MATRICES

Matrices

Matrices are 2 dimensional array. They have one more attribute then vector called dimension attribute. [Vector has 2 attribute, mode and length] The dimension attribute is itself an integer vector of length 2(nrow, ncol)

Matrix(data = NA, nrow=1, ncol=1, byrow=FALSE, dimnames = list(c(rowname),c(colname))

Byrow helps to decide whether rows are filled or columns are filled.

Dimnames allows to give names to the rows and columns.

Matrices

```
m <-matrix( c(1,2,3,4),nrow=2, ncol=2)
> m
      [1]  [2]
[1]   1     3
[2]   2     4
> dim (m)
[1] 2 3   # It shows that m has 2 rows and 3 columns
```

Matrices are constructed column-wise, so entries can be thought of starting in the "upper left" corner and running down the columns.

```
> m <-matrix(1:6, nrow=2, ncol =3)
> m
```

Matrices

Matrices can also be created by

```
> m<- 1:10
> m
[1] 1 2 3 4 5 6 7 8 9 10
> dim(m) <- c(2,5)
>m
     [,1] [,2] [,3] [,4] [,5]
[1,] 1    3    5    7    9
[2,] 2    4    6    8    10
```

Vectorized Matrix Operation

```
X<-matrix(1:4,2,2);y<-matrix(rep(10,4),2,2)
> x*y  ## element-wise multiplication
        [,1]   [,2]
[1,]   10     20
[2,]    20    40
> x/y
         [,1]   [,2]
[1,]     0.1   0.3
[2,]     0.2   0.4
> x %*%y    ## true matrix multiplication
         [,1]   [,2]
[1,]     40     40
[2,]     60     60
```

Cbind-ing and rbind-ing

- Matrices can be created by column-binding or row-binding with cbind() and rbind()

```
> x <-1:3
> Y <-10:12
cbind(x,y)
        x    y
[1,]   1    10
[2,]   2    11
[3,]   3    12
rbind(x,y)
     [,1]   [,2]   [,3]
x    1       2       3
y   10      11      12
```

Matrices name

```
> m <-matrix(c(30,35,40,45) , nrow = 2, ncol =
2)
> dimnames(m) <- list (c("Sumit","Nikita"), c("R
Prog", "C Prog"))
> m
          R Prog    C Prog
Sumit      30          40
Nikita      35          45
```

Accessing Matrix

- Matrix can be accessed in the ususal way with (i,j) type indices
- X<-matrix(1:6,2,3)
- >x[1,2]
- [1] 3
- >x[2,1]
- [1] 2
- Indices can be missing
- >x[1,]
- 1,3,5
- >x[,2]
- 3,4

Accessing Matrix

- By default when a single element of matrix is retrieved, it is returned as a vector of length 1 rather than a 1*1 matrix. This behaviour can be turned off by setting drop=FALSE

```
>X<-matrix(1:6,2,3)
X[1,2]
[1] 3
>x[1,2,drop=false]
      [,1]
[1,]    3
X[1, ]
[1] 1 3 5
X[1, ,drop = FALSE)
       [,1]   [,2]    [,3]
[1,]   1        3        5
```

Transpose

- Transpose is very important in matrix.
- For this R has function t()
- > m<-matrix(1:4,2,2)
- > m
- [1,] 1 3
- [2,] 2 4
- > t(m)
- [1,] 1 2
- [2,] 3 4

Mathematical functions on Matrix

- Diagonal of Matrix – diag(m)
- Eigenvalue & Eigenvectors
- Eigenvectors are denoted by A and defined as a vector that when multiplied by given matrix will just increase the magnitude of matrix by scalar value λ. They exists for square matrix.
- $A.V=\lambda V$
- e<-eigen(V)
- e$value
- e$vector

Other Matrix operations

- solve(a,b): solve a set of equations. (If b is not given then solve will return the inverse of a)
- ginv():Moore-Penrose generalized inverse of a matrix
- rowMeans: vector of row means
- rowSums: vector of row sums
- colMeans: vector of column means
- colSums: vector of column sums

QUESTIONS