

# Chapter 1

## Portfolio Theory with Matrix Algebra

When working with large portfolios, the algebra of representing portfolio expected returns and variances becomes cumbersome. The use of matrix (linear) algebra can greatly simplify many of the computations. Matrix algebra formulations are also very useful when it comes time to do actual computations on the computer. The matrix algebra formulas are easy to translate into matrix programming languages like R. Popular spreadsheet programs like Microsoft Excel, which are the workhorse programs of many financial houses, can also handle basic matrix calculations. All of this makes it worthwhile to become familiar with matrix techniques.

### 1.1 Portfolios with Three Risky Assets

Consider a three asset portfolio problem with assets denoted  $A$ ,  $B$  and  $C$ . Let  $R_i$  ( $i = A, B, C$ ) denote the return on asset  $i$  and assume that the constant expected return (CER) model holds:

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

**Example 1** *Three asset example data*

Table 1.1 gives example data on monthly means, variances and covariances for the continuously compounded returns on Microsoft, Norstrom and Starbucks

## 2CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

Stock $i$	$\mu_i$	$\sigma_i$	Pari (i,j)	$\sigma_{ij}$
A	0.0427	0.1000	(A,B)	0.0018
B	0.0015	0.1044	(A,C)	0.0011
C	0.0285	0.1411	(B,C)	0.0026

Table 1.1: Three asset example data.

(assets A, B and C) based on sample statistics computed over the five-year period January, 1995 through January, 2000<sup>1</sup>. The values of  $\mu_i$  and  $\sigma_i$  (risk-return tradeoffs) are shown in Figure 1.1. Clearly, Microsoft provides the best risk-return tradeoff and Nordstrom provides with worst. ■

Let  $x_i$  denote the share of wealth invested in asset  $i$  ( $i = A, B, C$ ), and assume that all wealth is invested in the three assets so that  $x_A + x_B + x_C = 1$ . The portfolio return,  $R_p$ , is the random variable

$$R_{p,x} = x_A R_A + x_B R_B + x_C R_C. \quad (1.1)$$

The subscript “ $x$ ” indicates that the portfolio is constructed using the  $x$ -weights  $x_A, x_B$  and  $x_C$ . The expected return on the portfolio is

$$\mu_{p,x} = E[R_{p,x}] = x_A \mu_A + x_B \mu_B + x_C \mu_C, \quad (1.2)$$

and the variance of the portfolio return is

$$\sigma_{p,x}^2 = \text{var}(R_{p,x}) = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + x_C^2 \sigma_C^2 + 2x_A x_B \sigma_{AB} + 2x_A x_C \sigma_{AC} + 2x_B x_C \sigma_{BC}. \quad (1.3)$$

Notice that variance of the portfolio return depends on three variance terms and six covariance terms. Hence, with three assets there are twice as many covariance terms than variance terms contributing to portfolio variance. Even with three assets, the algebra representing the portfolio characteristics (1.1) - (1.3) is cumbersome. We can greatly simplify the portfolio algebra using matrix notation.

<sup>1</sup>This example data is also analyzed in the Excel spreadsheet 3firmExample.xls.

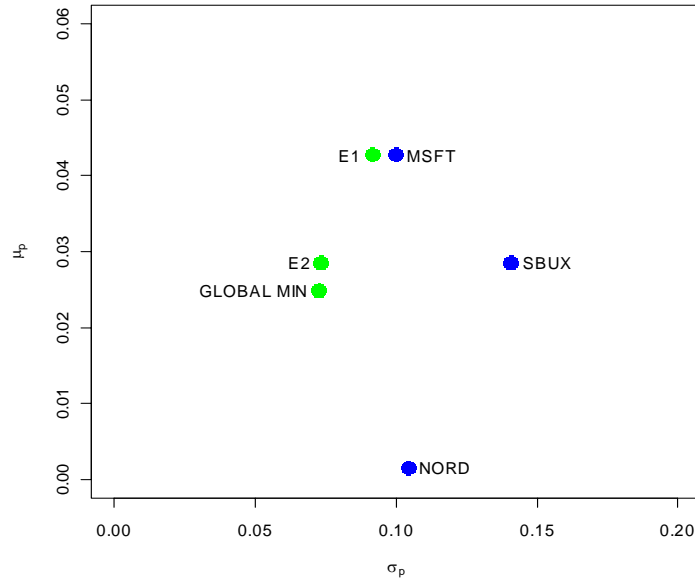


Figure 1.1: Risk-return tradeoffs among three asset portfolios. The portfolio labeled “e1” is the efficient portfolio with the same expected return as Microsoft; the portfolio labeled “e2” is the efficient portfolio with the same expected return as Starbux.

### 1.1.1 Portfolio Characteristics Using Matrix Notation

Define the following  $3 \times 1$  column vectors containing the asset returns and portfolio weights

$$\mathbf{R} = \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix}.$$

In matrix notation we can lump multiple returns in a single vector which we denote by  $\mathbf{R}$ . Since each of the elements in  $\mathbf{R}$  is a random variable we call  $\mathbf{R}$  a *random vector*. The probability distribution of the random vector  $\mathbf{R}$  is simply the joint distribution of the elements of  $\mathbf{R}$ . In the CER model all returns are jointly normally distributed and this joint distribution is com-

## 4CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

pletely characterized by the means, variances and covariances of the returns. We can easily express these values using matrix notation as follows. The  $3 \times 1$  vector of portfolio expected values is

$$E[\mathbf{R}] = E \left[ \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix} \right] = \begin{pmatrix} E[R_A] \\ E[R_B] \\ E[R_C] \end{pmatrix} = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix} = \boldsymbol{\mu},$$

and the  $3 \times 3$  covariance matrix of returns is

$$\begin{aligned} \text{var}(\mathbf{R}) &= \begin{pmatrix} \text{var}(R_A) & \text{cov}(R_A, R_B) & \text{cov}(R_A, R_C) \\ \text{cov}(R_B, R_A) & \text{var}(R_B) & \text{cov}(R_B, R_C) \\ \text{cov}(R_C, R_A) & \text{cov}(R_C, R_B) & \text{var}(R_C) \end{pmatrix} \\ &= \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} = \boldsymbol{\Sigma}. \end{aligned}$$

Notice that the covariance matrix is symmetric (elements off the diagonal are equal so that  $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}'$ , where  $\boldsymbol{\Sigma}'$  denotes the transpose of  $\boldsymbol{\Sigma}$ ) since  $\text{cov}(R_A, R_B) = \text{cov}(R_B, R_A)$ ,  $\text{cov}(R_A, R_C) = \text{cov}(R_C, R_A)$  and  $\text{cov}(R_B, R_C) = \text{cov}(R_C, R_B)$ .

**Example 2** *Example return data using matrix notation*

Using the example data in Table 1.1 we have

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix} = \begin{pmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{pmatrix},$$
$$\boldsymbol{\Sigma} = \begin{pmatrix} 0.0100 & 0.0018 & 0.0011 \\ 0.0018 & 0.0109 & 0.0026 \\ 0.0011 & 0.0026 & 0.0199 \end{pmatrix}.$$

In R, these values are created using

```

> 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)
> 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

```



The return on the portfolio using vector notation is

$$R_{p,x} = \mathbf{x}'\mathbf{R} = (x_A, x_B, x_C) \cdot \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix} = x_A R_A + x_B R_B + x_C R_C.$$

Similarly, the expected return on the portfolio is

$$\mu_{p,x} = E[\mathbf{x}'\mathbf{R}] = \mathbf{x}'E[\mathbf{R}] = \mathbf{x}'\boldsymbol{\mu} = (x_A, x_B, x_C) \cdot \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix} = x_A \mu_A + x_B \mu_B + x_C \mu_C.$$

The variance of the portfolio is

$$\begin{aligned} \sigma_{p,x}^2 &= \text{var}(\mathbf{x}'\mathbf{R}) = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} = (x_A, x_B, x_C) \cdot \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix} \\ &= x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + x_C^2 \sigma_C^2 + 2x_A x_B \sigma_{AB} + 2x_A x_C \sigma_{AC} + 2x_B x_C \sigma_{BC}. \end{aligned}$$

## 6 CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

Finally, the condition that the portfolio weights sum to one can be expressed as

$$\mathbf{x}'\mathbf{1} = (x_A, x_B, x_C) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = x_A + x_B + x_C = 1,$$

where  $\mathbf{1}$  is a  $3 \times 1$  vector with each element equal to 1.

Consider another portfolio with weights  $\mathbf{y} = (y_A, y_B, y_C)'$ . The return on this portfolio is

$$R_{p,y} = \mathbf{y}'\mathbf{R} = y_A R_A + y_B R_B + y_C R_C.$$

In the following we will need to compute the covariance between the return on portfolio  $\mathbf{x}$  and the return on portfolio  $\mathbf{y}$ ,  $\text{cov}(R_{p,x}, R_{p,y})$ . It can be shown that

$$\begin{aligned} \sigma_{xy} &= \text{cov}(R_{p,x}, R_{p,y}) = \text{cov}(\mathbf{x}'\mathbf{R}, \mathbf{y}'\mathbf{R}) \\ &= \mathbf{x}'\Sigma\mathbf{y} = (x_A, x_B, x_C) \cdot \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} \begin{pmatrix} y_A \\ y_B \\ y_C \end{pmatrix} \\ &= x_A y_A \sigma_A^2 + x_B y_B \sigma_B^2 + x_C y_C \sigma_C^2 \\ &\quad + (x_A y_B + x_B y_A) \sigma_{AB} + (x_A y_C + x_C y_A) \sigma_{AC} + (x_B y_C + x_C y_B) \sigma_{BC}. \end{aligned}$$

### Example 3 Portfolio computations in R

Consider an equally weighted portfolio with  $x_A = x_B = x_C = 1/3$ . This portfolio has return  $R_{p,x} = \mathbf{x}'\mathbf{R}$  where  $\mathbf{x} = (1/3, 1/3, 1/3)'$ . Using R, the portfolio mean and variance are

```
> 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)
> mu.p.x
[ ,1]
```

```
[1,] 0.02423
> sig.p.x
      [,1]
[1,] 0.07587
```

Next, consider another portfolio with weight vector  $\mathbf{y} = (y_A, y_B, y_C)' = (0.8, 0.4, -0.2)'$  and return  $R_{p,y} = \mathbf{y}'\mathbf{R}$ . The covariance between  $R_{p,x}$  and  $R_{p,y}$  is

```
> y.vec = c(0.8, 0.4, -0.2)
> names(x.vec) = asset.names
> sig.xy = t(x.vec)%*%sigma.mat%*%y.vec
> sig.xy
      [,1]
[1,] 0.003907
```

■

### 1.1.2 Finding the Global Minimum Variance Portfolio

The global minimum variance portfolio  $\mathbf{m} = (m_A, m_B, m_C)'$  for the three asset case solves the constrained minimization problem

$$\begin{aligned} \min_{m_A, m_B, m_C} \quad & \sigma_{p,m}^2 = m_A^2 \sigma_A^2 + m_B^2 \sigma_B^2 + m_C^2 \sigma_C^2 \\ & + 2m_A m_B \sigma_{AB} + 2m_A m_C \sigma_{AC} + 2m_B m_C \sigma_{BC} \\ \text{s.t.} \quad & m_A + m_B + m_C = 1. \end{aligned} \quad (1.4)$$

The Lagrangian for this problem is

$$\begin{aligned} L(m_A, m_B, m_C, \lambda) = & m_A^2 \sigma_A^2 + m_B^2 \sigma_B^2 + m_C^2 \sigma_C^2 + 2m_A m_B \sigma_{AB} + 2m_A m_C \sigma_{AC} + 2m_B m_C \sigma_{BC} \\ & + \lambda(m_A + m_B + m_C - 1), \end{aligned}$$

and the first order conditions for a minimum are

$$\begin{aligned} 0 &= \frac{\partial L}{\partial m_A} = 2m_A \sigma_A^2 + 2m_B \sigma_{AB} + 2m_C \sigma_{AC} + \lambda, \\ 0 &= \frac{\partial L}{\partial m_B} = 2m_B \sigma_B^2 + 2m_A \sigma_{AB} + 2m_C \sigma_{BC} + \lambda, \\ 0 &= \frac{\partial L}{\partial m_C} = 2m_C \sigma_C^2 + 2m_A \sigma_{AC} + 2m_B \sigma_{BC} + \lambda, \\ 0 &= \frac{\partial L}{\partial \lambda} = m_A + m_B + m_C - 1. \end{aligned} \quad (1.5)$$

## 8CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

This gives four linear equations in four unknowns which can be solved to find the global minimum variance portfolio weights  $m_A, m_B$  and  $m_C$ .

Using matrix notation, the problem (1.4) can be concisely expressed as

$$\min_{\mathbf{m}} \sigma_{p,m}^2 = \mathbf{m}'\Sigma\mathbf{m} \text{ s.t. } \mathbf{m}'\mathbf{1} = 1. \quad (1.6)$$

The four linear equation describing the first order conditions (1.5) has the matrix representation

$$\begin{pmatrix} 2\sigma_A^2 & 2\sigma_{AB} & 2\sigma_{AC} & 1 \\ 2\sigma_{AB} & 2\sigma_B^2 & 2\sigma_{BC} & 1 \\ 2\sigma_{AC} & 2\sigma_{BC} & 2\sigma_C^2 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} m_A \\ m_B \\ m_C \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

or, more concisely,

$$\begin{pmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}' & 0 \end{pmatrix} \begin{pmatrix} \mathbf{m} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}. \quad (1.7)$$

The system (1.7) is of the form

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

where

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

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

$$\mathbf{z}_m = \mathbf{A}_m^{-1} \mathbf{b}. \quad (1.8)$$

The first three elements of  $\mathbf{z}_m$  are the portfolio weights  $\mathbf{m} = (m_A, m_B, m_C)'$  for the global minimum variance portfolio with expected return  $\mu_{p,m} = \mathbf{m}'\boldsymbol{\mu}$  and variance  $\sigma_{p,m}^2 = \mathbf{m}'\Sigma\mathbf{m}$ .

**Example 4** *Global minimum variance portfolio for example data*

Using the data in Table 1, we can use R to compute the global minimum variance portfolio weights from (1.8) as follows:



```

> top = cbind(2*sigma.mat, rep(1, 3))
> bot = c(rep(1, 3), 0)
> Am.mat = rbind(top, bot)
> b.vec = c(rep(0, 3), 1)
> z.m = solve(Am.mat)%*%b.vec
> m = z.m[1:3]
> names(m) = asset.names
> m
  MSFT  NORD  SBUX
0.4411 0.3656 0.1933

```

Hence, the global minimum variance portfolio has portfolio weights  $m_{msft} = 0.4411$ ,  $m_{nord} = 0.3656$  and  $m_{sbux} = 0.1933$ , and is given by the vector

$$\mathbf{m} = (0.4411, 0.3656, 0.1933)'. \quad (1.9)$$

The expected return on this portfolio,  $\mu_{p,m} = \mathbf{m}'\boldsymbol{\mu}$ , is

```

> mu.gmin = crossprod(m, mu.vec)
> mu.gmin
  [,1]
[1,] 0.02489

```

The portfolio variance,  $\sigma_{p,m}^2 = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{m}$ , and standard deviation,  $\sigma_{p,m}$ , are

```

> sig2.gmin = t(m)%*%sigma.mat%*%m
> sig.gmin = sqrt(sig2.gmin)
> sig2.gmin
  [,1]
[1,] 0.005282
> sig.gmin
  [,1]
[1,] 0.07268

```

In figure 1.1, this portfolio is labeled “global min”. ■

**Alternative derivation of global minimum variance portfolio**

The first order conditions (1.5) from the optimization problem (1.6) can be expressed in matrix notation as

$$\underset{(3 \times 1)}{\mathbf{0}} = \frac{\partial L(\mathbf{m}, \lambda)}{\partial \mathbf{m}} = 2 \cdot \Sigma \mathbf{m} + \lambda \cdot \mathbf{1} \quad (1.10)$$

$$\underset{(1 \times 1)}{0} = \frac{\partial L(\mathbf{m}, \lambda)}{\partial \lambda} = \mathbf{m}' \mathbf{1} - 1 \quad (1.11)$$

Using (1.10), first solve for  $\mathbf{m}$  :

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

Next, multiply both sides by  $\mathbf{1}'$  and use (1.11) to solve for  $\lambda$ :

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

Finally, substitute the value for  $\lambda$  back into (1.10) to solve for  $\mathbf{m}$ :

$$\mathbf{m} = -\frac{1}{2} (-2) \frac{1}{\mathbf{1}' \Sigma^{-1} \mathbf{1}} \Sigma^{-1} \mathbf{1} = \frac{\Sigma^{-1} \mathbf{1}}{\mathbf{1}' \Sigma^{-1} \mathbf{1}}. \quad (1.12)$$

**Example 5** *Finding global minimum variance portfolio for example data*

Using the data in Table 1, we can use R to compute the global minimum variance portfolio weights from (1.12) as follows:

```
> one.vec = rep(1, 3)
> top = solve(sigma.mat)%*%one.vec
> bot = as.numeric((t(one.vec)%*%sigma.mat.inv)%*%one.vec)
> m = top/bot
> m
      [,1]
MSFT 0.4411
NORD 0.3656
SBUX 0.1933
```

■

### 1.1.3 Finding Efficient Portfolios

The investment opportunity set is the set of portfolio expected return and portfolio standard deviation values for all possible portfolios such that  $x_A + x_B + x_C = 1$ . As in the two risky asset case, this set can be described in a graph with  $\mu_p$  on the vertical axis and  $\sigma_p$  on the horizontal axis. With two assets, the investment opportunity set in  $(\mu_p, \sigma_p)$  space lies on a line (one side of a hyperbola). With three asset, the investment opportunity set in  $(\mu_p, \sigma_p)$  space is described by an infinite set of values whose general shape is complicated and depends crucially on the covariance terms  $\sigma_{ij}$ . However, we do not have to fully characterize the investment opportunity set. If we assume that investors choose portfolios to maximize expected return subject to a target level of risk, or, equivalently, to minimize risk subject to a target expected return, then we can simplify the asset allocation problem by only concentrating on the set of efficient portfolios consisting of assets A, B and C. This is the framework originally developed by Harry Markowitz, the father of portfolio theory and winner of the Nobel Prize in economics.

We assume that investors wish to find portfolios that have the best expected return-risk trade-off. Harry Markowitz characterized these efficient portfolios in two equivalent ways. In the first way, investors seek to find portfolios that maximize portfolio expected return for a given level of risk as measured by portfolio variance. Let  $\sigma_{p,0}^2$  denote a target level of risk. Then Harry Markowitz characterized the constrained maximization problem to find an efficient portfolio as

$$\begin{aligned} \max_{\mathbf{x}} \mu_p &= \mathbf{x}'\boldsymbol{\mu} \text{ s.t.} \\ \sigma_p^2 &= \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} = \sigma_{p,0}^2 \text{ and } \mathbf{x}'\mathbf{1} = 1. \end{aligned} \quad (1.13)$$

Markowitz showed that the investor's problem of maximizing portfolio expected return subject to a target level of risk has an equivalent dual representation in which the investor minimizes the risk of the portfolio (as measured by portfolio variance) subject to a target expected return level. Let  $\mu_{p,0}$  denote a target expected return level. Then the dual problem is the constrained minimization problem

$$\begin{aligned} \min_{\mathbf{x}} \sigma_{p,x}^2 &= \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} \text{ s.t.} \\ \mu_p &= \mathbf{x}'\boldsymbol{\mu} = \mu_{p,0}, \text{ and } \mathbf{x}'\mathbf{1} = 1, \end{aligned} \quad (1.14)$$

To find efficient portfolios of risky assets in practice, the dual problem (1.14) is most often solved. This is partially due to computational conveniences

## 12 CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

and partly due to investors being more willing to specify target expected returns rather than target risk levels. The efficient portfolio frontier is graph of  $\mu_p$  versus  $\sigma_p$  for the set of efficient portfolios generated by solving (1.14) for all possible target expected return levels  $\mu_{p,0}$  above the expected return on the global minimum variance portfolio. Just as in the two asset case, the resulting efficient frontier will resemble one side of an hyperbola and is often called the “Markowitz bullet”.

The Lagrangian function associated with (1.14) is

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

The first order conditions are

$$\frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \mathbf{x}} = 2\Sigma\mathbf{x} + \lambda_1\boldsymbol{\mu} + \lambda_2\mathbf{1} = \mathbf{0}, \quad (1.15)$$

$$\frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \lambda_1} = \mathbf{x}'\boldsymbol{\mu} - \mu_{p,0} = 0, \quad (1.16)$$

$$\frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \lambda_2} = \mathbf{x}'\mathbf{1} - 1 = 0. \quad (1.17)$$

These first order conditions consist of five linear equations in five unknowns ( $x_A, x_B, x_C, \lambda_1, \lambda_2$ ). We can represent the system of linear equations as

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

or

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

where

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

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

$$\mathbf{z}_x = \mathbf{A}_x^{-1}\mathbf{b}_0. \quad (1.18)$$

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

**Example 6** *Efficient portfolio with the same expected return as Microsoft*

Using the data in Table 1, consider finding an efficient portfolio with the same expected return as Microsoft. That is, consider solving (1.14) with target expected return  $\mu_{p,0} = \mu_{msft} = 0.0427$  using (1.18). The R calculations are

```
> top = cbind(2*sigma.mat, mu.vec, rep(1, 3))
> mid = c(mu.vec, 0, 0)
> bot = c(rep(1, 3), 0, 0)
> Ax.mat = rbind(top, mid, bot)
> bmsft.vec = c(rep(0, 3), mu.vec["MSFT"], 1)
> z.vec = solve(Ax.mat)%*%b.vec
> x.vec = z.vec[1:3]
> names(x.vec) = asset.names
> x.vec
      MSFT      NORD      SBUX
0.82745 -0.09075  0.26329
```

Hence, the efficient portfolio with the same expected return as Microsoft has portfolio weights  $x_{msft} = 0.82745$ ,  $x_{nord} = -0.09075$  and  $x_{sbux} = 0.26329$ , and is given by the vector

$$\mathbf{x} = (0.82745, -0.09075, 0.26329)'. \quad (1.19)$$

The expected return on this portfolio,  $\mu_{p,x} = \mathbf{x}'\boldsymbol{\mu}$ , is equal to the target return  $\mu_{msft}$ :

```
> mu.px = crossprod(x.vec, mu.vec)
> mu.px
      [,1]
[1,] 0.0427
```

The portfolio variance,  $\sigma_{p,x}^2 = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x}$ , and standard deviation,  $\sigma_{p,x}$ , are

```
> sig2.px = t(x.vec)%*%sigma.mat%*%x.vec
> sig.px = sqrt(sig2.px)
> sig2.px
      [,1]
[1,] 0.0084
> sig.px
      [,1]
[1,] 0.09166
```

## 14 CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

and are smaller than the corresponding values for Microsoft (see Table 1). This efficient portfolio is labeled “e1” in Figure 1.1.

Next, consider finding an efficient portfolio with the same expected return as Starbucks:

```
> bsbux.vec = c(rep(0, 3), mu.vec["SBUX"], 1)
> z.vec = solve(Ax.mat)%*%bsbux.vec
> y.vec = z.vec[1:3]
> names(y.vec) = asset.names
> y.vec
  MSFT  NORD  SBUX
0.5194 0.2732 0.2075
```

This efficient portfolio has weights  $y_{msft} = 0.5194$ ,  $y_{nord} = 0.2732$  and  $y_{sbux} = 0.2075$  and is given by the vector

$$\mathbf{y} = (0.5194, 0.2732, 0.2075)'. \quad (1.20)$$

The portfolio expected return, variance and standard deviation are:

```
> mu.py = crossprod(y.vec, mu.vec)
> sig2.py = t(y.vec)%*%sigma.mat%*%y.vec
> sig.py = sqrt(sig2.py)
> mu.py
  [,1]
[1,] 0.0285
> sig.py
  [,1]
[1,] 0.07355
```

This efficient portfolio is labeled “e2” in Figure 1.1. ■

### Alternative derivation of efficient portfolio

Consider the first order conditions (1.15)-(1.17) from the optimization problem (1.14). First, use (1.15) to solve for  $\mathbf{x}$ :

$$\mathbf{x} = \frac{-1}{2}\lambda_1\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{-1}{2}\lambda_1\boldsymbol{\Sigma}^{-1}\mathbf{1}. \quad (1.21)$$

Next, to find the values for  $\lambda_1$  and  $\lambda_2$ , pre-multiply (1.21) by  $\boldsymbol{\mu}'$  and use (1.16) to give

$$\mu_0 = \boldsymbol{\mu}'\mathbf{x} = \frac{-1}{2}\lambda_1\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{-1}{2}\lambda_1\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1}.$$

Similarly, pre-multiply (1.21) by  $\mathbf{1}'$  and use (1.17) to give

$$1 = \mathbf{1}'\mathbf{x} = \frac{-1}{2}\lambda_1\mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{-1}{2}\lambda_1\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}.$$

Now, we have two linear equations involving  $\lambda_1$  and  $\lambda_2$  which we can write in matrix notation as

$$\frac{-1}{2} \begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \mu_0 \\ 1 \end{pmatrix},$$

where  $A = \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ ,  $B = \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1}$  and  $C = \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}$ . The solution for  $(\lambda_1, \lambda_2)'$  is

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = -2 \begin{pmatrix} A & B \\ B & C \end{pmatrix}^{-1} \begin{pmatrix} \mu_0 \\ 1 \end{pmatrix}.$$

Using

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix}^{-1} = \frac{1}{AC - B^2} \begin{pmatrix} C & -B \\ -B & A \end{pmatrix}$$

we have

$$\lambda_1 = \frac{-2}{AC - B^2} (C\mu_0 - B), \quad \lambda_2 = \frac{-2}{AC - B^2} (A - B\mu_0)$$

Substituting these values back into (1.21) gives an explicit expression for the efficient portfolio weight vector  $\mathbf{x}$ :

$$\mathbf{x} = \frac{C\mu_0 - B}{AC - B^2}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{A - B\mu_0}{AC - B^2}\boldsymbol{\Sigma}^{-1}\mathbf{1}.$$

### 1.1.4 Computing the Efficient Frontier

The expression for an efficient portfolio derived in the previous sub-section showed that any efficient portfolio can be created as a convex combination of any two efficient portfolios with different target expected returns. If the expected return on the resulting portfolio is greater than the expected on the global minimum variance portfolio, then the portfolio is an efficient frontier portfolio. Otherwise, the portfolio is an inefficient frontier portfolio. As a result, to compute the portfolio frontier in  $(\mu_p, \sigma_p)$  space (Markowitz bullet) we only need to find two efficient portfolios. The remaining frontier portfolios can then be expressed as convex combinations of these two portfolios. The following proposition describes the process for the three risky asset case using matrix algebra.

**Proposition 7** *Creating a frontier portfolio from two efficient portfolios*

Let  $\mathbf{x} = (x_A, x_B, x_C)'$  and  $\mathbf{y} = (y_A, y_B, y_C)'$  be any two efficient portfolios with different target expected returns. That is,  $\mathbf{x}$  solves

$$\min_{\mathbf{x}} \sigma_{p,x}^2 = \mathbf{x}'\Sigma\mathbf{x} \quad \text{s.t.} \quad \mathbf{x}'\boldsymbol{\mu} = \mu_{p,0} \quad \text{and} \quad \mathbf{x}'\mathbf{1} = 1,$$

and  $\mathbf{y}$  solves

$$\min_{\mathbf{y}} \sigma_{p,y}^2 = \mathbf{y}'\Sigma\mathbf{y} \quad \text{s.t.} \quad \mathbf{y}'\boldsymbol{\mu} = \mu_{p,1} \quad \text{and} \quad \mathbf{y}'\mathbf{1} = 1,$$

where  $\mu_{p,0} \neq \mu_{p,1}$ . Let  $\alpha$  be any constant. Then the portfolio

$$\begin{aligned} \mathbf{z} &= \alpha \cdot \mathbf{x} + (1 - \alpha) \cdot \mathbf{y} \\ &= \begin{pmatrix} \alpha x_A + (1 - \alpha)y_A \\ \alpha x_B + (1 - \alpha)y_B \\ \alpha x_C + (1 - \alpha)y_C \end{pmatrix} \end{aligned} \tag{1.22}$$

is a frontier portfolio with

$$\mu_{p,z} = \mathbf{z}'\boldsymbol{\mu} = \alpha \cdot \mu_{p,x} + (1 - \alpha) \cdot \mu_{p,y}, \tag{1.23}$$

$$\sigma_{p,z}^2 = \mathbf{z}'\Sigma\mathbf{z} = \alpha^2 \sigma_{p,x}^2 + (1 - \alpha)^2 \sigma_{p,y}^2 + 2\alpha(1 - \alpha)\sigma_{xy}, \tag{1.24}$$

where

$$\sigma_{p,x}^2 = \mathbf{x}'\Sigma\mathbf{x}, \sigma_{p,y}^2 = \mathbf{y}'\Sigma\mathbf{y}, \sigma_{xy} = \mathbf{x}'\Sigma\mathbf{y}.$$

If  $\mu_{p,z} \geq \mu_{p,m}$ , where  $\mu_{p,m}$  is the expected return on the global minimum variance portfolio, then  $\mathbf{z}$  is an efficient portfolio.



**Example 8** *Creating an arbitrary frontier portfolio from two efficient portfolios*

Consider the data in Table 1 and the previously computed efficient portfolios (1.19) and (1.20) and let  $\alpha = 0.5$ . From (1.22), the frontier portfolio  $\mathbf{z}$  is constructed using

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

In R, the new frontier portfolio is computed using

```
> a = 0.5
> z.vec = a*x.vec + (1-a)*y.vec
> z.vec
  MSFT  NORD  SBUX
0.6734 0.0912 0.2354
```

Using  $\mu_{p,z} = \mathbf{z}'\boldsymbol{\mu}$  and  $\sigma_{p,z}^2 = \mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}$ , the expected return, variance and standard deviation of this portfolio are

```
> mu.pz = crossprod(z.vec, mu.vec)
> sig2.pz = t(z.vec)%*%sigma.mat%*%z.vec
> sig.pz = sqrt(sig2.pz)
> mu.pz
[ ,1]
```

## 18CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

```
[1,] 0.0356
> sig2.pz
      [,1]
[1,] 0.00641
> sig.pz
      [,1]
[1,] 0.08006
```

Equivalently, using  $\mu_{p,z} = \alpha\mu_{p,x} + (1-\alpha)\mu_{p,y}$  and  $\sigma_{p,z}^2 = \alpha^2\sigma_{p,x}^2 + (1-\alpha)^2\sigma_{p,y}^2 + 2\alpha(1-\alpha)\sigma_{xy}$  the expected return, variance and standard deviation of this portfolio are

```
> mu.pz = a*mu.px + (1-a)*mu.py
> sig.xy = t(x.vec)%*%sigma.mat%*%y.vec
> sig2.pz = a^2 * sig2.px + (1-a)^2 * sig2.py + 2*a*(1-a)*sig.xy
> sig.pz = sqrt(sig2.pz)
> mu.pz
      [,1]
[1,] 0.0356
> sig2.pz
      [,1]
[1,] 0.00641
> sig.pz
      [,1]
[1,] 0.08006
```

Because  $\mu_{p,z} = 0.0356 > \mu_{p,m} = 0.02489$  the frontier portfolio  $\mathbf{z}$  is an efficient portfolio. ■

**Example 9** *Creating a frontier portfolio with a given expected return from two efficient portfolios*

Given the two efficient portfolios (1.19) and (1.20) with target expected returns equal to the expected returns on Microsoft and Starbucks, respectively, consider creating a frontier portfolio with target expected return equal to the expected return on Nordstrom. Then

$$\mu_{p,z} = \alpha\mu_{p,x} + (1-\alpha)\mu_{p,y} = \mu_{nord} = 0.0015,$$

and we can solve for  $\alpha$  using

$$\alpha = \frac{\mu_{nord} - \mu_{p,y}}{\mu_{p,x} - \mu_{p,y}} = \frac{0.0015 - 0.0285}{0.0427 - 0.0285} = -1.901.$$

Using R, the weights in this frontier portfolio are

```
> z.nord = a.nord*x.vec + (1 - a.nord)*y.vec
> names(z.nord) = asset.names
> z.nord
      MSFT      NORD      SBUX
-0.06637  0.96509  0.10128
```

The expected return, variance and standard deviation on this portfolio are

```
> mu.pz.nord = a.nord*mu.px + (1-a.nord)*mu.py
> sig2.pz.nord = a.nord^2 * sig2.px + (1-a.nord)^2 * sig2.py +
+               2*a.nord*(1-a.nord)*sig.xy
> sig.pz.nord = sqrt(sig2.pz.nord)
> mu.pz.nord
      [,1]
[1,] 0.0015
> sig2.pz.nord
      [,1]
[1,] 0.01066
> sig.pz.nord
      [,1]
[1,] 0.1033
```

Because  $\mu_{p,z} = 0.0015 < \mu_{p,m} = 0.02489$  the frontier portfolio  $\mathbf{z}$  is an inefficient frontier portfolio. ■

The efficient frontier of portfolios, i.e., those frontier portfolios with expected return greater than the expected return on the global minimum variance portfolio, can be conveniently created using (1.22) with two specific efficient portfolios. The first efficient portfolio is the global minimum variance portfolio (1.4). The second efficient portfolio is the efficient portfolio whose target expected return is equal to the highest expected return among all of the assets under consideration. The steps for constructing the efficient frontier is then:

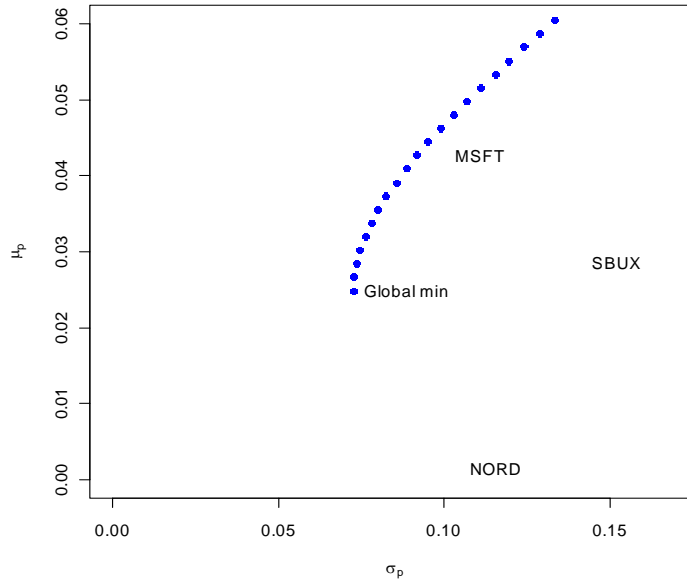


Figure 1.2: Efficient frontier of three risky assets.

1. Compute the global minimum variance portfolio  $\mathbf{m}$  by solving (1.6) and compute  $\mu_{p,m} = \mathbf{m}'\boldsymbol{\mu}$  and variance  $\sigma_{p,m}^2 = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{m}$ .
2. Compute the efficient portfolio  $\mathbf{x}$  by with target expected return equal to the maximum expected return of the assets under consideration. That is, solve (1.14) with  $\mu_0 = \max\{\mu_1, \mu_2, \mu_3\}$ , and compute  $\mu_{p,x} = \mathbf{x}'\boldsymbol{\mu}$  and variance  $\sigma_{p,m}^2 = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x}$ .
3. Compute  $\text{cov}(R_{p,m}, R_{p,x}) = \sigma_{mx} = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{x}$ .
4. Create an initial grid of  $\alpha$  values  $\{1, 0.9, \dots, -0.9, -1\}$ , compute the frontier portfolios  $\mathbf{z}$  and their expected returns and variances using (1.22), (1.23) and (1.24), respectively.
5. Plot  $\mu_{p,z}$  against  $\sigma_{p,z}$  and adjust the grid of  $\alpha$  values to create a nice plot.

**Example 10** *Compute and plot the efficient frontier of risky assets (Markowitz bullet)*

To compute and plot the efficient frontier from the three risky assets in Table 1.1 in R use

```

> a = seq(from=1, to=-1, by=-0.1)
> n.a = length(a)
> z.mat = matrix(0, n.a, 3)
> mu.z = rep(0, n.a)
> sig2.z = rep(0, n.a)
> sig.mx = t(m)%*%sigma.mat%*%x.vec
> for (i in 1:n.a) {
+   z.mat[i, ] = a[i]*m + (1-a[i])*x.vec
+   mu.z[i] = a[i]*mu.gmin + (1-a[i])*mu.px
+   sig2.z[i] = a[i]^2 * sig2.gmin + (1-a[i])^2 * sig2.px +
+             2*a[i]*(1-a[i])*sig.mx
+ }
> plot(sqrt(sig2.z), mu.z, type="b", ylim=c(0, 0.06), xlim=c(0, 0.17),
+       pch=16, col="blue", ylab=expression(mu[p]),
+       xlab=expression(sigma[p]))
> text(sig.gmin, mu.gmin, labels="Global min", pos=4)
> text(sd.vec, mu.vec, labels=asset.names, pos=4)

```

The variables `z.mat`, `mu.z` and `sig2.z` contain the weights, expected returns and variances, respectively, of the efficient frontier portfolios for a grid of  $\alpha$  values between 1 and  $-1$ . The resulting efficient frontier is illustrated in Figure 1.18. ■

### 1.1.5 Computing the Tangency Portfolio

The tangency portfolio is the portfolio of risky assets that has the highest Sharpe's slope. The tangency portfolio,  $\mathbf{t} = (t_{msft}, t_{nord}, t_{sbux})'$ , solves

$$\max_{\mathbf{t}} \frac{\mathbf{t}'\boldsymbol{\mu} - r_f}{(\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{\frac{1}{2}}} = \frac{\mu_{p,t} - r_f}{\sigma_{p,t}} \text{ s.t. } \mathbf{t}'\mathbf{1} = 1.$$

where  $\mu_{p,t} = \mathbf{t}'\boldsymbol{\mu}$  and  $\sigma_{p,t} = (\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{\frac{1}{2}}$ . The Lagrangian for this problem is

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

Using the chain rule, the first order conditions are

$$\begin{aligned} \frac{\partial L(\mathbf{t}, \lambda)}{\partial \mathbf{t}} &= \boldsymbol{\mu}(\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{-\frac{1}{2}} - (\mathbf{t}'\boldsymbol{\mu} - r_f) (\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{-3/2}\boldsymbol{\Sigma}\mathbf{t} + \lambda\mathbf{1} = 0 \\ \frac{\partial L(\mathbf{t}, \lambda)}{\partial \lambda} &= \mathbf{t}'\mathbf{1} - 1 = 0 \end{aligned}$$

## 22CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

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

$$\mathbf{t} = \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f \cdot \mathbf{1})}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f \cdot \mathbf{1})} \quad (1.25)$$

The location of the tangency portfolio, and the sign of the Sharpe ratio/slope, depends on the relationship between the risk free rate  $r_f$  and the expected return on the global minimum variance portfolio  $\mu_{p,m}$ . If  $\mu_{p,m} > r_f$ , which is the usual case, then the tangency portfolio will have a positive Sharpe slope. If  $\mu_{p,m} < r_f$ , which could occur when stock prices are falling and the economy is in a recession, then the tangency portfolio will have a negative Sharpe slope.

### Example 11 *Computing the tangency portfolio*

To compute the tangency portfolio (1.25) in R for the three risky assets in Table with  $r_f = 0.005$  use

```
> rf = 0.005
> sigma.mat.inv = solve(sigma.mat)
> one.vec = rep(1, 3)
> mu.minus.rf = mu.vec - rf*one.vec
> top = sigma.mat.inv*%*%mu.minus.rf
> bot = t(one.vec)%*%top
> t.vec = top/as.numeric(bot)
> t.vec
      [,1]
MSFT  1.0268
NORD -0.3263
SBUX  0.2994
```

The tangency portfolio weights  $t_{msft} = 1.0268$ ,  $t_{nord} = -0.3263$  and  $t_{sbux} = 0.2994$ , and is given by the vector

$$\mathbf{t} = (1.0268, -0.3263, 0.2994)'. \quad (1.26)$$

Notice that Nordstrom is sold short in this portfolio. The expected return on the tangency portfolio,  $\mu_{p,t} = \mathbf{t}'\boldsymbol{\mu}$ , is

```
> mu.t = crossprod(t.vec, mu.vec)
> mu.t
      [,1]
[1,] 0.05189
```

The portfolio variance,  $\sigma_{p,t}^2 = \mathbf{t}'\Sigma\mathbf{t}$ , and standard deviation,  $\sigma_{p,t}$ , are

```
> sig2.t = t(t.vec)%*%sigma.mat%*%t.vec
> sig.t = sqrt(sig2.t)
> sig2.t
      [,1]
[1,] 0.01245
> sig.t
      [,1]
[1,] 0.1116
```

Because  $r_f = 0.005 < \mu_{p,m} = 0.02489$  the tangency portfolio has a positive Sharpe's ratio/slope given by

$$SR_t = \frac{\mu_{p,t} - r_f}{\sigma_{p,t}} = \frac{0.05189 - 0.005}{0.1116} = .$$

■

### Alternative Derivation of the Tangency Portfolio

An alternative way to derive the tangency portfolio is to consider the

#### 1.1.6 Mutual Fund Separation Theorem Again

When there is a risk-free asset (T-bill) available, the efficient frontier of T-bills and risky assets consists of portfolios of T-bills and the tangency portfolio. The expected return and standard deviation of any such efficient portfolio are given by

$$\mu_p^e = r_f + x_t(\mu_{p,t} - r_f), \quad (1.27)$$

$$\sigma_p^e = x_t\sigma_{p,t}, \quad (1.28)$$

where  $x_t$  represents the fraction of wealth invested in the tangency portfolio ( $1 - x_T$  represents the fraction of wealth invested in T-Bills), and  $\mu_{p,t} = \mathbf{t}'\boldsymbol{\mu}$

and  $\sigma_{p,t} = (\mathbf{t}'\Sigma\mathbf{t})^{1/2}$  are the expected return and standard deviation on the tangency portfolio, respectively. Recall, this result is known as the *mutual fund separation theorem*. The tangency portfolio can be considered as a mutual fund of the risky assets, where the shares of the two assets in the mutual fund are determined by the tangency portfolio weights, and the T-bill can be considered as a mutual fund of risk free assets. The expected return-risk trade-off of these portfolios is given by the line connecting the risk-free rate to the tangency point on the efficient frontier of risky asset only portfolios. Which combination of the tangency portfolio and the T-bill an investor will choose depends on the investor's risk preferences. If the investor is very risk averse, then she will choose a combination with very little weight in the tangency portfolio and a lot of weight in the T-bill. This will produce a portfolio with an expected return close to the risk free rate and a variance that is close to zero. If the investor can tolerate a large amount of risk, then she would prefer a portfolio with highest expected return. This portfolio may involve borrowing at the risk-free rate (leveraging) and investing the proceeds in the tangency portfolio to achieve a high expected return.

**Example 12** *Efficient portfolios of three risky assets and T-bills*

■

## 1.2 Portfolio Analysis Functions in R

### 1.2.1 Eric Zivot's Portfolio Functions

The script file `portfolio.r` contains a few R functions for computing Markowitz mean-variance efficient portfolios allowing for short sales<sup>2</sup>. These functions are meant to be used for learning the basics of portfolio theory. They are summarized in Table 1.2.

The following examples illustrate the use of the functions in Table 1.2. These examples are also available in the R script file `testport.r` on the

---

<sup>2</sup>To use the functions in this script file, load them into R with the `source()` function. For example, if `portfolio.r` is located in `C:\mydata` run the command `source("C:/mydata/portfolio.r")` at the beginnin of your R session. Then the functions in the file will be available for your use.



Function	Description
<code>getPortfolio</code>	create portfolio object
<code>globalMin.portfolio</code>	compute global minimum variance portfolio
<code>efficient.portfolio</code>	compute minimum variance portfolio subject to target return
<code>tangency.portfolio</code>	compute tangency portfolio
<code>efficient.frontier</code>	compute efficient frontier of risky assets

Table 1.2: R functions for computing mean-variance efficient portfolios

class R hints web page. To begin, open the script file `portfolio.r` and select Edit/Run all to source the portfolio functions into memory.

Next, consider the input data ( $\mu$ ,  $\Sigma$  and  $r_f$ ) used for the three firm example used in the class lectures (see `3firmExample.xls` on the class syllabus web page).

```
> asset.names <- c("MSFT", "NORD", "SBUX")
> er <- c(0.0427, 0.0015, 0.0285)
> names(er) <- asset.names
> covmat <- matrix(c(0.0100, 0.0018, 0.0011,
+ 0.0018, 0.0109, 0.0026,
+ 0.0011, 0.0026, 0.0199),
+ nrow=3, ncol=3)
> rk.free <- 0.005
> dimnames(covmat) <- list(asset.names, asset.names)
```

To specify a portfolio, you need an expected return vector and covariance matrix for the assets under consideration as well as a vector of portfolio weights. To create an equally weighted portfolio use

```
> ew = rep(1,3)/3
> equalWeight.portfolio = getPortfolio(er=er,cov.mat=covmat,weights=ew)
> class(equalWeight.portfolio)
[1] "portfolio"
```

Portfolio objects have the following components

```
> names(equalWeight.portfolio)
[1] "call"    "er"      "sd"      "weights"
```

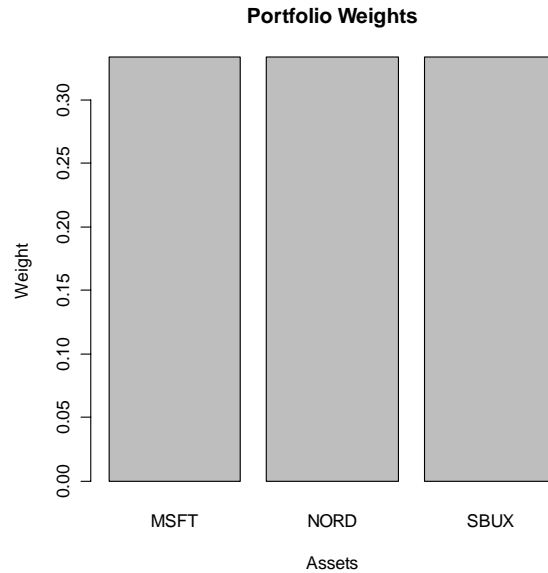


Figure 1.3:

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

```
> equalWeight.portfolio
```

```
Call:
```

```
getPortfolio(er = er, cov.mat = covmat, weights = ew)
```

```
Portfolio expected return:      0.02423333
```

```
Portfolio standard deviation:  0.07586538
```

```
Portfolio weights:
```

```
  MSFT  NORD  SBUX
0.3333 0.3333 0.3333
```

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

```
> plot(equalWeight.portfolio)
```

The global minimum variance portfolio (allowing for short sales)  $\mathbf{m}$  solves the optimization problem

$$\min_{\mathbf{m}} \mathbf{m}'\Sigma\mathbf{m} \text{ s.t. } \mathbf{m}'\mathbf{1} = 1.$$

To compute this portfolio use the function `globalMin.portfolio()`

```
> gmin.port <- globalMin.portfolio(er, covmat)
> attributes(gmin.port)
$names
[1] "call"      "er"        "sd"        "weights"

$class
[1] "portfolio"
> gmin.port
Call:
globalMin.portfolio(er = er, cov.mat = covmat)
```

```
Portfolio expected return:    0.02489184
Portfolio standard deviation: 0.07267607
Portfolio weights:
  MSFT  NORD  SBUX
0.4411 0.3656 0.1933
```

A mean-variance efficient portfolio  $\mathbf{x}$  that achieves the target expected return  $\mu_0$  solves the optimization problem

$$\min_x \mathbf{x}'\Sigma\mathbf{x} \text{ s.t. } \mathbf{x}'\mathbf{1} = 1 \text{ and } \mathbf{x}'\boldsymbol{\mu} = \mu_0.$$

To compute this portfolio for the target expected return  $\mu_0 = E[R_{msft}] = 0.04275$  use the `efficient.portfolio()` function

```
> target.return <- er[1]
> e.port.msft <- efficient.portfolio(er, covmat, target.return)
> e.port.msft
Call:
efficient.portfolio(er = er, cov.mat = covmat, target.return = target.return)
```

```
Portfolio expected return:    0.0427
Portfolio standard deviation: 0.091656
Portfolio weights:
  MSFT  NORD  SBUX
0.8275 -0.0907 0.2633
```

## 28 CHAPTER 1 PORTFOLIO THEORY WITH MATRIX ALGEBRA

The tangency portfolio  $\mathbf{t}$  is the portfolio of risky assets with the highest Sharpe's slope and solves the optimization problem

$$\max_{\mathbf{t}} \frac{\mathbf{t}'\boldsymbol{\mu} - r_f}{(\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{1/2}} \text{ s.t. } \mathbf{t}'\mathbf{1} = 1,$$

where  $r_f$  denotes the risk-free rate. To compute this portfolio with  $r_f = 0.005$  use the `tangency.portfolio()` function

```
> tan.port <- tangency.portfolio(er, covmat, rk.free)
> tan.port
Call:
tangency.portfolio(er = er, cov.mat = covmat, risk.free = rk.free)

Portfolio expected return:      0.05188967
Portfolio standard deviation:   0.1115816
Portfolio weights:
  MSFT   NORD   SBUX
1.0268 -0.3263  0.2994
```

The set of efficient portfolios of risky assets can be computed as a convex combination of any two efficient portfolios. It is convenient to use the global minimum variance portfolio as one portfolio and an efficient portfolio with target expected return equal to the maximum expected return of the assets under consideration as the other portfolio. Call these portfolios  $\mathbf{m}$  and  $\mathbf{x}$ , respectively. For any number  $\alpha$ , another efficient portfolio can be computed as

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

The function `efficient.frontier()` constructs the set of efficient portfolios using this method for a collection of  $\alpha$  values. For example, to compute 20 efficient portfolios for values of  $\alpha$  between  $-2$  and  $1.5$  use

```
> ef <- efficient.frontier(er, covmat, alpha.min=-2,
+                          alpha.max=1.5, nport=20)
> attributes(ef)
$names
[1] "call"   "er"     "sd"     "weights"

$class
```

```
[1] "Markowitz"
```

```
> ef
```

```
Call:
```

```
efficient.frontier(er = er, cov.mat = covmat, nport = 20, alpha.min = -2,
  alpha.max = 1.5)
```

Frontier portfolios' expected returns and standard deviations

```
  port 1 port 2 port 3 port 4 port 5 port 6 port 7
ER 0.0783 0.0750 0.0718 0.0685 0.0652 0.0619 0.0586
SD 0.1826 0.1732 0.1640 0.1548 0.1458 0.1370 0.1284
  port 8 port 9 port 10 port 11 port 12 port 13 port 14
ER 0.0554 0.0521 0.0488 0.0455 0.0422 0.039 0.0357
SD 0.1200 0.1120 0.1044 0.0973 0.0908 0.085 0.0802
  port 15 port 16 port 17 port 18 port 19 port 20
ER 0.0324 0.0291 0.0258 0.0225 0.0193 0.0160
SD 0.0764 0.0739 0.0727 0.0730 0.0748 0.0779
```

Use the `summary()` method to show the weights of these portfolios. Use the `plot()` method to plot the efficient frontier

```
> plot(ef)
```

The resulting plot is shown in Figure 1.4.

To create a plot of the efficient frontier showing the original assets and the tangency portfolio use

```
> plot(ef, plot.assets=T)
> points(gmin.port$sd, gmin.port$er, col="blue")
> points(tan.port$sd, tan.port$er, col="red")
> sr.tan = (tan.port$er - rk.free)/tan.port$sd
> abline(a=rk.free, b=sr.tan)
```

The resulting plot is shown in Figure 1.5.

### 1.2.2 Rmetrics Package fPortfolio

To be completed.

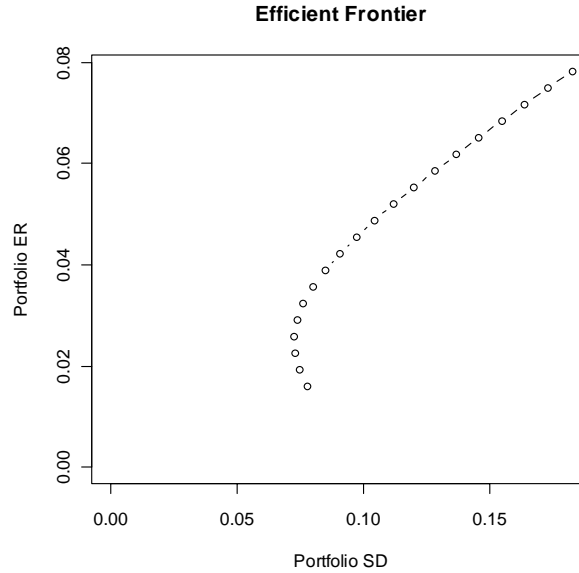


Figure 1.4: Plot method for Markowitz object.

## 1.3 Appendix

### 1.3.1 A Digression on the Covariance Matrix

The covariance matrix of returns,  $\Sigma$ , summarizes the variances and covariances of the individual returns in the return vector  $\mathbf{R}$ . In general, the covariance matrix of a random vector  $\mathbf{R}$  (sometimes simply called the variance of vector  $\mathbf{R}$ ) with mean vector  $\boldsymbol{\mu}$  is defined as

$$\text{cov}(\mathbf{R}) = E[(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})'] = \Sigma$$

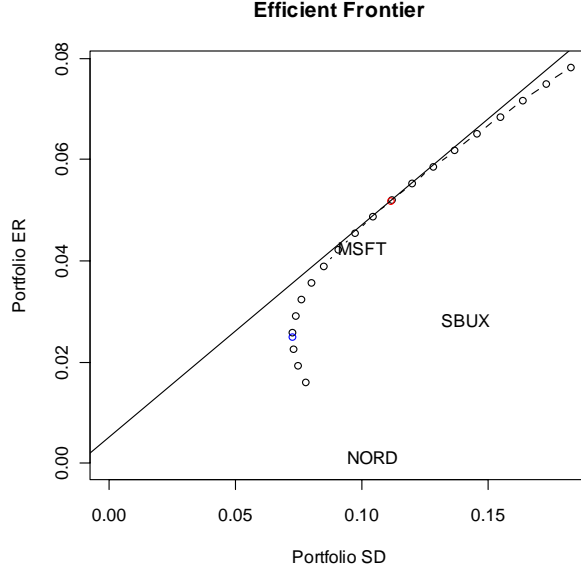


Figure 1.5: Efficient frontier for three firm example.

If  $\mathbf{R}$  has  $N$  elements then  $\Sigma$  will be an  $N \times N$  matrix. For the case  $N = 2$ , we have

$$\begin{aligned}
 E[(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})'] &= E \left[ \begin{pmatrix} R_A - \mu_A \\ R_B - \mu_B \end{pmatrix} \cdot (R_A - \mu_A, R_B - \mu_B) \right] \\
 &= E \left[ \begin{pmatrix} (R_A - \mu_A)^2 & (R_A - \mu_A)(R_B - \mu_B) \\ (R_B - \mu_B)(R_A - \mu_A) & (R_B - \mu_B)^2 \end{pmatrix} \right] \\
 &= \begin{pmatrix} E[(R_A - \mu_A)^2] & E[(R_A - \mu_A)(R_B - \mu_B)] \\ E[(R_B - \mu_B)(R_A - \mu_A)] & E[(R_B - \mu_B)^2] \end{pmatrix} \\
 &= \begin{pmatrix} \text{var}(R_A) & \text{cov}(R_A, R_B) \\ \text{cov}(R_B, R_A) & \text{var}(R_B) \end{pmatrix} = \begin{pmatrix} \sigma_A^2 & \sigma_{AB} \\ \sigma_{AB} & \sigma_B^2 \end{pmatrix} = \Sigma.
 \end{aligned}$$

We can use the formal definition of  $\text{cov}(\mathbf{R})$  to derive the variance of a portfolio. Consider again the two asset case. The variance of the portfolio

$R_p = \mathbf{x}'\mathbf{R}$  is given by

$$\text{var}(R_p) = \text{var}(\mathbf{x}'\mathbf{R}) = E[(\mathbf{x}'\mathbf{R} - \mathbf{x}'\boldsymbol{\mu})^2] = E[(\mathbf{x}'(\mathbf{R} - \boldsymbol{\mu}))^2]$$

since  $\mathbf{x}'\mathbf{R}$  is a scalar. Now we use a trick from matrix algebra. If  $z$  is a scalar (think of  $z = 2$ ) then  $z'z = z \cdot z' = z^2$ . Let  $z = \mathbf{x}'(\mathbf{R} - \boldsymbol{\mu})$  and so  $z \cdot z' = \mathbf{x}'(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})'\mathbf{x}$ . Then

$$\begin{aligned} \text{var}(R_p) &= E[z^2] = E[z \cdot z'] \\ &= E[\mathbf{x}'(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})'\mathbf{x}] \\ &= \mathbf{x}'E[(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})']\mathbf{x} \\ &= \mathbf{x}'\text{cov}(\mathbf{R})\mathbf{x} = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x}. \end{aligned}$$

Next consider determining the covariance between the returns on two portfolios  $\mathbf{x}$  and  $\mathbf{y}$ . The returns on these two portfolios are  $R_{p,x} = \mathbf{x}'\mathbf{R}$  and  $R_{p,y} = \mathbf{y}'\mathbf{R}$ . From the definition of covariance we have

$$\text{cov}(R_{p,x}, R_{p,y}) = E[(R_{p,x} - \mu_{p,x})(R_{p,y} - \mu_{p,y})]$$

which may be rewritten in matrix notation as

$$\begin{aligned} \text{cov}(\mathbf{x}'\mathbf{R}, \mathbf{y}'\mathbf{R}) &= E[(\mathbf{x}'\mathbf{R} - \mathbf{x}'\boldsymbol{\mu})(\mathbf{y}'\mathbf{R} - \mathbf{y}'\boldsymbol{\mu})] \\ &= E[\mathbf{x}'(\mathbf{R} - \boldsymbol{\mu})\mathbf{y}'(\mathbf{R} - \boldsymbol{\mu})] \\ &= E[\mathbf{x}'(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})'\mathbf{y}] \\ &= \mathbf{x}'E[(\mathbf{R} - \boldsymbol{\mu})(\mathbf{R} - \boldsymbol{\mu})']\mathbf{y} \\ &= \mathbf{x}'\boldsymbol{\Sigma}\mathbf{y}. \end{aligned}$$

### 1.3.2 Derivatives of Simple Matrix Functions

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

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}'\mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1} \mathbf{x}'\mathbf{y} \\ \vdots \\ \frac{\partial}{\partial x_n} \mathbf{x}'\mathbf{y} \end{pmatrix} = \mathbf{y}, \quad (1.29)$$

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}'\mathbf{A}\mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1} \mathbf{x}'\mathbf{A}\mathbf{x} \\ \vdots \\ \frac{\partial}{\partial x_n} \mathbf{x}'\mathbf{A}\mathbf{x} \end{pmatrix} = 2\mathbf{A}\mathbf{x}. \quad (1.30)$$



We will demonstrate these results with simple examples. Let

$$\mathbf{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

First, consider (1.29). Now

$$\mathbf{x}'\mathbf{y} = x_1y_1 + x_2y_2.$$

Then

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}'\mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1} \mathbf{x}'\mathbf{y} \\ \frac{\partial}{\partial x_2} \mathbf{x}'\mathbf{y} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} (x_1y_1 + x_2y_2) \\ \frac{\partial}{\partial x_2} (x_1y_1 + x_2y_2) \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{y}.$$

Next, consider (1.30). We have

$$\mathbf{x}'\mathbf{A}\mathbf{x} = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = ax_1^2 + 2bx_1x_2 + bx_2^2.$$

Then

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \mathbf{x}'\mathbf{A}\mathbf{x} &= \begin{pmatrix} \frac{\partial}{\partial x_1} (ax_1^2 + 2bx_1x_2 + bx_2^2) \\ \frac{\partial}{\partial x_2} (ax_1^2 + 2bx_1x_2 + bx_2^2) \end{pmatrix} = \begin{pmatrix} 2ax_1 + 2bx_2 \\ 2bx_1 + 2bx_2 \end{pmatrix} \\ &= 2\mathbf{A}\mathbf{x}. \end{aligned}$$

## 1.4 References