Matrix Calculations in R

updated 8 Feb 2004

R can be used to perform matrix multiplication and inversion. The syntax is a little odd, but straightforward. In the notes below, **>** indicates the **R** prompt, [1] the output from **R**

Defining Matrices

For starters, **R** is funny in that it works with column vectors. **R** starts with a list of elements and translates this into a matrix by filling up columns. The basic **R** command to define a matrix requires a list of elements (**`c(.,.,., .,)`**) and the number of rows **`nrow`** in the matrix. Consider the matrix

$$\mathbf{C} = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

To enter this matrix in **R**, we first have to write this as a single list, going down each column, i.e., **`c(1,2,3,4,5,6,7,8,9)`**. To use **R** to set the variable **`C`** equal to the matrix $\mathbf{C}$, we would use

```
> C <- matrix(c(1,2,3,4,5,6,7,8,9),nrow=3)
```

R uses the **`nrow`** command to set the dimension of the matrix. For example, if we entered

```
> C <- matrix(c(1,2,3,4,5,6,7,8,9),nrow=1)
```

This sets **`C`** equal to the matrix with a single row

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix}$$

By typing **`C`** and hitting return, **R** displays the matrix $\mathbf{C}$.

Conversely, you can instruct **R** to enter rows first by adding the command **`byrow=T`**, which enters the elements of the list as rows (the default is setting this option to false, entering this as columns). Thus entering

```
> D <- matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,byrow=T)
```

returns the matrix

$$\mathbf{D} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Individual elements can be extracted from a matrix **`C`** by using command **`C[i,j]`**, which extracts the element in the ith row and jth column of **`C`**.

Matrix Transposition, `t(C)`

There are two ways to compute a transpose in **C**. The simplest is to use the command **`t(C)`** to obtain the transpose of the matrix **C**. One can also compute the transpose when entering a matrix by using the **`byrow=T`** command.

Example 1: Using the **R** commands

```
> E <- matrix(c(1,2,3,4,5,6),nrow=2)
> F <- matrix(c(1,2,3,4,5,6),nrow=3,byrow=T)
```

Defines the matrices **E** and **F** as

$$\mathbf{E} = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

Note the $\mathbf{E}^T = \mathbf{F}$. Likewise, one could also use

```
> F <- t(E)
```

Matrix Multiplication: %*%

To multiply two matrices, **R** uses the command %*%. For example, using the matrices **C** and **D** above, the matrix product **CD** is computed in **R** by the command

```
> C%*%D

       [,1]  [,2]  [,3]
  [1,]  66    78    90
  [2,]  78    93    108
  [3,]  90    108   126
```

Conversely, the matrix product **DC** is given by

```
> D%*%C

       [,1]  [,2]  [,3]
  [1,]  14    32    50
  [2,]  32    77    122
  [3,]  50    122   194
```

Example 2: Consider the vector

$$\mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Use **R** to compute the inner product $\mathbf{b}^T\mathbf{b}$ and the outer product $\mathbf{b}\mathbf{b}^T$.

```
> b <- matrix(c(1,2,3),nrow=3)
> bt <- matrix(c(1,2,3),nrow=1)
> bt%*%b

      [,1]
[1,]  14
 > b%*%bt

      [,1]  [,2]  [,3]
[1,]  1     2     3
[2,]  2     4     6
[3,]  3     6     9
```

The Inverse of a Matrix

The inverse of $\mathbf{A}$ is obtained using the solve command, with $\mathbf{A}^{-1}$ computed by `solve(A)`.

Example 3: Consider the following system of equations

$$3x_1 + 4x_2 = 4$$
$$x_1 + 6x_2 = 2$$

In matrix from, this becomes $\mathbf{Ax{=}y}$ where

$$\mathbf{A} = \begin{pmatrix} 3 & 4 \\ 1 & 6 \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

Hence, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$,or in R

```
> A <- matrix(c(3,1,4,6),nrow=2)
> y <- matrix(c(4,2),nrow=2)
> x<- solve(X)%*%y
> x
```

R returns

```
        [,1]
[1,]   1.1428571
[2,]   0.1428571
```

We can check this by looking at the first equation, $3x_1 + 4x_2 = 4$

```
> 3*x[1,1]+4*x[2,1]
```

R returns

[1] 4.

Eigenvalues, vectors of a Matrix

The command **eigen(X)** returns the eigenvalues and vectors of for the square matrix **X**.

Example 4: Suppose we are still in **R** with **A** as in Example 3.

```
> eigen(A)
```

returns

```
$values
[1] 7 2
$vectors

        [,1]          [,2]
[1,]   -0.8246211    -0.9701425
[2,]   -0.8246211    0.2425356
```
