

# **Introduction to R Software**

## **Swayam Prabha**

## **Lecture 12**

### **Matrix Operations and Missing Data**

**Shalabh**

**Department of Mathematics and Statistics**

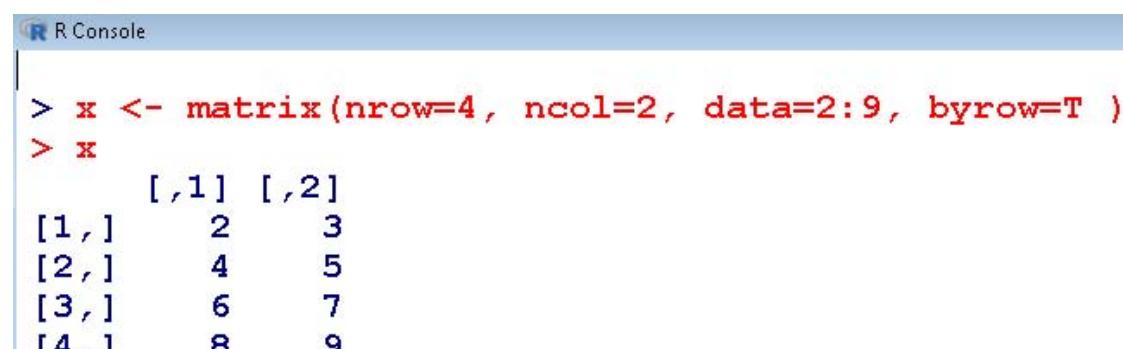
**Indian Institute of Technology Kanpur**

**Slides can be downloaded from  
<http://home.iitk.ac.in/~shalab/sp>**



## Transpose of a matrix X: X'

```
> x <- matrix(nrow=4, ncol=2, data=2:9, byrow=T )  
> x  
     [,1] [,2]  
[1,]    2    3  
[2,]    4    5  
[3,]    6    7  
[4,]    8    9
```



The screenshot shows the R console window with the title "R Console". Inside, the same R command and output are displayed as in the main text, demonstrating the transpose operation.

```
R Console  
> x <- matrix(nrow=4, ncol=2, data=2:9, byrow=T )  
> x  
     [,1] [,2]  
[1,]    2    3  
[2,]    4    5  
[3,]    6    7  
[4,]    8    9
```

## Transpose of a matrix $X$ : $X'$

```
> xt <- t(x)
```

```
> xt
```

	[,1]	[,2]	[,3]	[,4]
[1,]	2	4	6	8
[2,]	3	5	7	9

```
R Console  
> xt <- t(x)  
> xt  
[1,] 2 4 6 8  
[2,] 3 5 7 9  
>
```

## Multiplication of a matrix with a constant

```
> x <- matrix(nrow=4, ncol=2, data=1:8, byrow=T )  
  
> x  
     [,1]     [,2]  
[1,]    1    2  
[2,]    3    4  
[3,]    5    6  
[4,]    7    8
```

```
R Console  
> x <- matrix(nrow=4, ncol=2, data=1:8, byrow=T )  
>  
> x  
     [,1]     [,2]  
[1,]    1    2  
[2,]    3    4  
[3,]    5    6  
[4,]    7    8  
>
```

## Multiplication of a matrix with a constant

```
> 4*x  
      [,1]    [,2]  
[1,]    4     8  
[2,]   12    16  
[3,]   20    24  
[4,]   28    32
```

R Console

```
> 4*x  
      [,1]    [,2]  
[1,]    4     8  
[2,]   12    16  
[3,]   20    24  
[4,]   28    32  
>
```

## Matrix multiplication: operator `%*%`

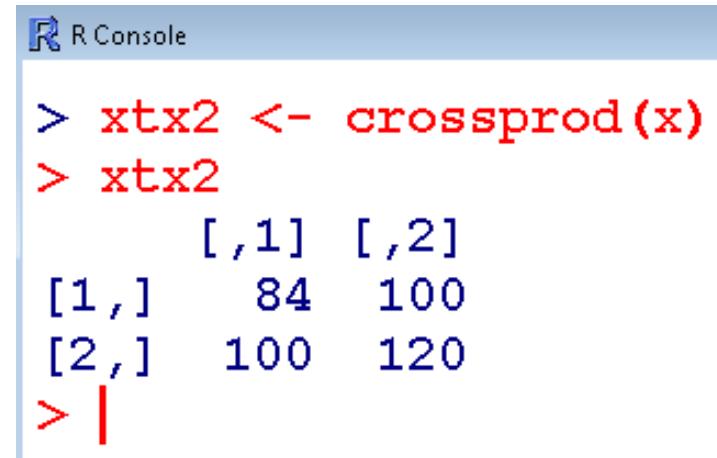
Consider the multiplication of  $X'$  with  $X$

```
> xtx <- t(x) %*% x  
  
> xtx  
      [,1]    [,2]  
[1,]   84    100  
[2,]  100    120
```

```
R Console  
> xtx <- t(x) %*% x  
>  
> xtx  
      [,1]    [,2]  
[1,]   84    100  
[2,]  100    120
```

## Cross product of a matrix $X \cdot X'X$ with a function `crossprod`

```
> xtx2 <- crossprod(x)  
  
> xtx2  
      [,1]   [,2]  
[1,]    84    100  
[2,]   100    120
```



R Console window showing the command `> xtx2 <- crossprod(x)` and its output. The output shows a 2x2 matrix:

	[,1]	[,2]
[1,]	84	100
[2,]	100	120

The cursor is positioned at the end of the output line.

**Note:** Command `crossprod( )` executes the multiplication faster than the conventional method with `t(x)%*%x`

Addition and subtraction of matrices (of same dimensions)  
can be executed with the usual operators + and -

```
> x <- matrix(nrow=4, ncol=2, data=1:8, byrow=T)
```

```
> x
```

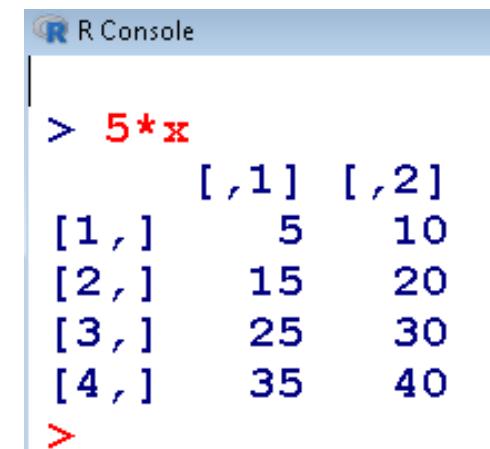
	[,1]	[,2]
[1, ]	1	2
[2, ]	3	4
[3, ]	5	6
[4, ]	7	8

R Console

```
> x <- matrix(nrow=4, ncol=2, data=1:8, byrow=T)
> x
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
```

**Addition and subtraction of matrices (of same dimensions!) can be executed with the usual operators + and -**

```
> 5*x  
      [,1]   [,2]  
[1,]    5    10  
[2,]   15    20  
[3,]   25    30  
[4,]   35    40
```

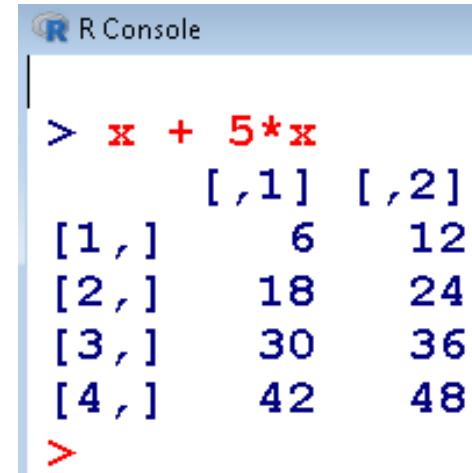


A screenshot of an R console window titled "R Console". The window shows the command "5\*x" followed by the resulting matrix output. The matrix has four rows and two columns, with values 5, 10, 15, 20 in the first column and 25, 30, 35, 40 in the second column.

```
> 5*x  
      [,1]   [,2]  
[1,]    5    10  
[2,]   15    20  
[3,]   25    30  
[4,]   35    40  
>
```

**Addition and subtraction of matrices (of same dimensions!) can be executed with the usual operators + and -**

```
> x + 5*x  
      [,1]  [,2]  
[1,]    6   12  
[2,]   18   24  
[3,]   30   36  
[4,]   42   48
```



R Console

```
> x + 5*x  
      [,1]  [,2]  
[1,]    6   12  
[2,]   18   24  
[3,]   30   36  
[4,]   42   48  
>
```

**Addition and subtraction of matrices (of same dimensions!) can be executed with the usual operators + and -**

```
> 5*x - x
      [,1]    [,2]
[1,]    4     8
[2,]   12    16
[3,]   20    24
[4,]   28    32
```

R Console

```
> 5*x - x
      [,1]    [,2]
[1,]    4     8
[2,]   12    16
[3,]   20    24
[4,]   28    32
>
```

## Inverse of a matrix

`solve()` finds the inverse of a positive definite matrix

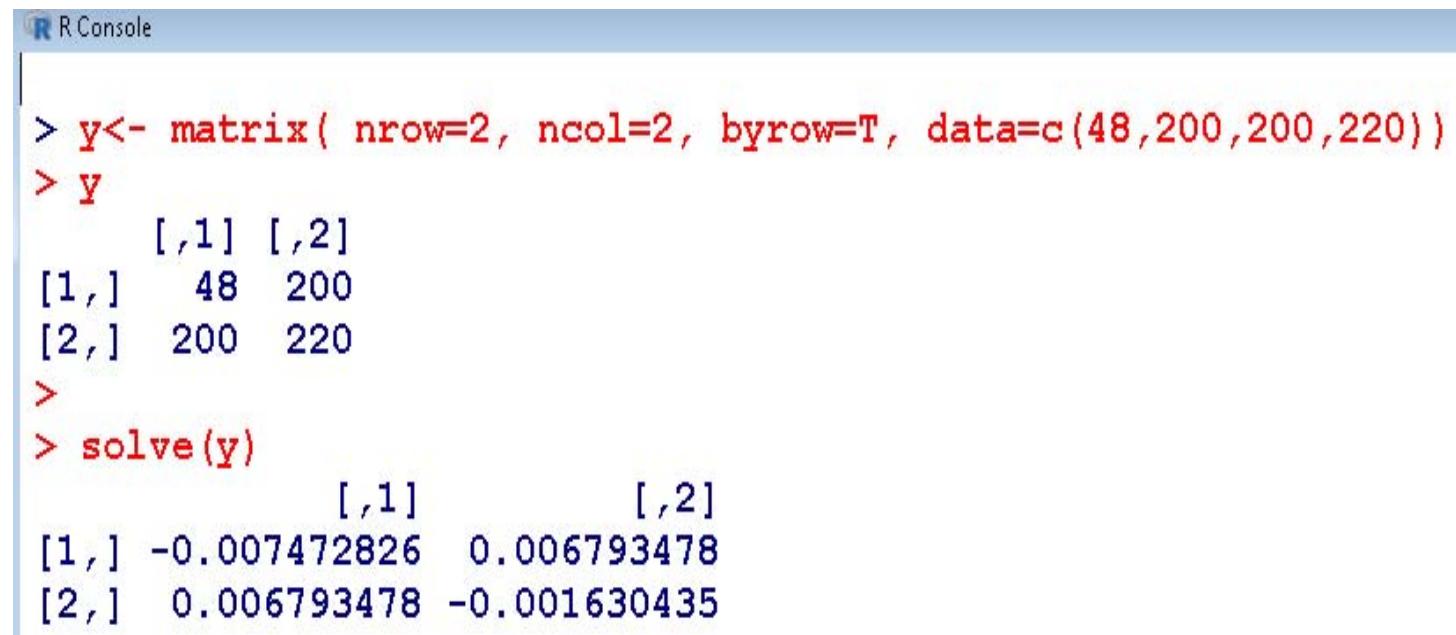
Example:

```
> y<- matrix( nrow=2, ncol=2, byrow=T,  
data=c(48,200,200,220))
```

```
> y  
      [,1] [,2]  
[1,]    48   200  
[2,]   200   220
```

## Inverse of a matrix

```
> solve(y)
      [,1]      [,2]
[1,] -0.007472826  0.006793478
[2,]  0.006793478 -0.001630435
```



The screenshot shows an R console window titled "R Console". Inside the window, the following R code is displayed:

```
> y<- matrix( nrow=2, ncol=2, byrow=T, data=c(48,200,200,220) )
> y
      [,1] [,2]
[1,]   48   200
[2,]   200  220
>
> solve(y)
      [,1]      [,2]
[1,] -0.007472826  0.006793478
[2,]  0.006793478 -0.001630435
```

# Eigen Values and Eigen Vectors

`eigen( )` finds the eigen values and eigen vectors of a positive definite matrix.

$$|A - \lambda I| = 0,$$

$$(A - \lambda I)b = 0$$

# Eigen Values and Eigen Vectors

Example:

```
> y
      [,1] [,2]
[1,]   48   200
[2,]  200   220

> eigen(y)

eigen() decomposition
$`values`
[1] 351.70622 -83.70622

$`vectors`
      [,1]      [,2]
[1,] 0.5499874 -0.8351730
[2,] 0.8351730  0.5499874
```

```
R Console
> y
      [,1] [,2]
[1,]   48   200
[2,]  200   220
>
> eigen(y)
eigen() decomposition
$`values`
[1] 351.70622 -83.70622

$`vectors`
      [,1]      [,2]
[1,] 0.5499874 -0.8351730
[2,] 0.8351730  0.5499874
```