2_3_Lab

xiyuan liu

```
### Chapter 2 Lab: Introduction to R

## 2.3.1 Basic Commands ---------------------------------------------
---------

# ------------------------------------------------------------------
---------
# Example. Joint together the numbers 1, 3, 2, and 5, and to save thme
as
#          a vector named x.
# ------------------------------------------------------------------
---------

x <- c(1, 3, 2, 5)
x

## [1] 1 3 2 5

# We can also save things "=" rather than "<-"
x = c(1, 6, 2)
x

## [1] 1 6 2

y = c(1, 4, 3)

# Type "?" to open a new help file window
?c

## starting httpd help server ... done

# ------------------------------------------------------------------
---------
# Example. Add two sets of numbers togeter.
# ------------------------------------------------------------------
---------

# Check their length using the "length()"
length(x)

## [1] 3

length(y)

## [1] 3
```

```r
# Add two sets of numbers together. However, x and y should be the same
length.
x+y
```

```
## [1]  2 10  5
```

```r
# ---------------------------------------------------------------------
---------
# Example. Delete both x and y
# ---------------------------------------------------------------------
---------

# Look at a list of the objects, show as data and functions.
ls()
```

```
## [1] "x" "y"
```

```r
# Delete both x and y
rm(x, y)

# Remove all objects at once
ls()
```

```
## character(0)
```

```r
rm(list=ls())



# ---------------------------------------------------------------------
---------
# Example. Create a simple matrix
# ---------------------------------------------------------------------
---------

?matrix
x = matrix(data=c(1, 2, 3, 4), nrow=2, ncol=2)
x
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```r
x = matrix(c(1, 2, 3, 4), 2, 2)
matrix(c(1, 2, 3, 4), 2, 2, byrow=TRUE)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

```r
# ------------------------------------------------------------------
---------
# Example. Take the square root of x and square of x
# ------------------------------------------------------------------
---------

sqrt(x)
```

```
##          [,1]     [,2]
## [1,] 1.000000 1.732051
## [2,] 1.414214 2.000000
```

```r
x^2
```

```
##      [,1] [,2]
## [1,]    1    9
## [2,]    4   16
```

```r
# ------------------------------------------------------------------
---------
# Example. Compute the correlation between x and y
# ------------------------------------------------------------------
---------

# Randomly genertate 50 obervations in a normal distribution.
x = rnorm(50)  # x ~ N(0, 1^2)
y = x + rnorm(50, mean=50, sd=.1)  # y = x + e, where e ~ N(50, 0.1^2)

# compute the correlation between x and y
cor(x, y)
```

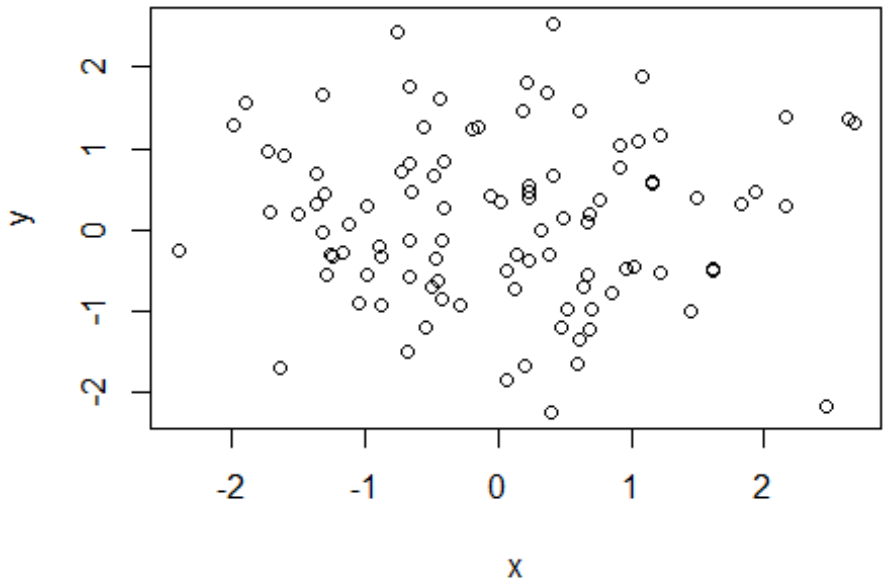
```
## [1] 0.9952733
```

```r
# ------------------------------------------------------------------
---------
# Example. Reproduce the exact same set of random numbers
# ------------------------------------------------------------------
---------

set.seed(1303)
rnorm(50)
```

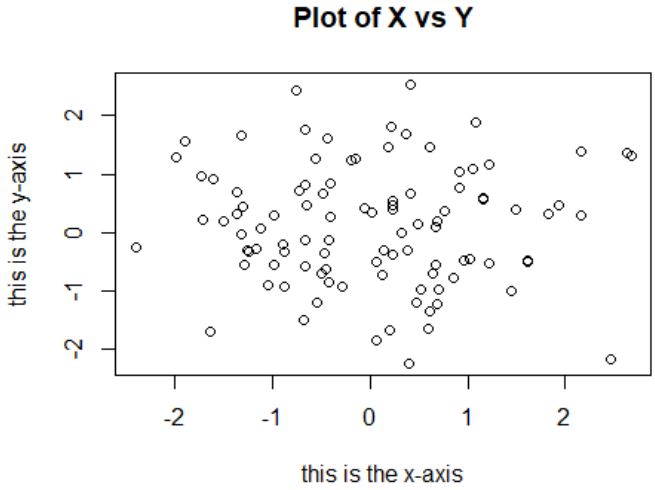
```
##  [1] -1.1439763145  1.3421293656  2.1853904757  0.5363925179
0.0631929665
##  [6]  0.5022344825 -0.0004167247  0.5658198405 -0.5725226890 -
1.1102250073
## [11] -0.0486871234 -0.6956562176  0.8289174803  0.2066528551 -
0.2356745091
## [16] -0.5563104914 -0.3647543571  0.8623550343 -0.6307715354
0.3136021252
## [21] -0.9314953177  0.8238676185  0.5233707021  0.7069214120
```

```
0.4202043256
## [26] -0.2690521547 -1.5103172999 -0.6902124766 -0.1434719524 -
1.0135274099
## [31]  1.5732737361  0.0127465055  0.8726470499  0.4220661905 -
0.0188157917
## [36]  2.6157489689 -0.6931401748 -0.2663217810 -0.7206364412
1.3677342065
## [41]  0.2640073322  0.6321868074 -1.3306509858  0.0268888182
1.0406363208
## [46]  1.3120237985 -0.0300020767 -0.2500257125  0.0234144857
1.6598706557

# ------------------------------------------------------------------
---------
# Example. Find the mean, the variance and the standard deviation for y
# ------------------------------------------------------------------
---------

set.seed(3)
y = rnorm(100)

# Compute the mean of y
mean(y)

## [1] 0.01103557

# Compute the variance of y
var(y)

## [1] 0.7328675

# Compute the standard deviation of y
sqrt(var(y))

## [1] 0.8560768

sd(y)

## [1] 0.8560768

## 2.3.2 Graphics ---------------------------------------------------
---------

# ------------------------------------------------------------------
---------
# Example. Draw a scatterplot using x and y
# ------------------------------------------------------------------
---------

x = rnorm(100)  # x ~ N(0, 1)
y = rnorm(100)  # y ~ N(0, 1)
```

```r
# Draw a simple scatterplot using x and y
plot(x, y)
```




```r
# Draw a scatterplot, and specify the x-axis and the y-axis label.
# Also, specify the main title for the plot.
plot(x, y, xlab="this is the x-axis",
           ylab="this is the y-axis",
           main="Plot of X vs Y")
```



```r
# ------------------------------------------------------------------------
---------
# Example. Export the plot to a PDF file
# ------------------------------------------------------------------------
---------

# Create a pdf file named Figure.pdf
pdf("Figure.pdf")  # You can use jpeg() to create a jpeg file.
plot(x, y, col="green")  # Specify the dot color as green.
dev.off()
```

```
## png
##   2
```

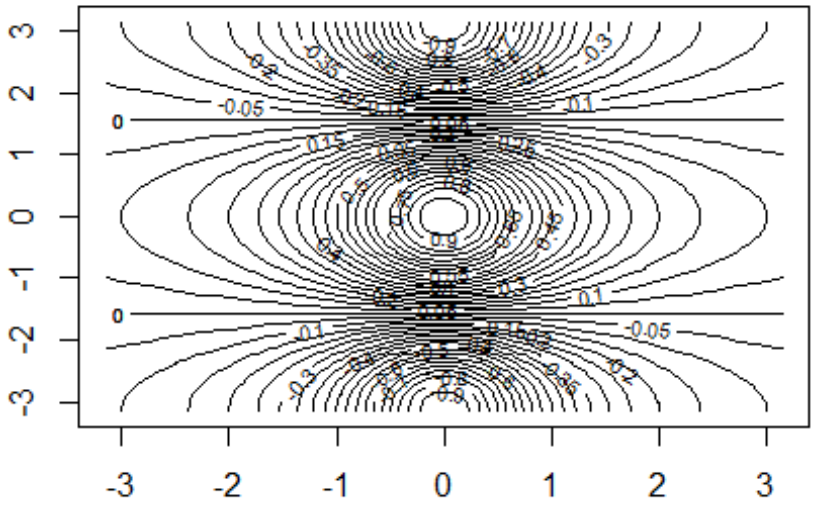
```r
# ------------------------------------------------------------------------
---------
# Example. Create a sequence of numbers
# ------------------------------------------------------------------------
---------

x = seq(1, 10)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

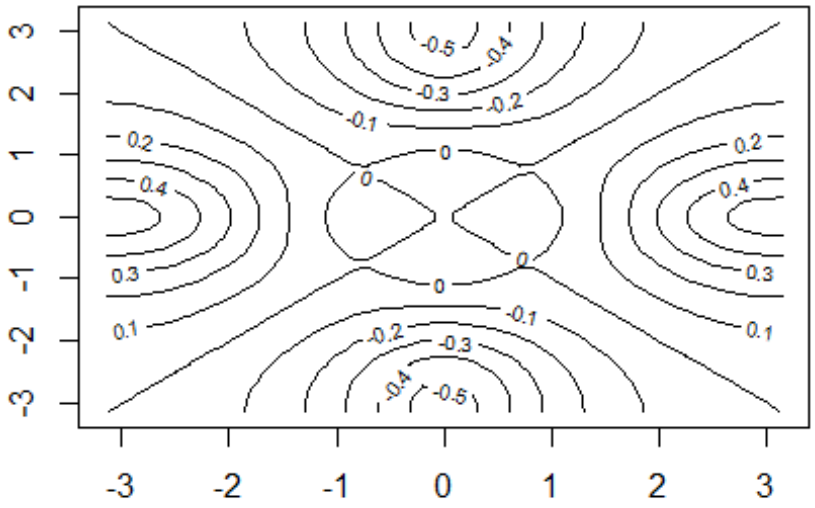
```r
x = 1:10  # This is a short cut for seq(1, 10)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

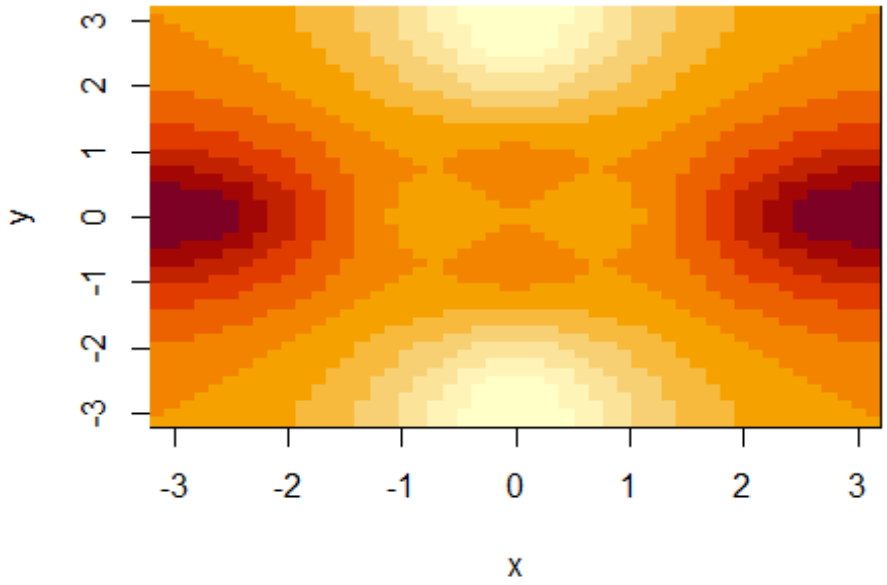
```r
# Create a sequence with a length of 50 between -3.14 and 3.14
x = seq(-pi, pi, length=50)


# ------------------------------------------------------------------------
---------
# Example. Create a contour plot
# ------------------------------------------------------------------------
---------

y = x
f = outer(x, y, function(x,y)cos(y)/(1+x^2))
contour(x, y, f)
contour(x, y, f, nlevels=45, add=T)
```



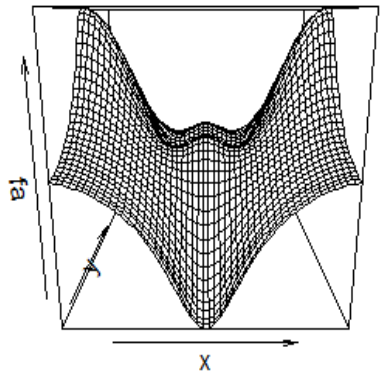
```
fa = (f-t(f))/2
contour(x, y, fa, nlevels=15)
```



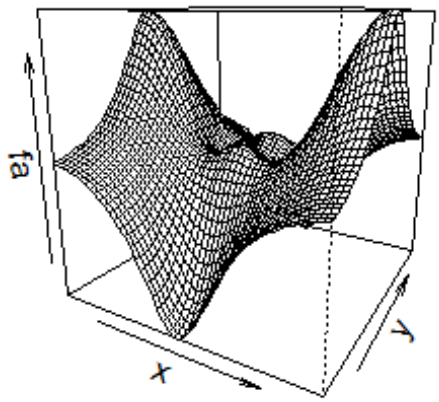
```r
# The image() works the same way as contour(), expect that it produces
# a color-coded plot whose colors depend on the z value.
image(x, y, fa)  # This is known as a heatmap
```



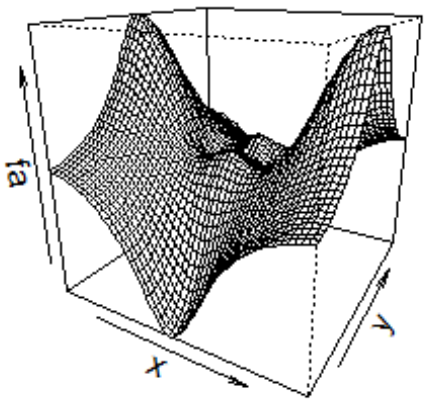

```r
# ---------------------------------------------------------------------
---------
# Example. Produce a three-dimensional plot. The arguments theta and
phi
#          control the angles at which the plot is viewed.
# ---------------------------------------------------------------------
---------

persp(x, y, fa)
```



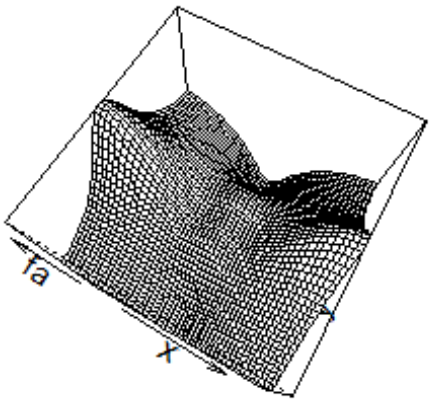
```r
persp(x, y, fa, theta=30)
```



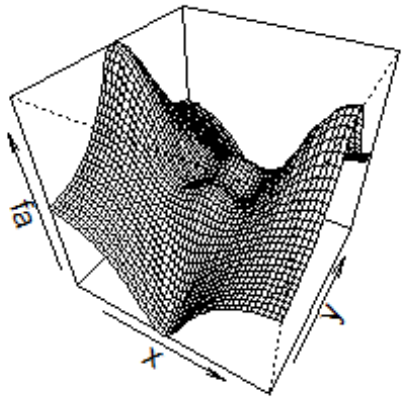

```r
persp(x, y, fa, theta=30, phi=20)
```




```r
persp(x, y, fa, theta=30, phi=70)
```



```r
persp(x, y, fa, theta=30, phi=40)
```




```r
## 2.3.3 Indexing Data ------------------------------------------------
---------

# ------------------------------------------------------------------------
---------
# Example. Examine part of a set of data.
# ------------------------------------------------------------------------
---------

A = matrix(1:16, 4, 4)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```r
# We want to check the element corresponding to
# the 2nd row and the 3rd column.
A[2,3]
```

```
## [1] 10
```

```r
# Check the element corresponding to the combination of
# the 1st, the 3rd rows, and the 2nd, the 4th columns.
A[c(1,3),c(2,4)]
```

```
##      [,1] [,2]
## [1,]    5   13
## [2,]    7   15
```

```r
# More options
A[1:3, 2:4]
```

```
##      [,1] [,2] [,3]
## [1,]    5    9   13
## [2,]    6   10   14
## [3,]    7   11   15
```

```
A[1:2, ]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
```

```
A[, 1:2]
```

```
##      [,1] [,2]
## [1,]    1    5
## [2,]    2    6
## [3,]    3    7
## [4,]    4    8
```

```
A[1, ]
```

```
## [1]  1  5  9 13
```

```
# Check all elements except the 1st row and the 3rd row.
A[-c(1,3), ]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    6   10   14
## [2,]    4    8   12   16
```

```
# More example.
A[-c(1,3), -c(1,3,4)]
```

```
## [1] 6 8
```

```
# Check the dimension of the matrix A.
dim(A)
```

```
## [1] 4 4
```

```
## Loading Data ------------------------------------------------------
---------

# ------------------------------------------------------------------
---------
# Example. Load the data in your computer
# ------------------------------------------------------------------
---------

# Let R search your data under your data file
setwd("C:\\Users\\gdsdl\\Desktop\\Lab_2_3")
```

```r
# Read the table using read.table()
Auto = read.table("Auto.data")  # The data will be stored as a data
frame.

# Use the fix() function to view your data in a spreadsheet like window
fix(Auto)

# The options: header and na.strings.
# header: Use the first line of the file as the variable names.
# na.string: Treat the "?" as a missing element of the data matrix.
Auto = read.table("Auto.data", header=T, na.strings="?")
fix(Auto)

# ---------------------------------------------------------------------
---------
# Example. Load a csv file (comma separated value) file
# ---------------------------------------------------------------------
---------

Auto = read.csv("Auto.csv", header=T, na.strings="?")
fix(Auto)
dim(Auto)
```

```
## [1] 397   9
```

```r
Auto[1:4, ]
```

```
##   mpg cylinders displacement horsepower weight acceleration year
origin
## 1  18         8          307        130   3504         12.0   70
1
## 2  15         8          350        165   3693         11.5   70
1
## 3  18         8          318        150   3436         11.0   70
1
## 4  16         8          304        150   3433         12.0   70
1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
```

```r
# Remove the rows (the observations) with any missing variables
Auto = na.omit(Auto)
dim(Auto)
```

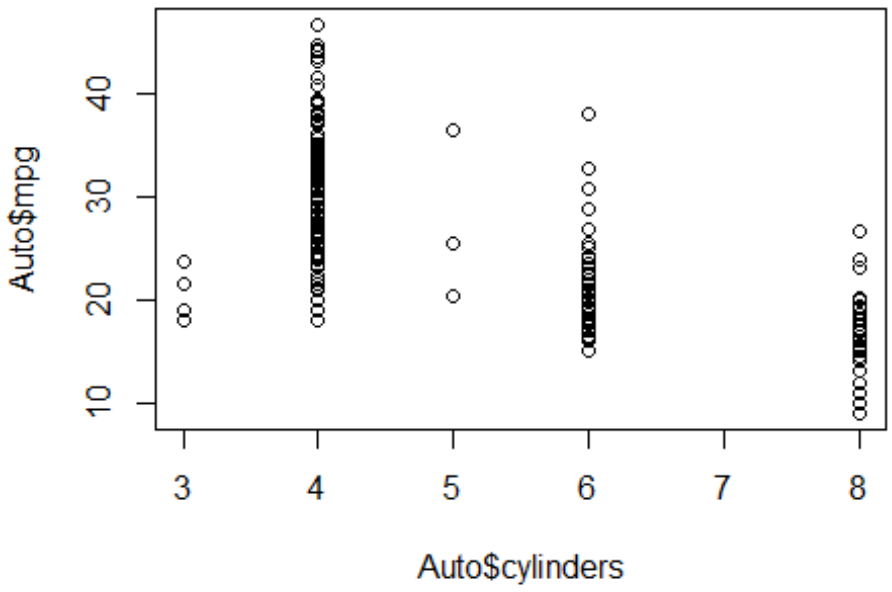
```
## [1] 392   9
```

```r
fix(Auto)
```

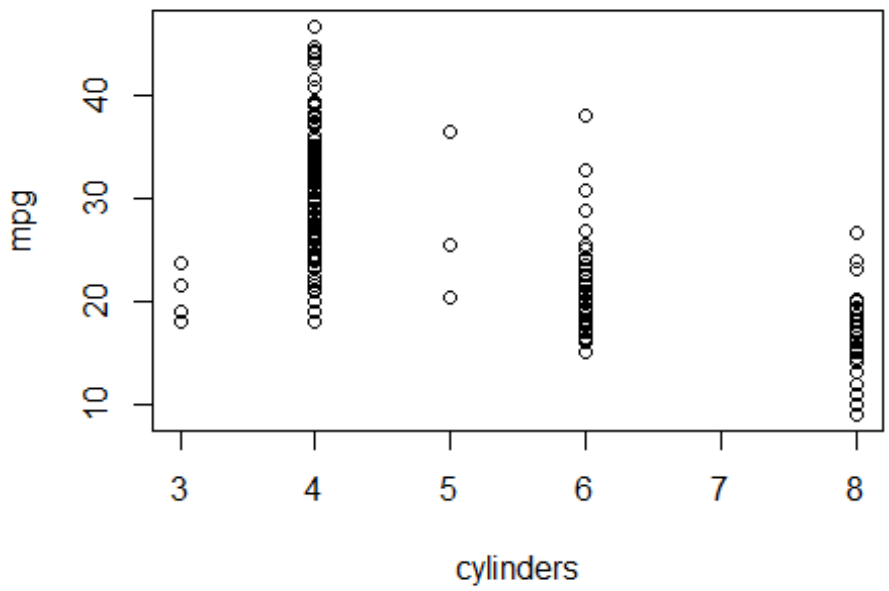
```r
# Check the variable names in our data
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"
"weight"
## [6] "acceleration" "year"         "origin"       "name"
```

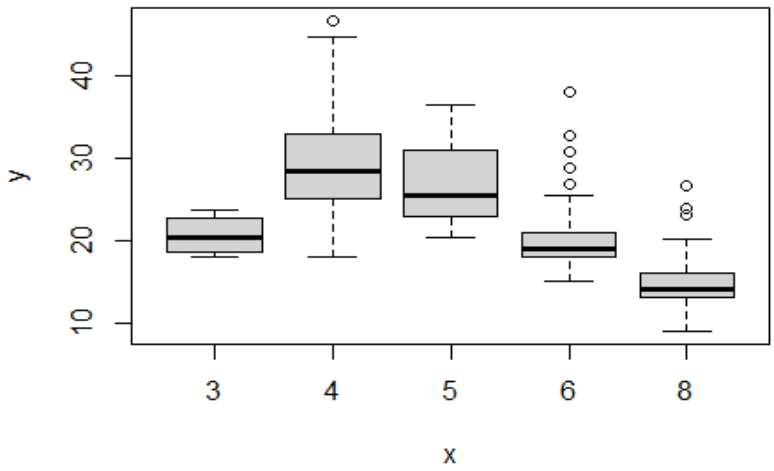
```r
## Additional Graphical and Numerical Summaries ----------------------
---------

# ------------------------------------------------------------------------
---------
# Example. Plot the cylinders and the mpg variables from the Auto data
# ------------------------------------------------------------------------
---------

# plot(cylinders, mpg)  # Obiviously, this cannot work
plot(Auto$cylinders, Auto$mpg)  # This is a regular scatterplot
```



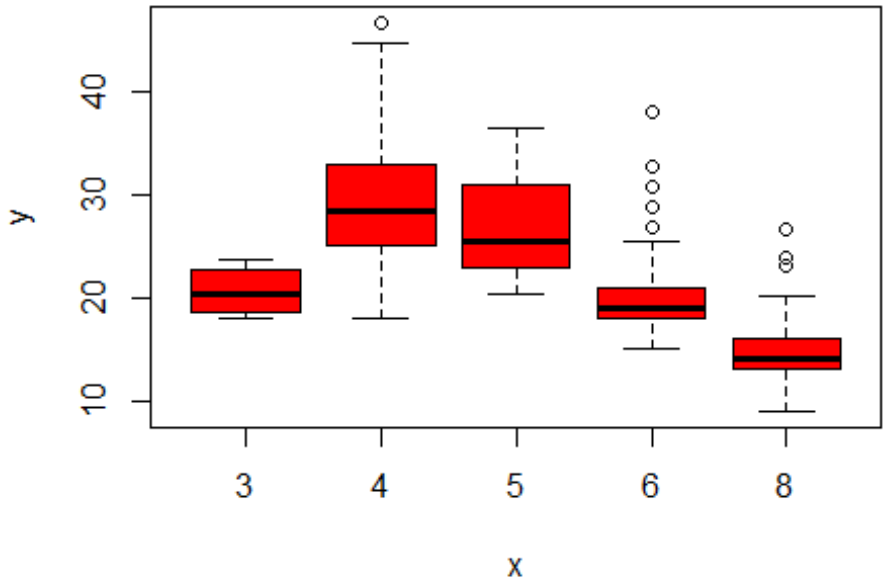

```r
# We can use attach() to make the variable in the data frame (Auto)
avaliable
# by name.
attach(Auto)
plot(cylinders, mpg)
```



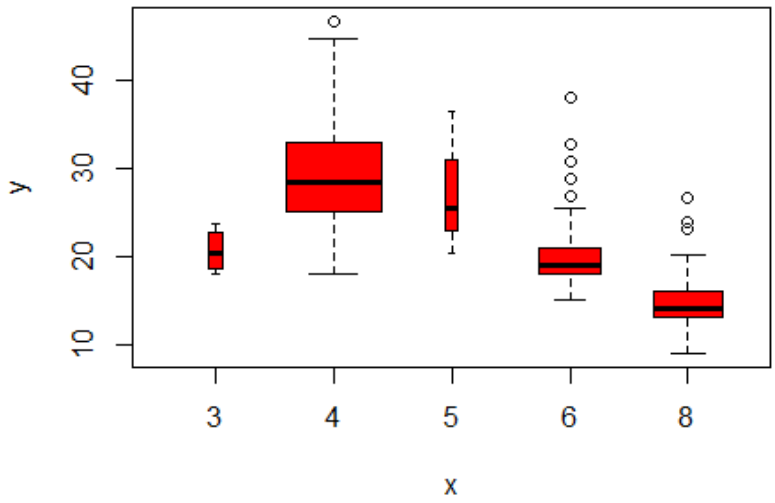

```r
# We want to treat cylinder as a categorical variable.
# (In R, its called factor.)
cylinders = as.factor(cylinders)

# Now, since the x-axis is categorial, the plot is changed into a
boxplot.
plot(cylinders, mpg)
```



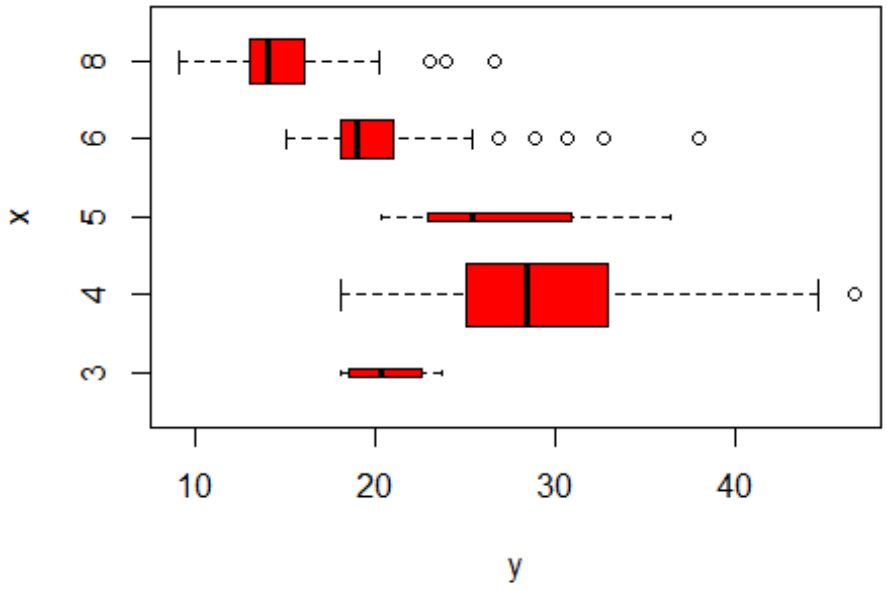
```r
plot(cylinders, mpg, col="red")  # Change the color.
```



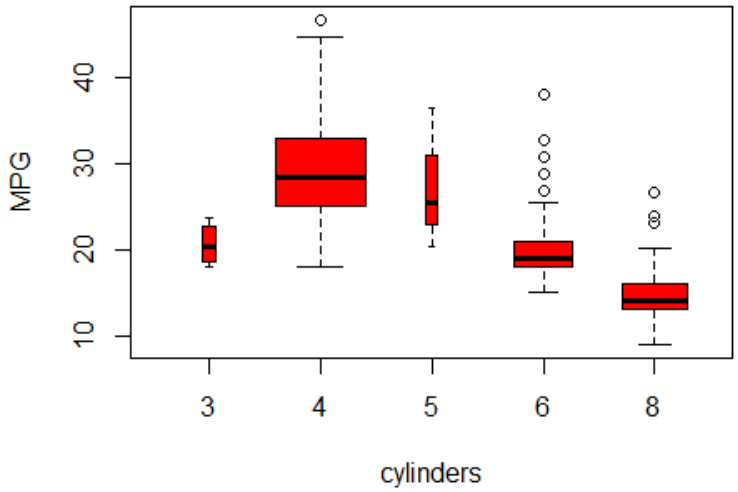

```r
# "varwidth=T": The box width will be proportional to the  square root of the
#               number of observations in the data.
plot(cylinders, mpg, col="red", varwidth=T)
```



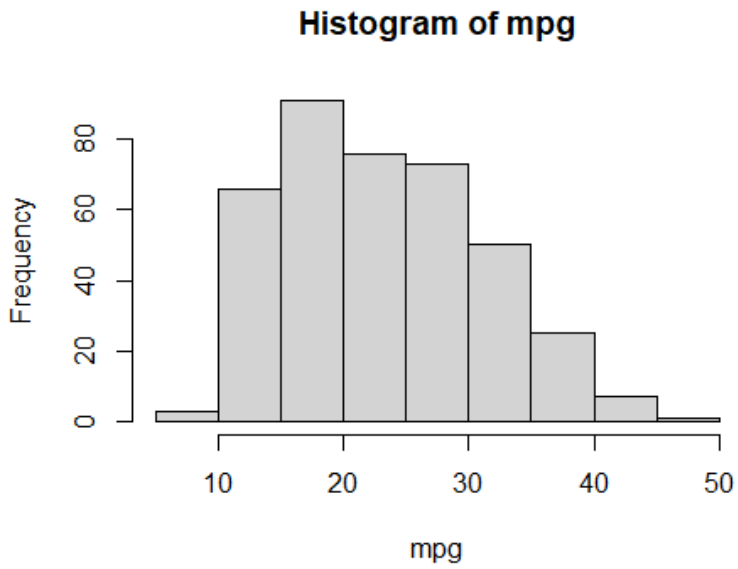
```r
# "horizontal=T": Creates horizontal boxes
plot(cylinders, mpg, col="red", varwidth=T, horizontal=T)
```



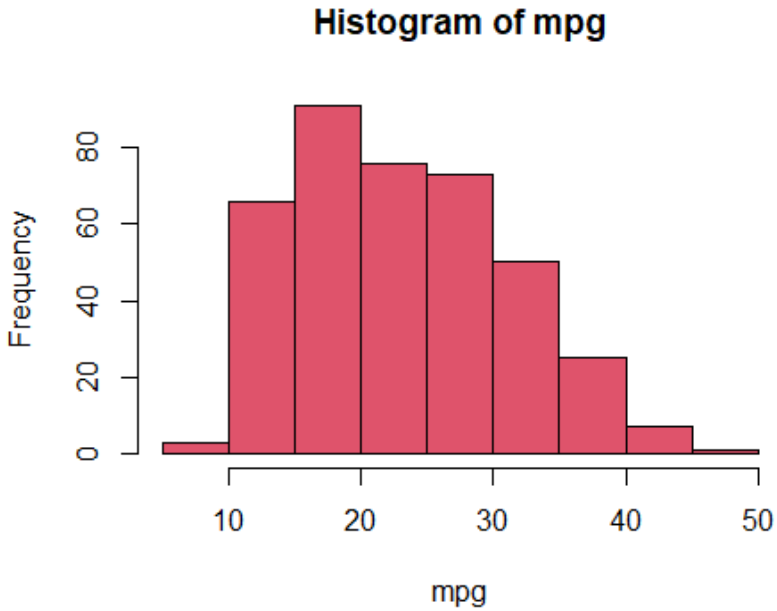

```r
# Create text labels for x-axis and y-axis
plot(cylinders, mpg, col="red", varwidth=T, xlab="cylinders",
ylab="MPG")
```



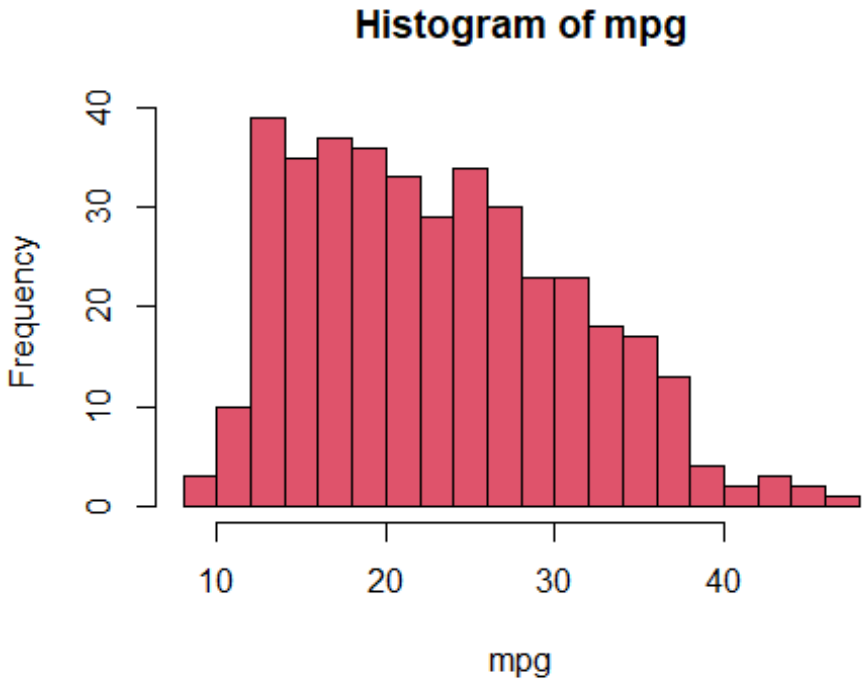
```r
# ---------------------------------------------------------------------
---------
# Example. Plot a histogram using the mpg variable
# ---------------------------------------------------------------------
---------

hist(mpg)
```



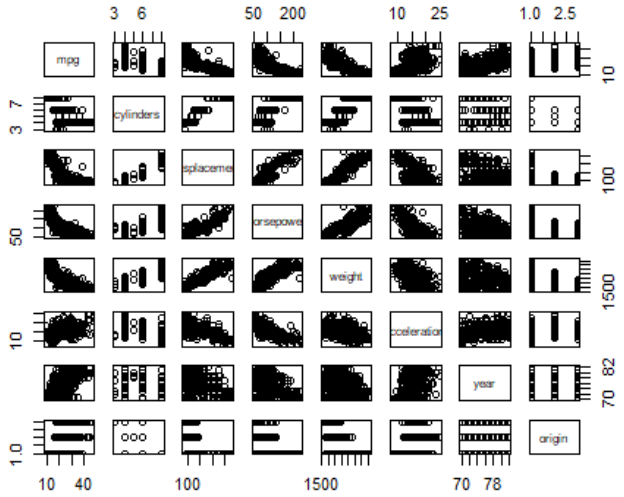

```r
hist(mpg, col=2)  # Change the color into red.
```



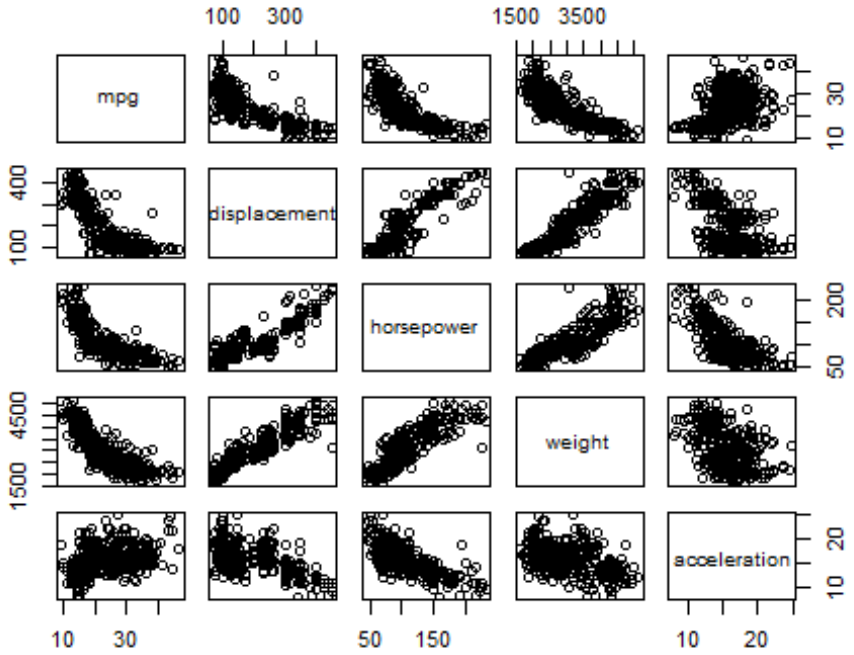
```r
hist(mpg, col=2, breaks=15)
```



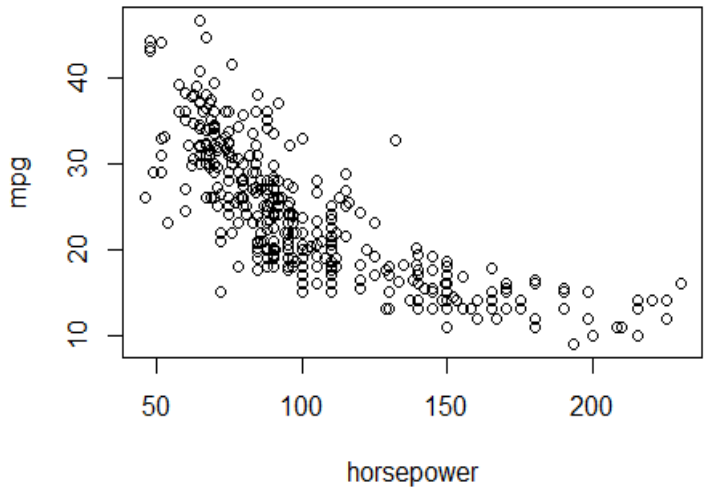

```r
# ---------------------------------------------------------------------
---------
# Example. Use pair() to create a scatterplot matrix
# ---------------------------------------------------------------------
---------

pairs(Auto[, -9])  # We can only use non-character variables.
```



```r
# We can also produce scatterplots for just a subset of the variables
pairs(~ mpg + displacement + horsepower + weight + acceleration, Auto)
```




```r
# ------------------------------------------------------------------
---------
# Example. identify() provides a useful interactive method for
identifying
#          the value for a particular variable for points on a plot.
# ------------------------------------------------------------------
---------

plot(horsepower, mpg)

# Three inputs for the identify function:
#   x: The x-axis.
#   y: The y-axis.
#   labels: the variable whose values we would like to see printed
#           for each point.
identify(x = horsepower, y = mpg, labels = name)
```




```
## integer(0)

# ------------------------------------------------------------------------
---------
# Example. Use summary() function to produce a numerical summary for
each
#          variable in the Auto data
# ------------------------------------------------------------------------
---------

summary(Auto)

##       mpg          cylinders      displacement     horsepower
weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
Min.   :1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st
Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
Median :2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
Mean   :2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd
Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
Max.   :5140
##   acceleration        year           origin          name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   Length:392
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   Class :character
##  Median :15.50   Median :76.00   Median :1.000   Mode  :character
##  Mean   :15.54   Mean   :75.98   Mean   :1.577
```

```
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000

# Can also produce a summary of just a single variable.
summary(mpg)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   17.00   22.75   23.45   29.00   46.60
```