R - Graphics

R - Line Graphs: A line chart is a graph that connects a series of points by drawing line segments between them.

The **plot()** function in R is used to create the line graph.

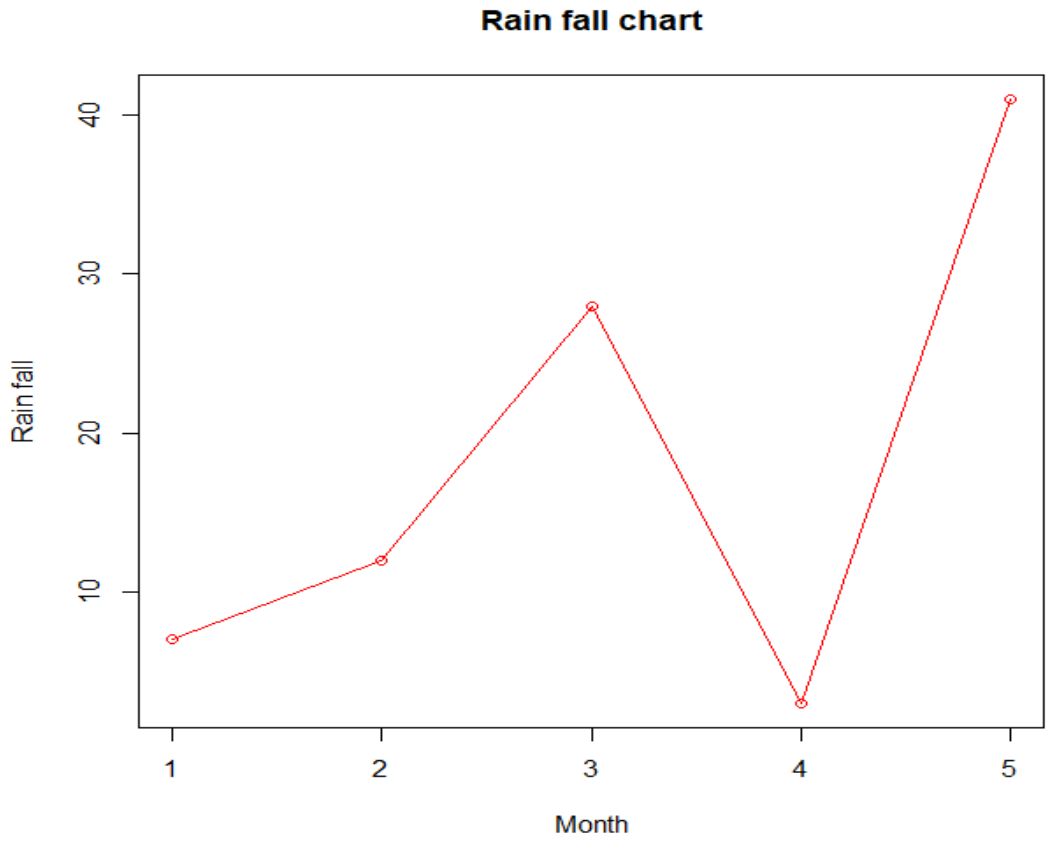
Syntax

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.

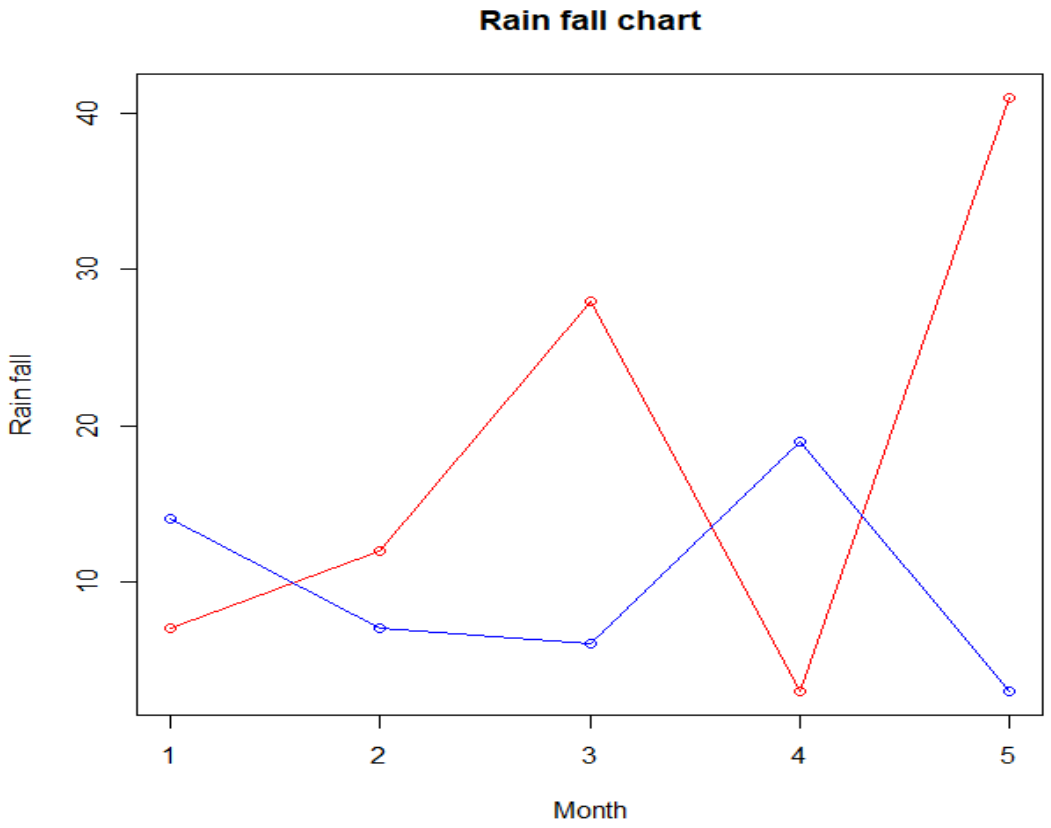
Example:

```
> # Create the data for the chart.
> 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: More than one line can be drawn on the same chart by using the **lines()**function.

```
> plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall", main = "Rain fall chart")
>
>lines(t, type = "o", col = "blue")
```



R - Bar Charts: A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable.

R uses the function **barplot()** to create bar charts.

Syntax

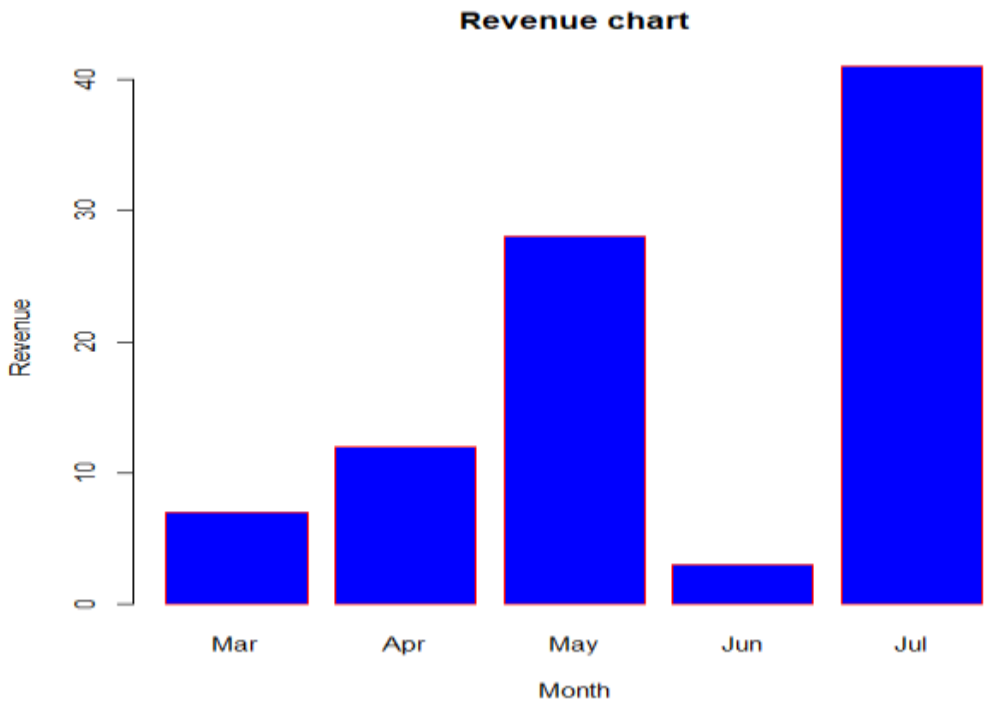
barplot(H,xlab,ylab,main, names.arg,col)

Following is the description of the parameters used −

- **H** is a vector or matrix containing numeric values used in bar chart.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the title of the bar chart.
- **names.arg** is a vector of names appearing under each bar.
- **col** is used to give colors to the bars in the graph.

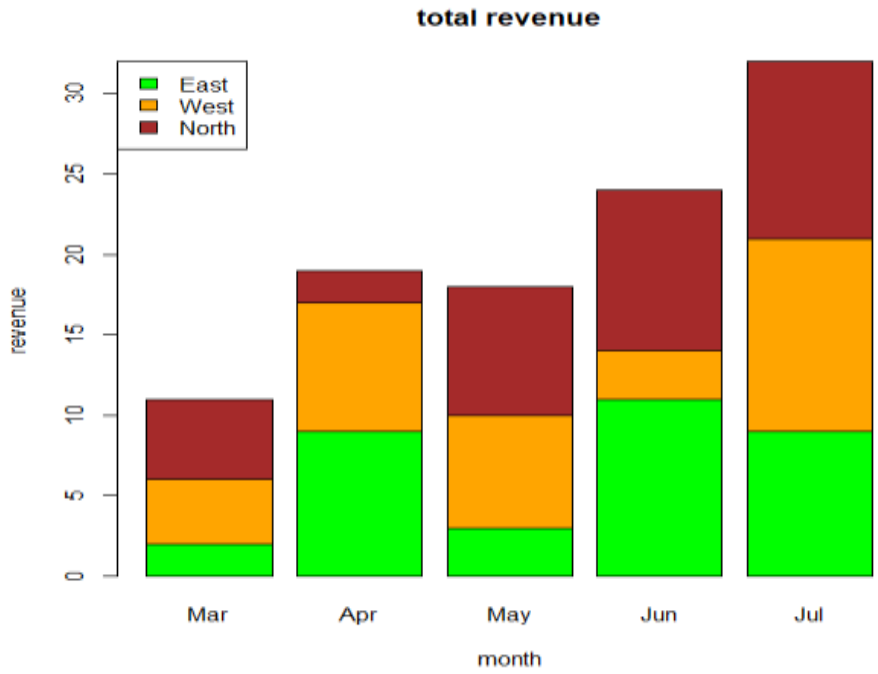
Example

```
> # Create the data for the chart
> H <- c(7,12,28,3,41)
> M <- c("Mar","Apr","May","Jun","Jul")
> # Plot the bar chart
> barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="blue“, main="Revenue chart",border="red")
```



Group Bar Chart and Stacked Bar Chart

```
> # Create the input vectors.
> colors = c("green","orange","brown")
> months <- c("Mar","Apr","May","Jun","Jul")
> regions <- c("East","West","North")
>
> # Create the matrix of the values.
> Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11), nrow = 3, ncol = 5, byrow = TRUE)
>
> # Create the bar chart
> barplot(Values, main = "total revenue", names.arg = months, xlab = "month", ylab = "revenue", col
= colors)
>
> # Add the legend to the chart
> legend("topleft", regions, fill = colors)
```



R - Pie Charts: pie chart is created using the **pie()** function which takes positive numbers as a vector input.

Syntax:

pie(x, labels, radius, main, col, clockwise)

Following is the description of the parameters used −

- **x** is a vector containing the numeric values used in the pie chart.
- **labels** is used to give description to the slices.
- **radius** indicates the radius of the circle of the pie chart.(value between −1 and +1).
- **main** indicates the title of the chart.
- **col** indicates the color palette.
- **clockwise** is a logical value indicating if the slices are drawn clockwise or anti clockwise.

```
> # Create data for the graph.
> x <- c(21, 62, 10, 53)
>labels <- c("London", "New York", "Singapore", "Mumbai")

> # Plot the chart with title and rainbow color pallet.
>pie(x, labels, main = "City pie chart", col = rainbow(length(x)))
> legend("topright", c("London","New York","Singapore","Mumbai"), fill=rainbow(length(x)))
```

R – Histograms: A histogram represents the frequencies of values of a variable bucketed into ranges. Histogram is similar to bar chat but the difference is it groups the values into continuous ranges.

R creates histogram using **hist()** function.

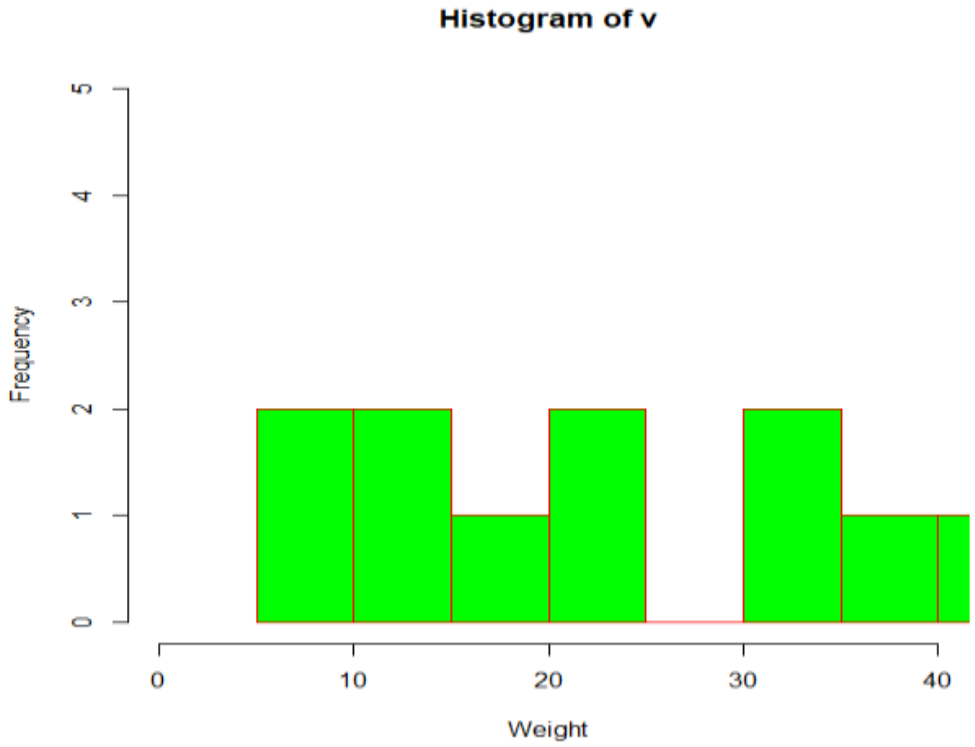
Syntax:

hist(v,main,xlab,xlim,ylim,breaks,col,border)

Following is the description of the parameters used −
- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to mention the width of each bar.

Example

```
> # Create data for the graph.
> v <- c(9,13,21,8,36,22,12,41,31,33,19)
>
> # Create the histogram.
> hist(v,xlab = "Weight",col = "green",border = "red", xlim = c(0,40), ylim = c(0,5), breaks = 5)
```



R – Scatterplots: Scatterplots show many points plotted in the Cartesian plane. The simple scatterplot is created using the **plot()** function.

Syntax

plot(x, y, main, xlab, ylab, xlim, ylim, axes)

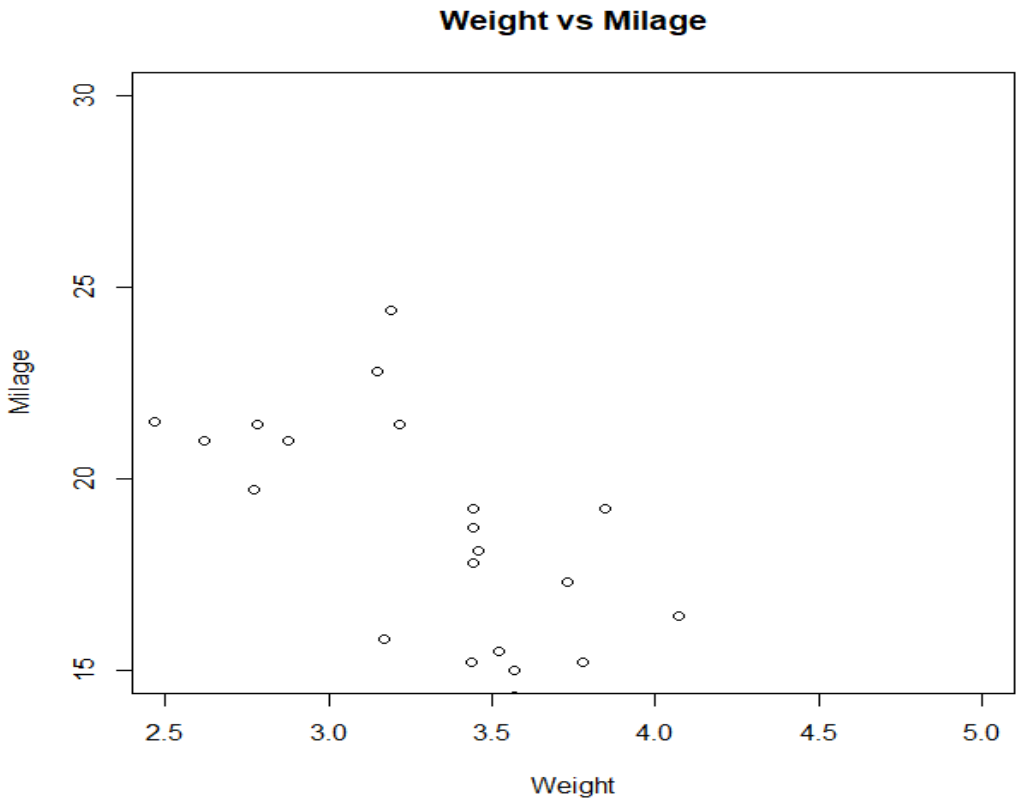
Following is the description of the parameters used −

- **x** is the data set whose values are the horizontal coordinates.
- **y** is the data set whose values are the vertical coordinates.
- **main** is the tile of the graph.
- **xlab** is the label in the horizontal axis.
- **ylab** is the label in the vertical axis.
- **xlim** is the limits of the values of x used for plotting.
- **ylim** is the limits of the values of y used for plotting.
- **axes** indicates whether both axes should be drawn on the plot.

Example

```
> # Get the input values.
>input <- mtcars[,c('wt','mpg')]

> # Plot the chart for cars with weight between 2.5 to 5 and mileage between 15 and 30.
> plot(x = input$wt,y = input$mpg,  xlab = "Weight",  ylab = "Milage",   xlim = c(2.5,5),
+    ylim = c(15,30),  main = "Weight vs Milage" )
```



R – Boxplots: Boxplots are a measure of how well distributed is the data in a data set. It divides the data set into three quartiles.

•This graph represents the minimum, maximum, median, first quartile and third quartile in the data set.

•Useful in comparing the distribution of data across data sets by drawing boxplots for each of them.

•Boxplots are created in R by using the **boxplot()** function.

Syntax:

boxplot(x, data, notch, varwidth, names, main)

Following is the description of the parameters used −

•**x** is a vector or a formula.

•**data** is the data frame.

•**notch** is a logical value. Set as TRUE to draw a notch.

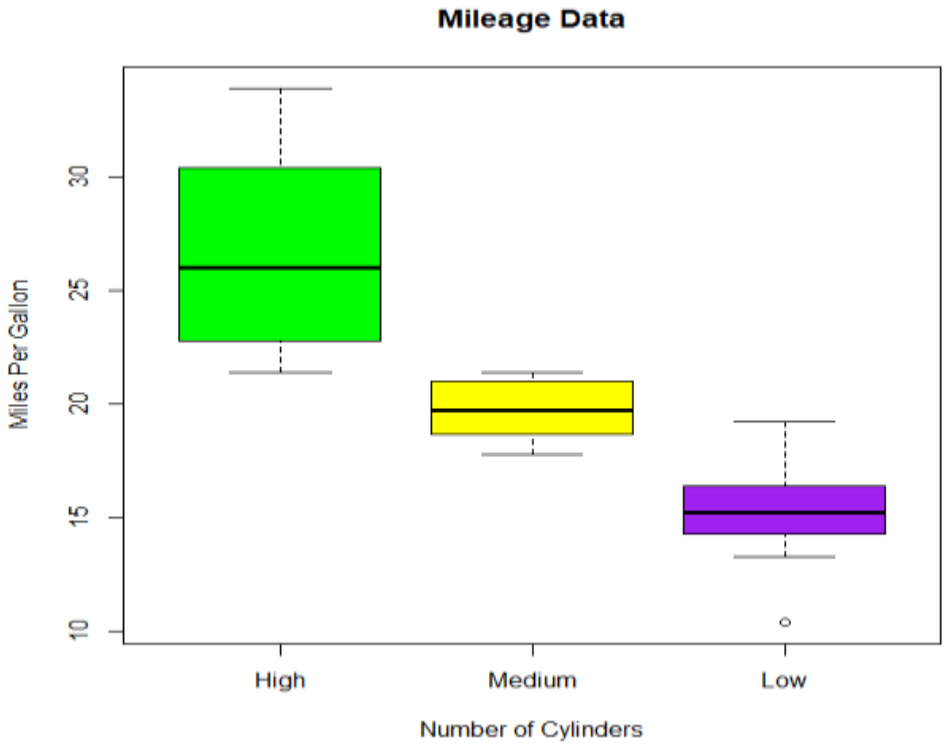
•**varwidth** is a logical value. Set as true to draw width of the box proportionate to the sample size.

•**names** are the group labels which will be printed under each boxplot.

•**main** is used to give a title to the graph.

Example

```
>input <- mtcars[,c('mpg','cyl')]
> # Plot the chart.
> boxplot(mpg~cyl, data = mtcars, xlab="Number of Cylinders", ylab="Miles Per Gallon",
names = c("High","Medium","Low"), col = c("green","yellow","purple"),main="Mileage
Data“)
```



R Graphical Devices

A graphics device is something where we can delineate a plot. When we make a plot in R, it has to be "sent" to a specific:

- Window on your computer (screen device)
- PDF file (file device)
- PNG or JPEG file (file device)
- Scalable vector graphics (SVG) file (file device)

The following devices are currently available:

- **PDF –** Write PDF graphics commands to file.
- **xfig –** Device for XFIG graphics file format.
- **pictex –** Writes TeX/PicTeX graphics commands to a file (of historical interest only).
- **postscript –** Writes PostScript graphics commands to a file.
- **bitmap –** bitmap pseudo-device via Ghostscript (if available).
- **SVG –** Device for the SVG File Format
- **libgd –** Supports fast image creation
- **GTK –** Supports gtkDevice.
- **quartz –** Graphics Device Driver for MacOS.
- **x11 –** Provides functions in Windows for opening interactive graphics device.

The following devices will be functional if R is compiled to use them:

- **png** – PNG bitmap device
- **jpeg** – JPEG bitmap device
- **bmp** – BMP bitmap device
- **tiff** – TIFF bitmap device
- **X11** – This is for the X11 windowing system.
- **svg** – SVG device based on Cairo graphics.
- **cario.pdf**, – cairo_ps PDF and PostScript devices based on Cairo graphics.
- **quartz** – The graphics device for the macOS native Quartz 2d graphics system.

R runs on many different operating systems and also supports different graphics formats.

Format	Driver	Notes
JPEG	jpeg	It is used everywhere but it does not facilitate resizing.
PNG	png	similar to JPEG and it does not resize.
WMF for	win.metafile	It is based on the Windows platform only. It is best suited MS word and it facilitates flexible resizing.
PDF	pdf	pdflatex; easily resizable
Postscript	postscript	latex and Open Office; easily resizable

In order to save graphics to an image file, there are three steps in R:

- You can create a graphics device of PNG format using png(), JPG format using jpg() and PDF format using pdf().
- Plot your data.
- Closing the graphics device and saving the image using dev.off.

```
> # 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()
```