**R Studio:-**

R is a programming language and software environment for statistical analysis, graphics representation and reporting. The core of R is an interpreted computer language which allows branching and looping as well as modular programming using functions is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac. R is free software distributed under a GNU-style copy left, and an official part of the GNU project called GNU S.

**Features:-**

* R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
* R has an effective data handling and storage facility,
* R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
* R provides a large, coherent and integrated collection of tools for data analysis.
* R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

**Data Types in R:-**

* + Vectors
  + Lists
  + Matrices
  + Arrays
  + Factors
  + Data Frames

**Vectors:-**

Vectors are the most basic R data objects and there are six types of atomic vectors. They are logical, integer, double, complex, character and raw.

**Creation:-**

**Single Element Vector:-**

**Atomic vector of type character:-**

print("abc") => "abc"

**Atomic vector of type double:-**

print(12.5) => 12.5

**Atomic vector of type integer:-**

print(63L) => 63

**Atomic vector of type logical:-**

print(TRUE) => TRUE

**Atomic vector of type complex:-**

print(2+3i) => 2+3i

**Atomic vector of type raw:-**

print(charToRaw('hello')) => 68 65 6c 6c 6f

**Multiple Elements Vector**

**Using colon operator with numeric data:-**

**Creating a sequence from 5 to 13**

v <- 5:13

print(v) => 5 6 7 8 9 10 11 12 13

**Creating a sequence from 6.6 to 12.6**

v <- 6.6:12.6

print(v) => 6.6 7.6 8.6 9.6 10.6 11.6 12.6

**If the final element specified does not belong to the sequence then it is discarded**

v <- 3.8:11.4

print(v) => 3.8 4.8 5.8 6.8 7.8 8.8 9.8 10.8

**Using the c() function:-**

The non-character values are coerced to character type if one of the elements is a character.

The logical and numeric values are converted to characters.

s <- c('apple','red',5,TRUE)

print(s) => "apple" "red" "5" "TRUE"

**Vector Manipulation:-**

**Vector arithmetic**

Two vectors of same length can be added, subtracted, multiplied or divided giving the result as a vector output.

**Vector Element Recycling**

If we apply arithmetic operations to two vectors of unequal length, then the elements of the shorter vector are recycled to complete the operations.

**Lists:-**

Lists are the R objects which contain elements of different types like − numbers, strings, vectors and another list inside it. A list can also contain a matrix or a function as its elements. List is created using list() function.

**Create a list:-**

**Create a list containing strings, numbers, vectors and a logical values:-**

list\_data <- list("Red", "Green", c(21,32,11), TRUE, 51.23, 119.1)

**Accessing List Elements:-**

Elements of the list can be accessed by the index of the element in the list. In case of named lists it can also be accessed using the names.

**Converting List to Vector:-**

A list can be converted to a vector so that the elements of the vector can be used for further manipulation. All the arithmetic operations on vectors can be applied after the list is converted into vectors. To do this conversion, we use the unlist() function. It takes the list as input and produces a vector.

**Matrices:-**

Matrices are the R objects in which the elements are arranged in a two-dimensional rectangular layout. They contain elements of the same atomic types. Though we can create a matrix containing only characters or only logical values, they are not of much use. We use matrices containing numeric elements to be used in mathematical calculations.

A Matrix is created using the matrix() function.

Syntax

The basic syntax for creating a matrix in R is −

**matrix (data, nrow, ncol, byrow, dimnames)**

**data** is the input vector which becomes the data elements of the matrix.

**nrow** is the number of rows to be created.

**ncol** is the number of columns to be created.

**byrow** is a logical clue. If TRUE then the input vector elements are arranged by row.

**dimname** is the names assigned to the rows and columns.

**Array:-**

Arrays are the R data objects which can store data in more than two dimensions. For example − If we create an array of dimension (2, 3, 4) then it creates 4 rectangular matrices each with 2 rows and 3 columns. Arrays can store only data type.

An array is created using the array() function. It takes vectors as input and uses the values in the dim parameter to create an array.

**Factors:-**

Factors are the data objects which are used to categorize the data and store it as levels. They can store both strings and integers. They are useful in the columns which have a limited number of unique values. Like "Male, "Female" and True, False etc. They are useful in data analysis for statistical modeling.

Factors are created using the factor () function by taking a vector as input.

**Data Frames:-**

A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

Following are the characteristics of a data frame.

* The column names should be non-empty.
* The row names should be unique.
* The data stored in a data frame can be of numeric, factor or character type.
* Each column should contain same number of data items.

**Pie Charts:-**

R Programming language has numerous libraries to create charts and graphs. A pie-chart is a representation of values as slices of a circle with different colors. The slices are labeled and the numbers corresponding to each slice is also represented in the chart.In R the pie chart is created using the pie() function which takes positive numbers as a vector input. The additional parameters are used to control labels, color, title.

**Parameters:-**

* **x** is a vector containing the numeric values used in the pie chart.
* **labels** is used to give description to the slices.
* **radius** indicates the radius of the circle of the pie chart.(value between −1 and +1).
* **main** indicates the title of the chart.
* **col** indicates the color palette.
* **clockwise** is a logical value indicating if the slices are drawn clockwise or anti clockwise.

**Bar Charts:-**

A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R uses the function barplot() to create bar charts. R can draw both vertical and Horizontal bars in the bar chart. In bar chart each of the bars can be given different colors.

**Syntax:-**

barplot(H,xlab,ylab,main, names.arg,col)

* **parameters:-**
* **H** is a vector or matrix containing numeric values used in bar chart.
* **xlab** is the label for x axis.
* **ylab** is the label for y axis.
* **main** is the title of the bar chart.
* **names.arg** is a vector of names appearing under each bar.
* **col** is used to give colors to the bars in the graph.

**Boxplots:-**

Boxplots are a measure of how well distributed is the data in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.Boxplots are created in R by using the boxplot() function.

**Syntax:-**

boxplot(x, data, notch, varwidth, names, main)

**Parameters:-**

**x** is a vector or a formula.

**data** is the data frame.

**notch** is a logical value. Set as TRUE to draw a notch.

**varwidth** is a logical value. Set as true to draw width of the box proportionate to the sample size.

**names** are the group labels which will be printed under each boxplot.

**main** is used to give a title to the graph.

**Histograms:-**

A histogram represents the frequencies of values of a variable bucketed into ranges. Histogram is similar to bar chat but the difference is it groups the values into continuous ranges. Each bar in histogram represents the height of the number of values present in that range.

R creates histogram using hist() function. This function takes a vector as an input and uses some more parameters to plot histograms.

**Syntax:-**

hist(v,main,xlab,xlim,ylim,breaks,col,border)

**Parameter:-**

**v** is a vector containing numeric values used in histogram.

**main** indicates title of the chart.

**col** is used to set color of the bars.

**border** is used to set border color of each bar.

**xlab** is used to give description of x-axis.

**xlim** is used to specify the range of values on the x-axis.

**ylim** is used to specify the range of values on the y-axis.

**breaks** is used to mention the width of each bar.

**Line Graphs:-**

A line chart is a graph that connects a series of points by drawing line segments between them. These points are ordered in one of their coordinate (usually the x-coordinate) value. Line charts are usually used in identifying the trends in data.

The plot() function in R is used to create the line graph.

**Syntax:-**

plot(v,type,col,xlab,ylab)

**Parameter:-**

**v** is a vector containing the numeric values.

**type** takes the value "p" to draw only the points, "l" to draw only the lines and "o" to draw both points and lines.

**xlab** is the label for x axis.

**ylab** is the label for y axis.

**main** is the Title of the chart.

**col** is used to give colors to both the points and lines.

**Scatterplots:-**

Scatterplots show many points plotted in the Cartesian plane. Each point represents the values of two variables. One variable is chosen in the horizontal axis and another in the vertical axis.

The simple scatterplot is created using the plot() function.

**Syntax:-**

**plot(x, y, main, xlab, ylab, xlim, ylim, axes)**

**Parameters:-**

**x** is the data set whose values are the horizontal coordinates.

**y** is the data set whose values are the vertical coordinates.

**main** is the tile of the graph.

**xlab** is the label in the horizontal axis.

**ylab** is the label in the vertical axis.

**xlim** is the limits of the values of x used for plotting.

**ylim** is the limits of the values of y used for plotting.

**axes** indicates whether both axes should be drawn on the plot.