Matrix Overview

Paul J. Hewson

August 28, 2010

Chapter 1

Matrix manipulation

It is convenient to represent multivariate data by means of $n \times p$ matrix such as X. We could consider the `USArrests` data in this way. We follow the convention of using n to denote the number of rows of individuals who have been observed, and p to denote the number of columns (variables). We will formalise some some aspects from linear algebra that will be important in understanding multivariate analysis. These are very brief notes, there is a wealth of readable material on linear algebra as well as material specific for statistical applications such as Healy (2000) and Schott (1997). There is also an interesting presentation from a more geometric perspective in Wickens (1995) which supplements more algebraic presentations of matrix concepts.

1.1 Vectors

Consider a vector $\boldsymbol{x} \in \mathbb{R}^p$, by convention this is thought of as a column vector:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

A row vector such as $\begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix}$ will be denoted by $\boldsymbol{x}^T$.

A vector is a basic unit of numbers within **R** , but the **R** objects don't entirely conform to a formal mathematical definition (look at the way vecctor recycling works for example) and some caution is needed. The following instruction:

```
> x <- c(3.289, 4.700, 10.400)
```

assigns the values to the object `x` creating the following **R** vector:

$$x = \begin{pmatrix} 3.289 \\ 4.700 \\ 10.400 \end{pmatrix}$$

The default print method in **R** gives these in the most compact form:

```
> x
[1]  [1]  3.289  4.700 10.400
```

but forcing this into a matrix object with `as.matrix()` confirms its dimensionality:

```
> as.matrix(x)
       [,1]
[1,]  3.289
[2,]  4.700
[3,] 10.400
```

and taking the transpose of this vector using `t()` does produce a row vector as expected:

```
> t(x)
      [,1] [,2] [,3]
[1,] 3.289  4.7 10.4
```

1.1.1 Vector multiplication; the inner product

We first define the inner product of two vectors. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^p$ this gives a scalar:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{y} = \sum_{j=1}^{p} x_j y_j = \boldsymbol{y}^T \boldsymbol{x}$$

In other words, we find the product of corresponding elements of each vector (the product of the first element of the row vector and the first element of the column vector), and then find the sum of all these products:

$$\begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{pmatrix} = \underbrace{x_1 y_1 + x_2 y_2 + \ldots + x_n y_n}_{\text{One number; the sum of all the individual products}}$$

To give a simple example, with $\boldsymbol{x}^T = (4, 1, 3, 2)$ and $\boldsymbol{y} = \begin{pmatrix} 1 \\ -1 \\ 3 \\ 0 \end{pmatrix}$ we have:

$$\begin{pmatrix} 4 & 1 & 3 & 2 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \\ 3 \\ 0 \end{pmatrix} = \underbrace{4 \times 1 + 1 \times (-1) + 3 \times 3 + 2 \times 0} = 12$$

In **R** the inner product can be simply obtained using `%*%`, for example:

```
> x <- c(4, 1, 3, 2)
> y <- c(1, -1, 3, 0)
> t(x) %*% y
     [,1]
[1,]   12
```

which returns the answer as a scalar. Note that using `*` without the enclosing `%%` yields a vector of the same length of $\boldsymbol{x}$ and $\boldsymbol{y}$ where each element is the product of the corresponding elements of $\boldsymbol{x}$ and $\boldsymbol{y}$, and may do other unexpected things using vector recycling.

1.1.2 Outer product

Note that if $\boldsymbol{x}^T\boldsymbol{y}$ is the inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the *outer* product is given by $\boldsymbol{x}\boldsymbol{y}^T$. For vectors, it can be computed by `x %*% t(y)`; but as we will find later, outer product operations are defined for arrays of more than one dimension as `x %o% y` and `outer(x,y)`

1.1.3 Vector length

An important concept is the length of a vector, also known as the Euclidean norm or the modulus. It is based on a geometric idea and expresses the distance of a given vector from the origin:

$$|\boldsymbol{x}| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{1/2} = \left(\sum_{j=1}^{p} x_j^2 \right)^{1/2}$$

A *normalised* vector is one scaled to have unit length, for the vector $\boldsymbol{x}$ this can be found by taking $\frac{1}{|x|}\boldsymbol{x}$ which is trivial in **R** :

```
> z <- x / sqrt(t(x) %*% x)
> z
[1] 0.7302967 0.1825742 0.5477226 0.3651484
> t(z) %*% z ## check the length of the normalised vector
```

```
     [,1]
[1,]    1
```

1.1.4 Orthogonality

Two vectors $\mathbf{x}$ and $\mathbf{y}$, of order $k \times 1$ are orthogonal if $\boldsymbol{x}\boldsymbol{y} = 0$. Furthermore, if two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are orthogonal *and* of unit length, i.e. if $\boldsymbol{x}\boldsymbol{y} = 0$, $\boldsymbol{x}^T\boldsymbol{x} = 1$ and $\boldsymbol{y}^T\boldsymbol{y} = 1$ then they are orthonormal.

More formally, a set $\{\boldsymbol{e}_i\}$ of vectors in $\mathbb{R}^p$ is orthonormal if

$$\boldsymbol{e}_i^T\boldsymbol{e}_j = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

Where δ_{ij} is referred to as the Kronecker delta.

1.1.5 Cauchy-Schwartz Inequality

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq |\boldsymbol{x}|\,|\boldsymbol{y}|, \text{ for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}$$

with equality if and only if $\boldsymbol{x} = \lambda\boldsymbol{y}$ for some $\lambda \in \mathbb{R}$. Proof of this inequality is given in many multivariate textbooks such as Bilodeau and Brenner (1999). We won't use this result itself, but will actually consider the extended Cauchy-Scwartz inequality later.

1.1.6 Angle between vectors

The cosine of the angle between two vectors is given by:

$$\cos(\theta) = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{|\boldsymbol{x}|\,|\boldsymbol{y}|}$$

It can be conveniently calculated in **R** :

```
> cor(x,y)
```

1.2 Matrices

We now consider some basic properties of matrices, and consider some basic operations on them that will become essential as we progress. Consider the data matrix $\boldsymbol{X}$, containing the `USArrests` data, a 50×4 matrix, i.e. with $n = 50$ rows refering to States and $p = 4$ columns refering to the variables measuring different arrest rates. To indicate the order of this matrix it could be described fully as $\boldsymbol{X}_{50,4}$; this convention is followed in **R** as a call to `dim(USArrests)` will confirm. Each element in this matrix can be denoted by x_{ij} where i denotes the particular row (here state) and j the particular column (here arrest rate). Hence $x_{6\,3} = 38.7$.

In order to create a matrix in **R** the dimension has to be specified in the call to `matrix()`. It should be very carefully noted that the default is to fill a matrix by columns, as indicated here:

```
> mydata <- c(1,2,3,4,5,6)
> A <- matrix(mydata, 3,2)
> A
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

If this is not convenient, **R** can be persuaded to fill matrices by rows rather than by columns by including the argument `byrow = TRUE` in the call to `matrix`. It is also possible to coerce other objects (such as data frames) to a matrix using `as.matrix()` and `data.matrix()`; the former producing a character matrix if there are any non-numeric variables present, the latter coercing everything to a numeric format.

1.2.1 Transposing matrices

Transposing matrices simply involves turning the first column into the first row. A transposed matrix is denoted by a superscripted T, in other words $\mathbf{A}^\mathbf{T}$ is the transpose of $\mathbf{A}$.

$$If\ \mathbf{A} = \begin{pmatrix} 3 & 1 \\ 5 & 6 \\ 4 & 4 \end{pmatrix} \text{ then } \mathbf{A}^\mathbf{T} = \begin{pmatrix} 3 & 5 & 4 \\ 1 & 6 & 4 \end{pmatrix}$$

As with vectors, transposing matrices in **R** simply requires a call to `t()`, the dimensions can be checked with `dim()`.

```
> Atrans <- t(A)
> Atrans
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> dim(Atrans)
[1] 2 3
```

1.2.2 Some special matrices

Symmetric matrices

We mention a few "special" matrix forms that will be encountered. We firstly note that *symmetric* matrices are symmetric around the diagonal $i = j$. For matrix $\boldsymbol{A}$, it is symmetric whenever $a_{ij} = a_{ji}$. The correlation matrix and the variance-covariance matrix are the most common symmetric matrices we will encounter, we will look at them in more detail later, for now note that we can obtain the (symmetric) correlation matrix as follows:

```
> cor(USArrests)
             Murder  Assault   UrbanPop      Rape
Murder   1.00000000 0.8018733 0.06957262 0.5635788
Assault  0.80187331 1.0000000 0.25887170 0.6652412
UrbanPop 0.06957262 0.2588717 1.00000000 0.4113412
Rape     0.56357883 0.6652412 0.41134124 1.0000000
```

Diagonal Matrices

Given it's name, it is perhaps obvious that a diagonal matrix has elements on the diagonal (where $i = j$) and zero elsewhere (where $i \neq j$). For example, the matrix $\boldsymbol{A}$ given as follows:

$$\mathbf{A} = \begin{pmatrix} 13 & 0 & 0 \\ 0 & 27 & 0 \\ 0 & 0 & 16 \end{pmatrix}$$

is a diagonal matrix. To save paper and ink, $\boldsymbol{A}$ can also be written as:

$$\mathbf{A} = diag\begin{pmatrix} 13 & 27 & 16 \end{pmatrix}$$

It is worth noting that the `diag()` command in **R** , as shown below, lets you both *overwrite* the diagonal elements of matrix and *extract* the diagonal elements depending how it is used:

```
> mydataD <- c(13, 27, 16)
> B <- diag(mydataD)
> B
     [,1] [,2] [,3]
[1,]   13    0    0
[2,]    0   27    0
[3,]    0    0   16
> diag(B)
[1] 13 27 16
```

It is also worth noting that when "overwriting", the size of the matrix to be overwritten can be inferred from the dimensionality of diagonal.

Identity Matrix

One special diagonal matrix is the identity matrix, which has a value of 1 at each position on the diagonal and 0 elsewhere. Here, all we need to know is the size. So

$\mathbf{I}_4$ tells us that we have the following matrix:

$$\mathbf{I}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This can be created in a variety of ways in **R** , such as `I4 <- diag(rep(1,4))`

Ones

We also need to define a vector of ones; $\mathbf{1}_p$, a $p \times 1$ matrix containing only the value 1. There is no inbuilt function in **R** to create this vector, it is easily added:

```
> ones <- function(p){
  Ones <- matrix(1,p,1)
  return(Ones)
}
```

Zero matrix

Finally, $\mathbf{0}$ denotes the zero matrix, a matrix of zeros. Unlike the previously mentioned matrices this matrix can be any shape you want. So, for example:

$$\mathbf{0}_{2\ 3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

1.2.3 Equality and addition

A little more care is needed in defining basic mathematical operations on matrices. Considering the two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we consider their equality $\boldsymbol{A} = \boldsymbol{B}$ if any only if:

- $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same size, and
- the ijth element of $\boldsymbol{A}$ is equal to the ijth element of $\boldsymbol{A}$ for all $1 \leq i \leq r$ and $1 \leq j \leq n$

A consequence of this is that the following two matrices are equal:

$$\begin{bmatrix} 138.8149 & 187.52 & 394.86 \\ 187.5200 & 267.00 & 559.00 \\ 394.8600 & 559.00 & 1200.00 \end{bmatrix} = \begin{bmatrix} 138.8149 & 187.52 & 394.86 \\ 187.5200 & 267.00 & 559.00 \\ 394.8600 & 559.00 & 1200.00 \end{bmatrix}$$

(which seems like an obvious and fussy thing to say) *but* the following two zero matrices are not equal:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Adding and subtracting are fairly straightforward. Provided $\boldsymbol{A}$ and $\boldsymbol{A}$ have the same size, $\boldsymbol{A} + \boldsymbol{B}$ and $\boldsymbol{A} - \boldsymbol{B}$ are defined by each of these operations being carried out on individual elements of the matrix. For example:

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 3 \\ -1 & -2 & -3 \end{pmatrix} = \begin{pmatrix} 1+0 & 3+2 & 5+3 \\ 2+-1 & 4+-2 & 6+-3 \end{pmatrix} = \begin{pmatrix} 1 & 5 & 8 \\ 1 & 2 & 3 \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix} - \begin{pmatrix} 0 & 2 & 3 \\ -1 & -2 & -3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 3 & 6 & 9 \end{pmatrix}$$

Addition and subtraction are straightforward enough in **R** :

```
> A <- matrix(c(1,2,3,4,5,6),2,3)
> A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> B <- matrix(c(0,-1,2,-2,3,-3),2,3)
> B
     [,1] [,2] [,3]
[1,]    0    2    3
[2,]   -1   -2   -3
> A + B
     [,1] [,2] [,3]
[1,]    1    5    8
[2,]    1    2    3
> A - B
     [,1] [,2] [,3]
[1,]    1    1    2
[2,]    3    6    9
```

Matrix addition follows all the normal arithmetic rules, i.e.

Commutative law $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$
Associative law $\mathbf{A} + (\mathbf{B} + \mathbf{C}) = (\mathbf{A} + \mathbf{B}) + \mathbf{C}$

Matrix multiplication however follows vector multiplication and therefore does not follow the same rules as basic multiplication.

1.2.4 Multiplication

A scalar is a matrix with just one row and one column, i.e. a single number. In other words, 0.4 could be a scalar or a 1×1 matrix. It's worth re-capping that multiplication by a scalar is easy enough, we just multiply every element in the matrix by the scalar.

So if $\mathbf{k} = 0.4$, and

$$\mathbf{A} = \begin{pmatrix} 1 & 5 & 8 \\ 1 & 2 & 3 \end{pmatrix}$$

we can calculate $\mathbf{kA}$ as:

$$\mathbf{kA} = 0.4 \times \begin{pmatrix} 1 & 5 & 8 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 0.4 & 2 & 3.2 \\ 0.4 & 0.8 & 1.6 \end{pmatrix}$$

When multiplying two matrices, it should be noted first that they must be conformable. The number of columns in the first matrix must match the number of rows in the second. As matrix multiplication has been defined, the result will be a matrix with as many rows as the first matrix and as many columns as the second. For example, with our vectors above in section 1.1.1 , we had $\boldsymbol{A}_{\mathbf{1\,4}} \times \boldsymbol{B}_{\mathbf{4\,1}} = \boldsymbol{C}_{\mathbf{1\,1}}$. More generally multiplication proceeds with matrix size as follows: $\boldsymbol{A}_{\boldsymbol{m\,n}} \times \boldsymbol{B}_{\boldsymbol{n\,p}} = \boldsymbol{C}_{\boldsymbol{m\,p}}$.

It *may* help to think about the vector operations and extend them to matrices. There are other ways of thinking about matrix multiplication, most multivariate text books have an appendix on matrix algebra and there are vast tomes available covering introductory linear algebra. However, one explanation of matrix multiplication is given here. We want to find $\boldsymbol{A} \times \boldsymbol{B}$ where

$$\mathbf{A} = \begin{pmatrix} 1 & 5 \\ 1 & 2 \\ 3 & 8 \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} 1 & 4 \\ 3 & 2 \end{pmatrix}$$

If $\mathbf{A}$ is of size $m \times n$ it could be considered as consisting of a row of vectors $\boldsymbol{a}_{\mathbf{1}}^{\boldsymbol{T}}, \boldsymbol{a}_{\mathbf{1}}^{\boldsymbol{T}}, \ldots, \boldsymbol{a}_{\boldsymbol{m}}^{\boldsymbol{T}}$, which in this case corresponds to $\boldsymbol{a}_{\mathbf{1}}^{\boldsymbol{T}} = (1, 5), \boldsymbol{a}_{\mathbf{2}}^{\boldsymbol{T}} = (1, 2)$ and $\boldsymbol{a}_{\mathbf{3}}^{\boldsymbol{T}} = (3, 8)$. Likewise, we can consider $\mathbf{B}$ consisting of $\boldsymbol{b}_{\mathbf{1}} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$ and $\boldsymbol{b}_{\mathbf{1}} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$. In other words, we are trying to multiply together:

$$\mathbf{A} = \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix} \text{and } \mathbf{B} = \begin{pmatrix} b_1 & b_2 \end{pmatrix}$$

We can define the multiplication operation for matrices generally as:

$$\mathbf{AB} = \begin{pmatrix} a_1^T \\ a_2^T \\ \ldots \\ a_m^T \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \ldots & b_p \end{pmatrix} = \begin{pmatrix} a_1^T b_1 & a_1^T b_2 & \ldots & a_1^T b_p \\ a_2^T b_1 & a_2^T b_2 & \ldots & a_1^T b_p \\ \vdots & \vdots & & \vdots \\ a_3^T b_1 & a_3^T b_2 & \ldots & a_m^T b_p \end{pmatrix}$$

In other words, we need to multiply row i of $\boldsymbol{A}$ by column j of $\boldsymbol{B}$ to give element ij of the result. For example, note that $\boldsymbol{a}_{\mathbf{1}}^T \boldsymbol{b}_{\mathbf{1}} = \begin{pmatrix} 1 & 5 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = 1 \times 1 + 5 \times 3 = 16$. Carrying out this operation on our matrices above gives:

$$\mathbf{AB} = \begin{pmatrix} 1 & 5 \\ 1 & 2 \\ 3 & 8 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} 16 & 14 \\ 7 & 8 \\ 27 & 28 \end{pmatrix}$$

In **R** , we only need to use the `%*%` operator to ensure we are getting matrix multiplication:

```
> A <- matrix(c(1,1,3,5,2,8),3,2)
> A
     [,1] [,2]
[1,]    1    5
[2,]    1    2
[3,]    3    8
> B <- matrix(c(1,3,4,2),2,2)
> B
     [,1] [,2]
[1,]    1    4
[2,]    3    2
> A %*% B
     [,1] [,2]
[1,]   16   14
[2,]    7    8
[3,]   27   28
```

Note that you can't multiply non-conformable matrices; this is one place in **R** where you get a clearly informative error message:

```
> B %*% A
Error in B %*% A : non-conformable arguments
```

It is particularly important to use the correct *matrix multiplication* argument. Depending on the matrices you are working with (if they both have the same dimensions), using the usual * multiplication operator will give you the Hadamard product, the element by element product of the two matrices which is rarely what you want:

```
> C <- matrix(c(1,1,3,5),2,2)
> C %*% B ## correct call for matrix multiplication
     [,1] [,2]
```

```
[1,]   10   10
[2,]   16   14
> C * B ## Hadamard Product!!!
     [,1] [,2]
[1,]    1   12
[2,]    3   10
```

We saw earlier that matrix addition was commutative and associative. But as you can imagine, given the need for comformability some differences may be anticipated between conventional multiplication and matrix multiplication. Generally speaking, matrix multiplication is not commutative (you may like to think of exceptions):

$$\begin{array}{rl} \text{(non-commutative)} & \mathbf{A} \times \mathbf{B} \neq \mathbf{B} \times \mathbf{A} \\ \text{Associative law} & \mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = (\mathbf{A} \times \mathbf{B}) \times \mathbf{C} \end{array}$$

And the distributive laws of multiplication over addition apply as much to matrix as conventional multiplication:

$$\mathbf{A} \times (\mathbf{B} + \mathbf{C}) = (\mathbf{A} \times \mathbf{B}) + (\mathbf{A} \times \mathbf{C})$$
$$(\mathbf{A} + \mathbf{B}) \times \mathbf{C} = (\mathbf{A} \times \mathbf{C}) + (\mathbf{B} \times \mathbf{C})$$

But there are a few pitfalls if we start working with transposes. Whilst

$$(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$$

note that:

$$(\mathbf{A} \times \mathbf{B})^T = \mathbf{B}^T \times \mathbf{A}^T$$

Trace of a matrix

The trace of a matrix is the quite simply the sum of its diagonal elements. This is an interesting concept in many ways, but it turns out in one specific context, when applied to the covariance matrix, this has an interpretation as the total sample variance. There is no inbuilt function in **R** to calculate this value, you need to use `sum(diag(X))`

Note that if you have two conformable matrices $\boldsymbol{A}$ e.g. $\begin{pmatrix} 2 & 5 \\ 0 & 7 \\ 4 & 3 \end{pmatrix}$ and $\boldsymbol{B}$ e.g. $\begin{pmatrix} 4 & 2 & 1 \\ 6 & 3 & 2 \end{pmatrix}$, $trace(\boldsymbol{AB}) = trace(\boldsymbol{BA})$

1.3 Crossproduct matrix

Given the data matrix $\boldsymbol{X}$, the crossproduct, sometimes more fully referred to as the "sum of squares and crossproducts" matrix is given by $\boldsymbol{X}^T\boldsymbol{X}$. The diagonals of this matrix are clearly the sum of squares of each column. Whilst this can be computed in **R** using `X %*% t(X)` there are some computational advantages in using the dedicated function `crossprod(X)` For example, coercing the `USArrests` data to a matrix we can obtain the sum of squares and crossproducts matrix for these data as follows:

```
B <- crossprod(as.matrix(USArrests))
```

So if $\boldsymbol{X}$ is the `USArrests` data,

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{bmatrix} 3962.20 & 80756.00 & 25736.20 & 9394.32 \\ 80756.00 & 1798262.00 & 574882.00 & 206723.00 \\ 25736.20 & 574882.00 & 225041.00 & 72309.90 \\ 9394.32 & 206723.00 & 72309.90 & 26838.62 \end{bmatrix}$$

If we define some sample estimators as follows:

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{1} \tag{1.1}$$

So for example we can find the sample mean for the USArrests data as:

```
> n <- dim(USArrests)[1] ## extract n; here 50
> one <- ones(n)
> 1/n * t(USArrests) %*% one
> mean(USArrests) ## check results against in-built function
```

We can use matrix algebra to obtain an unbiased estimate of the sample covariance matrix $\boldsymbol{S}$ as follows:

$$\begin{aligned} \boldsymbol{S} &= \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \\ &= \sum_{i=1}^{n}\boldsymbol{x}_i\sum_{i=1}^{n}\boldsymbol{x}_i^T - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T \\ &= \frac{1}{n-1}\boldsymbol{X}^T\boldsymbol{X} - \bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^T \\ &= \frac{1}{n-1}\left(\boldsymbol{X}^T\boldsymbol{X} - \frac{1}{n}\boldsymbol{X}^T\boldsymbol{1}\boldsymbol{1}^T\boldsymbol{X}\right) \end{aligned}$$

From this, we can define the centering matrix $\boldsymbol{H}$:

$$\boldsymbol{H} = \boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^T$$

and so arrive at an alternative expression for $\boldsymbol{S}$ using this centering matrix:

$$\boldsymbol{S} = \frac{1}{n-1}\boldsymbol{X}^T\boldsymbol{H}\boldsymbol{X} \tag{1.2}$$

Idempotent matrices

It may be noted that $\boldsymbol{H}$ is idempotent, i.e. $\boldsymbol{H} = \boldsymbol{H}^T$ and $\boldsymbol{H} = \boldsymbol{H}^2$.

In calculating $\boldsymbol{H}$ in **R** it might be clearer to set the steps out in a function:

```
centering <- function(n){
 I.mat <- diag(rep(1, n))
 Right.mat <- 1/n * ones(n) %*% t(ones(n))
 H.mat <- I.mat - Right.mat
 return(H.mat)
}
```

And our matrix method for finding an estimate of the sample covariance using this centering procedure can also be set out in a function:

```
S.mat <- function(X, H){
n <- dim(X)[1] ## number of rows
H.mat <- centering(n)
S <- 1/(n-1) * t(X) %*% H.mat %*% X
return(S)
}
```

So, to estimate the sample covariance with this function we need to make sure our data are in the form of matrix. We also compare the results with the inbuilt function `cov()`:

```
X <- as.matrix(USArrests)
S.mat(X)
cov(USArrests)
```

It may be worth clarifying the information contained in the matrix we have just obtained. The covariance matrix (more fully referred to as the variance-covariance matrix) contains information on the variance of each of the variables as well as information on pairwise covariance. We will formalise our understanding of estimators later, but for now note that it could be considered as an estimate of:

$$\boldsymbol{\Sigma} = \boldsymbol{V}\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} var(X_1) & cov(X_1, X_2) & cov(X_1, X_3) & cov(X_1, X_4) \\ cov(X_2, X_1) & var(X_2) & cov(X_2, X_3) & cov(X_2, X_4) \\ cov(X_3, X_1) & cov(X_3, X_2) & var(X_3) & cov(X_3, X_4) \\ cov(X_4, X_1) & cov(X_4, X_2) & cov(X_4, X_3) & var(X_4) \end{pmatrix}$$

For the US Arrests data, as we have seen:

$$\boldsymbol{S} = \begin{pmatrix} 18.97 & 291.06 & 4.39 & 22.99 \\ 291.06 & 6945.17 & 312.28 & 519.27 \\ 4.39 & 312.28 & 209.52 & 55.77 \\ 22.99 & 519.27 & 55.77 & 87.73 \end{pmatrix}$$

1.3.1 Powers of matrices

We set out some definitions of matrix powers as they will come in useful later.For all matrices, we define $\mathbf{A^0} = \mathbf{I}$, the identity matrix and $\mathbf{A^1} = \mathbf{A}$. We will next define $\mathbf{A^2} = \mathbf{AA}$ (if you think about it a bit you could see that $\mathbf{A}$ must be a square matrix, otherwise we couldn't carry out this multiplication). Using these definitions for matrix powers means that all the normal power arithmetic applies. For example, $\mathbf{A^m} \times \mathbf{A^n} = \mathbf{A^n} \times \mathbf{A^m} = \mathbf{A^{m+n}}$. If you look closely, you can also see that the powers of a matrix are commutative which means that we can do fairly standard algebraic factorisation. For example:

$$\mathbf{I} - \mathbf{A^2} = (\mathbf{I} + \mathbf{A})(\mathbf{I} - \mathbf{A})$$

which is a result we can use later.

1.3.2 Determinants

The determinant of a *square* $p \times p$ matrix $\mathbf{A}$ is denoted as $|A|$. Finding the determinant of a 2×2 matrix is easy:

$$|A| = det\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

For matrices of order > 2, partitioning the matrix into "minors" and "cofactors" is necessary. Consider the following 3×3 matrix.

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{13} \end{pmatrix}$$

Any element a_{ij} of this matrix has a corresponding square matrix formed by eliminating the row (i) and column (j) containing a_{ij}. So if we were considering a_{11}, we would be interested in the square matrix $\boldsymbol{A_{-11}} = \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{13} \end{pmatrix}$. The determinant of this reduced matrix, $|\boldsymbol{A_{-11}}|$ is called the minor of a_{11}, and the product $c_{ij} = (-1)^{i+j}|A_{-ij}| = -1^{1+1}|A_{-11}| = |A_{11}|$ is called the cofactor of a_{11}. The determinant of $\boldsymbol{A}$ can be expressed as the sum of minors and cofactors of any row or column of $\boldsymbol{A}$.

Thus:

$$|\boldsymbol{A}| = \Sigma_{j=1}^{p} a_{ij} c_{ij}$$

and as can be seen, this can get terribly recursive if you're working by hand! Working an example through:

$$\text{If} \boldsymbol{A} = \begin{pmatrix} 3 & 4 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{pmatrix}$$

Then $|\boldsymbol{A}|= a_{i1}c_{i1} + a_{i2}c_{i2} + a_{i3}c_{i3}$. If $i = 1$ then:

$$\begin{aligned} c_{11} &= (-1)^{1+1} \begin{vmatrix} 2 & 3 \\ 7 & 9 \end{vmatrix} = (18 - 21) = -3 \\ c_{11} &= (-1)^{1+2} \begin{vmatrix} 1 & 3 \\ 5 & 9 \end{vmatrix} = -(9 - 15) = 6 \\ c_{11} &= (-1)^{1+1} \begin{vmatrix} 1 & 2 \\ 5 & 7 \end{vmatrix} = (7 - 10) = -3 \end{aligned}$$

So $|\boldsymbol{A}| = 3(-3) + 4(6) + 6(-3) = -3$.

In **R** , `det()` tries to find the determinant of a matrix.

```
> D <- matrix(c(5,3,9,6),2,2)
> D
     [,1] [,2]
[1,]    5    9
[2,]    3    6
> det(D)
[1] 3
> E <- matrix(c(1,2,3,6),2,2)
> E
     [,1] [,2]
[1,]    1    3
[2,]    2    6
> det(E)
[1] 0
```

Some useful properties of determinants:

- The determinant of a diagonal matrix (or a triangular matrix for that matter) is the product of the diagonal elements. (Why?).

- For any scalar k, $|k\boldsymbol{A}| = k^n|\boldsymbol{A}|$, where $\boldsymbol{A}$ has size $n \times n$.
- If two rows or columns of a matrix are interchanged, the sign of the determinant changes.
- If two rows or columns are equal or proportional (see material on rank later), the determinant is zero.
- The determinant is unchanged by adding a multiple of some column (row) to any other column (row).
- If all the elements or a column / row are zero then the determinant is zero.
- If two $n \times n$ matrices are denoted by $\boldsymbol{A}$ and $\boldsymbol{B}$, then $|\boldsymbol{AB}| = |\boldsymbol{A}|.|\boldsymbol{B}|$.

The determinant of a variance-covariance has a rather challenging interpretation as the generalised variance.

1.3.3 Rank of a matrix

Rank denotes the number of linearly independent rows or columns. For example:

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 5 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

This matrix has dimension 3×3, but only has rank 2. The second column $\boldsymbol{a_2}$ can be found from the other two columns as $\boldsymbol{a_2} = 2\boldsymbol{a_1} - \boldsymbol{a_3}$.

If all the rows and columns of a square matrix $\mathbf{A}$ are linearly independent it is said to be of full rank and non-singular.

If $\mathbf{A}$ is singular, then $|A| = 0$.

1.4 Matrix inversion

If $\mathbf{A}$ is a non-singular $p \times p$ matrix, then there is a unique matrix $\mathbf{B}$ such that $\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}$, where $\mathbf{I}$ is the identity matrix given earlier. In this case, $\mathbf{B}$ is the inverse of $\mathbf{A}$, and denoted $\boldsymbol{A^{-1}}$.

Inversion is quite straightforward for a 2×2 matrix.

$$\text{If } \boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \text{ then } \boldsymbol{A}^{-1} = \frac{1}{|\boldsymbol{A}|}\begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

More generally for a matrix of order $n \times n$, the (j,k)th entry of $\boldsymbol{A^{-1}}$ is given by:

$$\left[\frac{|\boldsymbol{A}_{-jk}|}{|\boldsymbol{A}|}\right]^{(-1)^{j+k}},$$

where $\boldsymbol{A}_{-jk}$ is the matrix formed by deleting the jth row and kth column of $\boldsymbol{A}$. Note that a singular matrix has no inverse since its determinant is 0.

In **R** , we use `solve()` to invert a matrix (or solve a system of equations if you have a second matrix in the function call, if we don't specify a second matrix R assumes we want to solve against the identity matrix, which mean finding the inverse).

```
> D <- matrix(c(5,3,9,6),2,2)
> solve(D)
     [,1]      [,2]
[1,]    2 -3.000000
[2,]   -1  1.666667
```

Some properties of inverses:

- The inverse of a symmetric matrix is also symmetric.
- The inverse of the transpose of $\boldsymbol{A}$ is the transpose of $\boldsymbol{A}^{-1}$.
- The inverse of the product of several square matrices is a little more subtle: $(\boldsymbol{ABC})^{-1} = \boldsymbol{C}^{-1}\boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$. If c is a non-zero scalar then $(c\boldsymbol{A})^{-1} = c^{-1}\boldsymbol{A}^{-1}$.
- The inverse of a diagonal matrix is really easy - the reciprocals of the original elements.

1.5 Eigen values and eigen vectors

These decompositions will form the core of at least half our multivariate methods (although we need to mention at some point that we actually tend to use the singular value decomposition as a means of getting to these values). If $\boldsymbol{A}$ is a square $p \times p$ matrix, the eigenvalues (latent roots, characteristic roots) are the roots of the equation:

$$|\boldsymbol{A} - \lambda \boldsymbol{I}| = \boldsymbol{0}$$

This (characteristic) equation is a polynomial of degree p in λ. The roots, the eigenvalues of $\boldsymbol{A}$ are denoted by $\lambda_1, \lambda_2, \ldots, \lambda_p$. For each eigen value λ_i there is a corresponding eigen vector $\boldsymbol{e}_i$ which can be found by solving:

$$(\boldsymbol{A} - \lambda_i \boldsymbol{I})\boldsymbol{e_i} = \boldsymbol{0}$$

There are many solutions for $\boldsymbol{e_i}$. For our (statistical) purposes, we usually set it to have length 1, i.e. we obtain a normalised eigenvector for λ_i by $\boldsymbol{a_i} = \frac{e_i}{\sqrt{{e_i}^T e_i}}$

We pause to mention a couple of results that will be explored in much more detail later:

(a) $trace(\boldsymbol{A}) = \Sigma_{i=1}^{p} \lambda_i$

(b) $|\boldsymbol{A}| = \prod_{i=1}^{p} \lambda_i$

Also, if $\boldsymbol{A}$ is symmetric:

(c) The normalised eigenvectors corresponding to unequal eigenvalues are orthonormal (this is a bit of circular definition, if the eigenvalues are equal the corresponding eigenvectors are not unique, and one "fix" is to choose orthonormal eigenvectors).

(d) Correlation and covariance matrices: are symmetric positive definite (or semi-definite). If such a matrix is of full rank p then all the eigen values are positive. If the matrix is of rank $m < p$ then there will be m positive eigenvalues and $p - m$ zero eigenvalues.

We will look at the `eigen()` function in **R** to carry out these decompositions later.

1.6 Singular Value Decomposition

To be added.

1.7 Extended Cauchy-Schwarz Inequality

We met the rather amazing Cauchy Schwartz inequality earlier in section 1.1.5. Beautiful as this result may be, we actually need to use the *extended* Cauchy Schwartz inequality. For any non-zero vectors $\boldsymbol{x} \in \mathbb{R}$ and $\boldsymbol{y} \in \mathbb{R}$, with any positive definite $p \times p$ matrix $\boldsymbol{S}$:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2 \leq (\boldsymbol{x}^T \boldsymbol{S} \boldsymbol{x})(\boldsymbol{y}^T \boldsymbol{S}^{-1} \boldsymbol{y}), \text{ for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}$$

with equality if and only if $\boldsymbol{x} = \lambda \boldsymbol{S} \boldsymbol{y}$ for some $\lambda \in \mathbb{R}$. Proofs are available for this result (Flury, 1997, page 291). We will use this result when developing methods for discriminant analysis.

1.8 Partitioning

Finally, note that we can partition a large matrix into smaller ones:

$$\left(\begin{array}{cc|c} 2 & 5 & 4 \\ \hline 0 & 7 & 8 \\ 4 & 3 & 4 \end{array}\right)$$

So we could work with submatrices such as $\begin{pmatrix} 0 & 7 \\ 4 & 3 \end{pmatrix}$.

e.g. If $\boldsymbol{X}$ was partitioned as $\begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{pmatrix}$ and $\begin{pmatrix} \boldsymbol{Y}_1 & \boldsymbol{Y}_2 & \boldsymbol{Y}_3 \end{pmatrix}$ then:

$$\boldsymbol{XY} = \begin{pmatrix} \boldsymbol{X}_1\boldsymbol{Y}_1 & \boldsymbol{X}_1\boldsymbol{Y}_2 & \boldsymbol{X}_1\boldsymbol{Y}_3 \\ \boldsymbol{X}_2\boldsymbol{Y}_1 & \boldsymbol{X}_2\boldsymbol{Y}_2 & \boldsymbol{X}_2\boldsymbol{Y}_3 \end{pmatrix}$$

1.9 Exercises

1. Which of the following are orthogonal to each other:

$$\boldsymbol{x} = \begin{pmatrix} 1 \\ -2 \\ 3 \\ -4 \end{pmatrix} \boldsymbol{y} = \begin{pmatrix} 6 \\ 7 \\ 1 \\ -2 \end{pmatrix} \boldsymbol{z} = \begin{pmatrix} 5 \\ -4 \\ 5 \\ 7 \end{pmatrix}$$

   Normalise each of the two orthogonal vectors.

2. Find vectors which are orthogonal to:

$$\boldsymbol{u} = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \boldsymbol{v} = \begin{pmatrix} 2 \\ 4 \\ -1 \\ 2 \end{pmatrix}$$

3. Find vectors which are orthonormal to:

$$\boldsymbol{x} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \boldsymbol{y} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{5}{6} \end{pmatrix}$$

4. What are the determinants of:

$$(a) \begin{pmatrix} 1 & 3 \\ 6 & 4 \end{pmatrix} (b) \begin{pmatrix} 3 & 1 & 6 \\ 7 & 4 & 5 \\ 2 & -7 & 1 \end{pmatrix}$$

5. Invert the following matrices:

$$(a) \begin{pmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{pmatrix} (b) \begin{pmatrix} 2 & 3 \\ 1 & 5 \end{pmatrix} (c) \begin{pmatrix} 3 & 2 & -1 \\ 1 & 4 & 7 \\ 0 & 4 & 2 \end{pmatrix} (d) \begin{pmatrix} 1 & 1 & 1 \\ 2 & 5 & -1 \\ 3 & 1 & -1 \end{pmatrix}$$

6. Find eigenvalues and corresponding eigen vectors for the following matrices:

$$\boldsymbol{a} = \begin{pmatrix} 1 & 4 \\ 2 & 3 \end{pmatrix} \boldsymbol{b} = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} \boldsymbol{c} = \begin{pmatrix} 2 & -2 \\ -2 & 5 \end{pmatrix} \boldsymbol{d} = \begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix}$$

$$\boldsymbol{e} = \begin{pmatrix} 1 & 4 & 0 \\ 4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{f} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{g} = \begin{pmatrix} 13 & -4 & 2 \\ -4 & 13 & -2 \\ 2 & -2 & 10 \end{pmatrix}$$

7. Convert the following covariance matrix (you've seen it earlier) to a correlation matrix, calculate the eigenvalues and eigenvectors and verify that the eigen vectors are orthogonal.

$$g = \begin{pmatrix} 13 & -4 & 2 \\ -4 & 13 & -2 \\ 2 & -2 & 10 \end{pmatrix}$$

Bibliography

Bilodeau, M. and D. Brenner (1999). *Theory of Multivariate Statistics*. New York: Springer.

Flury, B. (1997). *A First Course in Multivariate Statistics*. New York: Springer.

Healy, M. (2000). *Matrices for Statistics* (2nd ed.). Oxford: Clarendon Press.

Schott, J. R. (Ed.) (1997). *Matrix Analysis for Statistics*. New York: Wiley.

Wickens, T. D. (1995). *The Geometry of Multivariate Statistics*. New Jersey: Lawrence Erlbaum Associates.