**Data Visualization in R**

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns.

For this section we will work with **mtcars** data set which is a built-in data set in R. You can get the help file by just typing ?**mtcars**. To load the datasets just use the following code.

?mtacars  
library(datasets)

#OR

data("mtcars")

#So, this is how the data looks:

## mpg cyl disp hp drat wt qsec vs am gear carb

## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4

## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4

## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1

## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1

## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2

## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1

## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4

## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2

## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2

## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4

## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4

## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3

## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3

## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3

## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4

## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4

## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4

## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1

## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2

## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1

## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1

## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2

## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2

## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4

## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2

## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1

## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2

## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2

## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4

## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6

## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8

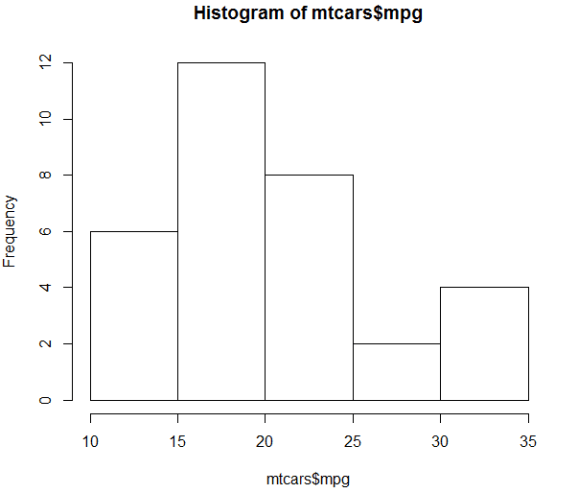
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

The data was extracted from the 1974 Motor Trend US magzine, and comprises fuel consumption and 10 aspects of automobile design and performance of 32 automobiles for the models built between 1973-74.

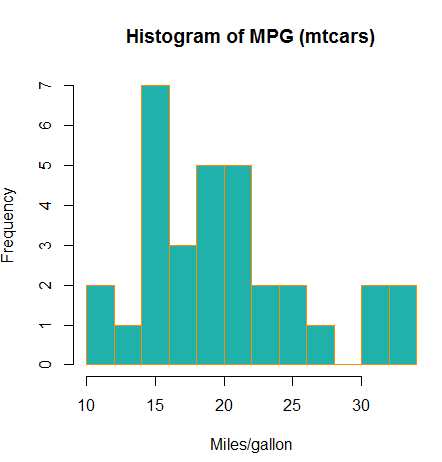
##### **Histograms:**

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)



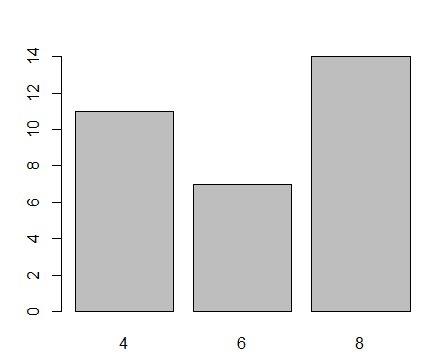
hist(mtcars$mpg, xlab = "Miles/gallon", main = "Histogram of MPG (mtcars)", breaks = 12, col = "lightseagreen", border = "darkorange")



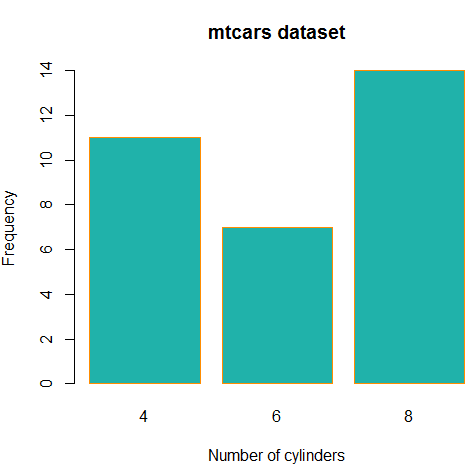
##### **Barplots:**

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))



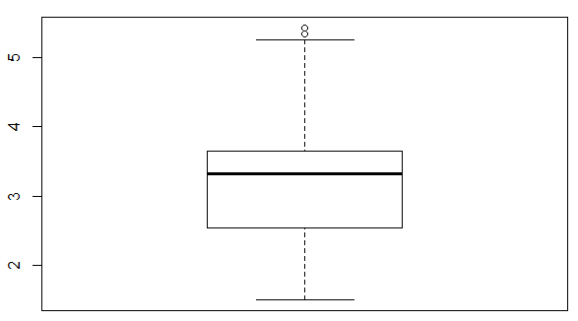
barplot(table(mtcars$cyl), xlab = "Number of cylinders", ylab = "Frequency", main = "mtcars dataset", col = "lightseagreen", border = "darkorange")



##### **Boxplots:**

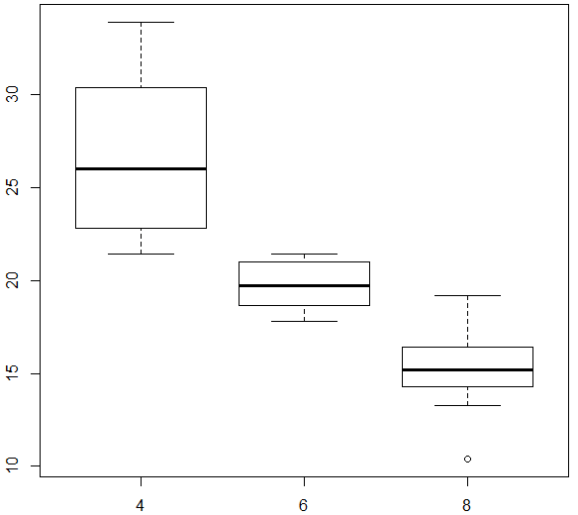
we can use a single boxplot as an alternative to a histogram for visualizing a single numerical variable. Let’s do a boxplot for Weight column in mtcars.

boxplot(mtcars$wt)



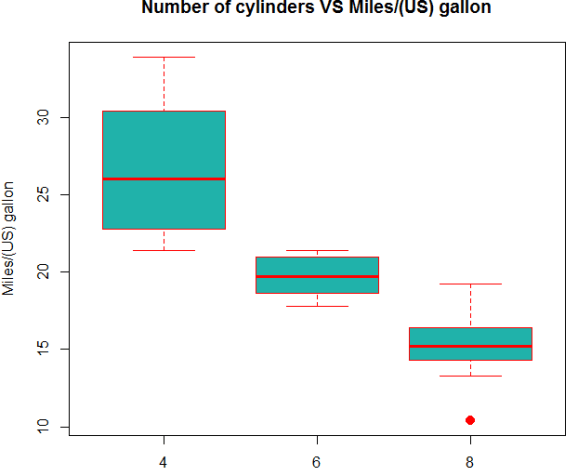
To visualize the relationship between a numerical and categorical variable, we can use a boxplot. Here mpg is a numerical variable and Number of cylinders is categorical.

boxplot(mpg ~ cyl , data = mtcars)



We can make the box plot more attractive by setting some of its parameters.

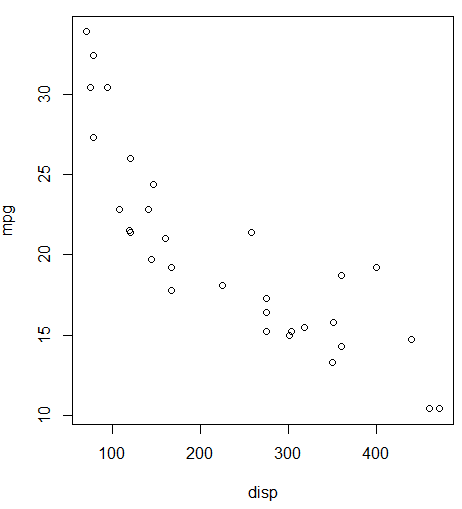
boxplot(mpg ~ cyl, data = mtcars,  
xlab = "Number of cylinders",  
ylab = "Miles/(US) gallon",  
main = "Number of cylinders VS Miles/(US) gallon",  
pch = 20,  
cex = 2,  
col = "lightseagreen",  
border = "red")



##### **Scatterplots:**

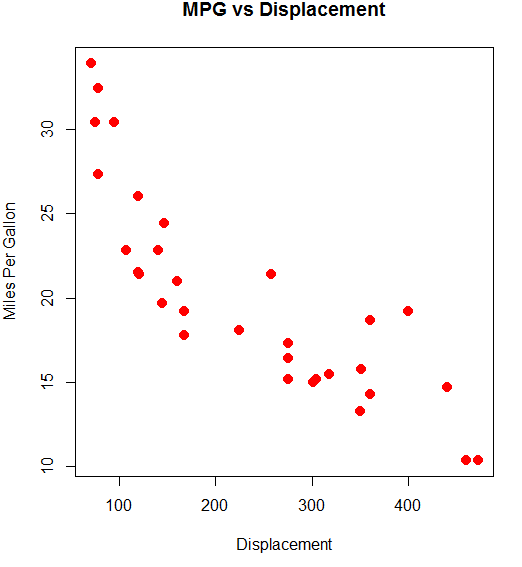
To visualize the relationship between two numeric variables we will use a scatterplot. This can be done with the plot() function and the ~ syntax we just used with a boxplot.

plot(mpg~disp, data=mtcars)



You can use the above graph more beautiful by using the below code.

plot(mpg ~ disp, data = mtcars,  
xlab = "Displacement",  
ylab = "Miles Per Gallon",  
main = "MPG vs Displacement",  
pch = 20,  
cex = 2,  
col = "red")



##### **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.

### Example

A simple line chart is created using the input vector and the type parameter as "O". The below script will create and save a line chart in the current R working directory.

# Create the data for the chart.

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

# Give the chart file a name.

png(file = "line\_chart.jpg")

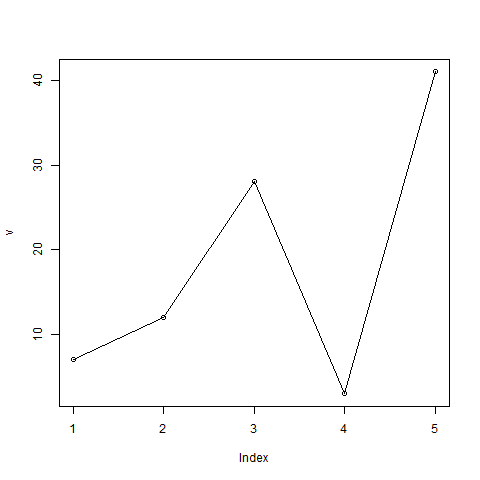
# Plot the bar chart.

plot(v,type = "o")

# Save the file.

dev.off()

When we execute the above code, it produces the following result −



## Line Chart Title, Color and Labels

The features of the line chart can be expanded by using additional parameters. We add color to the points and lines, give a title to the chart and add labels to the axes.

### Example

# Create the data for the chart.

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

# Give the chart file a name.

png(file = "line\_chart\_label\_colored.jpg")

# Plot the bar chart.

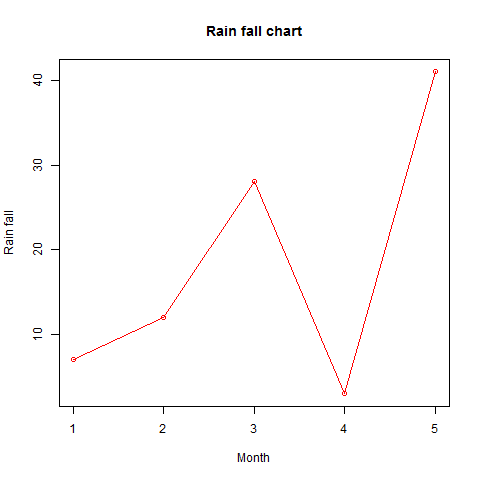
plot(v,type = "o", col = "red", xlab = "Month", ylab = "Rain fall",

main = "Rain fall chart")

# Save the file.

dev.off()

When we execute the above code, it produces the following result −



## Multiple Lines in a Line Chart

More than one line can be drawn on the same chart by using the **lines()**function.

After the first line is plotted, the lines() function can use an additional vector as input to draw the second line in the chart,

# Create the data for the chart.

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

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

# Give the chart file a name.

png(file = "line\_chart\_2\_lines.jpg")

# Plot the bar chart.

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

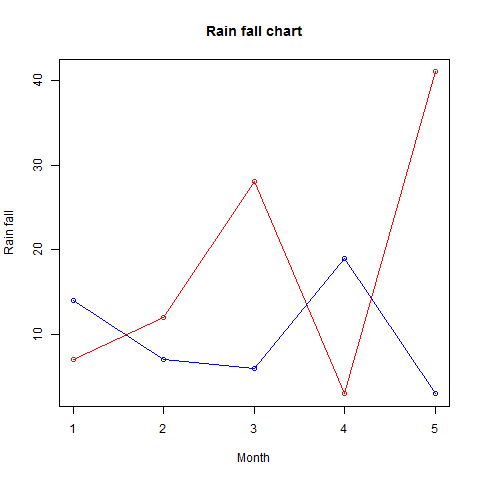
main = "Rain fall chart")

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

# Save the file.

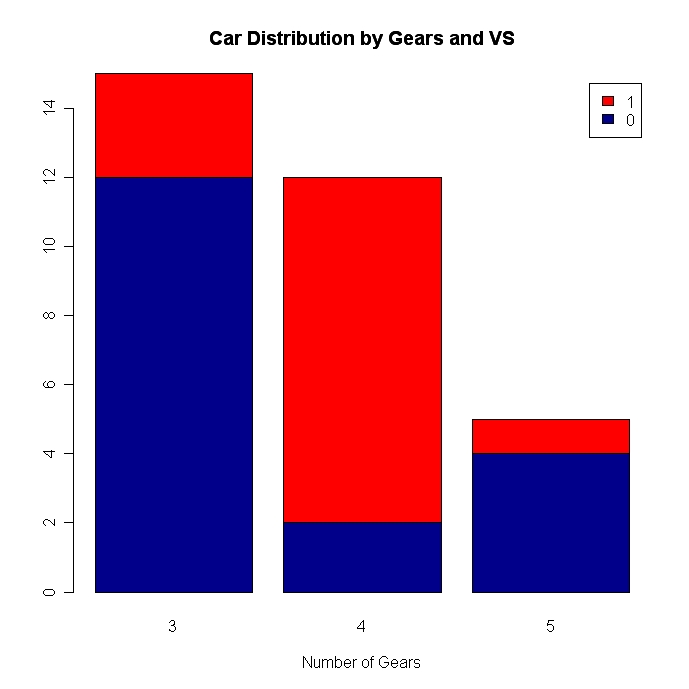
dev.off()

When we execute the above code, it produces the following result −



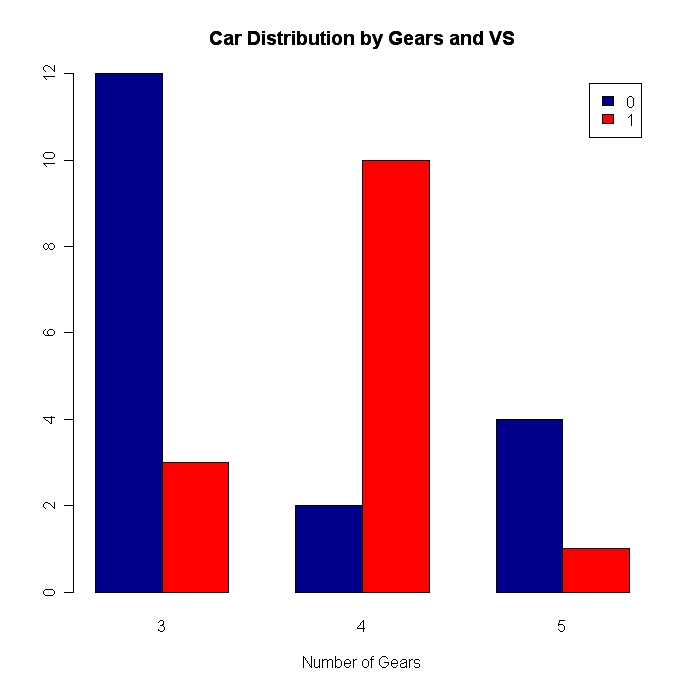
## Stacked Bar Plot [[8]](https://www.statmethods.net/graphs/bar.html)

# Stacked Bar Plot with Colors and Legend  
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

# Grouped Bar Plot  
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)



## Notes:

Bar plots need not be based on counts or frequencies. You can create bar plots that represent means, medians, standard deviations, etc. Use the [aggregate( )](https://www.statmethods.net/management/aggregate.html)function and pass the results to the barplot( ) function.

By default, the categorical axis line is suppressed. Include the option **axis.lty=1** to draw it.

With many bars, bar labels may start to overlap. You can decrease the font size using the **cex.names =** option. Values smaller than one will shrink the size of the label.

**Pie Chart**

Basic pie chart to show the proportion of cars from the mtcars data set that have 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

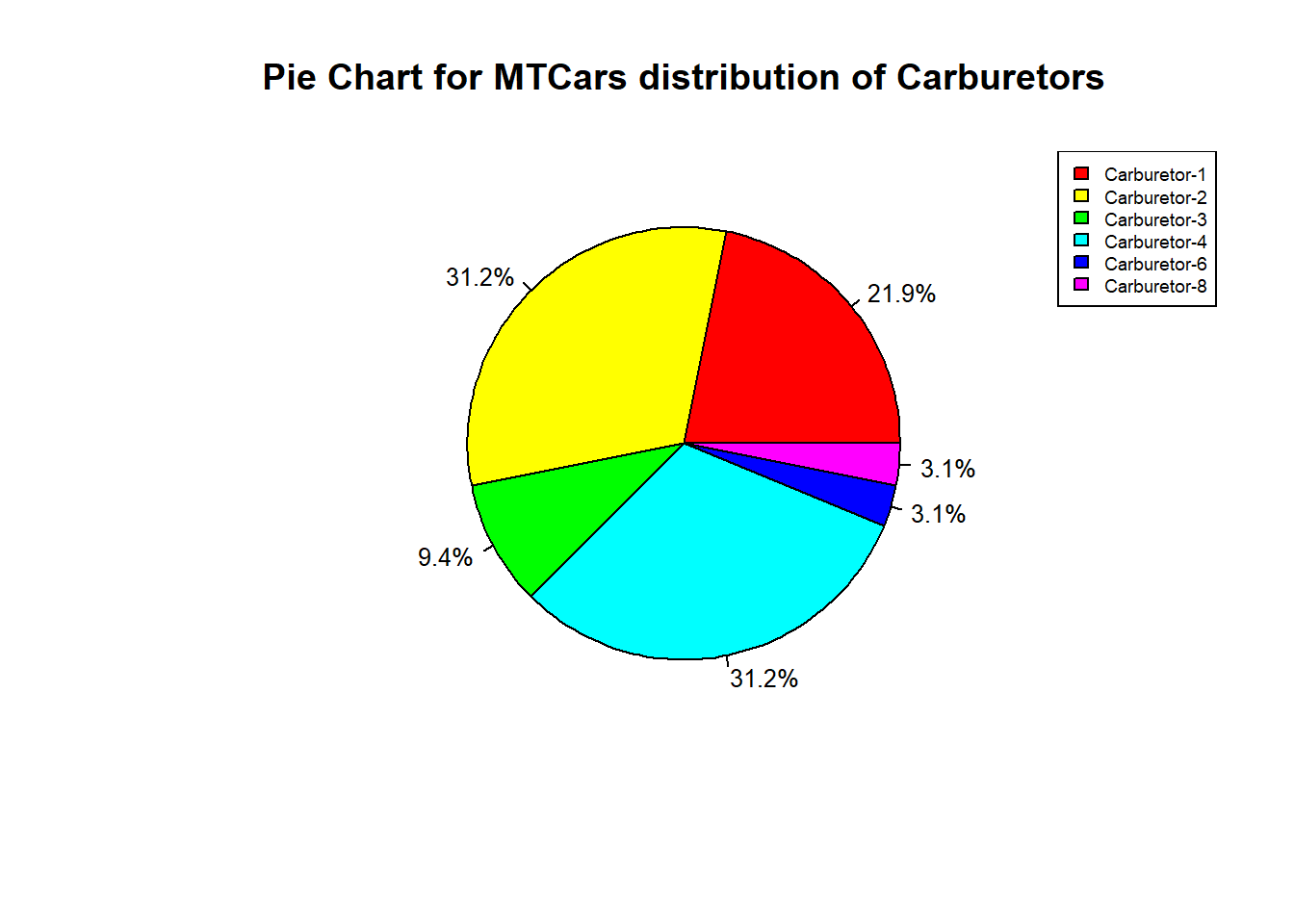
pielabels<- paste(percentlabels, "%", sep="")

#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)))

****