Portfolio Theory with Short Sales Restrictions

Eric Zivot

4/12/2021

Short Sale

- Borrow asset from broker and sell now
- To close short position, buy back asset and return to broker
- Profit if asset price drops after short sale
- If asset i is sold short then

$$x_i < 0$$

where x_i = share of wealth in asset i

No Short Sale Restrictions

- Exchanges (e.g. NYSE, NASDAQ) may prevent short sales in some assets
- Some institutions (e.g. pension funds) are prevented from short-selling assets
- Certain accounts (e.g. retirement accounts) do not allow short sales
- Certain assets (e.g. mutual funds) cannot be shorted
- Short selling often requires substantial credit qualifications

Markowitz Algorithm with No Short Sales Restrictions

$$\begin{aligned}\min_{\mathbf{x}} \sigma^2_{p,x} &= \mathbf{x}'\Sigma\mathbf{x} \quad \text{s.t.} \\ \mu_{p,x} &= \mathbf{x}'\mu = \mu^0_p \\ \mathbf{x}'\mathbf{1} &= 1 \\ x_i &\geq 0 \ (i = 1, \ldots, n)\end{aligned}$$

Remark: With inequality constraints, the Lagrange multiplier method no longer works because it imposes an equality in the constraint.

Markowitz Algorithm with No Short Sales Restrictions

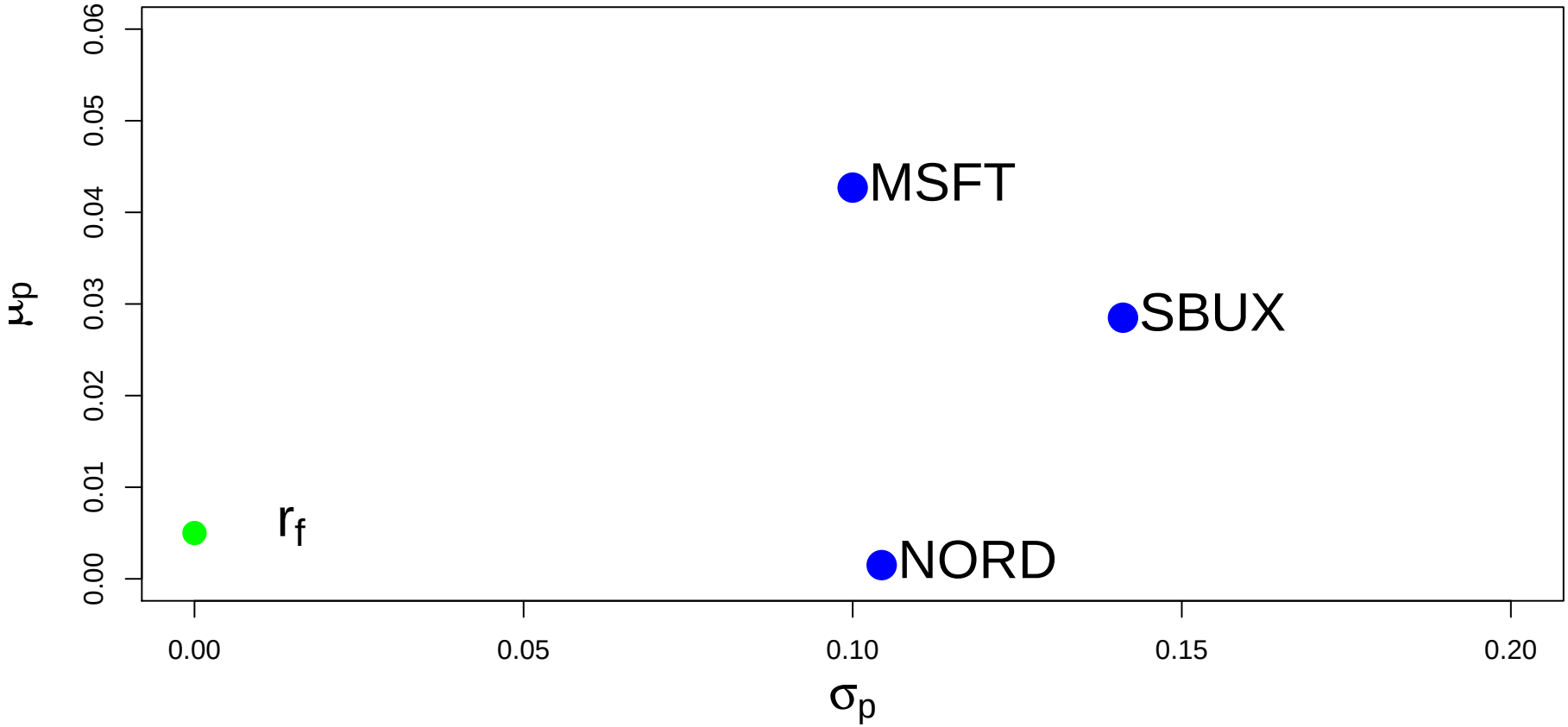
- Problem must be solved numerically, e.g. using the function `solve.QP()` in the R package **quadprog** or the `Solver` in Excel.
- Portfolio frontier can no longer be constructed from any two efficient portfolios (cannot guarantee positive weights). It has to be computed by brute force for each portfolio with target expected return above the global minimum variance expected return.
- There may not be a feasible solution; i.e., there may not exist a no-short sale portfolio that reaches the target return μ_p^0.
- No short sale portfolio frontier must lie "inside" the portfolio frontier that allows short sales

Familiar Three Asset Example

Estimates of GWN model for Microsoft, Nordstrom and Starbucks stock from monthly returns over the period January 1995 to January 2000.

```
asset.names <- c("MSFT", "NORD", "SBUX")
mu.vec = c(0.0427, 0.0015, 0.0285)
names(mu.vec) = asset.names
sigma.mat = matrix(c(0.0100, 0.0018, 0.0011,
           0.0018, 0.0109, 0.0026,
           0.0011, 0.0026, 0.0199),
         nrow=3, ncol=3)
dimnames(sigma.mat) = list(asset.names, asset.names)
r.f = 0.005
sd.vec = sqrt(diag(sigma.mat))
```

Risk return characteristics



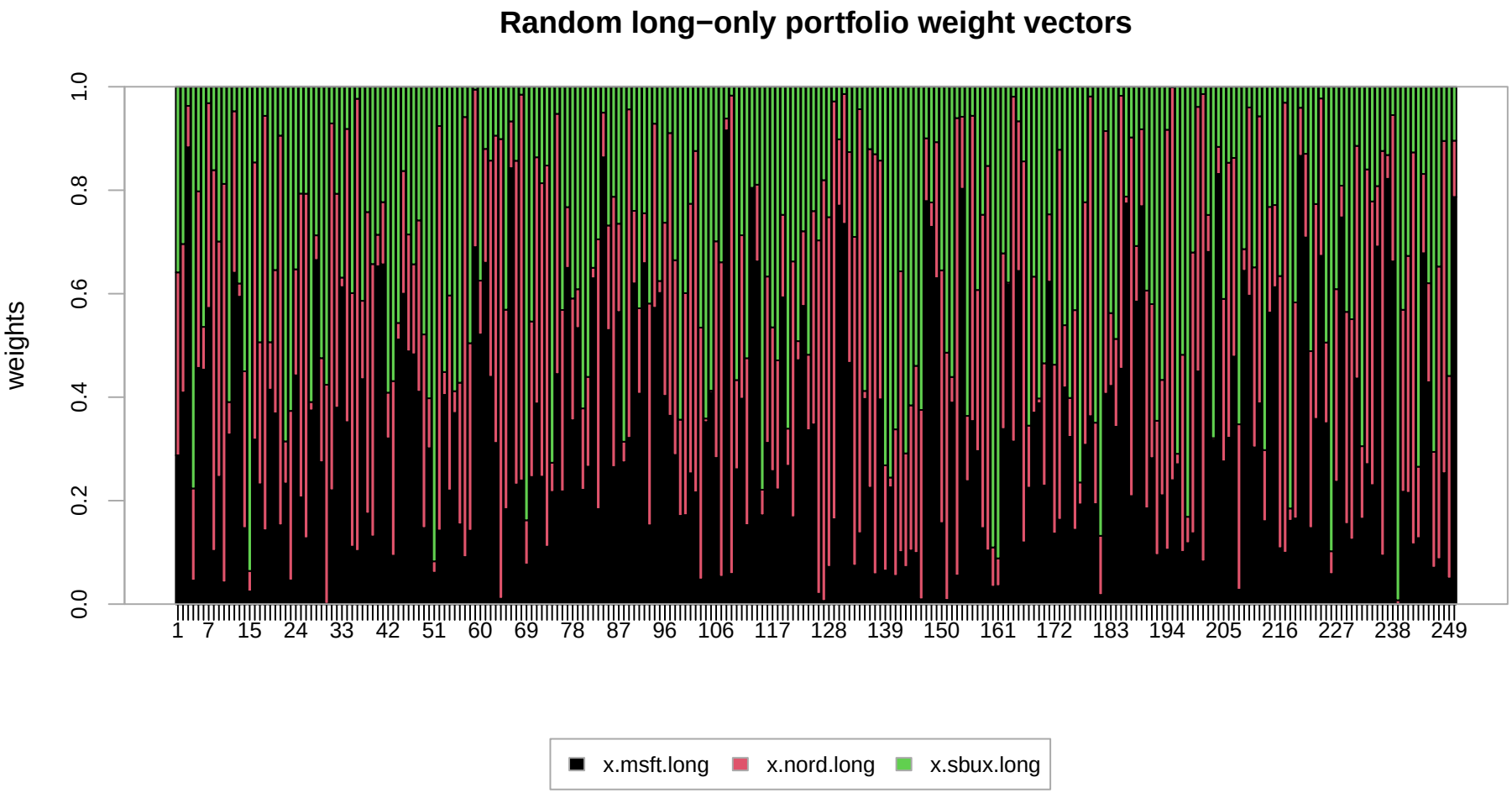
500 Random Portfolios: (No) Short sales

```
set.seed(123)
x.msft = runif(500, min=0, max=1)
x.nord = runif(500, min=0, max=1)
x.sbux = 1 - x.msft - x.nord
long.only = which(x.msft > 0 & x.nord > 0 & x.sbux > 0)
x.msft.long = x.msft[long.only]
x.nord.long = x.nord[long.only]
x.sbux.long = x.sbux[long.only]
length(long.only)
```

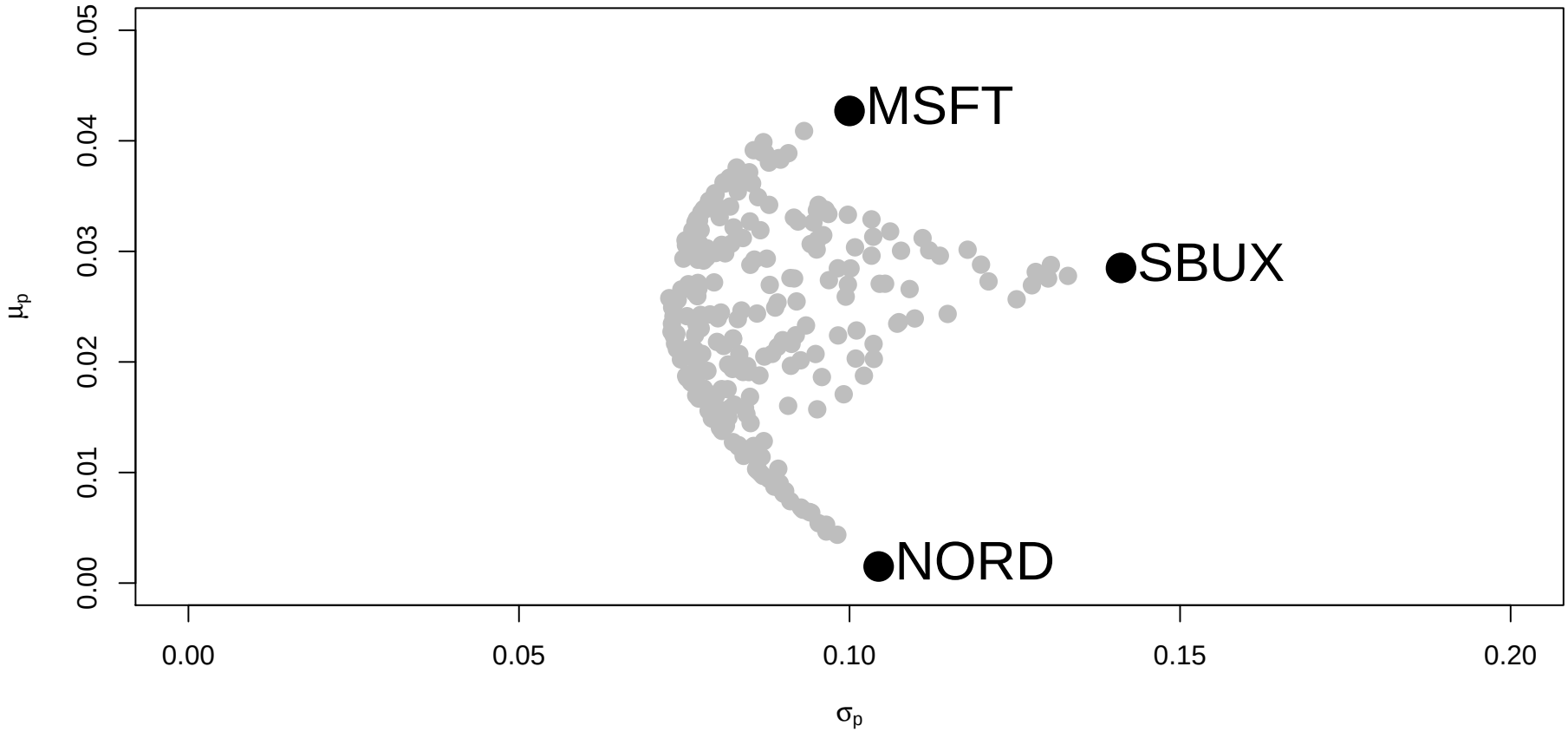
```
## [1] 250
```

- Of the 500 generated random portfolios only 250 portfolios are long only.

Random Portfolios: (No) Short sales



Long-only Random Portfolios



- Notice that it is not possible to find a no-short sales portfolio with a higher mean return than MSFT

R functions for Minimum Variance Portfolios with No Short Sales Restrictions

- **IntroCompFinR** functions allow for no-short sale restrictions (set optional argument `no.shorts=TRUE`)
- Use R package **quadprog** function `solve.QP()` - quadratic optimization subject to linear equality and inequality constraints
- **PortfolioAnalytics** package functions (from the authors of PerformanceAnalytics) available on R-forge (not ready yet for CRAN)

The Markowitz Algorithm with No Short Sales

In the R package **quadprog**, *Quadratic programming* problems have the form

$$\min_x \ \frac{1}{2}\mathbf{x}'\mathbf{D}\mathbf{x} - \mathbf{d}'\mathbf{x}$$
$$\mathbf{A}'_{neq}\mathbf{x} \geq \mathbf{b}_{neq} \text{ for } m \text{ inequality constraints}$$
$$\mathbf{A}'_{eq}\mathbf{x} = \mathbf{b}_{eq} \text{ for } l \text{ equality constraints}$$

where

- $\mathbf{D}$ is an $n \times n$ matrix, $\mathbf{x}$ and $\mathbf{d}$ are $n \times 1$ vectors,
- $\mathbf{A}'_{neq}$ is an $m \times n$ matrix, $\mathbf{b}_{neq}$ is an $m \times 1$ vector,
- $\mathbf{A}'_{eq}$ is an $l \times n$ matrix, and $\mathbf{b}_{eq}$ is an $l \times 1$ vector.

The Markowitz Algorithm with No Short Sales

Consider the portfolio optimization problem

$$\begin{aligned}
\min_{\mathbf{x}} \sigma_{p,x}^2 &= \mathbf{x}'\Sigma\mathbf{x} \quad \text{s.t.} \\
\mu_{p,x} &= \mathbf{x}'\mu = \mu_p^0 \\
\mathbf{x}'\mathbf{1} &= 1 \\
x_i &\geq 0 \ (i = 1, \ldots, n)
\end{aligned}$$

The Markowitz Algorithm with No Short Sales

For the objective function $\frac{1}{2}\mathbf{x}'\mathbf{D}\mathbf{x} - \mathbf{d}'\mathbf{x}$, set

$$\mathbf{D} = 2 \cdot \Sigma \text{ and } \mathbf{d} = (0, \ldots, 0)'$$

Then

$$\frac{1}{2}\mathbf{x}'\mathbf{D}\mathbf{x} - \mathbf{d}'\mathbf{x} = \mathbf{x}'\Sigma\mathbf{x}$$

The Markowitz Algorithm with No Short Sales

For the $l = 2$ equality constraints $\mathbf{A}'_{eq}\mathbf{x} = \mathbf{b}_{eq}$, we have

$$\begin{aligned}\mathbf{x}'\mu &= \mu'\mathbf{x} = \mu_p^0,\\ \mathbf{x}'\mathbf{1} &= \mathbf{1}'\mathbf{x} = 1,\end{aligned}$$

and so set

$$\underset{(2\times n)}{\mathbf{A}'_{eq}} = \begin{pmatrix} \mu' \\ \mathbf{1}' \end{pmatrix},$$

$$\underset{(2\times 1)}{\mathbf{b}_{eq}} = \begin{pmatrix} \mu_p^0 \\ 1 \end{pmatrix}.$$

The Markowitz Algorithm with No Short Sales

Then

$$\mathbf{A}'_{eq}\mathbf{x} = \begin{pmatrix} \mu' \\ \mathbf{1}' \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mu'\mathbf{x} \\ \mathbf{1}'\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mu_p^0 \\ 1 \end{pmatrix}$$

For the $m = n$ inequality constraints $\mathbf{A}'_{neq}\mathbf{x} \geq \mathbf{b}_{neq}$ we have

$$\begin{aligned} x_i \geq 0, \;\; i = 1, \ldots, n \\ \Rightarrow \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

so set

$$\underset{(n \times n)}{\mathbf{A}'_{neq}} = \mathbf{I}_n,$$

The Markowitz Algorithm with No Short Sales

The function `solve.QP()` assumes that the inequality constraints and equality constraints are combined into a single $(m + l) \times n$ matrix $\mathbf{A}'$ and a single $(m + l) \times 1$ vector $\mathbf{b}$ of the form

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}'_{eq} \\ \mathbf{A}'_{neq} \end{bmatrix}, \ \mathbf{b} = \begin{pmatrix} \mathbf{b}_{eq} \\ \mathbf{b}_{neq} \end{pmatrix}$$

For the portfolio problem we have

$$\mathbf{A}' = \begin{pmatrix} \mu' \\ \mathbf{1}' \\ \mathbf{I}_n \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} \mu_p^0 \\ 1 \\ \mathbf{0}_n \end{pmatrix}$$

where $\mathbf{0}_n = (0, \ldots, 0)'$.

The Global Minimum Variance Portfolio with No Short Sales

The optimization problem is

$$\min_{\mathbf{m}} \sigma^2_{p,m} = \mathbf{m}'\Sigma\mathbf{m} \text{ s.t.}$$
$$\mathbf{m}'\mathbf{1} = 1, \ \mathbf{m} \geq 0$$

Here, the restriction matrices are

$$\underset{(1\times n)}{\mathbf{A}'_{eq}} = \mathbf{1}', \ \underset{(1\times 1)}{\mathbf{b}_{eq}} = 1$$
$$\underset{(n\times n)}{\mathbf{A}'_{neq}} = \mathbf{I}_n \text{ and } \underset{(n\times 1)}{\mathbf{b}_{neq}} = (0,\ldots,0)'$$

The Global Minimum Variance Portfolio with No Short Sales

So that

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}'_{eq} \\ \mathbf{A}'_{neq} \end{bmatrix} = \begin{bmatrix} \mathbf{1}' \\ \mathbf{I}_n \end{bmatrix}, \ \mathbf{b} = \begin{pmatrix} \mathbf{b}_{eq} \\ \mathbf{b}_{neq} \end{pmatrix} = \begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix}$$

R Examples

Use the **IntroCompFinR** package function `globalMin.portfolio()` to find the global minimum variance portfolio allowing for short-sales

```
gmin.port = globalMin.portfolio(mu.vec, sigma.mat)
gmin.port
```

```
## Call:
## globalMin.portfolio(er = mu.vec, cov.mat = sigma.mat)
##
## Portfolio expected return:     0.0249
## Portfolio standard deviation:  0.0727
## Portfolio weights:
##  MSFT  NORD  SBUX
## 0.441 0.366 0.193
```

- Short sales constraint is not binding (no negative weights)

R Examples

Here, we illustrate how to use the **quadprog** function `solve.QP()` to find the short-sales restricted global minimum variance portfolio.

The function `solve.QP()` takes the restrictions matrices as inputs

```
args(solve.QP)
```

```
## function (Dmat, dvec, Amat, bvec, meq = 0, factorized = FAI
## NULL
```

R Examples

The restriction matrices for the global minimum variance optimization problem are:

```
D.mat = 2*sigma.mat
d.vec = rep(0, 3)
A.mat = cbind(rep(1,3), diag(3))
b.vec = c(1, rep(0,3))
```

R Examples

`solve.QP()` returns a list object containing information about the optimization

```
qp.out = solve.QP(Dmat=D.mat, dvec=d.vec,
                  Amat=A.mat, bvec=b.vec, meq=1)
names(qp.out)
```

```
## [1] "solution"
## [2] "value"
## [3] "unconstrained.solution"
## [4] "iterations"
## [5] "Lagrangian"
## [6] "iact"
```

R Examples

The portfolio weights are in the `solution` component

```
qp.out$solution
```

```
## [1] 0.441 0.366 0.193
```

Solution satisfies the constraints (weights sum to one and are positive)

```
sum(qp.out$solution)
```

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

The minimized value of the objective function (portfolio variance) is in the `value` component

```
qp.out$value
```

```
## [1] 0.00528
```

R Examples

Compute the mean and volatility of the minimum variance portfolio

```
er.gmin.ns = as.numeric(crossprod(qp.out$solution, mu.vec))
sd.gmin.ns = sqrt(qp.out$value)
c(er.gmin.ns, sd.gmin.ns)
```

```
## [1] 0.0249 0.0727
```

R Examples

The **IntroCompFinR** function `globlalMin.portfolio()` has an optional argument shorts=FALSE to impose no-short sales restrictions and uses `solve.QP()` to do the optimization.

```
gmin.port = globalMin.portfolio(mu.vec, sigma.mat,
                                shorts=FALSE)
gmin.port
```

```
## Call:
## globalMin.portfolio(er = mu.vec, cov.mat = sigma.mat, short
##
## Portfolio expected return:     0.0249
## Portfolio standard deviation:  0.0727
## Portfolio weights:
##  MSFT  NORD  SBUX
## 0.441 0.366 0.193
```

R Examples

Here we compute minimum variance portfolios with same mean as Microsoft. First find the minimum variance portfolio allowing short sales

```
eMsft.port = efficient.portfolio(mu.vec, sigma.mat,
                                 target.return = mu.vec["MSFT"])
eMsft.port
```

```
## Call:
## efficient.portfolio(er = mu.vec, cov.mat = sigma.mat, targe
##
## Portfolio expected return:     0.0427
## Portfolio standard deviation:  0.0917
## Portfolio weights:
##    MSFT    NORD    SBUX
##  0.8275 -0.0907  0.2633
```

- NORD is sold short in the unconstrained minimum variance portfolio.

R Examples

To find the the minimum variance portfolio not allowing short sales, set up the appropriate restriction matrices required by the function `solve.QP()`

```
D.mat = 2*sigma.mat
d.vec = rep(0, 3)
A.mat = cbind(mu.vec, rep(1,3), diag(3))
b.vec = c(mu.vec["MSFT"], 1, rep(0,3))
```

R Examples

Then use `solve.QP()` to find the solution

```
qp.out = solve.QP(Dmat=D.mat, dvec=d.vec,
                  Amat=A.mat, bvec=b.vec, meq=2)
names(qp.out$solution) = names(mu.vec)
round(qp.out$solution, digits=3)
```

```
## MSFT NORD SBUX
##    1    0    0
```

- The argument `meq = 2` specifies 2 equality constraints. It tells `solve.QP()` how to separate A_{eq} and A_{neq} from A
- Notice that the no-short sales solution forces the weights on Nordstrom and Starbucks to 0
- It is not possible to find an no-short sale efficient portfolio that has a higher mean than Microsoft

R Examples

The **IntroCompFinR** function `efficient.portfolio()` with `shorts=FALSE` uses `solve.QP()` to compute a no-short sales efficient portfolio:

```
efficient.portfolio(mu.vec, sigma.mat,
                    target.return = mu.vec["MSFT"],
                    shorts = FALSE)
```

```
## Call:
## efficient.portfolio(er = mu.vec, cov.mat = sigma.mat, targe
##     shorts = FALSE)
##
## Portfolio expected return:     0.0427
## Portfolio standard deviation:  0.1
## Portfolio weights:
## MSFT NORD SBUX
##    1    0    0
```

R Examples

Suppose you try to find an efficient portfolio with target return higher than the mean for Microsoft

```
# efficient.portfolio(mu.vec, sigma.mat,
# target.return = mu.vec["MSFT"]+0.01, shorts = FALSE)
# Error in quadprog::solve.QP(Dmat = Dmat, dvec = dvec,
# Amat = Amat, bvec = bvec,  :
#  constraints are inconsistent, no solution!
```

- Notice that `solve.QP()` throws an error saying there is no compatible solution

Finding the Efficient Frontier with No Short Sales

- Compute the global minimum variance portfolio with no-short sales
- Set an initial grid of target expected returns between the expected return on the global minimum variance portfolio with no short sales and the highest single asset expected return
- Solve the Markowitz algorithm with no-short sales for each target expected return in the grid
- No feasible solutions exist for target expected returns above the highest single asset expected return

R Examples

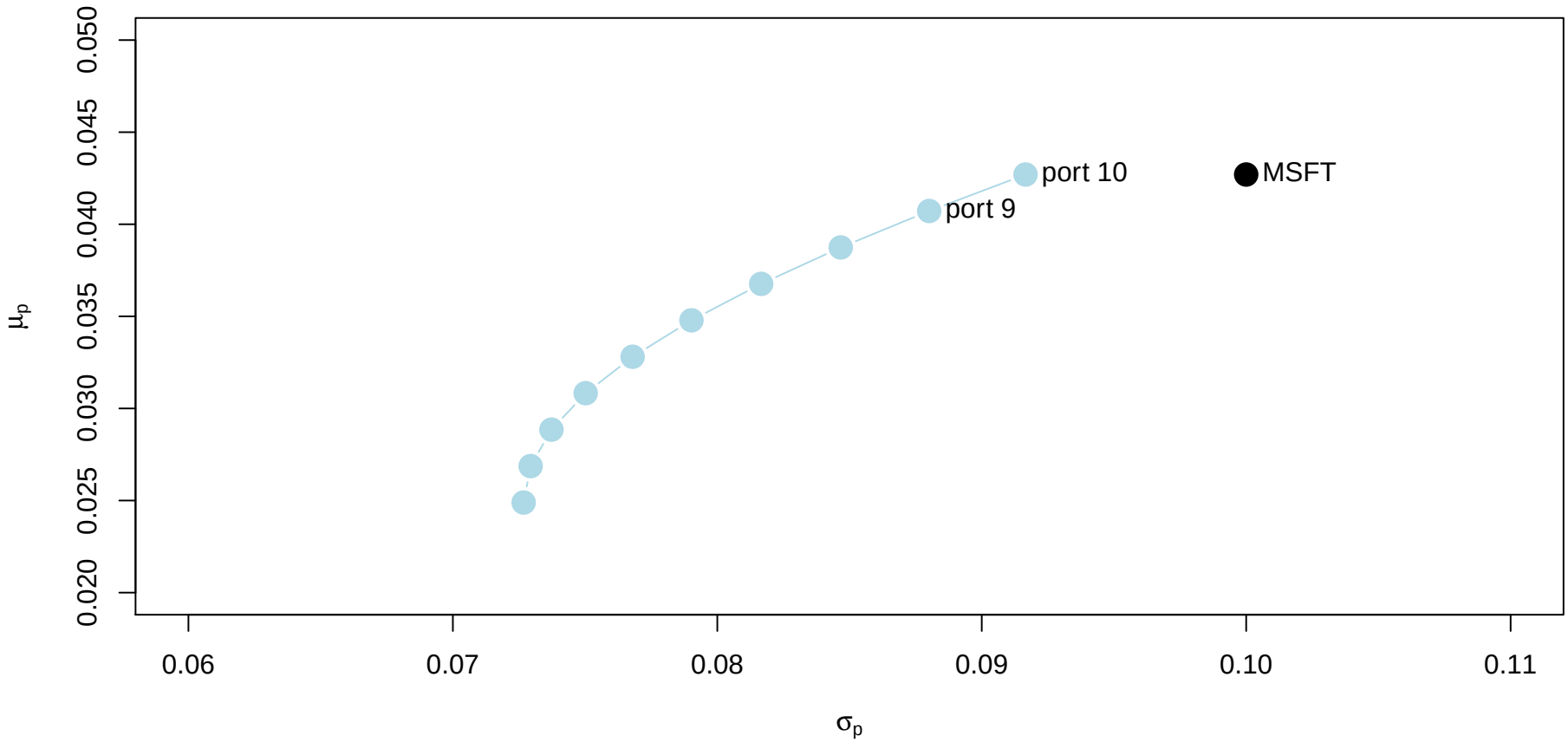
First, use the **IntroCompFinR** function `efficient.frontier()` to compute the efficient set of frontier portfolios that allow short sales

```
ef = efficient.frontier(mu.vec, sigma.mat, alpha.min=0,
                        alpha.max=1, nport=10)
tail(ef$weights, n=3)
```

```
##            MSFT    NORD  SBUX
## port 8   0.742  0.0107 0.248
## port 9   0.785 -0.0400 0.256
## port 10 0.827 -0.0907 0.263
```

- Notice that two efficient portfolios (Port 9 and Port 10) have negative weights (shorts) in NORD

R Examples



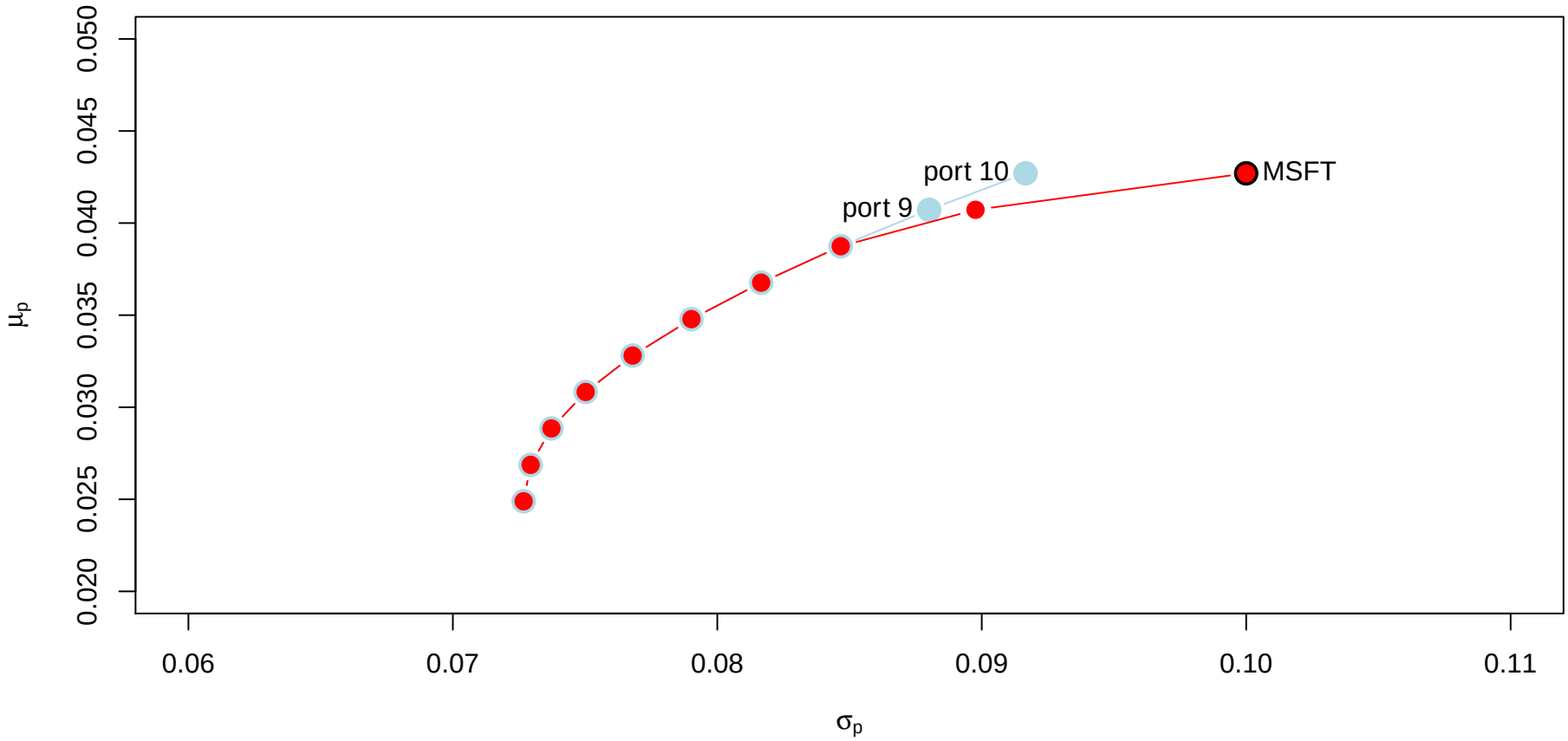
- port 9 and port 10 have negative weights in Nordstrom

R Examples

Compute short-sales restricted efficient frontier by running a simple loop

```
mu.vals = seq(gmin.port$er, max(mu.vec), length.out=10)
w.mat = matrix(0, length(mu.vals), 3)
sd.vals = rep(0, length(sd.vec))
colnames(w.mat) = names(mu.vec)
D.mat = 2*sigma.mat
d.vec = rep(0, 3)
A.mat = cbind(mu.vec, rep(1,3), diag(3))
for (i in 1:length(mu.vals)) {
  b.vec = c(mu.vals[i],1,rep(0,3))
  qp.out = solve.QP(Dmat=D.mat, dvec=d.vec,
                    Amat=A.mat, bvec=b.vec, meq=2)
  w.mat[i, ] = qp.out$solution
  sd.vals[i] = sqrt(qp.out$value)
}
```

R Examples



- Short-sales restricted efficient frontiner is in red

R Examples

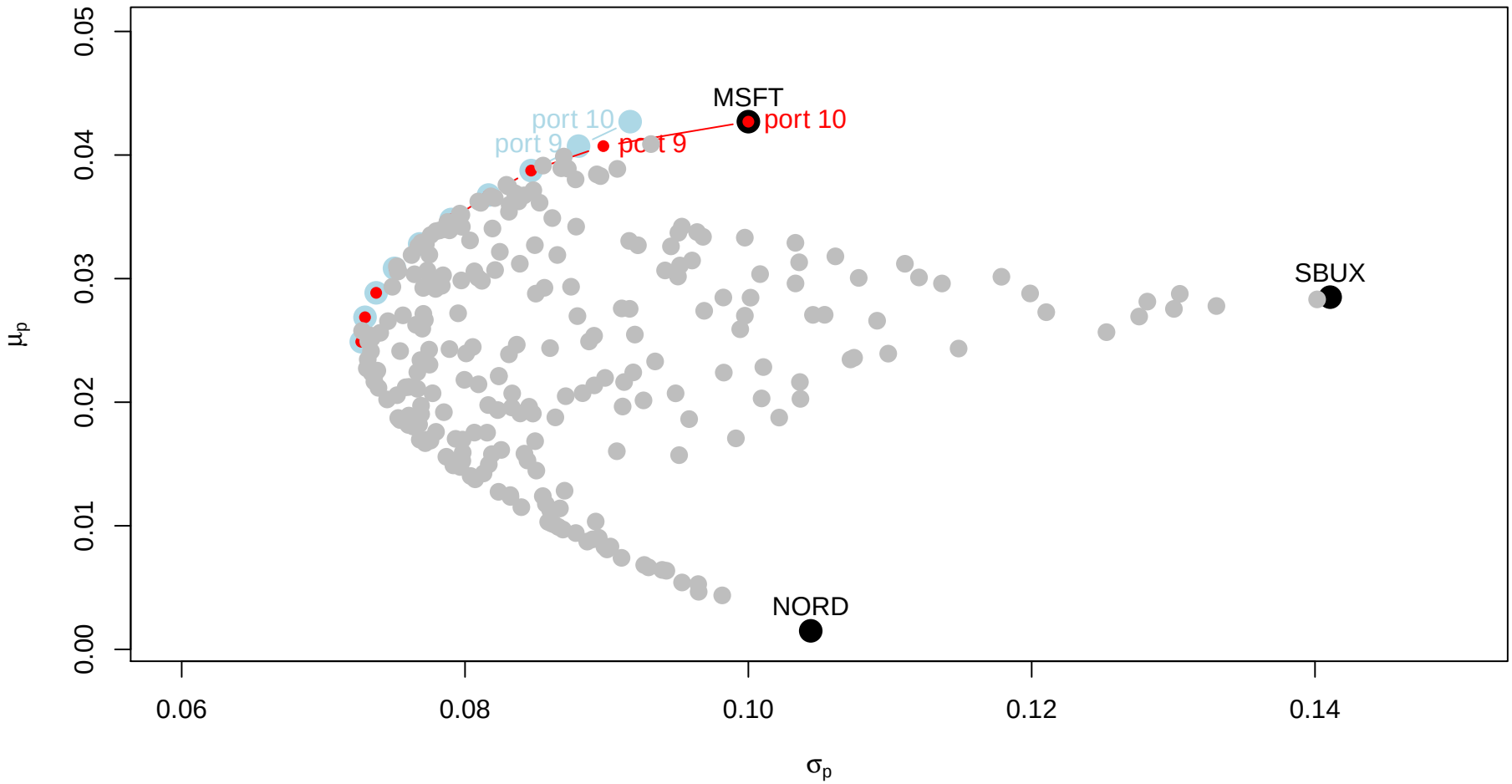
Use the **IntroCompFinR** function `efficient.frontier()` with `shorts=FALSE` to compute the efficient set of frontier portfolios that allow short sales

```
ef.ns = efficient.frontier(mu.vec, sigma.mat, alpha.min=0,
                           alpha.max=1, nport=10, shorts=FALSE)
tail(ef.ns$weights, n=3)
```

```
##            MSFT   NORD  SBUX
## port 8   0.742 0.0107 0.248
## port 9   0.861 0.0000 0.139
## port 10 1.000 0.0000 0.000
```

- Port 9 and Port 10 now have non-negative weights in NORD

R Examples


MSFT
port 10
port 10
port 9
port 9
SBUX
NORD
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.08
0.10
0.12
0.14
μ_p
σ_p

Short-Sales Restricted Tangency Portfolio

Maximize portfolio Sharpe ratio with added restriction of no short sales:

$$\max_{\mathbf{t}} \text{ Sharpe ratio} = \frac{\mu_{p,t} - r_f}{\sigma_{p,t}}$$

subject to

$$\mathbf{t}'\mathbf{1} = 1, \text{ and } \mathbf{t} \geq \mathbf{0}$$

Note: This maximization problem is not a quadratic programming problem. However, it can be re-formulated as a quadratic programming problem. See textbook for reformulation.

R Examples

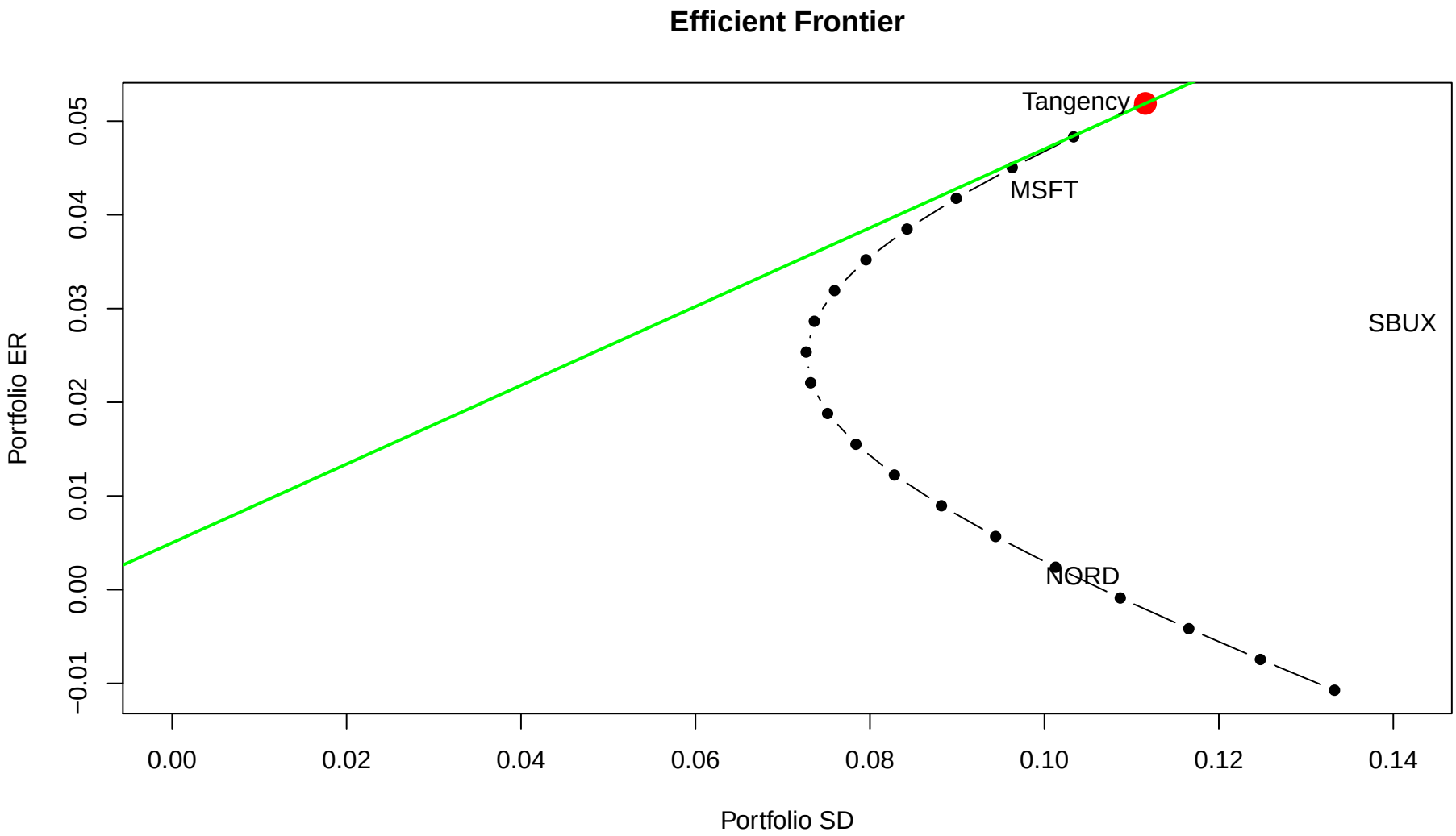
Unconstrained maximum Sharpe ratio (tangency) portfolio

First, compute maximum Sharpe ratio portfolio allowing short-sales:

```
r.f = 0.005
tan.port = tangency.portfolio(mu.vec, sigma.mat, r.f)
summary(tan.port, r.f)
```

```
## Call:
## tangency.portfolio(er = mu.vec, cov.mat = sigma.mat, risk.f
##
## Portfolio expected return:     0.0519
## Portfolio standard deviation:  0.112
## Portfolio Sharpe Ratio:        0.42
## Portfolio weights:
##    MSFT   NORD   SBUX
##   1.027 -0.326  0.299
```

R Examples


Efficient Frontier
Tangency
MSFT
SBUX
NORD
Portfolio ER
Portfolio SD
0.05
0.04
0.03
0.02
0.01
0.00
−0.01
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

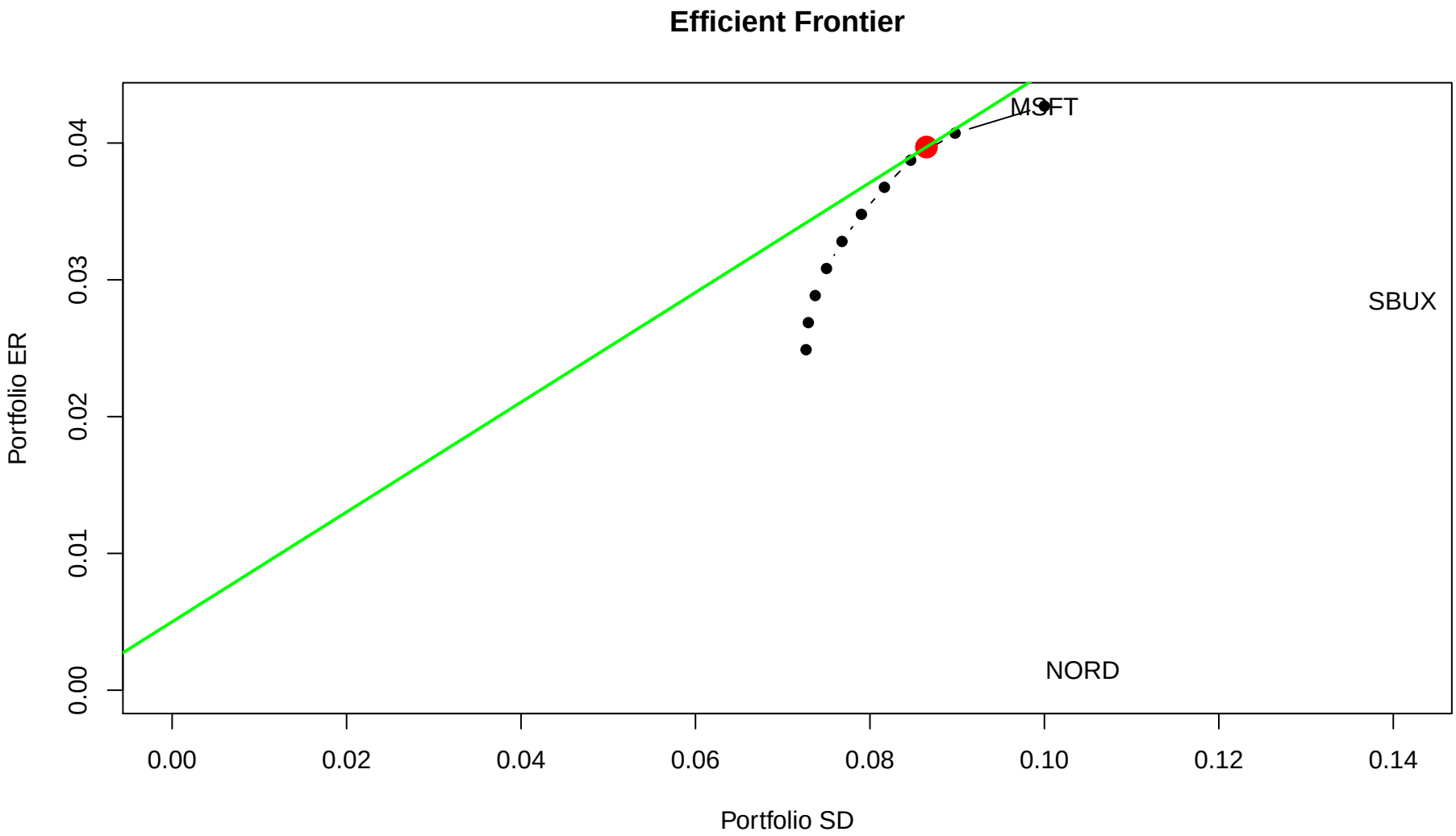
R Examples

Now, compute maximum Sharpe ratio portfolio not allowing short-sales:

```
tan.port.ns = tangency.portfolio(mu.vec, sigma.mat, r.f,
                                 shorts=FALSE)
summary(tan.port.ns, r.f)
```

```
## Call:
## tangency.portfolio(er = mu.vec, cov.mat = sigma.mat, risk.f
##     shorts = FALSE)
##
## Portfolio expected return:     0.0397
## Portfolio standard deviation:  0.0865
## Portfolio Sharpe Ratio:        0.401
## Portfolio weights:
##  MSFT  NORD  SBUX
## 0.789 0.000 0.211
```

R Examples


Efficient Frontier
MSFT
SBUX
NORD
Portfolio ER
0.00
0.01
0.02
0.03
0.04
Portfolio SD
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14