Data Visualization Using R

Kundan Kumar Jha (22MSM40052)

Abstract

Human eyes are drawn to colors and patterns. We can quickly identify red from blue, and squares from circles.

Our culture is visual, including everything from art and advertisements to TV and movies. Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we quickly see trends and outliers. If you've ever stared at a massive spreadsheet of data and couldn't see a trend, you know how much more effective a visualization can be.

Data visualization is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization software. Data visualizations are surprisingly common in our everyday life, but they often appear in the form of well-known charts and graphs. It can be used to discover unknown facts and trends etc.

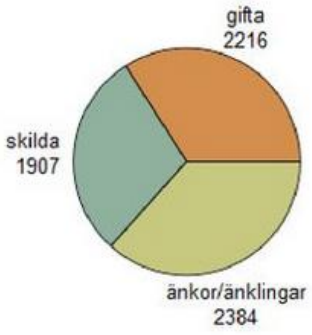
It can give the boost to business owner and stack holder in a way they can understand.

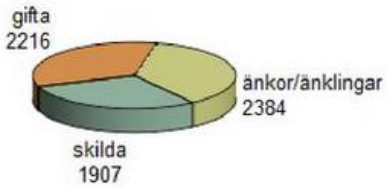
Table of Content

Introduction

Data visualisation is used to communicate information clearly and efficiently to users by the usage of information graphics such as tables and charts. It helps users in analysing a large amount of data in a simple way. It makes computer data more accessible, understandable, and usable It is concerned with the design, development and application of computer generated graphical representation of data. This enables decision makers to see analytics in visual form make it easier for them to make sense of data. It helps them discover patterns, comprehend, information and form an opinion

But Human visual system has limitations. These limitations may lead to wrong/Incomplete analysis of graphs. Misleading graphs may lead to be avoided.





Why there is a need for Data Visualization?

The increased popularity of big data and data analyst projects have made visualisation more important than ever. Companies are increasingly using Machine Learning to gather massive amount of data that can be difficult and slow to sort. Visualisation provides a way to represent the unstructured data in a structured format. That is why visualisation is necessary.

In designing and building a data visualization prototype, one must be guided by how the visualization will be applied. Data visualisation is more than just representing numbers, it involves selecting and rethinking the numbers on which the visualization is based.

DATA VISUALISATION TOOLS:

A **data visualisation** software is a form of software that is designed to visualize data. Each visualisation tool has its own specification but all visualisation software allows you to put input a dataset and visually manipulate it.

Software of the Most Used Data visualisation tools are

- **Tableau:** It is one of the most powerful visualization tool. They are more focused on Business Intelligence. It can be used to create the dashboard for companies which tells how the company is performing in the market. It expensive but it has mobile applications as well. Almost any data can be imported on these software.

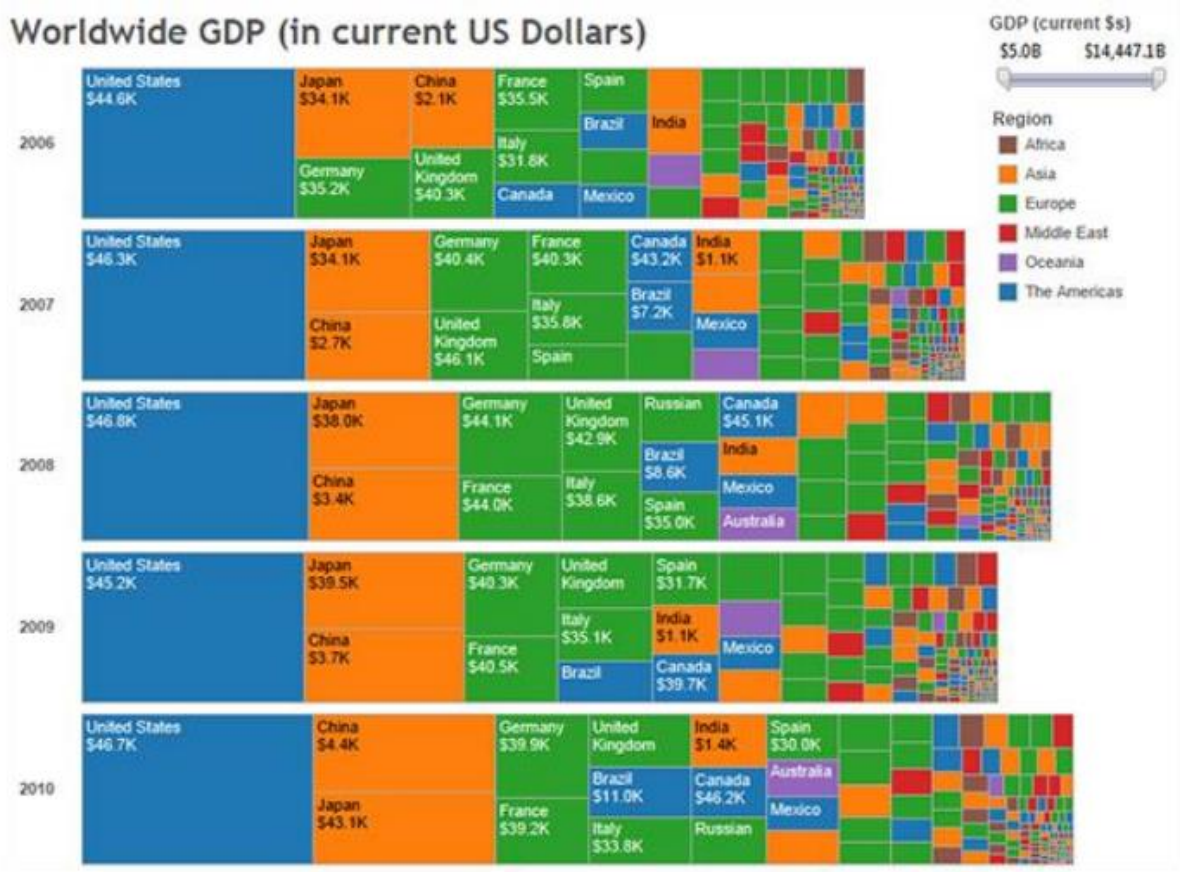
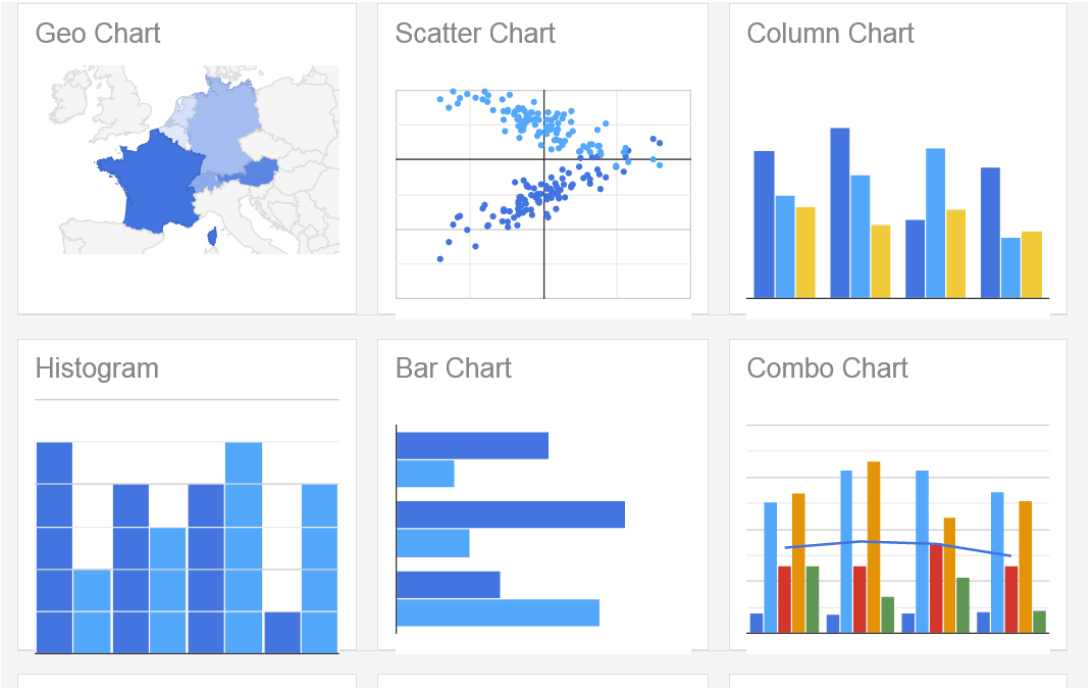


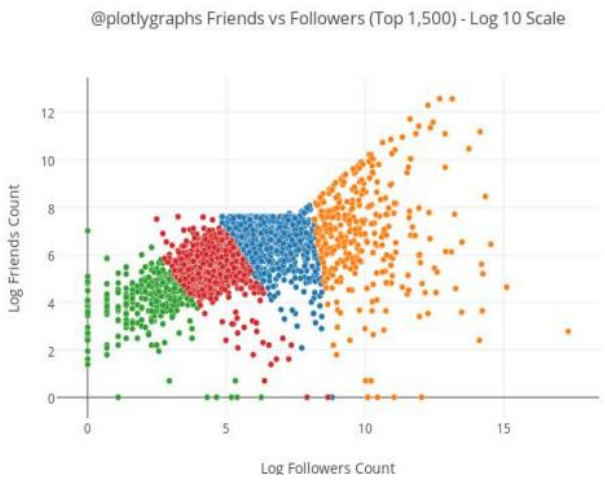
Fig. 4.3 World GDP visualization in Tableau

- **Google charts**: Google charts provides a perfect way to visualize data on website. These charts are pure based on **HTML5/SVG technology,** so no plugins are required.



- **Plotty-** Plotty is a free and online tool with good Graphical User Interface. It has libraries for **Python, R, MATLAB**, etc to visualize the data. It is an open source JavaScript library for creating graphs and customs dashboards.

  The following graph show how many friends they follow and how many followers they have



Implementation of "*different types of visualization"* in R

In R there are various libraries to plot graphs, charts etc like **lattice, highcharter, ggplot2, Plotly etc** but we use the most used library i.e **ggplot2 t**o visualize data on different types of visualization.

(1) Bar Plot: -Bar plot or bar chart is one of the most common type of graph which uses **rectangular bar** to visualize data.

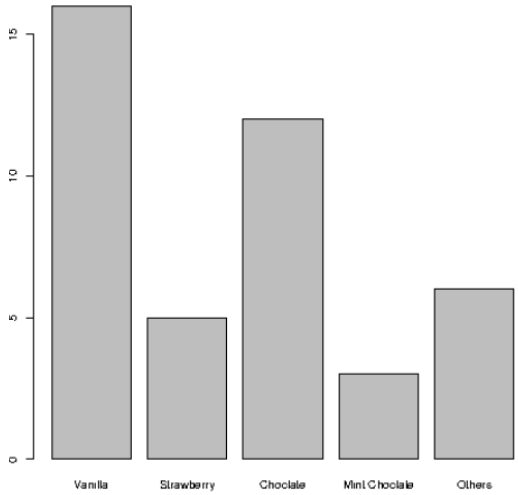
It shows the relationship between the numerical and categorical variable. It can be displayed horizontally or vertically. The height and length of the bar is proportional to the value they represent.

For eg: Consumer were polled about there favourite ice cream flavours in a survey. Draw the bar graph of the following data

Flavours of Icream	Feedback
Vanilla	16
Strawberry	5
Choclate	12
Mini Cholate	3
Others	6

```
# x-axis values
Icecream_flavour <- c("Vanilla", "Strawberry","Choclate", "Mint Choclate", "Others")
# y-axis values
feedback <- c(16, 5, 12, 3, 6 )
# barplot is used to draw bar graph
barplot(y, names.arg = Icecream_flavour)
```

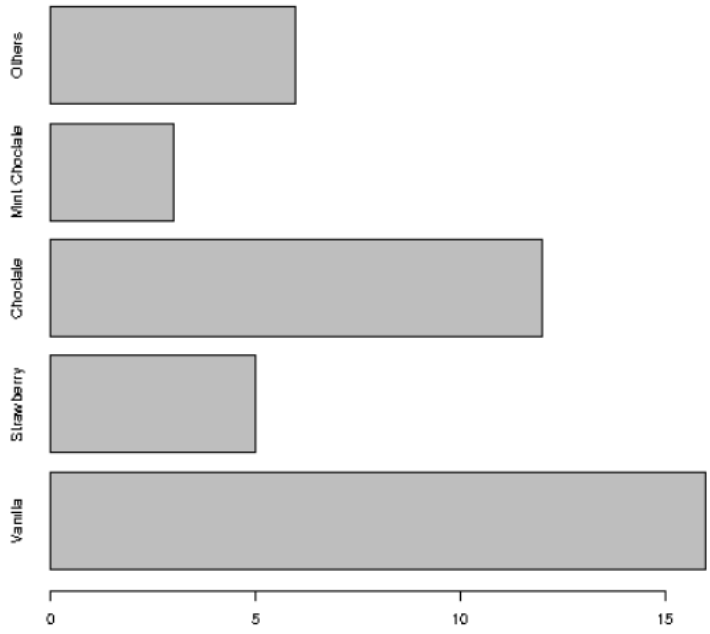
Output:



Horizontal Bar graph can be drawn by setting the "***horiz"*** argument **TRUE**

```
Icecream_flavour<-c("Vanilla","Strawberry","Choclate","Mint Choclate","Others")
# y-axis values
y <- c(16, 5, 12, 3, 6 )
# barplot is used to draw bar graph
barplot(y, names.arg = Icecream_flavour, horiz= TRUE)
```

Output:

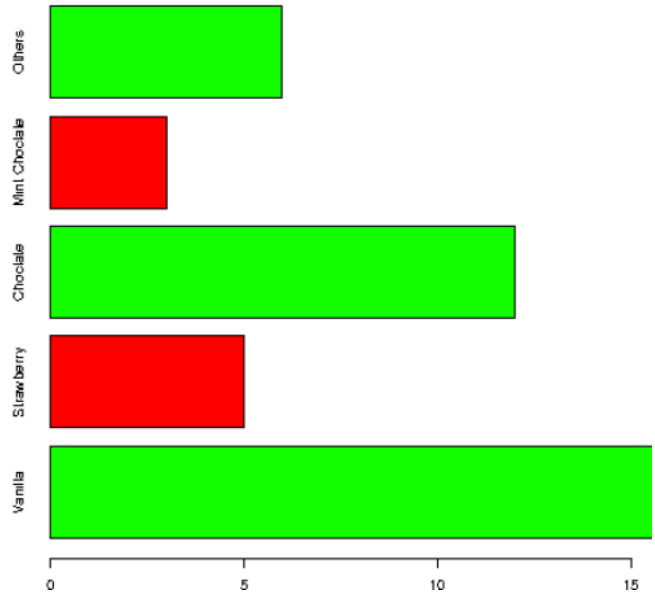


T***o look the graph more effective we can give color to bar***

It can be done by passing "***col argument"*** with the color name that you want to give.

barplot(y, names.arg = Icecream_flavour, horiz= TRUE, **col = c("green","red"))**

OUTPUT:



From the above graph, it is concluded that customers like the *Vanilla flavour* the most.

(2) Pie chart: A Pie Chart is a type of graph that displays data in a circular graph. The pieces of the graph are proportional to the fraction of the whole in each category. In other words, **each slice of the pie is relative to the size of that category** in the group as a whole. The entire "pie" represents 100 percent of a whole, while the pie "slices" represent portions of the whole.

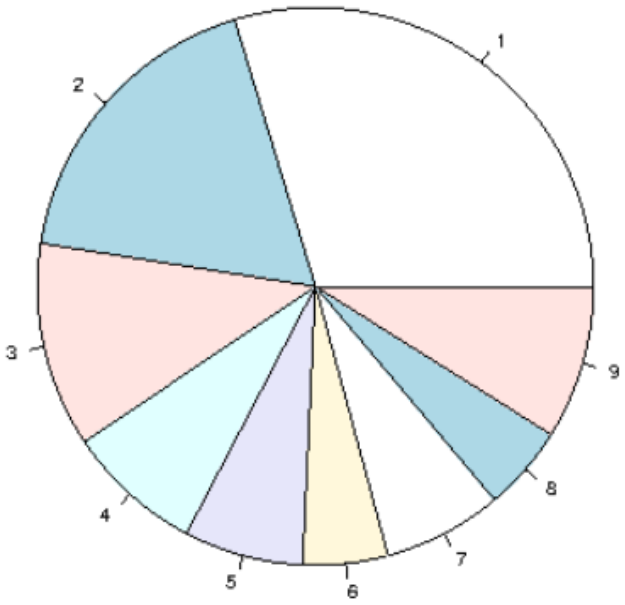
It can be ploted with **pie().**

For eg: Consider the following data

Sate:	Maharastra	West Bengal	Tamil Nadu	U.P.	Karnataka	Bihar	Other State
Popul.:	3000	1800	1200	800	700	500	900

population <- c(3000, 1800, 1200, 800, 700, 500, 700, 500, 900)
pie(population)

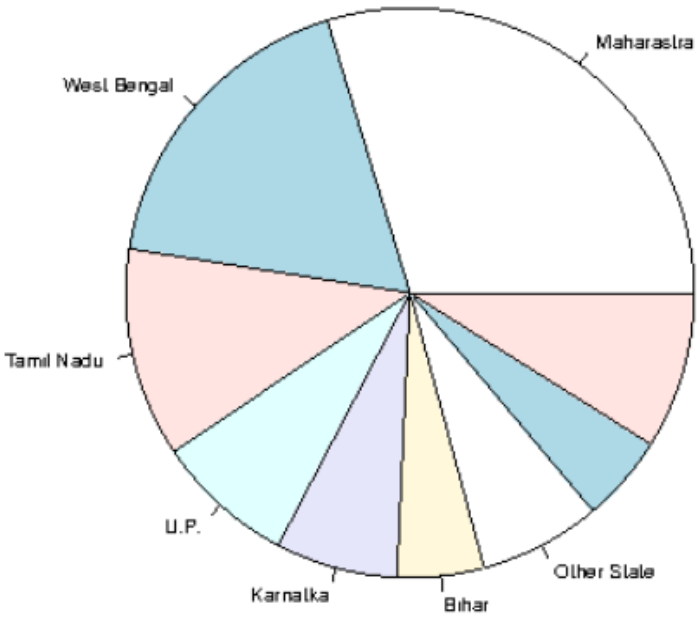
Output:



We can give labels to pie chart by which you know which sector occupies more space – It can be done by passing ***label argument.***

state <- c("Maharastra","West Bengal","Tamil Nadu", "U.P.", "Bihar", "Other state"
population <- c(3000, 1800, 1200, 800, 700, 500, 700, 500, 900)
pie(population, **label = state**)

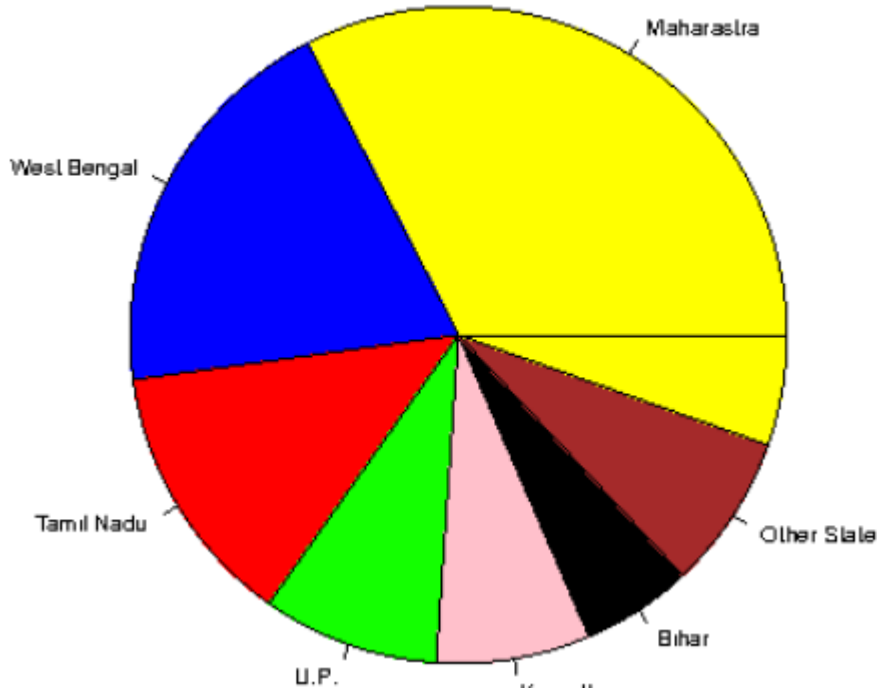
Output:



To make the pie chart attractive we can give the color to slices of the Pie chart: It can be by passing the "**col" argument** with the name of the color that you want to give.

```
state <- c("Maharastra","West Bengal","Tamil Nadu", "U.P.", "Karnatka", "Bihar",
"Other State")
population <- c(3000, 1800, 1200, 800, 700, 500, 700, 500)
color =c("yellow", "blue", "red", "green", "Pink", "black", "brown")
# Display the pie chart
pie(population, label = state, col = color)
```

Output:



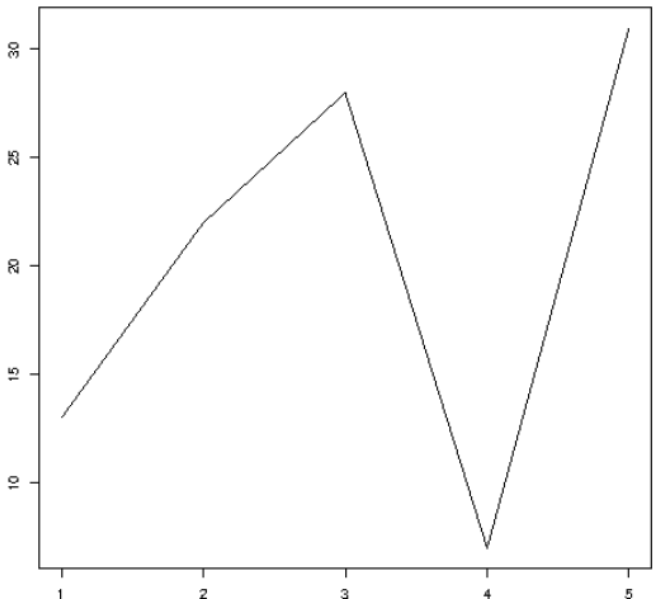
From the above graph, you conclude that *Maharashtra* is the most populated State.

(3) Line Chart: A line chart is a type of chart used to show information that changes over time. When smaller changes exist, line graphs are better to use than bar graph.

It can be draw by **plot()** by passing **type ="l"** argument in function.

```
a <- c(13, 22, 28, 7,31)
plot(a,type = "l")
```

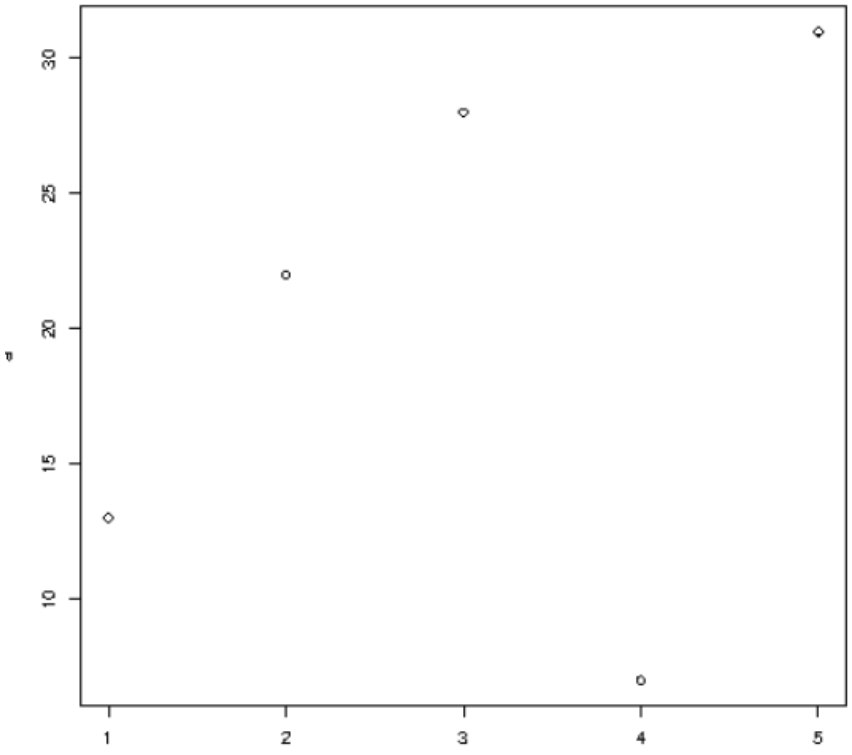
Output:



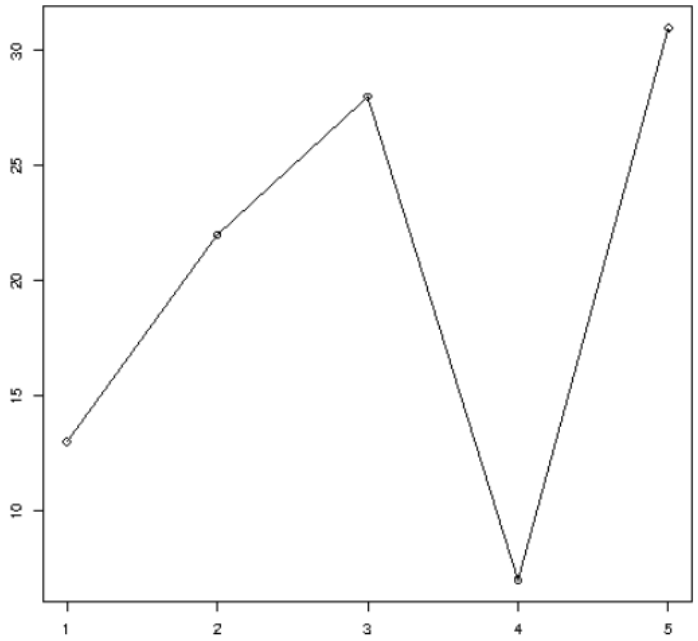
In the above program **type = ”l”** is used to draw a line. If it is not used then we get the output as

```
a <- c(13, 22, 28, 7,31)
plot(a)
```

Output:



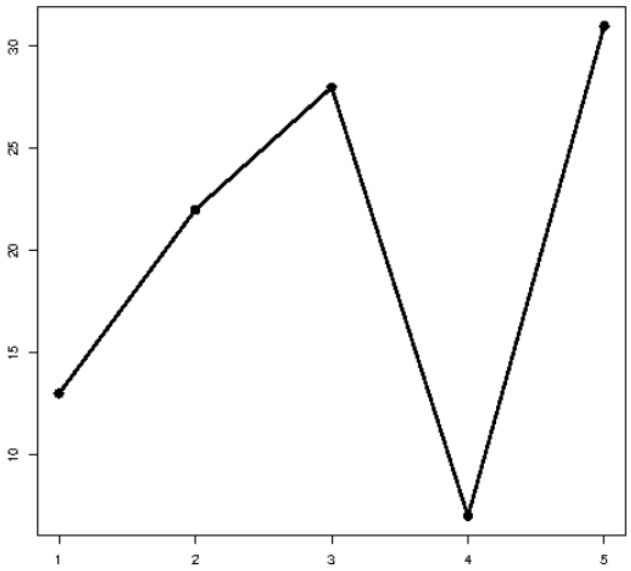
If we use **type = "o"** argument we get the output as



Line width: The width of the line can be changed by the passing ***lwd*** *parameter* (1 is default, while 0.5 means 50% smaller, and 2 means 100% larger)

```
a <- c(13, 22, 28, 7,31)
plot(a, type = "o", lwd = 2)
```

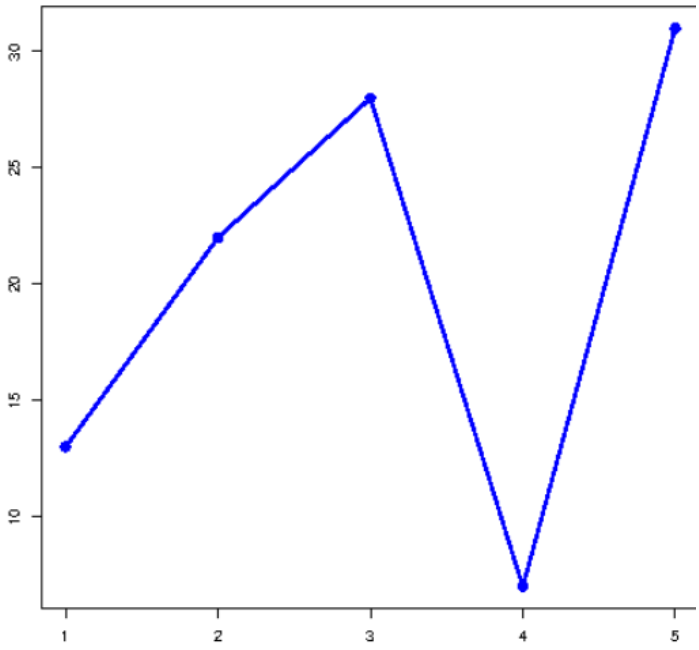
Output:



Line colour: Line colour is black default but it can be changed by passing the ***col*** *argument* in plot() function.

```
a <- c(13, 22, 28, 7,31)
plot(a, type = "o", lwd = 2, col = "blue")
```

Output:

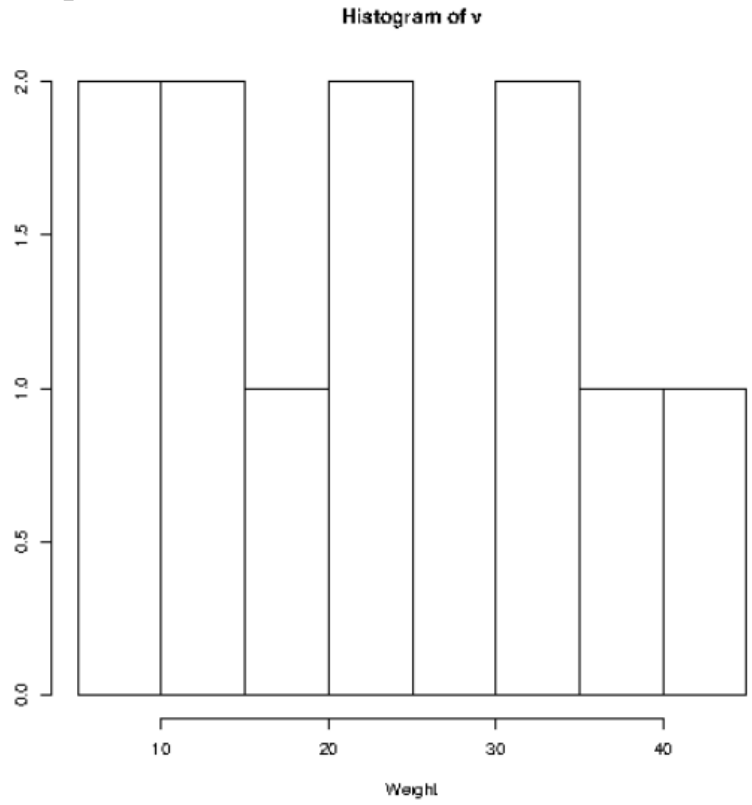


(4) Histogram: A histogram contains a rectangular area to display the statistical information which is proportional to the frequency of a variable and its width depends on the size of numerical intervals.

It can be drawn by **hist()** function**.** It takes vector as input and uses some more parameters to plot diagram

```
v <-  c(9,13,21,8,36,22,12,41,31,33,19)
hist(v, xlab = "Weight")
```

Output:

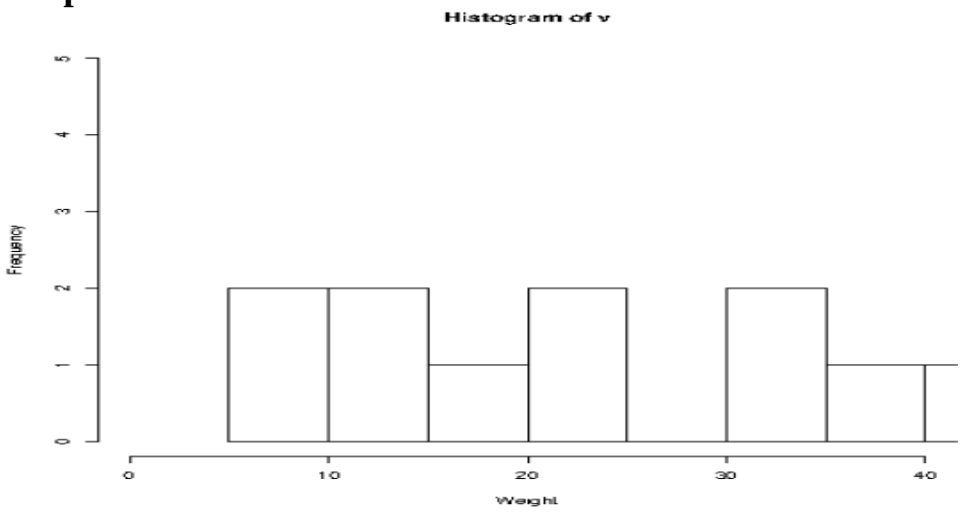


In the above program ***xlab*** *argument* is used to give label to **x-axis.**

Range of X and Y values
For this we use **xlim** and **ylim** parameters in **hist()** function.

v <- c(9,13,21,8,36,22,12,41,31,33,19)
hist(v,xlab = "Weight", **xlim = c(0,40), ylim = c(0,5)**)

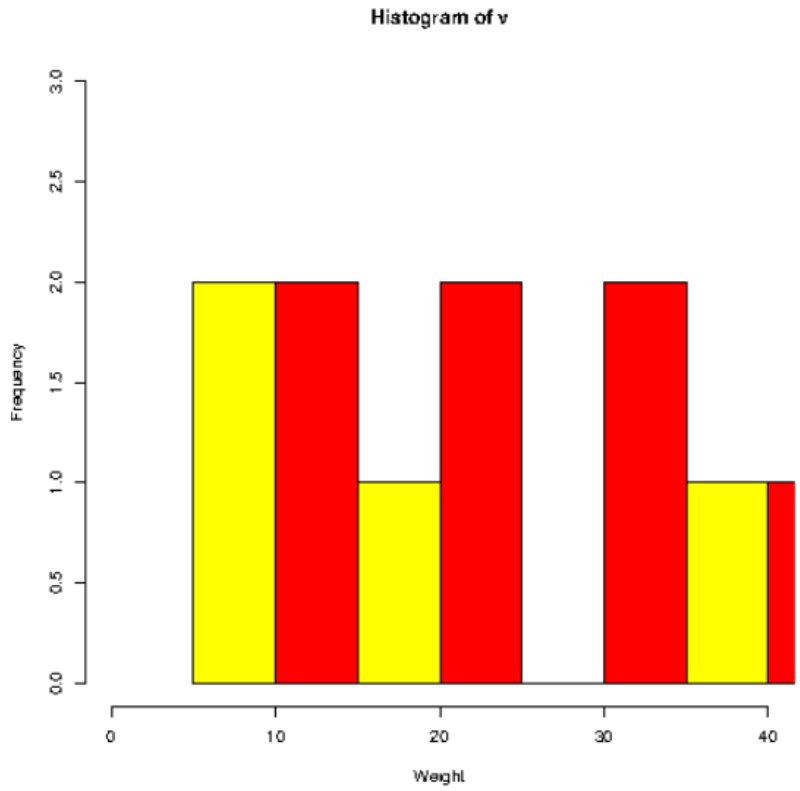
Output:



The above graph can be made more attractive by adding colours to it

```
v <-  c(9,13,21,8,36,22,12,41,31,33,19)
hist(v,xlab = "Weight",xlim = c(0,40), ylim = c(0,3), col= c('Yellow','Red'))
```

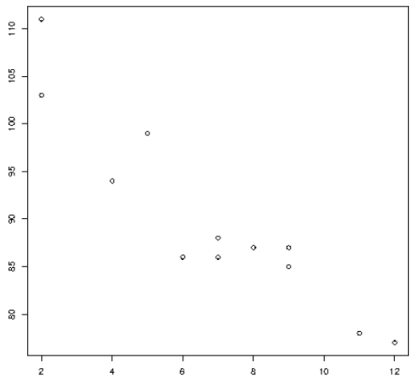
Output:



(5.) Scatter Plot: It is a type of plot used to display the relationship between two variables and plot one variable for each observation. It need two vectors of same length, one for x-axis (horizontal) and y-axis(vertical).

```
x <- c(5,7,8,7,2,2,9,4,11,12,9,6)
y <- c(99,86,87,88,111,103,87,94,78,77,85,86)
plot(x, y)
```

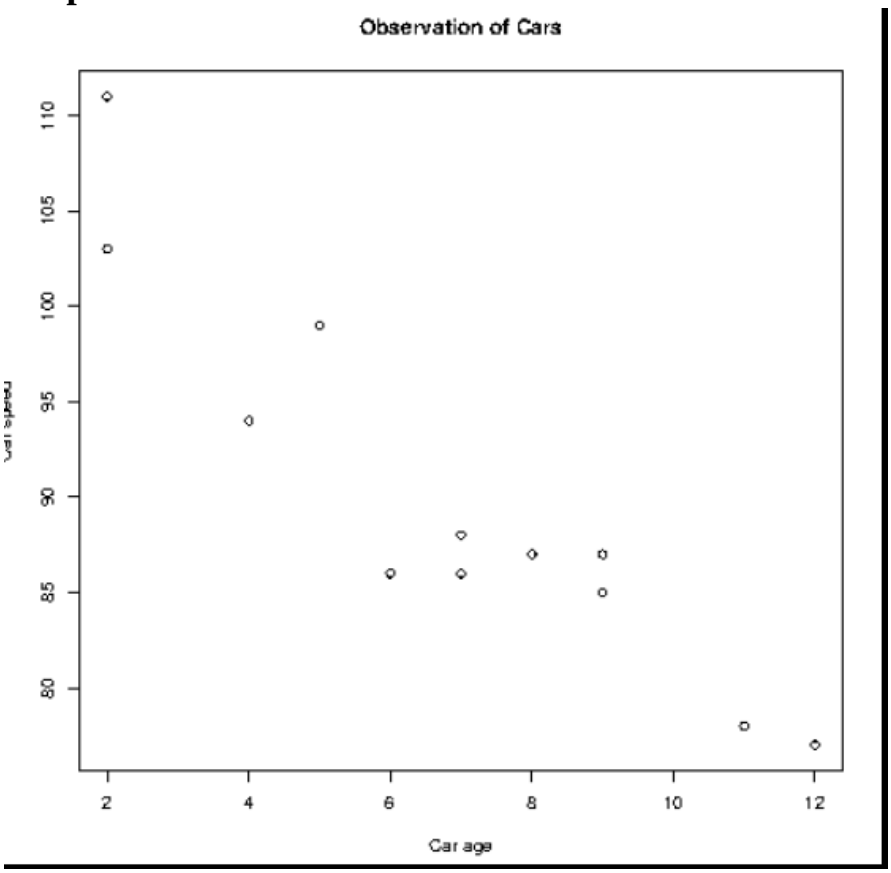
Output:



The graph may not be clear for those who watch it for the first time, so its better to add header and different labels to describe the scatter plot.

plot(x, y, main="Observation of Cars", **xlab="Car age", ylab="Car speed")**

Output:



Application of Data Visualization

(1.) **Data Science:** Data scientists generally create visualisations for their personal use or to communicate information to a small group of people. Visualization libraries for the specified programming languages and tools are used to create the visual representations. Open source programming languages such as Python and R are commonly used by data scientist.

(2.) **Fraud detection:** Data visualization is important in the early stage of fraud investigation. After processing and analysing the data we can visualize the data to determine how many transactions are fraud or not.

(3.) **Education:** The use of data visualisation help teachers to monitor the student progress throughout the semester.

(4.) **Marketing:** In marketing analysis, we may use visuals and reports to analyse various patterns and trends analysis, such as sales analysis, market research analysis, customer analysis, defect analysis, cost analysis, and forecasting.

Conclusion

Information perception is the way toward addressing information in a graphical or pictorial manner in an unmistakable and compelling way. It has arisen as an incredible and generally relevant device for breaking down and deciphering huge and complex information. It has become a speedy, simple method for passing on ideas in a general configuration. It should discuss complex thoughts with clearness, precision, and productivity. These advantages have permitted information perception to be valuable in numerous fields of study.

REFERENCES

https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/pie-chart/
https://mode.com/blog/r-data-visualization-packages/
https://www.javatpoint.com/r-line-graphs
https://www.analyticssteps.com/blogs/10-applications-data-visualization
https://analyticsindiamag.com/top-10-r-packages-for-data-visualisation/