Foundations of R Software

Lecture 12

Basics of Calculations

::::

Matrices

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Matrix

Matrices are important objects in any calculation.

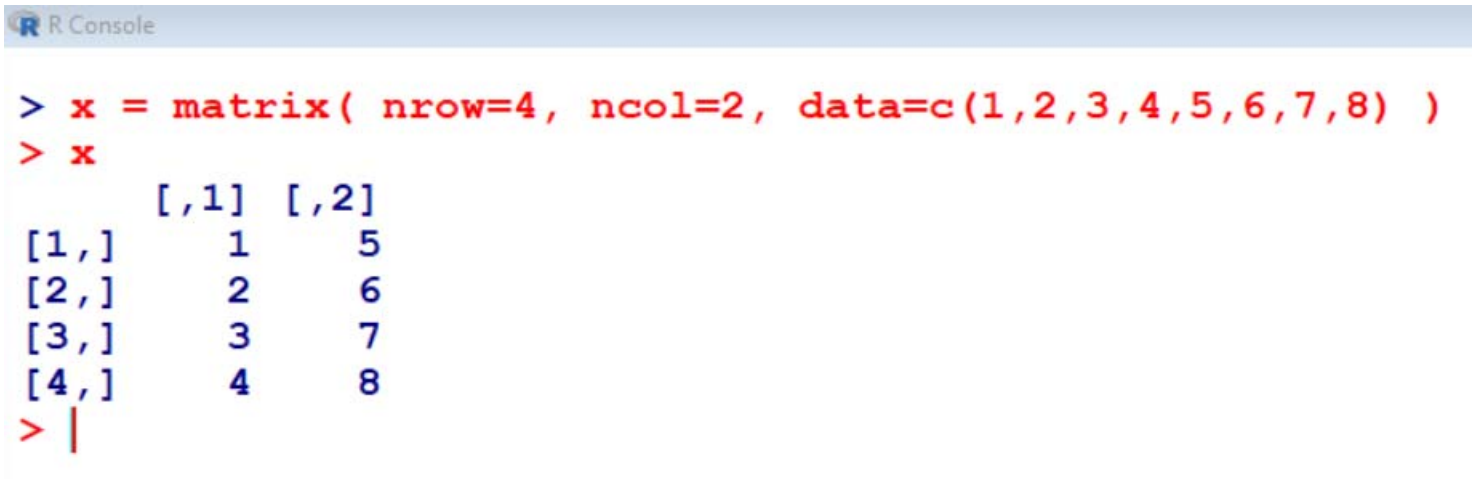
A matrix is a rectangular array with *p* rows and *n* columns.

An element in the *i*-th row and *j*-th column is denoted by X_{ij} (book version) or *X*[*i*, *j*] ("program version"), *i* = 1,2,...,*n*, *j* = 1,2,...,*p*.

An element of a matrix can also be an object, for example a string. However, in mathematics, we are mostly interested in numerical matrices, whose elements are generally real numbers

In R, a 4 × 2-matrix *X* can be created with a following command:

```
> x = matrix( nrow=4, ncol=2,
                        data=c(1,2,3,4,5,6,7,8) )
> x
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```



Matrix parameters

We see:

The parameter `nrow` defines the row number of a matrix.

The parameter `ncol` defines the column number of a matrix.

The parameter `data` assigns specified values to the matrix elements.

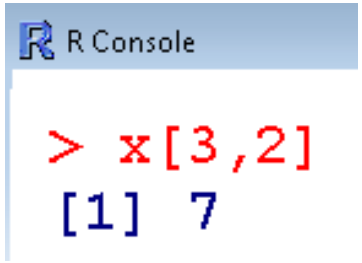
The data values from the parameters are written column-wise in matrix.

Accessing any element of a matrix

```
> x
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```

One can access a single element of a matrix with `x[i,j]`:

```
> x[3,2]
[1] 7
```



Entering data column wise in a matrix

In case, the data has to be entered column wise, then a 4 × 2-matrix

***X* can be created with**

```
> x = matrix( nrow=4, ncol=2,
                        data=c(1,2,3,4,5,6,7,8))
```

OR

```
x = matrix( nrow=4, ncol=2,
        data=c(1,2,3,4,5,6,7,8), byrow = FALSE)
> x
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```

Entering data row wise in a matrix

In case, the data has to be entered row wise, then a 4 × 2-matrix *X* can be created with

```
> y = matrix( nrow=4, ncol=2,
          data=c(1,2,3,4,5,6,7,8), byrow = TRUE)
> y
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
```

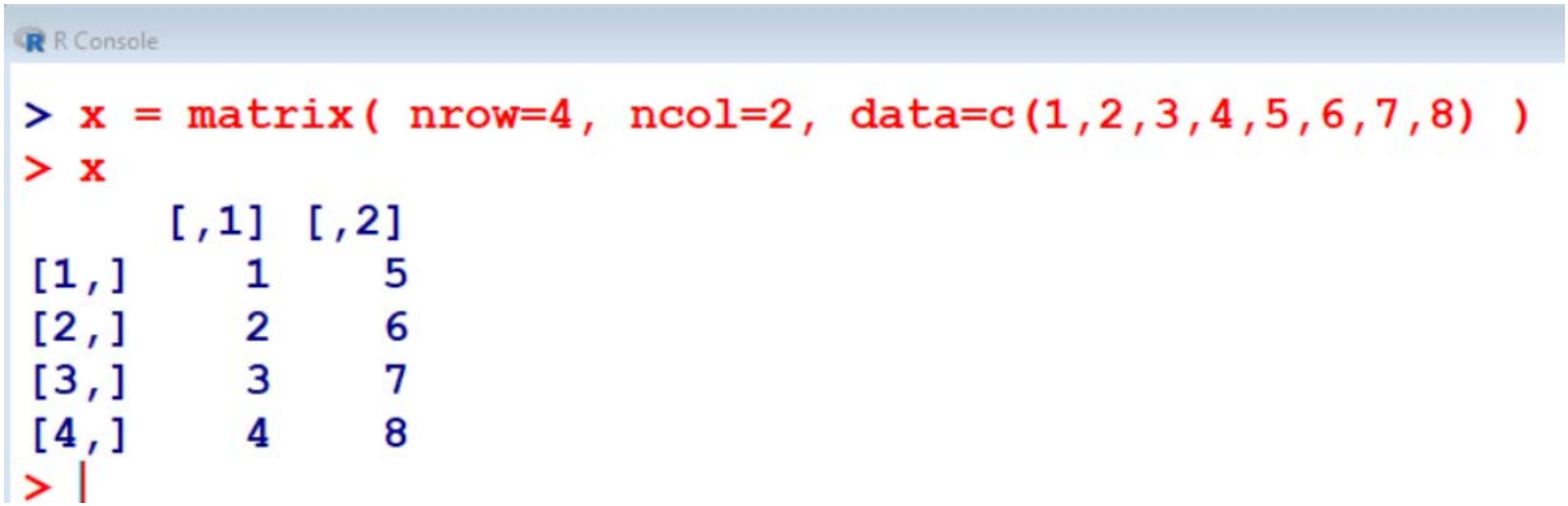
Entering data row wise and column wise in a matrix

R Console

```
> x = matrix( nrow=4, ncol=2, data=c(1,2,3,4,5,6,7,8) )
> x
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
> y = matrix( nrow=4, ncol=2, data=c(1,2,3,4,5,6,7,8), byrow = TRUE )
> y
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
>
```

In R, a 4 × 2-matrix *X* can be created with a following command:

```
> x = matrix( nrow=4, ncol=2,
                        data=c(1,2,3,4,5,6,7,8) )
> x
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```



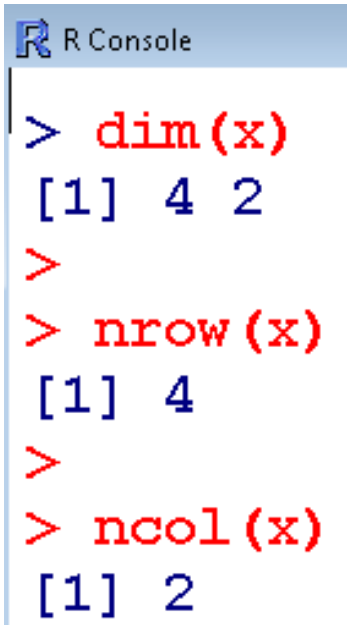
Properties of a Matrix

We can get specific *properties* of a matrix:

```
> dim(x)  # tells the dimension of matrix
[1] 4 2
```

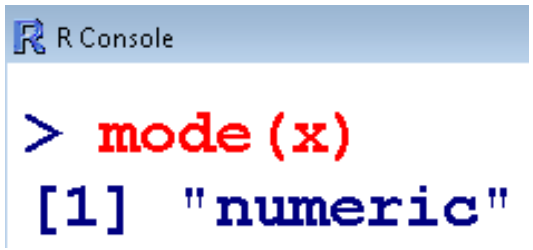
```
> nrow(x) # tells the number of rows
[1] 4
```

```
> ncol(x) # tells the number of columns
[1] 2
```



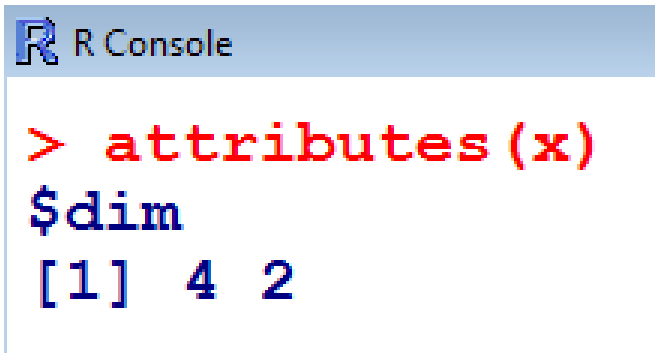
Properties of a Matrix

```
> mode(x)        # Informs the type or storage
                   mode of an object, e.g.,
                   numerical, logical etc.
[1] "numeric"
```



attributes provides all the attributes of an object

```
> attributes(x) #Informs the dimension of matrix
$dim [1] 4 2
```



Help on the Object "Matrix"

To know more about these important objects, we use R–help on "matrix".

```
> help("matrix")
matrix      package:base      R Documentation

Matrices

Description:
```

'matrix' creates a matrix from the given set of values.

'as.matrix' attempts to turn its argument into a matrix.

'is.matrix' tests if its argument is a (strict) matrix. It is generic: you can write methods to handle specific classes of objects, see Internal Methods.

Then we get an overview on how a matrix can be created and what parameters are available:

Usage:

```
matrix(data [= NA,nrow = 1,ncol = 1,byrow = FALSE,
                                    dimnames = NULL)
as.matrix(x)
is.matrix(x)
```

Arguments:

`data`: an optional data vector.

`nrow`: the desired number of rows

`ncol`: the desired number of columns

`byrow`: `logical.` If `'FALSE'`(the default) the matrix is filled by columns, otherwise the matrix is filled by rows.

`dimnames`: A `'dimnames'` attribute for the matrix: a `'list'` of length 2.

`x`: an R object.

Then, the meaning of each parameter is explained:

Details:

If either of 'nrow' or 'ncol' is not given, an attempt is made to infer it from the length of 'data' and the other parameter.

If there are too few elements in 'data' to fill the array, then the elements in 'data' are recycled. If 'data' has length zero, 'NA' of an appropriate type is used for atomic vectors and 'NULL' for lists.

'is.matrix' returns 'TRUE' if 'x' is a matrix (i.e., it is _not_ a 'data.frame' and has a 'dim' attribute of length 2) and 'FALSE' otherwise.

'as.matrix' is a generic function. The method for data frames will convert any non-numeric/complex column into a character vector using 'format' and so return a character matrix, except that all-logical data frames will be coerced to a logical matrix. `

Finally, references and cross-references are displayed...

References:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S Language_. Wadsworth & Brooks/Cole.

See Also:

'data.matrix', which attempts to convert to a numeric matrix.

.. as well as an example:

Examples:

```
is.matrix(as.matrix(1:10))
data(warpbreaks)
!is.matrix(warpbreaks)# data.frame, NOT matrix!
warpbreaks[1:10,]
as.matrix(warpbreaks[1:10,]) #using
as.matrix.data.frame(.) method
```