Chapter 1

Matrix Algebra Review

This chapter reviews some basic matrix algebra concepts that we will use throughout the book.

Updated: August 15, 2013.

1.1 Matrices and Vectors

A *matrix* is just an array of numbers. The *dimension* of a matrix is determined by the number of its rows and columns. For example, a matrix $\mathbf{A}$ with n rows and m columns is illustrated below

$$\underset{(n\times m)}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \vdots & \vdots & \ldots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{bmatrix}$$

where a_{ij} denotes the i^{th} row and j^{th} column element of $\mathbf{A}$.

A *vector* is simply a matrix with 1 column. For example,

$$\underset{(n\times 1)}{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

is an $n \times 1$ vector with elements $x_1, x_2, \ldots, x_n$. Vectors and matrices are often written in bold type (or underlined) to distinguish them from scalars (single elements of vectors or matrices).

Example 1 *Matrix creation in R*

In R, matrix objects are created using the `matrix()` function. For example, to create the 2×3 matrix

$$\underset{(2\times3)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

use

```
> matA = matrix(data=c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=TRUE)
> matA
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> class(matA)
[1] "matrix"
```

The optional argument `byrow=TRUE` fills the matrix row by row.[1] The default is `byrow=FALSE` which fills the matrix column by column:

```
> matrix(data=c(1,2,3,4,5,6),nrow=2,ncol=3)
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Matrix objects have row and column dimension attributes which can be examined with the `dim()` function:

```
> dim(matA)
[1] 2 3
```

[1]When specifying logical variables in R always spell out `TRUE` and `FALSE` instead of using `T` and `F`. Upon startup R defines the variables `T=TRUE` and `F=FALSE` so that `T` and `F` can be used as substitutes for `TRUE` and `FALSE`, respectively. However, this shortcut is not recommended because the variables `T` and `F` could be reassigned during subsequent programming.

The rows and columns can be given names using

```
> dimnames(matA) = list(c("row1","row2"),c("col1","col2","col3"))
> matA
     col1 col2 col3
row1    1    2    3
row2    4    5    6
```

or

```
> colnames(matA) = c("Col1", "Col2", "Col3")
> rownames(matA) = c("Row1", "Row2")
> matA
     Col1 Col2 Col3
Row1    1    2    3
Row2    4    5    6
```

The elements of a matrix can extracted or subsetted as follows:

```
> matA[1, 2]
[1] 2
> matA["Row1", "Col1"]
[1] 1
> matA[1, ]
Col1 Col2 Col3
   1    2    3
> matA[, 2]
Row1 Row2
   2    5
```

To preserve the dimension attributes when subsetting use the `drop=FALSE` option:

```
> matA[1, , drop=FALSE]
     Col1 Col2 Col3
Row1    1    2    3
> matA[, 2, drop=FALSE]
     Col2
Row1    2
Row2    5
```

■

Example 2 *Creating vectors in R*

Vectors can be created in R using a variety of methods:

```
> xvec = c(1,2,3)
> xvec
[1] 1 2 3
> xvec = 1:3
> xvec
[1] 1 2 3
> xvec = seq(from=1,to=3,by=1)
> xvec
[1] 1 2 3
```

Vectors in R are of class `numeric` and do not have a dimension attribute:

```
> class(xvec)
[1] "numeric"
> dim(xvec)
NULL
```

The elements of a vector can be assigned names using the `names()` function:

```
> names(xvec) = c("x1", "x2", "x3")
> xvec
x1 x2 x3
 1  2  3
```

To force a dimension attribute onto a vector, coerce it to a `matrix` object using `as.matrix()`:

```
> xvec = as.matrix(xvec)
> xvec
   [,1]
x1    1
x2    2
x3    3
> class(xvec)
[1] "matrix"
> dim(xvec)
[1] 3 1
```

■

The *transpose* of an $n \times m$ matrix $\mathbf{A}$ is a new matrix with the rows and columns of $\mathbf{A}$ interchanged, and is denoted $\mathbf{A}'$ or $\mathbf{A}^{\intercal}$. For example,

$$\underset{(2\times3)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \ \underset{(3\times2)}{\mathbf{A}'} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\underset{(3\times1)}{\mathbf{x}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \ \underset{(1\times3)}{\mathbf{x}'} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}.$$

A *symmetric* matrix $\mathbf{A}$ is such that $\mathbf{A} = \mathbf{A}'$. Obviously, this can only occur if $\mathbf{A}$ is a *square* matrix; i.e., the number of rows of $\mathbf{A}$ is equal to the number of columns. For example, consider the 2×2 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Then,

$$\mathbf{A}' = \mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Example 3 *Transpose of a matrix in R*

To take the transpose of a matrix or vector use the `t()` function

```
> matA = matrix(data=c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=TRUE)
> t(matA)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> xvec = c(1,2,3)
> t(xvec)
     [,1] [,2] [,3]
[1,]    1    2    3
```

Notice that, when applied to a vector with n elements, the `t()` function returns a `matrix` object with dimension $1 \times n$. ■

1.2 Basic Matrix Operations

In this section we review the basic matrix operations of addition, subtraction, scalar multiplication and multiplication.

1.2.1 Addition and subtraction

Matrix addition and subtraction are element by element operations and only apply to matrices of the same dimension. For example, let

$$\mathbf{A} = \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix}.$$

Then

$$\begin{aligned} \mathbf{A}+\mathbf{B} &= \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} = \begin{bmatrix} 4+2 & 9+0 \\ 2+0 & 1+7 \end{bmatrix} = \begin{bmatrix} 6 & 9 \\ 2 & 8 \end{bmatrix}, \\ \mathbf{A}-\mathbf{B} &= \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} = \begin{bmatrix} 4-2 & 9-0 \\ 2-0 & 1-7 \end{bmatrix} = \begin{bmatrix} 2 & 9 \\ 2 & -6 \end{bmatrix}. \end{aligned}$$

Example 4 *Matrix addition and subtraction in R*

Matrix addition and subtraction is straightforward in R:

```
> matA = matrix(c(4,9,2,1),2,2,byrow=TRUE)
> matB = matrix(c(2,0,0,7),2,2,byrow=TRUE)
> matA
     [,1] [,2]
[1,]    4    9
[2,]    2    1
> matB
     [,1] [,2]
[1,]    2    0
```

```
[2,]    0    7
> # matrix addition
> matC = matA + matB
> matC
     [,1] [,2]
[1,]    6    9
[2,]    2    8
> # matrix subtraction
> matC = matA - matB
> matC
     [,1] [,2]
[1,]    2    9
[2,]    2   -6
```

■

1.2.2 Scalar Multiplication

Here we refer to the multiplication of a matrix by a scalar number. This is also an element-by-element operation. For example, let $c = 2$ and

$$\mathbf{A} = \begin{bmatrix} 3 & -1 \\ 0 & 5 \end{bmatrix}.$$

Then

$$c \cdot \mathbf{A} = \begin{bmatrix} 2 \cdot 3 & 2 \cdot (-1) \\ 2 \cdot (0) & 2 \cdot 5 \end{bmatrix} = \begin{bmatrix} 6 & -2 \\ 0 & 10 \end{bmatrix}.$$

Example 5 *Scalar multiplication in R*

```
> matA = matrix(c(3,-1,0,5),2,2,byrow=TRUE)
> matC = 2*matA
> matC
     [,1] [,2]
[1,]    6   -2
[2,]    0   10
```

■

1.2.3 Matrix Multiplication

Matrix multiplication only applies to *conformable* matrices. **A** and **B** are conformable matrices if the number of columns in **A** is equal to the number of rows in **B**. For example, if **A** is $n \times m$ and **B** is $m \times p$ then **A** and **B** are conformable and the matrix product of **A** and **B** has dimension $n \times p$. The mechanics of matrix multiplication is best explained by example. Let

$$\underset{(2\times 2)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } \underset{(2\times 3)}{\mathbf{B}} = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 2 \end{bmatrix}.$$

Then

$$\begin{aligned} \underset{(2\times 2)}{\mathbf{A}} \cdot \underset{(2\times 3)}{\mathbf{B}} &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 1\cdot 1+2\cdot 3 & 1\cdot 2+2\cdot 4 & 1\cdot 1+2\cdot 2 \\ 3\cdot 1+4\cdot 3 & 3\cdot 2+4\cdot 4 & 3\cdot 1+4\cdot 2 \end{bmatrix} \\ &= \begin{bmatrix} 7 & 10 & 5 \\ 15 & 22 & 11 \end{bmatrix} = \underset{(2\times 3)}{\mathbf{C}} \end{aligned}$$

The resulting matrix **C** has 2 rows and 3 columns. In general, if **A** is $n \times m$ and **B** is $m \times p$ then $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ is $n \times p$.

As another example, let

$$\underset{(2\times 2)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } \underset{(2\times 1)}{\mathbf{B}} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}.$$

Then

$$\underset{(2\times 2)}{\mathbf{A}} \cdot \underset{(2\times 1)}{\mathbf{B}} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 6 \end{bmatrix} = \begin{bmatrix} 1\cdot 2 + 2\cdot 6 \\ 3\cdot 2 + 4\cdot 6 \end{bmatrix} = \begin{bmatrix} 14 \\ 30 \end{bmatrix}.$$

As a final example, let

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}.$$

Then

$$\mathbf{x}'\mathbf{y} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = 1\cdot 4 + 2\cdot 5 + 3\cdot 6 = 32$$

Example 6 *Matrix multiplication in R*

In R, matrix multiplication is performed with the `%*%` operator. For example:

```
> matA = matrix(1:4,2,2,byrow=TRUE)
> matB = matrix(c(1,2,1,3,4,2),2,3,byrow=TRUE)
> matA
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> matB
     [,1] [,2] [,3]
[1,]    1    2    1
```

```
[2,]    3    4    2
> dim(matA)
[1] 2 2
> dim(matB)
[1] 2 3
> matC = matA%*%matB
> matC
     [,1] [,2] [,3]
[1,]    7   10    5
[2,]   15   22   11
> # note: B%*%A doesn't work b/c B and A are not comformable
> matB%*%matA
Error in matB %*% matA : non-conformable arguments
```

The next example shows matrix multiplication in R also works on numeric vectors:

```
> matA = matrix(c(1,2,3,4), 2, 2, byrow=TRUE)
> vecB = c(2,6)
> matA%*%vecB
     [,1]
[1,]   14
[2,]   30
> vecX = c(1,2,3)
> vecY = c(4,5,6)
> t(vecX)%*%vecY
     [,1]
[1,]   32
> crossprod(vecX, vecY)
     [,1]
[1,]   32
```

■

1.2.4 The Identity Matrix

The identity matrix plays a similar role as the number 1. Multiplying any number by 1 gives back that number. In matrix algebra, pre or post multiplying a matrix **A** by a conformable identity matrix gives back the matrix

A. To illustrate, let

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

denote the 2 dimensional identity matrix and let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

denote an arbitrary 2×2 matrix. Then

$$\begin{aligned} \mathbf{I}_2 \cdot \mathbf{A} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}, \end{aligned}$$

and

$$\begin{aligned} \mathbf{A} \cdot \mathbf{I}_2 &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}. \end{aligned}$$

Example 7 *The identity matrix in R*

Use the `diag()` function to create an identity matrix:

```
> matI = diag(2)
> matI
     [,1] [,2]
[1,]    1    0
[2,]    0    1
> matA = matrix(c(1,2,3,4), 2, 2, byrow=TRUE)
> matI%*%matA
```

```
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> matA%*%matI
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

■

1.3 Representing Summation Using Vector Notation

Consider the sum

$$\sum_{k=1}^{n} x_k = x_1 + \cdots + x_n.$$

Let $\mathbf{x} = (x_1, \ldots, x_n)'$ be an $n \times 1$ vector and $\mathbf{1} = (1, \ldots, 1)'$ be an $n \times 1$ vector of ones. Then

$$\mathbf{x}'\mathbf{1} = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = x_1 + \cdots + x_n = \sum_{k=1}^{n} x_k,$$

and

$$\mathbf{1}'\mathbf{x} = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 + \cdots + x_n = \sum_{k=1}^{n} x_k.$$

Next, consider the sum of squared x values

$$\sum_{k=1}^{n} x_k^2 = x_1^2 + \cdots + x_n^2.$$

This sum can be conveniently represented as

$$\mathbf{x}'\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1^2 + \cdots + x_n^2 = \sum_{k=1}^{n} x_k^2.$$

Last, consider the sum of cross products

$$\sum_{k=1}^{n} x_k y_k = x_1 y_1 + \cdots x_n y_n.$$

This sum can be compactly represented by

$$\mathbf{x}'\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + \cdots x_n y_n = \sum_{k=1}^{n} x_k y_k.$$

Note that $\mathbf{x}'\mathbf{y} = \mathbf{y}'\mathbf{x}$.

Example 8 *Computing sums in R*

In R, summing the elements in a vector can be done using matrix algebra.

```
# create vector of 1's and a vector x
> onevec = rep(1,3)
> onevec
[1] 1 1 1
> xvec = c(1,2,3)
> xvec
[1] 1 2 3
# sum elements in x
> t(xvec)%*%onevec
     [,1]
[1,]    6
```

The functions `crossprod()` and `sum()` are generally computationally more efficient:

```
> crossprod(xvec,onevec)
     [,1]
[1,]    6
> sum(xvec)
[1] 6
```

Sums of squares are best computed using

```
> crossprod(xvec)
     [,1]
[1,]   14
> sum(xvec^2)
[1] 14
```

The dot-product or cross-product of two vectors can be conveniently computed using the `crossprod()` function:

```
> yvec = 4:6
> xvec
[1] 1 2 3
> yvec
[1] 4 5 6
> crossprod(xvec,yvec)
     [,1]
[1,]   32
> crossprod(yvec,xvec)
     [,1]
[1,]   32
```

■

1.4 Systems of Linear Equations

Consider the system of two linear equations

$$x + y = 1 \tag{1.1}$$
$$2x - y = 1 \tag{1.2}$$

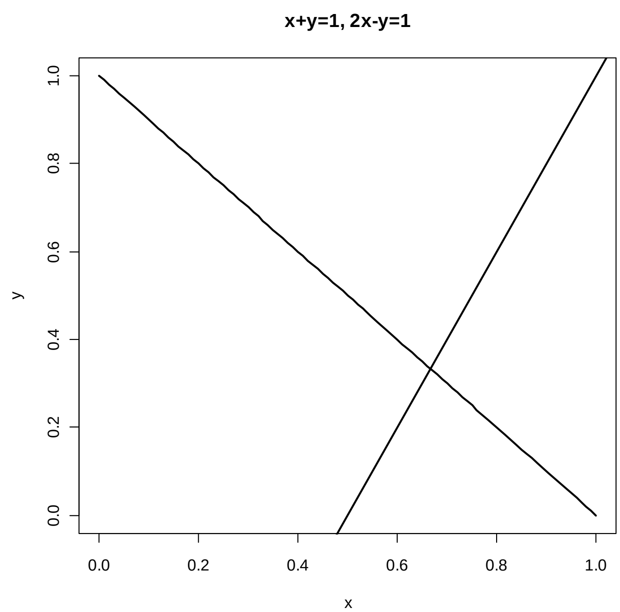

Figure 1.1: System of two linear equations.

As shown in Figure 1.1, equations (1.1) and (1.2) represent two straight lines which intersect at the point $x = \frac{2}{3}$ and $y = \frac{1}{3}$. This point of intersection is determined by solving for the values of x and y such that $x + y = 2x - y$.[2]

The two linear equations can be written in matrix form as

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

or

$$\mathbf{A} \cdot \mathbf{z} = \mathbf{b},$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \ \mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

[2]Soving for x gives $x = 2y$. Substituting this value into the equation $x + y = 1$ gives $2y + y = 1$ and solving for y gives $y = 1/3$. Solving for x then gives $x = 2/3$.

If there was a (2×2) matrix $\mathbf{B}$, with elements b_{ij}, such that $\mathbf{B} \cdot \mathbf{A} = \mathbf{I}_2$, where $\mathbf{I}_2$ is the (2×2) identity matrix, then we could *solve* for the elements in $\mathbf{z}$ as follows. In the equation $\mathbf{A} \cdot \mathbf{z} = \mathbf{b}$, pre-multiply both sides by $\mathbf{B}$ to give

$$\begin{aligned} \mathbf{B} \cdot \mathbf{A} \cdot \mathbf{z} &= \mathbf{B} \cdot \mathbf{b} \\ &\Longrightarrow \mathbf{I} \cdot \mathbf{z} = \mathbf{B} \cdot \mathbf{b} \\ &\Longrightarrow \mathbf{z} = \mathbf{B} \cdot \mathbf{b}, \end{aligned}$$

or

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} \cdot 1 + b_{12} \cdot 1 \\ b_{21} \cdot 1 + b_{22} \cdot 1 \end{bmatrix}$$

If such a matrix $\mathbf{B}$ exists it is called the *inverse* of $\mathbf{A}$ and is denoted $\mathbf{A}^{-1}$. Intuitively, the inverse matrix $\mathbf{A}^{-1}$ plays a similar role as the inverse of a number. Suppose a is a number; e.g., $a = 2$. Then we know that $\frac{1}{a} \cdot a = a^{-1}a = 1$. Similarly, in matrix algebra $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_2$ where $\mathbf{I}_2$ is the identity matrix. Now, consider solving the equation $a \cdot x = 1$. By simple division we have that $x = \frac{1}{a}x = a^{-1}x$. Similarly, in matrix algebra if we want to solve the system of linear equations $\mathbf{Ax} = \mathbf{b}$ we multiply by $\mathbf{A}^{-1}$ and get the solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

Using $\mathbf{B} = \mathbf{A}^{-1}$, we may express the solution for $\mathbf{z}$ as

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

As long as we can determine the elements in $\mathbf{A}^{-1}$ then we can solve for the values of x and y in the vector $\mathbf{z}$. The system of linear equations has a solution as long as the two lines intersect, so we can determine the elements in $\mathbf{A}^{-1}$ provided the two lines are not parallel. If the two lines are parallel, then one of the equations is a multiple of the other. In this case we say that $\mathbf{A}$ is *not invertible*.

There are general numerical algorithms for finding the elements of $\mathbf{A}^{-1}$ (e.g., so-called Gaussian elimination) and matrix programming languages and spreadsheets have these algorithms available. However, if $\mathbf{A}$ is a (2×2) matrix then there is a simple formula for $\mathbf{A}^{-1}$. Let $\mathbf{A}$ be a (2×2) matrix such that

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Then

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix},$$

where $\det(\mathbf{A}) = a_{11}a_{22} - a_{21}a_{12}$ denotes the determinant of $\mathbf{A}$ and is assumed to be not equal to zero. By brute force matrix multiplication we can verify this formula:

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{A} &= \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ &= \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22}a_{11} - a_{12}a_{21} & a_{22}a_{12} - a_{12}a_{22} \\ -a_{21}a_{11} + a_{11}a_{21} & -a_{21}a_{12} + a_{11}a_{22} \end{bmatrix} \\ &= \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22}a_{11} - a_{12}a_{21} & 0 \\ 0 & -a_{21}a_{12} + a_{11}a_{22} \end{bmatrix} \\ &= \begin{bmatrix} \frac{a_{22}a_{11} - a_{12}a_{21}}{a_{11}a_{22} - a_{21}a_{12}} & 0 \\ 0 & \frac{-a_{21}a_{12} + a_{11}a_{22}}{a_{11}a_{22} - a_{21}a_{12}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

Let's apply the above rule to find the inverse of $\mathbf{A}$ in our example linear system (1.1)-(1.2):

$$\mathbf{A}^{-1} = \frac{1}{-1-2} \begin{bmatrix} -1 & -1 \\ -2 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix}.$$

Notice that

$$\mathbf{A}^{-1}\mathbf{A} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Our solution for $\mathbf{z}$ is then

$$\begin{aligned}\mathbf{z} &= \mathbf{A}^{-1}\mathbf{b} \\ &= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}\end{aligned}$$

so that $x = \frac{2}{3}$ and $y = \frac{1}{3}$.

Example 9 *Solving systems of linear equations in R*

In R, the `solve()` function is used to compute the inverse of a matrix and solve a system of linear equations. The linear system $x+y = 1$ and $2x-y = 1$ can be represented using

```
matA = matrix(c(1,1,2,-1), 2, 2, byrow=TRUE)
vecB = c(1,1)
```

First we solve for $\mathbf{A}^{-1}$:[3]

```
> matA.inv = solve(matA)
> matA.inv
       [,1]    [,2]
[1,] 0.3333  0.3333
[2,] 0.6667 -0.3333
> matA.inv%*%matA
     [,1]       [,2]
[1,]    1 -5.551e-17
[2,]    0  1.000e+00
> matA%*%matA.inv
     [,1]      [,2]
[1,]    1 5.551e-17
[2,]    0 1.000e+00
```

[3] Notice that the calculations in R do not show $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ exactly. The $(1, 2)$ element of $\mathbf{A}^{-1}\mathbf{A}$ is -5.552e-17, which for all practical purposes is zero. However, due to the limitations of machine calculations the result is not exactly zero.

Then we solve the system $\mathbf{z} = \mathbf{A}^{-1}\mathbf{b}$:

```
> z = matA.inv%*%vecB
> z
       [,1]
[1,] 0.6667
[2,] 0.3333
```

■

In general, if we have n linear equations in n unknown variables we may write the system of equations as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

which we may then express in matrix form as

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

or

$$\underset{(n\times n)}{\mathbf{A}} \cdot \underset{(n\times 1)}{\mathbf{x}} = \underset{(n\times 1)}{\mathbf{b}}.$$

The solution to the system of equations is given by

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

where $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ and $\mathbf{I}$ is the $(n \times n)$ identity matrix. If the number of equations is greater than two, then we generally use numerical algorithms to find the elements in $\mathbf{A}^{-1}$.

1.4.1 Partitioned Matrices and Partitioned Inverses

To be completed

1.5 Positive Definite Matrices

To be completed

1.6 Multivariate Probability Distributions Using Matrix Algebra

In this section, we show how matrix algebra can be used to simplify many of the messy expressions concerning expectations and covariances between multiple random variables, and we show how certain multivariate probability distributions (e.g., the multivariate normal distribution) can be expressed using matrix algebra.

1.6.1 Random Vectors

Let $X_1, \ldots, X_n$ denote n random variables for $i = 1, \ldots, n$ let $\mu_i = E[X_i]$ and $\sigma_i^2 = \text{var}(X_i)$, and let $\sigma_{ij} = \text{cov}(X_i, X_j)$ for $i \neq j$. Define the $n \times 1$ random vector $\mathbf{X} = (X_1, \ldots, X_n)'$. Associated with $\mathbf{X}$ is the $n \times 1$ vector of expected values

$$\underset{n\times 1}{\boldsymbol{\mu}} = E[\mathbf{X}] = \begin{pmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}.$$

1.6.2 Covariance Matrix

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

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

If $\mathbf{X}$ has n elements then $\mathbf{\Sigma}$ will be the symmetric $n \times n$ matrix

$$\underset{n\times n}{\mathbf{\Sigma}} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{pmatrix}.$$

For the case $N = 2$, we have

$$\begin{aligned} E[(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})'] &= E\left[\begin{pmatrix} X_1-\mu_1 \\ X_2-\mu_2 \end{pmatrix} \cdot (X_1-\mu_1, X_2-\mu_2)\right] \\ &= E\left[\begin{pmatrix} (X_1-\mu_1)^2 & (X_1-\mu_1)(X_2-\mu_2) \\ (X_2-\mu_2)(X_1-\mu_1) & (X_2-\mu_2)^2 \end{pmatrix}\right] \\ &= \begin{pmatrix} E[(X_1-\mu_1)^2] & E[(X_1-\mu_1)(X_2-\mu_2)] \\ E[(X_2-\mu_2)(X_1-\mu_1)] & E[(X_2-\mu_2)^2] \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{var}(X_1) & \mathrm{cov}(X_1,X_2) \\ \mathrm{cov}(X_2,X_1) & \mathrm{var}(X_2) \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \mathbf{\Sigma}. \end{aligned}$$

1.6.3 Variance of Linear Combination of Random Vectors

Consider the $n \times 1$ random vector $\mathbf{X}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$. Let $\mathbf{a} = (a_1, \ldots, a_n)'$ be an $n \times 1$ vector of constants and consider the random variable $Y = \mathbf{a}'\mathbf{X} = a_1X_1 + \cdots + a_nX_n$. Then

$$\mu_Y = E[Y] = E[\mathbf{a}'\mathbf{X}] = \mathbf{a}'E[\mathbf{X}] = \mathbf{a}'\boldsymbol{\mu}.$$

and

$$\mathrm{var}(Y) = \mathrm{var}(\mathbf{a}'\mathbf{X}) = E[(\mathbf{a}'\mathbf{X} - \mathbf{a}'\boldsymbol{\mu})^2] = E[(\mathbf{a}'(\mathbf{X}-\boldsymbol{\mu}))^2]$$

since $Y = \mathbf{a}'\mathbf{X}$ 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 $z = \mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})$ and so

$z \cdot z' = \mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})'\mathbf{a}$. Then

$$\begin{aligned}
\mathrm{var}(Y) &= E[z^2] = E[z \cdot z'] \\
&= E[\mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})'\mathbf{a}] \\
&= \mathbf{a}'E[(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})']\mathbf{a} \\
&= \mathbf{a}'\mathrm{cov}(\mathbf{X})\mathbf{a} = \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}.
\end{aligned}$$

1.6.4 Covariance Between Linear Combination of Two Random Vectors

Consider the $n \times 1$ random vector $\mathbf{X}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Let $\mathbf{a} = (a_1, \ldots, a_n)'$ and $\mathbf{b} = (b_1, \ldots, b_n)'$ be $n \times 1$ vectors of constants, and consider the random variable $Y = \mathbf{a}'\mathbf{X} = a_1X_1 + \cdots + a_nX_n$ and $Z = \mathbf{b}'\mathbf{X} = b_1X_1 + \cdots + b_nX_n$. From the definition of covariance we have

$$\mathrm{cov}(Y, Z) = E[(Y - E[Y])(Z - E[Z])]$$

which may be rewritten in matrix notation as

$$\begin{aligned}
\mathrm{cov}(\mathbf{a}'\mathbf{X}, \mathbf{b}'\mathbf{X}) &= E[(\mathbf{a}'\mathbf{X}-\mathbf{a}'\boldsymbol{\mu})(\mathbf{b}'\mathbf{X}-\mathbf{b}'\boldsymbol{\mu})] \\
&= E[\mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})\mathbf{b}'(\mathbf{X}-\boldsymbol{\mu})] \\
&= E[\mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})'\mathbf{b}] \\
&= \mathbf{a}'E[(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})']\mathbf{b} \\
&= \mathbf{a}'\boldsymbol{\Sigma}\mathbf{b}.
\end{aligned}$$

Since $\mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})$ and $\mathbf{b}'(\mathbf{X}-\boldsymbol{\mu})$ are scalars, we can use the trick

$$\mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})\mathbf{b}'(\mathbf{X}-\boldsymbol{\mu}) = \mathbf{a}'(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})'\mathbf{b}.$$

1.6.5 Bivariate Normal Distribution

Let X and Y be distributed bivariate normal. The joint pdf is given by

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}} \times \tag{1.3}$$

$$\exp\left\{-\frac{1}{2(1-\rho_{XY}^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 - \frac{2\rho_{XY}(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right\}$$

where $E[X] = \mu_X$, $E[Y] = \mu_Y$, $\mathrm{sd}(X) = \sigma_X$, $\mathrm{sd}(Y) = \sigma_Y$, and $\rho_{XY} = \mathrm{cor}(X,Y)$. The correlation coefficient ρ_{XY} describes the dependence between X and Y. If $\rho_{XY} = 0$ then X and Y are independent and the pdf collapses to the pdf of the standard bivariate normal distribution.

The formula for the bivariate normal distribution (1.3) is a bit messy. We can greatly simplify the formula by using matrix algebra. Define the 2×1 vectors $\mathbf{x} = (x, y)'$ and $\boldsymbol{\mu} = (\mu_X, \mu_y)'$, and the 2×2 matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix}.$$

Then the bivariate normal distribution (1.3) may be compactly expressed as

$$f(\mathbf{x}) = \frac{1}{2\pi \det(\boldsymbol{\Sigma})^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})},$$

where

$$\det(\boldsymbol{\Sigma}) = \sigma_X^2\sigma_Y^2 - \sigma_{XY}^2 = \sigma_X^2\sigma_Y^2 - \sigma_X^2\sigma_Y^2\rho_{XY}^2 = \sigma_X^2\sigma_Y^2(1-\rho_{XY}^2).$$

1.6.6 Multivariate Normal Distribution

Consider n random variables $X_1, \ldots, X_n$ and assume they are jointly normally distributed. Define the $n\times 1$ vectors $\mathbf{X} = (X_1, \ldots, X_n)'$, $\mathbf{x} = (x_1, \ldots, x_n)'$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)'$, and the $n \times n$ covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{pmatrix}.$$

Then $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ means that the random vector $\mathbf{X}$ has a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The pdf of the multivariate normal distribution can be compactly expressed as

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2\pi^{n/2}\det(\boldsymbol{\Sigma})^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \\ &= (2\pi)^{-n/2}\det(\boldsymbol{\Sigma})^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \end{aligned}$$

1.7 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

$$\underset{n\times 1}{\frac{\partial}{\partial \mathbf{x}}}\mathbf{x}'\mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{y} \\ \vdots \\ \frac{\partial}{\partial x_n}\mathbf{x}'\mathbf{y} \end{pmatrix} = \mathbf{y}, \tag{1.4}$$

$$\underset{n\times 1}{\frac{\partial}{\partial \mathbf{x}}}\mathbf{A}\mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1}(\mathbf{A}\mathbf{x})' \\ \vdots \\ \frac{\partial}{\partial x_n}(\mathbf{A}\mathbf{x})' \end{pmatrix} = \mathbf{A}, \tag{1.5}$$

$$\underset{n\times 1}{\frac{\partial}{\partial \mathbf{x}}}\mathbf{x}'\mathbf{A}\mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1}\mathbf{x}'\mathbf{A}\mathbf{x} \\ \vdots \\ \frac{\partial}{\partial x_n}\mathbf{x}'\mathbf{A}\mathbf{x} \end{pmatrix} = 2\mathbf{A}\mathbf{x}. \tag{1.6}$$

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 (1.4). Now

$$\mathbf{x}'\mathbf{y} = x_1 y_1 + x_2 y_2.$$

Then

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

Next, consider (1.5). Note that

$$\mathbf{A}\mathbf{x} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a x_1 + b x_2 \\ b x_1 + c x_2 \end{pmatrix}$$

and

$$(\mathbf{Ax})' = (ax_1 + bx_2, bx_1 + cx_2)$$

Then

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{Ax} = \begin{pmatrix} \frac{\partial}{\partial x_1}(ax_1 + bx_2, bx_1 + cx_2) \\ \frac{\partial}{\partial x_2}(ax_1 + bx_2, bx_1 + cx_2) \end{pmatrix} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} = \mathbf{A}$$

Finally, consider (1.6). We have

$$\mathbf{x}'\mathbf{Ax} = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = ax_1^2 + 2bx_1x_2 + cx_2^2.$$

Then

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}}\mathbf{x}'\mathbf{Ax} &= \begin{pmatrix} \frac{\partial}{\partial x_1}(ax_1^2 + 2bx_1x_2 + cx_2^2) \\ \frac{\partial}{\partial x_2}(ax_1^2 + 2bx_1x_2 + cx_2^2) \end{pmatrix} = \begin{pmatrix} 2ax_1 + 2bx_2 \\ 2bx_1 + 2cx_2 \end{pmatrix} \\ &= 2\begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2\mathbf{Ax}. \end{aligned}$$

1.8 Portfolio Math Using Matrix Algebra

Let R_i denote the return on asset $i = A, B, C$ and assume that R_A, R_B and R_C are jointly normally distributed with means, variances and covariances:

$$\mu_i = E[R_i],\ \sigma_i^2 = \text{var}(R_i),\ \text{cov}(R_i, R_j) = \sigma_{ij}.$$

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, $R_{p,x}$, is the random variable

$$R_{p,x} = x_A R_A + x_B R_B + x_C R_C. \tag{1.7}$$

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

$$\mu_{p,x} = E[R_{p,x}] = x_A\mu_A + x_B\mu_B + x_C\mu_C, \tag{1.8}$$

and the variance of the portfolio return is

$$\sigma_{p,x}^2 = \text{var}(R_{p,x}) = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + x_C^2\sigma_C^2 + 2x_Ax_B\sigma_{AB} + 2x_Ax_C\sigma_{AC} + 2x_Bx_C\sigma_{BC}. \tag{1.9}$$

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 (1.7) - (1.9) is cumbersome. We can greatly simplify the portfolio algebra using matrix notation.

Define the following 3×1 column vectors containing the asset returns and portfolio weights

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

The probability distribution of the random return vector $\mathbf{R}$ is simply the joint distribution of the elements of $\mathbf{R}$. Here all returns are jointly normally distributed and this joint distribution is completely characterized by the means, variances and covariances of the returns. We can easily express these values using matrix notation as follows. The 3×1 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 3×3 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 $\mathbf{\Sigma} = \mathbf{\Sigma}'$, where $\mathbf{\Sigma}'$ denotes the transpose of $\mathbf{\Sigma}$) 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 $\text{cov}(R_B, R_C) = \text{cov}(R_C, R_B)$.

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

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

The variance of the portfolio is

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

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 3×1 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

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

We often need to compute the covariance between the return on portfolio $\mathbf{x}$ and the return on portfolio $\mathbf{y}$, $\text{cov}(R_{p,x}, R_{p,y})$. This can be easily expressed using matrix algebra:

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

Notice that

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

which is quite a messy expression!

The global minimum variance portfolio $\mathbf{m}$ solves the constrained minimization problem

$$\min_{\mathbf{m}} \ \sigma_{p,m}^2 = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{m} \text{ s.t. } \mathbf{m}'\mathbf{1} = 1. \tag{1.10}$$

The Lagrangian function is

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

The first order conditions can be expressed in matrix notation as

$$\underset{(3\times1)}{\mathbf{0}} = \frac{\partial L(\mathbf{m}, \lambda)}{\partial \mathbf{m}} = \frac{\partial}{\partial \mathbf{m}}\mathbf{m}'\boldsymbol{\Sigma}\mathbf{m} + \frac{\partial}{\partial \mathbf{m}}\boldsymbol{\lambda}(\mathbf{m}'\mathbf{1} - 1) = 2 \cdot \boldsymbol{\Sigma}\mathbf{m} + \lambda \cdot \mathbf{1} \tag{1.11}$$

$$\underset{(1\times1)}{0} = \frac{\partial L(\mathbf{m}, \lambda)}{\partial \lambda} = \frac{\partial}{\partial \lambda}\mathbf{m}'\boldsymbol{\Sigma}\mathbf{m} + \frac{\partial}{\partial \lambda}\boldsymbol{\lambda}(\mathbf{m}'\mathbf{1} - 1) = \mathbf{m}'\mathbf{1} - 1 \tag{1.12}$$

1.9 Further Reading

A classic textbook on linear algebra is Strang (1980). Reviews of matrix algebra with applications in economics and finance are given in Chang (1984). Excellent treatments of portfolio theory using matrix algebra are given in Ingersol (1987), Huang and Litzenberger (1988) and Campbell, Lo and MacKinlay (1996).

Bibliography

[1] Campbell, J.Y., Lo, A.W., and MacKinlay, A.C. (1997). *The Econometrics of Financial Markets*. Priceton, New Jersey: Princeton University Press.

[2] Chiang, A. (1984). *Fundamental Methods of Mathematical Economics, Third Edition*. McGraw-Hill, New York.

[3] Huang, C.-F., and Litzenbeger, R.H. (1988). *Foundations for Financial Economics*. New York: North-Holland.

[4] Ingersoll, J.E. (1987). *Theory of Financial Decision Making*. Totowa, New Jersey: Rowman & Littlefield.

[5] Strang, G. (1980). *Linear Algebra and Its Applications*. Academic Press.