Portfolio Theory with Matrix Algebra

Eric Zivot

4/12/2021

Portfolio Math with Matrix Algebra

Three Risky Asset Example

Let R_i $(i = M, N, S)$ denote the return on asset i and assume that R_i follows GWN model:

$$R_i \sim iid \; N(\mu_i, \sigma_i^2)$$
$$\text{cov}(R_i, R_j) = \sigma_{ij}$$

Portfolio **x**

$$x_i = \text{share of wealth in asset } i$$
$$x_M + x_N + x_S = 1$$
$$R_{p,x} = x_M R_M + x_N R_N + x_S R_S.$$

Example Data

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

Stock i	μ_i	σ_i	Pair (i,j)	σ_{ij}
M (Microsoft)	0.0427	0.1000	(M,N)	0.0018
N (Nordstrom)	0.0015	0.1044	(M,S)	0.0011
S (Starbucks)	0.0285	0.1411	(N,S)	0.0026

Table 1: Three asset example data.

Risk free rate (to be used later): $r_f = 0.005$

Example Data

In matrix algebra, we have

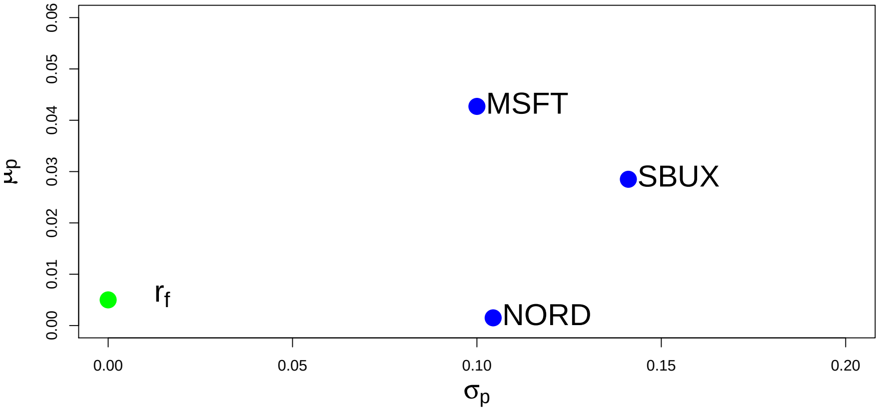
$$\mu = \begin{pmatrix} \mu_M \\ \mu_N \\ \mu_S \end{pmatrix} = \begin{pmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \sigma_M^2 & \sigma_{MN} & \sigma_{MS} \\ \sigma_{MN} & \sigma_N^2 & \sigma_{NS} \\ \sigma_{MS} & \sigma_{NS} & \sigma_S^2 \end{pmatrix} = \begin{pmatrix} (0.1000)^2 & 0.0018 & 0.0011 \\ 0.0018 & (0.1044)^2 & 0.0026 \\ 0.0011 & 0.0026 & (0.1411)^2 \end{pmatrix}$$

Example Data in R

```
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
```

Risk-Return Characteristics



- MSFT has highest Sharpe ratio, NORD has lowest Sharpe ratio

Portfolios with Matrix Algebra

$$\mathbf{R} = \begin{pmatrix} R_M \\ R_N \\ R_S \end{pmatrix}, \ \mu = \begin{pmatrix} \mu_M \\ \mu_N \\ \mu_S \end{pmatrix}, \ \mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} x_M \\ x_N \\ x_S \end{pmatrix}, \ \Sigma = \begin{pmatrix} \sigma_M^2 & \sigma_{MN} & \sigma_{MS} \\ \sigma_{MN} & \sigma_N^2 & \sigma_{NS} \\ \sigma_{MS} & \sigma_{NS} & \sigma_S^2 \end{pmatrix}$$

Portfolio weights sum to 1

$$\begin{aligned} \mathbf{x}'\mathbf{1} &= \begin{pmatrix} x_M & x_N & x_S \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\ &= x_M + x_N + x_S = 1 \end{aligned}$$

Portfolios with Matrix Algebra

Portfolio return

$$R_{p,x} = \mathbf{x}'\mathbf{R} = \begin{pmatrix} x_M & x_N & x_S \end{pmatrix} \begin{pmatrix} R_M \\ R_N \\ R_S \end{pmatrix}$$
$$= x_M R_M + x_N R_N + x_S R_S$$

Portfolio expected return

$$\mu_{p,x} = \mathbf{x}'\mu = \begin{pmatrix} x_M & x_N & x_S \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix}$$
$$= x_M \mu_M + x_N \mu_N + x_S \mu_S$$

Portfolios with Matrix Algebra

Portfolio variance

$$
\begin{aligned}
\sigma^2_{p,x} &= \mathbf{x}'\Sigma\mathbf{x} \\
&= \begin{pmatrix} x_M & x_N & x_S \end{pmatrix} \begin{pmatrix} \sigma^2_M & \sigma_{MN} & \sigma_{MS} \\ \sigma_{MN} & \sigma^2_N & \sigma_{NS} \\ \sigma_{MS} & \sigma_{NS} & \sigma^2_S \end{pmatrix} \begin{pmatrix} x_M \\ x_N \\ x_S \end{pmatrix} \\
&= x^2_M\sigma^2_M + x^2_N\sigma^2_N + x^2_S\sigma^2_S \\
&+ 2x_M x_N \sigma_{MN} + 2x_M x_S \sigma_{MS} + 2x_N x_S \sigma_{NS}
\end{aligned}
$$

Portfolio distribution

$$
R_{p,x} \sim N(\mu_{p,x}, \sigma^2_{p,x})
$$

Portfolios with Matrix Algebra

Covariance Between 2 Portfolio Returns

$$\mathbf{x} = \begin{pmatrix} x_M \\ x_N \\ x_S \end{pmatrix}, \ \mathbf{y} = \begin{pmatrix} y_M \\ y_N \\ y_S \end{pmatrix}$$
$$\mathbf{x}'\mathbf{1} = 1, \ \mathbf{y}'\mathbf{1} = 1$$

Portfolio returns

$$R_{p,x} = \mathbf{x}'\mathbf{R}, \ R_{p,y} = \mathbf{y}'\mathbf{R}$$

Covariance

$$\mathrm{cov}(R_{p,x}, R_{p,y}) = \mathbf{x}'\Sigma\mathbf{y} = \mathbf{y}'\Sigma\mathbf{x}$$

R Formulas

Let `x` and `y` be the vectors of portfolio weights, `muvec` the vector of expected returns, and `Sigma` the covariance matrix. Then

- $\mu_{p,x}$ = `t(x)*muvec` = `t(muvec)*x` = `crossprod(x,muvec)`
- $\sigma_{p,x}$ = `t(x)%*%Sigma%*%x`
- σ_{xy} = `t(x)%*%Sigma%*%y` = `t(y)%*%Sigma%*%x`

Example Portfolio: Equally-Weighted

```r
x.vec = rep(1,3)/3
names(x.vec) = asset.names
mu.p.x = crossprod(x.vec,mu.vec)
sig2.p.x = t(x.vec)%*%sigma.mat%*%x.vec
sig.p.x = sqrt(sig2.p.x)
```

Mean and volatility are:

```r
c(mu.p.x, sig.p.x)
```

```
## [1] 0.0242 0.0759
```

Example Portfolio: Long-Short

```
y.vec = c(0.8, 0.4, -0.2)
names(y.vec) = asset.names
mu.p.y = crossprod(y.vec,mu.vec)
sig2.p.y = t(y.vec)%*%sigma.mat%*%y.vec
sig.p.y = sqrt(sig2.p.y)
```

Mean and volatility are:

```
c(mu.p.y,sig.p.y)
```

```
## [1] 0.0291 0.0966
```

Covariance and Correlation between Example Portfolio Returns

Covariance:

```
sig.xy = t(x.vec)%*%sigma.mat%*%y.vec
sig.xy
```

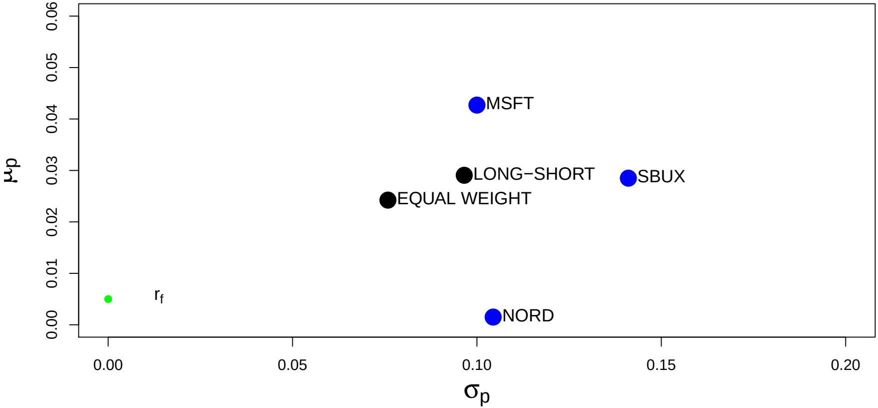
```
##          [,1]
## [1,] 0.00391
```

Correlation:

```
rho.xy = sig.xy/(sig.p.x*sig.p.y)
rho.xy
```

```
##        [,1]
## [1,] 0.533
```

Risk-Return characteristics: Example Portfolios



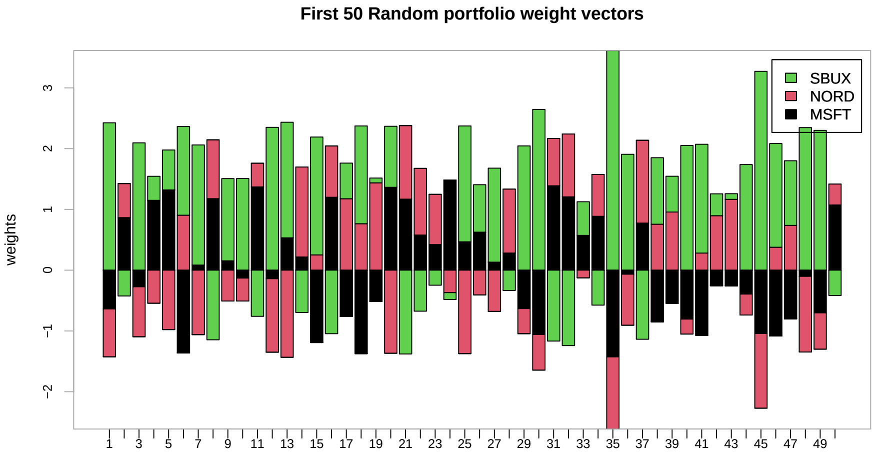
Random Portfolios

Create 600 random portfolio vectors with weights that sum to one.

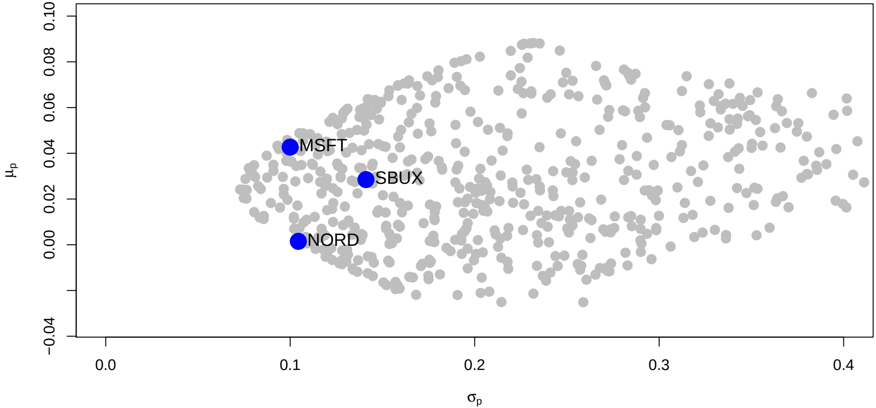
```
set.seed(123)
x.msft = runif(600, min=-1.5, max=1.5)
x.nord = runif(600, min=-1.5, max=1.5)
x.sbux = 1 - x.msft - x.nord
head(cbind(x.msft, x.nord, x.sbux), n=3)
```

```
##      x.msft x.nord x.sbux
## [1,] -0.637 -0.788  2.426
## [2,]  0.865  0.559 -0.424
## [3,] -0.273 -0.823  2.096
```

Portfolio Weights: Random Portfolios



Risk-Return Characteristics: Random Portfolios



Risk-Return Characteristics: Random Portfolios

Comments:

- With more than two assets, set of feasible portfolios is no longer one side of a hyperbole
- Set of feasible portfolios is a solid space (e.g. grey dots fill out solid space as number of portfolios increase)
- Efficient portfolios are on the upper boundary (above minimimum variance portfolio)

Global Minimum Variance Portfolio

Problem: Find the portfolio $\mathbf{m} = (m_M, m_N, m_S)'$ that solves

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

That is, find the portfolio that has the smallest possible variance (volatility). We can do this in two ways:

- Analytic solution using calculus and matrix algebra
- Numerical Solution in R or Excel Using numerical optimizers

Review: Derivatives of Simple Matrix Functions

Let $\mathbf{A}$ be an $n \times n$ symmetric matrix, and let $\mathbf{x}$ and $\mathbf{y}$ be an $n \times 1$ vectors. Then

$$\underset{n\times 1}{\frac{\partial}{\partial \mathbf{x}}}\mathbf{x}'\mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{y} \\ \vdots \\ \frac{\partial}{\partial x_n}\mathbf{x}'\mathbf{y} \end{pmatrix} = \mathbf{y}, \tag{1}$$

$$\underset{n\times 1}{\frac{\partial}{\partial \mathbf{x}}}\mathbf{x}'\mathbf{A}\mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{A}\mathbf{x} \\ \vdots \\ \frac{\partial}{\partial x_n}\mathbf{x}'\mathbf{A}\mathbf{x} \end{pmatrix} = 2\mathbf{A}\mathbf{x}. \tag{2}$$

Analytic solution using matrix algebra

The Lagrangian is

$$L(\mathbf{m},\lambda) = \mathbf{m}'\Sigma\mathbf{m} + \lambda(\mathbf{m}'\mathbf{1} - 1)$$

First order conditions (use matrix derivative results)

$$\underset{(3\times1)}{\mathbf{0}} = \frac{\partial L(\mathbf{m},\lambda)}{\partial\mathbf{m}} = \frac{\partial\mathbf{m}'\Sigma\mathbf{m}}{\partial\mathbf{m}} + \frac{\partial}{\partial\mathbf{m}}\lambda(\mathbf{m}'\mathbf{1}-1) = 2\cdot\Sigma\mathbf{m}+\lambda\mathbf{1}$$
$$\underset{(1\times1)}{0} = \frac{\partial L(\mathbf{m},\lambda)}{\partial\lambda} = \frac{\partial\mathbf{m}'\Sigma\mathbf{m}}{\partial\lambda} + \frac{\partial}{\partial\lambda}\lambda(\mathbf{m}'\mathbf{1}-1) = \mathbf{m}'\mathbf{1}-1$$

Analytic solution using matrix algebra

Write FOCs in matrix form as

$$\begin{pmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}' & 0 \end{pmatrix} \begin{pmatrix} \mathbf{m} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \begin{matrix} 3 \times 1 \\ 1 \times 1 \end{matrix} .$$

The FOCs are the linear system

$$\mathbf{A}_m \mathbf{z}_m = \mathbf{b}$$

where

$$\mathbf{A}_m = \begin{pmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}' & 0 \end{pmatrix}, \ \mathbf{z}_m = \begin{pmatrix} \mathbf{m} \\ \lambda \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}.$$

Analytic solution using matrix algebra

The solution for $\mathbf{z}_m$ is

$$\mathbf{z}_m = \mathbf{A}_m^{-1}\mathbf{b}.$$

- The first three elements of $\mathbf{z}_m$ are the portfolio weights $\mathbf{m} = (m_M, m_N, m_S)'$ for the global minimum variance portfolio with expected return $\mu_{p,m} = \mathbf{m}'\mu$ and variance $\sigma^2_{p,m} = \mathbf{m}'\Sigma\mathbf{m}$.
- The fourth element is the Lagrange multiplier λ

Alternative Derivation of Global Minimum Variance Portfolio

The first order conditions from the optimization problem can be expressed in matrix notation as

$$\underset{(3\times 1)}{\mathbf{0}} = \frac{\partial L(\mathbf{m},\lambda)}{\partial \mathbf{m}} = 2\cdot\Sigma\mathbf{m} + \lambda\cdot\mathbf{1},$$

$$\underset{(1\times 1)}{0} = \frac{\partial L(\mathbf{m},\lambda)}{\partial \lambda} = \mathbf{m}'\mathbf{1} - 1.$$

Using first equation, solve for $\mathbf{m}$:

$$\mathbf{m} = -\frac{1}{2}\cdot\lambda\Sigma^{-1}\mathbf{1}.$$

Alternative Derivation of Global Minimum Variance Portfolio

Next, multiply both sides by $\mathbf{1}'$ and use second equation to solve for λ:

$$\begin{aligned}\mathbf{1}'\mathbf{m} &= -\frac{1}{2}\cdot\lambda\mathbf{1}'\Sigma^{-1}\mathbf{1} = 1 \\ &\Rightarrow \lambda = -2\cdot\frac{1}{\mathbf{1}'\Sigma^{-1}\mathbf{1}}.\end{aligned}$$

Finally, substitute the value for λ in the equation for $\mathbf{m}$:

$$\begin{aligned}\mathbf{m} &= -\frac{1}{2}\left(-2\right)\frac{1}{\mathbf{1}'\Sigma^{-1}\mathbf{1}}\Sigma^{-1}\mathbf{1} \\ &= \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}'\Sigma^{-1}\mathbf{1}}.\end{aligned}$$

Global Minimum Variance Portfolio: Example Data Method 1

Use $\mathbf{z}_m = \mathbf{A}_m^{-1}\mathbf{b}$.

```
top.mat = cbind(2*sigma.mat, rep(1, 3))
bot.vec = c(rep(1, 3), 0)
Am.mat = rbind(top.mat, bot.vec)
b.vec = c(rep(0, 3), 1)
z.m.mat = solve(Am.mat)%*%b.vec
m.vec = z.m.mat[1:3,1]
m.vec
```

```
##  MSFT  NORD  SBUX
## 0.441 0.366 0.193
```

Global Minimum Variance Portfolio: Example Data Method 1

Mean and volatility of minimum variance portfolio

```
mu.gmin = as.numeric(crossprod(m.vec, mu.vec))
sig2.gmin = as.numeric(t(m.vec)%*%sigma.mat%*%m.vec)
sig.gmin = sqrt(sig2.gmin)
c(mu.gmin, sig.gmin)
```

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

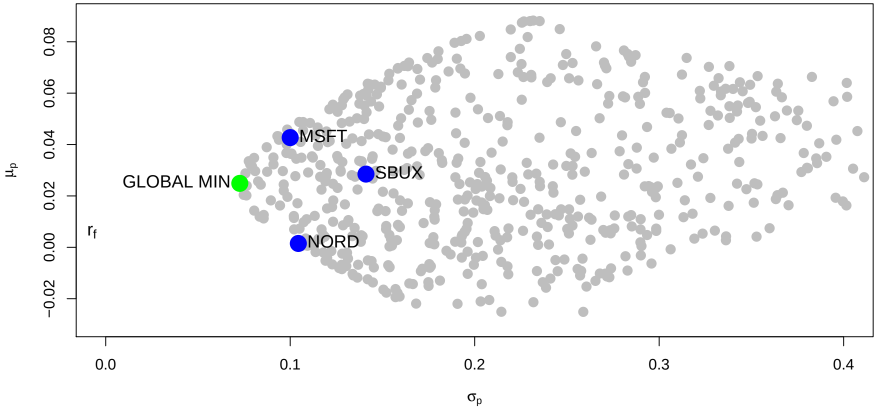
Global Minimum Variance Portfolio: Example Data Method 2

Use analytic formula for minimum variance portfolio

```
one.vec = rep(1, 3)
sigma.inv.mat = solve(sigma.mat)
top.mat = sigma.inv.mat%*%one.vec
bot.val = as.numeric((t(one.vec)%*%sigma.inv.mat%*%one.vec))
m.mat = top.mat/bot.val
m.mat[, 1]
```

```
##  MSFT  NORD  SBUX
## 0.441 0.366 0.193
```

Plot the Global Minimum Variance Portfolio



Efficient Portfolios of Risky Assets: Markowitz Algorithm

Problem 1: find portfolio $\mathbf{x}$ that has the highest expected return for a given level of risk as measured by portfolio variance

$$
\begin{aligned}
\max_{x_A, x_B, x_C} \mu_{p,x} &= \mathbf{x}'\mu \text{ s.t} \\
\sigma^2_{p,x} &= \mathbf{x}'\Sigma\mathbf{x} = \sigma^0_p = \text{ target risk} \\
\mathbf{x}'\mathbf{1} &= 1
\end{aligned}
$$

Efficient Portfolios of Risky Assets: Markowitz Algorithm

Efficient Portfolios of Risky Assets: Markowitz Algorithm

Problem 2 (Dual): find portfolio $\mathbf{x}$ that has the smallest risk, measured by portfolio variance, that achieves a target expected return.

$$\begin{aligned} \min_{x_A, x_B, x_C} \sigma^2_{p,x} &= \mathbf{x}'\Sigma\mathbf{x} \text{ s.t.} \\ \mu_{p,x} &= \mathbf{x}'\mu = \mu^0_p = \text{targetreturn} \\ \mathbf{x}'\mathbf{1} &= 1 \end{aligned}$$

Remark: Problem 2 is usually solved in practice by varying the target return between a given range.

Efficient Portfolios of Risky Assets: Markowitz Algorithm

Solving for Efficient Portfolios

- Analytic solution using matrix algebra
- Numerical solution using optimizers in R or Excel

Analytic solution using matrix algebra

The Lagrangian function associated with Problem 2 is

$$L(x, \lambda_1, \lambda_2) = \mathbf{x}'\Sigma\mathbf{x} + \lambda_1(\mathbf{x}'\mu - \mu_{p,0}) + \lambda_2(\mathbf{x}'\mathbf{1} - 1)$$

The FOCs are

$$\begin{aligned}
\underset{(3\times1)}{\mathbf{0}} &= \frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \mathbf{x}} = 2\Sigma\mathbf{x} + \lambda_1\mu + \lambda_2\mathbf{1}, \\
\underset{(1\times1)}{0} &= \frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \lambda_1} = \mathbf{x}'\mu - \mu_{p,0}, \\
\underset{(1\times1)}{0} &= \frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \lambda_2} = \mathbf{x}'\mathbf{1} - 1.
\end{aligned}$$

These FOCs consist of five linear equations in five unknowns

Analytic solution using matrix algebra

We can represent the FOCs in matrix notation as

$$\begin{pmatrix} 2\Sigma & \mu & \mathbf{1} \\ \mu' & 0 & 0 \\ \mathbf{1}' & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mu_{p,0} \\ 1 \end{pmatrix}$$

or

$$\mathbf{A}_x \mathbf{z}_x = \mathbf{b}_0$$

where

$$\mathbf{A}_x = \begin{pmatrix} 2\Sigma & \mu & \mathbf{1} \\ \mu' & 0 & 0 \\ \mathbf{1}' & 0 & 0 \end{pmatrix}, \ \mathbf{z}_x = \begin{pmatrix} \mathbf{x} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \text{ and } \mathbf{b}_0 = \begin{pmatrix} \mathbf{0} \\ \mu_{p,0} \\ 1 \end{pmatrix}$$

Analytic solution using matrix algebra

The solution for $\mathbf{z}_x$ is then

$$\mathbf{z}_x = \mathbf{A}_x^{-1}\mathbf{b}_0.$$

The first three elements of $\mathbf{z}_x$ are the portfolio weights $\mathbf{x} = (x_M, x_N, x_S)'$ for the efficient portfolio with expected return $\mu_{p,x} = \mu_{p,0}$.

Example: Find efficient portfolios with the same expected return as MSFT and SBUX

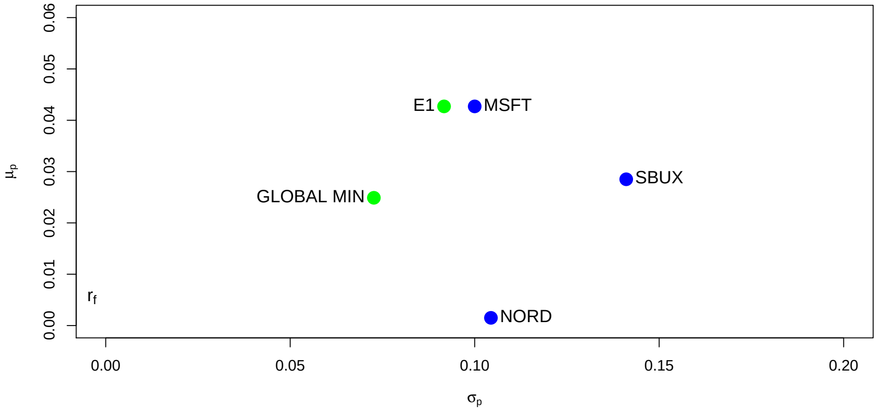
For MSFT, we solve

$$\begin{aligned} \min_{x_A, x_B, x_C} \sigma^2_{p,x} &= \mathbf{x}'\Sigma\mathbf{x} \text{ s.t.} \\ \mu_{p,x} &= \mathbf{x}'\mu = \mu_{MSFT} = 0.0427 \\ \mathbf{x}'\mathbf{1} &= 1 \end{aligned}$$

For SBUX, we solve

$$\begin{aligned} \min_{y_A, y_B, y_C} \sigma^2_{p,x} &= \mathbf{y}'\Sigma\mathbf{y} \text{ s.t.} \\ \mu_{p,y} &= \mathbf{y}'\mu = \mu_{SBUX} = 0.0285 \\ \mathbf{y}'\mathbf{1} &= 1 \end{aligned}$$

Efficient portfolio with the same mean as Microsoft



- Point E1 is the efficient portfolio with the same mean as Microsoft

Efficient portfolio with the same mean as Microsoft

Use matrix algebra formula to compute efficient portfolio.

```
top.mat = cbind(2*sigma.mat, mu.vec, rep(1, 3))
mid.vec = c(mu.vec, 0, 0)
bot.vec = c(rep(1, 3), 0, 0)
Ax.mat = rbind(top.mat, mid.vec, bot.vec)
bmsft.vec = c(rep(0, 3), mu.vec["MSFT"], 1)
z.mat = solve(Ax.mat)%*%bmsft.vec
x.vec = z.mat[1:3,]
x.vec
```

```
##    MSFT    NORD    SBUX
##  0.8275 -0.0907  0.2633
```

Efficient portfolio with the same mean as MSFT

Compute mean and volatility of efficient portfolio.

```
mu.px = as.numeric(crossprod(x.vec, mu.vec))
sig2.px = as.numeric(t(x.vec)%*%sigma.mat%*%x.vec)
sig.px = sqrt(sig2.px)
c(mu.px,sig.px)
```

```
## [1] 0.0427 0.0917
```

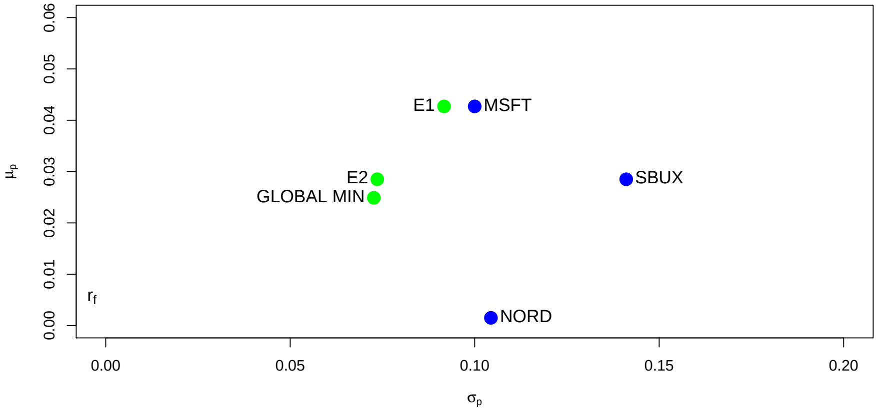
Compare with mean and volatility of MSFT.

```
c(mu.vec["MSFT"],sd.vec["MSFT"])
```

```
##   MSFT   MSFT
## 0.0427 0.1000
```

- Efficient portfolio has slightly smaller volatility than Microsoft.
- Microsoft is near the efficient frontier boundary

Efficient portfolio with the same mean as SBUX



- Point E2 is the efficient portfolio withe the same mean as Starbucks

Efficient portfolio with the same mean as SBUX

Use matrix algebra formula to compute efficient portfolio. Note: only have to change $\mathbf{b_0}$.

```
bsbux.vec = c(rep(0, 3), mu.vec["SBUX"], 1)
z.mat = solve(Ax.mat)%*%bsbux.vec
y.vec = z.mat[1:3,]
y.vec
```

```
##  MSFT  NORD  SBUX
## 0.519 0.273 0.207
```

Efficient portfolio with the same mean as SBUX

Compute mean and volatility of efficient portfolio.

```
mu.py = as.numeric(crossprod(y.vec, mu.vec))
sig2.py = as.numeric(t(y.vec)%*%sigma.mat%*%y.vec)
sig.py = sqrt(sig2.py)
c(mu.py,sig.py)
```

```
## [1] 0.0285 0.0736
```

Compare with mean and volatility of SBUX.

```
c(mu.vec["SBUX"],sd.vec["SBUX"])
```

```
##   SBUX   SBUX
## 0.0285 0.1411
```

- Efficient portfolio has much smaller volatility than Starbucks!
- Starbucks is far away from the efficient frontier boundary

Covariance between Efficient Portfolio Returns

Later on, we will use the covariance between the two efficient portfolios.

```
sigma.xy = as.numeric(t(x.vec)%*%sigma.mat%*%y.vec)
rho.xy = sigma.xy/(sig.px*sig.py)
c(sigma.xy, rho.xy)
```

```
## [1] 0.00591 0.87722
```

Computing the Portfolio Frontier

Result: The portfolio frontier can be represented as convex combinations of any two frontier portfolios. Let $\mathbf{x}$ be a frontier portfolio that solves

$$
\begin{aligned}
\min_{\mathbf{x}} \sigma_{p,x}^2 &= \mathbf{x}'\Sigma\mathbf{x} \text{ s.t.} \\
\mu_{p,x} &= \mathbf{x}'\mu = \mu_p^0 \\
\mathbf{x}'\mathbf{1} &= 1
\end{aligned}
$$

Let $\mathbf{y} \neq \mathbf{x}$ be another frontier portfolio that solves

$$
\begin{aligned}
\min_{\mathbf{y}} \sigma_{p,y}^2 &= \mathbf{y}'\Sigma\mathbf{y} \text{ s.t.} \\
\mu_{p,y} &= \mathbf{y}'\mu = \mu_p^1 \neq \mu_p^0 \\
\mathbf{y}'\mathbf{1} &= 1
\end{aligned}
$$

Computing the Portfolio Frontier

Let α be any constant. Then the portfolio

$$\mathbf{z} = \alpha \cdot \mathbf{x} + (1-\alpha) \cdot \mathbf{y}$$

is a frontier portfolio. Furthermore

$$\begin{aligned}
\mu_{p,z} &= \mathbf{z}'\mu = \alpha \cdot \mu_{p,x} + (1-\alpha)\mu_{p,y} \\
\sigma_{p,z}^2 &= \mathbf{z}'\Sigma\mathbf{z} \\
&= \alpha^2\sigma_{p,x}^2 + (1-\alpha)^2\sigma_{p,y}^2 + 2\alpha(1-\alpha)\sigma_{x,y} \\
\sigma_{x,y} &= \mathrm{cov}(R_{p,x}, R_{p,y}) = \mathbf{x}'\Sigma\mathbf{y}
\end{aligned}$$

Example: 3 Asset Case

A convex combination of two frontier portfolios is another frontier portfolio:

$$
\begin{aligned}
\mathbf{z} &= \alpha \cdot \mathbf{x} + (1-\alpha) \cdot \mathbf{y} \\
&= \alpha \cdot \begin{pmatrix} x_M \\ x_N \\ x_S \end{pmatrix} + (1-\alpha) \begin{pmatrix} y_M \\ y_N \\ y_S \end{pmatrix} \\
&= \begin{pmatrix} \alpha x_M + (1-\alpha) y_M \\ \alpha x_N + (1-\alpha) y_N \\ \alpha x_S + (1-\alpha) y_S \end{pmatrix} = \begin{pmatrix} z_M \\ z_N \\ z_S \end{pmatrix}
\end{aligned}
$$

Example: 3 Asset Case

- Compute efficient portfolio as convex combination of efficient portfolio with same mean as MSFT and efficient portfolio with same mean as SBUX.
- Let **x** denote the efficient portfolio with the same mean as MSFT, **y** denote the efficient portfolio with the same mean as SBUX, and let $\alpha = 0.5$. Then

$$
\begin{aligned}
\mathbf{z} &= \alpha \cdot \mathbf{x} + (1-\alpha)\cdot \mathbf{y} \\
&= 0.5 \cdot \begin{pmatrix} 0.82745 \\ -0.09075 \\ 0.26329 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 0.5194 \\ 0.2732 \\ 0.2075 \end{pmatrix} \\
&= \begin{pmatrix} (0.5)(0.82745) \\ (0.5)(-0.09075) \\ (0.5)(0.26329) \end{pmatrix} + \begin{pmatrix} (0.5)(0.5194) \\ (0.5)(0.2732) \\ (0.5)(0.2075) \end{pmatrix} = \begin{pmatrix} 0.6734 \\ 0.0912 \\ 0.2354 \end{pmatrix} = \begin{pmatrix} z_M \\ z_N \\ z_S \end{pmatrix}
\end{aligned}
$$

Example: 3 Asset Case

The mean of this portfolio can be computed using:

$$\mu_{p,z} = \mathbf{z}'\mu = (0.6734, 0.0912, 0.2354)' \begin{pmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{pmatrix} = 0.0356$$

$$\mu_{p,z} = \alpha \cdot \mu_{p,x} + (1-\alpha)\mu_{p,y} = 0.5(0.0427) + (0.5)(0.0285) = 0.0356$$

The variance can be computed using

$$\begin{aligned} \sigma^2_{p,z} &= \mathbf{z}'\Sigma\mathbf{z} = 0.00641 \\ \sigma^2_{p,z} &= \alpha^2\sigma^2_{p,x} + (1-\alpha)^2\sigma^2_{p,y} + 2\alpha(1-\alpha)\sigma_{xy} \\ &= (0.5)^2(0.09166)^2 + (0.5)^2(0.07355)^2 + 2(0.5)(0.5)(0.005914) = 0.006 \end{aligned}$$

Example: R Code

The weight vector **z** is determined using

```
a = 0.5
z.vec = a*x.vec + (1-a)*y.vec
z.vec
```

```
##   MSFT   NORD   SBUX
## 0.6734 0.0912 0.2354
```

The mean and volatility are

```
sigma.xy = as.numeric(t(x.vec)%*%sigma.mat%*%y.vec)
mu.pz = as.numeric(crossprod(z.vec, mu.vec))
sig2.pz = as.numeric(t(z.vec)%*%sigma.mat%*%z.vec)
sig.pz = sqrt(sig2.pz)
c(mu.pz, sig.pz)
```

```
## [1] 0.0356 0.0801
```

Example: 3 Asset Case

Next, find an efficient portfolio with expected return 0.05 from two efficient portfolios. Let **x** denote the efficient portfolio with the same mean as MSFT, **y** denote the efficient portfolio with the same mean as SBUX, and let $\mu_{p,z} = 0.05$. Then use

$$0.05 = \mu_{p,z} = \alpha \cdot \mu_{p,x} + (1 - \alpha)\mu_{p,y}$$

to solve for α :

$$\alpha = \frac{0.05 - \mu_{p,y}}{\mu_{p,x} - \mu_{p,y}} = \frac{0.05 - 0.0285}{0.0427 - 0.0285} = 1.514$$

Example: 3 Asset Case

Solve for portfolio weights using

$$
\begin{aligned}
\mathbf{z} &= \alpha \cdot \mathbf{x} + (1-\alpha) \cdot \mathbf{y} \\
&= 1.514 \begin{pmatrix} 0.8275 \\ -0.0908 \\ 0.2633 \end{pmatrix} - 0.514 \begin{pmatrix} 0.5194 \\ 0.2732 \\ 0.2075 \end{pmatrix} = \begin{pmatrix} 0.9858 \\ -0.2778 \\ 0.2920 \end{pmatrix}
\end{aligned}
$$

Example: R Code

Given a target mean value, $\mu_0 = 0.05$, you can solve for α.

```
a.05 = (0.05 - mu.py)/(mu.px - mu.py)
a.05
```

```
## [1] 1.51
```

Given $\alpha = 1.514$ solve for $\mathbf{z}$.

```
z.05 = a.05*x.vec + (1 - a.05)*y.vec
z.05
```

```
##   MSFT   NORD   SBUX
##  0.986 -0.278  0.292
```

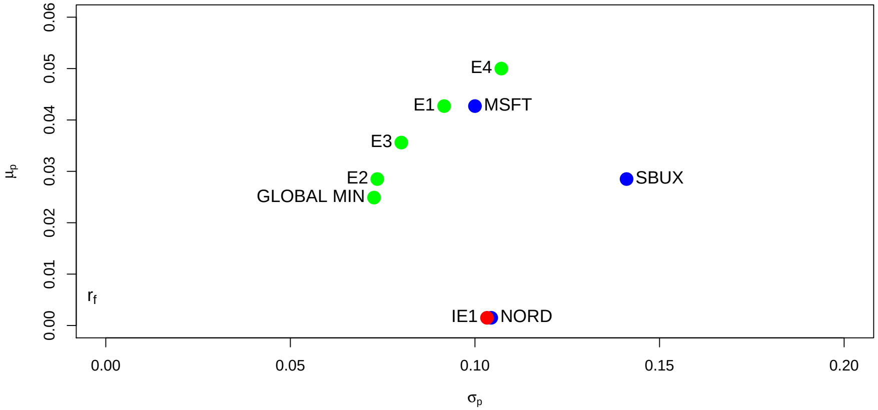
Example: R Code

Compute the mean and volatility.

```
mu.pz.05 = as.numeric(crossprod(z.05,mu.vec))
sig.pz.05 = as.numeric(sqrt(t(z.05)%*%sigma.mat%*%z.05))
c(mu.pz.05,sig.pz.05)
```

```
## [1] 0.050 0.107
```

Show Example Frontier Portfolios



- Point E3 is the efficient portfolio with $\alpha = 0.5$
- Point E4 is the efficient portfolio with expected return 0.05

Strategy for Plotting Portfolio Frontier

1. Set global minimum variance portfolio = first frontier portfolio

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

and compute $\mu_{p,m} = \mathbf{m}'\mu$

2. Find asset i that has highest expected return. Set target return to $\mu^0 = \max(\mu)$ and solve

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

Strategy for Plotting Portfolio Frontier

3. Create grid of α values, initially between 0 and 1, and compute

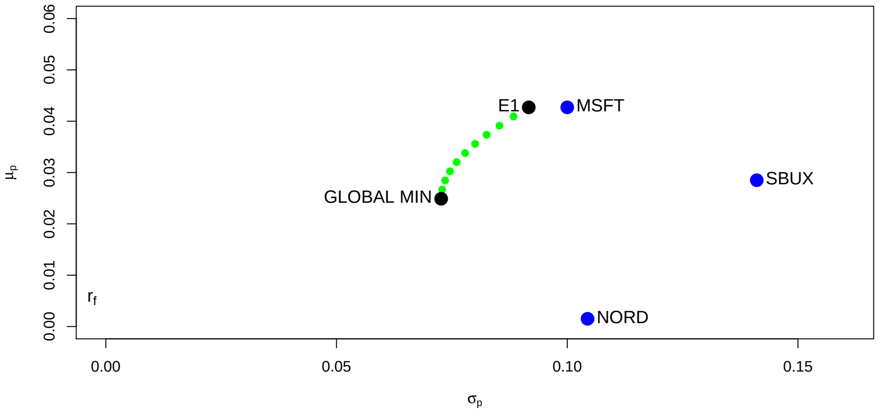
$$
\begin{aligned}
\mathbf{z} &= \alpha \cdot \mathbf{x} + (1-\alpha) \cdot \mathbf{m} \\
\mu_{p,z} &= \alpha \cdot \mu_{p,x} + (1-\alpha)\mu_{p,m} \\
\sigma_{p,z}^2 &= \alpha^2 \sigma_{p,x}^2 + (1-\alpha)^2 \sigma_{p,m}^2 + 2\alpha(1-\alpha)\sigma_{m,x} \\
\sigma_{m,x} &= \mathbf{m}'\Sigma\mathbf{x}
\end{aligned}
$$

4. Plot $\mu_{p,z}$ against $\sigma_{p,z}$. Expand or contract the grid of α values if necessary to improve the plot

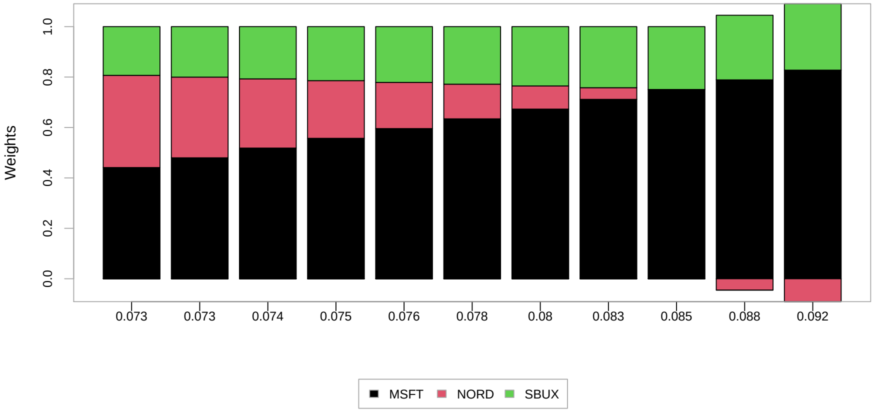
Example: R Code

```
a = seq(from=0, to=1, by=0.1)
n.a = length(a)
z.mat = matrix(0, n.a, 3)
colnames(z.mat) = names(mu.vec)
mu.z = rep(0, n.a)
sig2.z = rep(0, n.a)
sig.mx = t(m.vec)%*%sigma.mat%*%x.vec
for (i in 1:n.a) {
  z.mat[i, ] = a[i]*x.vec + (1-a[i])*m.vec
  mu.z[i] = a[i]*mu.px + (1-a[i])*mu.gmin
  sig2.z[i] = a[i]^2 * sig2.px + (1-a[i])^2 * sig2.gmin +
    2*a[i]*(1-a[i])*sig.mx
}
```

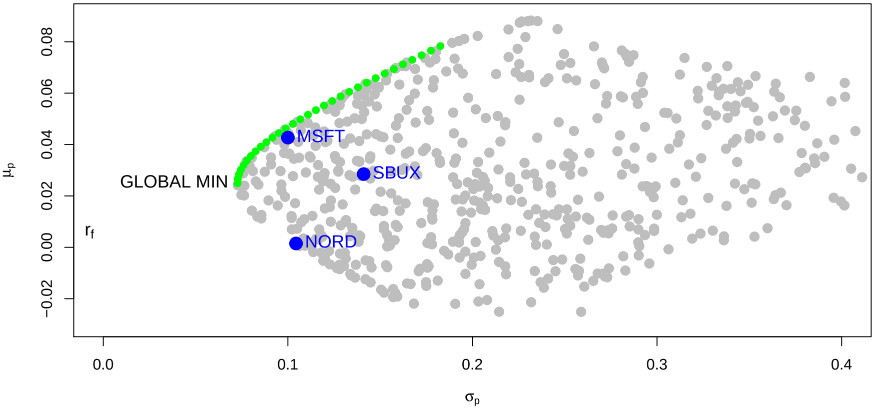
Plot the Efficient Frontier



Show the Weights of the Efficient Portfolios



Plot the Efficient Frontier with Random Portfolios



The Tangency Portfolio

The tangency portfolio $\mathbf{t}$ is the portfolio of risky assets that maximizes Sharpe's slope:

$$\max_{\mathbf{t}} \text{ Sharpe's ratio} = \frac{\mu_{p,t} - r_f}{\sigma_{p,t}}$$

subject to

$$\mathbf{t}'\mathbf{1} = 1$$

In matrix notation,

$$\text{Sharpe's ratio } = \frac{\mathbf{t}'\mu - r_f}{(\mathbf{t}'\Sigma\mathbf{t})^{1/2}}$$

Analytic solution using matrix algebra

The Lagrangian for this problem is

$$L(\mathbf{t}, \lambda) = \left(\mathbf{t}'\mu - r_f\right)\left(\mathbf{t}'\Sigma\mathbf{t}\right)^{-\frac{1}{2}} + \lambda(\mathbf{t}'\mathbf{1} - 1)$$

Using the chain rule, the first order conditions are

$$\underset{(3\times1)}{\mathbf{0}} = \frac{\partial L(\mathbf{t}, \lambda)}{\partial \mathbf{t}} = \mu(\mathbf{t}'\Sigma\mathbf{t})^{-\frac{1}{2}} - \left(\mathbf{t}'\mu - r_f\right)\left(\mathbf{t}'\Sigma\mathbf{t}\right)^{-3/2} \circ \mathbf{t} + \lambda\mathbf{1}$$

$$\underset{(1\times1)}{0} = \frac{\partial L(\mathbf{t}, \lambda)}{\partial \lambda} = \mathbf{t}'\mathbf{1} - 1 = 0$$

Analytic solution using matrix algebra

After much tedious algebra, it can be shown that the solution for $\mathbf{t}$ is

$$\mathbf{t} = \frac{\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}{\mathbf{1}'\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}$$

The Tangency Portfolio

- If the risk free rate, r_f, is less than the expected return on the global minimum variance portfolio, $\mu_{g\min}$, then the tangency portfolio has a positive Sharpe slope
- If the risk free rate, r_f, is equal to the expected return on the global minimum variance portfolio, $\mu_{g\min}$ then the tangency portfolio is not defined
- If the risk free rate, r_f, is greater than the expected return on the global minimum variance portfolio, $\mu_{g\min}$, then the tangency portfolio has a negative Sharpe slope.

The Tangency Portfolio

Example: Finding the Tangency Portfolio for 3 Asset Case

```
rf = 0.005
sigma.inv.mat = solve(sigma.mat)
one.vec = rep(1, 3)
mu.minus.rf = mu.vec - rf*one.vec
top.mat = sigma.inv.mat%*%mu.minus.rf
bot.val = as.numeric(t(one.vec)%*%top.mat)
t.vec = top.mat[,1]/bot.val
t.vec
```

```
##   MSFT   NORD   SBUX
##  1.027 -0.326  0.299
```

Example: Finding the Tangency Portfolio for 3 Asset Case

Compute mean and volatility of tangency portfolio

```
mu.t = as.numeric(crossprod(t.vec, mu.vec))
sig2.t = as.numeric(t(t.vec)%*%sigma.mat%*%t.vec)
sig.t = sqrt(sig2.t)
c(mu.t, sig.t)
```

```
## [1] 0.0519 0.1116
```

Compute Sharpe ratio of tangency portfolio

```
SR.t = (mu.t - r.f)/sig.t
SR.t
```

```
## [1] 0.42
```

Mutual Fund Separation Theorem Again

Efficient Portfolios of T-bills and Risky assets are combinations of two portfolios (mutual funds)

- T-bills
- Tangency portfolio

Efficient Portfolios

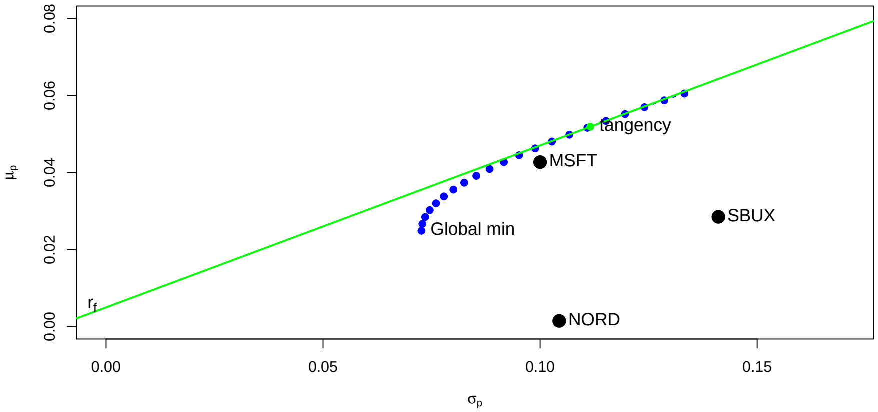
$$
\begin{aligned}
x_t &= \text{ share of wealth in tangency portfolio } \mathbf{t} \\
x_f &= \text{share of wealth in T-bills} \\
x_t + x_f &= 1 \Rightarrow \; x_f = 1 - x_t \\
\mu_p^e &= r_f + x_t(\mu_{p,t} - r_f), \; \mu_{p,t} = \mathbf{t}'\mu \\
\sigma_p^e &= x_t \sigma_{p,t}, \; \sigma_{p,t} = \left(\mathbf{t}'\Sigma\mathbf{t}\right)^{1/2}
\end{aligned}
$$

Efficient Portfolios

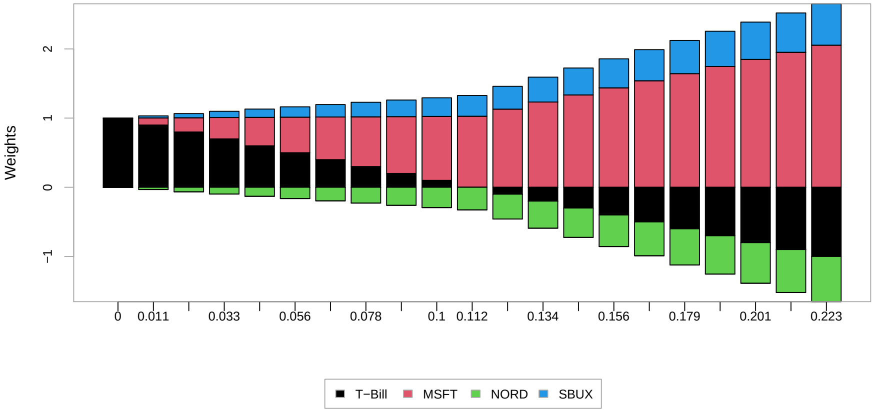
The weights x_t and x_f are determined by an investor's risk preferences

- Risk averse investors hold mostly T-Bills ($x_t \approx 0$)
- Risk tolerant investors hold mostly tangency portfolio ($x_t \approx 1$)
- If Sharpe's slope for the tangency portfolio is negative then the efficient portfolio involves shorting the tangency portfolio

Example: Efficient Portfolios



Show the efficient portfolio weights



Example: Find efficient portfolio with target risk (SD) equal to 0.02

- Recall, tangency portfolio has $\mu_t = 0.05189$ and $\sigma_t = 0.1116$. Use equation for volatility of efficient portfolio and solve for x_t

$$
\begin{aligned}
0.02 = \sigma_p^e = x_t \sigma_{p,t} = x_t \times (0.1116) \\
\Rightarrow x_t = \frac{0.02}{0.1116} = 0.1792 \\
x_f = 1 - x_t = 0.8208
\end{aligned}
$$

- Efficient portfolio with $\sigma_p^e = 0.02$ has 18% invested in tangency portfolio and 82% invested in T-Bills.

Example: Find efficient portfolio with target risk (SD) equal to 0.02

- To find expected return on efficient portfolio use

$$\begin{aligned}\mu_p^e &= r_f + x_t(\mu_{p,t} - r_f) \\ &= 0.005 + (0.1116)(0.05189 - 0.005) = 0.0134\end{aligned}$$

Example: R Code

Find the weight in the tangency portoflio:

```
x.t.02 = 0.02/sig.t
x.t.02
```

```
## [1] 0.179
```

The mean and volatility of this efficient portfolio are:

```
mu.t.02 = x.t.02*mu.t + (1-x.t.02)*r.f
sig.t.02 = x.t.02*sig.t
c(mu.t.02, sig.t.02)
```

```
## [1] 0.0134 0.0200
```

Example: Find efficient portfolio with target ER equal to 0.07

Every efficient portfolio is a combination of T-bills and the tangency portfolio. The mean of such an efficient portfolio is:

$$\mu_e = r_f + x_t \times (\mu_t - r_f)$$

Given a target mean, $\mu_0 = 0.07$, you can solve for x_t and $x_f = 1 - x_t$:

$$\begin{aligned} 0.07 = \mu_p^e &= r_f + x_t(\mu_{p,t} - r_f) \\ \Rightarrow x_t &= \frac{0.07 - r_f}{\mu_{p,t} - r_f} = \frac{0.07 - 0.005}{0.05189 - 0.005} = 1.386 \end{aligned}$$

Example: Find efficient portfolio with target ER equal to 0.07

To find the volatility of the efficient portfolio use

$$\sigma_p^e = x_t \sigma_{p,t} = (1.386)(0.1116) = 0.1547$$

Example: R Code

Find weight in tangency portfolio

```
x.t.07 = (0.07 - rf)/(mu.t - r.f)
x.t.07
```

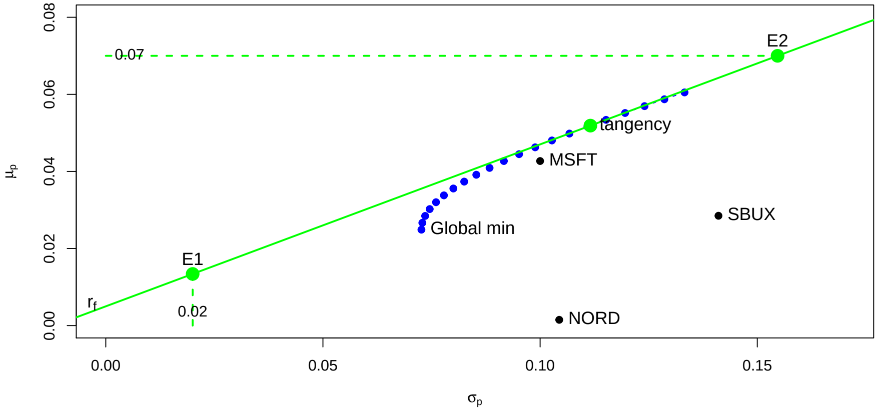
```
## [1] 1.39
```

The mean and volatility of this efficient portfolio are:

```
mu.t.07 = x.t.07*mu.t + (1-x.t.07)*r.f
sig.t.07 = x.t.07*sig.t
c(mu.t.07, sig.t.07)
```

```
## [1] 0.070 0.155
```

Efficient Portfolios with Target Mean and Target Volatility



Portfolio functions in IntroCompFinR

- The package **IntroCompFinR** contains a few R functions for computing Markowitz mean-variance efficient portfolios allowing for short sales using matrix algebra computations.
- These functions allow for the easy computation of the global minimum variance portfolio, an efficient portfolio with a given target expected return, the tangency portfolio, and the efficient frontier.

Portfolio functions in IntroCompFinR

Function	Description
`getPortfolio`	create "`portfolio`" object
`globalMin.portfolio`	compute global minimum variance portfolio
`efficient.portfolio`	compute min var portfolio subject to target return
`tangency.portfolio`	compute tangency portfolio
`efficient.frontier`	compute efficient frontier of risky assets

Functions require expected return vector, covariance matrix and optionally a risk-free rate

Example Data (Same as Before)

```
mu.vec
```

```
##   MSFT   NORD   SBUX
## 0.0427 0.0015 0.0285
```

```
sigma.mat
```

```
##        MSFT   NORD   SBUX
## MSFT 0.0100 0.0018 0.0011
## NORD 0.0018 0.0109 0.0026
## SBUX 0.0011 0.0026 0.0199
```

```
r.f
```

```
## [1] 0.005
```

getPortfolio()

Create equally weighted portfolio object:

```
library(IntroCompFinR)
ew = rep(1,3)/3
equalWeight.portfolio = getPortfolio(er=mu.vec,
                                     cov.mat=sigma.mat,
                                     weights=ew)
class(equalWeight.portfolio)
```

```
## [1] "portfolio"
```

getPortfolio()

"portfolio" objects have the following components:

```
names(equalWeight.portfolio)
```

```
## [1] "call"    "er"      "sd"      "weights"
```

Extract components using $

```
equalWeight.portfolio$weights
```

```
##  MSFT  NORD  SBUX
## 0.333 0.333 0.333
```

Method functions for "portfolio" objects

There are print(), summary() and plot() methods for "portfolio" objects. The print() method gives:

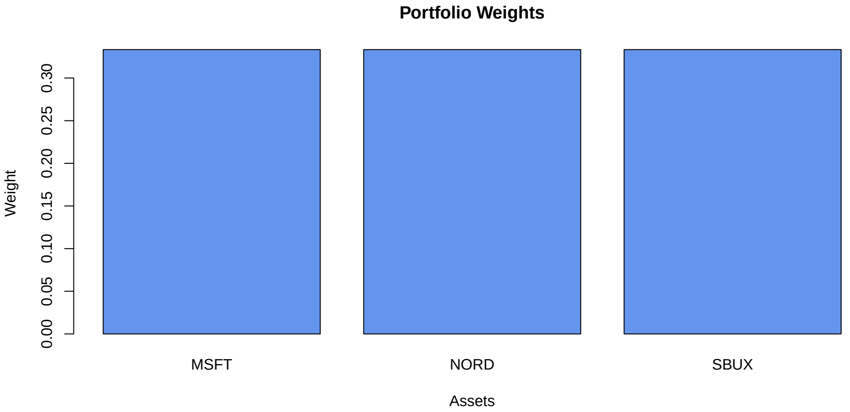
```
equalWeight.portfolio
```

```
## Call:
## getPortfolio(er = mu.vec, cov.mat = sigma.mat, weights = ew
##
## Portfolio expected return:     0.0242
## Portfolio standard deviation:  0.0759
## Portfolio weights:
##  MSFT  NORD  SBUX
## 0.333 0.333 0.333
```

Method functions for "portfolio" objects

The plot() method shows a bar chart of the portfolio weights:

```
plot(equalWeight.portfolio, col="cornflowerblue")
```



globalMin.portfolio()

To compute the global minimum variance portfolio use the function globalMin.portfolio():

```
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
```

efficient.portfolio()

Use the efficient.portfolio() function to compute a mean-variance efficient portfolio with the same mean as MSFT:

```
target.return = mu.vec[1]
e.port.msft = efficient.portfolio(mu.vec, sigma.mat, target.re
e.port.msft
```

```
## 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
```

tangent.portfolio()

To compute the tangency portfolio with $r_f = 0.005$ use the `tangency.portfolio()` function:

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

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

efficient.frontier()

The function `efficient.frontier()` constructs the set of efficient portfolios for a collection of α values on an equally spaced grid between α_{min} and α_{max}. For example, to compute 20 efficient portfolios for values of α between -2 and 1.5 use:

```
ef = efficient.frontier(mu.vec, sigma.mat, alpha.min=-2,
                        alpha.max=1.5, nport=20)
attributes(ef)
```

```
## $names
## [1] "call"    "er"      "sd"      "weights"
##
## $class
## [1] "Markowitz"
```

efficient.frontier()

Each component of ef has information for 20 frontier portfolios

```
head(cbind(ef$er, ef$sd, ef$weights), n=5)
```

```
##                              MSFT  NORD   SBUX
## port 1 -0.010724 0.133 -0.3316 1.278 0.0532
## port 2 -0.007444 0.125 -0.2604 1.194 0.0661
## port 3 -0.004164 0.117 -0.1892 1.110 0.0790
## port 4 -0.000883 0.109 -0.1181 1.026 0.0919
## port 5  0.002397 0.101 -0.0469 0.942 0.1048
```

efficient.frontier()

Use the plot() method to create a simple plot the efficient frontier:

```
plot(ef, plot.assets=TRUE, col="blue", pch=16)
```