



# PROGRAMMING : MATRIX

---

# Objectives :

- Introduction
- Create matrices
- Index/Subset matrix
- More about matrix
- Applications of R matrices

# INTRODUCTION.....

A matrix is a rectangular array of data elements, arranged in rows and columns. Matrices in R are homogenous i.e., they can hold a single type of data. In the first section, we look at various methods of creating matrices in R.

	COLUMNS		
R			
O			
W			
S			

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

# Creating Matrix.....

The easiest way to create a matrix in R is to use the `matrix()` function.

```
Console Terminal x Jobs x
~/
> #Form a matrix of order 2*3 using the elements 1,4,7,2,5,8
> m<-matrix(c(1,4,7,2,5,8),2,3)
> m #display the matrix
      [,1] [,2] [,3]
[1,]   1   7   5
[2,]   4   2   8
> |
```

```
Console Terminal x Jobs x
~/
> #Form a matrix of order 2*3 using the elements 1,3,4,2,5,8,6,9
> m<-matrix(data=c(1,3,4,2,5,8,6,9), nrow=2,ncol=4)
> m #display the matrix
      [,1] [,2] [,3] [,4]
[1,]   1   4   5   6
[2,]   3   2   8   9
> |
```

# cbind and rbind

```
> mat1<-c(1,2,3,4)
> mat2<-c(9,8,7,6)
> #merge vector column wise
> matcb<-cbind(mat1,mat2)
> matcb
      mat1 mat2
[1,]    1    9
[2,]    2    8
[3,]    3    7
[4,]    4    6
> #merge vector row wise
> matr<-rbind(mat1,mat2)
> matr
      [,1] [,2] [,3] [,4]
mat1    1    2    3    4
mat2    9    8    7    6
```

**These functions are used to form a matrix using two or more vectors. cbind and rbind append the vectors columnwise and rowwise respectively.**

# How to convert a list into a matrix?

`as.matrix` is used to coerce a data structure to the type matrix. Therefore, a list can also be changed into matrix by using `as.matrix()` command.

```
Console Terminal x Jobs x
~/
> l<-list(x=c(3,2,0),y=1:3,z=c(4,5,6))
> l
$x
[1] 3 2 0

$y
[1] 1 2 3

$z
[1] 4 5 6

> z=as.data.frame(l)
> z
  x y z
1 3 1 4
2 2 2 5
3 0 3 6
> w=as.matrix(z)
> w
      x y z
[1,] 3 1 4
[2,] 2 2 5
[3,] 0 3 6
> |
```

# Index/Subset Matrices

- ❑ The [ ] operator can be used to index the elements but since matrices are 2D, we need to specify both row number and column number.

```
Console Terminal x Jobs x
~/
> m<-matrix(data=1:8,4,2)
> m[1,]#display first row
[1] 1 5
> m[,1]#display 1st column
[1] 1 2 3 4
> m[c(1,3),]#display all elements 1st and 3rd column
      [,1] [,2]
[1,]    1    5
[2,]    3    7
> m[-1,c(1,2)]#display elements of 1st and 2nd column in all rows except 1st row
      [,1] [,2]
[1,]    2    6
[2,]    3    7
[3,]    4    8
> sort(m[1,])#elements of 1st row in ascending order
[1] 1 5
> sort(m[,2])#elements of 2nd column in ascending order
[1] 5 6 7 8
> rev(sort(m[1,]))#elements of 1st row in descending order
[1] 5 1
> rev(sort(m[,1]))#elements of 1st column in descending order
[1] 4 3 2 1
> |
```

# More about Matrices

✓ Command for any matrix

✓ Command for a square matrix

```
Console Terminal x Jobs x
~/
> m<-matrix(data=1:8,4,2)
> rowSums(m)#use to find sum of each row in a matrix
[1] 6 8 10 12
> rowMeans(m)#use to find mean of each row in a matrix
[1] 3 4 5 6
> colSums(m)#use to find sum of each column in a matrix
[1] 10 26
> colMeans(m)#use to find mean of each column in a matrix
[1] 2.5 6.5
> mean(m)
[1] 4.5
> sd(m)
[1] 2.44949
> var(m)
      [,1] [,2]
[1,] 1.666667 1.666667
[2,] 1.666667 1.666667
> |
```

```
Console Terminal x Jobs x
~/
> m<-matrix(data=c(1,2,5,7),2,2)
> solve(m)# gives inverse of matrix
      [,1] [,2]
[1,] -2.3333333 1.6666667
[2,] 0.6666667 -0.3333333
> eigen(m)#gives eigenvalues and eigenvectors
eigen() decomposition
$values
[1] 8.3588989 -0.3588989

$vectors
      [,1] [,2]
[1,] -0.5619986 -0.9649955
[2,] -0.8271382 0.2622663

> det(m)# used to find determinant of matrix
[1] -3
> t(m)#gives transpose of matrix
      [,1] [,2]
[1,] 1 2
[2,] 5 7
> |
```



# Changing Row names and Column names

- **rownames()** and **colnames()** are the commands that are used to add row names and column names of a matrix.

```
Console Terminal x Jobs x
~/
> m<-matrix(c(1,5,3,8,9,7),2,3)
> m
      [,1] [,2] [,3]
[1,]    1    3    9
[2,]    5    8    7
> rownames(m)<-c("Vanilla","strawberry")#2 rows define 2 names
> m
      [,1] [,2] [,3]
Vanilla    1    3    9
strawberry  5    8    7
> colnames(m)<-c("Rasperry","Butterscotch","Blackforest")
> m
      Rasperry Butterscotch Blackforest
Vanilla      1          3          9
strawberry   5          8          7
>
```



# Applications

- ✓ In geology, matrices are used for taking surveys and also used for plotting graphs.
- ✓ In robotics and communication, matrices are the best elements for robotic movements.
- ✓ Matrices are used in calculating the gross domestic products in economics.
- ✓ In physical, matrices can be applied in study of an electrical circuit.

**T H E E N D**

**~ By Chetna Mehra ~  
Mat/19/9**