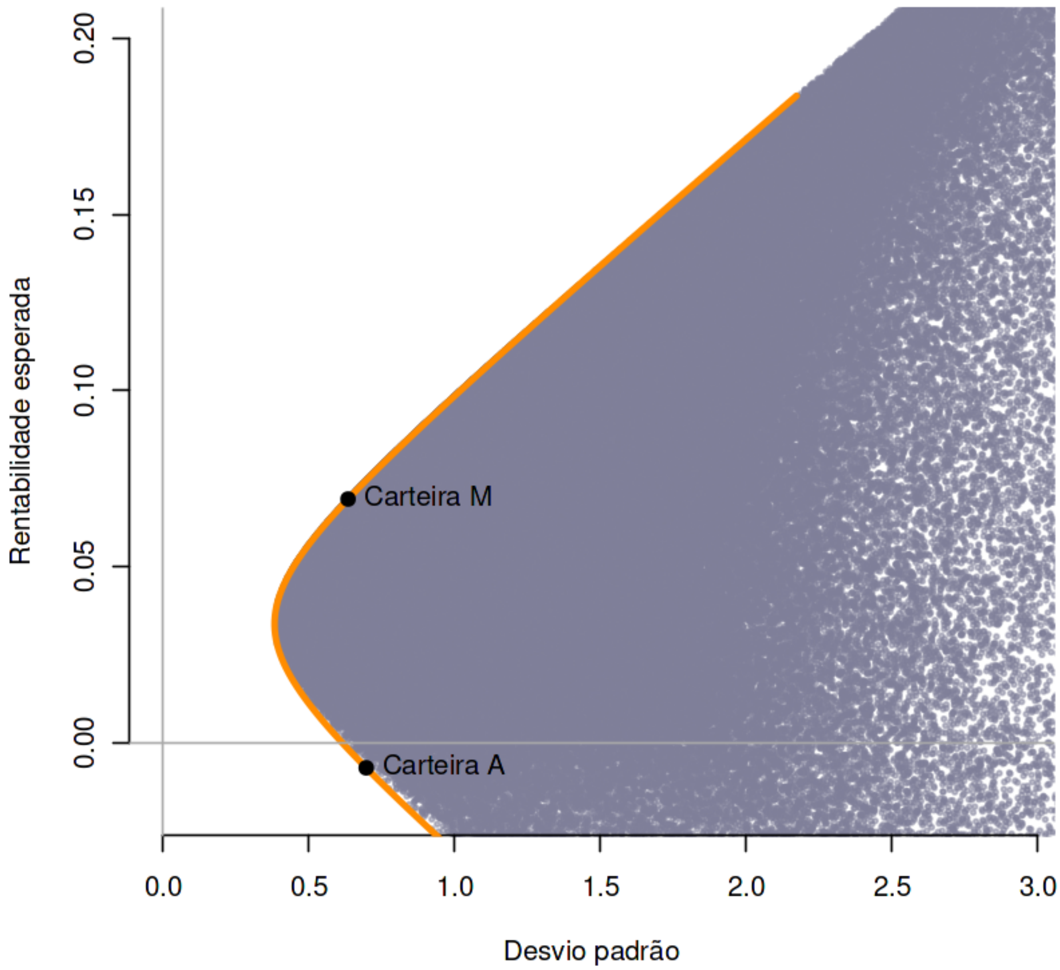
[27]:
```
rm <- rfvm[idxm]
dpm <- dsvpdsb[idxm]
wm <- wfvm[,idxm]
ra <- rfvm[idxa]
dpa <- dsvpdsb[idxa]
wa <- wfvm[,idxa]
covam <- (wa %*% mcov %*% wm)[1]
```

Como iremos descrever graficamente a relação entre desvio padrão e rentabilidade esperada resultante de outras combinações de carteiras de ativos, criamos uma função para que isso possa ser feito automaticamente

[28]:
```
combinaa <- function(wa, wb, amax, amin, cor=NULL, esp=NULL, rotulos=NULL){
  sa <- clc_dp(wa) # calcula o desvio padrão da carteira com pesos wa
  sb <- clc_dp(wb) # idem pesos wb
  ra  <- sum(wa * re) # calcula a rentabilidade esperada da carteira wa
  rb  <- sum(wb * re) # idem wb
  alfas <- seq(amin, amax, length.out=200) # vetor de pesos para combinar as␣
 ↪duas carteiras
  ws <- sapply(alfas, function(a) a * wa + (1-a) * wb) # carteiras resultantes
  re <- apply(ws, 2, function(x) sum(re * x)) # vetor de rentabiĺidades␣
 ↪esperadas
  dp <- apply(ws, 2, clc_dp) # vetor de desvios padrões
  lines(dp, re, col=cor, lw=esp) # instrução para plotar as linhas no gráfico
  points(c(sa, sb), c(ra, rb), pch=19) # marcar os pontos referentes às␣
 ↪carteiras wa e wb
  text(c(sa, sb), c(ra, rb), rotulos, pos=4)
}
```

Agora usamos essa função para desenhar a FVM como um conjunto de combinações afim entre as carteiras m e a

[29]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3, col=rgb(.5, .5, 0.6,␣
 ↪alpha=0.5), bty='n')
combinaa(wm, wa, 2.5, -1, cor='darkorange', esp=3, rotulos=c('Carteira M',␣
 ↪'Carteira A'))
abline(h=0, v=0, col='darkgrey')
```



Também podemos construir a fronteira de variância mínima usando a fórmula

$$\sigma_R^2 = \frac{\sigma_{11} + \sigma_{22} - 2\sigma_{12}}{\left(\overline{r}_1 - \overline{r}_2\right)^2}\overline{R}^2 - 2\frac{\overline{r}_2\sigma_{11} + \overline{r}_1\sigma_{22} - \sigma_{12}\left(\overline{r}_1 + \overline{r}_2\right)}{\left(\overline{r}_1 - \overline{r}_2\right)^2}\overline{R} + \frac{\overline{r}_2^2\sigma_{11} + \overline{r}_1^2\sigma_{22} - 2\overline{r}_1\overline{r}_2\sigma_{12}}{\left(\overline{r}_1 - \overline{r}_2\right)^2}.$$

Na qual os subscritos 1 e 2 referem-se a duas carteiras já conhecidas sobre essa fronteira. Tomemos duas carteiras que já calculamos: a carteira de variância mínima, que pode ser a nossa carteira 1, e a carteira x, que pode ser a nossa carteira 2. Notando por $\mathbf{w}_1$ o vetor de pesos da carteira 1 e por $\mathbf{w}_2$ o vetor de pesos da carteira 2, a covariância entre as rentabilidades dessas duas carteiras é dada por

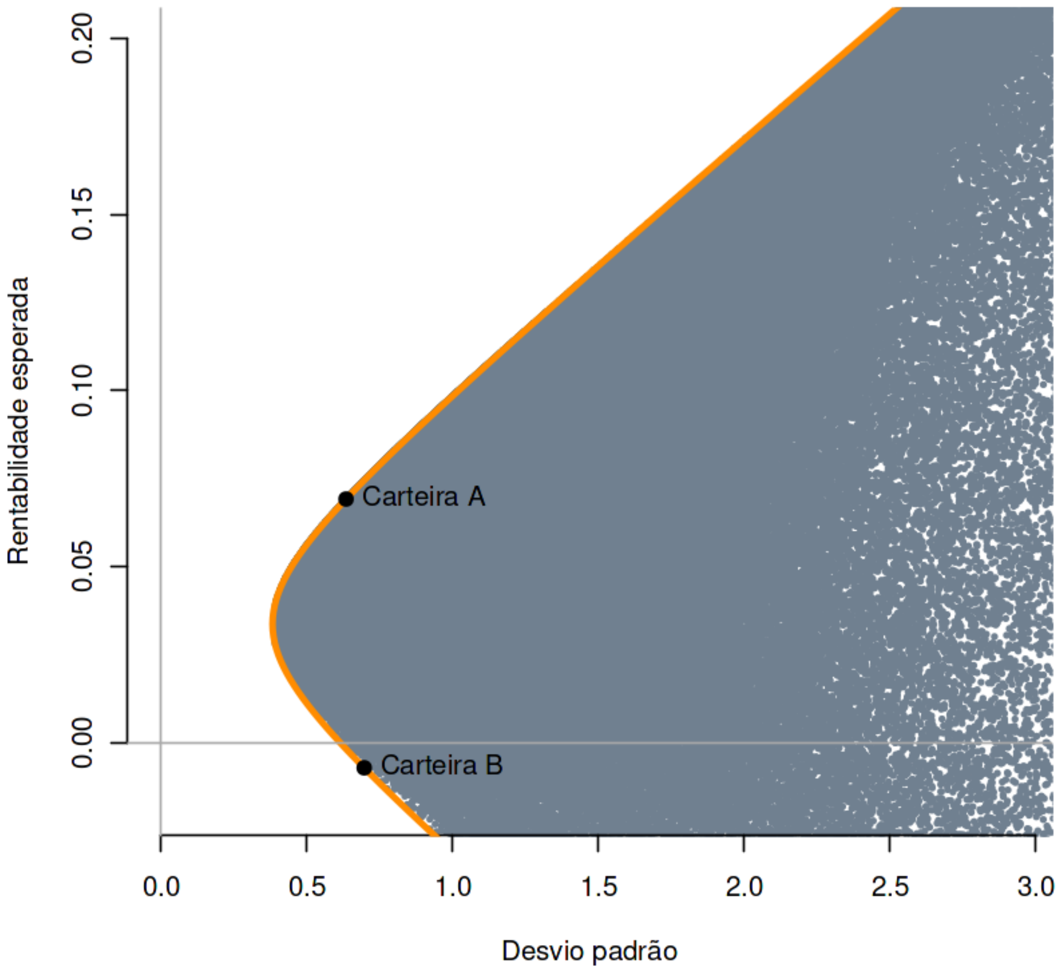
$$\sigma_{12} = \mathbf{w}_1'\Omega\mathbf{w}_2$$

[30]:
```
combinab <- function(re, ss, sab, rmin, rmax,
                     rotulos=NULL,
```

```
                    cor=NULL, esp=NULL, n=200
                       ){
  a <- (ss[1]^2 + ss[2]^2 - 2 * sab)
  b <- - 2 * (re[2] * ss[1]^2 + re[1] * ss[2]^2 - sab * (re[1] + re[2]))
  c_ <- (re[2]^2 * ss[1]^2 + re[1]^2 * ss[2]^2 - 2 * re[1] * re[2] * sab)
  rsd <- seq(rmin, rmax, length.out=n)
  dsvpdsd <- sqrt((a * rsd^2 + b * rsd + c_)/(re[1]-re[2])^2)
  lines(dsvpdsd, rsd, col=cor, lw=esp)
  points(ss, re, pch=20, cex=1.5)
  text(ss, re, rotulos, pos=4)
}
```

[31]:

```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3,
     col='slategray', bty='n')
combinab(
        c(rm, ra), c(dpm, dpa), covam, -0.1, 0.25, cor='darkorange', esp=3,
        rotulos=c('Carteira A', 'Carteira B')
)
abline(h=0, v=0, col='darkgrey')
```



4 Encontrando a carteira zero-beta

Conforme vimos, se as carteiras m e a pertencem à fronteira de variância mínima, então qualquer combinação afim das duas também pertencerá a essa fronteira. Em particular a carteira

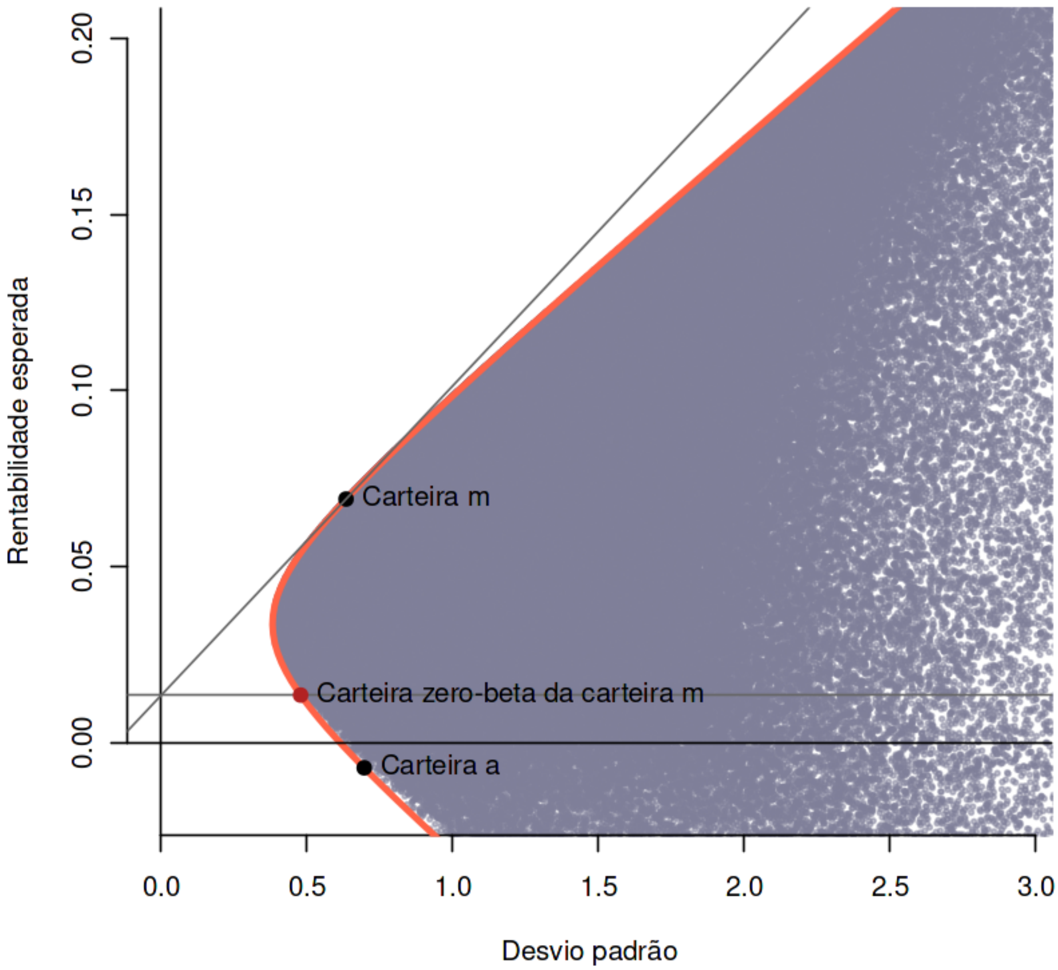
$$\mathbf{w}_{om} = \frac{\beta_{am}}{\beta_{am} - 1}\mathbf{w}_m - \frac{1}{\beta_{am} - 1}\mathbf{w}_a$$

na qual $\beta_{am} = \sigma_{am}/\sigma_{mm}$, pertence à FVM e tem covariância nula com a carteira m. Essa carteira é chamada de carteira zero-beta da carteira m. A seguir escolhemos duas carteiras sobre a fronteira de variância mínima, calculamos, a partir dessas duas carteiras a carteira zero-beta de uma delas e marcamos a posição dessa carteira na FVM.

Uma propriedade notável da carteira zero-beta é que a inclinação da FVM no ponto correspondente à carteira a tem inclinação $\frac{\bar{R}_m - \bar{R}om}{\sigma_m}$.

[32]:
```
beta <- covam / dpm^2
alfa <- beta / (beta-1)
wo <- alfa * wm + (1 - alfa) * wa
ro <- sum(wo * re)
dpo <- clc_dp(wo)
```

[33]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3, col=rgb(.5, .5, 0.6,
 ↪alpha=0.5), bty='n')
combinaa(wm, wa, 5, -2, cor='tomato', esp=3, rotulos=c("Carteira m", "Carteira
 ↪a"))
abline(h=0, v=0)
abline(a=ro, b=(rm-ro)/dpm, h=ro, col=rgb(0.4, 0.4, 0.4))
points(dpo, ro, pch=19, col="firebrick")
text(dpo, ro, "Carteira zero-beta da carteira m", pos=4)
```

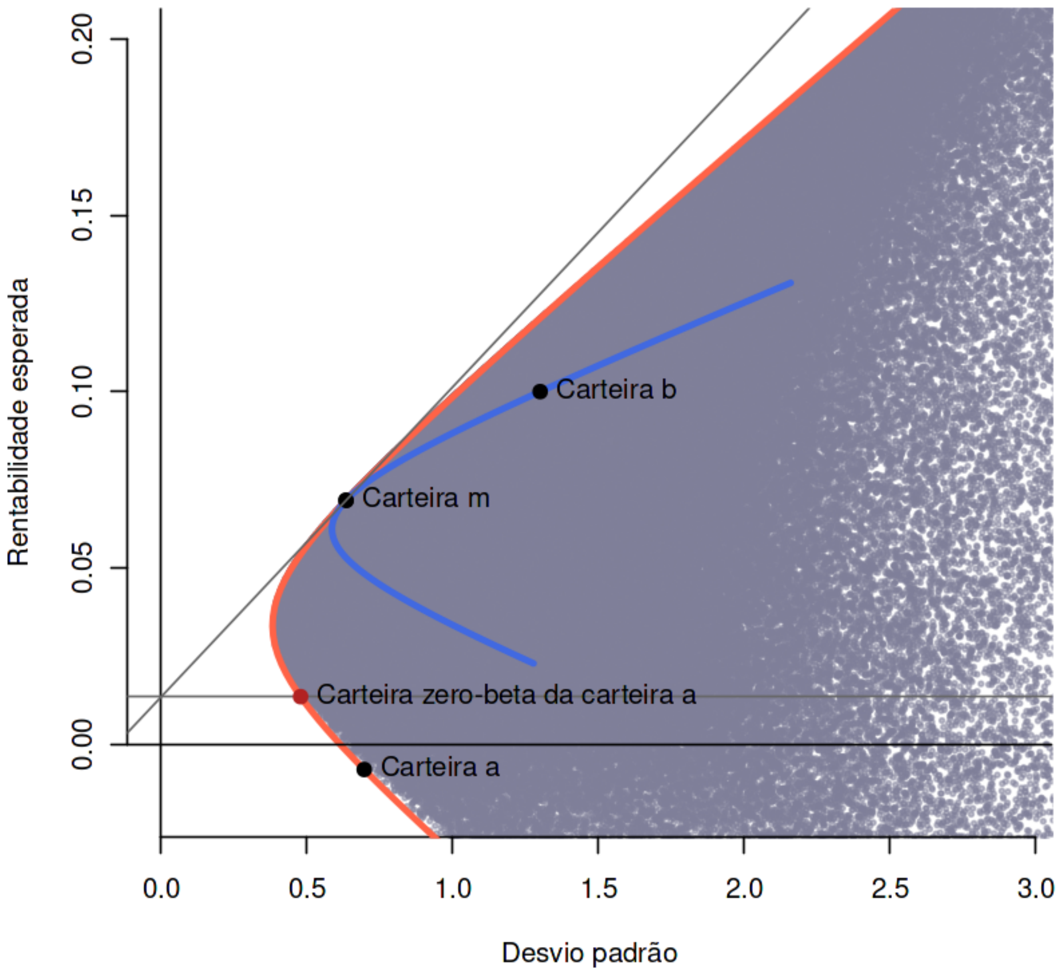


Agora, vamos também desenhar a curva descrevendo a rentabilidade esperada e o desvio padrão de carteiras resultantes de combinações afim da carteira a com uma carteira ou ativo fora da fronteira eficiente, para mostrar que essa curva tangencia à direita a FVM no ponto correspondente à carteira a. Para ilustrar, escolhemos fazer combinações afim entre o ativo 2 e a carteira a.

[34]:
```
wb <- replace(rep(0, n), 2, 1) # Uma carteira contendo apenas o ativo 2
```

[35]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3, col=rgb(.5, .5, 0.6,␣
 ↪alpha=0.5), bty='n')
combinaa(wm, wa, 5, -2, cor='tomato', esp=3, rotulos=c("Carteira m", "Carteira␣
 ↪a"))
combinaa(wm, wb, 2.5, -1, cor='RoyalBlue', esp=3, rotulos=c("", "Carteira b"))
```

```
abline(h=0, v=0)
abline(a=ro, b=(rm-ro)/dpm, h=ro, col=rgb(0.4, 0.4, 0.4))
points(dpo, ro, pch=19, col="firebrick")
text(dpo, ro, "Carteira zero-beta da carteira a", pos=4)
```



5 Carteira ótima

No vídeo teórico vimos que a carteira ótima pode ser derivada a partir da fórmula

$$\mathbf{w} = \frac{\Omega^{-1}(\bar{\mathbf{r}} - R_f \cdot \mathbf{1})}{\mathbf{1}'\Omega^{-1}(\bar{\mathbf{r}} - R_f \cdot \mathbf{1})}$$

na qual Ω é a matriz de covariância dos ativos de risco e R_f é a rentabilidade do ativo livre de risco.

Vamos assumir em nosso exemplo que, além dos três ativos, haja um ativo livre de risco com rentabilidade igual a 1%.

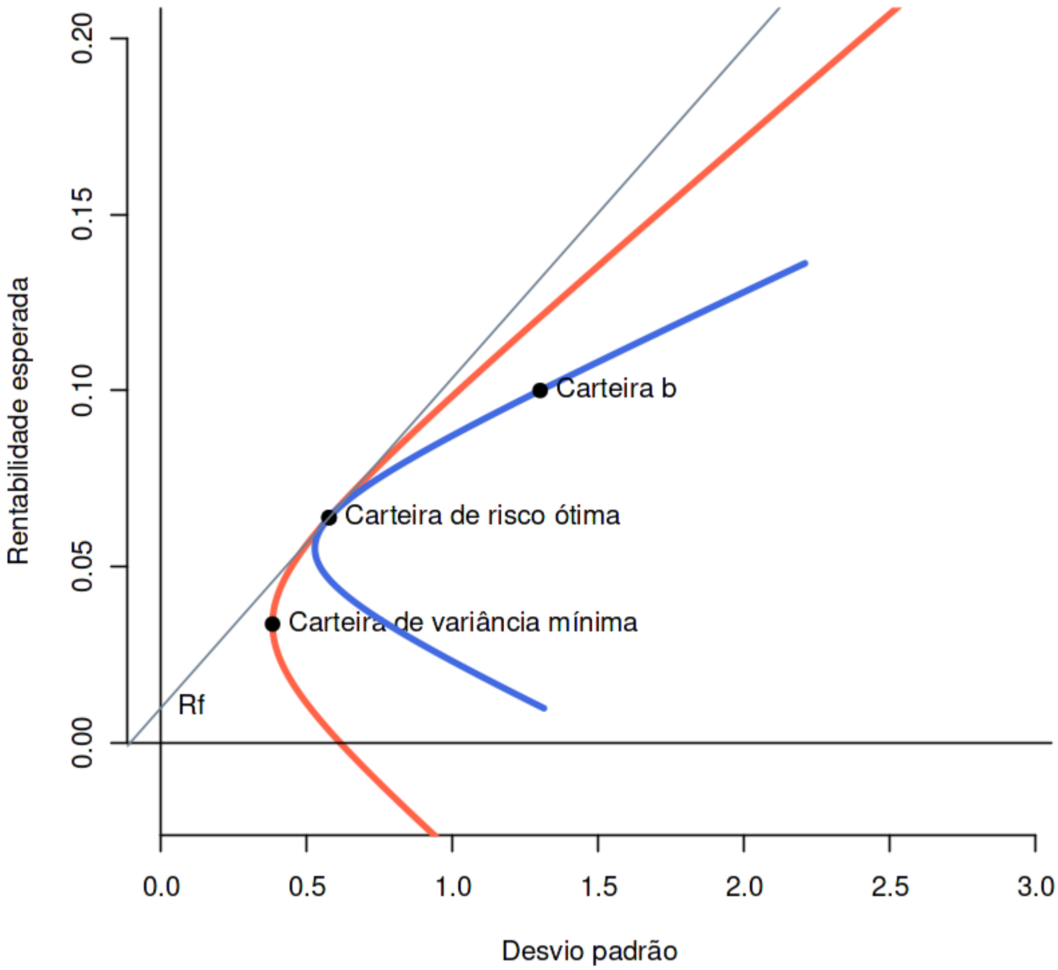
[36]:
```
rf = 0.01
```

[37]:
```
mcov_inv = solve(mcov)
wotm = c(mcov_inv %*% (re - rf) / sum(mcov_inv %*% (re - rf)))
```

[38]:
```
rotm = sum(wotm * re)
dpotm = clc_dp(wotm)
shrp_otm = (rotm - rf) / dpotm
rslt  <- c(rotm, dpotm, shrp_otm)
names(rslt) <- c('rentabiliade', 'desvio padrão', 'índice de Sharpe')
```

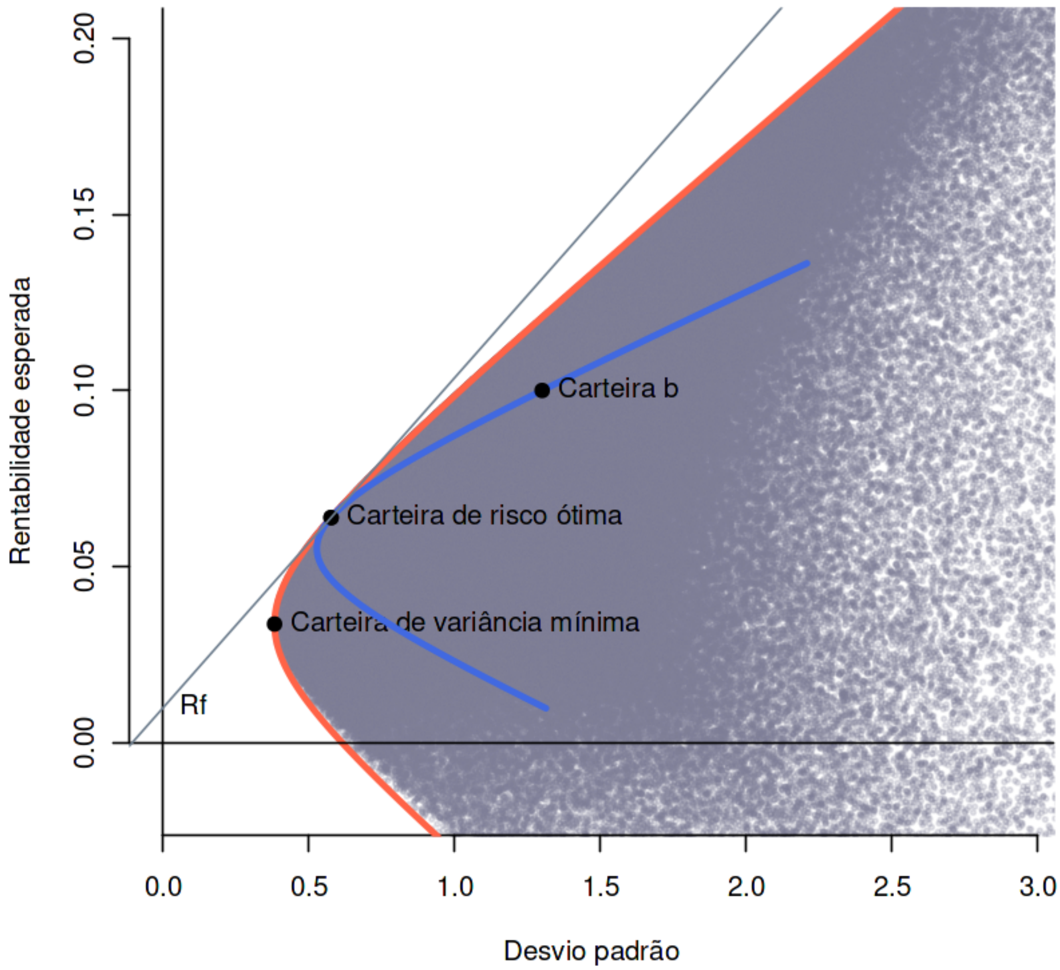
Podemos agora representar o gráfico da verdadeira fronteira eficiente (considerando a carteira final, com ativos com risco e o ativo livre de risco).

[39]:
```
plot(NULL, NULL, xlab="Desvio padrão", ylab='Rentabilidade esperada',
    xlim=c(0, xmax), ylim=c(ymin, ymax), bty='n')
combinaa(wotm, wvmin, 6, -2, cor='Tomato', esp=3, rotulos=c("Carteira de risco␣
 ↪ótima", 'Carteira de variância mínima'))
combinaa(wotm, wb, 2.5, -1, cor='RoyalBlue', esp=3, rotulos=c("", "Carteira b"))
abline(h=0, v=0)
abline(rf, shrp_otm, col='slategray', lwd=1)
text( 0, rf, 'Rf', pos=4)
```



Abaixo, um gráfico contendo, além das curvas acima, os pontos representando as combinações de desvio padrão e rentabilidade esperada de carteiras de ativos aleatório

[40]:
```
plot(dp_simul, re_simul, xlab="Desvio padrão", ylab="Rentabilidade esperada",
     xlim=c(0, xmax), ylim=c(ymin, ymax), pch=19, cex=0.3, col=rgb(.5, .5, 0.6,␣
 ↪alpha=0.2), bty='n')
combinaa(wotm, wvmin, 6, -2, cor='Tomato', esp=3, rotulos=c("Carteira de risco␣
 ↪ótima", 'Carteira de variância mínima'))
combinaa(wotm, wb, 2.5, -1, cor='RoyalBlue', esp=3, rotulos=c("", "Carteira b"))
abline(h=0, v=0)
abline(rf, shrp_otm, col='slategray', lwd=1)
text( 0, rf, 'Rf', pos=4)
```


Rentabilidade esperada
Desvio padrão
0.00
0.05
0.10
0.15
0.20
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Carteira b
Carteira de risco ótima
Carteira de variância mínima
Rf