Features of R

As stated earlier, R is a programming language and software environment for statistical analysis, graphics representation and reporting.

The following are the important features of R −

- 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 Structures in R Programming

A data structure is a particular way of organizing data in a computer so that it can be used effectively. The idea is to reduce the space and time complexities of different tasks.

Data structures in R programming are tools for holding multiple values.

R's base data structures are often organized by their dimensionality (1D, 2D, or nD) and whether they're homogeneous (all elements must be of the identical type) or heterogeneous (the elements are often of various types).

This gives rise to the six data types which are most frequently utilized in data analysis.

1. **Vectors**
2. **Lists**
3. **Dataframes**
4. **Matrix**
5. **Arrays**
6. **Factors**

Vector:

One of the basic data structures in R is the vector, used to group together multiple values of the same data type.

Ex. A collection of numbers, such as (1,2,3,4,5)

Vectors are one-dimensional data structures.

Vectors can be created using the c() function, like this:

> thisVector <- c(1:60)

Creates a vector containing 60 numbers, from 1 through 60 that is assigned to object 'thisVector'

> characterVector <- c("one", "two", "three")

Creates a vector containing character values, that is assigned to object "**characterVector**"

Vectors have two different flavors: atomic vectors and lists. They have three common properties:

1. Type – Describes what it is (**typeof()**)
2. Length – Tells how many elements it contains (**length()**)
3. Attributes – Gives us information about additional arbitrary metadata (**attributes()**)

List:

In R lists act as containers. Unlike atomic vectors, the contents of a list are not restricted to a single mode and can encompass any mixture of data types.

Lists are sometimes called generic vectors, because the elements of a list can be of any type of R object, even lists containing further lists. This property makes them fundamentally different from atomic vectors.

A list is a special type of vector. Each element can be a different type.

Ex. A collection of object, such as (1,2, “one”, “two”)

List can be created using the list() function, like this:

```
> myList <- list(1, "a", TRUE, 1+4i)
```

Dataframe

Dataframes are generic data objects of R which are used to store data in tabular format i.e., in two dimensions, rows and columns, much like a spreadsheet

They are two-dimensional, heterogeneous data structures. These are lists of vectors of equal lengths.

Data frames have the following constraints placed upon them:

- A data-frame must have column names and every row should have a unique name.
- Each column must have the identical number of items i.e. length of each column should be same.
- Each item in a single column must be of the same data type.
- Different columns may have different data types.

To create a data frame we use the data.frame() function.

The following code creates 3 different vectors named 'rollNo' and 'marks’ and copies them into the data frame named 'result'.

> rollNo <- c(1:3)

> marks <- c(80, 30, 90)

> result <- data.frame(rollNo, marks)

> result

	rollNo	marks
1	1	80
2	2	30
3	3	90

Useful Data Frame Functions

- `head()` - shows first 6 rows
- `tail()` - shows last 6 rows
- `dim()` - returns the dimensions of data frame (i.e. number of rows and number of columns)
- `nrow()` - number of rows
- `ncol()` - number of columns

Matrix

A matrix is a data structure that is something between a vector and a data frame.

Like a vector, it can hold values of only one data type. But, like a data frame, it can store and display that data in tabular format, columns, and rows, like a spreadsheet.

the elements of a matrix must be of the same data type.

Here's how to create a matrix, using the matrix() function:

```
> mat1 <- matrix(nrow = 2, ncol = 2)
> mat1
    [,1] [,2]
[1,]   NA   NA
[2,]   NA   NA

> mat2 <- matrix(1:4, nrow = 2, ncol = 2)
    [,1] [,2]
[1,]   1   3
[2,]   2   4
```

Array:

Arrays are the R data objects which store the data in more than two dimensions.

Arrays are n-dimensional data structures. For example, if we create an array of dimensions (2, 3, 3) then it creates 3 rectangular matrices each with 2 rows and 3 columns. They are homogeneous data structures.

An array is created using the *array()* function. We can use vectors as input. To create an array, we can use these values in the dim parameter.

For example:

In this following example, we will create an array in R of two 3×3 matrices each with 3 rows and 3 columns.

```
# Create two vectors of different lengths.
> output <- array(  c(1,2,3,4,5,6,7,8),   #Taking sequence of elements
                    dim = c(2,2,2))     #Creating two rectangular matrices
                                        #Each with two rows and two columns
> output
, , 1

     [,1] [,2]
[1,]    1    3
[2,]    2    4


, , 2
```

```
     [,1] [,2]
[1,]    5    7
[2,]    6    8
```

Factor:

Factors are the data objects which are used to categorize the data and store it as levels.

They are useful for storing categorical data.

They can store both strings and integers. They are useful to categorize unique values in columns like “TRUE” or “FALSE”, or “MALE” or “FEMALE”, etc..

They are useful in data analysis for statistical modeling.

To create a factor in R you need to use the function called factor(). The argument to this factor() is the vector.

> fac <- factor(c(“male”, “female”, “male”, “male”, “male”, “female”))

> fact

[1] Male Female Male Male Female Male Female

Levels: Female Male

DATA VISUALIZATION

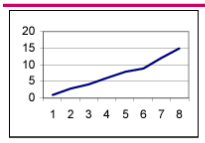
Data visualization is the process of creating graphical representations of information. This process helps the presenter communicate data in a way that's easy for the viewer to interpret and draw conclusions.

DATA VISUALIZATION TECHNIQUES

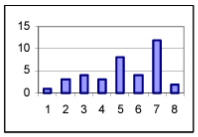
The type of data visualization technique you leverage will vary based on the type of data you're working with, in addition to the story you're telling with your data.

Here are some important data visualization techniques to know:

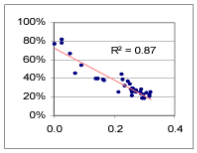
- Pie Chart
- Bar Chart
- Histogram
- Gantt Chart
- Heat Map
- Scatter Plot
- Area Chart
- Timeline
- Box and Whisker Plot
- Waterfall Chart
- Pictogram Chart
- Highlight Table
- Bullet Graph
- Choropleth Map
- Word Cloud
- Network Diagram
- Correlation Matrices



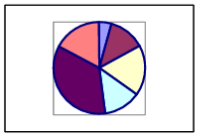
Line Graph
- x-axis requires quantitative variable
- Variables have contiguous values
- Familiar/conventional ordering among ordinals



Bar Graph
- Comparison of relative point values



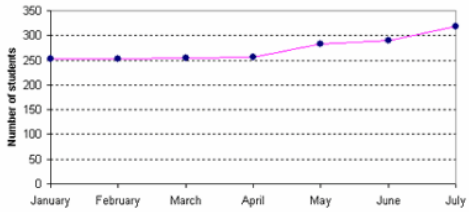
Scatter Plot
- Convey overall impression of relationship between two variables



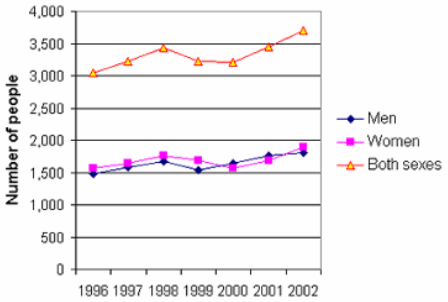
Pie Chart
- Emphasizing differences in proportion among a few numbers

Line Graph – Trend visualization

- Fundamental technique of data presentation
- Used to compare two variables
  - X-axis is often the control variable
  - Y-axis is the response variable
- Good at:
  - Showing specific values
  - Trends
  - Trends in groups (using multiple line graphs)



Students participating in sporting activities



Mobile Phone use

Note: graph labelling is fundamental

Bar Plot

Bar Plot (or barchart) is one of the most common type of graphic.

It shows the relationship between a numeric variable and a categoric variable.

Bar Plot are classified into four types of graphs - bar graph or bar chart, line graph, pie chart, and diagram.

Limitations of Bar Plot:

When we try to display changes in speeds such as acceleration, Bar graphs wont help us.

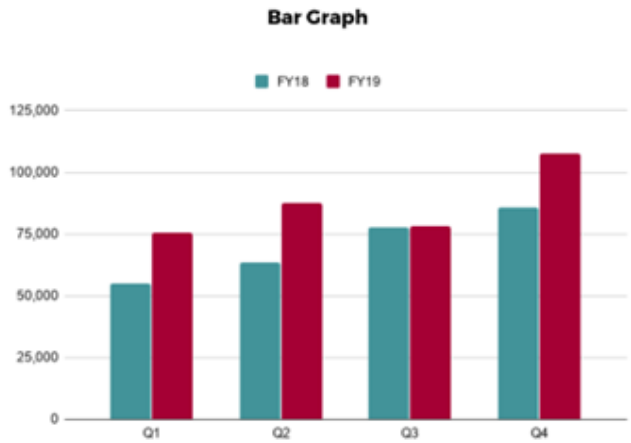
Advantages of Bar plot:

• Bar charts are easy to understand and interpret.

• Relationship between size and value helps for in easy comparison.

• They're simple to create.

• They can help in presenting very large or very small values easily.



Histogram

A histogram represents the frequency distribution of continuous variables. while, a bar graph is a diagrammatic comparison of discrete variables.

Histogram presents numerical data whereas bar graph shows categorical data. The histogram is drawn in such a way that there is no gap between the bars.

Limitations of Histogram:

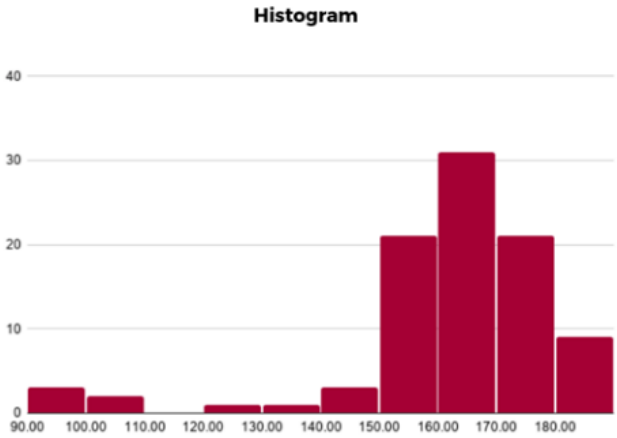
A histogram can present data that is misleading as it has many bars.

Only two sets of data are used, but to analyse certain types of statistical data, more than two sets of data are necessary

Advantages of Histogram:

Histogram helps to identify different data, the frequency of the data occurring in the dataset and categories which are difficult to interpret in a tabular form.

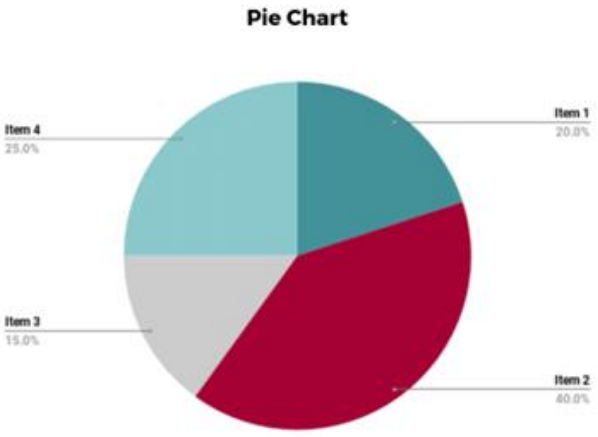
It helps to visualize the distribution of the data.



Pie charts

Pie charts are one of the most common and basic data visualization techniques, used across a wide range of applications. Pie charts are ideal for illustrating proportions, or part-to-whole comparisons.

Because pie charts are relatively simple and easy to read, they're best suited for audiences who might be unfamiliar with the information or are only interested in the key takeaways. For viewers who require a more thorough explanation of the data, pie charts fall short in their ability to display complex information.



Scatter Plot :

A Scatter Plot is a graph in which the values of two variables are plotted along two axes, the pattern of the resulting points revealing any correlation present. With scatter plots we can explain how the variables relate to each other. Which is defined as correlation. Positive, Negative, and None (no correlation) are the three types of correlation.

Advantages of a Scatter Diagram

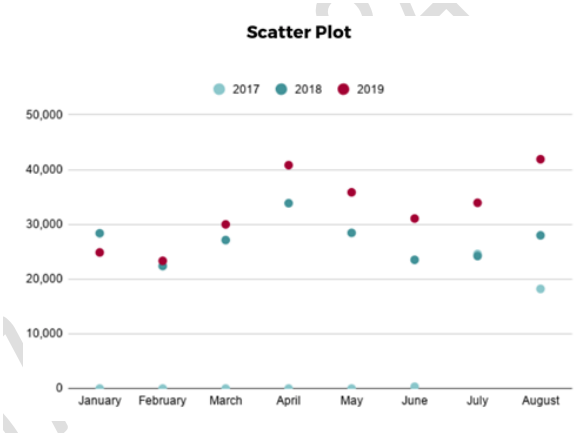
• Relationship between two variables can be viewed.

• For non-linear pattern, this is the best method.

• Maximum and minimum value can be easily determined.

• Observation and reading is easy to understand

• Plotting the diagram is very simple.



Limitations of a Scatter Diagram

• With Scatter diagrams we cannot get the exact extent of correlation.

• Quantitative measure of the relationship between the variable cannot be viewed. Only shows the quantitative expression.

• The relationship can only show for two variables.

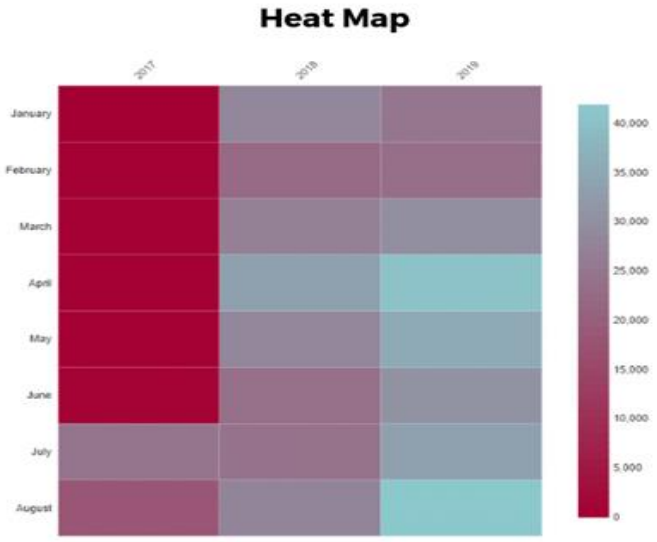
Gantt Chart

Gantt charts are particularly common in project management, as they're useful in illustrating a project timeline or progression of tasks. In this type of chart, tasks to be performed are listed on the vertical axis and time intervals on the horizontal axis. Horizontal bars in the body of the chart represent the duration of each activity.

Utilizing Gantt charts to display timelines can be incredibly helpful, and enable team members to keep track of every aspect of a project. Even if you're not a project management professional, familiarizing yourself with Gantt charts can help you stay organized.

Heat Map

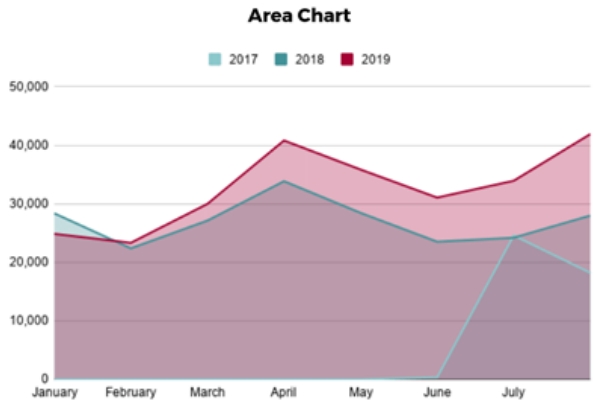
A heat map is a type of visualization used to show differences in data through variations in color. These charts use color to communicate values in a way that makes it easy for the viewer to quickly identify trends. Having a clear legend is necessary in order for a user to successfully read and interpret a heatmap. There are many possible applications of heat maps. For example, if you want to analyze which time of day a retail store makes the most sales, you can use a heat map that shows the day of the week on the vertical axis and time of day on the horizontal axis. Then, by shading in the matrix with colors that correspond to the number of sales at each time of day, you can identify trends in the data that allow you to determine the exact times your store experiences the most sales.



Area Chart

An area chart, or area graph, is a variation on a basic line graph in which the area underneath the line is shaded to represent the total value of each data point. When several data series must be compared on the same graph, stacked area charts are used.

This method of data visualization is useful for showing changes in one or more quantities over time, as well as showing how each quantity combines to make up the whole. Stacked area charts are effective in showing part-to-whole comparisons.



Timeline

Timelines are the most effective way to visualize a sequence of events in chronological order. They're typically linear, with key events outlined along the axis. Timelines are used to communicate time-related information and display historical data.

Timelines allow you to highlight the most important events that occurred, or need to occur in the future, and make it easy for the viewer to identify any patterns appearing within the selected time period. While timelines are often relatively simple linear visualizations, they can be made more visually appealing by adding images, colors, fonts, and decorative shapes.