**Data Preparation and Manipulation in R**

>data("cars")

>str(cars)

>data("mtcars")

>str(mtcars)

>mtcars$vs<-as.factor(mtcars$vs)

>mtcars$am<-as.factor(mtcars$am)

>library(dplyr)

>summary(mtcars)

>library(dplyr)

#Filter

>car<-mtcars%>%filter(vs=="0")

#arrange

>mtcars%>%filter(vs=="0")%>%arrange(desc(hp))

#summarise

>mtcars%>%filter(vs=="0")%>%group\_by(am)%>%summarise(Average=mean(hp))

>mtcars%>%filter(vs=="0")%>%group\_by(am)%>%summarise(Average=mean(hp),SD=sd(hp),Max=max(hp),Min=min(hp),Count=n())

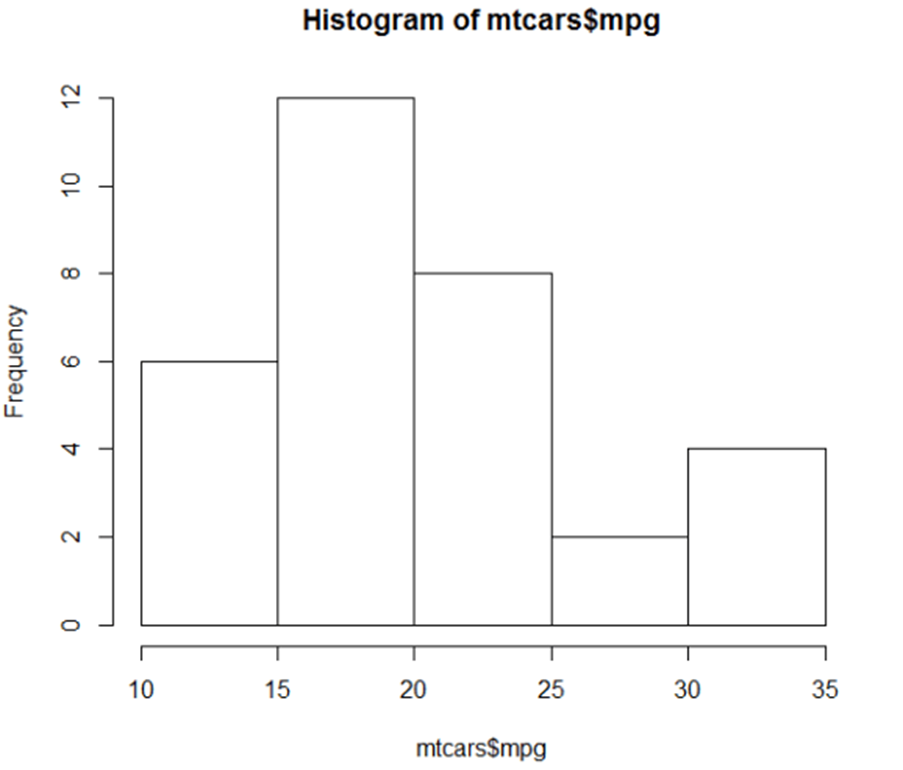
>mtcars%>%filter(vs=="0")%>%group\_by(am)%>%summarise(Average=mean(hp),SD=sd(hp),Max=max(hp),Min=min(hp),Count=n())%>%arrange(desc(Average))

Visualization

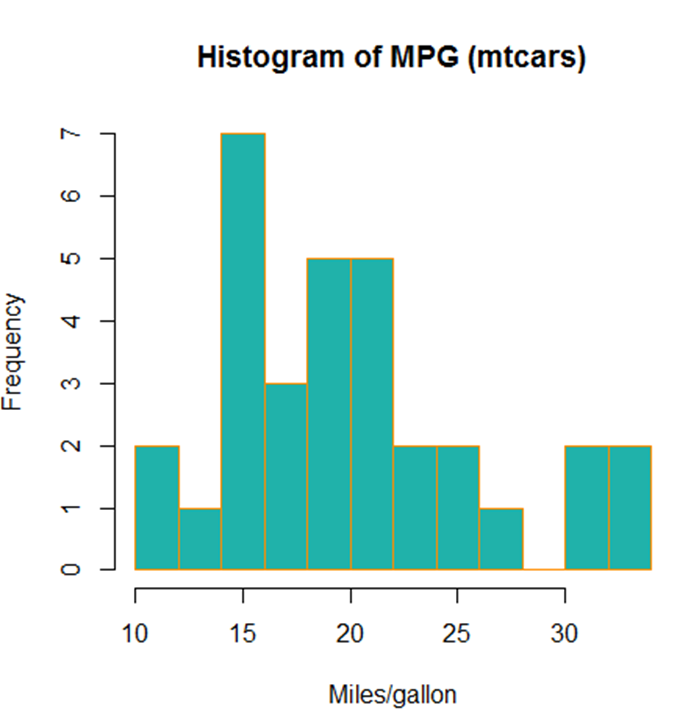
Histograms:

When visualizing a single numerical variable, a histogram will be our go-to tool, which can be created in R using the hist() function. There are various variables in dataset “mtcars”, we will be using variable mpg(miles per gallon) for this visualization:

hist(mtcars$mpg)



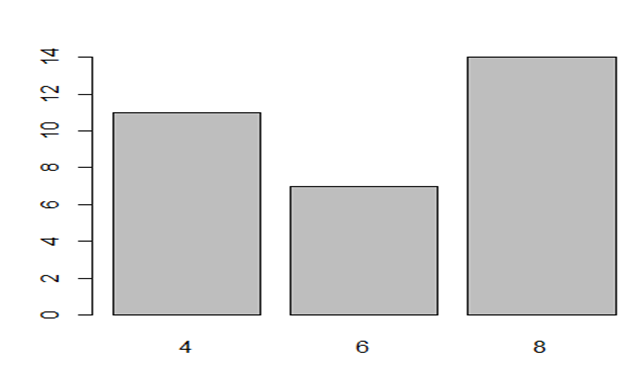
hist(mtcars$mpg, xlab = "Miles/gallon", main = "Histogram of MPG (mtcars)", breaks = 12, col = "lightseagreen", border = "darkorange")



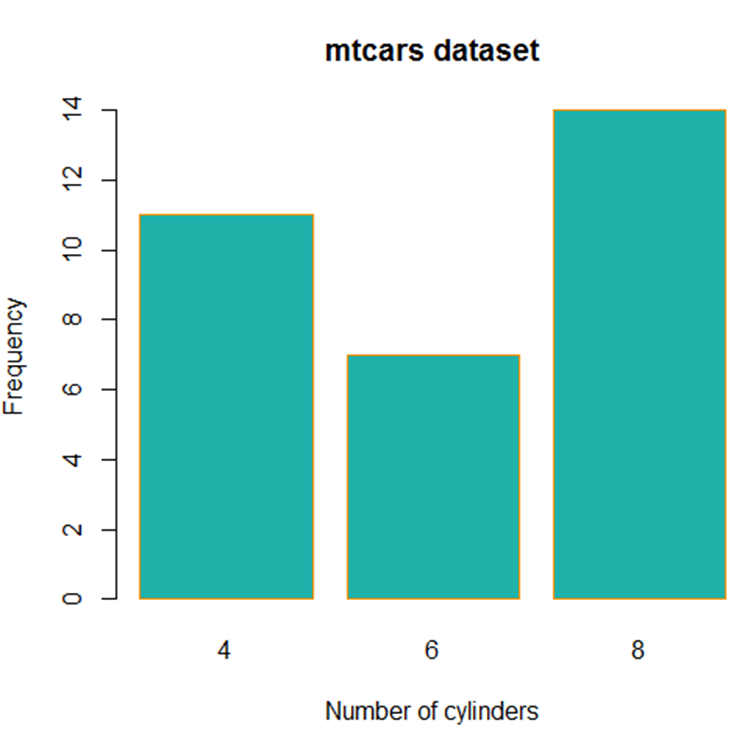
Barplots:

A barplot can provide a visual summary of a categorical variable, or a numeric variable with a finite number of values, like a ranking from 1 to 10. For drawing barplot I will use cyl variable which is nothing but Number of cylinders in mtcars dataset.

barplot(table(mtcars$cyl))



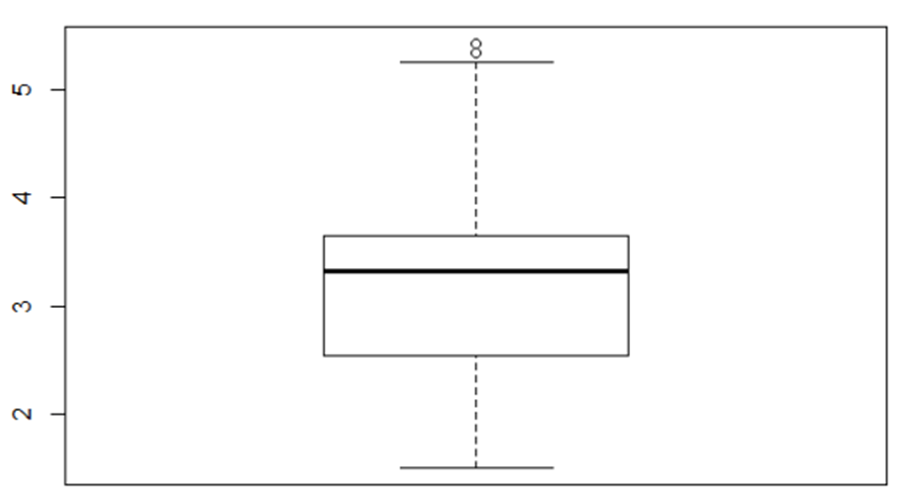
barplot(table(mtcars$cyl), xlab = "Number of cylinders", ylab = "Frequency", main = "mtcars dataset", col = "lightseagreen", border = "darkorange")

****

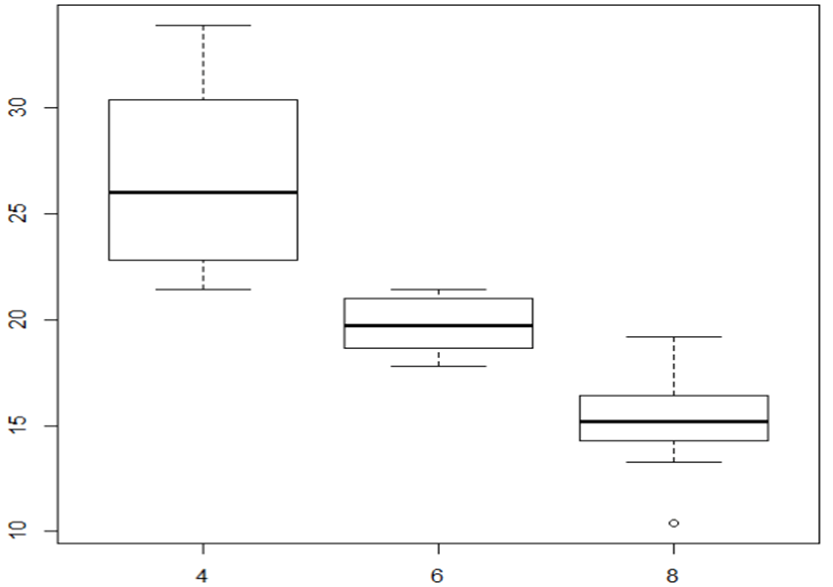
**Boxplots:**

we can use a single boxplot as an alternative to a histogram for visualizing a single numerical variable. Let’s do a boxplot for Weight column in mtcars.

boxplot(mtcars$wt)



boxplot(mpg ~ cyl , data = mtcars)



boxplot(mpg ~ cyl, data = mtcars,

xlab = "Number of cylinders",

ylab = "Miles/(US) gallon",

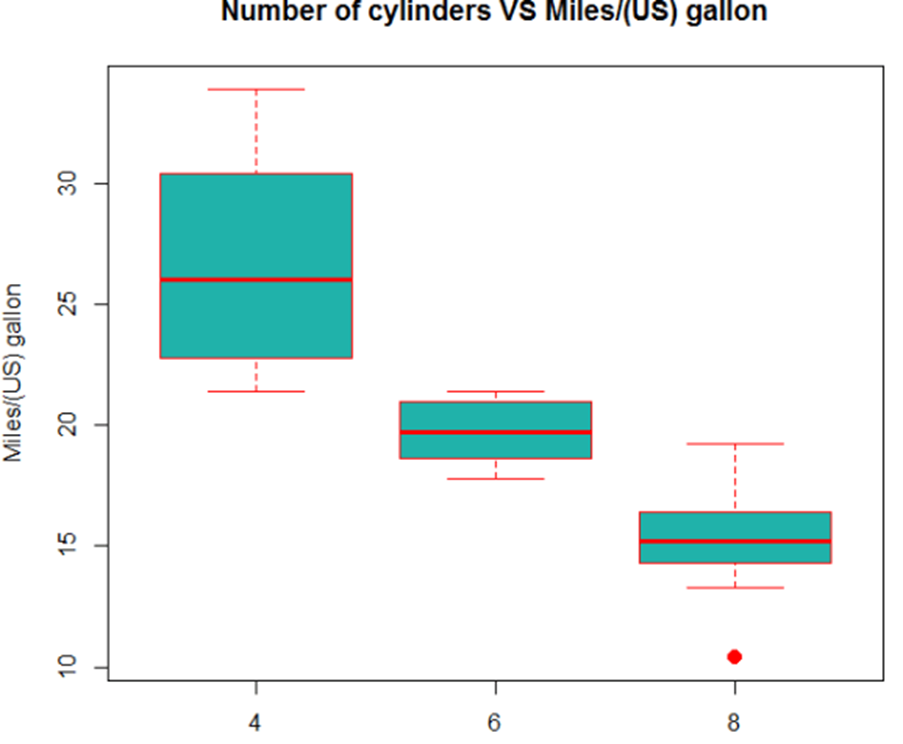
main = "Number of cylinders VS Miles/(US) gallon",

pch = 20,

cex = 2,

col = "lightseagreen",

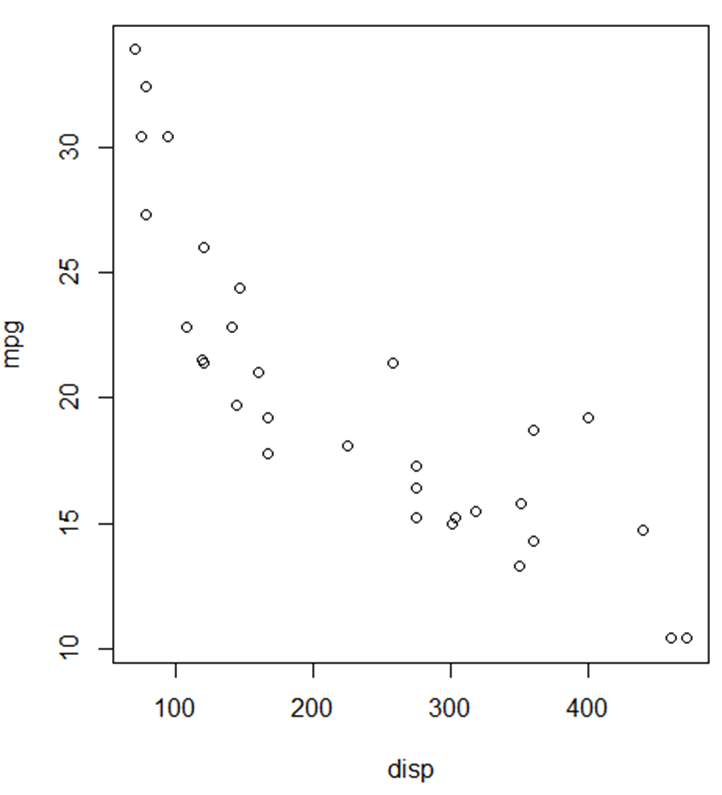
border = "red")



Scatterplots:

To visualize the relationship between two numeric variables we will use a scatterplot. This can be done with the plot() function and the ~ syntax we just used with a boxplot.

plot(mpg~disp, data=mtcars)



plot(mpg ~ disp, data = mtcars,

xlab = "Displacement",

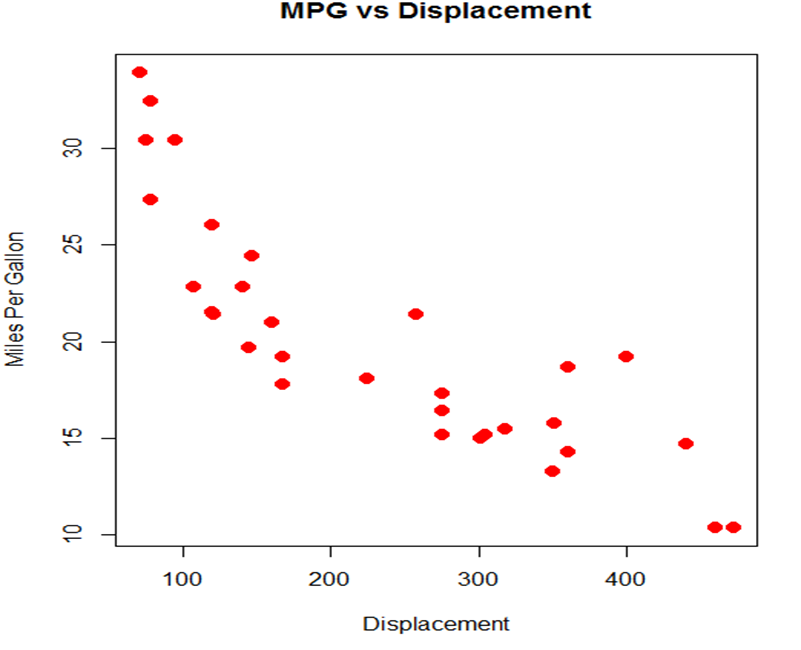
ylab = "Miles Per Gallon",

main = "MPG vs Displacement",

pch = 20,

cex = 2,

col = "red")



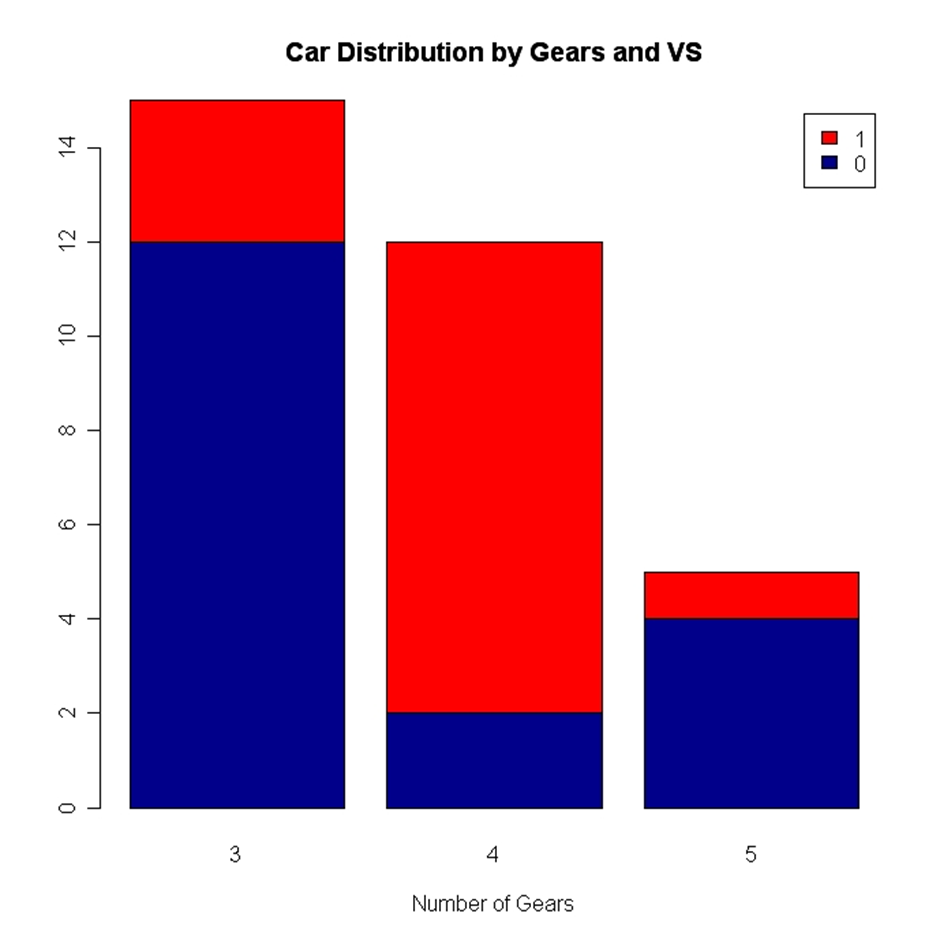
Stacked Bar Plot: subdivided bar diagram

counts <- table(mtcars$vs, mtcars$gear)

barplot(counts, main="Car Distribution by Gears and VS",

xlab="Number of Gears", col=c("darkblue","red"),

legend = rownames(counts))



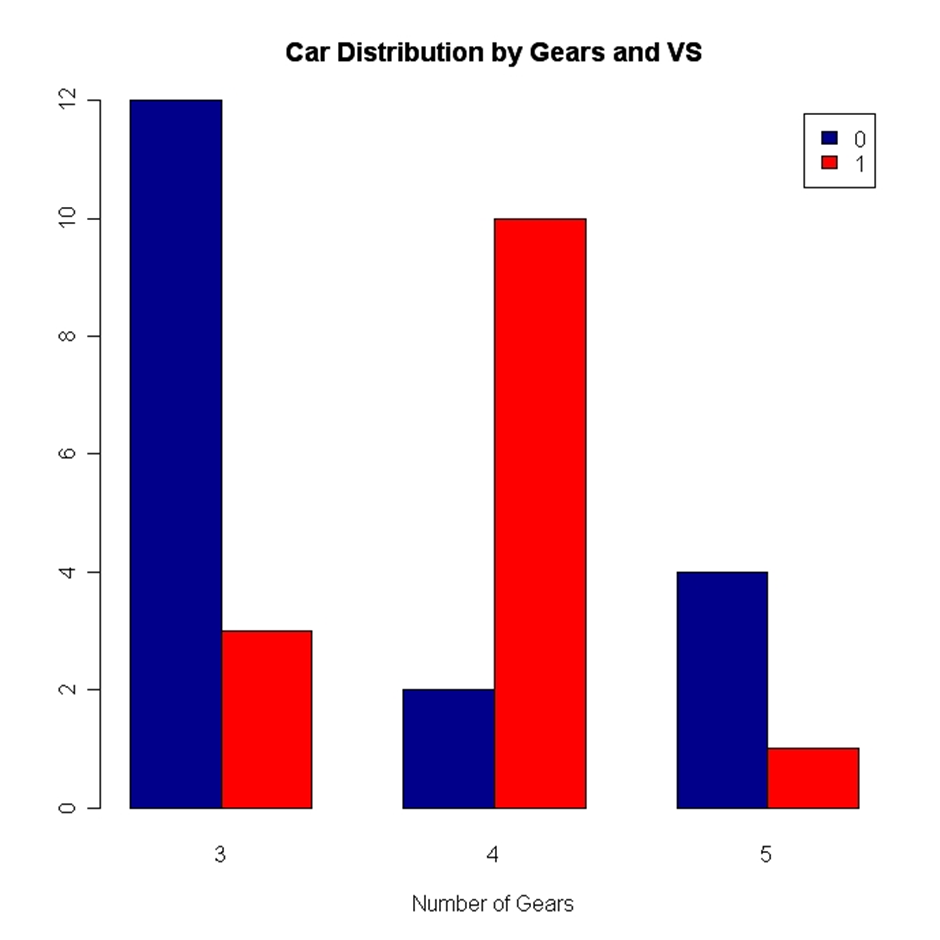
Grouped Bar Plot- Multiple bar diagram

counts <- table(mtcars$vs, mtcars$gear)

barplot(counts, main="Car Distribution by Gears and VS",

xlab="Number of Gears", col=c("darkblue","red"),

legend = rownames(counts), beside=TRUE)



Pie Chart

Basic pie chart to show the proportion of cars from the mtcars data set that has different carburetor values.

#Calculate the frequency of different carb values using table function

mtcarscarb = table(mtcars$carb)

#Create percent label values

percentlabels<- round(100\*mtcarscarb/sum(mtcarscarb), 1)

#Create labels for each pie in the chart

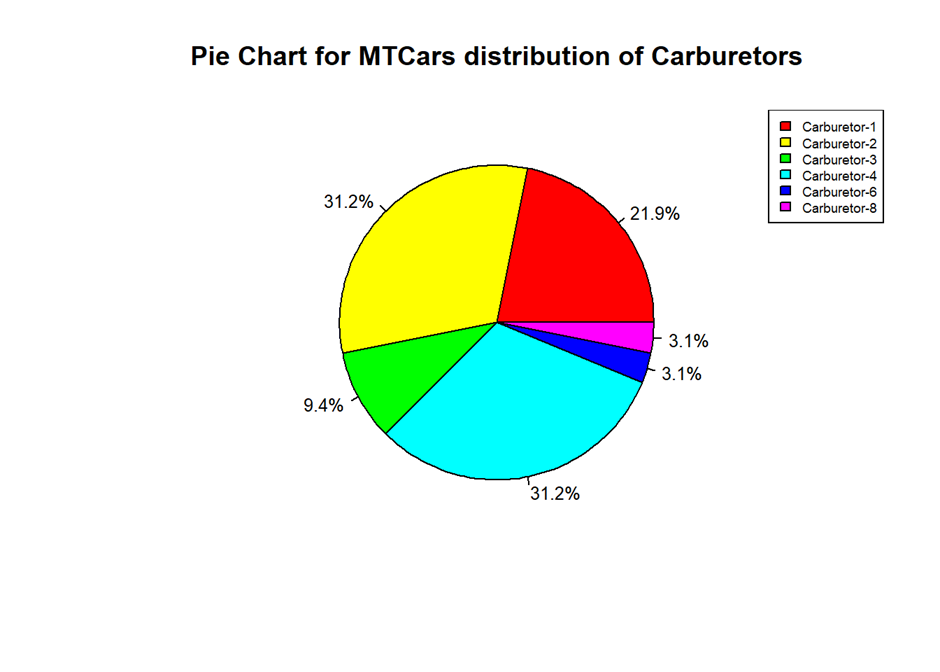
pielabels<- paste(percentlabels, "%", sep="")

#R code to create the Pie Chart

pie(mtcarscarb,col = rainbow(length(mtcarscarb)), labels = pielabels , main = 'Pie Chart for MTCars distribution of Carburetors', cex = 0.8)

#Legend for the pie chart

legend("topright", c("Carburetor-1","Carburetor-2","Carburetor-3","Carburetor-4","Carburetor-6","Carburetor-8"), cex=0.6, fill= rainbow(length(mtcarscarb)))



Line Graph

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

The basic syntax to create a line chart in R is −

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

Following is the description of the parameters used −

• 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.

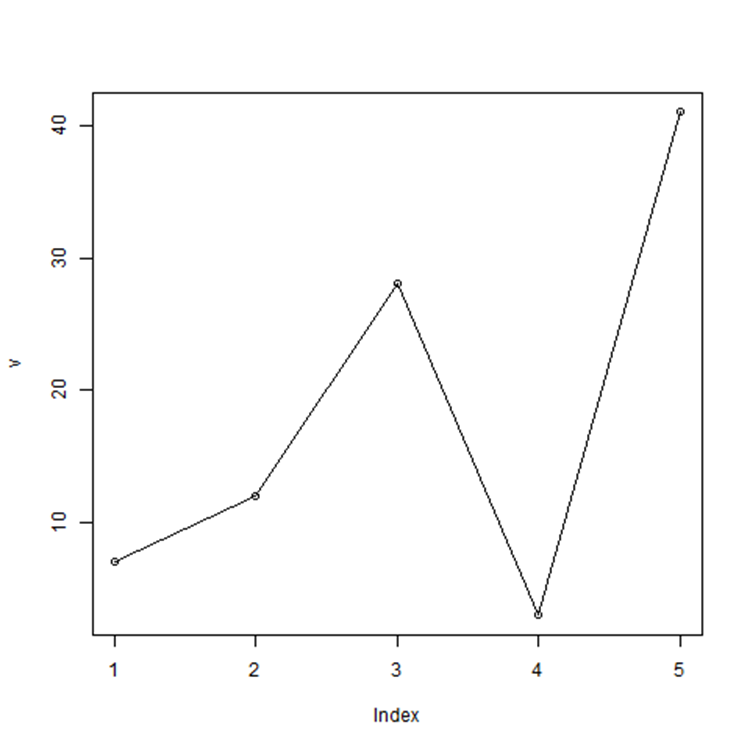
• ylab is the label for y axis.

• main is the Title of the chart.

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

v <- c(7,12,28,3,41)

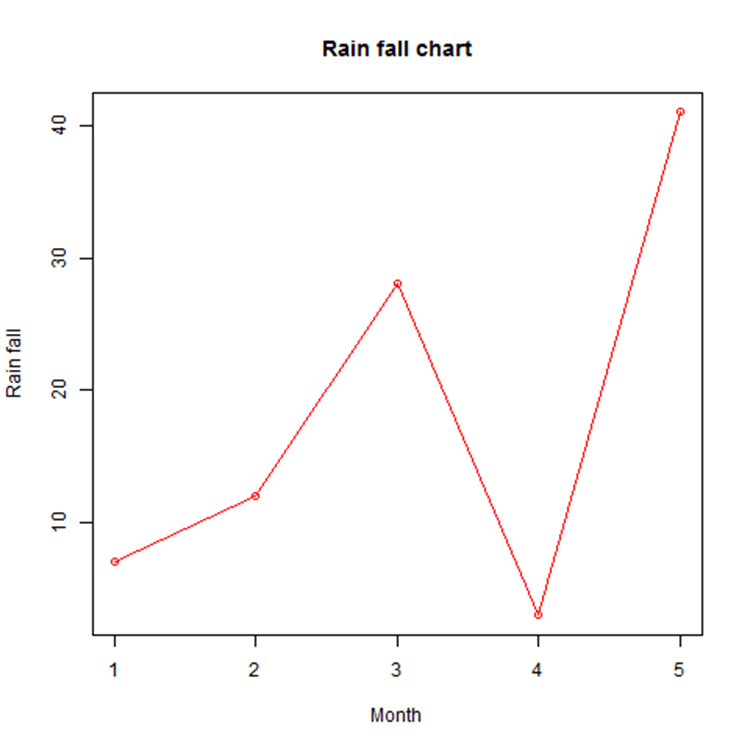
plot(v,type = "o")



v <- c(7,12,28,3,41)

plot(v,type = "o", col = "red", xlab = "Month", ylab = "Rain fall",

main = "Rain fall chart")



Multiple Lines in a Line Chart

v <- c(7,12,28,3,41)

t <- c(14,7,6,19,3)

plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall",

main = "Rain fall chart")

lines(t, type = "o", col = "blue")

