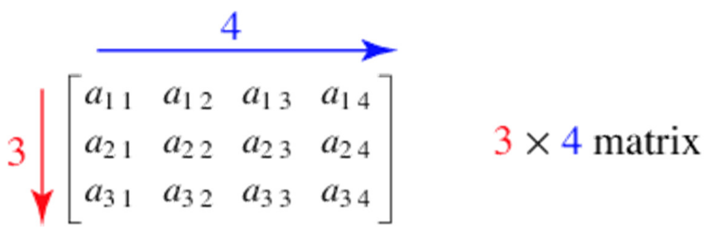
Simple R data analysis on matrices

I intend to show in the next few notes on the applications of some simple R statistical programming language to arouse the interest in using R data analysis package.

To start with, we know a matrix is a rectangular table of cells with number entries in rows and columns, as shown in the diagram below which is a two-dimension matrix with 3 rows and 4 columns:



In general, we write to define an *m* x *n* matrix *A* to consist of *m* rows and *n* columns with each entry in the matrix called $a_{i,j}$ for all $1 \leq i \leq m$ and $1 \leq j \leq n$. These number entries represent the variable data. Of course, a one-dimension matrix is a simple row or column of figures. A matrix can also have three-dimension arrangement.

The R programming language for matrix is matrix() which covers:

```
matrix (data, nrow=1, ncol=1, byrow=FALSE, dimnames=NULL)
```

where: data is a must, nrow is the number of rows, ncol, the number of columns, byrow controls the data set listing with default FALSE for ascending order and dimnames provides the names for the rows and columns.

Let us see some simple examples:

```
> matrix(1:12,nrow=4,ncol=3)   #by default
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

```
> matrix(1:12,nrow=4,ncol=3,byrow=T) #ascending numbers by row
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
>

> A=matrix(1:12,nrow=3,ncol=4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
>
> t(A)   #transforming A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
>

> A=B=matrix(1:12,nrow=3)
> A+B
     [,1] [,2] [,3] [,4]
[1,]    2    8   14   20
[2,]    4   10   16   22
[3,]    6   12   18   24
>

> 5*A
     [,1] [,2] [,3] [,4]
[1,]    5   20   35   50
[2,]   10   25   40   55
[3,]   15   30   45   60
>
> A*A
     [,1] [,2] [,3] [,4]
[1,]    1   16   49  100
[2,]    4   25   64  121
[3,]    9   36   81  144
>
```

```
> A^2
     [,1] [,2] [,3] [,4]
[1,]    1   16   49  100
[2,]    4   25   64  121
[3,]    9   36   81  144
>
```

```
> A/B
     [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    1    1    1    1
[3,]    1    1    1    1
>
```

A matrix diagonal extracts or replaces the diagonals of a matrix, or constructs a diagonal matrix.

Let's consider another set of examples. If z is a matrix then diag(z) returns the diagonal of x.

```
> z=matrix(1:16,nrow=4)
> z
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
>
> diag(z)    #diagonals of z
[1]  1  6 11 16
>
> diag(diag(z))
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    6    0    0
[3,]    0    0   11    0
[4,]    0    0    0   16
>
> diag(3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
>
> diag(4)
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
>
```

```
# If:
> X=matrix(1:4,nrow=2,ncol=2)
> Y=matrix(5:8,nrow=2,ncol=2)
> X
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> Y
     [,1] [,2]
[1,]    5    7
[2,]    6    8

X*Y   # Simple multiplications of X and Y only
     [,1] [,2]
[1,]    5   21
[2,]   12   32

> X%*%Y   # Note the use of %*% for multiplying matrix X with matrix Y
     [,1] [,2]
[1,]   23   31
[2,]   34   46

> Y%*%X   # Note the different results of XY and YX
     [,1] [,2]
[1,]   19   43
[2,]   22   50
```

It is interesting to note that matrix ***X*** post-multiplied ***Y*** is different from ***Y*** post-multiplied ***X***, i.e. $XY \neq YX$.