Chapter 02. Introduction To R

Feathers

1. Good at data analysis

2. It is totally free

3. Lots of built-in packages

4. Beautiful graphics

---

00. Load necessary packages

```
# install.packages('ISLR')
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.1.2
```

---

01. Basic command

```
# vectors, data, matrices, subsetting
x=c(2,7,5)
x
```

```
## [1] 2 7 5
```

```
y=seq(from=4, length=3, by=3)
?seq
```

```
## starting httpd help server ... done
```

```
y
```

```
## [1]  4  7 10
```

```
# vector operations in parallel
x+y
```

```
## [1]  6 14 15
```

```
# element-wise division
x/y
```

```
## [1] 0.5 1.0 0.5
```

```
x^y
```

```
## [1]      16  823543 9765625
```

```
x[2]
```

```
## [1] 7
```

```
x[2:3]
```

```
## [1] 7 5
```

```
# Use '-' to remove elements
x[-2]
```

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

```
# Remove the collection of indices 1 and 2,
x[-c(1,2)]
```

```
## [1] 5
```

```
# Matrix : two way array
z=matrix(seq(1,12),4,3)
z
```

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

```r
# Subset matrix
z[3:4,2:3]
```

```
##      [,1] [,2]
## [1,]    7   11
## [2,]    8   12
```

```r
z[,2:3]
```

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

```r
### When we took just the first column of z, that became a vector
z[,1]
```

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

```r
### Use drop=FALSE to keep it as matrix
z[,1,drop=FALSE]
```

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

```r
### Dimensions of the matrix.
dim(z)
```

```
## [1] 4 3
```

```r
### Tell you what you have available in your working directory.
ls()
```

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

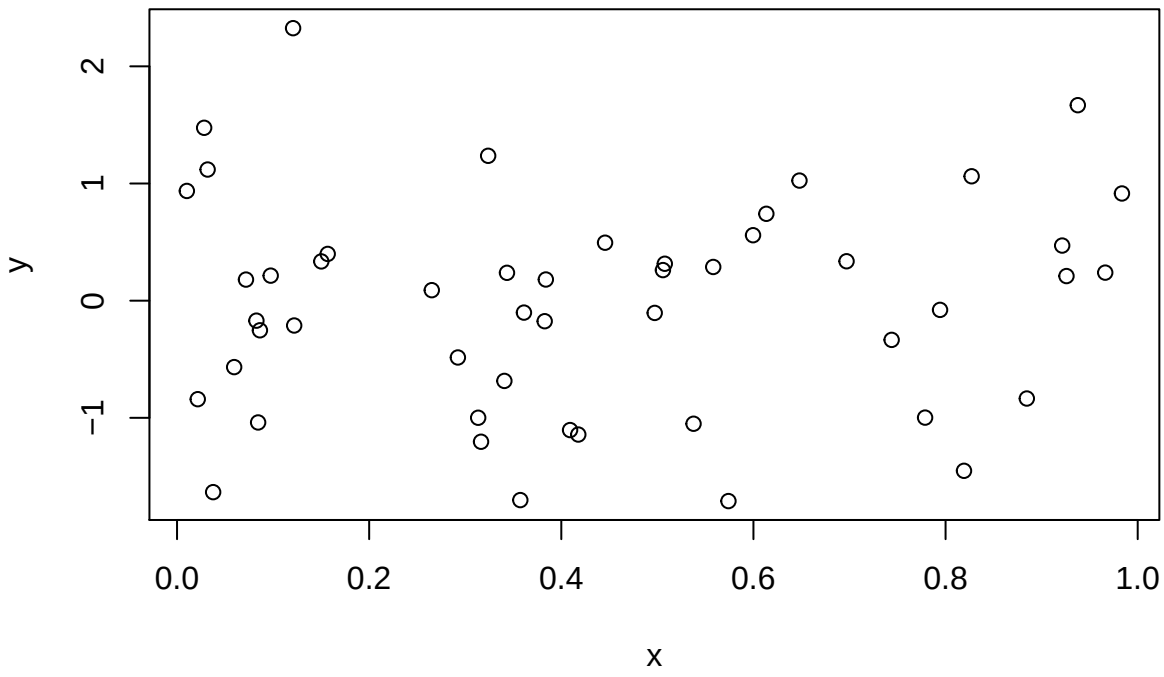
```r
### Use rm to clean up your working directory
rm(y)
ls()
```

```
## [1] "x"  "Xy" "z"
```

---

02. Generate data and Simulation

```r
# Generating random data, graphics
?runif

# random uniform
x=runif(50)

# random Gaussion
y=rnorm(50)

plot(x,y)
```



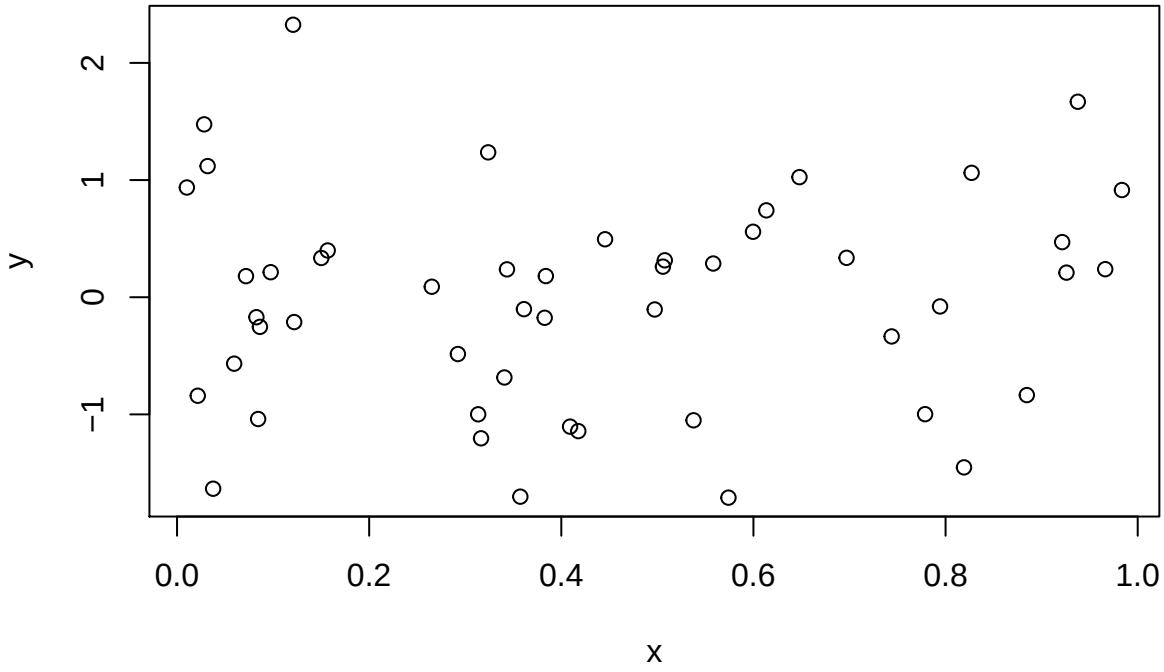
---

03. Plotting

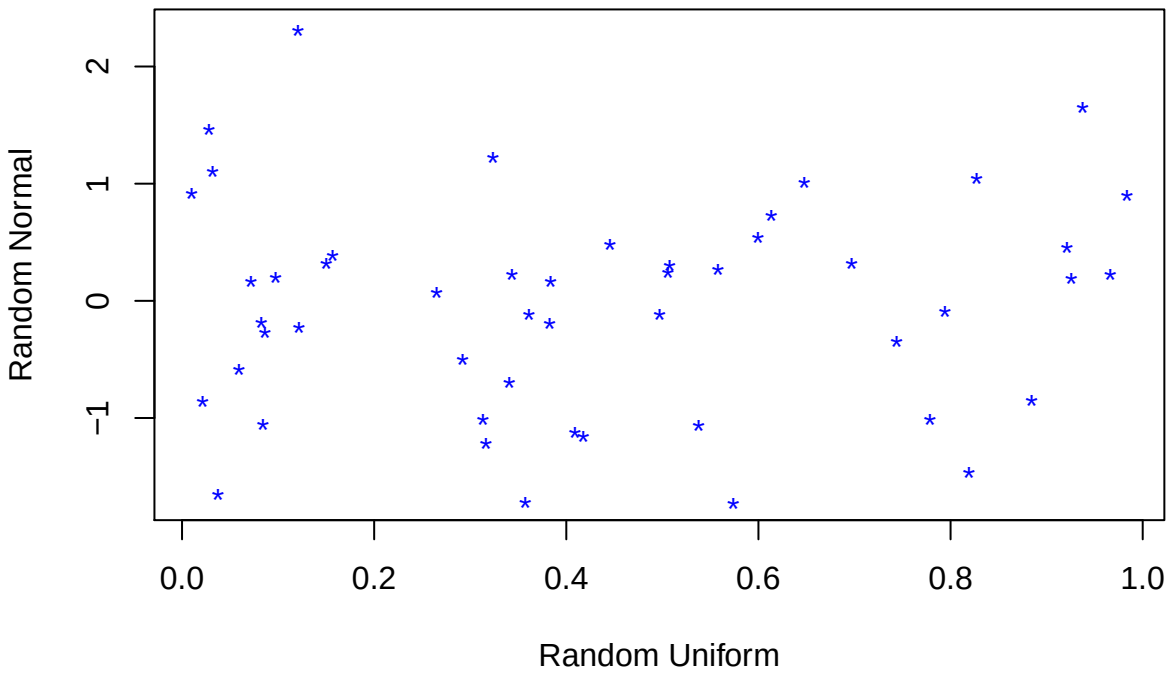
```r
plot(x,y)
```

One doesn't think too much about the design of graphics, but a lot goes into it, such as aspect ratios, how much space to put around the points on a plot, between the edge of

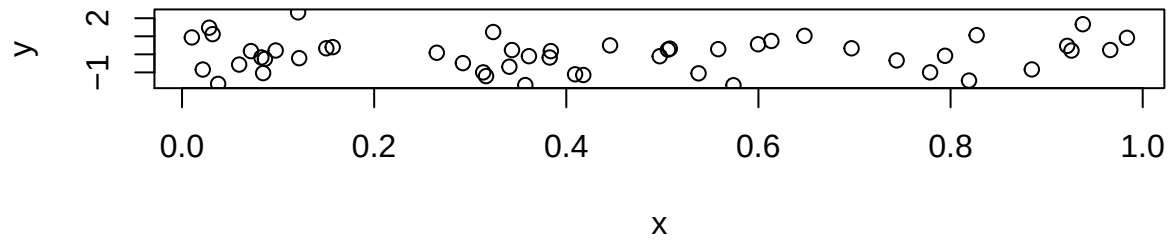
the points and the axes. Just things like spacing of the axes, how many ticks, and so on

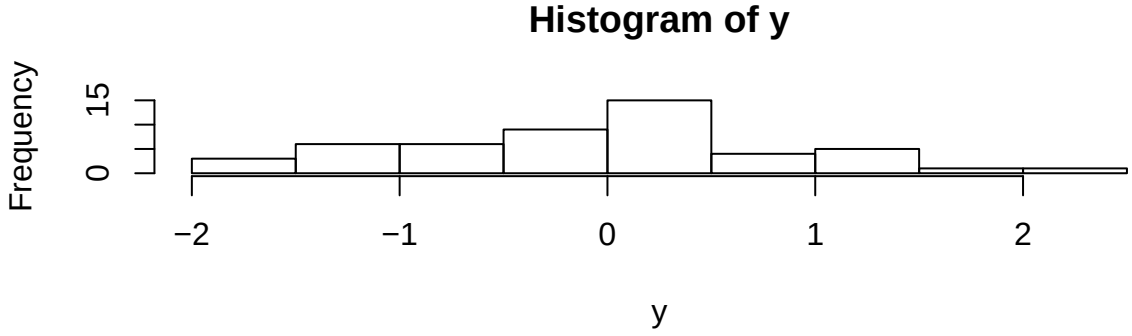



```
plot(x,y,xlab="Random Uniform",ylab="Random Normal",pch="*",col="blue")
```




```
### a panel of plots with two rows and one column.
par(mfrow=c(2,1))
plot(x,y)
hist(y)
```






```
###  reset it with another mfrow command.
par(mfrow=c(1,1))
```

---

04. Reading in data

```
### Reading in data
### Auto is in the ISLR package

names(Auto)
```

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

```
dim(Auto)
```

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

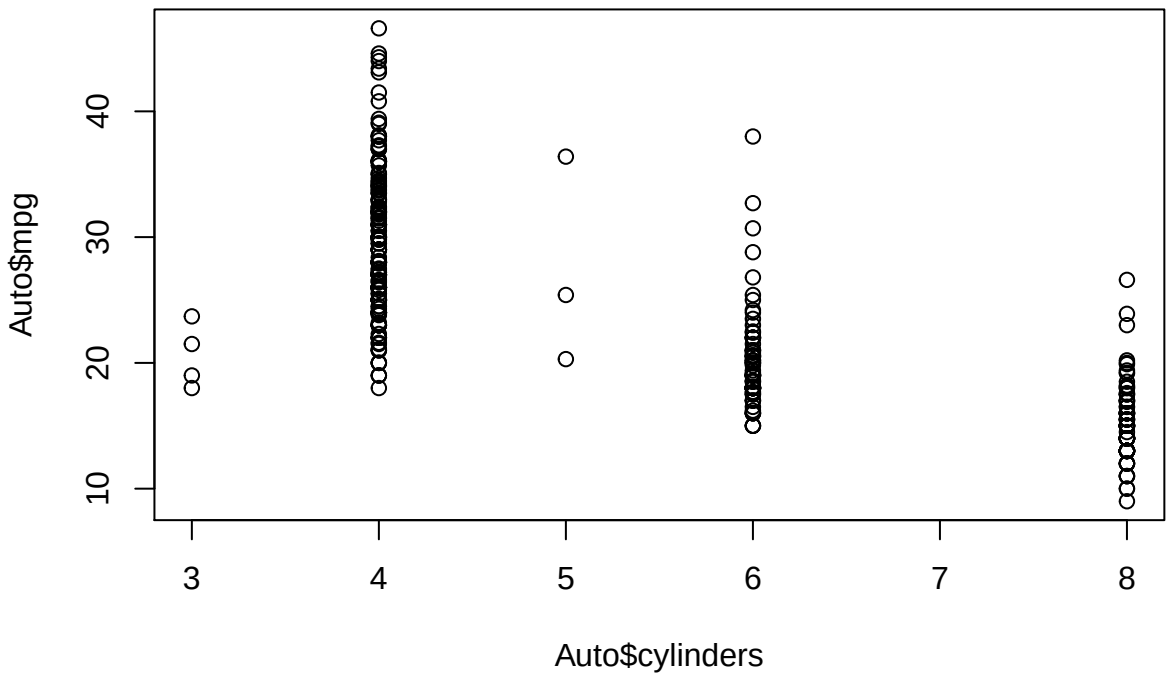
```
### Data frame
### It's sort of like a matrix, except that the columns can ### be variables of different kinds. So you
### matrices, and so on, which is really the way we think ###   of observations in statistics.

class(Auto)
```

```
## [1] "data.frame"
```

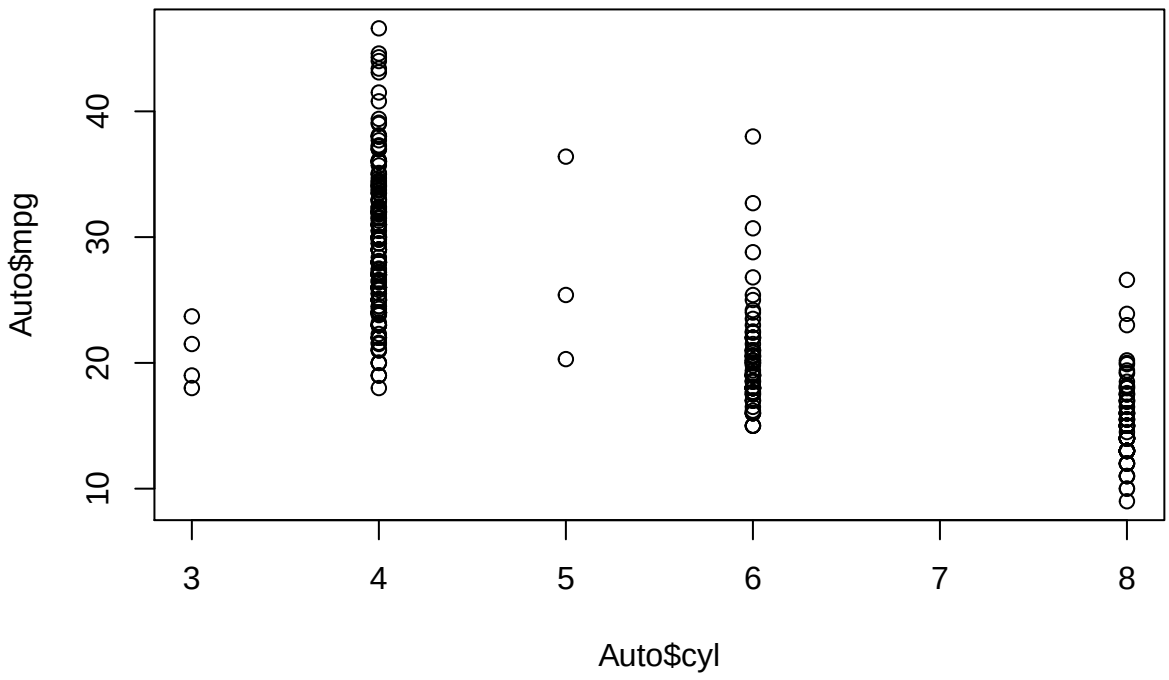
```
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##                  name
##  amc matador       :  5
##  ford pinto        :  5
##  toyota corolla    :  5
##  amc gremlin       :  4
##  amc hornet        :  4
##  chevrolet chevette:  4
##  (Other)           :365
```

```
### Data frame is also a list. getting the element of a list we can use $

plot(Auto$cylinders,Auto$mpg)
```



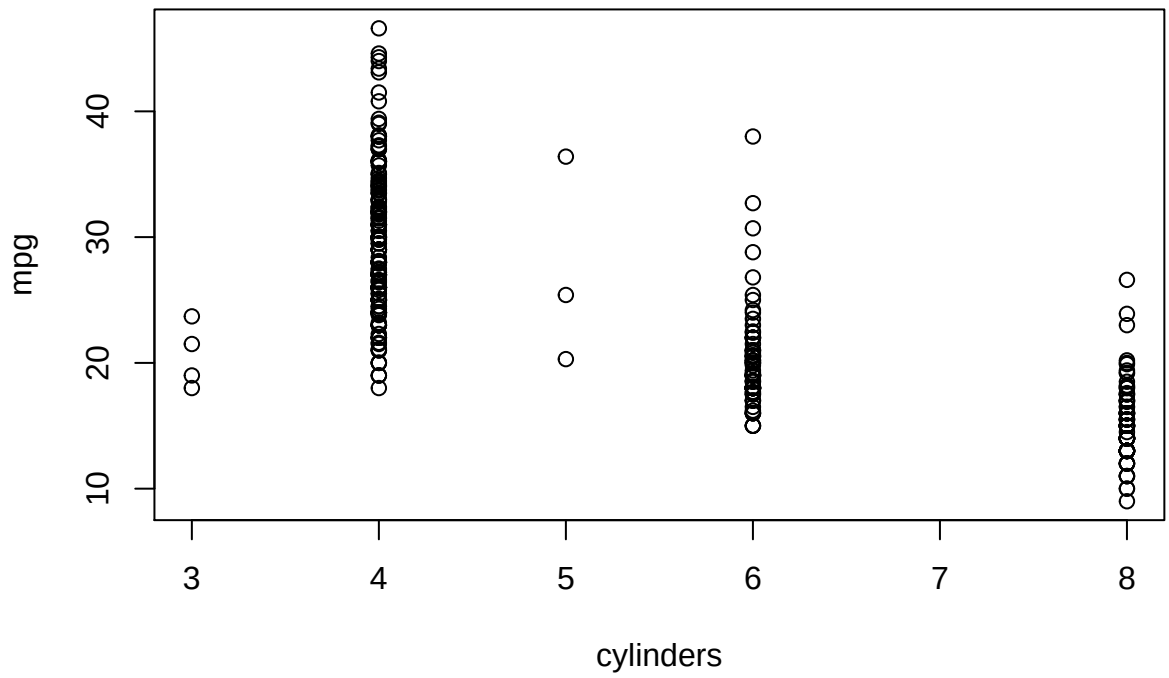

```
plot(Auto$cyl,Auto$mpg)
```



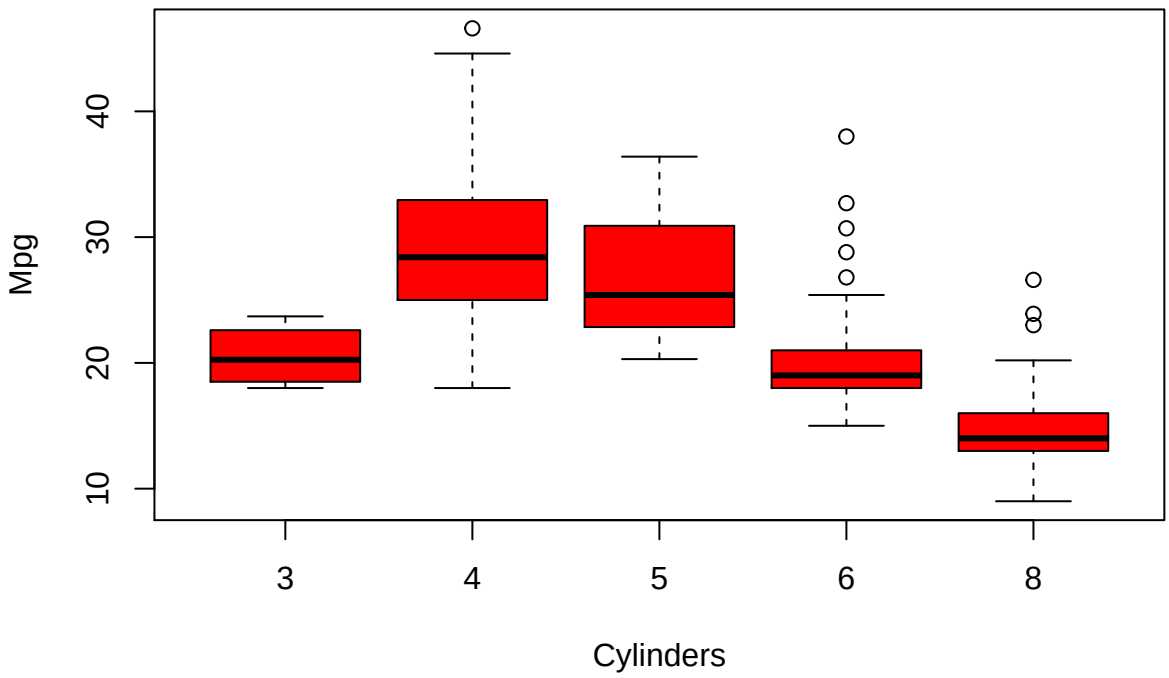

```
### attach creates a workspace with all the named variables as now variables in your workspace.So now y
attach(Auto)

### command Search, it tells us our various workspaces.
search()
```

```
##  [1] ".GlobalEnv"        "Auto"              "package:ISLR"
##  [4] "package:stats"     "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"  "package:methods"
## [10] "Autoloads"         "package:base"
```

```
plot(cylinders,mpg)
```



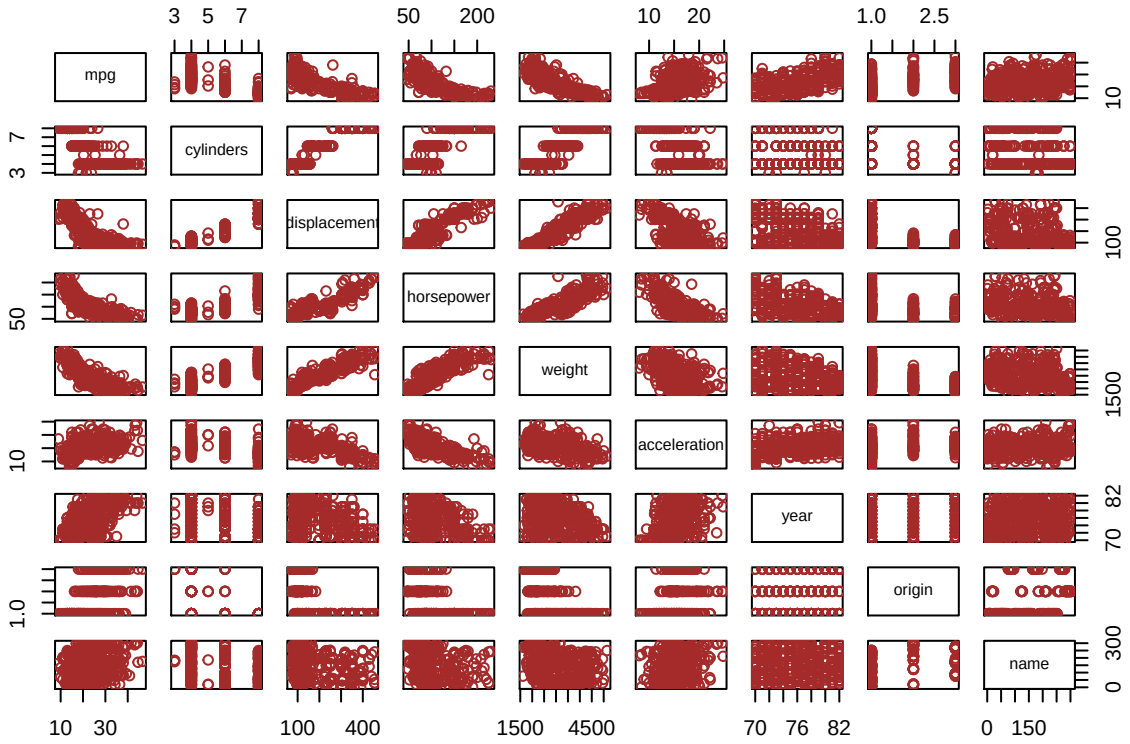

```
cylinders=as.factor(cylinders)
plot(cylinders,mpg,xlab="Cylinders",ylab="Mpg",col="red")
```



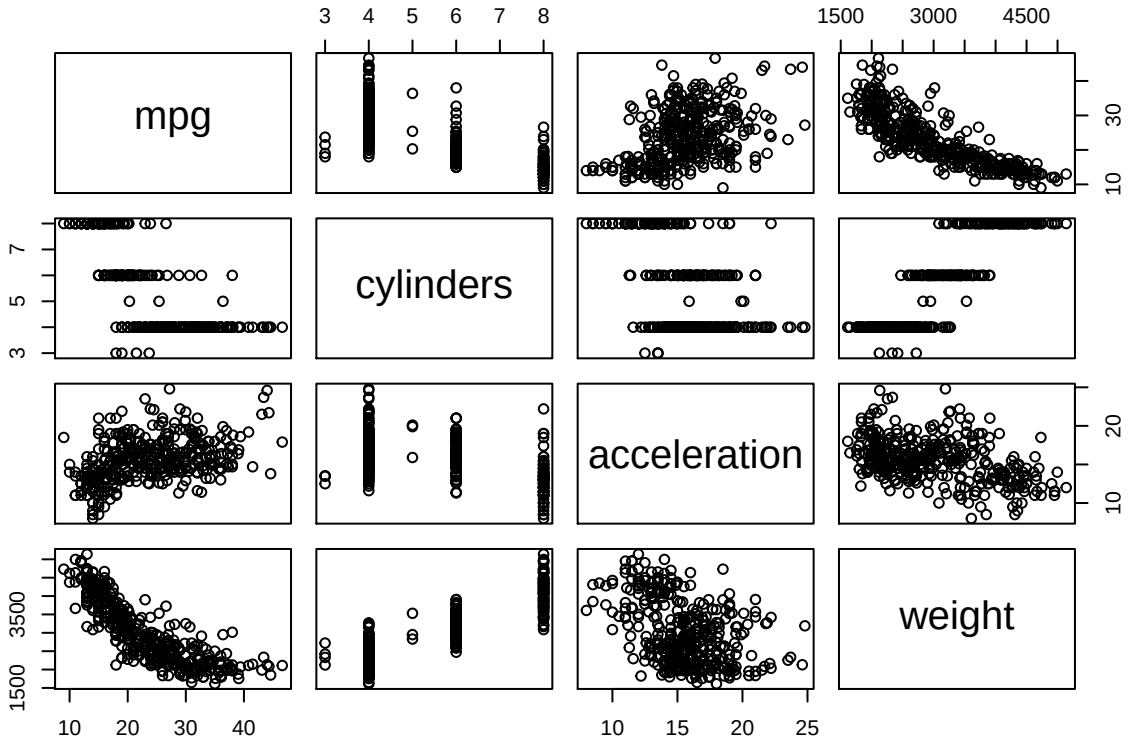

```
pdf(file="../mpg.pdf")
plot(cylinders,mpg,xlab="Cylinders",ylab="Mpg",col="red")
dev.off()
```

```
## pdf
##   2
```

```
pairs(Auto,col="brown")
```




```
pairs(mpg~cylinders+acceleration+weight,Auto)
```


mpg
cylinders
acceleration
weight
3 4 5 6 7 8
1500 3000 4500
10 30
3 5 7
10 20
1500 3500
10 20 30 40
10 15 20 25