Experiment-09

Table of Contents

Aim:

Implementation of Data Visualization in R.

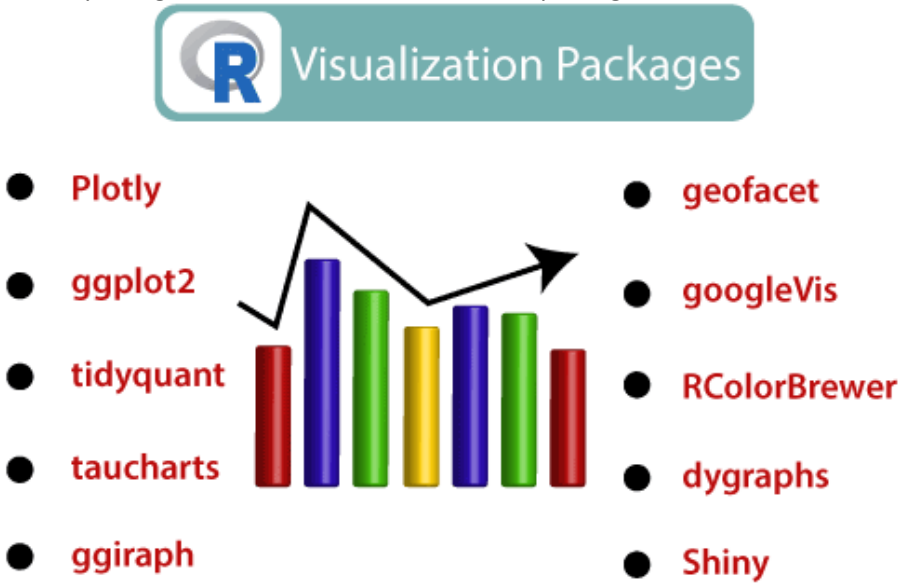
Procedure:

R Data Visualization:

In R, we can create visually appealing data visualizations by writing few lines of code. For this purpose, we use the diverse functionalities of R. Data visualization is an efficient technique for gaining insight about data through a visual medium. With the help of visualization techniques, a human can easily obtain information about hidden patterns in data that might be neglected.
By using the data visualization technique, we can work with large datasets to efficiently obtain key insights about it.

R Visualization Packages:

R provides a series of packages for data visualization. These packages are as follows:

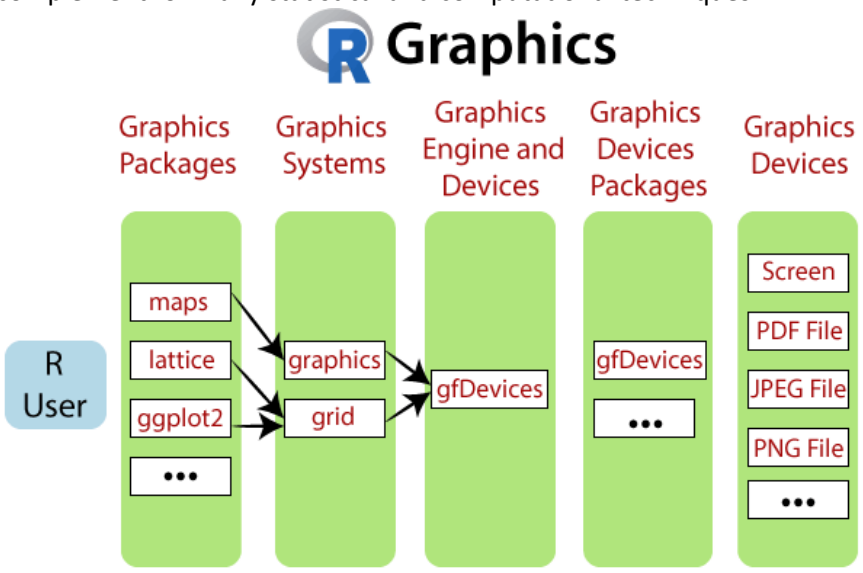


1) plotly : The plotly package provides online interactive and quality graphs. This package extends upon the JavaScript library ? plotly.js.
2) ggplot2 : R allows us to create graphics declaratively. R provides the ggplot package for this purpose. This package is famous for its elegant and quality graphs, which sets it apart from other visualization packages.
3) tidyquant : The tidyquant is a financial package that is used for carrying out quantitative financial analysis. This package adds under tidyverse universe as a financial package that is used for importing, analyzing, and visualizing the data.
4) taucharts : Data plays an important role in taucharts. The library provides a declarative interface for rapid mapping of data fields to visual properties.
5) ggiraph : It is a tool that allows us to create dynamic ggplot graphs. This package allows us to add tooltips, JavaScript actions, and animations to the graphics.
6) geofacets : This package provides geofaceting functionality for 'ggplot2'. Geofaceting arranges a sequence of plots for different geographical entities into a grid that preserves some of the geographical orientation.
7) googleVis : googleVis provides an interface between R and Google's charts tools. With the help of this package, we can create web pages with interactive charts based on R data frames.

8) RColorBrewer : This package provides color schemes for maps and other graphics, which are designed by Cynthia Brewer.
9) dygraphs : The dygraphs package is an R interface to the dygraphs JavaScript charting library. It provides rich features for charting time-series data in R.
10) shiny : R allows us to develop interactive and aesthetically pleasing web apps by providing a shiny package. This package provides various extensions with HTML widgets, CSS, and JavaScript.

R Graphics

Graphics play an important role in carrying out the important features of the data. Graphics are used to examine marginal distributions, relationships between variables, and summary of very large data. It is a very important complement for many statistical and computational techniques.



Standard Graphics

R standard graphics are available through package graphics, include several functions which provide statistical plots, like:

- Scatterplots
- Piecharts
- Boxplots
- Barplots etc.

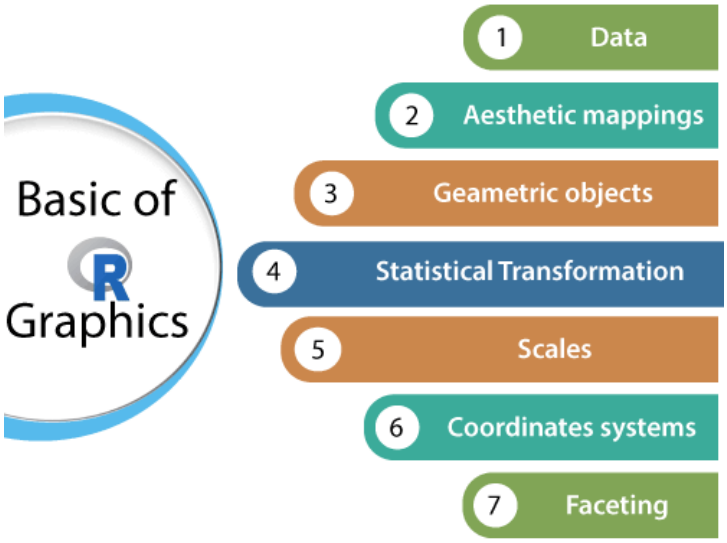
We use the above graphs that are typically a single function call.

Graphics Devices

It is something where we can make a plot to appear. A graphics device is a window on your computer (screen device), a PDF file (file device), a Scalable Vector Graphics (SVG) file (file device), or a PNG or JPEG file (file device).
There are some of the following points which are essential to understand:
The functions of graphics devices produce output, which depends on the active graphics device.
A screen is the default and most frequently used device.
R graphical devices such as the PDF device, the JPEG device, etc. are used.
We just need to open the graphics output device which we want. Therefore, R takes care of producing the type of output which is required by the device.
For producing a certain plot on the screen or as a GIF R graphics file, the R code should exactly be the same. We only need to open the target output device before.
Several devices can be open at the same time, but there will be only one active device.

The basics of the grammar of graphics

There are some key elements of a statistical graphic. These elements are the basics of the grammar of graphics. Let's discuss each of the elements one by one to gain the basic knowledge of graphics.



1) Data: Data is the most crucial thing which is processed and generates an output.
2) Aesthetic Mappings: Aesthetic mappings are one of the most important elements of a statistical graphic. It controls the relation between graphics variables and data variables. In a scatter plot, it also helps to map the temperature variable of a data set into the X variable.
In graphics, it helps to map the species of a plant into the color of dots.
3) Geometric Objects: Geometric objects are used to express each observation by a point using the aesthetic mappings. It maps two variables in the data set into the x,y variables of the plot.
4) Statistical Transformations: Statistical transformations allow us to calculate the statistical analysis of the data in the plot.The statistical transformation uses the data and approximates it with the help of a regression line having x,y coordinates, and counts occurrences of certain values.
5) Scales: It is used to map the data values into values present in the coordinate system of the graphics device.
6) Coordinate system: The coordinate system plays an important role in the plotting of the data.

- Cartesian
- Plot

7) Faceting: Faceting is used to split the data into subgroups and draw sub-graphs for each group.

Advantages of Data Visualization in R

1. Understanding: It can be more attractive to look at the business. And, it is easier to understand through graphics and charts than a written document with text and numbers. Thus, it can attract a wider range of audiences. Also, it promotes the widespread use of business insights that come to make better decisions.
2. Efficiency: Its applications allow us to display a lot of information in a small space. Although, the decision-making process in business is inherently complex and multifunctional, displaying evaluation findings in a graph can allow companies to organize a lot of interrelated information in useful ways.
3. Location: Its app utilizing features such as Geographic Maps and GIS can be particularly relevant to wider business when the location is a very relevant factor. We will use maps to show business insights from various locations, also consider the seriousness of the issues, the reasons behind them, and working groups to address them.
4. libraries: R offers a broad collection of visualization libraries along with extensive online guidance on their usage.
5. visualization: R also offers data visualization in the form of 3D models and multipaned charts.
Through R, we can easily customize our data visualization by changing axes, fonts, legends, annotations, and labels.

Disadvantages of Data Visualization in R

1. Cost: R application development range a good amount of money. It may not be possible, especially for small companies, that many resources can be spent on purchasing them. To generate reports, many companies may employ professionals to create charts that can increase costs. Small enterprises are often operating in resource-limited settings and are also receiving timely evaluation results that can often be of high importance.
2. Distraction: However, at times, data visualization apps create highly complex and fancy graphics-rich reports and charts, which may entice users to focus more on the form than the function. If we first add visual appeal, then the overall value of the graphic representation will be minimal. In resource-setting, it is required to understand how resources can be best used. And it is also not caught in the graphics trend without a clear purpose.
3. Standalone: R is only preferred for data visualization when done on an individual standalone server.
4. visualization: Data visualization using R is slow for large amounts of data as compared to other counterparts.

Application Areas:

- Presenting analytical conclusions of the data to the non-analysts departments of your company.
- Health monitoring devices use data visualization to track any anomaly in blood pressure, cholesterol and others.
- To discover repeating patterns and trends in consumer and marketing data.
- Meteorologists use data visualization for assessing prevalent weather changes throughout the world.
- Real-time maps and geo-positioning systems use visualization for traffic monitoring and estimating travel time.

Example

Consider the following airquality data set for visualization in R:

Ozone	Solar R.	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5
28	NA	14.9	66	5	6

Types of Data Visualizations:

Some of the various types of visualizations offered by R are

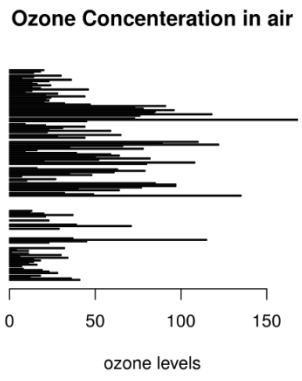
Bar Plot

There are two types of bar plots- horizontal and vertical which represent data points as horizontal or vertical bars of certain lengths proportional to the value of the data item. They are generally used for continuous and categorical variable plotting. By setting the horiz parameter to true and false, we can get horizontal and vertical bar plots respectively.

Example 1 for Bar Plot:

```
# Horizontal Bar Plot for
# Ozone concentration in air
barplot(airquality$Ozone,
        main = 'Ozone Concenteration in air',
        xlab = 'ozone levels', horiz = TRUE)
```

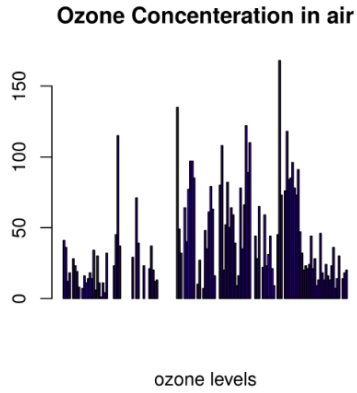
Output:



Example 2 for Bar Plot:

```
# Vertical Bar Plot for
# Ozone concentration in air
barplot(airquality$Ozone, main = 'Ozone Concenteration in air',
        xlab = 'ozone levels', col ='blue', horiz = FALSE)
```

Output:



Bar plots are used for the following scenarios:
To perform a comparative study between the various data categories in the data set.
To analyze the change of a variable over time in months or years.

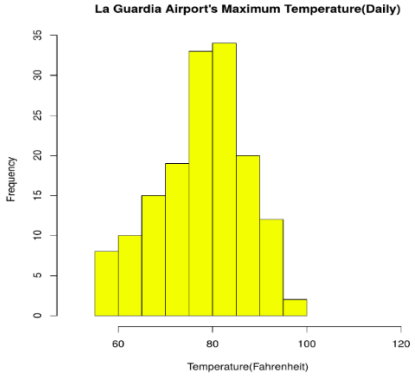
Histogram

A histogram is like a bar chart as it uses bars of varying height to represent data distribution. However, in histogram values are grouped into consecutive intervals called bins. In a Histogram, continuous values are grouped and displayed in these bins whose size can be varied

Example:

```
# Histogram for Maximum Daily Temperature

data(airquality)


hist(airquality$Temp, main ="La Guardia Airport's\

Maximum Temperature(Daily)",

    xlab ="Temperature(Fahrenheit)",

    xlim = c(50, 125), col ="yellow",

    freq = TRUE)
```

Output:



For a histogram, the parameter xlim can be used to specify the interval within which all values are to be displayed.
Another parameter freq when set to TRUE denotes the frequency of the various values in the histogram and when set to FALSE, the probability densities are represented on the y-axis such that they are of the histogram adds up to one.
Histograms are used in the following scenarios:

- To verify an equal and symmetric distribution of the data.
- To identify deviations from expected values.

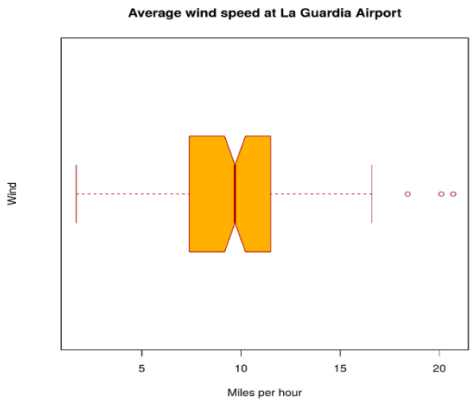
Box Plot:

The statistical summary of the given data is presented graphically using a boxplot. A boxplot depicts information like the minimum and maximum data point, the median value, first and third quartile, and interquartile range.

Example:

```
# Box plot for average wind speed

data(airquality)


boxplot(airquality$Wind, main = "Average wind speed\

at La Guardia Airport",

        xlab = "Miles per hour", ylab = "Wind",

        col = "orange", border = "brown",

        horizontal = TRUE, notch = TRUE)
```

Output:

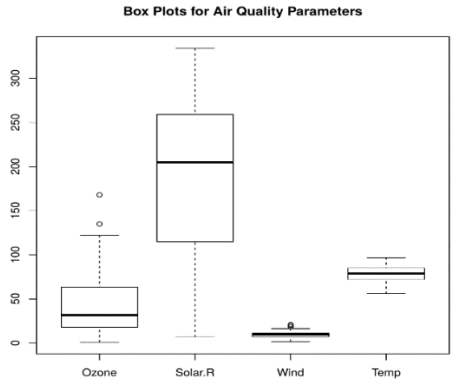


Multiple box plots can also be generated at once through the following code:

Example:

```
# Multiple Box plots, each representing

# an Air Quality Parameter

boxplot(airquality[, 0:4],

        main ='Box Plots for Air Quality Parameters')
```

Output:



Box Plots are used for:

- To give a comprehensive statistical description of the data through a visual cue.
- To identify the outlier points that do not lie in the inter-quartile range of data.

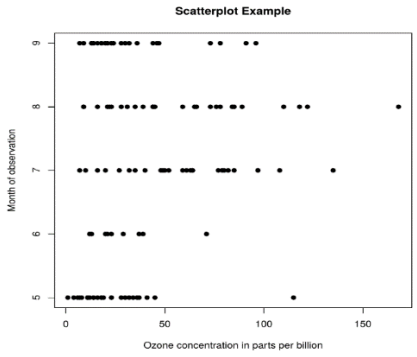
Scatter Plot

A scatter plot is composed of many points on a Cartesian plane. Each point denotes the value taken by two parameters and helps us easily identify the relationship between them.

Example:

```
# Scatter plot for Ozone Concentration per month
data(airquality)


plot(airquality$Ozone, airquality$Month,
     main ="Scatterplot Example",
    xlab ="Ozone Concentration in parts per billion",
    ylab =" Month of observation ", pch = 19)
```

Output:



Scatter Plots are used in the following scenarios:

- To show whether an association exists between bivariate data.
- To measure the strength and direction of such a relationship.

Heat Map:

Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. heatmap() function is used to plot heatmap.

Syntax: heatmap(data)

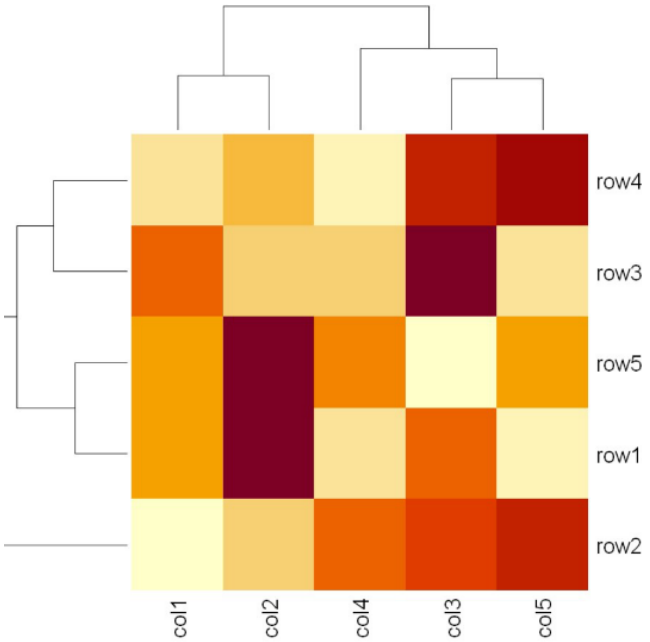
Parameters: data: It represent matrix data, such as values of rows and columns

Return: This function draws a heatmap.

Example:

```
# Set seed for reproducibility

# set.seed(110)
# Create example data
data <- matrix(rnorm(50, 0, 5), nrow = 5, ncol = 5)
# Column names
colnames(data) <- paste0("col", 1:5)

rownames(data) <- paste0("row", 1:5)
# Draw a heatmap

heatmap(data)
```

Output:



Map visualization in R

Here we are using maps package to visualize and display geographical maps using an R programming language.

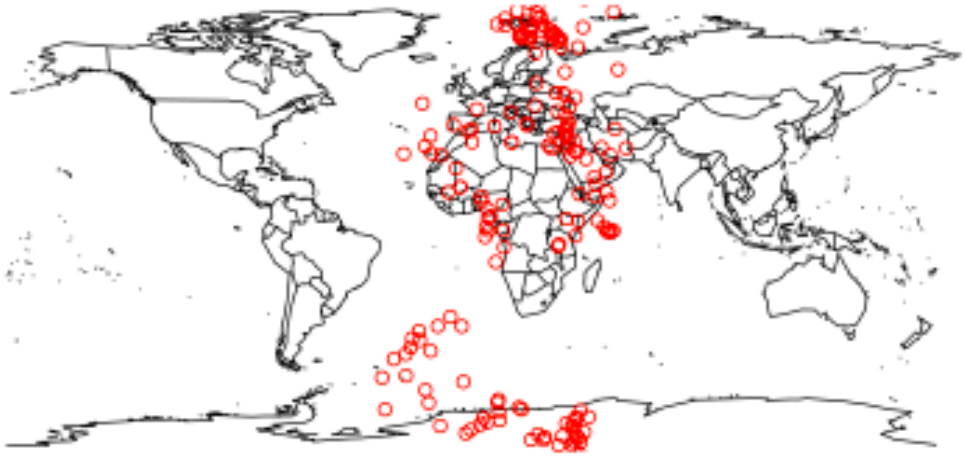
install.packages("maps")

Link of the dataset: worldcities.csv

Example:

```
# Read dataset and convert it into

# Dataframe

data <- read.csv("worldcities.csv")

df <- data.frame(data)

# Load the required libraries

library(maps)

map(database = "world")

# marking points on map

points(x = df$lat[1:500], y = df$lng[1:500], col = "Red")
```

Output:



3D Graphs in R

Here we will use preps() function, This function is used to create 3D surfaces in perspective view. This function will draw perspective plots of a surface over the x–y plane.

Syntax: persp(x, y, z)

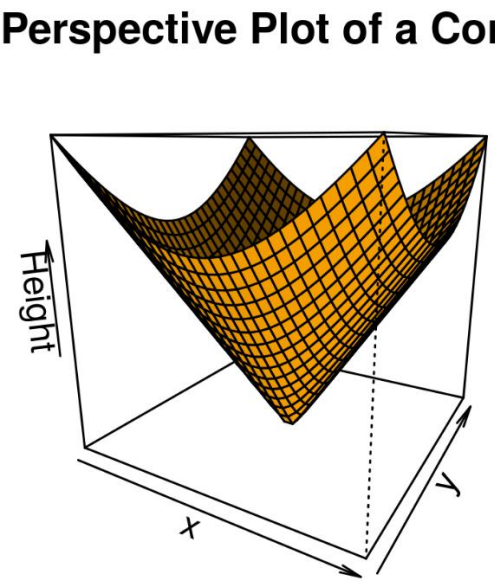
Parameter: This function accepts different parameters i.e. x, y and z where x and y are vectors defining the location along x- and y-axis. z-axis will be the height of the surface in the matrix z.

Return Value: persp() returns the viewing transformation matrix for projecting 3D coordinates (x, y, z) into the 2D plane using homogeneous 4D coordinates (x, y, z, t).

Example:

```
# Adding Titles and Labeling Axes to Plot

cone <- function(x, y){

sqrt(x ^ 2 + y ^ 2)

}

# prepare variables.

x <- y <- seq(-1, 1, length = 30)

z <- outer(x, y, cone)

# plot the 3D surface

# Adding Titles and Labeling Axes to Plot

persp(x, y, z,

main="Perspective Plot of a Cone",

zlab = "Height",

theta = 30, phi = 15,

col = "orange", shade = 0.4)
```

Output:



Exercises

1. Write a R program for Display All X-Axis Labels of Barplot in R
2. Write a R program for Adding Colors to Charts in R Programming
3. Write a R program for How to show legend in heatmap in R?