CM 2062 - Statistical Computing with R
Lab Sheet 4

Visualization of Data in R

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.

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.

Some of the various types of visualizations offered by R are: Pie Chart, Bar Chart, Line Chart, Histograms, Box Plot, Scatter Plot ... etc.

Visualization of Categorical Data in R

Basically Pie Chart and Bar Chart can be used to visualize categorical data.

Pie Chart in R

In R the pie chart is created using the **pie()** function which takes positive numbers as a vector input. The additional parameters are used to control labels, color, title etc.

```
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.

Example
Let's consider a survey was conducted of a group of 190 individuals, who were asked "What's your favorite fruit?"

Fruit:	Apple	Kiwi	Grapes	Banana	Pears	Orange
People:	40	15	30	50	20	35

```
# Create data for the graph.
x <- c(40, 15, 30, 50, 20, 35 )
labels <- c("apple","kiwi","grape", "banana", "pear", "orange")
pie(x,labels)

# Plot the chart with title and rainbow color pallet.
pie(x, labels, main = "Favorite fruit pie chart", col = rainbow(length(x)))

# Give colours manually
pie(x, labels, main = "Favorite fruit pie chart",
    col=c("red","orange","yellow","blue", "green", "purple"))

# Another way to give colors
pie(x, labels, main = "Favorite fruit pie chart",
   col=gray(seq(0.4, 1.0, length = 6)))
```

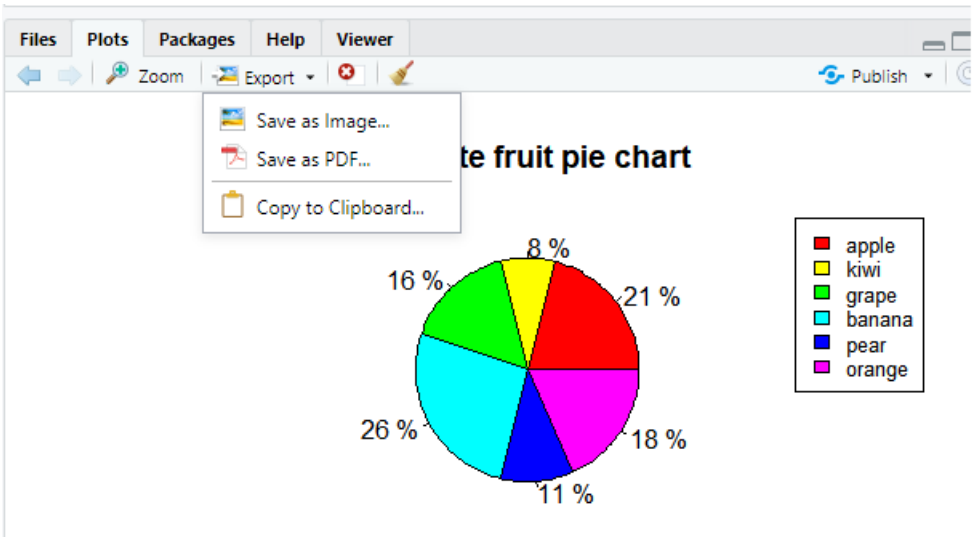
Exercise
Search for other ways to change colours in a Pie chart.

```
# Add slice Percentages and Chart Legend.
piepercent <- round(100*x/sum(x), 1)
piepercent <- round(100*x/sum(x))
# Add % sign
lbls <- paste(piepercent, "%")

pie(x, labels = lbls, main = "Favorite fruit pie chart",
    col = rainbow(length(x)))
legend("topright", c("apple","kiwi","grape", "banana", "pear", "orange"),
cex = 0.8, fill = rainbow(length(x)))
```

In R, you can save any plot toa pdf or as an image using "export" wizard.

Exercise:
Draw a 3D Pie Chart for the above example.



Bar Chart in R

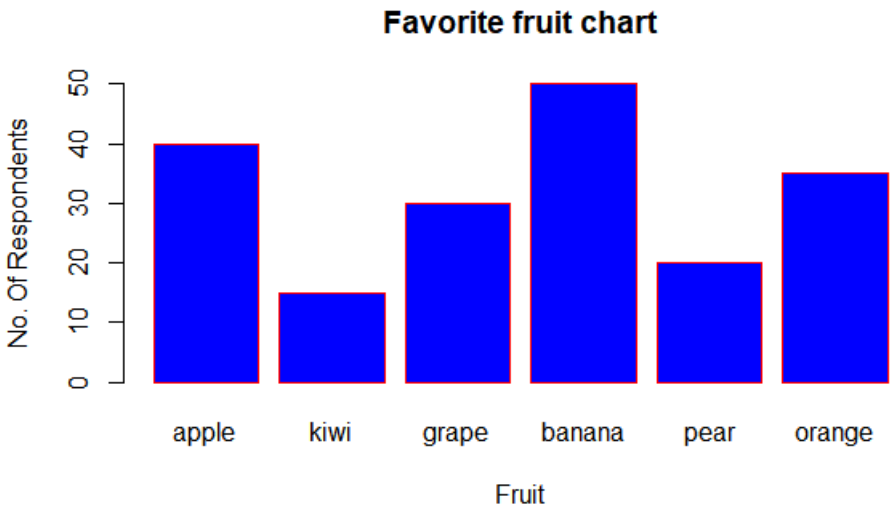
A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R uses the function **barplot()** to create bar charts. R can draw both vertical and Horizontal bars in the bar chart. In bar chart each of the bars can be given different colors.

The basic syntax to create a bar-chart in R is -

```
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.

```
# Create the data for the chart
x <- c(40, 15, 30, 50, 20, 35 )
labels <- c("apple","kiwi","grape", "banana", "pear", "orange")
# Plot the bar chart
barplot(x, names.arg=labels)
barplot(x, names.arg=labels, xlab="Fruit",ylab="No. Of Respondents",col="blue",
        main="Favorite fruit chart",border="red")
```



Exercise

Draw a horizontal bar chart for the above example.

Stacked Bar Chart

We can create bar chart with stacks in each bar by using a matrix as input values.

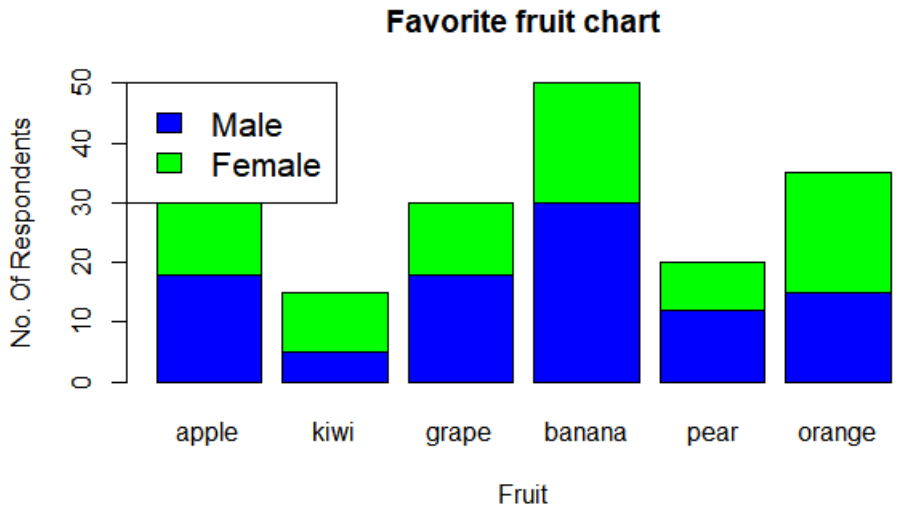
More than two variables are represented as a matrix which is used to create the group bar chart and stacked bar chart.

Let's consider the previous example and now assume there are both Male and Female respondents as given in below table.

	apple	kiwi	grape	banana	pear	orange
Male	18	5	18	30	12	15
Female	22	10	12	20	8	20

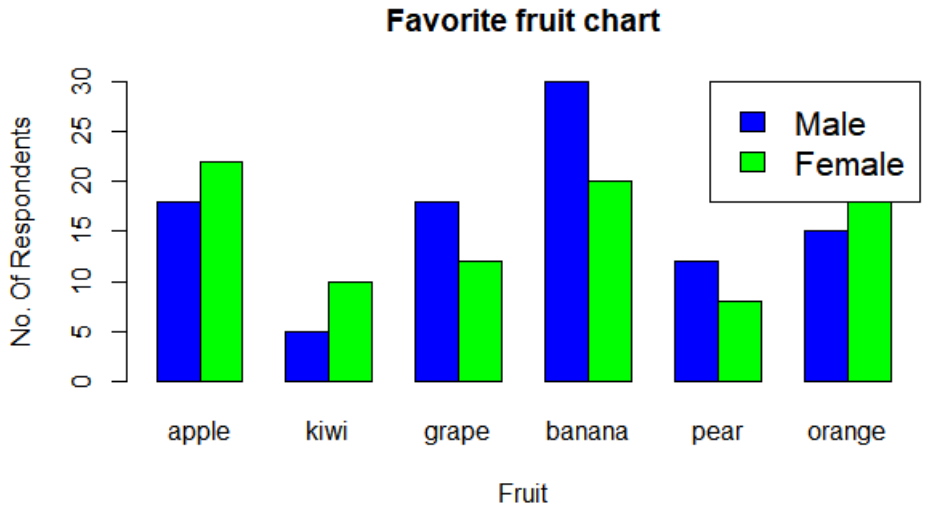
```
# Create the input vectors.
Gender <- c("Male", "Female")
Fruit <- c("apple","kiwi","grape", "banana", "pear", "orange")
# Create the matrix of the values.
Values <- matrix(c(18,5,18,30,12,15,22,10,12,20,8,20),
            nrow = 2, ncol = 6, byrow = TRUE)
# Give colors
colors = c("blue","green")
# Create the bar chart
barplot(Values, main = "Favorite fruit chart", names.arg = Fruit,
    xlab = "Fruit", ylab = "No. Of Respondents", col = colors)
# Add the legend to the chart
```

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



Group Bar Chart

```
barplot(Values, main = "Favorite fruit chart", names.arg = Fruit,
        xlab = "Fruit", ylab = "No. Of Respondents", col = colors, beside=TRUE)
legend("topright", Gender, cex = 1.3, fill = colors)
```



Visualization of Numerical Data in R

There are many charts in R to visualize numerical data. Among them we are discussing here only a few charts such as line chart, Histogram, box plot and scatter plot.

Line chart in R

A 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.

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

```
Temp <- c(30, 35, 40, 36, 31, 30 ,27, 42, 34, 25, 33, 36)
# Plot the line chart.
plot(Temp,type = "o")
```

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.

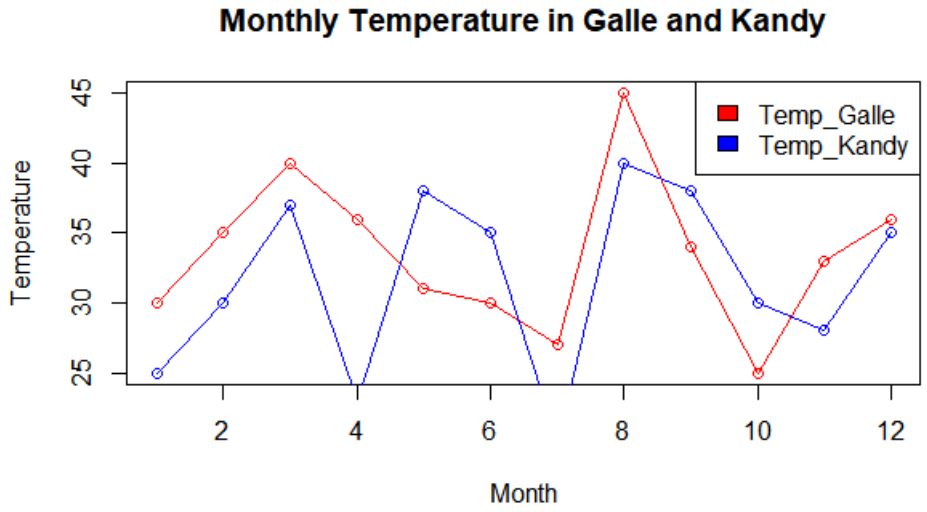
```
# Line Chart Title, Color and Labels
plot(Temp,type = "o", col = "red", xlab = "Month", ylab = "Temperature",
     main = "Monthly Temperature")
```

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.

```
# Multiple Lines in a Line Chart
Temp_Galle <- c(30, 35, 40, 36, 31, 30 ,27, 45, 34, 25, 33, 36)
Temp_Kandy <- c(25, 30, 37, 23, 38, 35, 20,  40, 38, 30, 28, 35)
```

```
plot(Temp_Galle,type = "o", col = "red", xlab = "Month", ylab = "Temperature",
     main = "Monthly Temperature in Galle and Kandy")
lines(Temp_Kandy, type = "o", col = "blue")
legend("topright", c("Temp_Galle", "Temp_Kandy"), fill=c("red", "blue"))
```



Exercise
Try to change the "type", line type, line width, line colour, legend position ... etc in the above line chart.

Histogram in R

R creates histogram using **hist()** function. This function takes a vector as an input and uses some more parameters to plot histograms.

The basic syntax for creating a histogram using R is -

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

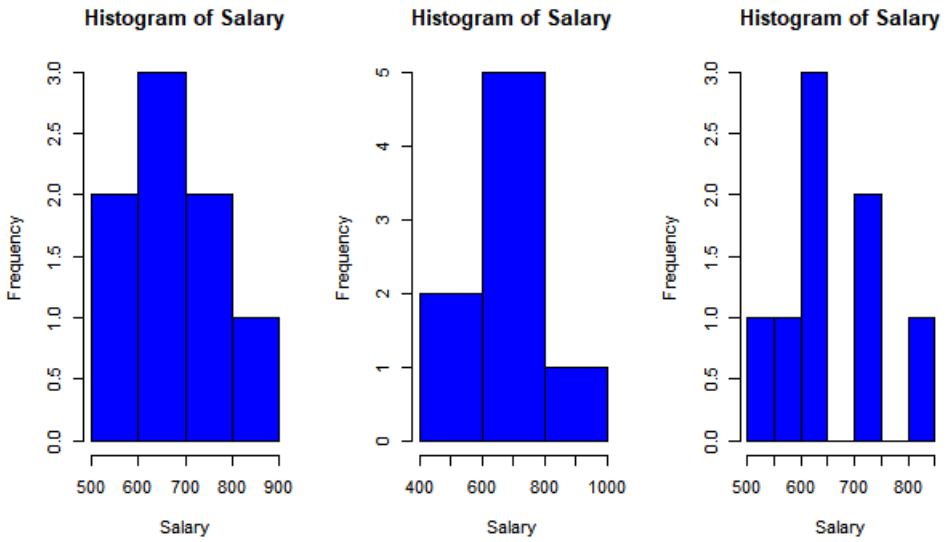
Following is the description of the parameters used -
v is a vector containing numeric values used in histogram.
main indicates title of the chart.
col is used to set color of the bars.
border is used to set border color of each bar.
xlab is used to give description of x-axis.
xlim is used to specify 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

Recall the "Salary" variable in "EmpFinalData" data frame.

```
Salary <- c(623.3, 515.2, 611, 729, 843.25, 578, 722.50, 632.80)
# Use to get more graphs in a same window
par(mfrow = c(1, 3))
# Create the histogram.
hist(Salary,xlab = "Salary",col = "blue",border = "black")
hist(Salary,xlab = "Salary",col = "blue",border = "black", breaks=2)
hist(Salary,xlab = "Salary",col = "blue",border = "black", breaks=5)
```



Box Plot in R

Box plots 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.

Box plots are created in R by using the **boxplot()** function.

The basic syntax to create a box plot in R is

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

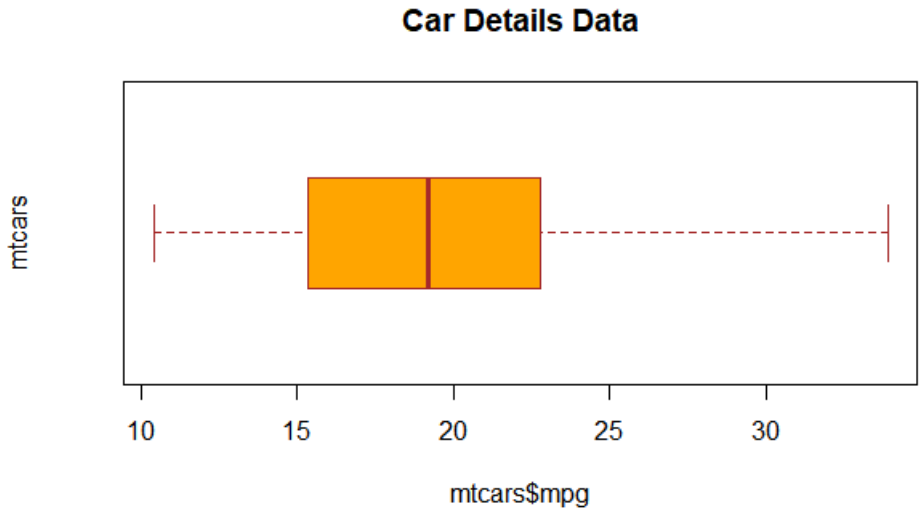
Following is the description of the parameters used-
x is a vector or a formula.

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 box plot.
main is used to give a title to the graph.

Example

We use the data set "mtcars" available in the R environment to create a basic box plot. Let's look at the columns "mpg" and "cyl" in mtcars.

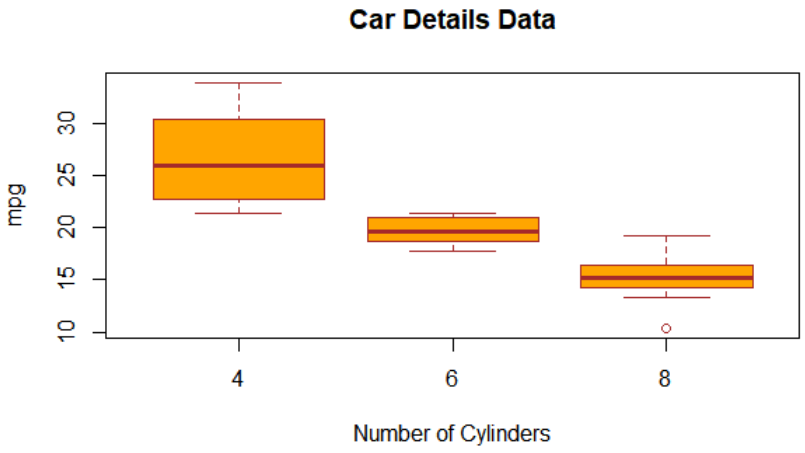
```
boxplot(mtcars$mpg)
boxplot(mtcars$mpg,
        main = "Car Details Data",
        xlab = "mtcars$mpg",
        ylab = "mtcars",
        col = "orange",
        border = "brown",
        horizontal = TRUE
)
```



The below script will create a box plot graph for the relation between mpg (miles per gallon) and cyl (number of cylinders).

```
boxplot(mpg ~ cyl,
        data=mtcars,
        main="Car Details Data",
        xlab="Number of Cylinders",
        ylab="mpg",
        col="orange",
        border="brown"
```

)



Exercise
Recall the data frame call "EmpFinalData" and draw the box plot for Salary variable.

Scatter Plot in R

Scatter plots 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.

The simple scatter plot is created using the **plot()** function.

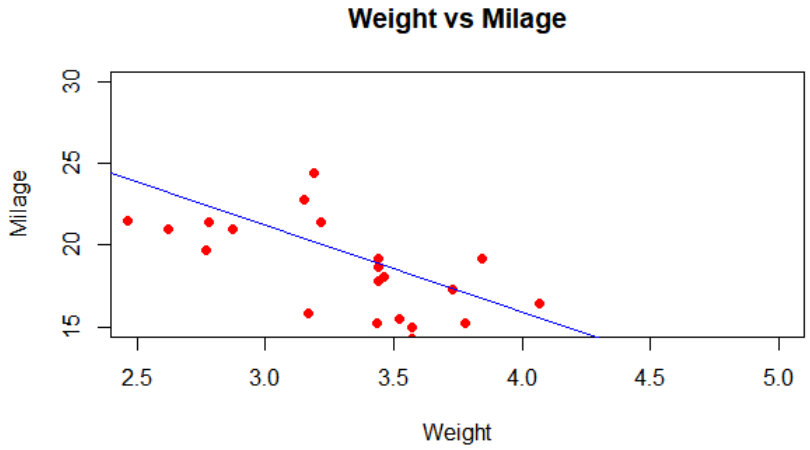
The basic syntax for creating scatter plot in R is-

```
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
We use the data set "mtcars" available in the R environment to create a basic scatter plot. Let's use the columns "wt" and "mpg" in mtcars.

```
mtcars
plot(mtcars$wt,mtcars$mpg,
     xlab = "Weight",
     ylab = "Milage",
     main = "Weight vs Milage"
)
# Plot the chart for cars with weight between 2.5 to 5 and
# mileage between 15 and 30.
plot(mtcars$wt,mtcars$mpg,
     xlab = "Weight",
     ylab = "Milage",
     xlim = c(2.5,5),
     ylim = c(15,30),
     main = "Weight vs Milage",
     pch = 19, # Change the pattern use for the data points
     col = "red"
)
# Add regression line
abline(lm(mtcars$mpg ~ mtcars$wt, data = mtcars), col = "blue")
```



According to the scatter plot there is a negative liner relationship between weight and millage.

Scatter plot Matrices

When we have more than two variables and we want to find the correlation between one variable versus the remaining ones we use scatter plot matrix. We use pairs() function to create matrices of scatter plots. Each variable is paired up with each of the remaining variable. A scatter plot is plotted for each pair.

The basic syntax for creating scatter plot matrices in R is-

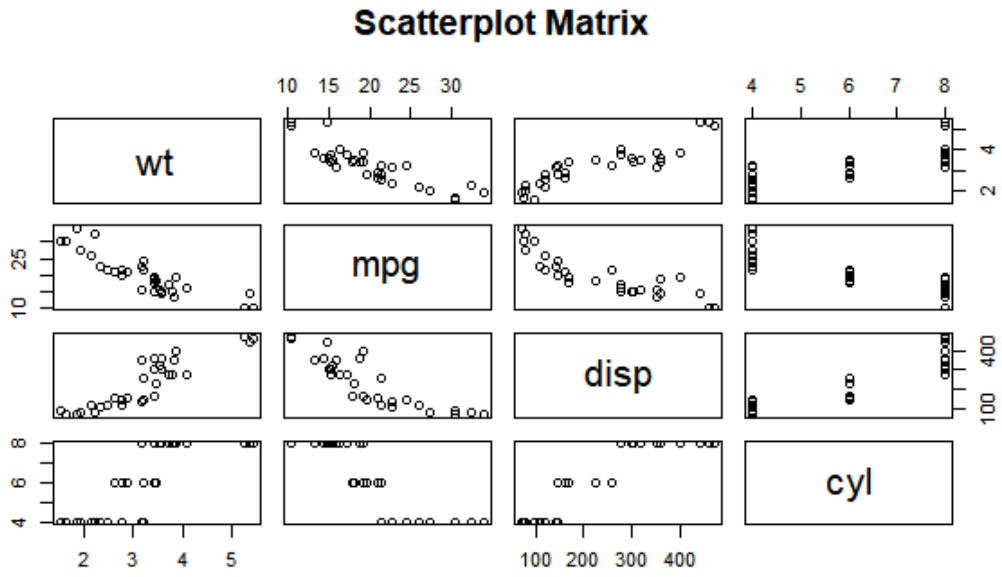
```
pairs(formula, data)
```

Following is the description of the parameters used-
formula represents the series of variables used in pairs.
data represents the data set from which the variables will be taken.

Example

```
# Plot the matrices between 4 variables giving 12 plots.
# One variable with 3 others and total 4 variables.

pairs(~wt+mpg+disp+cyl,data = mtcars,
      main = "Scatterplot Matrix")
```



Exercise

Recall the "Students" data frame in R.
(a) Draw the scatter plot between student's weight and height and interpret it.
(b) Draw the scatter plot between student's weight and age and interpret it.
(c) Draw the scatter plot between student's height and age and interpret it.
(d) Draw the scatter plot Matrix for student's weight. height and age.

Note: You can find out about the full capabilities of R´s graphics system by typing,

demo(graphics)

which will create several built-in demo graphs.