# LAB100

## Week 13: Matrices

Your aim for this workshop is to learn how to construct matrices and to extract information from them. Upon completing this workshop you should be able to:

- Use the `matrix` command
- Combine vectors to construct a matrix
- Subset a matrix to obtain a particular value, vector or sub-matrix
- Construct a matrix using a `for` loop
- Use R to find the determinant and inverse of a square matrix

Remember that you need to change the working directory and open a new R script at the start of each workshop.

## 1 Constructing a Matrix

Matrices can be constructed in RStudio using the `matrix` command:

```
matrix( data, nrow, ncol, byrow )
```

The input arguments for this command are:

| Argument | Description |
|---|---|
| `data` | A vector containing all of the values to be entered into the matrix. |
| `nrow` | The number of rows that the matrix should have. |
| `ncol` | The number of columns that the matrix should have. |
| `byrow` | A logical argument denoting whether the contents of `data` should be inserted by row (`TRUE`) or by column (`FALSE` - default) |

As an illustration, lets create the three matrices below.

$$A = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 9 & -13 \\ 20 & 5 & -6 \end{bmatrix} \qquad C = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix}$$

```
A <- matrix( 1:9, nrow=3, ncol=3, byrow=FALSE )
B <- matrix( c(1, 9, -13, 20, 5, -6), nrow=2, ncol=3, byrow=TRUE )
C <- matrix( rep(2,6), nrow=3, ncol=2, byrow=FALSE )
```

Note that the length of the data vector must be equivalent to the number of elements in the matrix, i.e. for a `nrow`×`ncol` matrix, the length of the data vector must be `nrow*ncol`. If this is not the case, then RStudio would will cycle through the data vector until a value has been entered into each element of the matrix. Furthermore, if the number of matrix elements is not a multiple of the data vector's length, then a warning message is produced in the console. Try the following example to understand how the data is entered into matrices

```
a <- 1:5
matrix(a, ncol=5, nrow=2, byrow=FALSE)
matrix(a, ncol=2, nrow=2, byrow=FALSE)
matrix(a, ncol=3, nrow=4, byrow=FALSE)
```

# 2 Combining Vectors

Another method for creating matrices is to combine a number of vectors, either by row or by column. Consider the following matrix that has been divided according to its columns:

$$D = \left[\begin{array}{c:c} 7 & 3 \\ 2 & 5 \\ 6 & 8 \\ 9 & 0 \end{array}\right]$$

We can easily construct two vectors using the `c` command that contains the values in each of the matrix's columns.

```
col1 <- c(7, 2, 6, 9)
col2 <- c(3, 5, 8, 0)
```

To form matrix D, the `cbind` command combines vectors of equal length such that each makes one of the matrix's columns.

```
D <- cbind(col1, col2)
D
```

Alternatively, consider matrix E that has been divided according to its rows:

$$E = \left[\begin{array}{cccc} 3 & 5 & 4 & 3 \\ \hdashline 1 & 3 & 2 & 1 \\ \hdashline 2 & 4 & 3 & 2 \end{array}\right]$$

Once again it is simple to construct three vectors containing the values within each row.

```
row1 <- c(3, 5, 4, 3)
row2 <- c(1, 3, 2, 1)
row3 <- c(2, 4, 3, 2)
row4 <- c(0, 2, 1, 3)
```

To form matrix E, the `rbind` command combines vectors of equal length such that each makes one of the matrix's rows.

```
E <- rbind(row1, row2, row3, row4)
E
```

## 2.1 Constructing a complicated matrix

In some circumstances, we might need to construct a matrix with a particular pattern. For instance, suppose we want to construct a matrix of the form

$$\begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ -1 & 1 & -1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix}$$

First we can create a matrix of the correct dimension

```
M <- matrix(0,10,10)
```

One way to proceed is to use a `for` loop

```
for (i in 1:10) {
   M[i,i]<-1
   if (i>1) M[i,i-1]<--1
   if (i<10) M[i,i+1]<--1
}
```

## 2.2 Constructing a diagonal matrix

In some cases we may want to construct a diagonal matrix. The function `diag` is available for this. `diag` may be used in several ways. If a single scalar input `n` is supplied then the $n \times n$ identity matrix $\mathbf{I}_n$ is created. For instance

```
I <- diag(10)
```

creates a $10 \times 10$ identity matrix. If two scalar inputs `k` and `n` are supplied then the $n \times n$ matrix $k\mathbf{I}_n$ is created, e.g.

```
I <- diag(5,7)
```

creates a $7 \times 7$ matrix with diagonal elements 5. If a single vector input is supplied then a diagonal matrix with diagonal elements corresponding to the elements of that vector is created e.g.

```
H <- diag(c(1,2,3,4))
```

creates the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

In the next section we will see that `diag` can also be used to extract the diagonal elements of a matrix.

# 3 Extracting Data from a Matrix

In workshop week 8 you were introduced to subsetting a vector using the square brackets `[ ]`. We also use this technique to extract information from a matrix, but we need to provide the row number and the column number of the matrix element that we are interested in. In general, to extract the value contained in the $m^{\text{th}}$ row and $n^{\text{th}}$ column cell of matrix $F$, we would execute the command:

```
F[ m, n ]
```

Using the matrices that you have created in this workshop, execute the following extraction commands. Remember that the row number is first and the column number is second.

```
A[ 3, 3 ]
B[ 2, 1 ]
D[ 4, 2 ]
E[ 1, 3 ]
```

Note that the number you put into the square brackets cannot be greater than the dimension of the matrix. For example, the command `C[ 2, 3 ]` produces the message `Error: subscript out of bounds`. Note that we can obtain the dimensions of a matrix using the function `dim`, e.g. `dim(A)` returns `c(3,2)`.

Now that we know how to extract data from a matrix, we can edit its content by using subsetting to identify a particular element and assigning a new value to that position. For example:

```
C
C[ 2, 2 ] <- 4
C
```

Note that the number in the second row and second column had changed from 2 to 4.

# 4 Further Subsetting

As well as extracting a single number from a matrix, the square brackets can also be used to extract vectors. In this case we only need to specify one number either before or after the comma depending on whether we wish to extract a row or column respectively. In general, to extract the $m^{\text{th}}$ row for matrix $F$ we would execute:

```
F[ m, ]
```

Alternatively we can extract the $n^{\text{th}}$ column from matrix $F$ by:

```
F[ , n ]
```

Try the following examples using the matrices that you created earlier in the workshop and note whether the command extracted a row or a column to create the vector.

```
A[ 2, ]
B[ , 3 ]
C[ 1, ]
C[ , 2 ]
D[ , 1 ]
E[ 3, ]
```

In addition to extracting vectors, we can use negative subsetting extract everything from the matrix excluding the specified row or column. For example, reporting matrix $C$ without the third row can be obtained by executing:

```
C[ -3, ]
```

Or, to report matrix $A$ without the first column can be obtained by executing:

```
A[ , -1 ]
```

Finally, if we supply the function `diag` with a square matrix then it will extract the diagonal elements

```
diag(H)
```

## 5 Functions for Matrices

The determinant of a square matrix can be computed in R by using the function `det` for instance

```
det(E)
```

The transpose of a matrix can be obtained by using `t()` e.g.

```
t(E)
```

The inverse of a square matrix can be obtained by using the function `solve`

```
solve(E)
```

---

## Quiz 1: Constructing a matrix (I)

Which of the following commands creates the matrix

$$\begin{bmatrix} 2 & 17 & 32 & 47 & 62 \\ 5 & 20 & 35 & 50 & 65 \\ 8 & 23 & 38 & 53 & 68 \\ 11 & 26 & 41 & 56 & 71 \\ 14 & 29 & 44 & 59 & 74 \end{bmatrix}$$

(**A**) `matrix((2:74),5,5,byrow=TRUE)`

(**B**) `matrix(3*(1:25) - 1, 5,5,byrow=TRUE)`

(**C**) `matrix(rep(2:74,3),5,5,byrow=TRUE)`

(**D**) `matrix(rep(c(2,17,32,47,62),5),5,5,byrow=FALSE)`

(**E**) `matrix(seq(2,74,by=3),5,5,byrow=FALSE)`

---

---

## Quiz 2: Inverting a $5 \times 5$ matrix

Find the inverse of the matrix

$$\begin{bmatrix} 1 & -1 & 0 & 3 & 1 \\ 0 & -1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & -1 \\ 0 & 2 & 1 & 1 & 1 \\ -1 & 0 & 0 & 0 & 2 \end{bmatrix}$$

(**A**)

$$\begin{bmatrix} -1 & 2 & 1 & 0 & -1 \\ -1/4 & 0 & 3/4 & 0 & 1/2 \\ 1/4 & 0 & -7/4 & 1 & -3/2 \\ 3/4 & 1 & 1/4 & 0 & 1/2 \\ -1/2 & 1 & -1/2 & 0 & 0 \end{bmatrix}$$

(**B**)

$$\begin{bmatrix} -1 & 2 & 0 & 1 & -1 \\ -1/4 & 0 & 3/4 & 0 & 1/2 \\ 7/4 & 0 & -1/4 & 1 & -1/2 \\ -1/4 & -1 & 1/4 & 0 & 3/2 \\ 1/2 & 0 & 1/2 & 0 & 0 \end{bmatrix}$$

(**C**)

$$\begin{bmatrix} -1 & -1/4 & 1/4 & 3/4 & -1/2 \\ 2 & 0 & 0 & -1 & 1 \\ 1 & 3/4 & -7/4 & -1/4 & 1/2 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 1/2 & -3/2 & 1/2 & 0 \end{bmatrix}$$

(**D**)

$$\begin{bmatrix} -1 & 2 & 1 & 0 & -1 \\ -1/4 & 0 & 3/4 & 0 & 1/2 \\ 1/4 & 0 & -7/4 & 1 & -3/2 \\ 3/4 & -1 & -1/4 & 0 & 1/2 \\ -1/2 & 1 & 1/2 & 0 & 0 \end{bmatrix}$$

(**E**)

$$\begin{bmatrix} -1 & -1/4 & 1/4 & 3/4 & -1/2 \\ 2 & 0 & 0 & -1 & 1 \\ 1 & 3/4 & -7/4 & -1/4 & 1/2 \\ 0 & 0 & 1 & 0 & 0 \\ -1 & 1/2 & -3/2 & 1/2 & 0 \end{bmatrix}$$

## Quiz 3: Constructing a matrix (II)

Construct a $10 \times 10$ matrix of the form:

$$\mathbf{V} = \begin{bmatrix} 1 & -1 & -2 & 0 & \dots & 0 \\ -1 & 1 & -1 & -2 & \ddots & \vdots \\ -2 & -1 & \ddots & \ddots & \ddots & 0 \\ 0 & -2 & \ddots & \ddots & \ddots & -2 \\ \vdots & \ddots & \ddots & -1 & \ddots & -1 \\ 0 & \dots & 0 & -2 & -1 & 1 \end{bmatrix}$$

Find the determinant of the matrix.

## Quiz 4: Inverse of a matrix

Find the inverse of the matrix $\mathbf{V}$ defined in the previous question. Find the (7,8) element of $\mathbf{V}^{-1}$ to 3 decimal places.

## Quiz 5: Determinants of large matrix

Consider other matrices of the same general form as $\mathbf{V}$. What size $n$ of $n \times n$ matrix leads to a matrix with determinant 190231?

## Quiz 6: Largest determinant of a $3 \times 3$ matrix

What is the largest possible determinant that a **symmetric** $3 \times 3$ matrix, all of whose entries lie in the set of integers $\{1, 2, 3, 4, 5, 6\}$ can have? Use R to determine the answer.