R

Chris Qi from Data Maniac

2018-09-11

Contents

Chapter 1

21

R R R .... R (http://blog.fens.me/r-apply/)

1.1 R

R S

- S 1976
- R . (Ross Ihaka) . Robert Gentleman S 1993 1995
- R R tnitr, rmarkdown, bookdown
- R R R ’base datasets utils grDevices graphics stats’ Google Hal Varian R
- R

1.2 R

R,

- CRAN Comprehensive R Archive Network http://www.r-project.org/ R Linux Mac OS X Windows http://ftp.ctex.org/mirrors/CRAN/
- Rstudio http://www.rstudio.com/ide/download/ Rstudio desktop

R R Rstudio Rstudio R RStudio RStudio R

1.3 RStudio

https://www.dropbox.com/s/cy1ls5p6f4qqcya/rstudio.png?dl=0

/Users/yuandong/Dropbox/Public/rstudio.png

knitr::include_graphics(rep(“images/knit-logo.png”, 3))

Chapter 2

R Basics 1

R > R

2.1 R

* : + * : - * : : / * : ^ * : %%

```
# An addition
5 + 5
```

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

```
# A subtraction
5 - 5
```

```
## [1] 0
```

```
# A multiplication
3 * 5
```

```
## [1] 15
```

```
 # A division
(5 + 5) / 2
```

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

```
# Exponentiation
2^5
```

```
## [1] 32
```

```r
# Modulo
28%%6
```

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

expression

2.2

R <-

```r
my_var<-42
```

my_var RStudio “environment” my_var

my_var R Console 42

```r
my_var
```

```
## [1] 42
```

R

c() function c combine

```r
#
lucky_numbers <- c(7, 77)
lucky_numbers
```

```
## [1]  7 77
```

R

2.3 ? or help()

?c help(c) RStudio Help

2.4

c()

R base datasets utils grDevices graphics stats methods search()

install.packages()

```r
install.packages("dplyr")
```

update.packages()

R library()

```r
library(dplyr)
```

help(package="package_name") help()

2.5

* getwd() * setwd(yourpath) * RStudio ,Files tab

2.6

- numerics (1, 2.5)
- logical (TRUE or FALSE)
- characters
- factors

```r
# Change my_numeric to be 42
my_numeric <- 42

# Change my_character to be "universe"
my_character <- "universe"

# Change my_logical to be FALSE
my_logical <- FALSE

#
```

factors

class()

```r
# Declare variables of different types:
my_numeric <- 42
my_character <- "universe"
my_logical <- FALSE

# Check class of my_numeric
class(my_numeric)
```

```
## [1] "numeric"
```

```r
# Check class of my_character
class(my_character)
```

```
## [1] "character"
```

```r
# Check class of my_logical
class(my_logical)
```

```
## [1] "logical"
```

2.7

vector matrix dataframe list * * * list

2.7.1 vector

c()

```r
a <- c(1, 2, 5, 3, 6, -2, 4)
b <- c("apple", "pear", "orange")
c <- c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE)
```

a b c

’a[c(2)]’ a

```r
a[c(2)]
```

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

```r
b[c(1,3)]
```

```
## [1] "apple"  "orange"
```

```r
c[c(2:4)]
```

```
## [1] FALSE  TRUE FALSE
```

,

2.7.2 matrix

matrix()

```
myymatrix <- matrix(vector, nrow=number_of_rows, ncol=number_of_columns,
                    byrow=logical_value)
```

vector nrow ncol dimnames byrow byrow=TRUE byrow=FALSE

```
myMatrix <- matrix(1:15, nrow=3, ncol=5)
myMatrix
```

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

‘r X[i,]’ X i ’r X[,j]’ j ’r X[i, j]’ i j i j

```
y <- matrix(1:18, nrow=2)
y
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    3    5    7    9   11   13   15   17
## [2,]    2    4    6    8   10   12   14   16   18
```

```
y[2,]
```

```
## [1]  2  4  6  8 10 12 14 16 18
```

```
y[,1]
```

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

```
y[2,c(3:5)]
```

```
## [1]  6  8 10
```

2.7.3 dataframe

R

2-1

```
students <- c("A", "B", "C", "D")
math_score<-c(100, 80, 70, 95)
english_score<-c(96, 86, 77, 99)
students_scores<-data.frame(students, math_score,english_score)
```

* * *

```
students_scores[,2]
```

```
## [1] 100  80  70  95
```

```
students_scores[,"math_score"]
```

```
## [1] 100  80  70  95
```

```
students_scores$math_score
```

```
## [1] 100  80  70  95
```

$, students math

```
data.frame(students_scores$students, students_scores$math_score)
```

```
##   students_scores.students students_scores.math_score
## 1                        A                        100
## 2                        B                         80
## 3                        C                         70
## 4                        D                         95
```

2.7.4 factor

factor * *

95 90 5

R

factor() [1…k] k

```
excellence<- c("excellent", "bad", "good", "okay", "bad")
excellence<- factor(excellence)
excellence
```

```
## [1] excellent bad       good      okay      bad
## Levels: bad excellent good okay
```

```
excellence <- factor(excellence, order=TRUE,
                     levels=c("bad", "okay","good","excellent"))
excellence
```

```
## [1] excellent bad       good      okay      bad
## Levels: bad < okay < good < excellent
```

excellence

levels labels 1 2

```
sex<-c(1,2,2,1,2,1,1,3)
sex
```

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

```
sex <- factor(sex, levels=c(1, 2), labels=c("Male", "Female"))
sex
```

```
## [1] Male   Female Female Male   Female Male   Male   <NA>
## Levels: Male Female
```

“Male” “Female” 1 2 1 2

2.7.5 list

list R component list()

```
a <- "My First List"
b <- c(25, 26, 18, 39)
c <- matrix(1:10, nrow=5)
d <- c("one", "two", "three")
mylist <- list(title=a ,b,c,d)
mylist
```

```
## $title
## [1] "My First List"
##
## [[2]]
## [1] 25 26 18 39
##
## [[3]]
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
##
## [[4]]
## [1] "one"   "two"   "three"
```

2.8

length(object) /

```
length(mtcars)
```

```
## [1] 11
```

```
length(mtcars$mpg)
```

```
## [1] 32
```

dim(object)

```
dim(mtcars)
```

```
## [1] 32 11
```

str(object)

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

class(object)

```
class(mtcars)
```

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

names(object)

```
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

c(object, object,...)

```
c(2, 20)
```

```
## [1]  2 20
```

cbind(object, object, ...)

```
cbind(students, math_score)
```

```
##      students math_score
## [1,] "A"      "100"
## [2,] "B"      "80"
## [3,] "C"      "70"
## [4,] "D"      "95"
```

rbind(object, object, ...)

```
rbind(students, math_score)
```

```
##            [,1]  [,2] [,3] [,4]
## students   "A"   "B"  "C"  "D"
## math_score "100" "80" "70" "95"
```

head(object)

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

tail(object)

```
tail(mtcars)
```

```
##                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

ls()

```r
ls()
```

```
##  [1] "a"               "b"               "c"
##  [4] "d"               "english_score"   "excellence"
##  [7] "lucky_numbers"   "math_score"      "my_character"
## [10] "my_logical"      "my_numeric"      "my_var"
## [13] "mylist"          "myMatrix"        "sex"
## [16] "students"        "students_scores" "y"
```

rm(object, object, ...)

```r
rm(a, b, c)
ls()
```

```
##  [1] "d"               "english_score"   "excellence"
##  [4] "lucky_numbers"   "math_score"      "my_character"
##  [7] "my_logical"      "my_numeric"      "my_var"
## [10] "mylist"          "myMatrix"        "sex"
## [13] "students"        "students_scores" "y"
```

rm(list = ls())   4

Chapter 3

R Basics 2

3.1

“ ”

[]!(/Users/yuandong/Dropbox/Public/stats.png)

-
-
-

3.2

Summary statistics

3.2.1 : Mean) (Median) (Mode) (percentile)

- Mean):

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

- (Median):

```
median(mtcars$mpg)
```

```
## [1] 19.2
```

* (Mode): R

```
names(table(mtcars$mpg))[which.max(table(mtcars$mpg))]
```

```
## [1] "10.4"
```

* (percentile)

```
quantile(mtcars$mpg)
```

```
##     0%    25%    50%    75%   100% 
## 10.400 15.425 19.200 22.800 33.900 
```

3.2.2 var) range)

- Variance): . (sample variance) , s2:

$$s2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

*

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

```
var(mtcars$mpg)
```

```
## [1] 36.3241
```

- Range): . ,

```
range(mtcars$mpg)
```

```
## [1] 10.4 33.9
```

```
summary(mtcars$mpg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   10.40   15.43   19.20   20.09   22.80   33.90 
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

3.3

R Actions in R

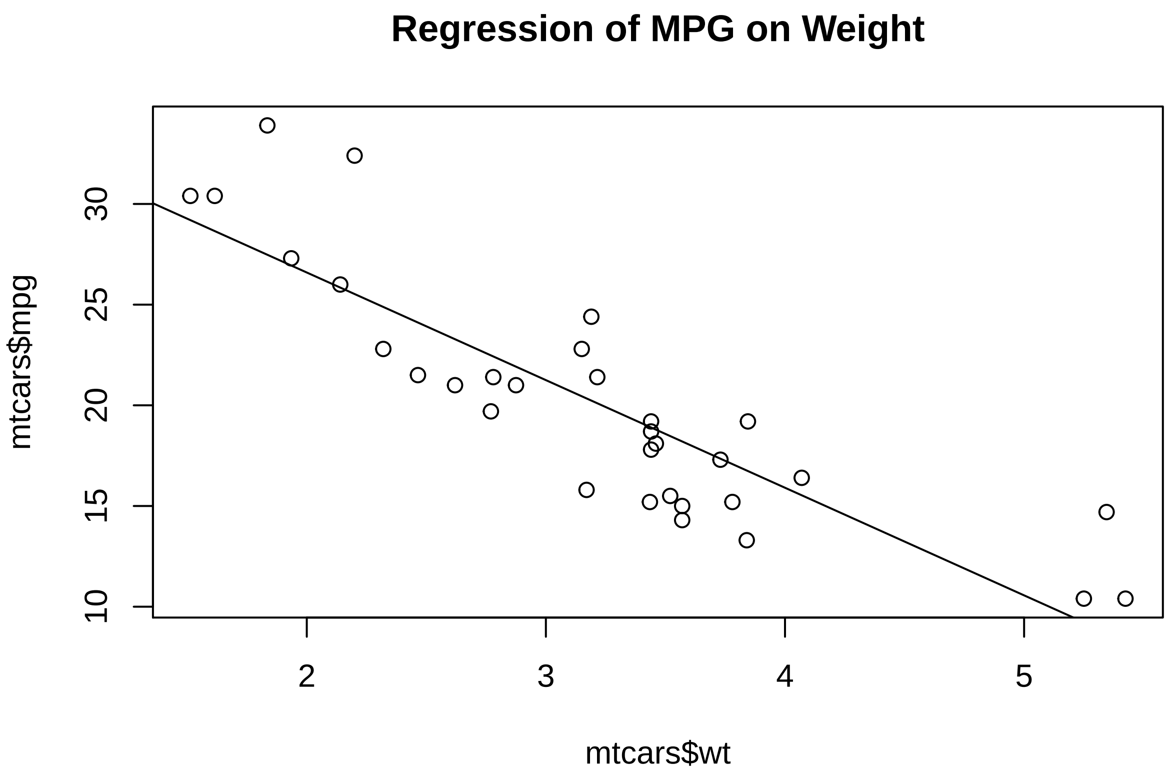
R

plot() R ( )

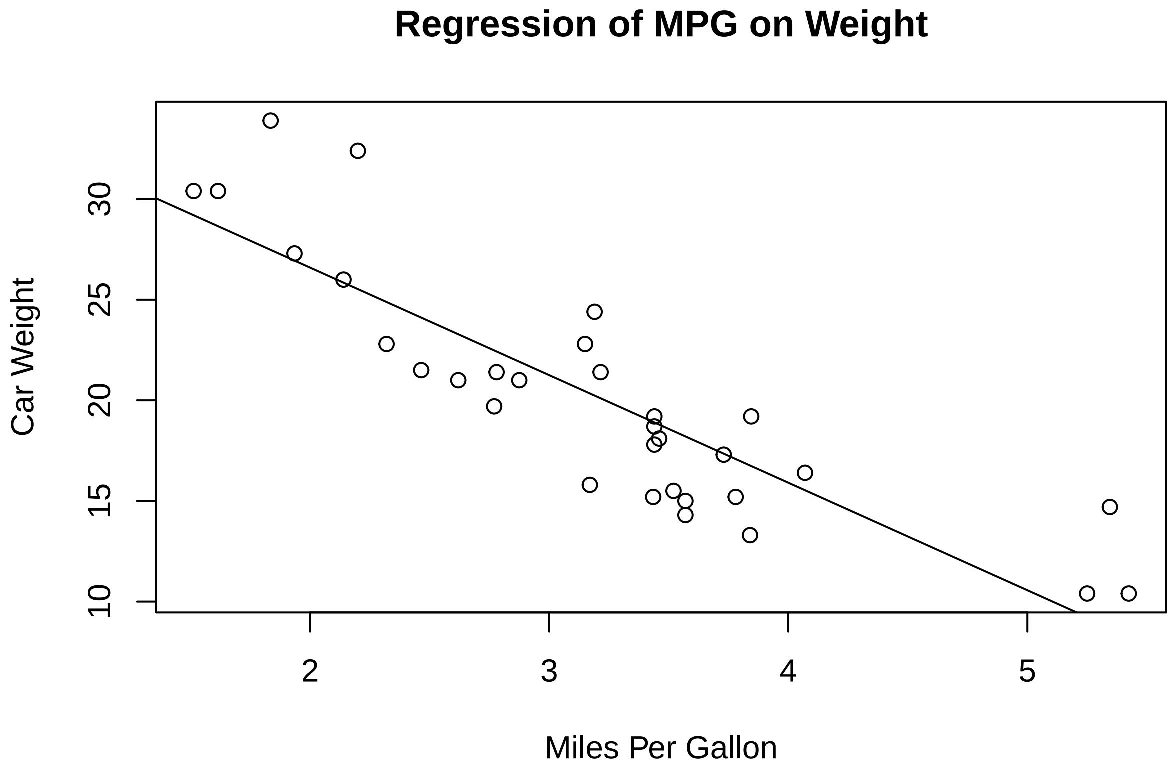
```
plot(x, y, type="b")
```

x y (x, y) help(plot)

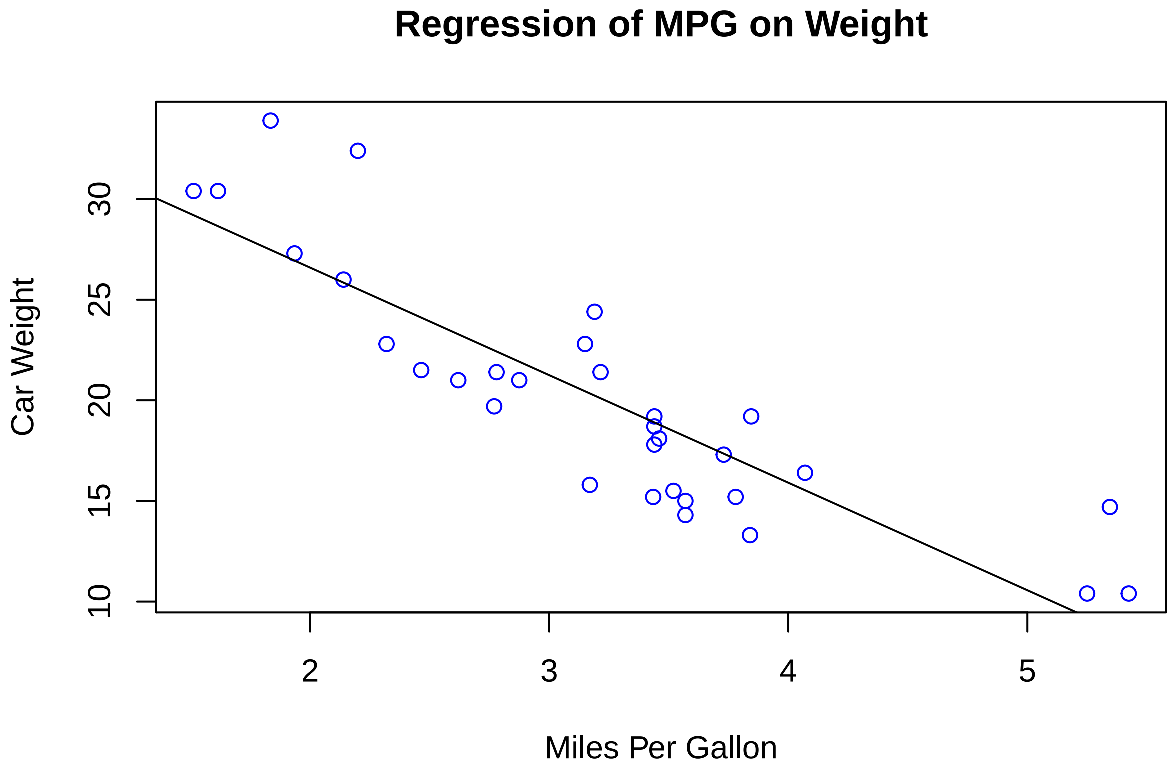
```
plot(mtcars$wt, mtcars$mpg)
abline(lm(mtcars$mpg~mtcars$wt))
title("Regression of MPG on Weight")
```



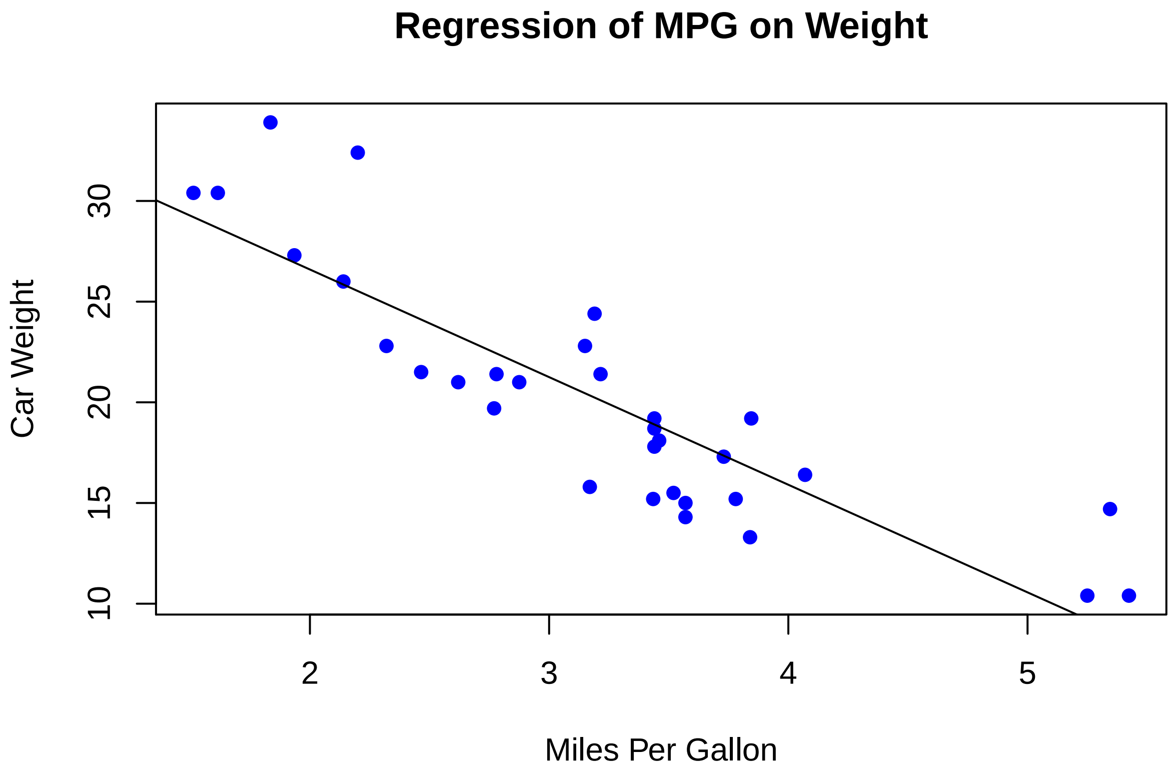
```
plot(mtcars$wt, mtcars$mpg,
         xlab="Miles Per Gallon",
         ylab="Car Weight")
abline(lm(mtcars$mpg~mtcars$wt))
title("Regression of MPG on Weight")
```



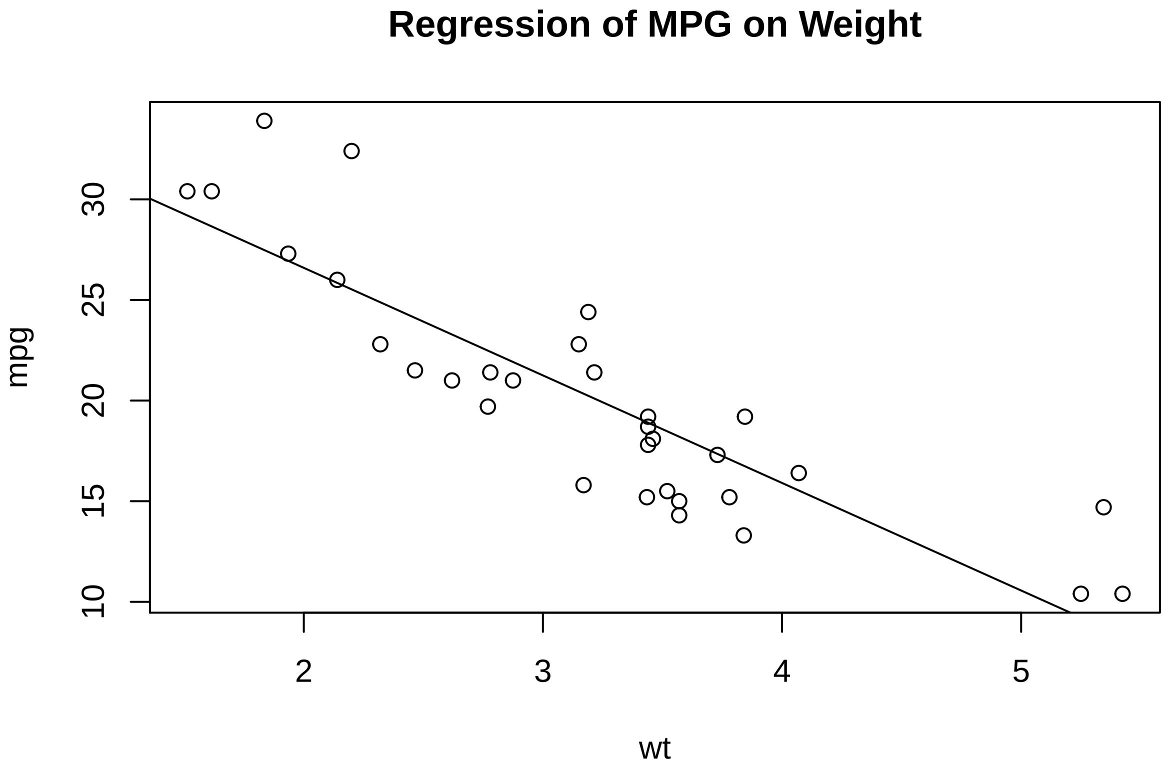
```
plot(mtcars$wt, mtcars$mpg,
         xlab="Miles Per Gallon",
         ylab="Car Weight",
     col=4)
abline(lm(mtcars$mpg~mtcars$wt))
title("Regression of MPG on Weight")
```



```r
plot(mtcars$wt, mtcars$mpg,
         xlab="Miles Per Gallon",
         ylab="Car Weight",
     col=4,
     pch=16)
abline(lm(mtcars$mpg~mtcars$wt))
title("Regression of MPG on Weight")
```



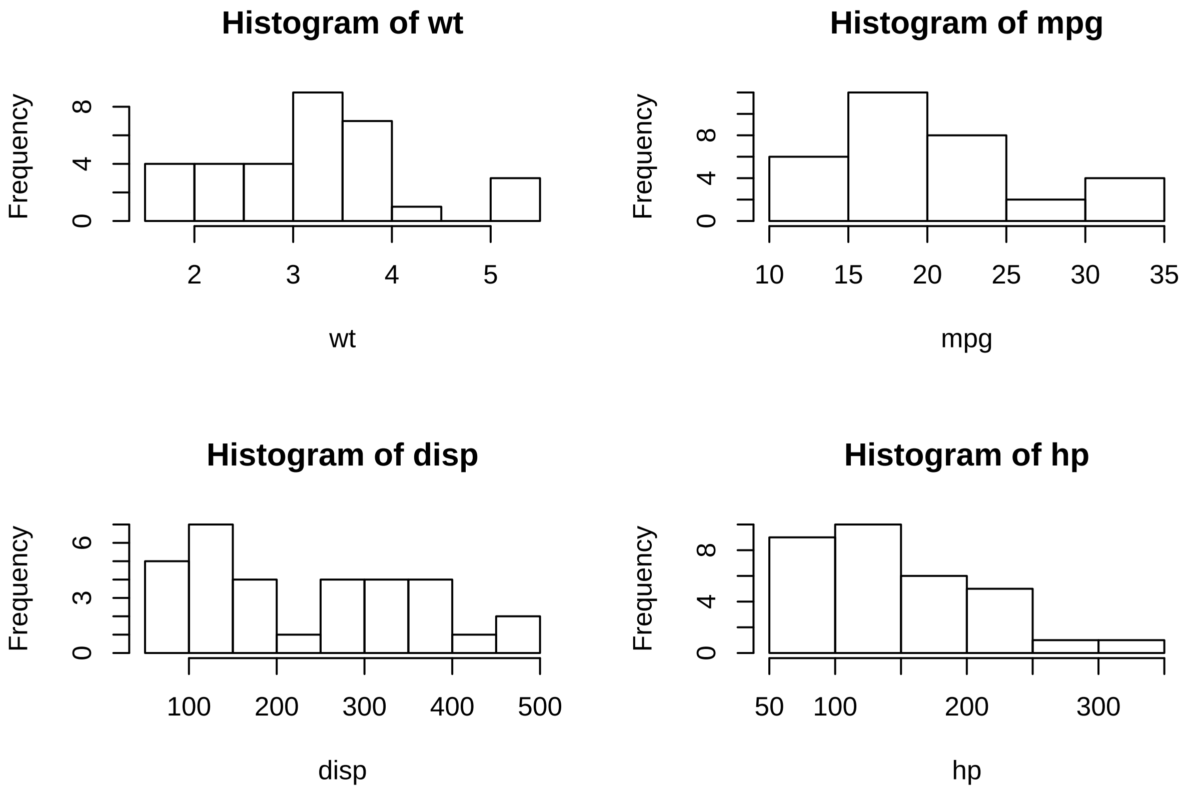
```
with(mtcars,{
plot(wt, mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
}
)
```



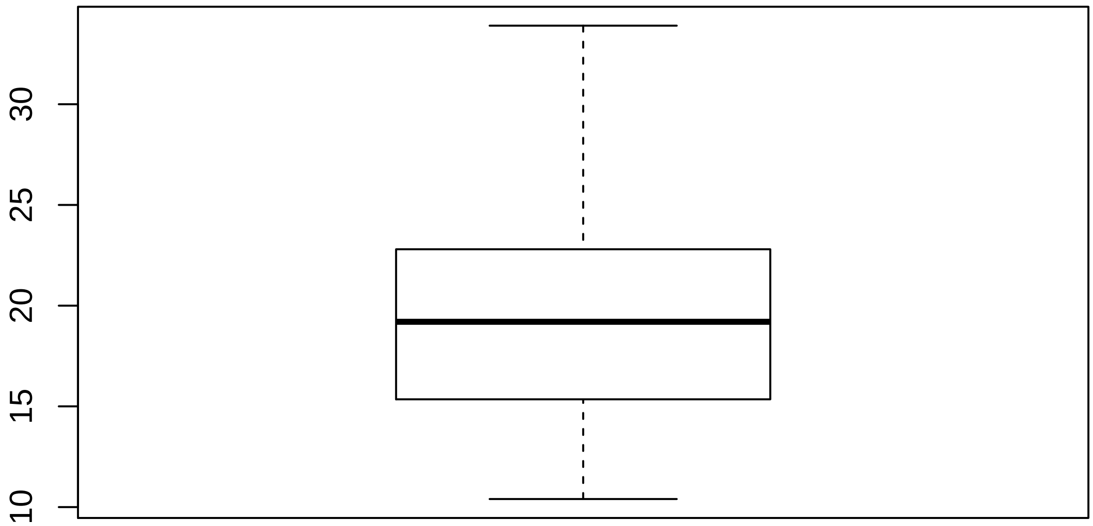
, 2 2

```r
par(mfrow=c(3,1))
```

```r
with(mtcars,{
  par(mfrow=c(2,2))
  hist(wt)
  hist(mpg)
  hist(disp)
  hist(hp)
})
```



```r
boxplot(mtcars$mpg)
```



mygraph.pdf PDF (R in Action):

```r
pdf("mygraph.pdf")
      attach(mtcars)
```

```
## The following object is masked from package:ggplot2:
```

```
##
##     mpg
```

```
      plot(wt, mpg)
      abline(lm(mpg~wt))
      title("Regression of MPG on Weight")
      detach(mtcars)
dev.off()
```

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

pdf() win.metafile() png() jpeg() bmp()

RStudio “Export”

Chapter 4

4.1

4.2

4.3

4.4

Chapter 5

R apply

http://blog.fens.me/r-apply/

R R

R for while R C C apply apply, sapply, tapply, mapply, lapply, rapply, vapply, eapply

5.1 apply

apply R apply R apply apply

R for apply R C for R

apply apply 8

apply sapply 8

5.2 apply

apply for apply ( ) FUN

```
apply(X, MARGIN, FUN, ...)
```

- X:
- MARGIN: 1 2
- FUN:
- ...:

apply

```
x<-matrix(1:12,ncol=3)
apply(x,1,sum)
```

```
## [1] 15 18 21 24
```

x1 1 x1,x2

data.frame

```
x <- cbind(x1 = 3, x2 = c(4:1, 2:5))
x
```

```
##      x1 x2
## [1,]  3  4
## [2,]  3  3
## [3,]  3  2
## [4,]  3  1
## [5,]  3  2
## [6,]  3  3
## [7,]  3  4
## [8,]  3  5
```

```
#    myFUN    x
#             apply '...'
myFUN<- function(x, c1, c2) {
c(sum(x[c1],1), mean(x[c2]))
}

#            myFUN   c1,c2 myFUN
apply(x,1,myFUN,c1='x1',c2=c('x1','x2'))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]  4.0    4  4.0    4  4.0    4  4.0    4
## [2,]  3.5    3  2.5    2  2.5    3  3.5    4
```

myFUN

for

```
#
df<-data.frame()

#  for
for(i in 1:nrow(x)){
  row<-x[i,]                                        #
  df<-rbind(df,rbind(c(sum(row[1],1), mean(row))))    #
  }

#
df
```

for

R

```
data.frame(x1=x[,1] 1,x2=rowMeans(x))
```

3

```
#
 rm(list=ls())

#  fun1
 fun1<-function(x){
   myFUN<- function(x, c1, c2) {
     c(sum(x[c1],1), mean(x[c2]))
   }
   apply(x,1,myFUN,c1='x1',c2=c('x1','x2'))
 }

#  fun2
 fun2<-function(x){
   df<-data.frame()
   for(i in 1:nrow(x)){
     row<-x[i,]
     df<-rbind(df,rbind(c(sum(row[1],1), mean(row))))
   }
 }

#  fun3
 fun3<-function(x){
   data.frame(x1=x[,1]+1,x2=rowMeans(x))
 }

#
x <- cbind(x1=3, x2 = c(400:1, 2:500))

#   3  CPU
 system.time(fun1(x))
```

```
##    user  system elapsed
##   0.010   0.000   0.011
```

```
system.time(fun2(x))
```

```
##    user  system elapsed
##   0.163   0.008   0.171
```

```
system.time(fun3(x))
```

```
##    user  system elapsed
##       0       0       0
```

CPU for apply R R apply for,while

5.3 lapply

lapply list data.frame X list lapply ’l’

```
lapply(X, FUN, ...)
```

5.4 dplyr package

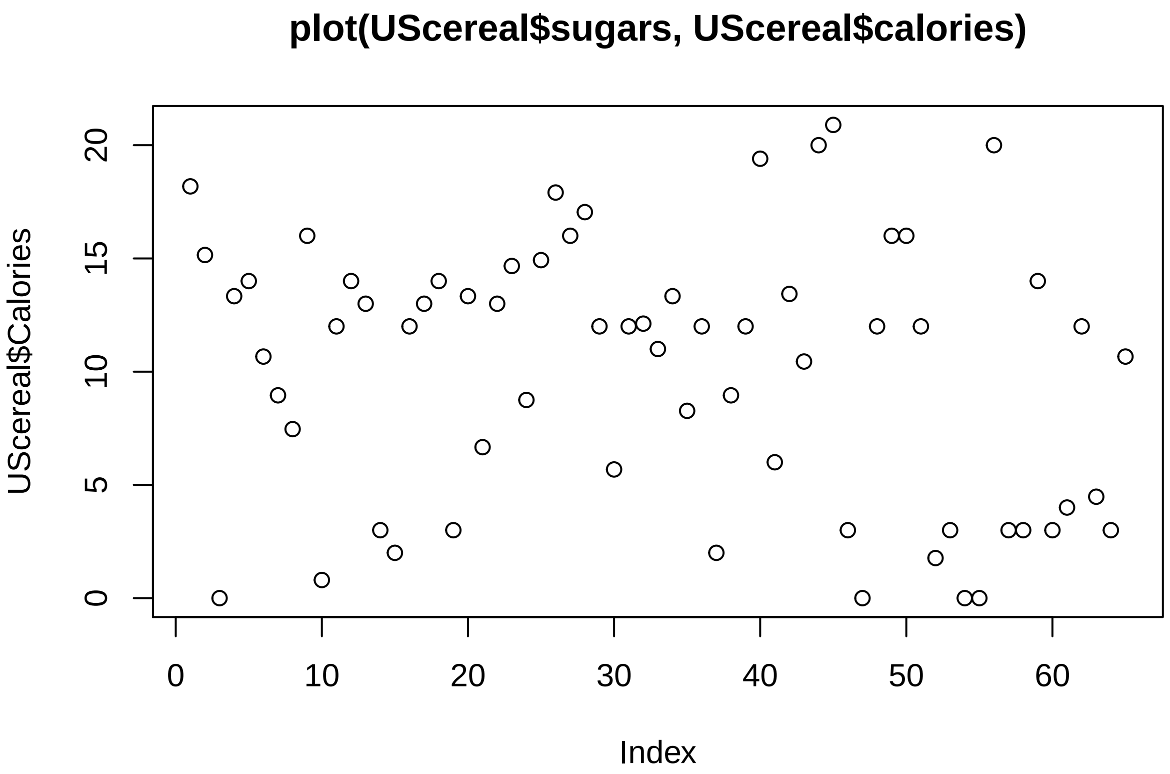
Chapter 6

R base graphics in R

- Base graphics: , ,
- Grid graphics:
- Lattice graphics: grid graphics
- ggplot2:

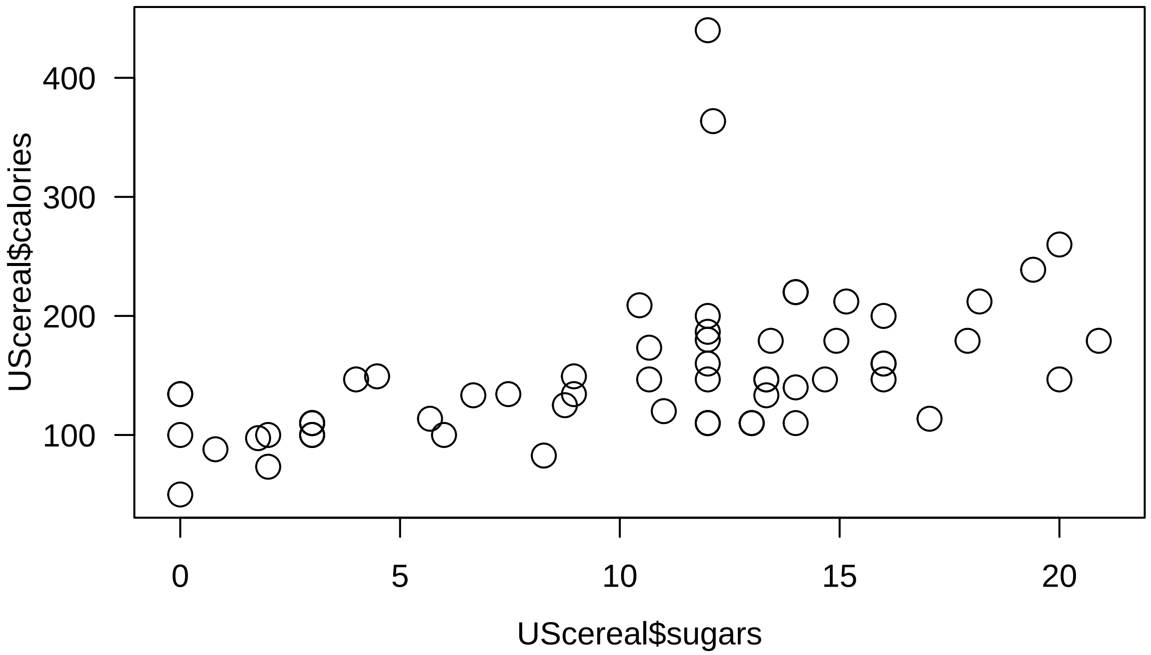
Base graphics:

```r
 library(MASS)
plot(UScereal$sugars, UScereal$Calories)
title("plot(UScereal$sugars, UScereal$calories)")
```



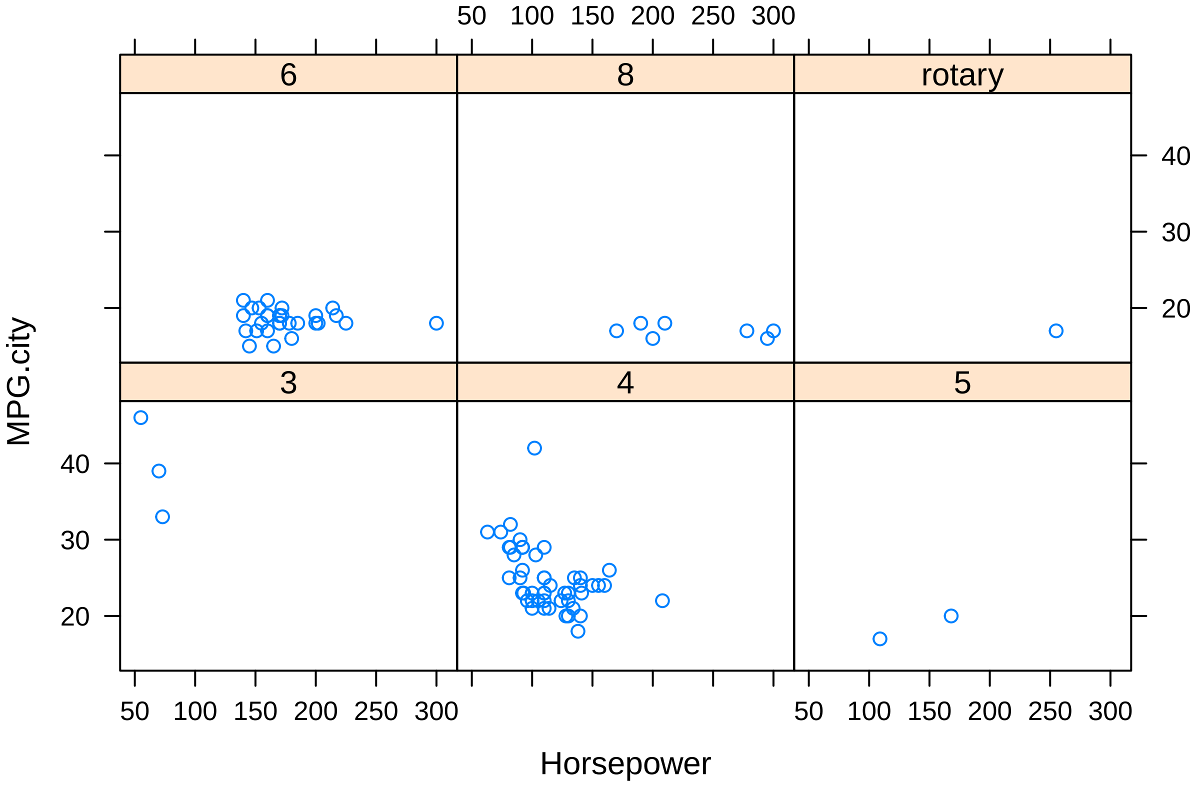
Grid graphics :

```
# Get the data and load the grid package
library(MASS)
x <- UScereal$sugars
y <- UScereal$calories
library(grid)
# This is the grid code required to generate the plot
pushViewport(plotViewport())
pushViewport(dataViewport(x, y))
grid.rect()
grid.xaxis()
grid.yaxis()
grid.points(x, y)
grid.text("UScereal$calories", x = unit(-3, "lines"), rot = 90)
grid.text("UScereal$sugars", y = unit(-3, "lines"), rot = 0)
popViewport(2)
```



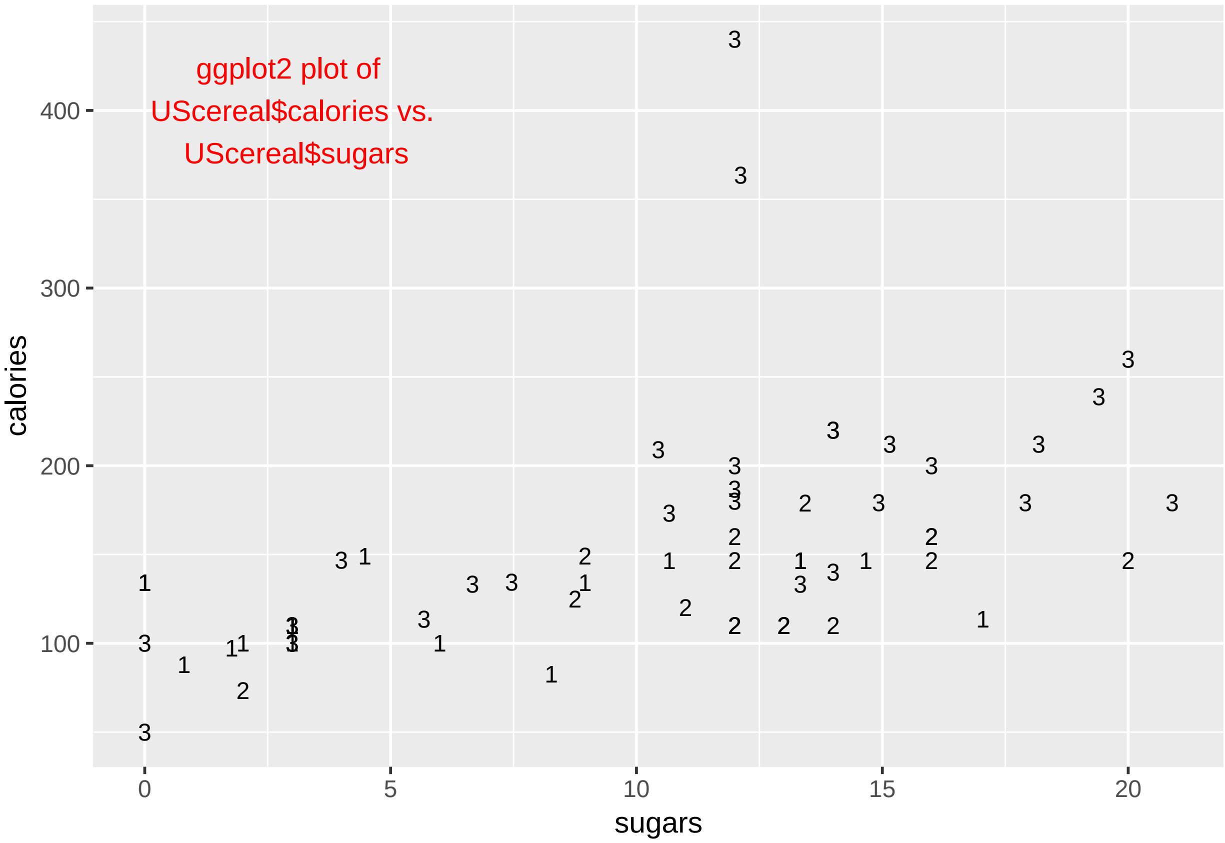
Lattice graphics:

```
library(MASS)
library(lattice)
xyplot(MPG.city ~ Horsepower | Cylinders, data = Cars93)
```



ggplot2:

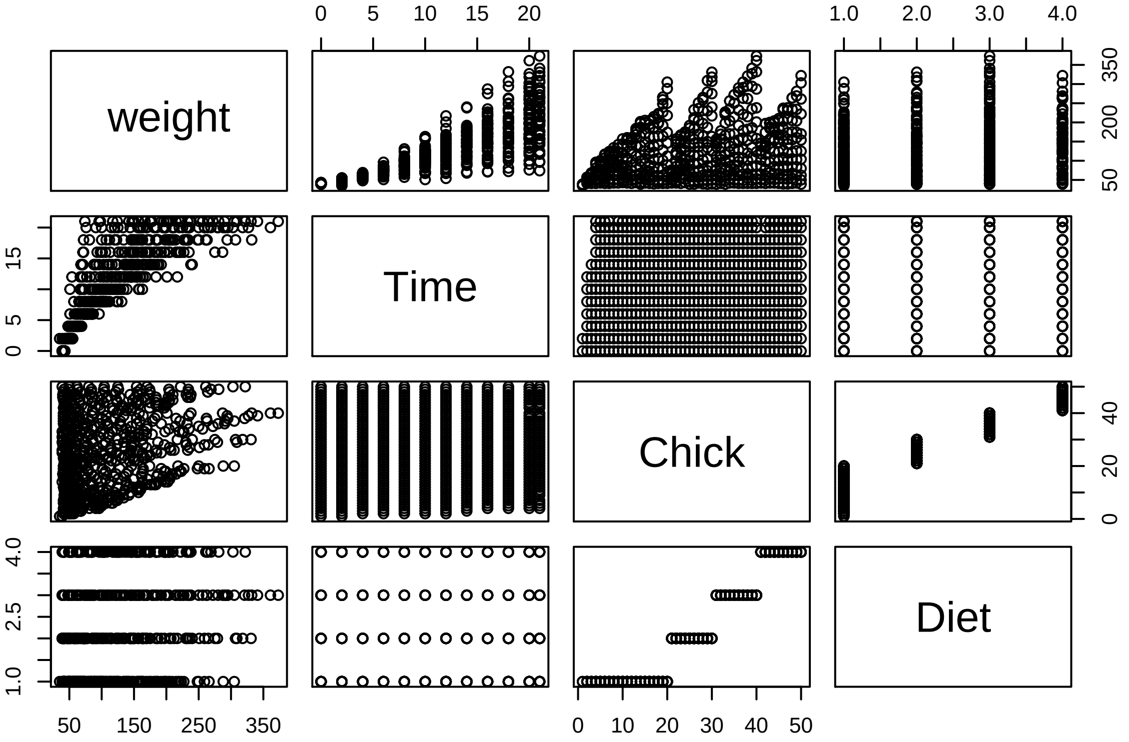
```
library(MASS)
library(ggplot2)
title <-
 "ggplot2 plot of \n UScereal$calories vs. \n UScereal$sugars"
basePlot <- ggplot(UScereal, aes(x = sugars, y = calories))
basePlot +
 geom_point(shape = as.character(UScereal$shelf), size = 3) +
 annotate("text", label = title, x = 3, y = 400,
 colour = "red")
```



- :

ChickWeight

```
plot(ChickWeight)
```



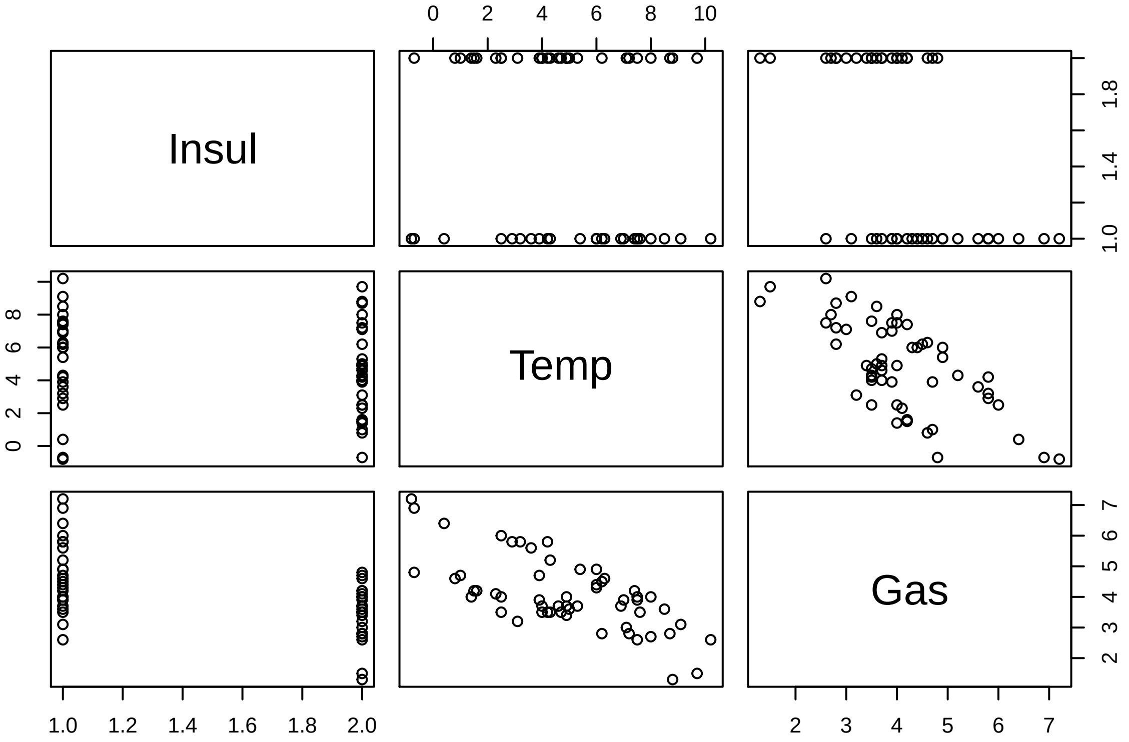
-

```
install.packages("insuranceData")
install.packages("MASS")
install.packages("robustbase")
install.packages("car")
install.packages("aplpack")
install.packages("corrplot")
install.packages("rpart")
```

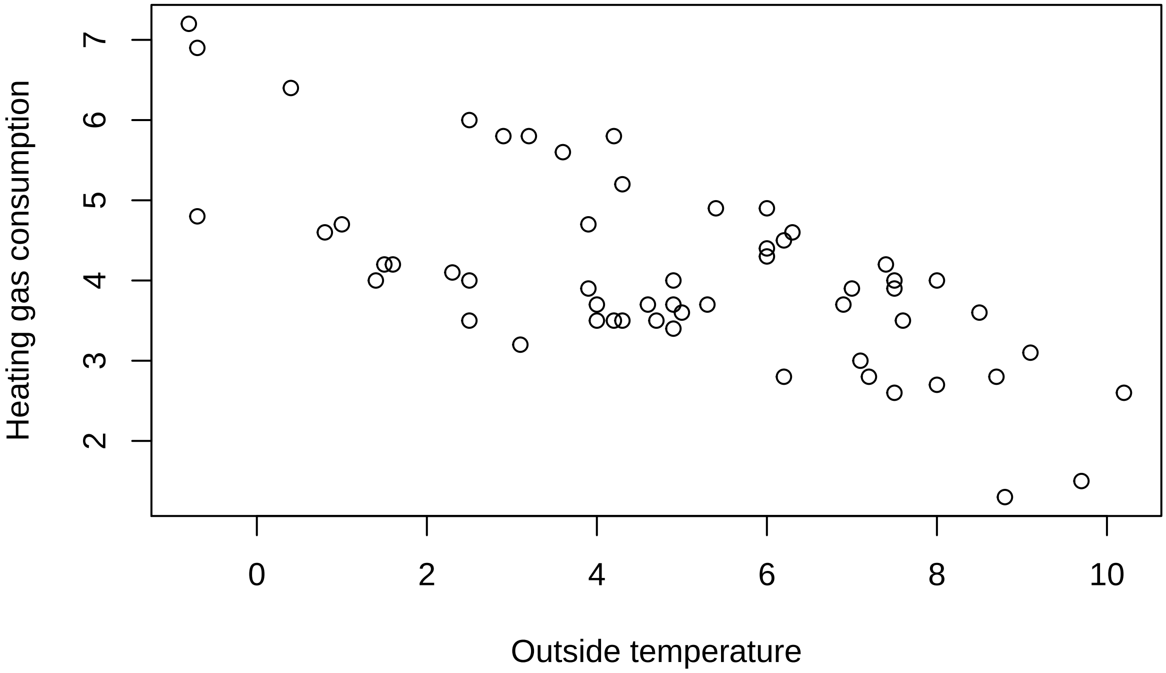
```
# Load MASS package
library(dplyr)
library(MASS)
library(robustbase)
library(insuranceData)
library(car)
```

Temp: Gas: Insul:

```
# Plot whiteside data
plot(whiteside)
```

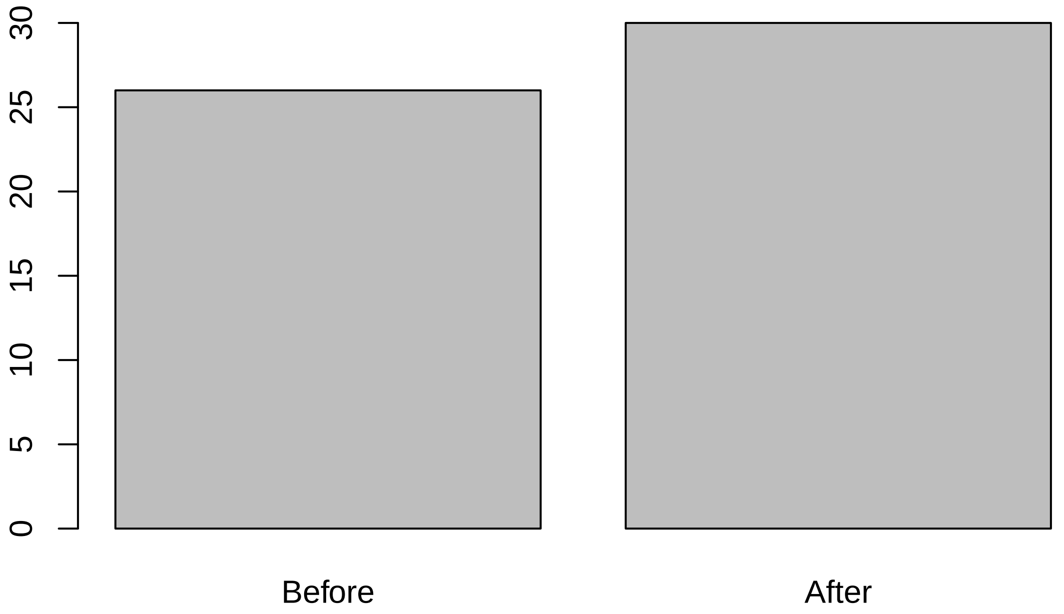


```
plot(whiteside$Temp,whiteside$Gas,
     xlab="Outside temperature",
     ylab= "Heating gas consumption")
```



`plot()` `whiteside$Insul`

```r
plot(whiteside$Insul)
```



Cars93

Price: Max.Price: Min.Price:

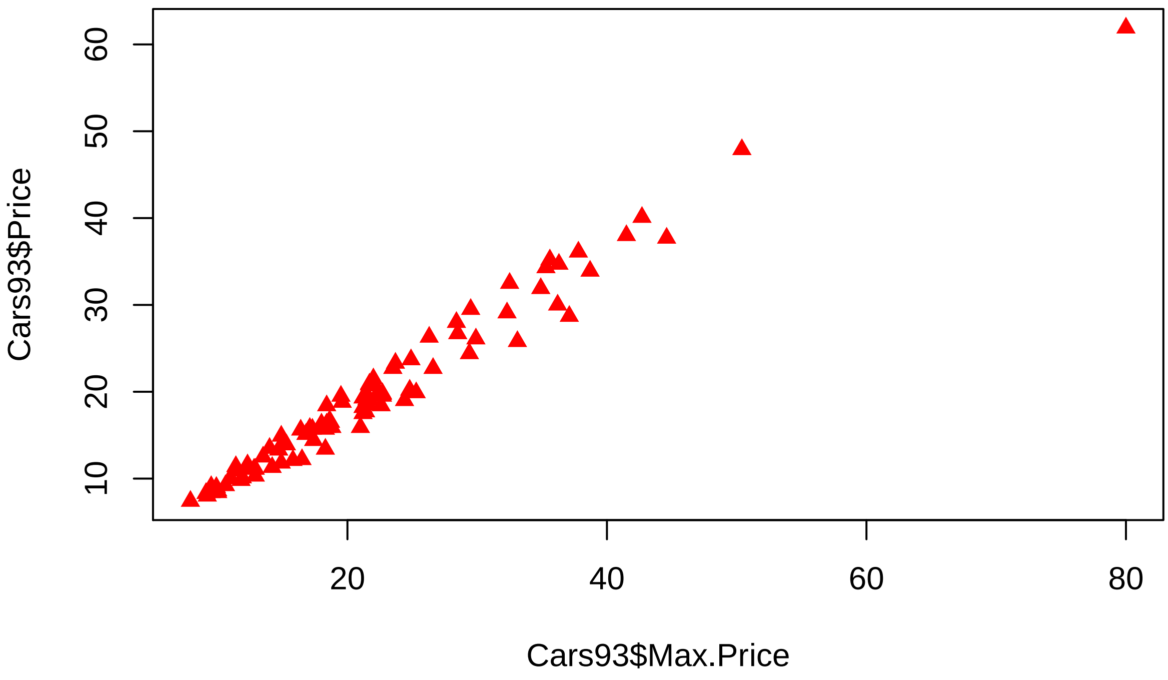
glimpse() Cars93

```r
glimpse(Cars93)
```

```
## Observations: 93
## Variables: 27
## $ Manufacturer       <fct> Acura, Acura, Audi, Audi, BMW, Buick, Buick...
## $ Model              <fct> Integra, Legend, 90, 100, 535i, Century, Le...
## $ Type               <fct> Small, Midsize, Compact, Midsize, Midsize, ...
## $ Min.Price          <dbl> 12.9, 29.2, 25.9, 30.8, 23.7, 14.2, 19.9, 2...
## $ Price              <dbl> 15.9, 33.9, 29.1, 37.7, 30.0, 15.7, 20.8, 2...
## $ Max.Price          <dbl> 18.8, 38.7, 32.3, 44.6, 36.2, 17.3, 21.7, 2...
## $ MPG.city           <int> 25, 18, 20, 19, 22, 22, 19, 16, 19, 16, 16,...
## $ MPG.highway        <int> 31, 25, 26, 26, 30, 31, 28, 25, 27, 25, 25,...
## $ AirBags            <fct> None, Driver & Passenger, Driver only, Driv...
## $ DriveTrain         <fct> Front, Front, Front, Front, Rear, Front, Fr...
## $ Cylinders          <fct> 4, 6, 6, 6, 4, 4, 6, 6, 6, 8, 8, 4, 4, 6, 4...
## $ EngineSize         <dbl> 1.8, 3.2, 2.8, 2.8, 3.5, 2.2, 3.8, 5.7, 3.8...
## $ Horsepower         <int> 140, 200, 172, 172, 208, 110, 170, 180, 170...
## $ RPM                <int> 6300, 5500, 5500, 5500, 5700, 5200, 4800, 4...
## $ Rev.per.mile       <int> 2890, 2335, 2280, 2535, 2545, 2565, 1570, 1...
## $ Man.trans.avail    <fct> Yes, Yes, Yes, Yes, Yes, No, No, No, No, No...
## $ Fuel.tank.capacity <dbl> 13.2, 18.0, 16.9, 21.1, 21.1, 16.4, 18.0, 2...
## $ Passengers         <int> 5, 5, 5, 6, 4, 6, 6, 6, 5, 6, 5, 5, 5, 4, 6...
## $ Length             <int> 177, 195, 180, 193, 186, 189, 200, 216, 198...
## $ Wheelbase          <int> 102, 115, 102, 106, 109, 105, 111, 116, 108...
## $ Width              <int> 68, 71, 67, 70, 69, 69, 74, 78, 73, 73, 74,...
## $ Turn.circle        <int> 37, 38, 37, 37, 39, 41, 42, 45, 41, 43, 44,...
## $ Rear.seat.room     <dbl> 26.5, 30.0, 28.0, 31.0, 27.0, 28.0, 30.5, 3...
## $ Luggage.room       <int> 11, 15, 14, 17, 13, 16, 17, 21, 14, 18, 14,...
## $ Weight             <int> 2705, 3560, 3375, 3405, 3640, 2880, 3470, 4...
## $ Origin             <fct> non-USA, non-USA, non-USA, non-USA, non-USA...
## $ Make               <fct> Acura Integra, Acura Legend, Audi 90, Audi ...
```

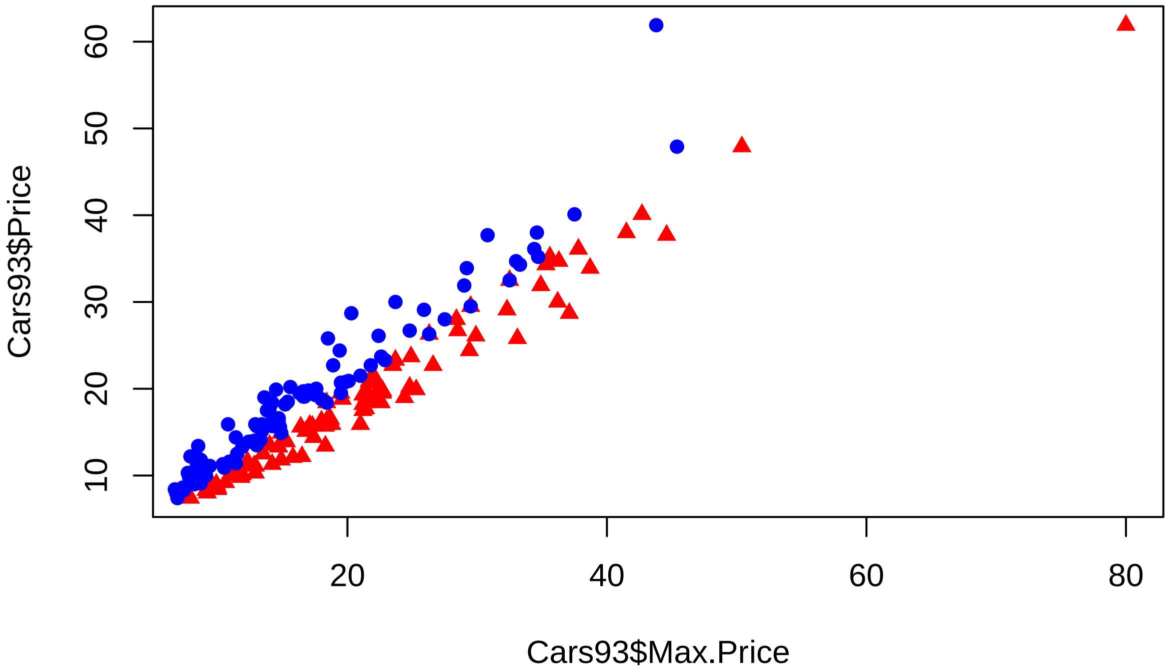
vs.

```
plot(Cars93$Max.Price, Cars93$Price,col="red",pch=17)
```



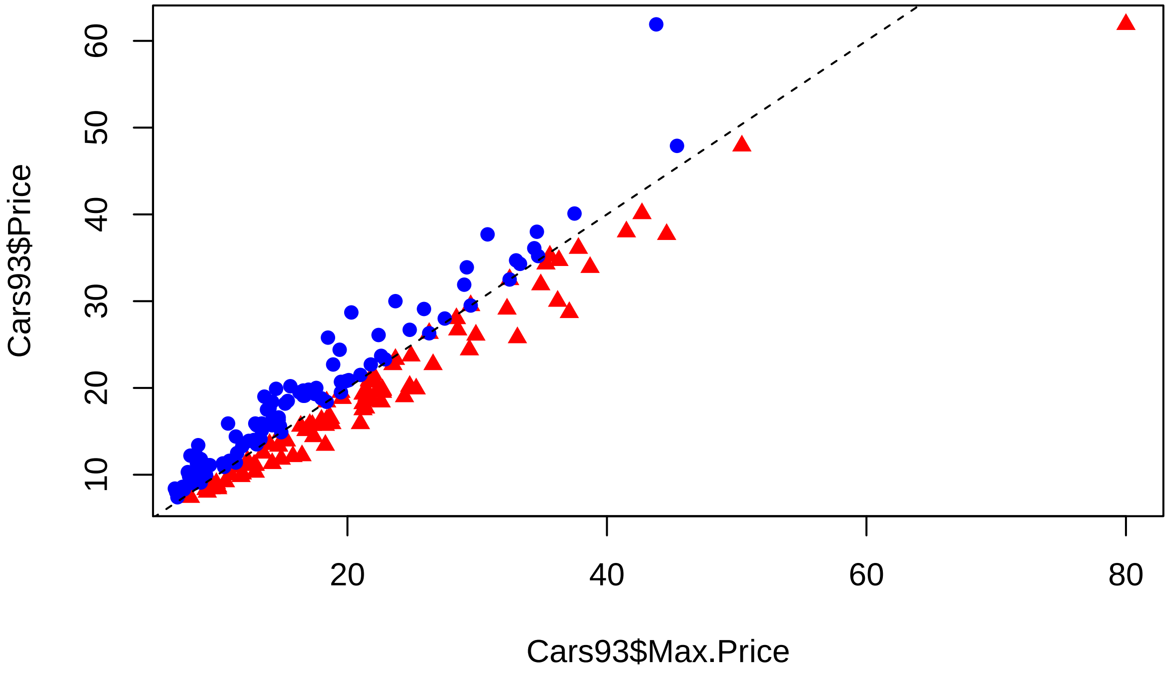
vs.

```
plot(Cars93$Max.Price, Cars93$Price,col="red",pch=17)
points(Cars93$Min.Price,Cars93$Price,col="blue",pch=16)
```



0 1

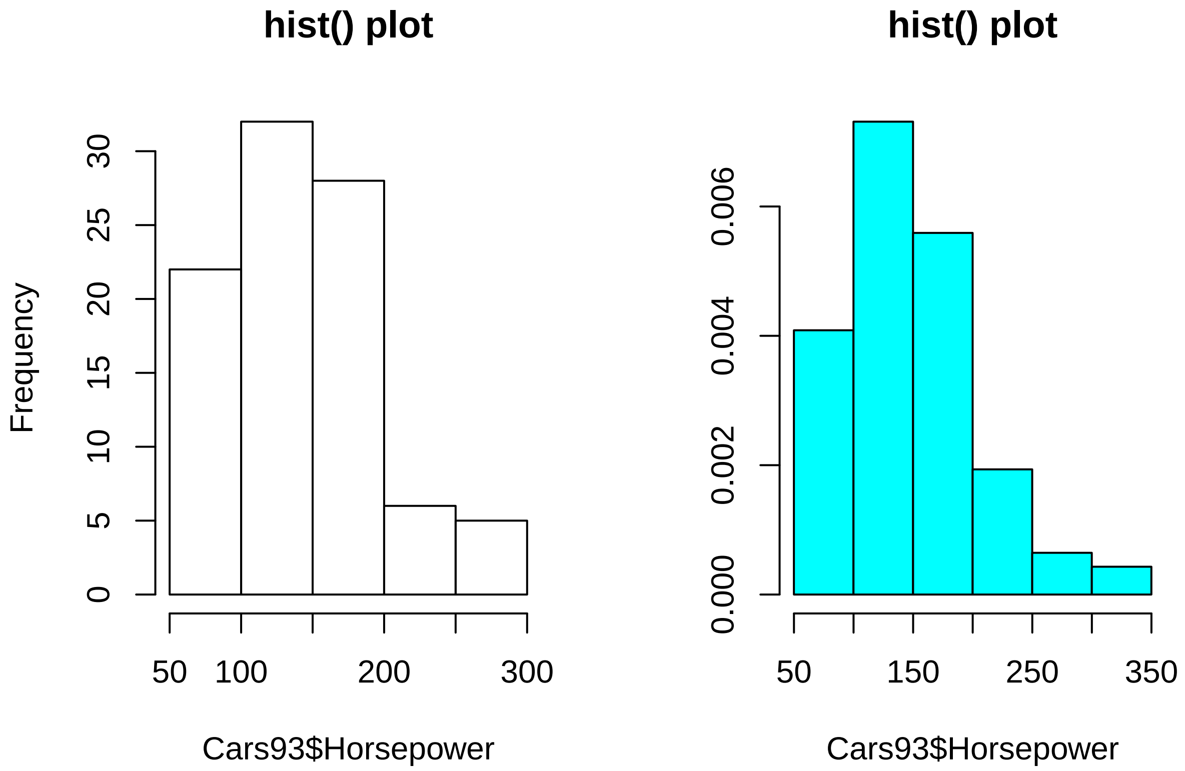
```
plot(Cars93$Max.Price, Cars93$Price,col="red",pch=17)
points(Cars93$Min.Price,Cars93$Price,col="blue",pch=16)
abline(a = 0, b = 1, lty = 2)
```



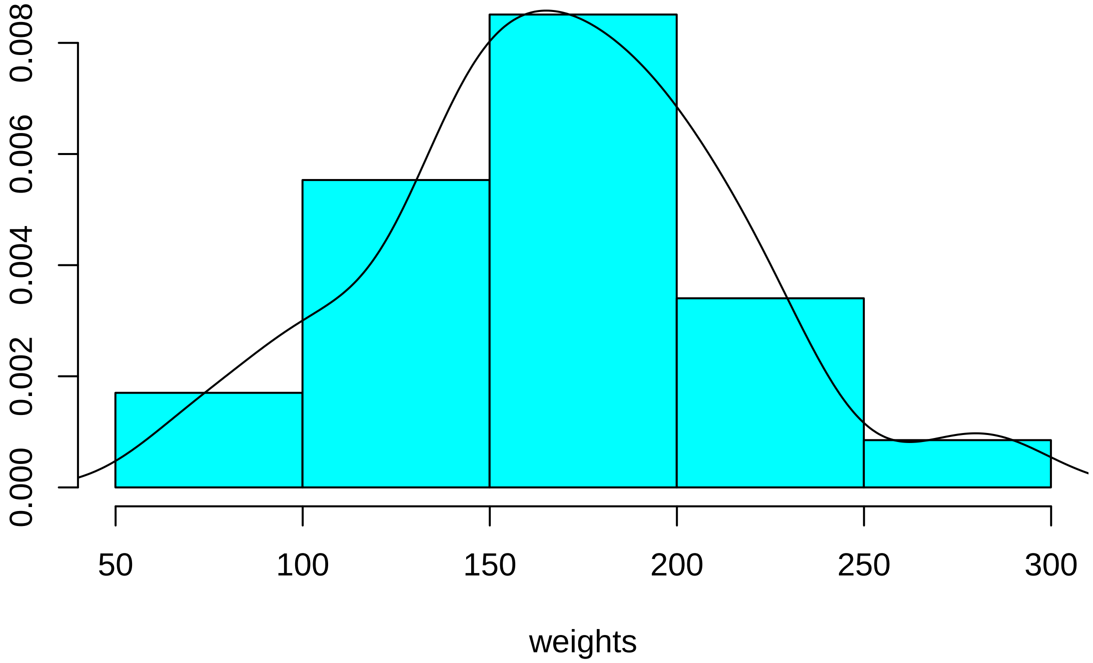
truehist() MASS

```
# Set up a side-by-side plot array
par(mfrow=c(1,2))
```

```
# Create a histogram of counts with hist()
hist(Cars93$Horsepower,main="hist() plot")

# Create a normalized histogram with truehist()
truehist(Cars93$Horsepower,main="hist() plot")
```



```
# Create index16, pointing to 16-week chicks
index16 <- which(ChickWeight$Time == 16)

# Get the 16-week chick weights
weights <- ChickWeight$weight[index16]

# Plot the normalized histogram
truehist(weights)

# Add the density curve to the histogram
lines(density(weights))
```



par()

```
par("bg") #
```

```
## [1] "transparent"
```

```
par("col") #
```

```
## [1] "black"
```

```
par("mar") #(bottom, left, top, right)
```

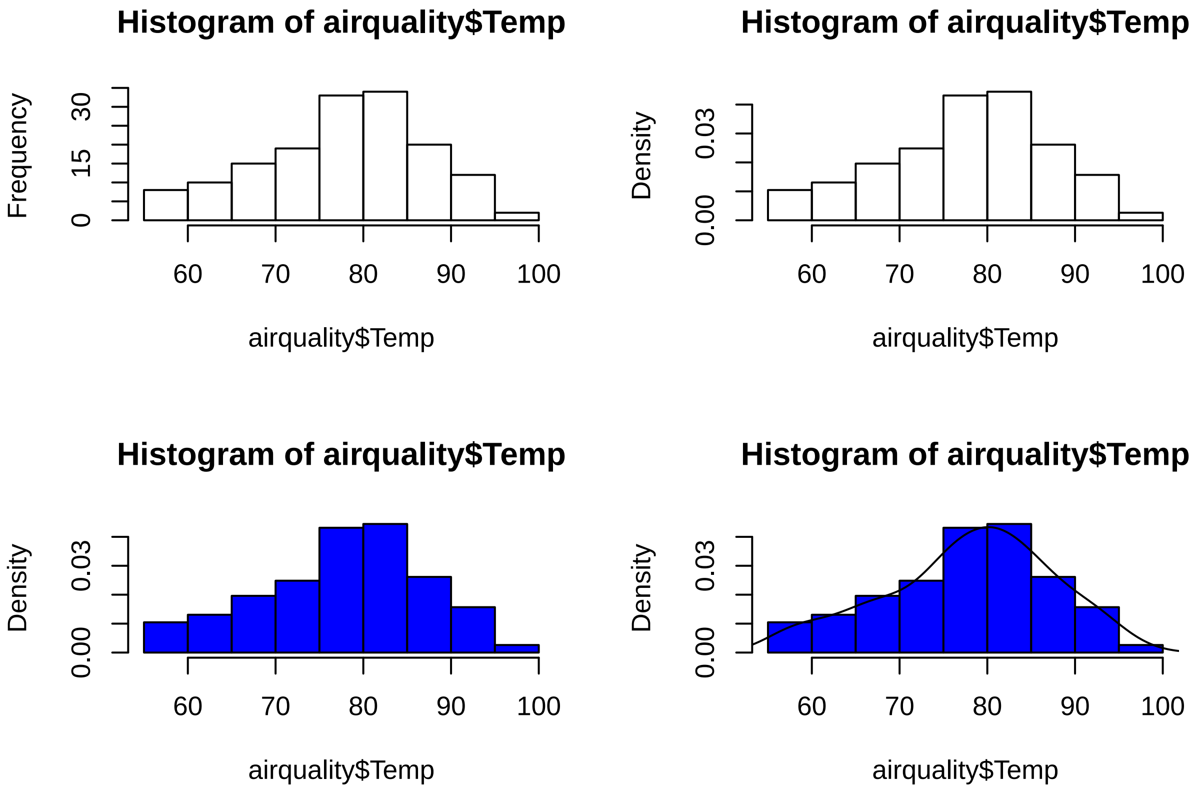
```
## [1] 5.1 4.1 4.1 2.1
```

```
par("mfrow") #
```

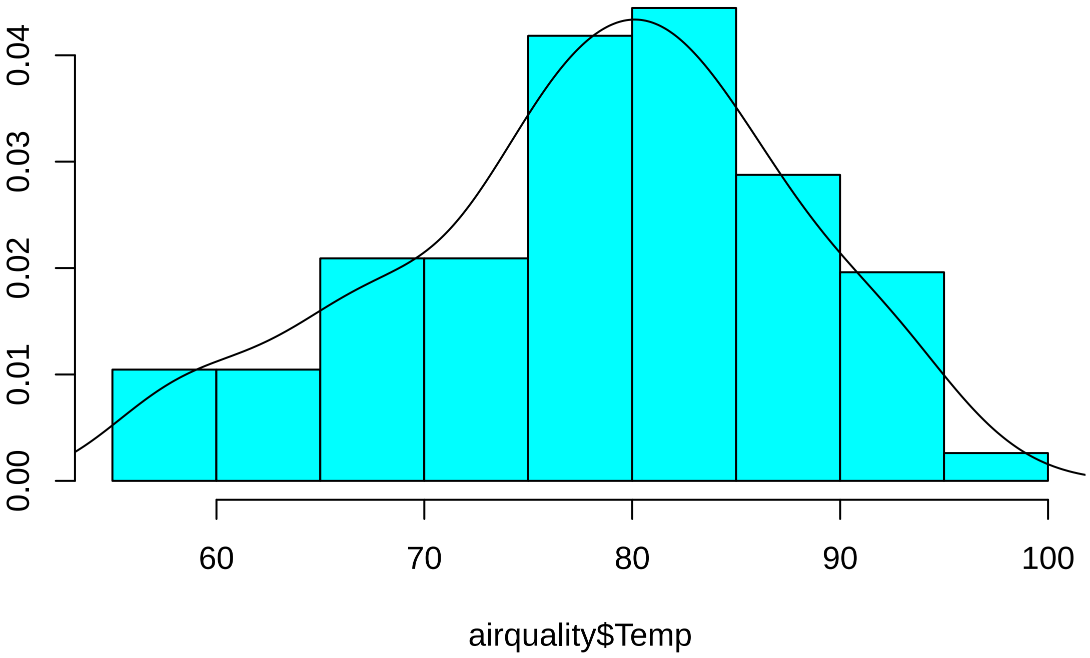
```
## [1] 1 1
```

```
?par
```

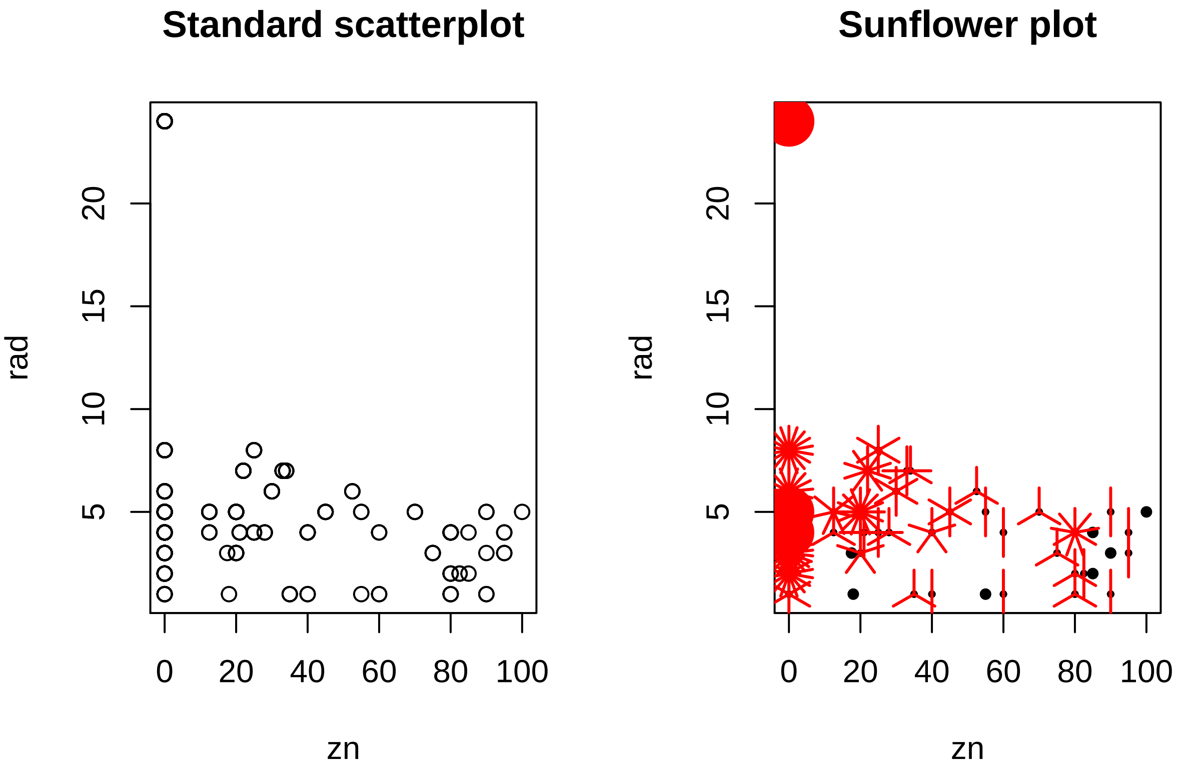
```
#
par(mfrow = c(2,2)) #  2
hist(airquality$Temp)
hist(airquality$Temp, freq = FALSE)
hist(airquality$Temp, freq = FALSE, col="blue")

hist(airquality$Temp, freq = FALSE, col="blue")
lines(density(airquality$Temp))
```



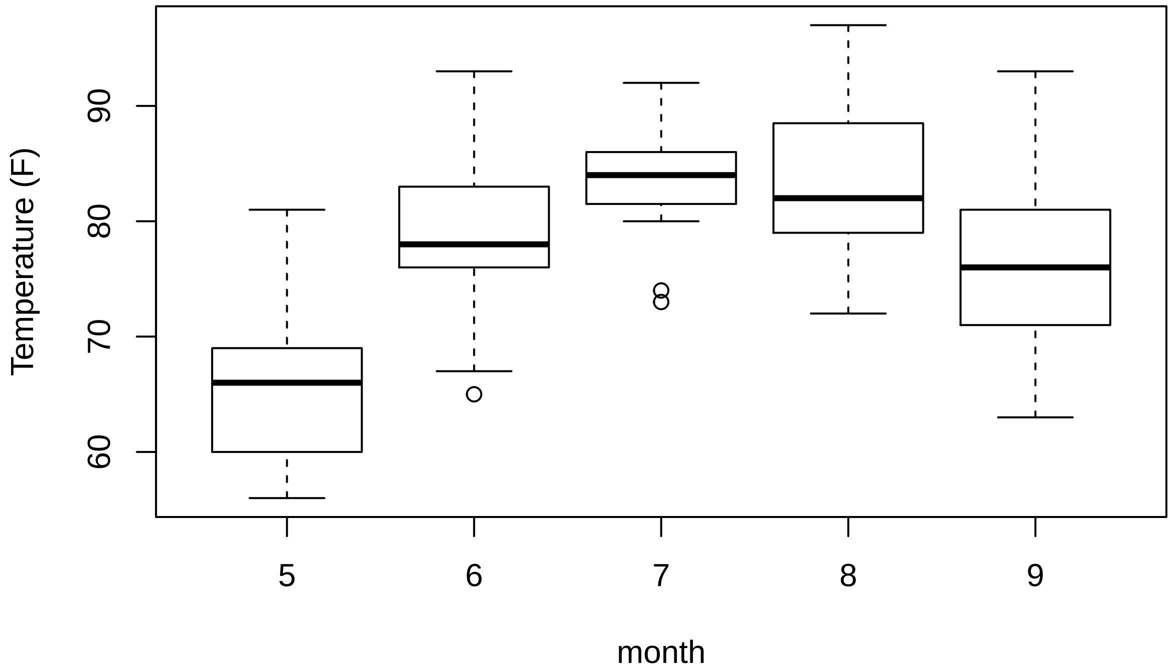
```r
par(mfrow = c(1,1))
truehist(airquality$Temp)
lines(density(airquality$Temp))
```



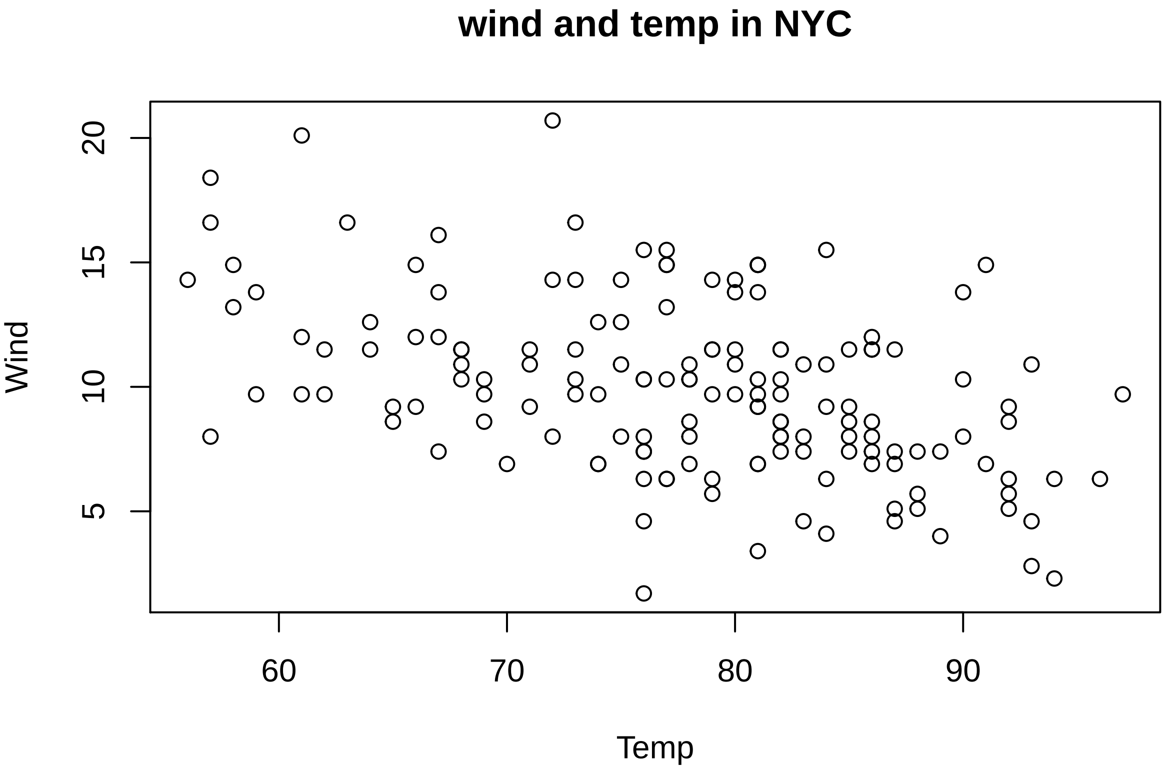
```
# Set up a side-by-side plot array
par(mfrow=c(1,2))

# Create the standard scatterplot
plot(rad~zn,data=Boston)

# Add the title
title("Standard scatterplot")

# Create the sunflowerplot
sunflowerplot(rad~zn,data=Boston)

# Add the title
title("Sunflower plot")
```



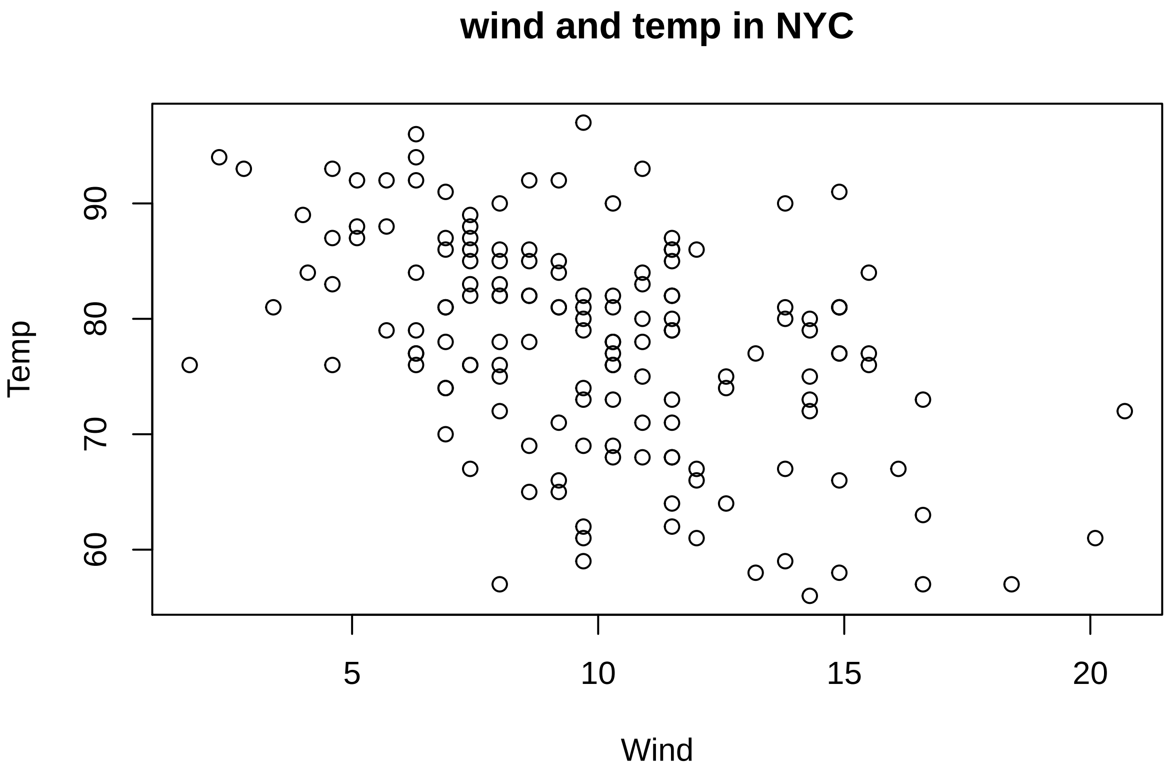
```
#
#month
boxplot(Temp~Month,airquality,xlab="month",
        ylab="Temperature (F)")
```



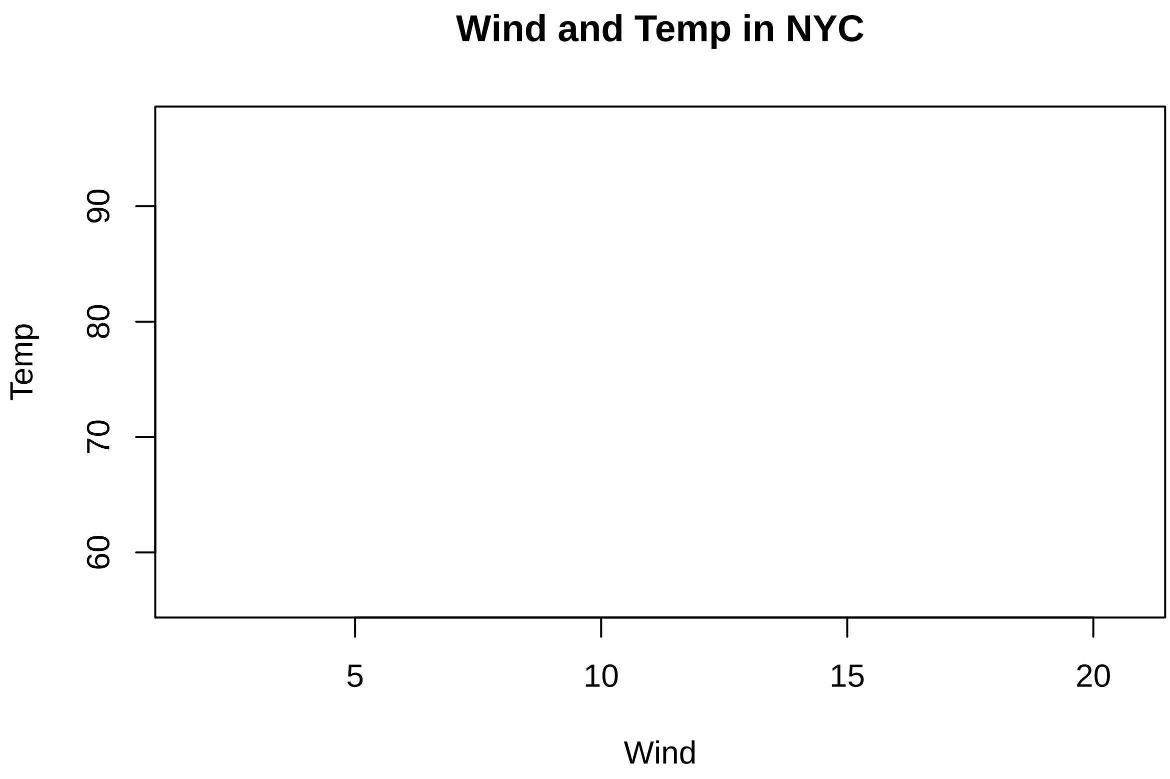
```
#with()
#
with(airquality, plot(Wind~Temp))
title(main="wind and temp in NYC") # title
```



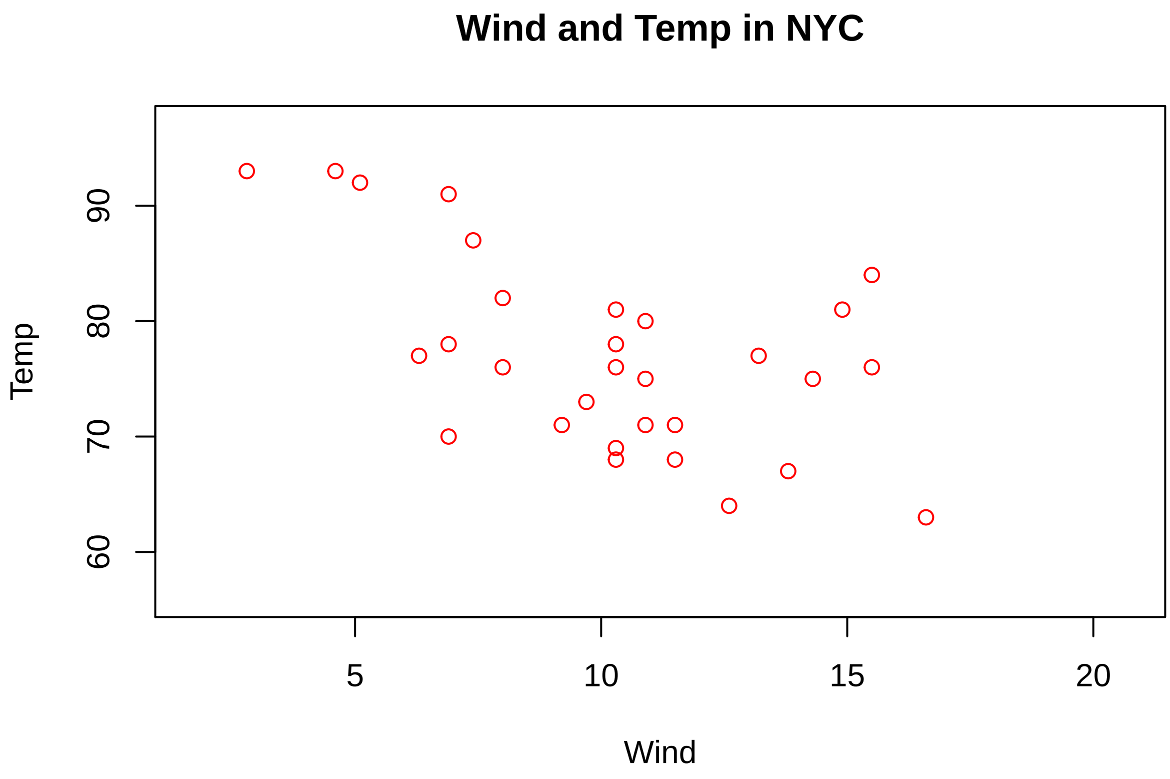
```
#
with(airquality, plot(Wind, Temp,
     main="wind and temp in NYC"))
```



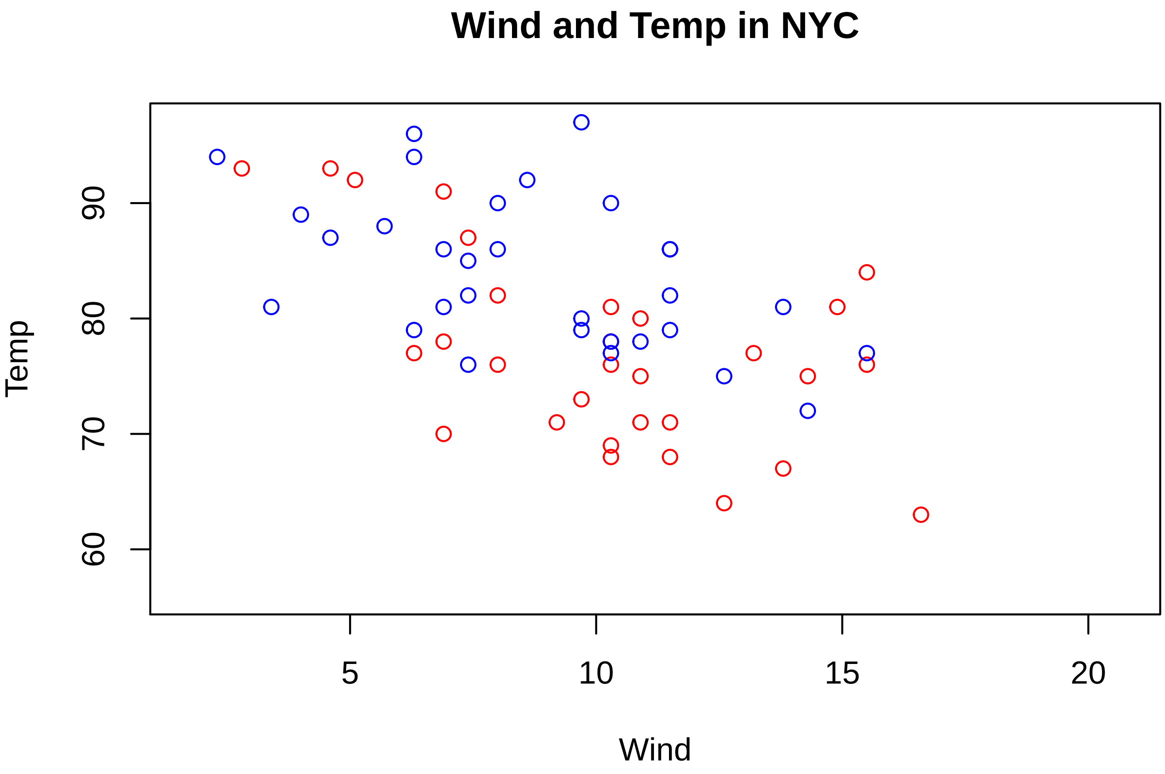
```
#   type="n"
with(airquality, plot(Wind, Temp,
                      main="Wind and Temp in NYC",
                      type="n"))
```



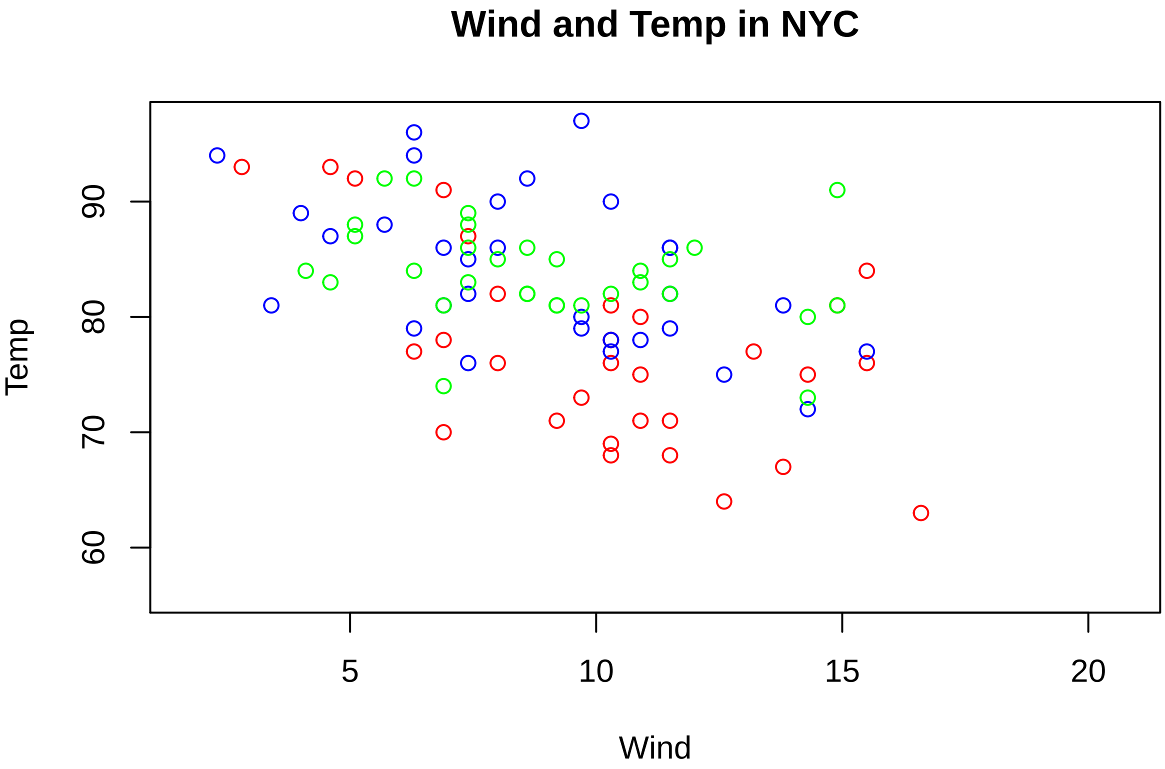
```
with(airquality, plot(Wind, Temp,
                      main="Wind and Temp in NYC",
                      type="n"))
with(subset(airquality, Month==9),
     points(Wind, Temp, col="red"))
```



```
with(airquality, plot(Wind, Temp,
                      main="Wind and Temp in NYC",
                      type="n"))
with(subset(airquality, Month==9),
     points(Wind, Temp, col="red"))
with(subset(airquality, Month==8),
     points(Wind, Temp, col="blue"))
```

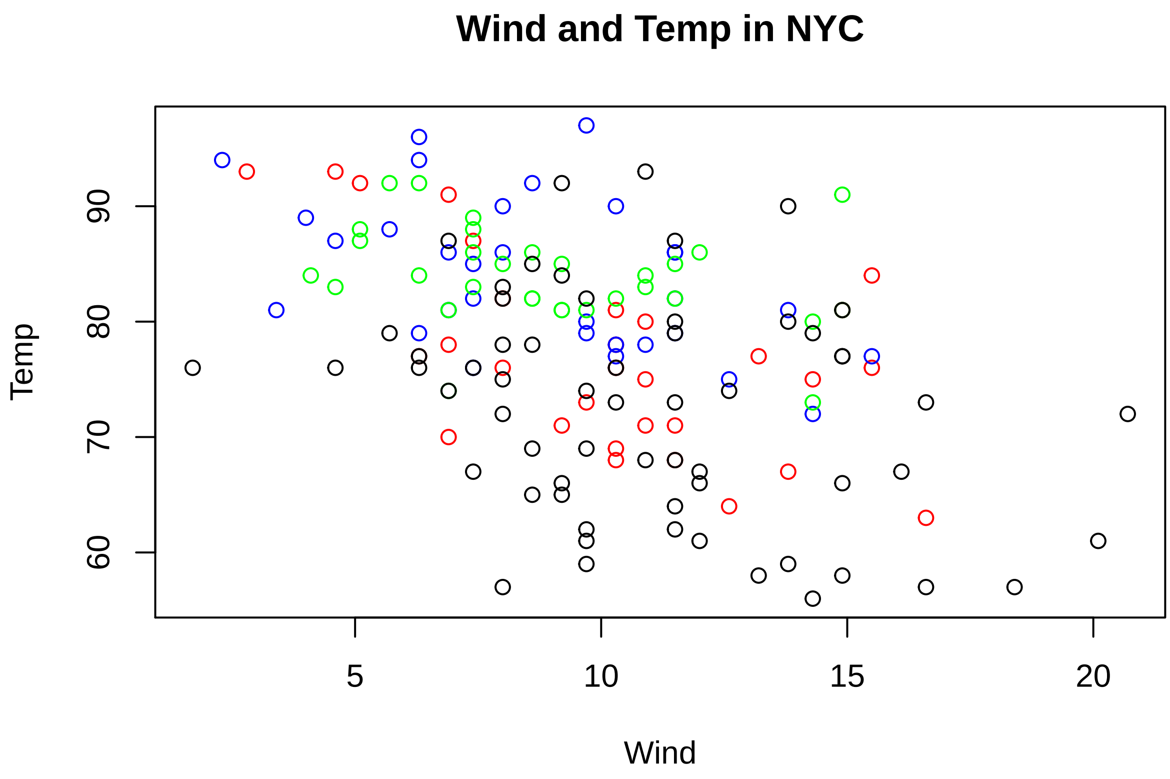


```
with(airquality, plot(Wind, Temp,
                      main="Wind and Temp in NYC",
                      type="n"))
with(subset(airquality, Month==9),
     points(Wind, Temp, col="red"))

with(subset(airquality, Month==8),
     points(Wind, Temp, col="blue"))

with(subset(airquality, Month==7),
     points(Wind, Temp, col="green"))
```

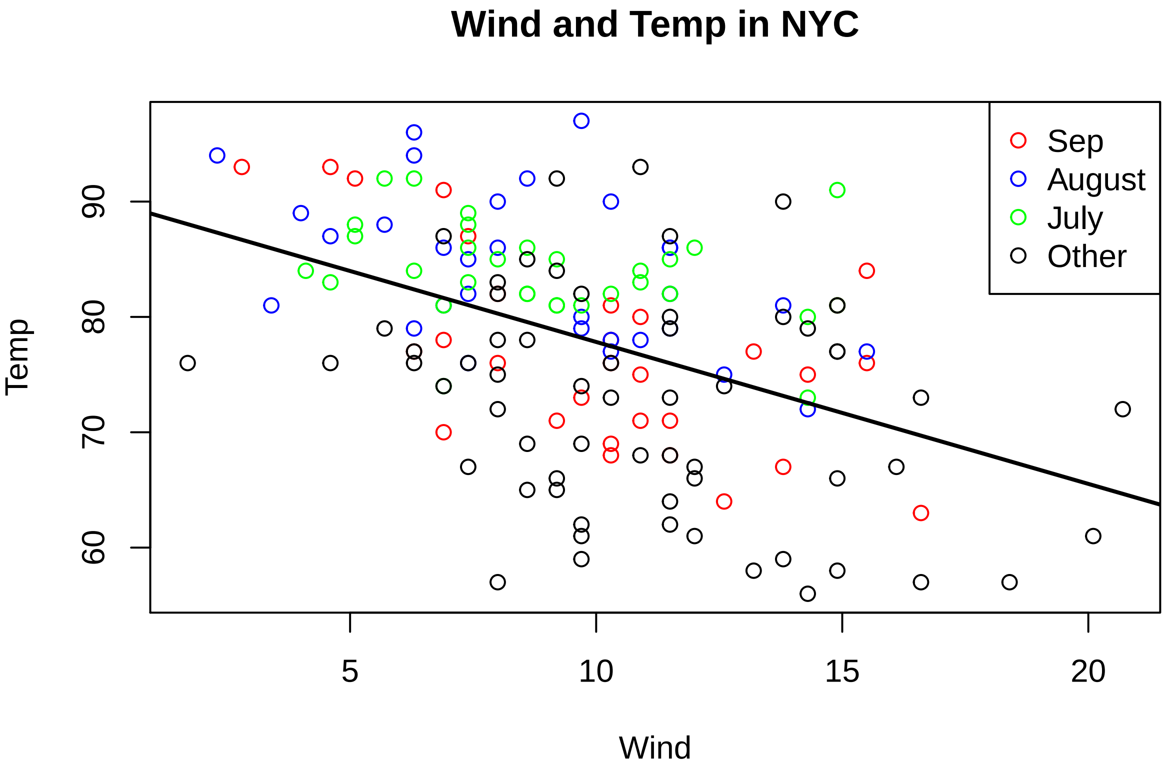


```
#
```

```
with(airquality, plot(Wind, Temp,
                      main="Wind and Temp in NYC",
                      type="n"))
with(subset(airquality, Month==9),
     points(Wind, Temp, col="red"))

with(subset(airquality, Month==8),
     points(Wind, Temp, col="blue"))

with(subset(airquality, Month==7),
     points(Wind, Temp, col="green"))

with(subset(airquality, Month %in% c(5,6)),
     points(Wind, Temp, col="black"))
```



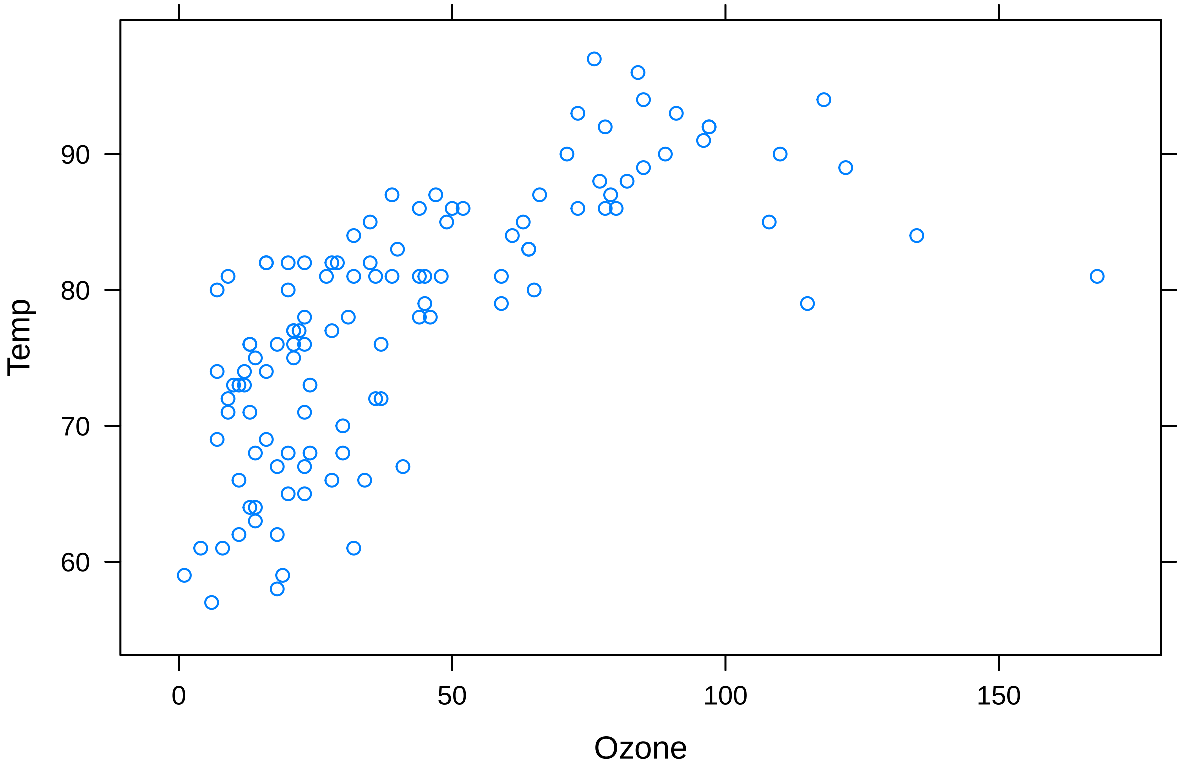
```
with(airquality, plot(Wind, Temp,
                      main="Wind and Temp in NYC",
                      type="n"))
with(subset(airquality, Month==9),
     points(Wind, Temp, col="red"))

with(subset(airquality, Month==8),
     points(Wind, Temp, col="blue"))

with(subset(airquality, Month==7),
     points(Wind, Temp, col="green"))

with(subset(airquality, Month %in% c(5,6)),
     points(Wind, Temp, col="black"))

fit<-lm(Temp~Wind, airquality) #
abline(fit,lwd=2) #

#
legend("topright", pch=1, #
       col=c("red","blue","green","black"),
       legend=c("Sep","August","July","Other"))
```



```
#
#
#
```

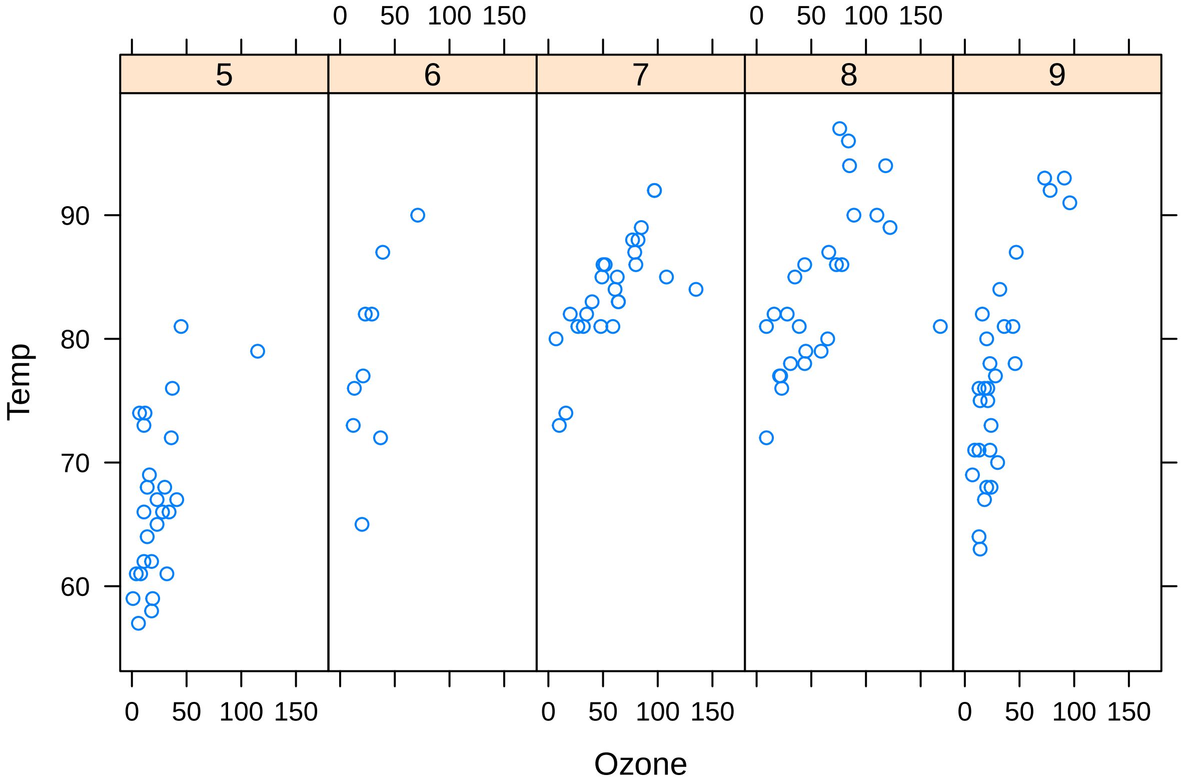
lattice

```
library(lattice) #   install.packages("lattice")
xyplot(Temp~Ozone, data=airquality) #  ,
```



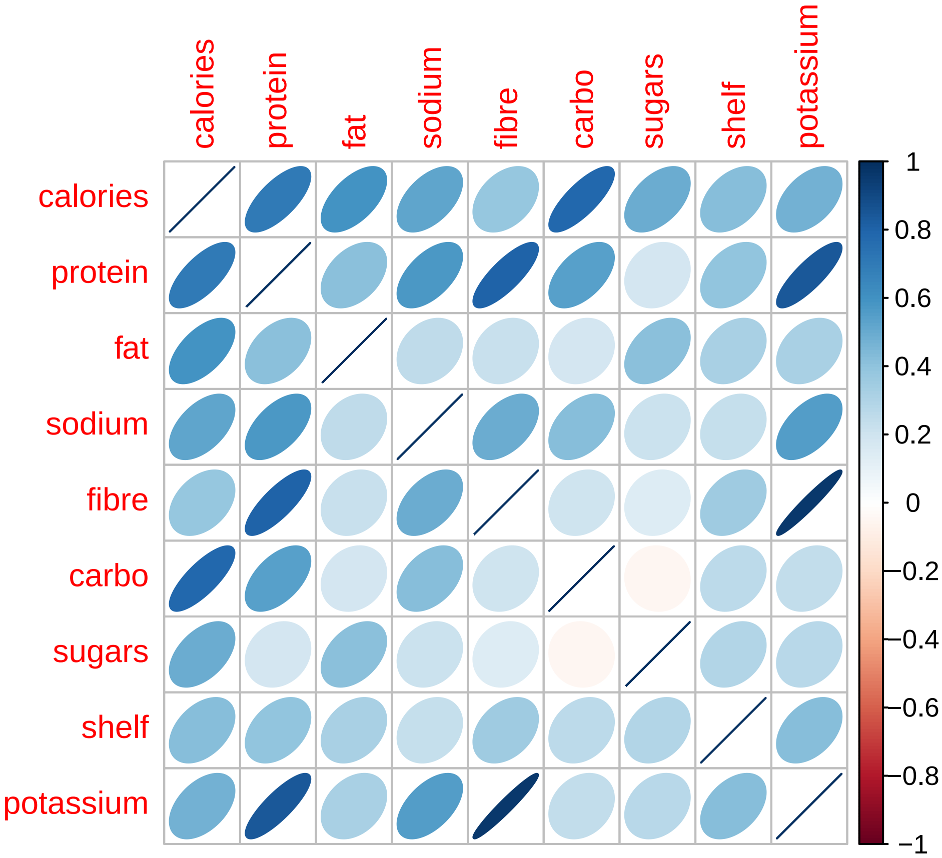
```
airquality$Month<-factor(airquality$Month)
```

```
#
xyplot(Temp~Ozone|Month, data=airquality,
       layout=c(5,1)) # 1 5
```



1, 0

```r
# Load the corrplot library for the corrplot() function
library(corrplot)

# Extract the numerical variables from UScereal
numericalVars <- UScereal[, 2:10]

# Compute the correlation matrix for these variables
corrMat <- cor(numericalVars)

# Generate the correlation ellipse plot
corrplot(corrMat, method = "ellipse")
```



data visulization in datacamp

Chapter 7

7.1 ggplot2

7.2

7.3

7.4

7.5

7.6

Chapter 8

8.1

Chapter 9

9.1

9.1.1

9.1.2

9.1.3

9.2

9.2.1

9.2.2

9.2.3

Chapter 10

10.1 R

10.1.1

10.1.2

10.1.3

10.1.4

10.2

10.2.1

10.2.2 R

10.2.3

10.2.4

Chapter 11

11.1

11.2

Chapter 12

dplyr