**data visualization in R**

**data visualization** in R. Visualizations are a key component of data analysis as they help in interpreting trends, patterns, and insights within data.

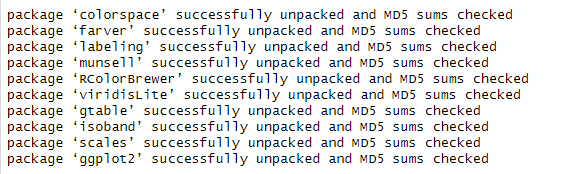
**Data Visualization in R with ggplot2**

The **ggplot2** package, part of the tidyverse, is the most popular tool for creating elegant and versatile data visualizations in R. It uses a **grammar of graphics** approach, making it easy to layer different components in a plot.

To get started with **ggplot2**, you’ll need to install and load it:

install.packages("ggplot2")

after running this code you’ll be getting this



library(ggplot2)

**The Basic Structure of ggplot2**

ggplot2 plots are built up in layers:

1. **Data**: The data frame you’re visualizing.
2. **Aesthetics (aes)**: Mappings of variables to visual properties like x, y, color, size, etc.
3. **Geometries (geom)**: The type of plot (e.g., points, lines, bars).

Example:

ggplot(data = df, aes(x = Age, y = Score)) +

geom\_point()

**Common Types of Plots**

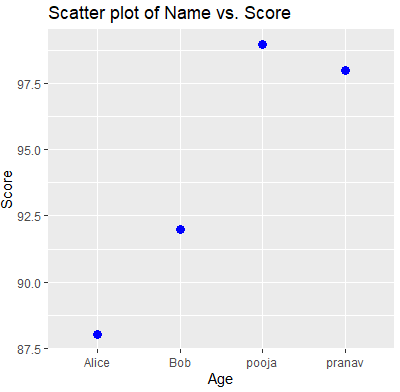
**1. Scatter Plot**

* Useful for visualizing the relationship between two continuous variables.

ggplot(df, aes(x = Name, y = Score)) +

geom\_point(color = "blue", size = 3) +

labs(title = "Scatter Plot of Name vs. Score", x = "Age", y = "Score")



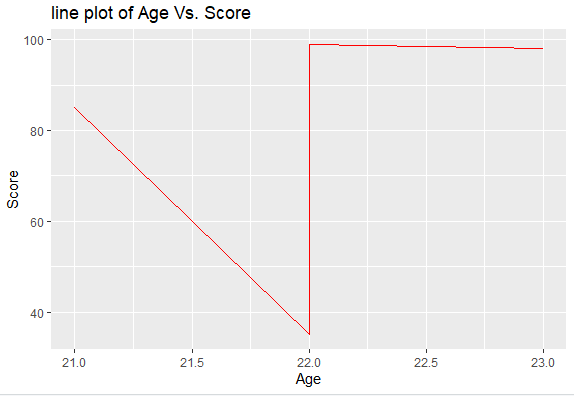
**2. Line Plot**

* Great for time-series data or showing trends over a continuous range.

ggplot(df, aes(x = Age, y = Score)) +

geom\_line(color = "red") +

labs(title = "Line Plot of Age vs. Score", x = "Age", y = "Score")



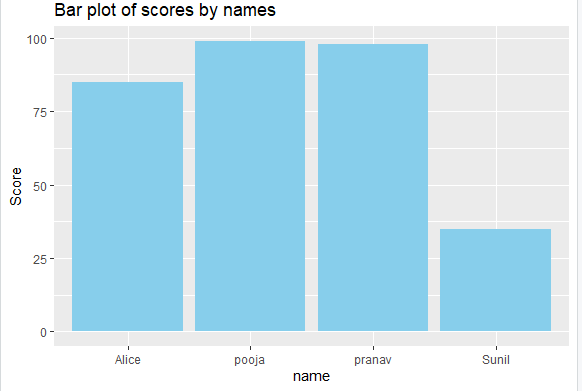
**3. Bar Plot**

* Ideal for showing counts or comparing values across categories.

ggplot(df, aes(x = Name, y = Score)) +

geom\_bar(stat = "identity", fill = "skyblue") +

labs(title = "Bar Plot of Scores by Name", x = "Name", y = "Score")



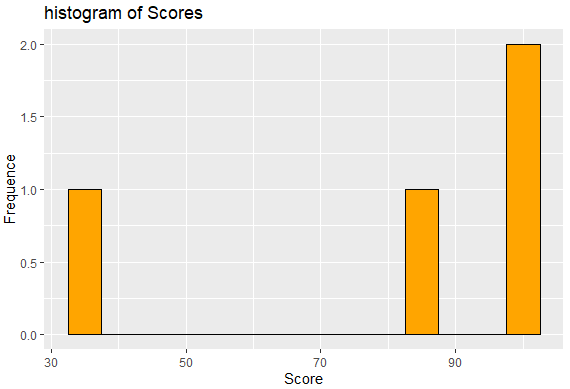
**4. Histogram**

* Useful for viewing the distribution of a single continuous variable.

ggplot(df, aes(x = Score)) +

geom\_histogram(binwidth = 5, fill = "orange", color = "black") +

labs(title = "Histogram of Scores", x = "Score", y = "Frequency")



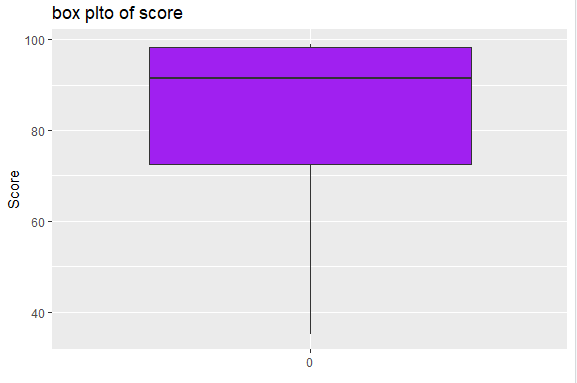
**5. Box Plot**

* Shows the distribution of a variable, with quartiles and outliers.

ggplot(df, aes(x = factor(0), y = Score)) +

geom\_boxplot(fill = "purple") +

labs(title = "Box Plot of Scores", x = "", y = "Score")



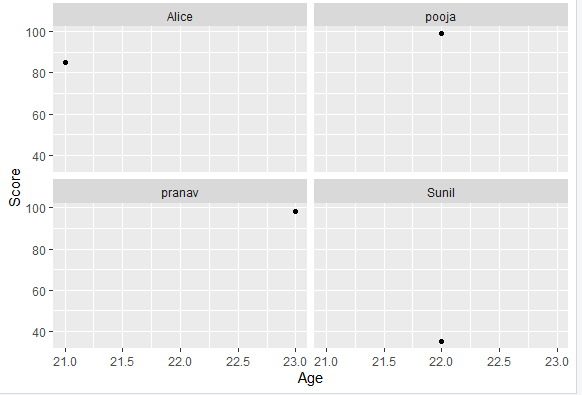
**Adding Layers to Your Plot**

1. **Faceting**: Create multiple panels for each category.

ggplot(df, aes(x = Age, y = Score)) +

geom\_point() +

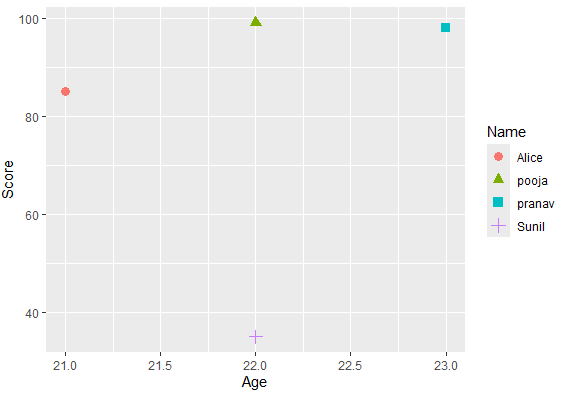
facet\_wrap(~ Name)



1. **Colors and Shapes**: Customize colors or shapes based on another variable.

ggplot(df, aes(x = Age, y = Score, color = Name, shape = Name)) +

geom\_point(size = 3)



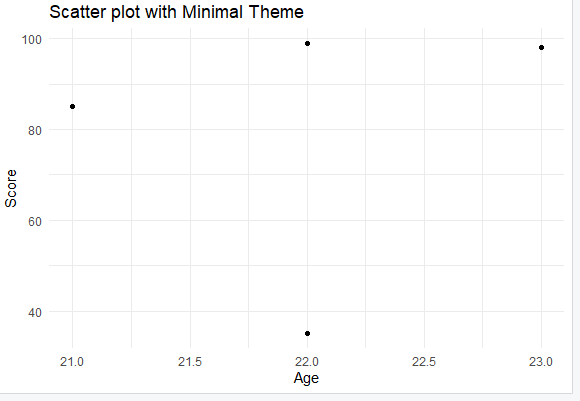
1. **Themes**: Modify the appearance of your plot.

ggplot(df, aes(x = Age, y = Score)) +

geom\_point() +

theme\_minimal() +

labs(title = "Scatter Plot with Minimal Theme")



1. **Titles, Labels, and Legends**: Add custom titles, axis labels, and adjust legends.

r

Copy code

ggplot(df, aes(x = Age, y = Score)) +

geom\_point(color = "darkgreen") +

labs(

title = "Scatter Plot of Age vs. Score",

subtitle = "A closer look at the relationship between Age and Score",

x = "Age",

y = "Score"

)

**Exercises**

1. **Create a Scatter Plot**
   * Plot the relationship between Age and Score with customized colors and point sizes.
2. **Faceted Scatter Plot**
   * Use facet\_wrap() to create a scatter plot of Age vs. Score for each Name.
3. **Box Plot**
   * Create a box plot showing the distribution of Score and customize it with colors.
4. **Histogram**
   * Visualize the distribution of Score using a histogram, and experiment with binwidth and color options.

**Summary**

ggplot2 offers a flexible and powerful way to create informative and visually appealing graphics in R. Mastering ggplot2 will allow you to convey insights more effectively through data visualization. Once you try out these examples, let me know if you'd like more advanced plotting techniques or if you're ready to move on to another topic!

# Getting started with Data Visualization in R

Last Updated : 05 Jul, 2024

**Data visualization** is the technique used to deliver insights in data using visual cues such as graphs, charts, maps, and many others. This is useful as it helps in intuitive and easy understanding of the large quantities of data and thereby make better decisions regarding it.

## Data Visualization in R Programming Language

The popular data visualization tools that are available are Tableau, Plotly, R, Google Charts, Infogram, and Kibana. The various data visualization platforms have different capabilities, functionality, and use cases. They also require a different skill set. This article discusses the use of R for data visualization.

R is a language that is designed for statistical computing, graphical data analysis, and scientific research. It is usually preferred for data visualization as it offers flexibility and minimum required coding through its packages.

### 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 | 8.0 | 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:**

R

*# Horizontal Bar Plot for*

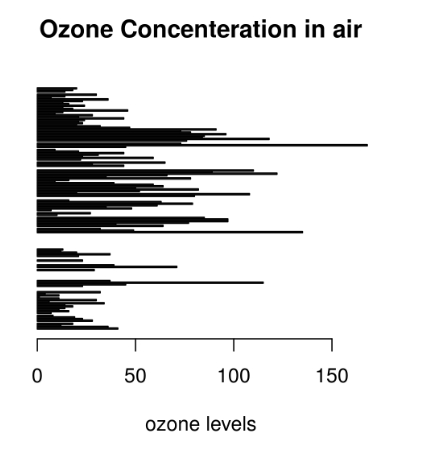
*# Ozone concentration in air*

barplot(airquality$Ozone,

main = 'Ozone Concenteration in air',

xlab = 'ozone levels', horiz = **TRUE**)

**Output:**



**Example 2:**

R

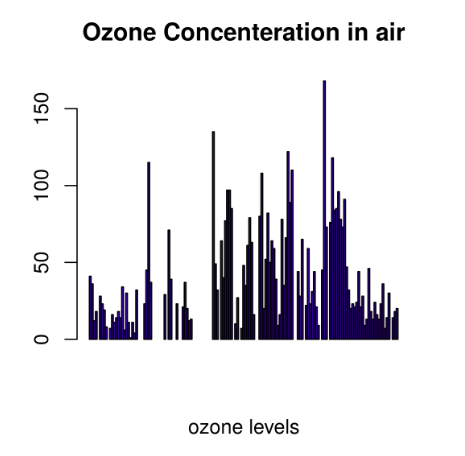
*# Vertical Bar Plot for*

*# 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 a 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:**

R

*# Histogram for Maximum Daily Temperature*

data(airquality)

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

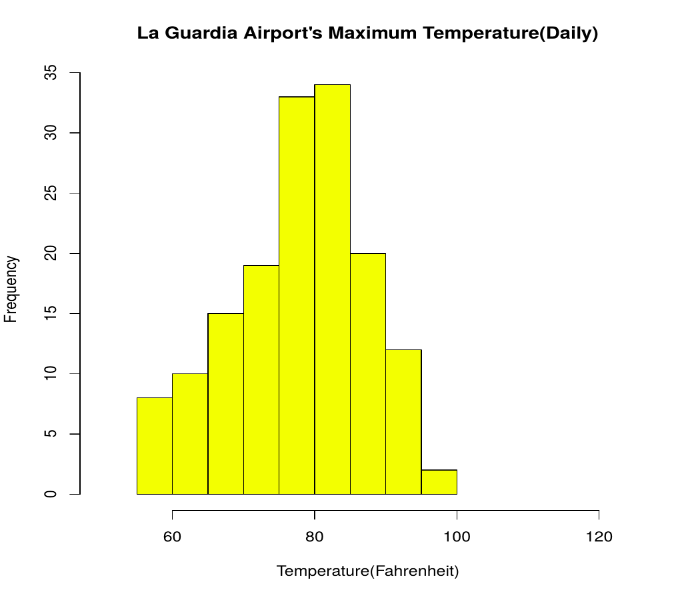
Maximum Temperature(Daily)",

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:**

R

*# Box plot for average wind speed*

data(airquality)

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

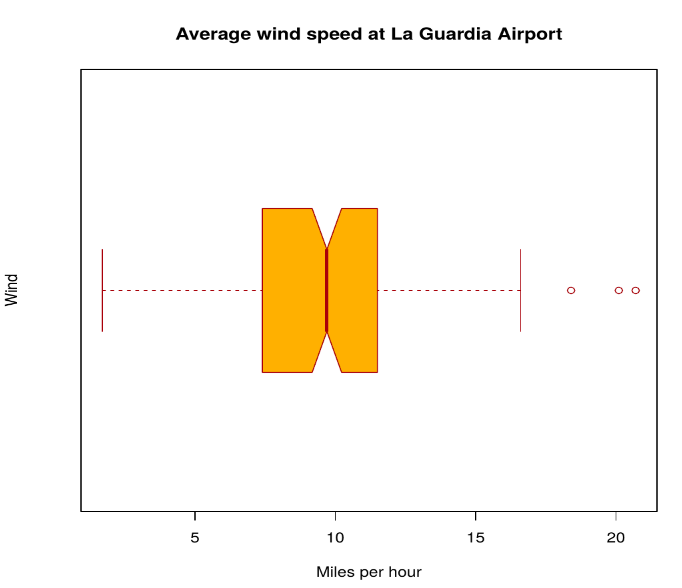
at La Guardia Airport",

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:**

R

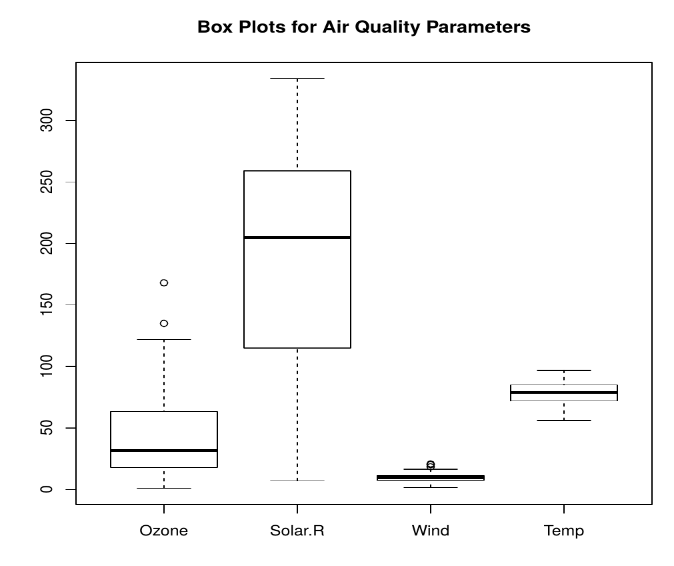
*# Multiple Box plots, each representing*

*# 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:**

R

*# Scatter plot for Ozone Concentration per month*

data(airquality)

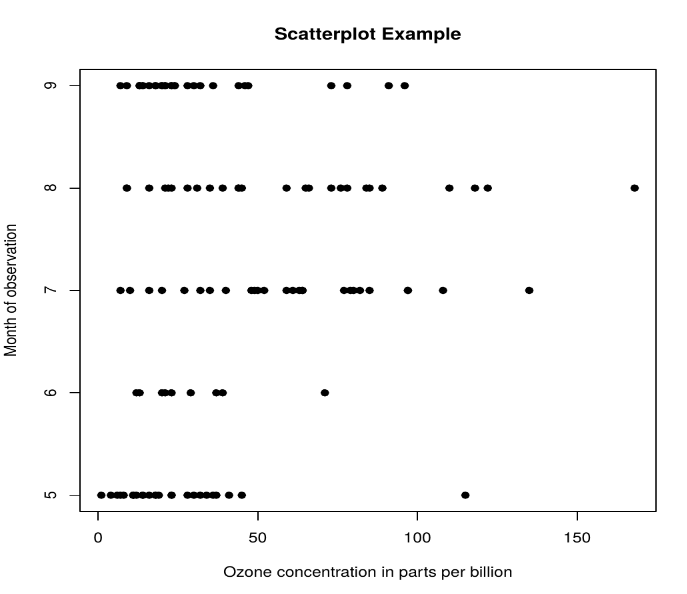
plot(airquality$Ozone, airquality$Month,

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.*

R

*# Set seed for reproducibility*

*# set.seed(110)*

*# Create example data*

data <- matrix(rnorm(50, 0, 5), nrow = 5, ncol = 5)

*# Column names*

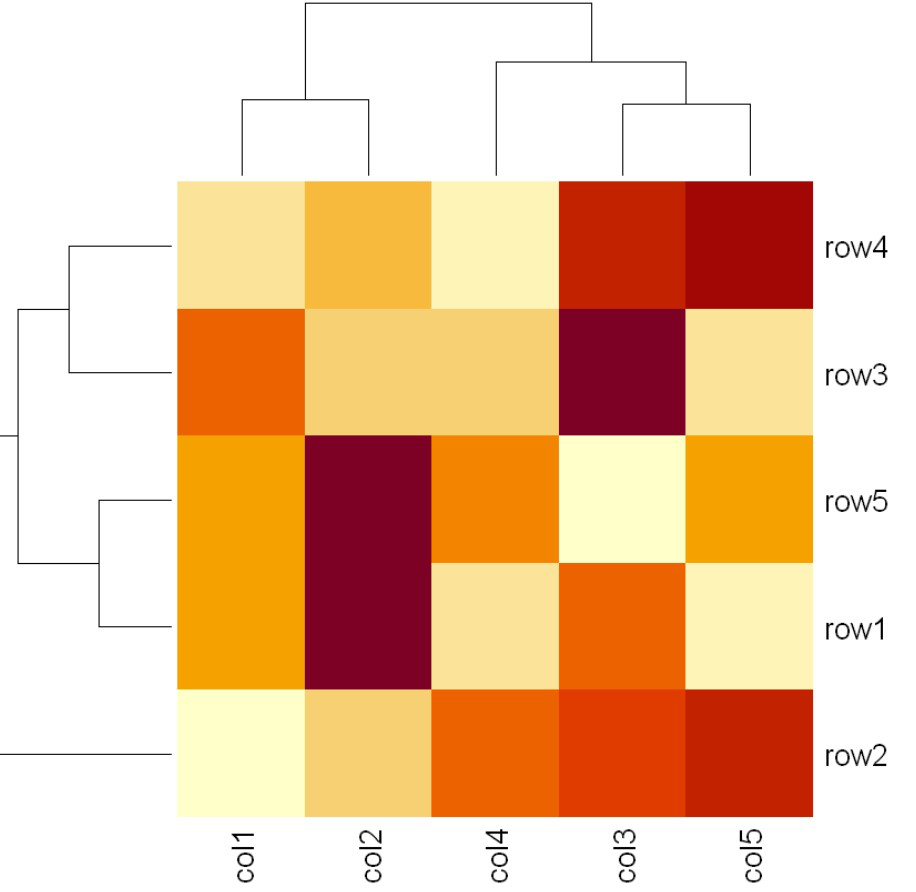
colnames(data) <- paste0("col", 1:5)

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")

R

*# Read dataset and convert it into*

*# Dataframe*

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

df <- data.frame(data)

*# Load the required libraries*

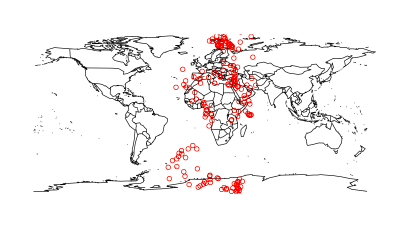
library(maps)

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).*

R

*# 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,

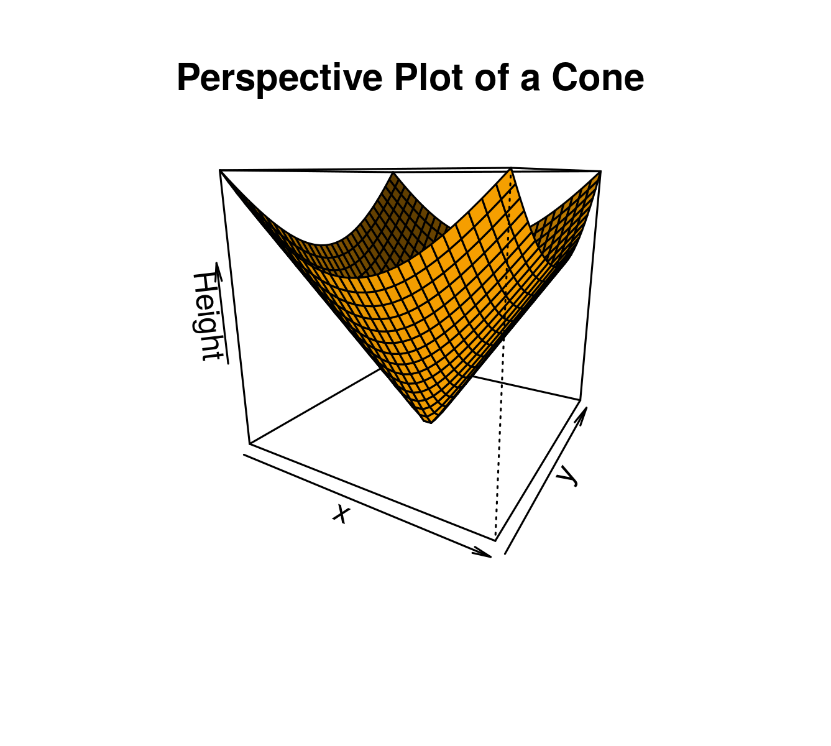
main="Perspective Plot of a Cone",

zlab = "Height",

theta = 30, phi = 15,

col = "orange", shade = 0.4)

**Output:**



## **Advantages of Data Visualization in R:**

R has the following advantages over other tools for data visualization:

* R offers a broad collection of visualization libraries along with extensive online guidance on their usage.
* R also offers data visualization in the form of 3D models and multipanel charts.
* Through R, we can easily customize our data visualization by changing axes, fonts, legends, annotations, and labels.

## **Disadvantages of Data Visualization in R:**

R also has the following disadvantages:

* R is only preferred for data visualization when done on an individual standalone server.
* 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.