

About R Language

- ✓ R is a computer language for carrying out statistical computations.
- ✓ R is Free Software, and runs on a variety of platforms
- ✓ Command-line execution based on function calls.
- ✓ Workspace containing data and functions.
- ✓ Extensible with user functions.
- ✓ Graphics devices.
- ✓ R packages can contain not only code, but also other resources like documentation and sample data sets.
- ✓ Well-defined format that ensures easy installation, a basic standard of documentation, and enhances portability and reliability.
- ✓ The basic mode of interaction is ‘read – evaluate – print’.
- ✓ The R Project is an international collaboration of researchers in statistical computing.
- ✓ There are roughly 20 members of the “R Core Team” who maintain and enhance R.
- ✓ Releases of the R environment are made through the CRAN (comprehensive R archive network) twice per year.
- ✓ The software is released under a “free software” license, which makes it possible for anyone to download and use it.
- ✓ There are over 3500 extension packages that have been contributed to CRAN.
- ✓ R is a computer language which is processed by a special program called an interpreter. This program reads and evaluates R language expressions, and prints the values determined for the expressions.

Installation

- ✓ R can be downloaded from one of the mirror sites in <http://cran.r-project.org/mirrors.html>. You should pick your nearest location.

Using External Data

- ✓ R offers plenty of options for loading external data, including Excel, Minitab, SAS and SPSS files.

Experiment-I

Basic Concepts in R, Understanding Data types, importing/exporting data

- ✓ After R is started, there is a console awaiting for input. At the prompt (>), you can enter numbers and perform calculations.

```
> 2+3
[1] 5
> 100+200+300
[1] 600
```

```
> options(digits=10)
> 20/3
[1] 6.666666667
> exp(2)
[1] 7.389056099
> pi
[1] 3.141592654
> options(digits=7) #default
```

Methods of data Input:

C function :-

R functions are invoked by its name, followed by the parenthesis and arguments. The function c is used to combine three numeric values into a vector

```
> c(1,2,3)
[1] 1 2 3
> c(10,20,30)
[1] 10 20 30
```

- ✓ All text after the pound sign "#" within the same line is considered a comment.

```
> 5                # type 5 at the prompt
[1] 5                # here 5 is returned
> 3 + 4            # adding two numbers
[1] 7
> 5^3              # will compute 5^3
```

```
[1] 125
> pi                # pi value
[1] 3.141593
> 1 + 2 * 3         # Normal arithmetic rules apply
[1] 7
```

- ✓ R's basic operators have the following precedence (listed in highest-to-lowest order)

\wedge	exponentiation
- +	unary minus and plus
:	sequence operator
%/% %%	integer division, remainder
* /	multiplication, division
+ -	addition, subtraction

Example:-

>2^3^2 [1] 512	> (2^3)^2 [1] 64	> 2^(3^2) [1] 512	>sqrt(2) [1] 1.414214
> log(10) [1] 2.302585	> log10 (10) [1] 1	> sin(1) [1] 0.841471	>4*atan(1) [1] 3.141593

Seq operator and seq function:-

- ✓ The expression $n1:n2$, generates the sequence of integers from $n1$ to $n2$.

> 1:15 #print the numbers 1 to 15

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

> 5:-5# print the numbers 5 to -5

[1] 5 4 3 2 1 0 -1 -2 -3 -4 -5

> seq(2,8,by=2) #specifies interval and increment

[1] 2 4 6 8

> seq(0,1,length=11) # specifies interval and the number of elements

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

- ✓ NA is used to indicate that a value is missing or not available. Any arithmetic expression which contains NA will produce NA as a result.

```
> 1/0      >Inf-Inf      >sqrt(-1)    > 1+sin(NA)
[1] Inf     [1] NaN           [1] NaN     [1] NA
```

rep function:

In order to enter the data containing repeated values ,rep function

```
> x<-c(rep(1,4),rep(2,2),rep(3,5),7,rep(9,2));
```

```
> x
```

```
[1] 1 1 1 1 2 2 3 3 3 3 3 7 9 9
```

Matrix form:-

```
> fr.dist<-matrix(c(10,5,20,10,30,120,40,22,50,13,60,52),nrow=6,byrow=T)
```

```
> fr.dist
```

```
 [1] [2]
```

```
[1,] 10  5
```

```
[2,] 20 10
```

```
[3,] 30 120
```

```
[4,] 40  22
```

```
[5,] 50  13
```

```
[6,] 60  52
```

Assignment:-

Values are stored by assigning them a name.

- The following statements all store the value 18 under the name .

```
>x = 18      >x <- 18      >18 -> x
```

- Variables can be used in expressions in the same way as numbers.

```
> x=22
```

```
> x=x+25
>x
[1] 47
```

- Individual values can be combined into a vector by using the c function.

```
> x=c(1,2,3,4)
>x
[1] 1 2 3 4
```

Character:- (A character object is used to represent string values in R)

- as.character() function is used to convert objects into character values:

```
> x=as.character(4.58)
>x
[1] "4.58"
```

- Strings can be concatenated by using paste function.

```
>paste("First", "Second", "Third")
[1] "First Second Third"
```

```
>paste("First", "Second", "Third", sep = ":")
[1] "First:Second:Third"
```

```
>fname = "Sri"; lname ="Ram"
>paste(fname)
>paste(fname,lname)
[1] "Sri Ram"
```

Vector Arithmetic:-

- Arithmetic operations of vectors are performed member wise.

```
> a = c(1, 3, 5, 7)
> b = c(1, 2, 4, 8)
```

If we add a and b, the sum would be a vector whose members are the sum of the corresponding members from a and b.

```
>a+b
[1] 2 5 9 15
```

If we multiply a by 5, we get a vector with each of its members multiplied by 5.

```
> 5*a
```

```
[1] 5 15 25 35
```

Similarly for subtraction, multiplication and division, we get new vectors via member wise operations.

```
>a-b
```

```
[1] 0 1 1 -1
```

```
>a*b
```

```
[1] 1 6 20 56
```

```
>a/b
```

```
[1] 1.000 1.500 1.250 0.875
```

```
> a=c(1,2,3,4)
```

```
> 2*a+1
```

```
[1] 3 5 7 9
```

- If two vectors are of unequal length, the shorter one will be recycled in order to match the longer vector

```
> u=c(10,20,30)
```

```
> v=c(1,2,3,4,5,6,7,8,9)
```

```
>u+v
```

```
[1] 11 22 33 14 25 36 17 28 39
```

Data Frame:

A **data frame** is used for storing data tables. It is a list of vectors of equal length. For example, the following variable df is a data frame containing three vectors n, s, b.

```
>n = c(2, 3, 5)
> s = c("aa", "bb", "cc")
> b = c(TRUE, FALSE, TRUE)
>df = data.frame (n, s, b)      # df is a data frame
>df
  n s    b
1 2 aa TRUE
2 3 bb FALSE
3 5 cc TRUE
```

Data Import:

CSV File: The sample data can also be in comma separated values (CSV) format. The first row of the data file should contain the column names instead of the actual data.

Important Note:

Enter the following data(or any data) in Excel sheet and save it as CSV file. That is Easy and fast for further analysis

```
col1 col2 col3
34  23  76
56  54  43
76  34  24
54  76  67
32  24  54
```

Code:

```
> mydata=read.csv("C:\\Users\\admin\\Desktop\\mokesh\\workdata.csv")
```

```
# select your file based on your path
```