Data Visualization in R

Soumen Ghosh

Indian Institute of Information Technology Chittoor,
Sri City, Andhra Pradesh.

February 28, 2018

Need of Data Visualization???

The way human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports.

Before we perform any analysis and come up with any assumptions about the distributions of and relationships between variables in our datasets, it is always a good idea to visualize our data in order to understand their properties and identify appropriate analytics techniques.

Tools for Data Visualization

The basic six type of visualization tools are available in R. These are given bellow:

- Histograms
- Boxplots
- Scatterplots
- Line Graphs
- Pie Charts
- Bar Charts

Histogram

Histogram is basically a plot that breaks the data into bins (or breaks) and shows frequency distribution of these bins. You can change the breaks also and see the effect it has data visualization in terms of understandability. The basic syntax for creating a histogram using R is

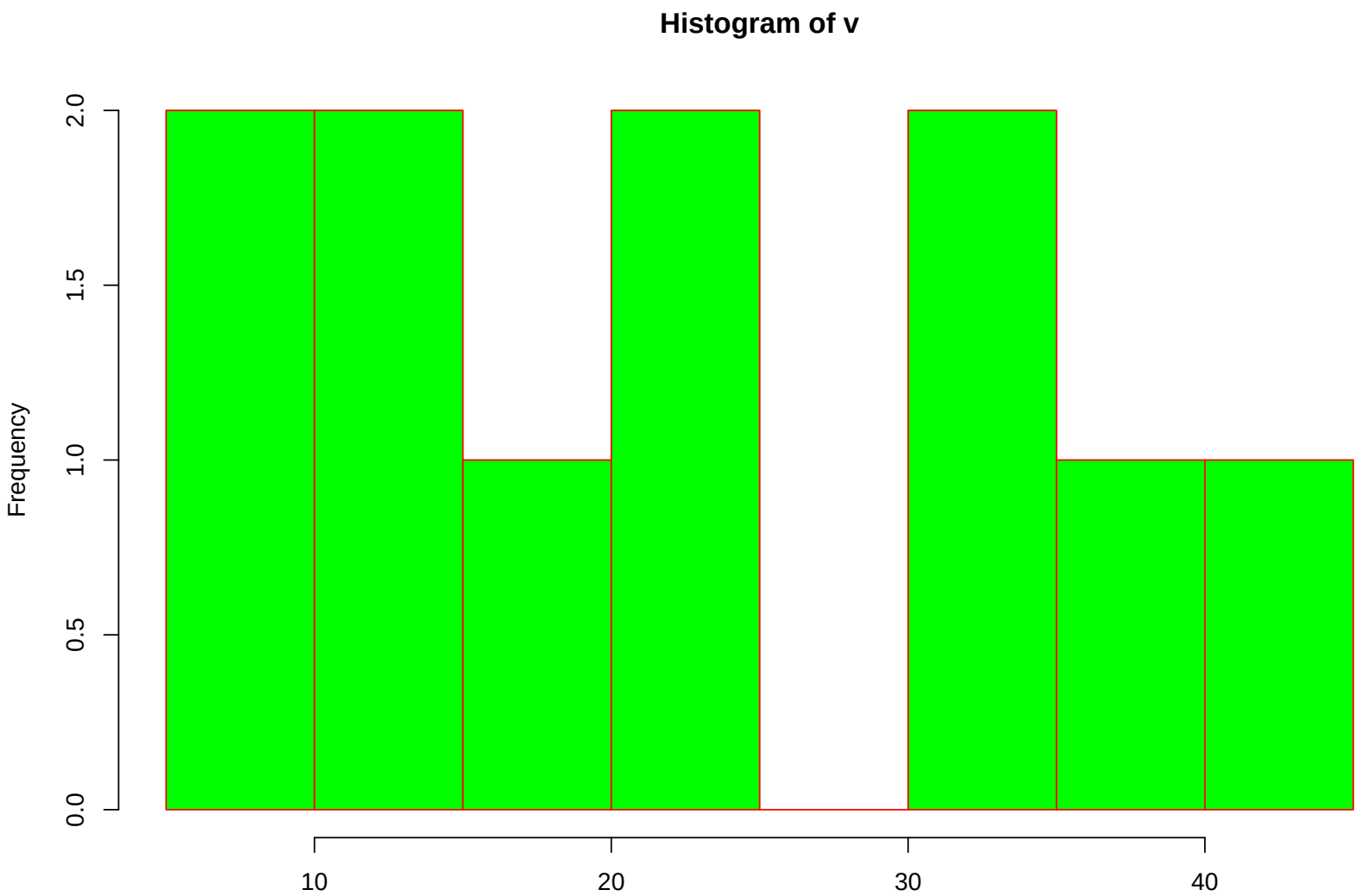
hist(v, main, xlab, xlim, ylim, breaks, col, border)

- **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.

```
v <-  c(9,13,21,8,36,22,12,41,31,33,19)
```

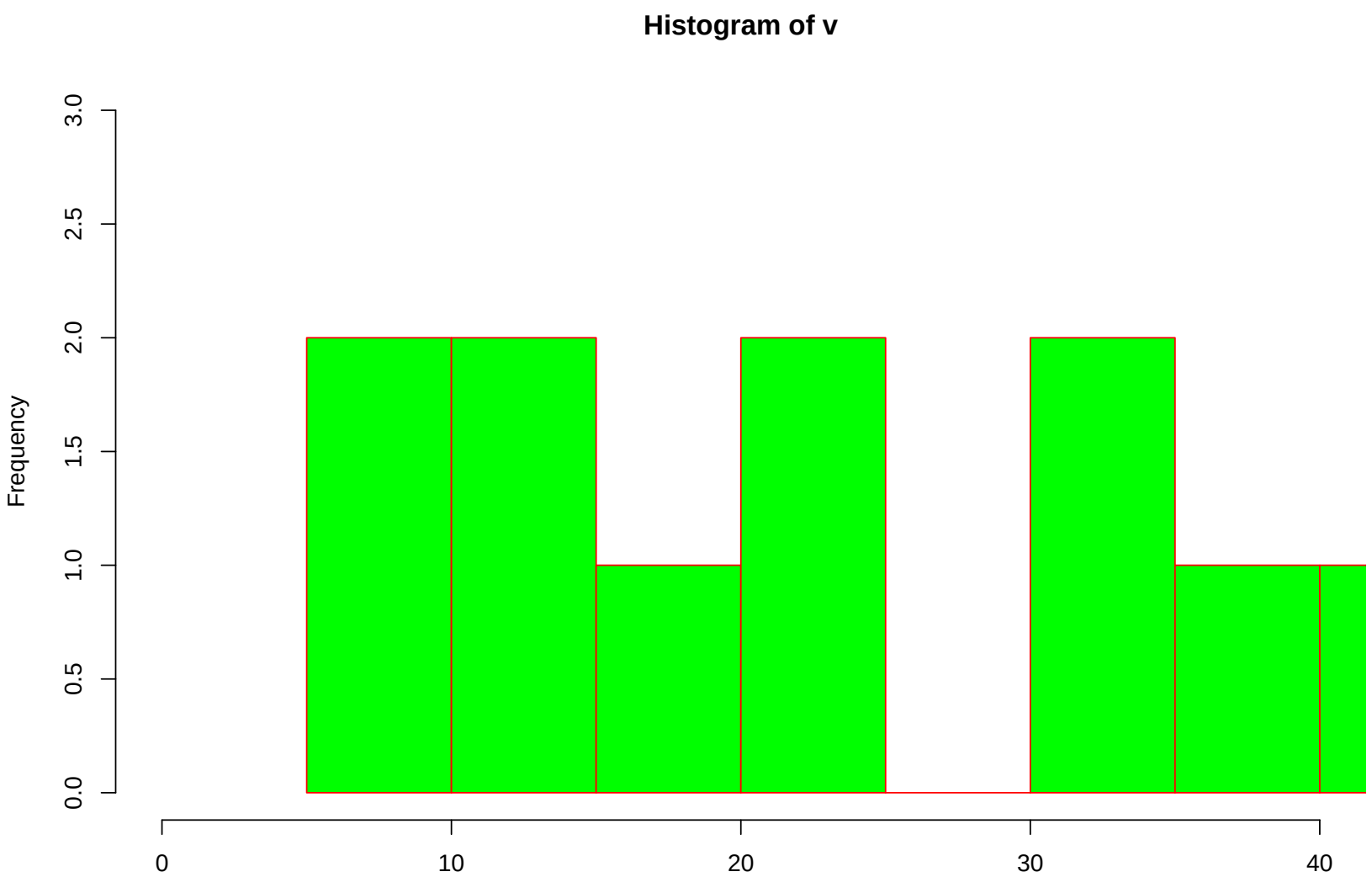
Create the Histogram

```
hist(v,xlab = "Weight", col = "green", border = "red")
```



Histogram with X and Y limit

```
hist(v, xlim = c(0,40), ylim = c(0,3), breaks = 5, xlab = "Wei
```



Structure and Summary of the Data

```
str(VADeaths)
```

```
##  num [1:5, 1:4] 11.7 18.1 26.9 41 66 8.7 11.7 20.3 30.9 54.
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:5] "50-54" "55-59" "60-64" "65-69" ...
##   ..$ : chr [1:4] "Rural Male" "Rural Female" "Urban Male"
```

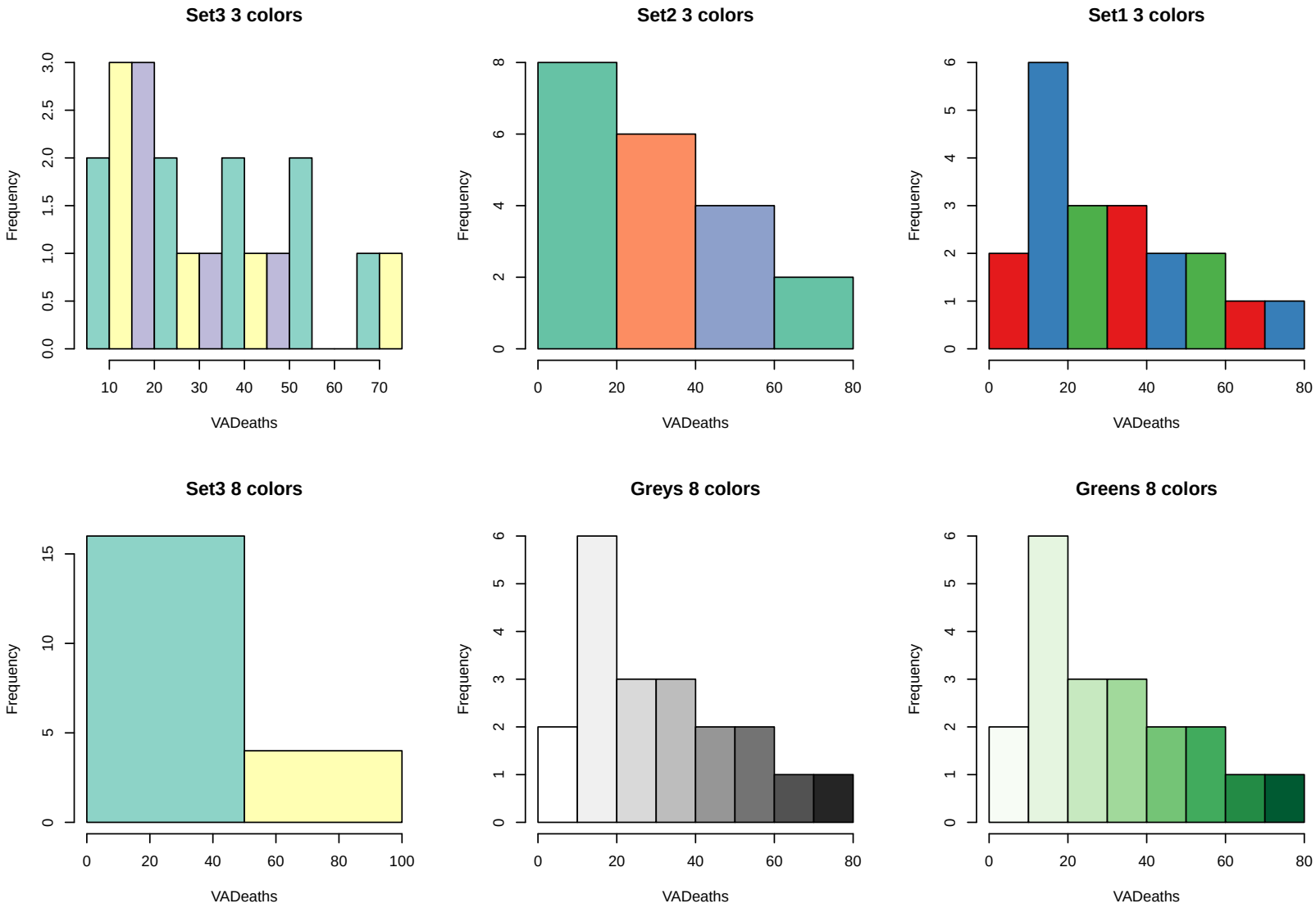
```
summary(VADeaths)
```

```
##    Rural Male      Rural Female     Urban Male      Urban Fem
##  Min.   :11.70   Min.   : 8.70   Min.   :15.40   Min.   : 8
##  1st Qu.:18.10   1st Qu.:11.70   1st Qu.:24.30   1st Qu.:13
##  Median :26.90   Median :20.30   Median :37.00   Median :19
##  Mean   :32.74   Mean   :25.18   Mean   :40.48   Mean   :25
##  3rd Qu.:41.00   3rd Qu.:30.90   3rd Qu.:54.60   3rd Qu.:35
##  Max.   :66.00   Max.   :54.30   Max.   :71.10   Max.   :50
```

Histogram with Different Color I

```
library(RColorBrewer)
data(VADeaths)
par(mfrow=c(2,3))
hist(VADeaths,breaks=10, col=brewer.pal(3,"Set3"),
     main="Set3 3 colors")
hist(VADeaths,breaks=3 ,col=brewer.pal(3,"Set2"),
     main="Set2 3 colors")
hist(VADeaths,breaks=7, col=brewer.pal(3,"Set1"),
     main="Set1 3 colors")
hist(VADeaths,,breaks= 2, col=brewer.pal(8,"Set3"),
     main="Set3 8 colors")
hist(VADeaths,col=brewer.pal(8,"Greys"),
     main="Greys 8 colors")
hist(VADeaths,col=brewer.pal(8,"Greens"),
     main="Greens 8 colors")
```

Histogram with Different Color II



Boxplot

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. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.

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

- **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.

Loading Data

```
data(iris)
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 .
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",.
```

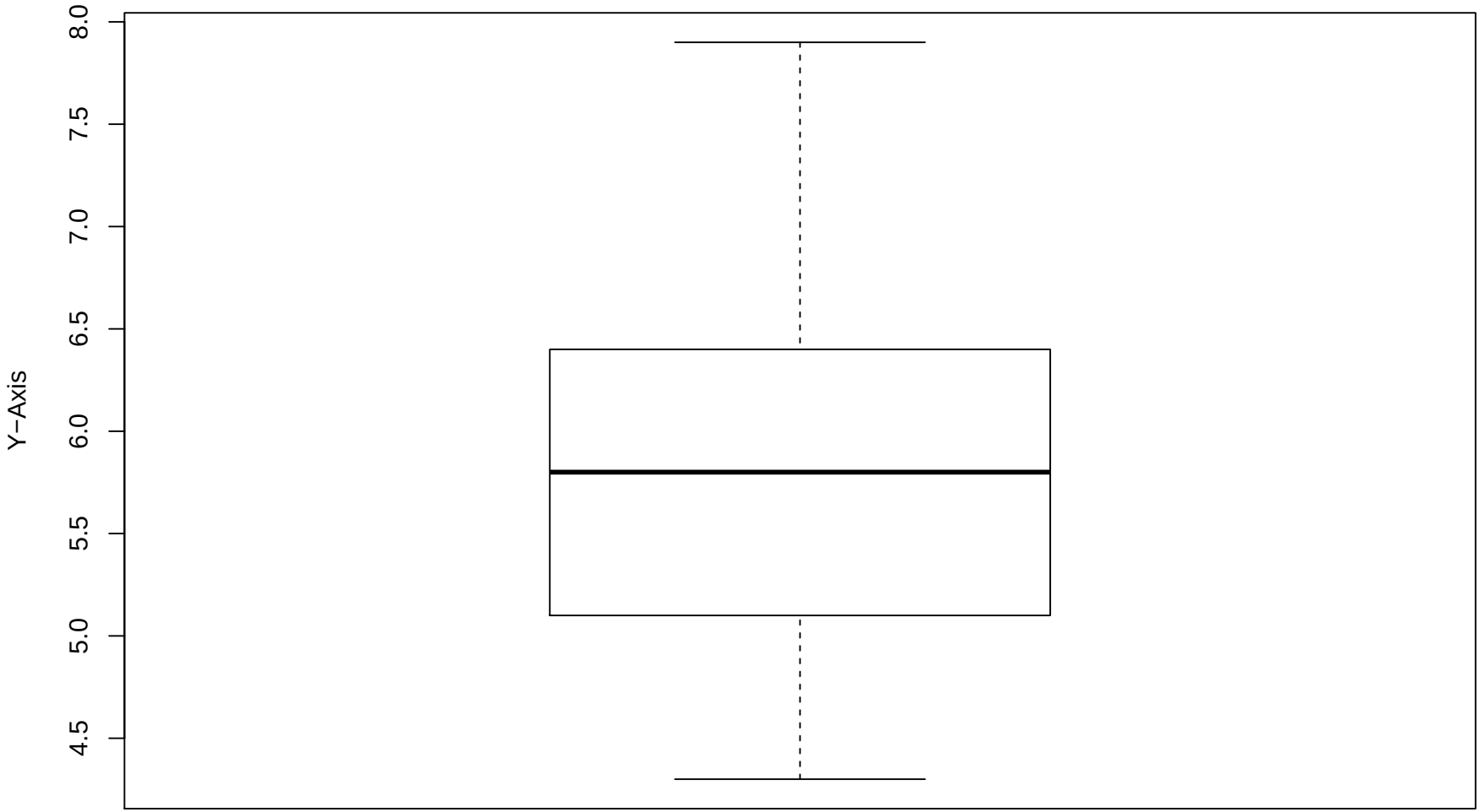
Summary of the Data

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Wid
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.
##  Median :5.800   Median :3.000   Median :4.350   Median :1.
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Creating a Boxplot

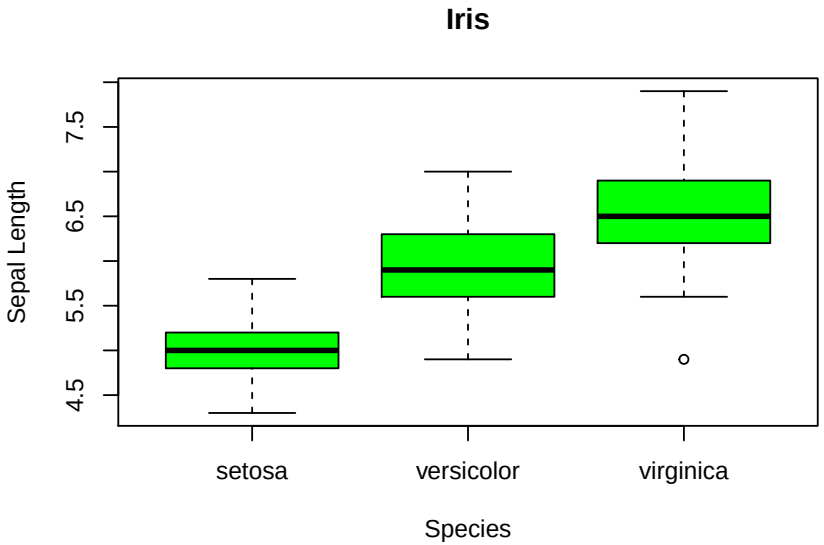
```
boxplot(iris$Sepal.Length, data = iris, xlab = "X-Axis", ylab
```

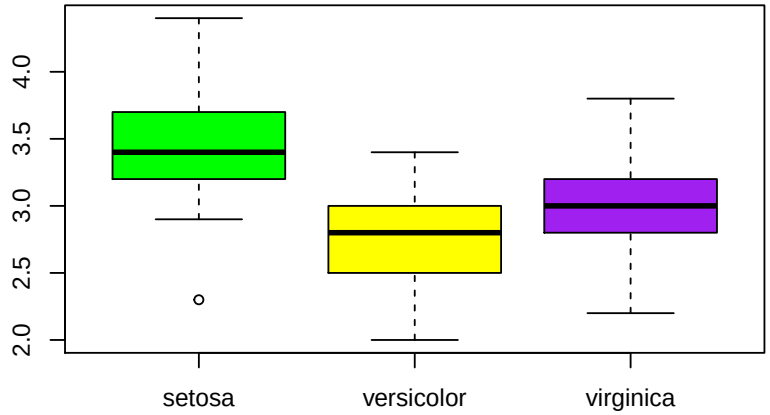


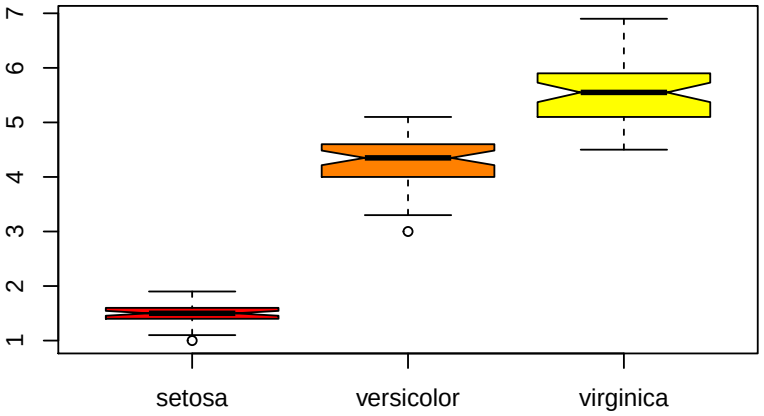
Boxplot for Iris Data I

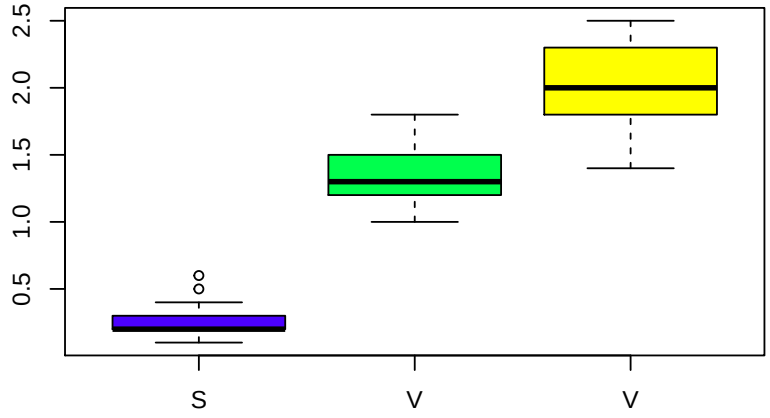
```
par(mfrow=c(2,2))
boxplot(iris$Sepal.Length~iris$Species, col = "green",
        xlab = "Species", ylab = "Sepal Length",
        main = "Iris")
boxplot(iris$Sepal.Width~iris$Species,
        col = c("green","yellow","purple"))
boxplot(iris$Petal.Length~iris$Species, col = heat.colors(3),
        notch=TRUE, varwidth = TRUE)
boxplot(iris$Petal.Width~iris$Species, col = topo.colors(3),
        names = c("S", "V", "V"))
```

Boxplot for Iris Data II









Scatterplot

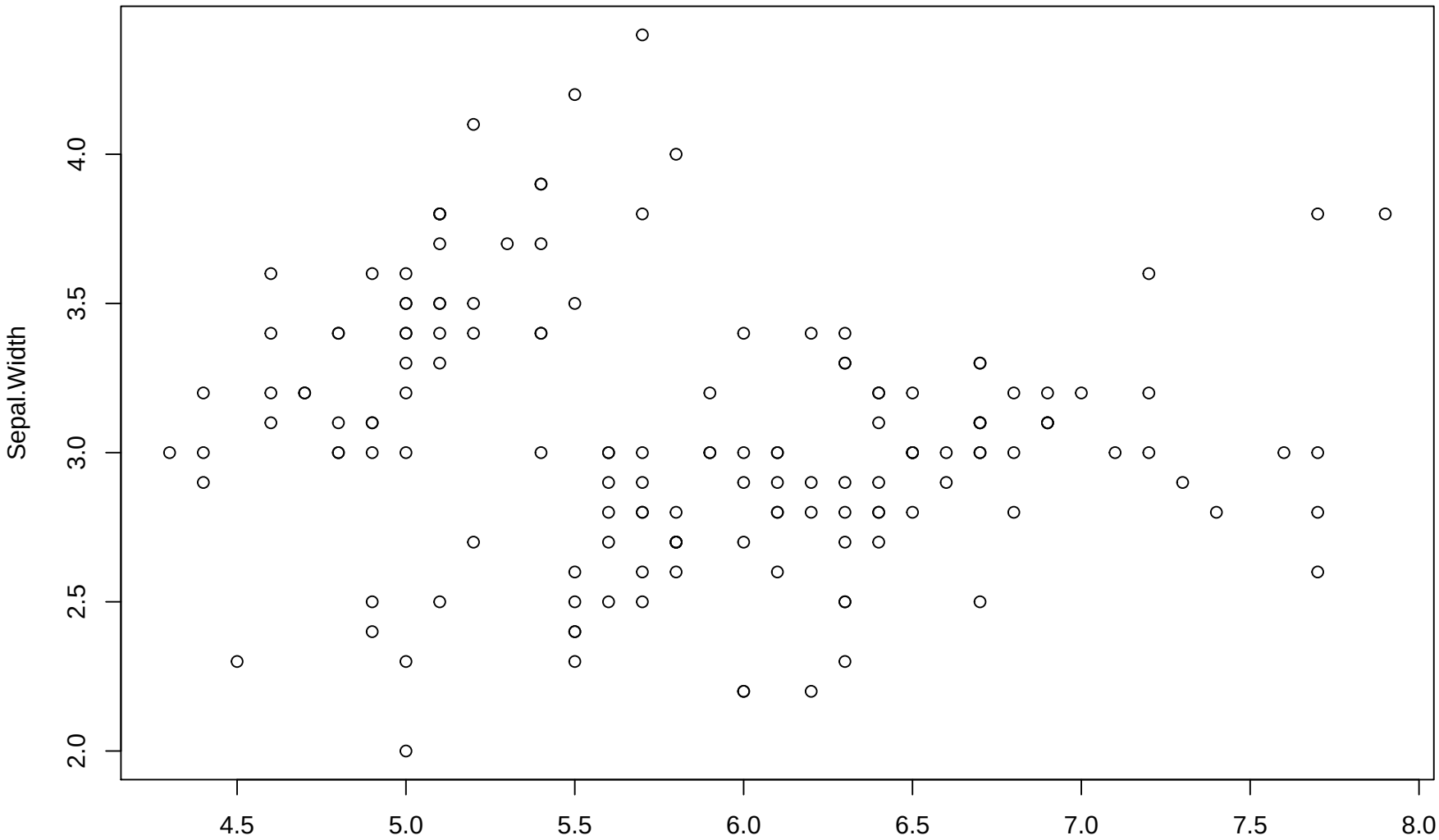
Scatterplots show many points plotted in the Cartesian plane. Each point represents the values of two variables. One variable is chosen in the horizontal axis and another in the vertical axis.

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

- **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.

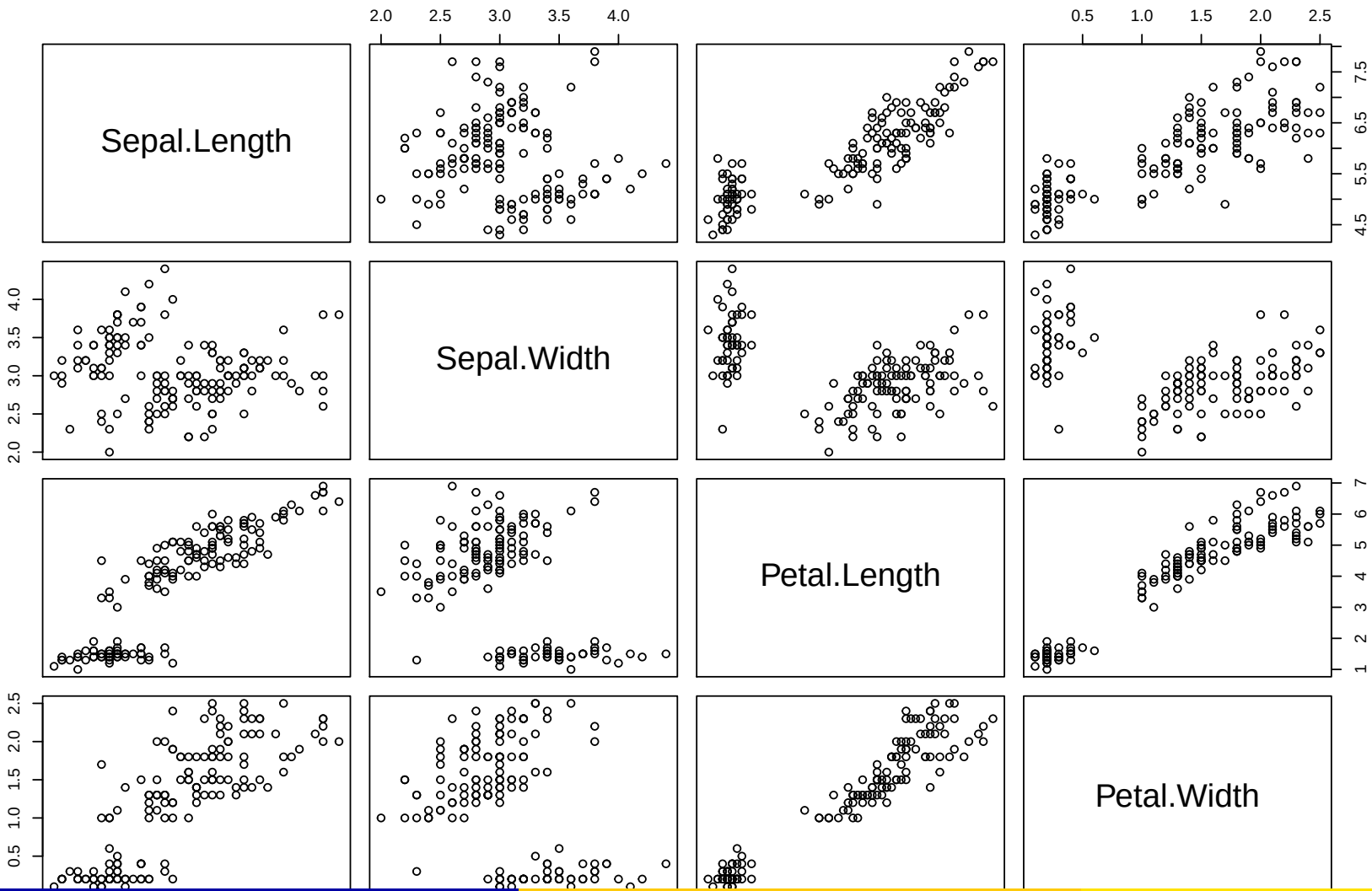
Creating a Scatterplot

```
plot(x = iris$Sepal.Length, y = iris$Sepal.Width, type="p", xl
```



Scatterplot for Iris Data

```
plot(iris[,-5])
```



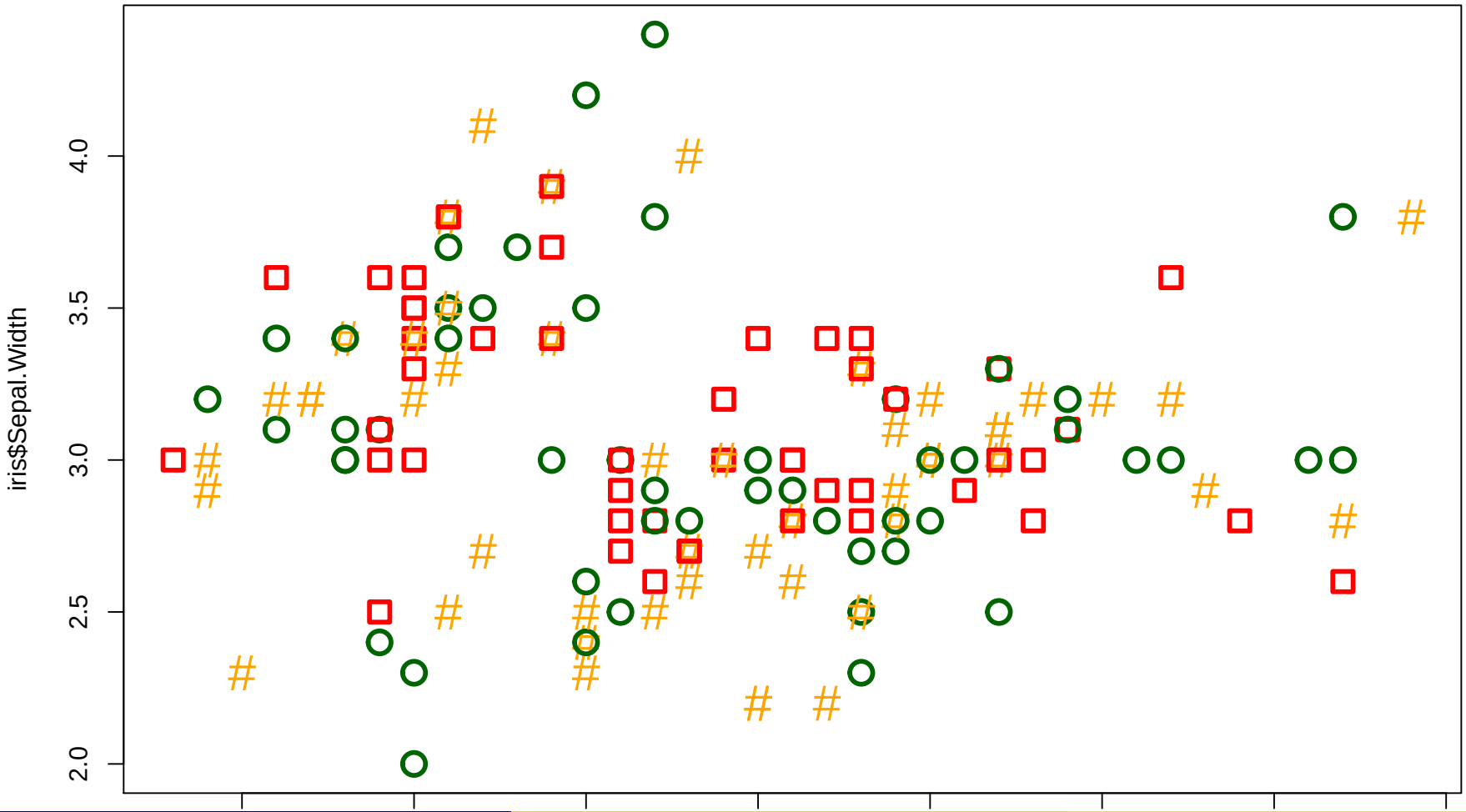
Advanced Scatterplot

There are numerous graphical arguments available to functions in R. In this tutorial, just a few of the common aesthetic options will be addressed below

- **col:** determines the colors used for points and lines; accepts character strings of color names (i.e. “red”, “green”, etc.)
- **pch:** the type of point to use (i.e. circle, square, triangle, etc.); accepts values 0-25 for symbols and 32-255 for characters
- **cex:** the amount to scale the size of points; accepts a numeric value; default is 1
- **lty:** defines the line type; accepts various character strings (i.e. “solid”, “dashed”, “dotted”, etc.)
- **lwd:** defines the line width; accepts a positive number; default is 1

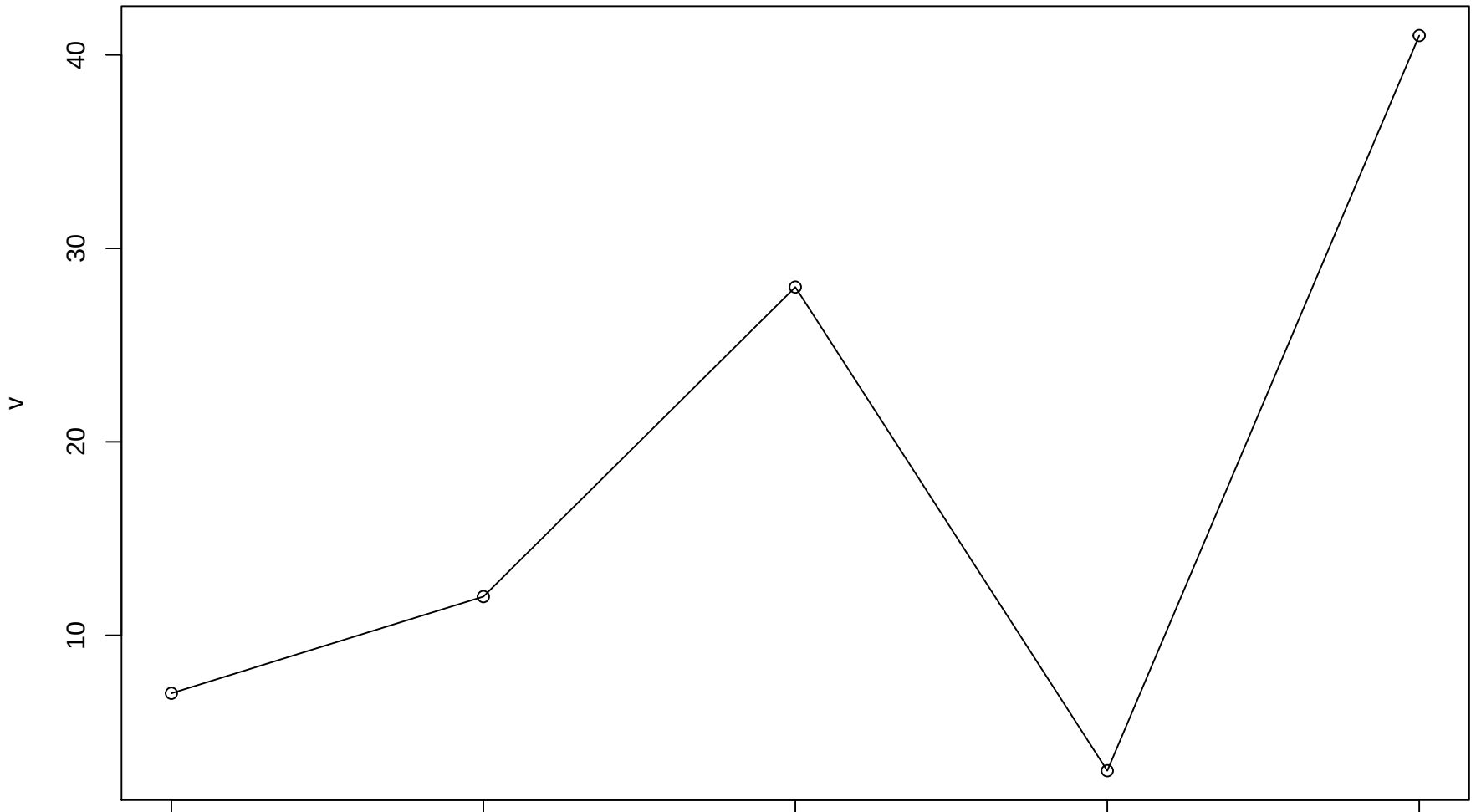
Creating Advanced Scatterplot

```
plot(x = iris$Sepal.Length, y = iris$Sepal.Width, col = c("dar
     pch = c(21, 22, 35), cex = 2, lty = "solid", lwd = 3)
```



Line Graphs

```
v <- c(7,12,28,3,41)
plot(v, type = "o")
```



Type of Plot

what type of plot should be drawn. Possible types are

- “p” for points,
- “l” for lines,
- “b” for both,
- “c” for the lines part alone of “b”,
- “o” for both ‘overplotted’,
- “h” for ‘histogram’ like (or ‘high-density’) vertical lines,
- “s” for stair steps,
- “S” for other steps, see ‘Details’ below,
- “n” for no plotting.

Pie Charts

A pie-chart is a representation of values as slices of a circle with different colors. The slices are labeled and the numbers corresponding to each slice is also represented in the chart.

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

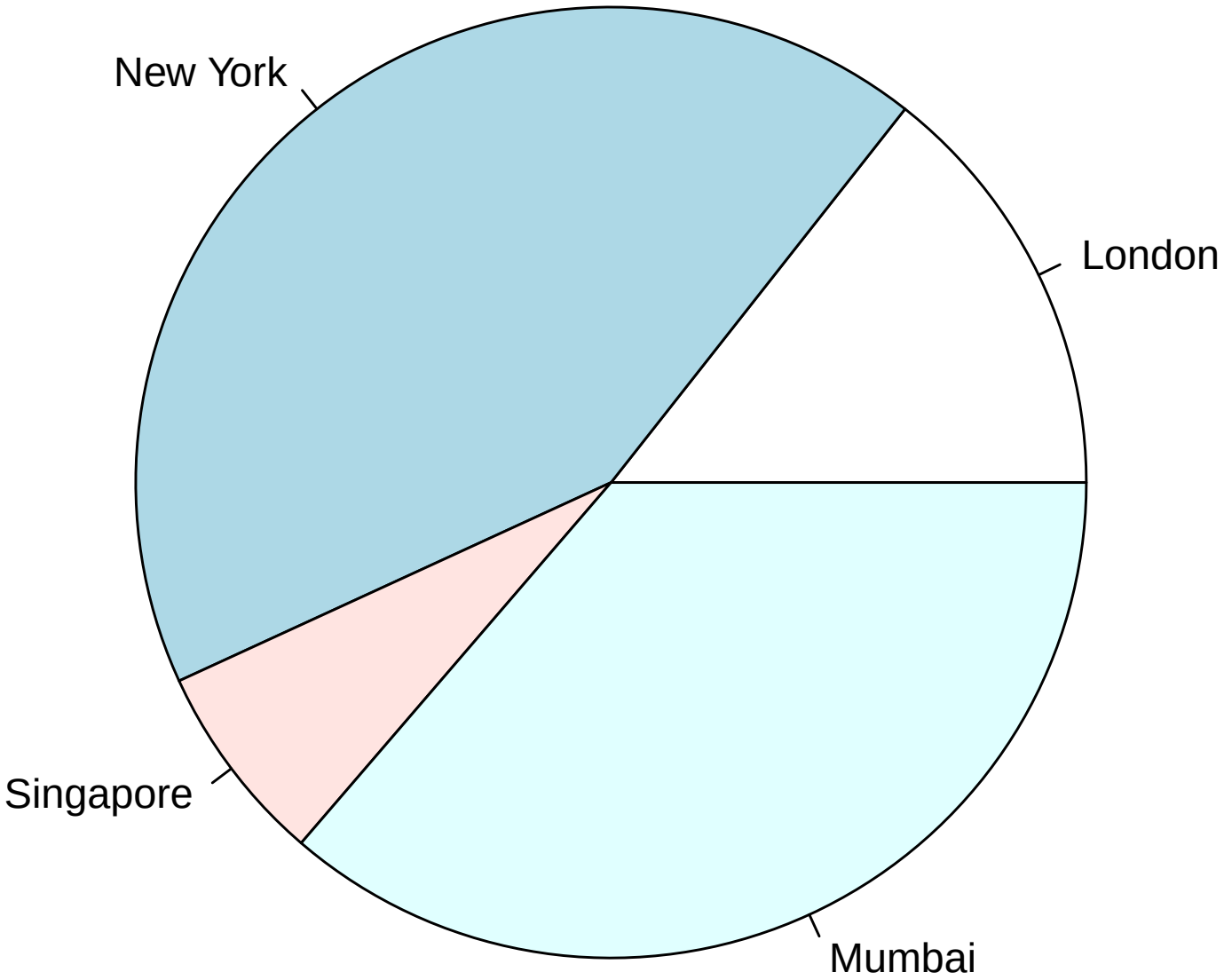
- **x** is a vector containing the numeric values used in the pie chart.
- **labels** is used to give description to the slices.
- **radius** indicates the radius of the circle of the pie chart.(value between -1 and $+1$).
- **main** indicates the title of the chart.
- **col** indicates the color palette.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

Creating Pie Chart I

```
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore", "Mumbai")

pie(x,labels)
```

Creating Pie Chart II



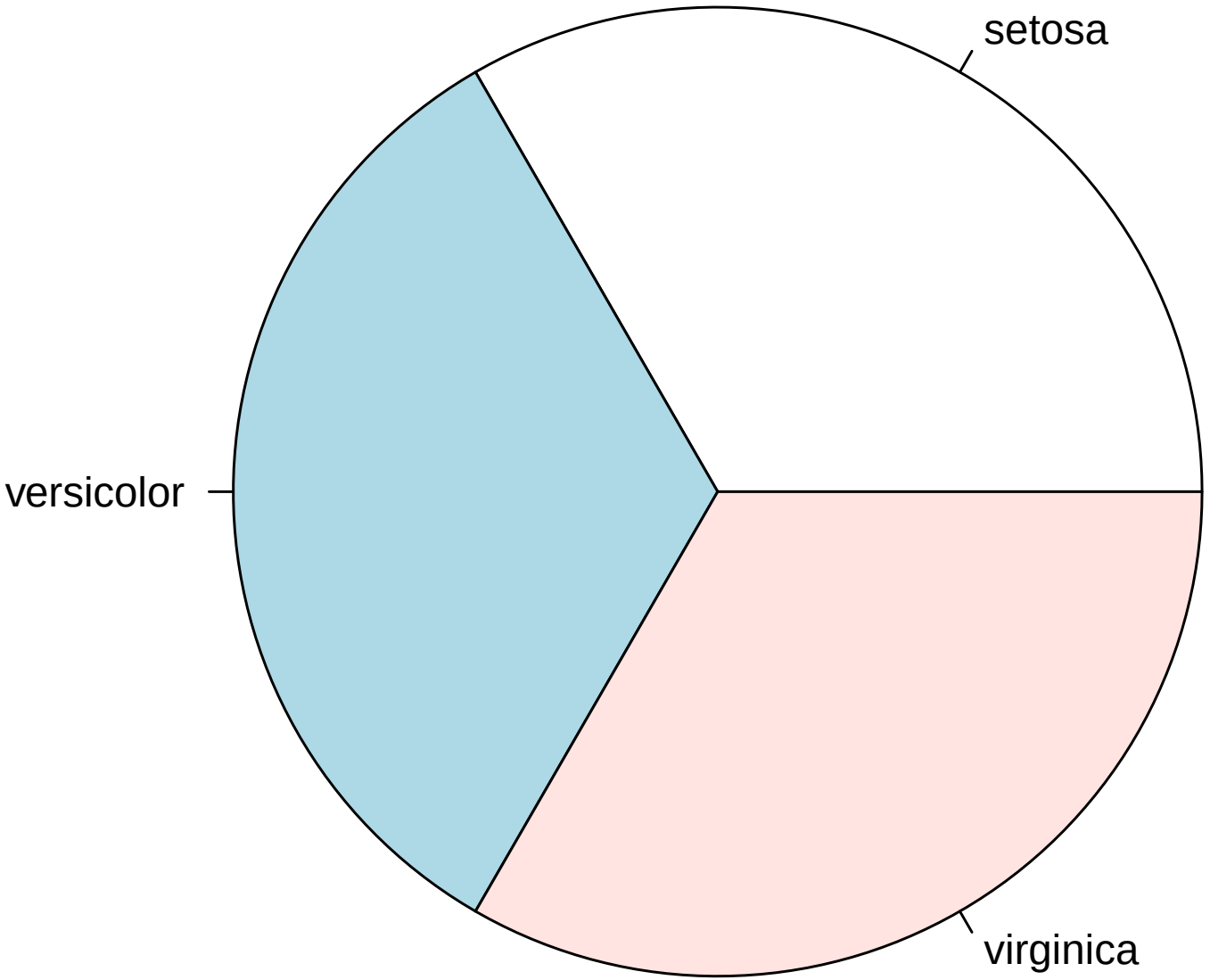
Pie Chart for Iris Data

```
data("iris")
summary(iris[,5])
```

```
##     setosa versicolor  virginica
##         50         50         50
```

```
a <- summary(iris[,5])
pie(a)
```

Pie Chart for Iris Data

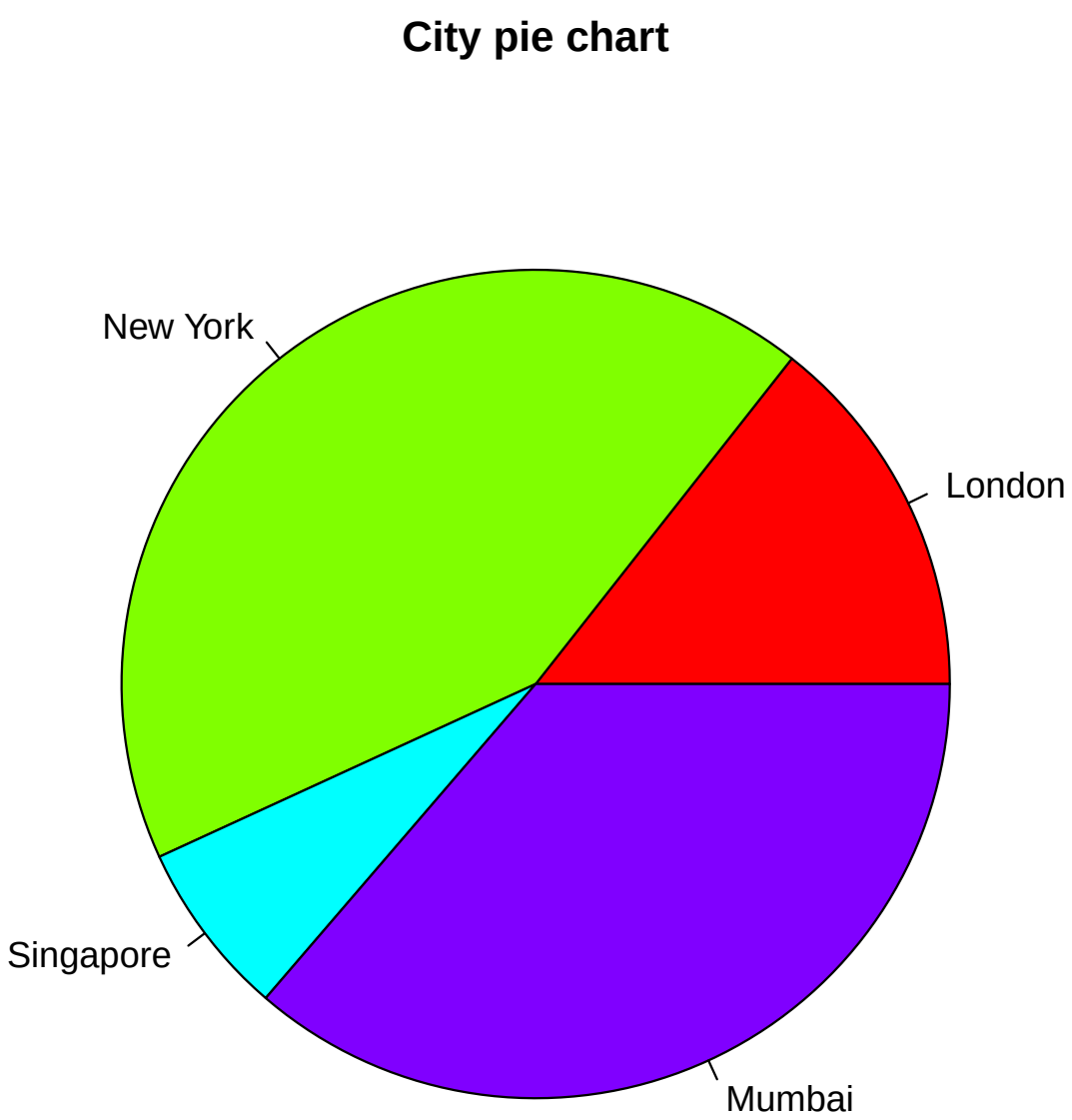


Pie Chart with Colors I

```
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore", "Mumbai")

pie(x, labels, main = "City pie chart",
    col = rainbow(length(x)))
```

Pie Chart with Colors II

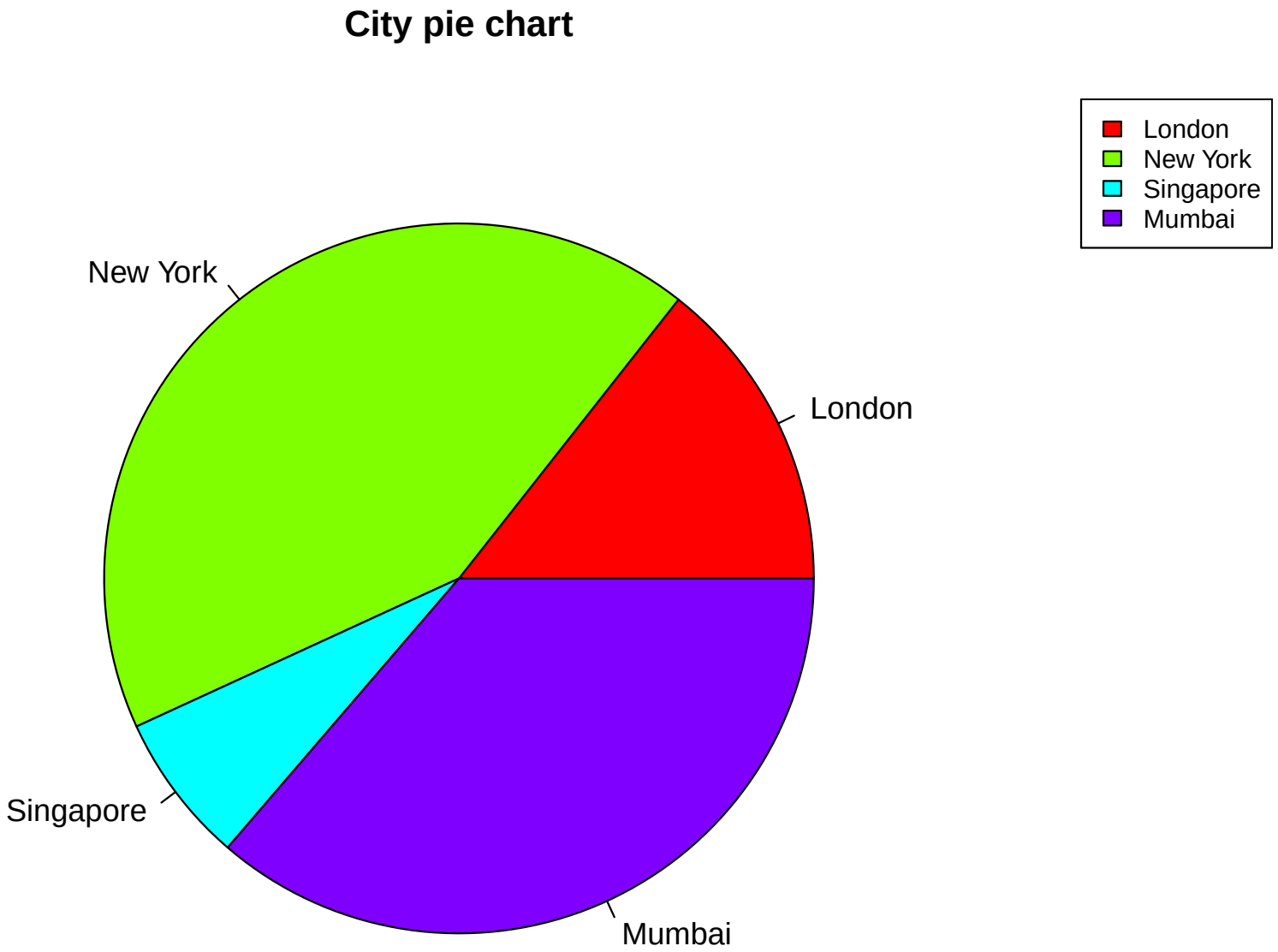


Pie Chart with Legend I

```
x <- c(21, 62, 10, 53)


pie(x, labels <- c("London", "New York", "Singapore", "Mumbai"
legend("topright", c("London","New York","Singapore","Mumbai"
   fill = rainbow(length(x)))
```

Pie Chart with Legend II



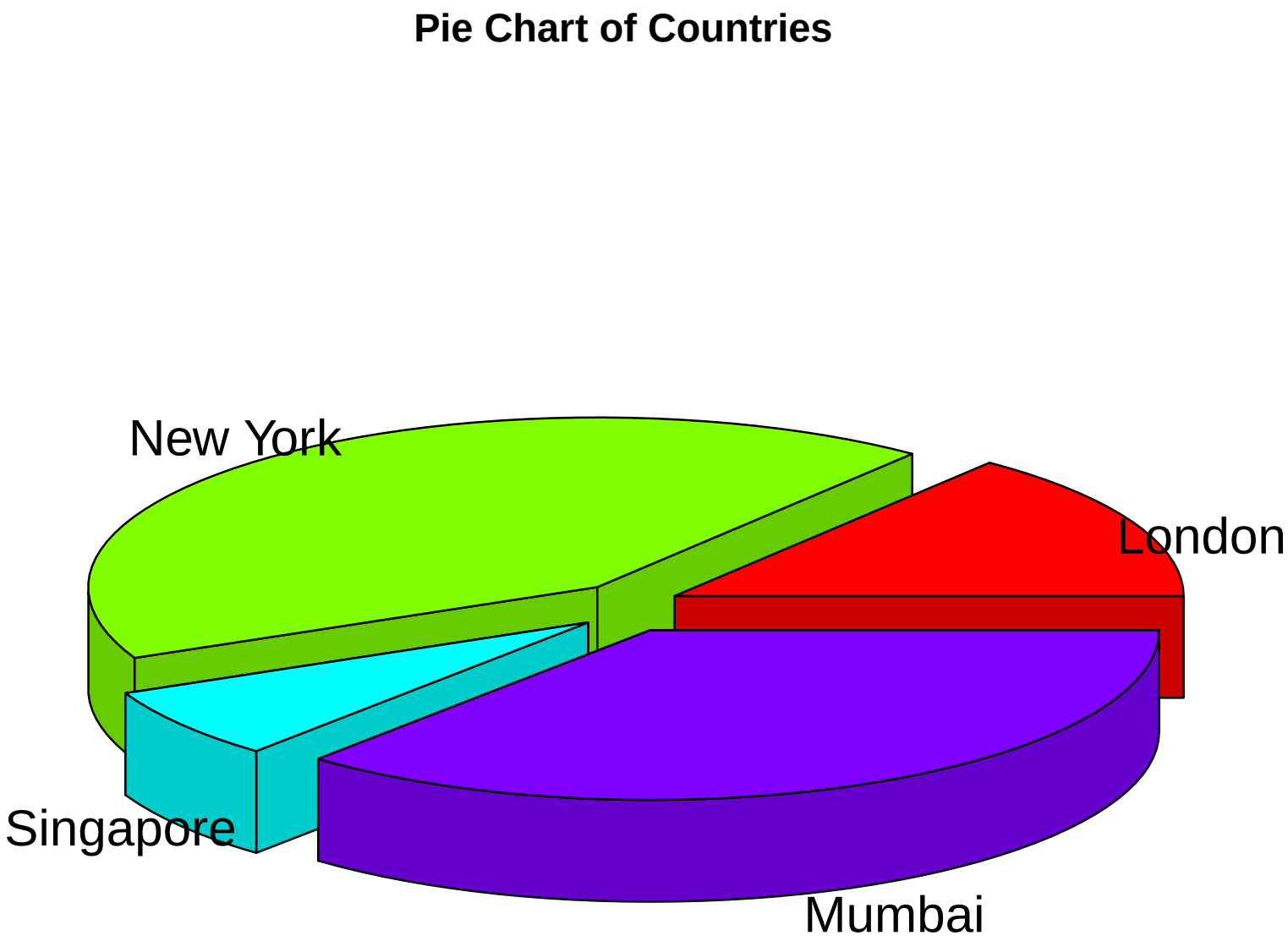
3D Pie Chart I

```
library(plotrix)

x <-  c(21, 62, 10,53)

pie3D(x,labels = c("London","New York","Singapore","Mumbai"),
      explode = 0.1, main = "Pie Chart of Countries ")
```

3D Pie Chart II



Bar Charts

A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R can draw both vertical and horizontal bars in the bar chart.

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

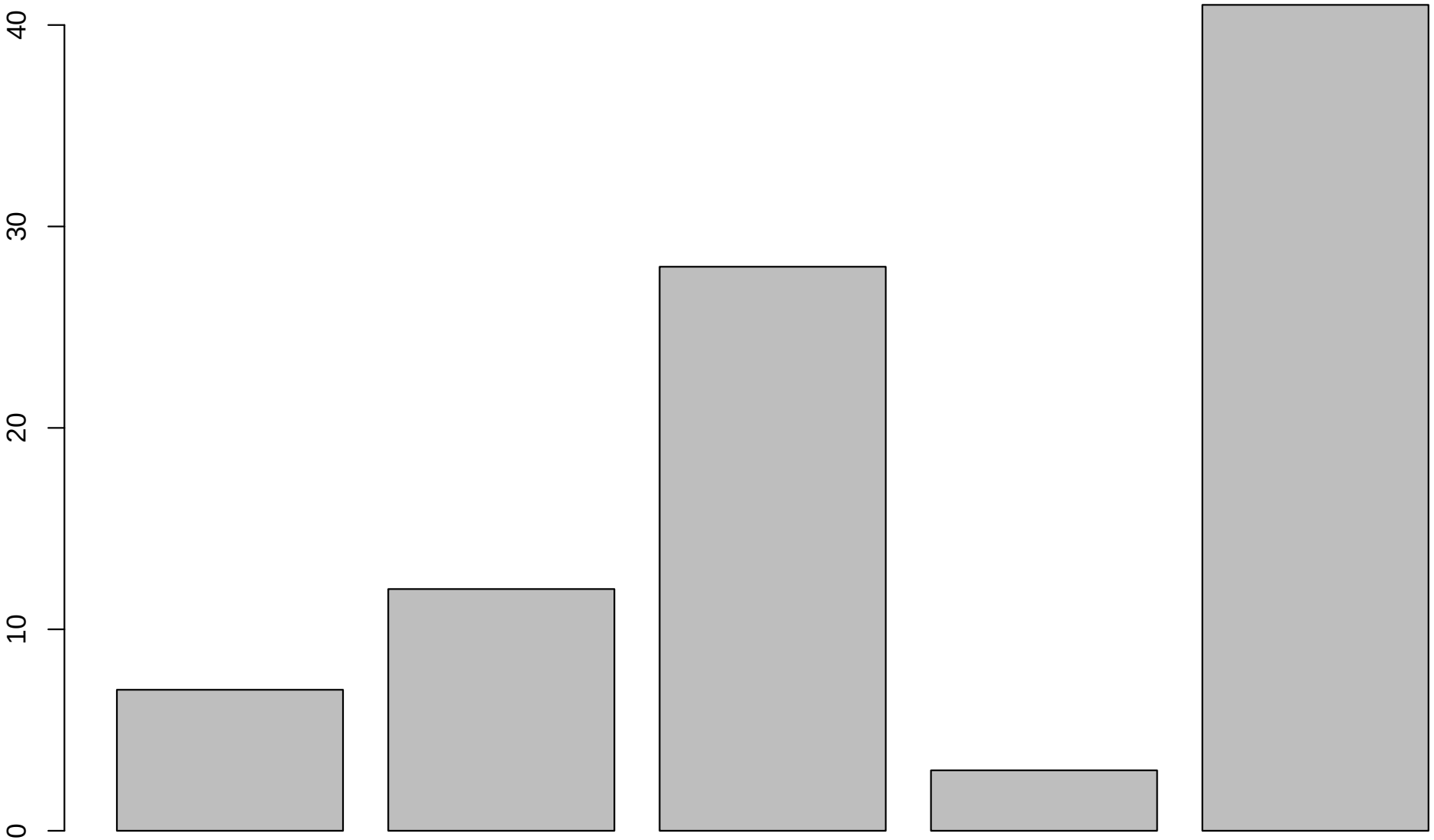
- **H** is a vector or matrix containing numeric values used in bar chart.
- **xlab** is the label for x axis.
- **ylab** is the label for y axis.
- **main** is the title of the bar chart.
- **names.arg** is a vector of names appearing under each bar.
- **col** is used to give colors to the bars in the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

Creating a Bar Chart I

```
H <- c(7,12,28,3,41)

barplot(H)
```

Creating a Bar Chart II

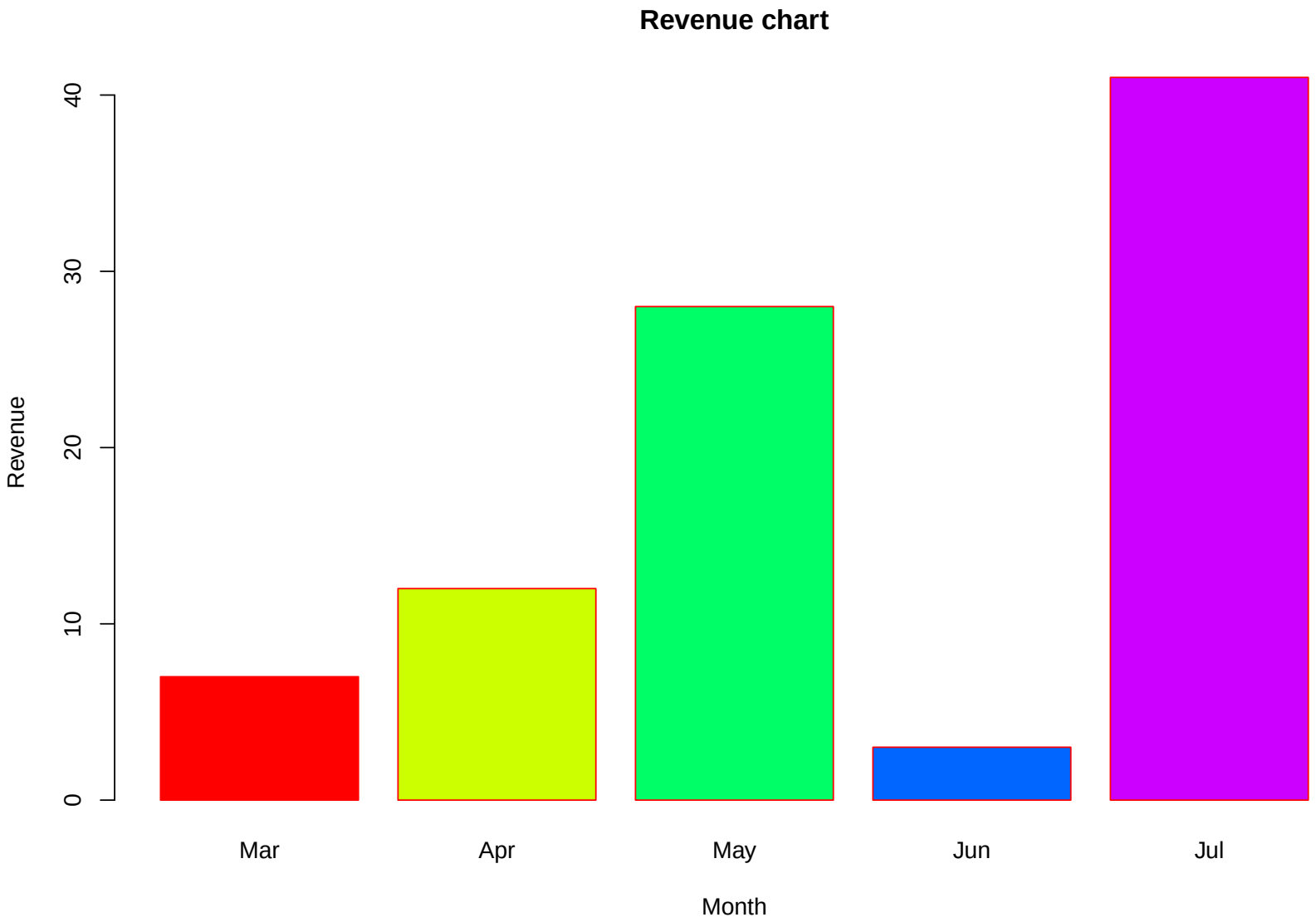


Bar Chart with Labels, Title and Colors I

```
H <- c(7,12,28,3,41)
M <- c("Mar","Apr","May","Jun","Jul")

barplot(H,names.arg = M,xlab = "Month",ylab = "Revenue",
        col = rainbow(length(H)), main = "Revenue chart",borde
```

Bar Chart with Labels, Title and Colors II



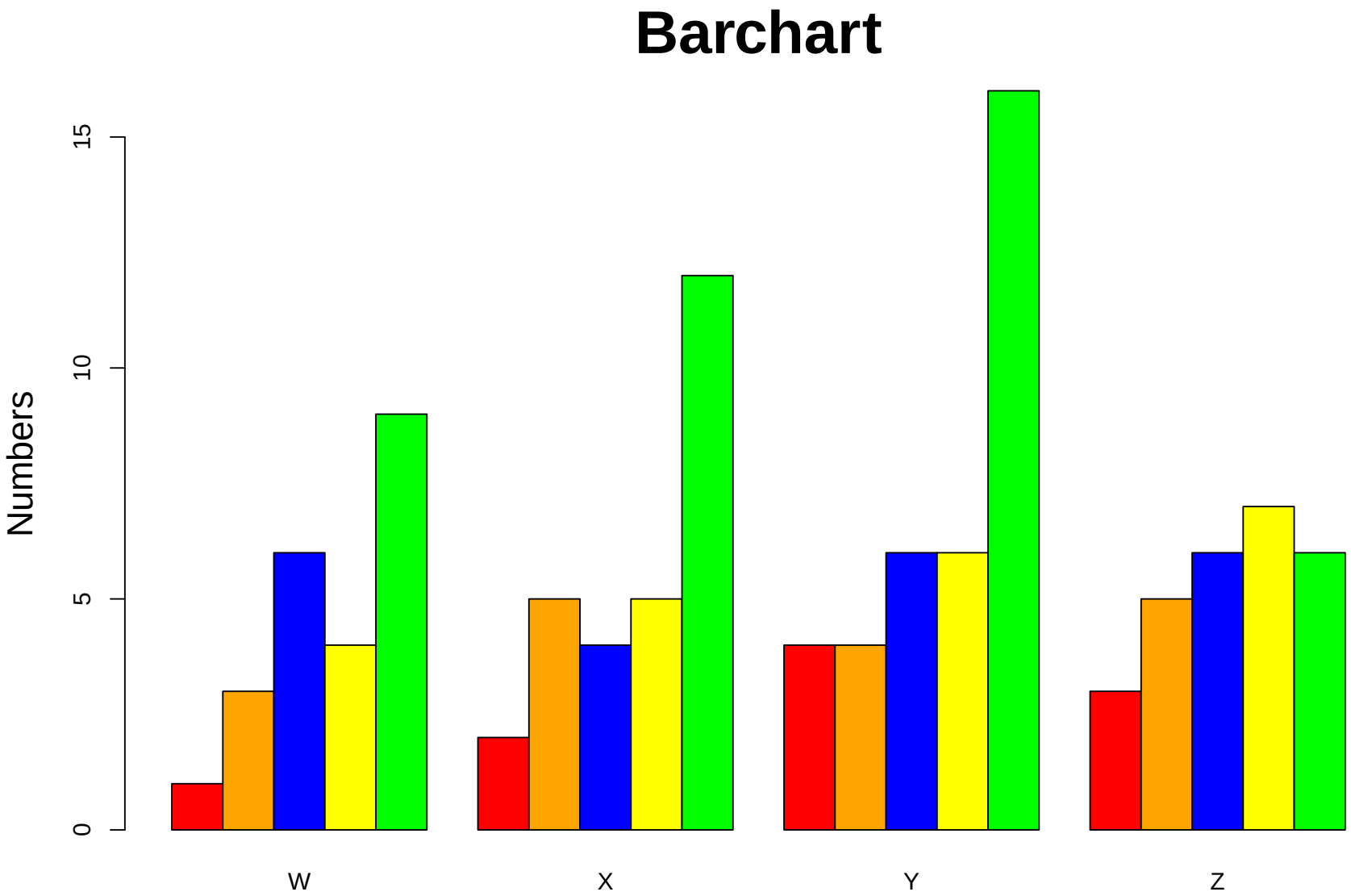
Bar Chart of a Dataset I

```
data <- structure(list(W= c(1L, 3L, 6L, 4L, 9L),
        X = c(2L, 5L, 4L, 5L, 12L), Y = c(4L, 4L, 6L, 6L, 16L)
        Z = c(3L, 5L, 6L, 7L, 6L)), Names = c("W", "X", "Y", "
        class = "data.frame", row.names = c(NA, -5L))
print(data)
```

```
##   W  X  Y Z
## 1 1  2  4 3
## 2 3  5  4 5
## 3 6  4  6 6
## 4 4  5  6 7
## 5 9 12 16 6
```

```
colours <- c("red", "orange", "blue", "yellow", "green")
barplot(as.matrix(data), main="Barchart", ylab = "Numbers",
      cex.lab = 1.5, cex.main = 2.5, beside=TRUE, col=colours)
```

Bar Chart of a Dataset II



Group Bar Chart and Stacked Bar Chart I

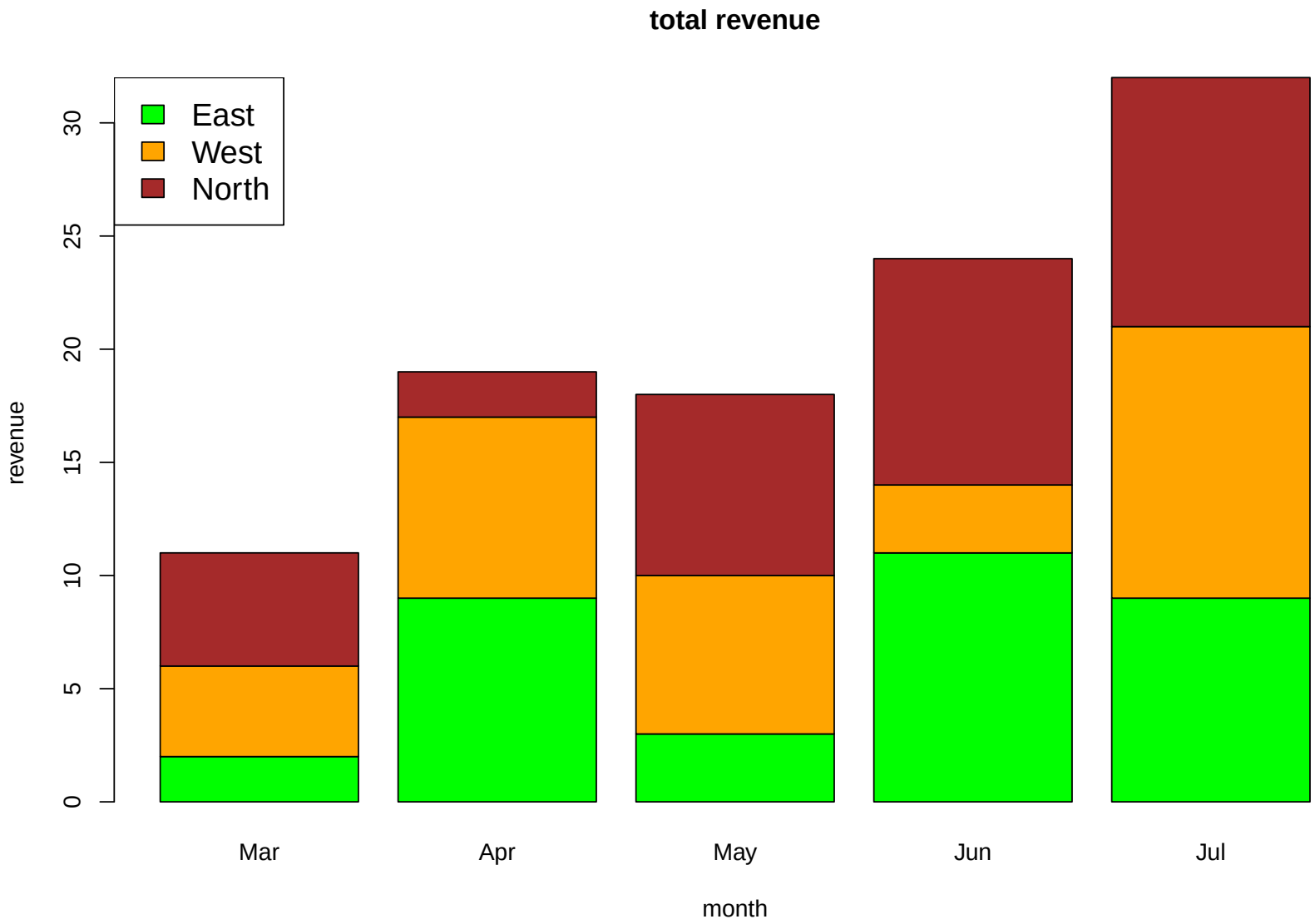
```
colors <- c("green","orange","brown")
months <- c("Mar","Apr","May","Jun","Jul")
regions <- c("East","West","North")

Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11),
                 nrow = 3,ncol = 5,byrow = TRUE)

barplot(Values,main = "total revenue",names.arg = months,
        xlab = "month",ylab = "revenue", col = colors)

legend("topleft", regions, cex = 1.3, fill = colors)
```

Group Bar Chart and Stacked Bar Chart II



Thank You