4_R프로그램 기초 - R 그래픽스

Jinseog Kim
Dongguk University
jskim1986@gmail.com

2018-03-06

Contents

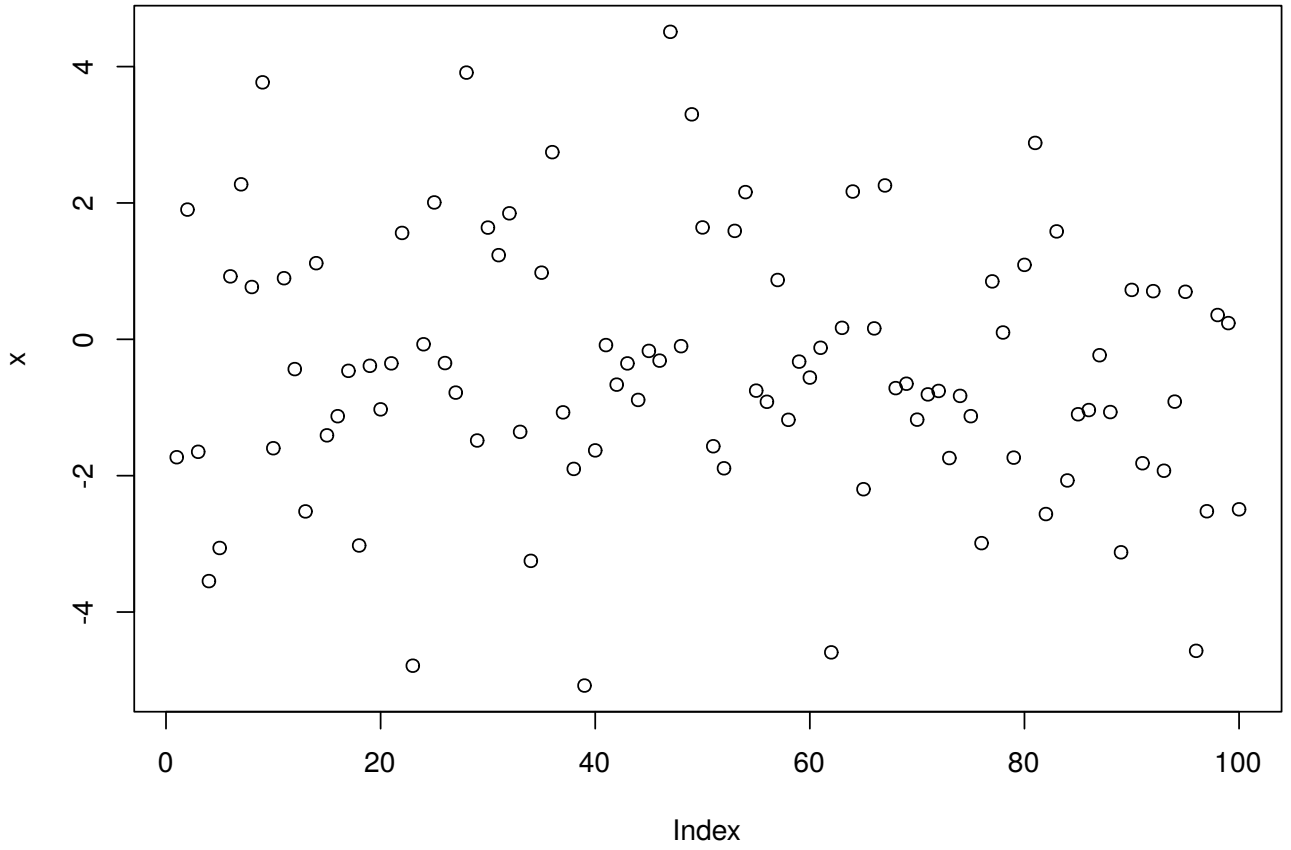
1 R 그래프를 위한 기본함수

1. 고수준 함수: plot(), barplot(), boxplot(), hist(), pie(), persp()
2. 저수준함수
    1. 점그리기: points()
    2. 선그리기: lines(), abline(), arrows()
    3. 문자출력: text()
    4. 도형: rect(), ploygon()
    5. 좌표축: axis()
    6. 격자표현: grid()

1.1 plot() 함수

- plot

```
x <- rnorm(100, sd=2); y <- 0.3 + 2*x + rnorm(100, sd=1)
plot(x)
```



 심볼 모양 : plot(, pch=)

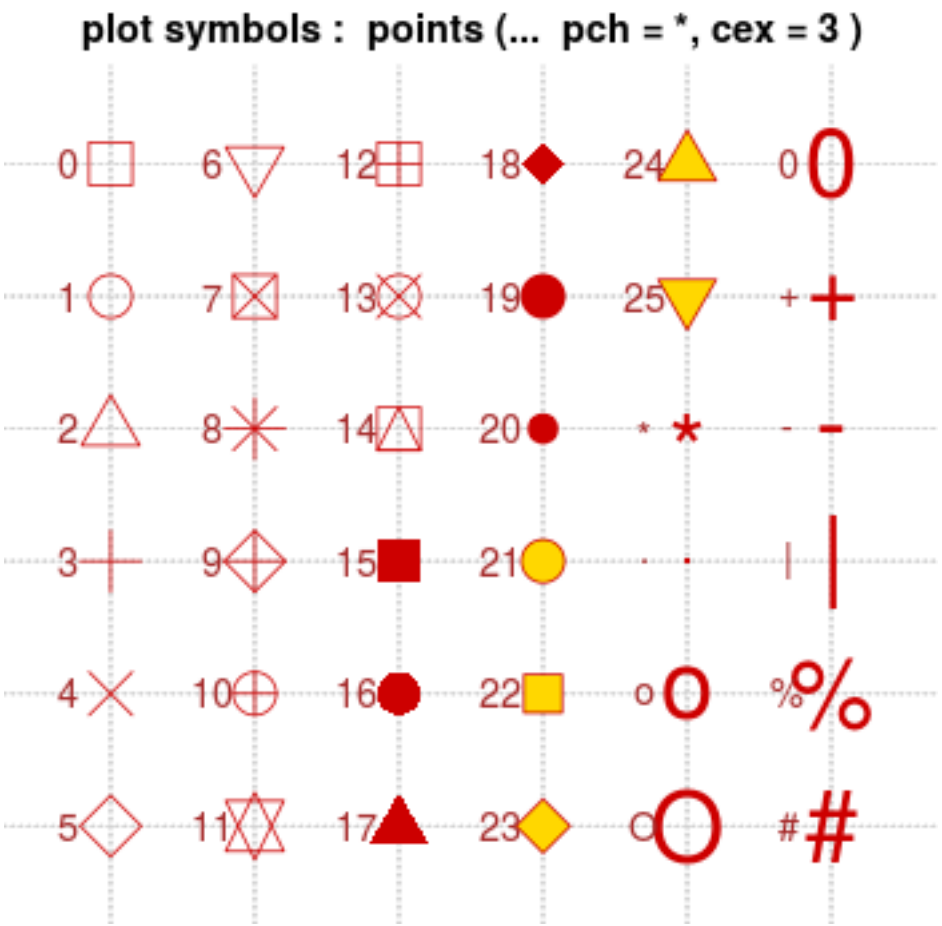
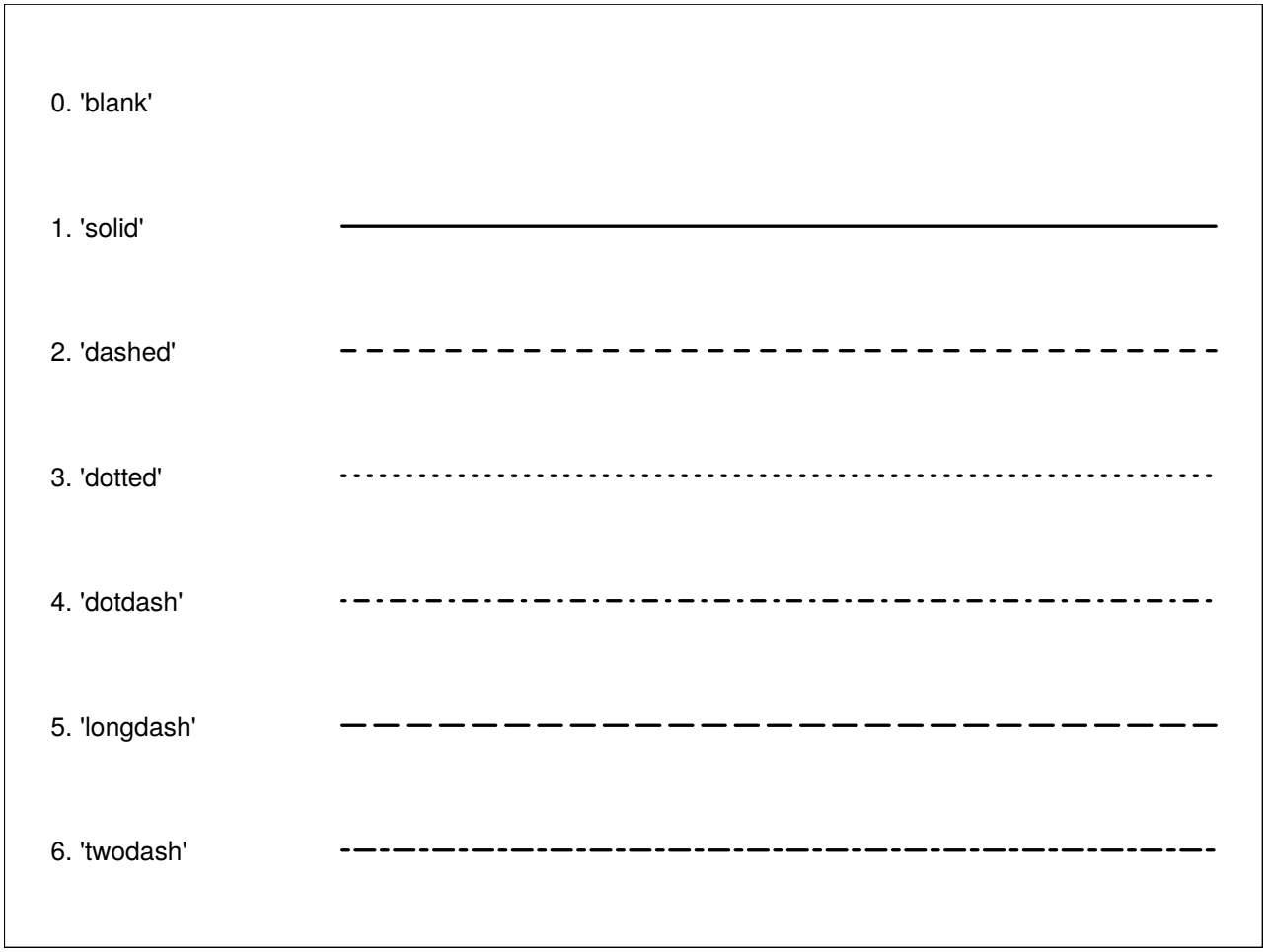


Figure 1: 심볼 모양

 선의 종류 : * plot(..., lty=)



 산점도 + 텍스트 추가

 Animals data: Average brain and body weights for 28 species of land animals.

```
library(MASS)
head(Animals)
```

```
##                    body brain
## Mountain beaver     1.35   8.1
## Cow               465.00 423.0
## Grey wolf          36.33 119.5
## Goat               27.66 115.0
## Guinea pig          1.04   5.5
## Dipliodocus     11700.00  50.0
```

```r
plot(x=Animals$body , y=Animals$brain, pch=16, col="blue",
 xlab="Body weight (kg)", ylab="Brain weight (g)", xlim=c(0,250), ylim=c(0, 1400))
text(x=Animals$body, y=Animals$brain, labels=row.names(Animals), pos=4)
```

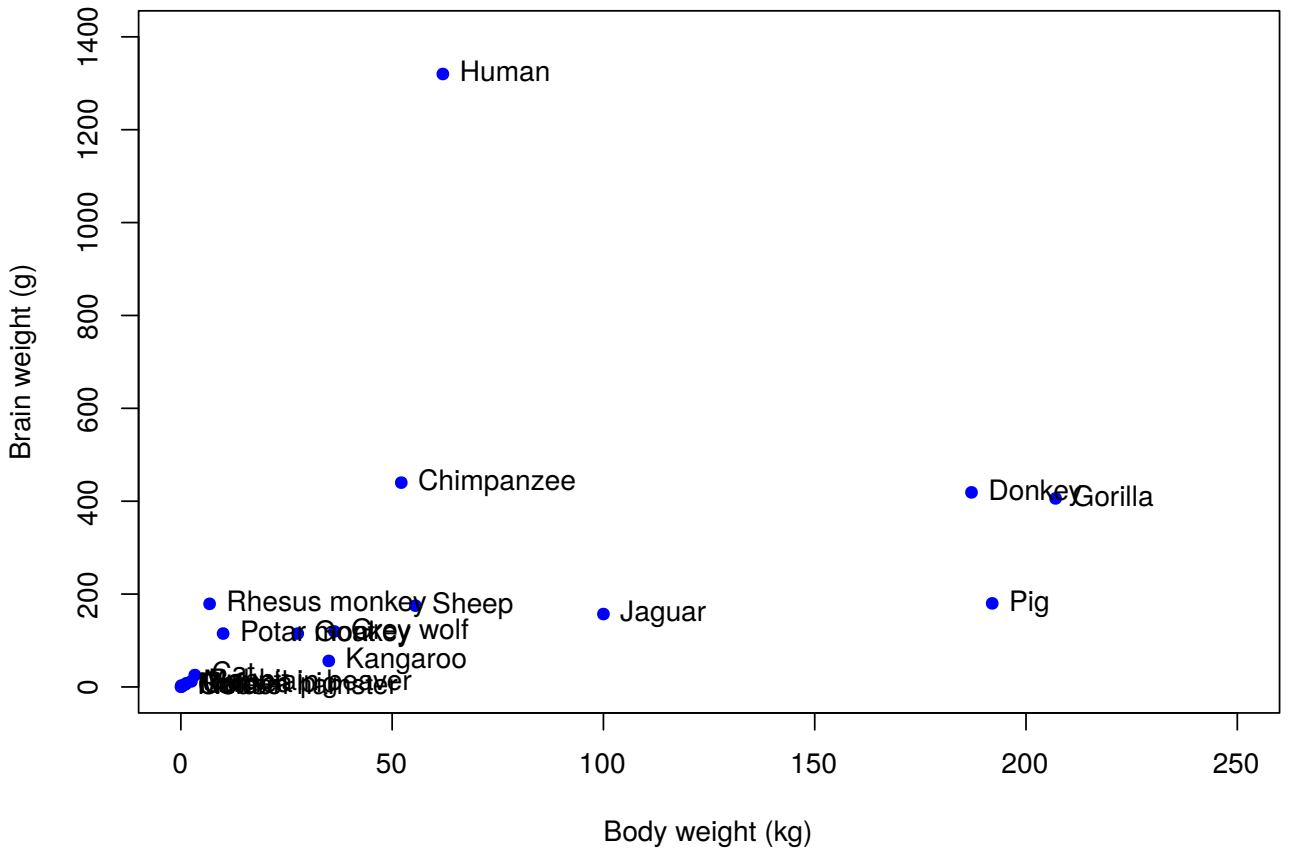
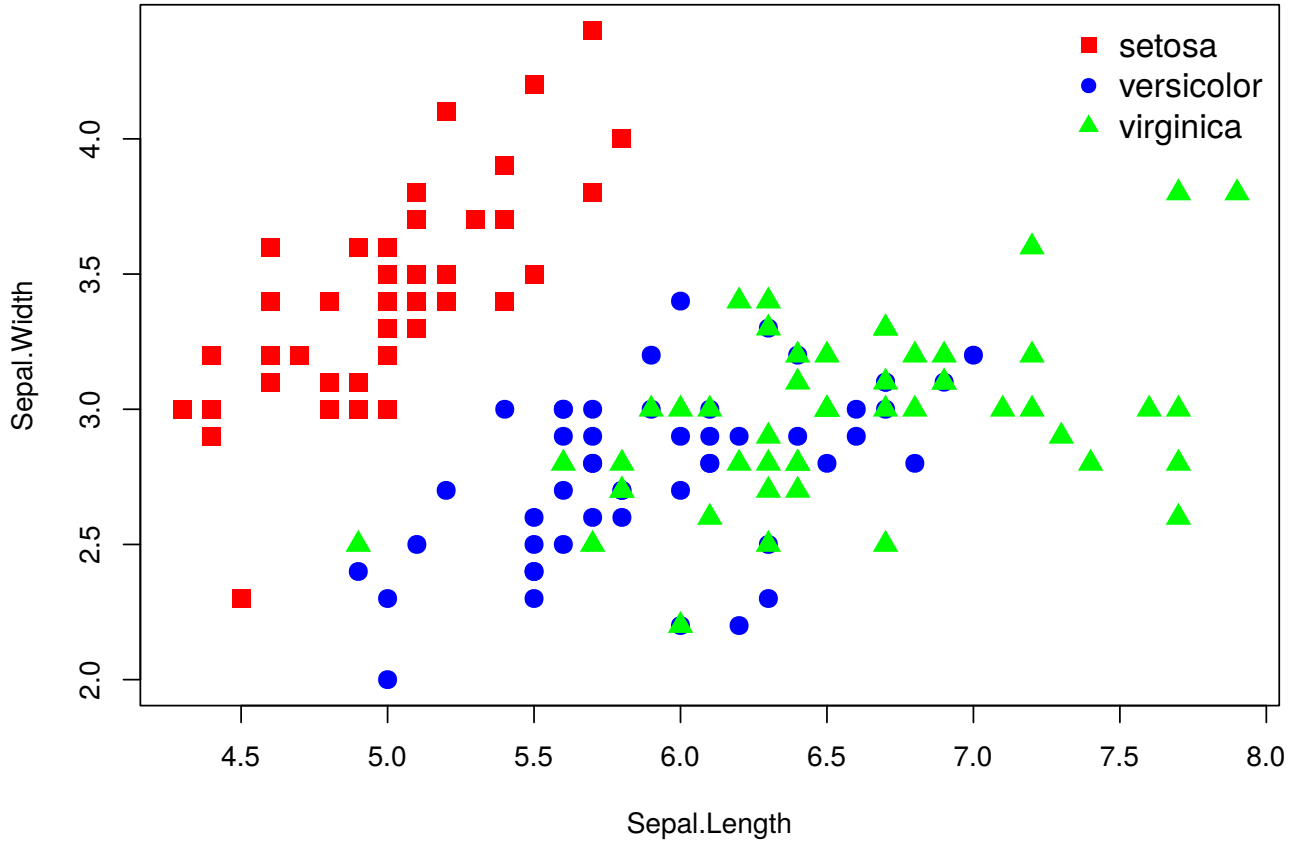


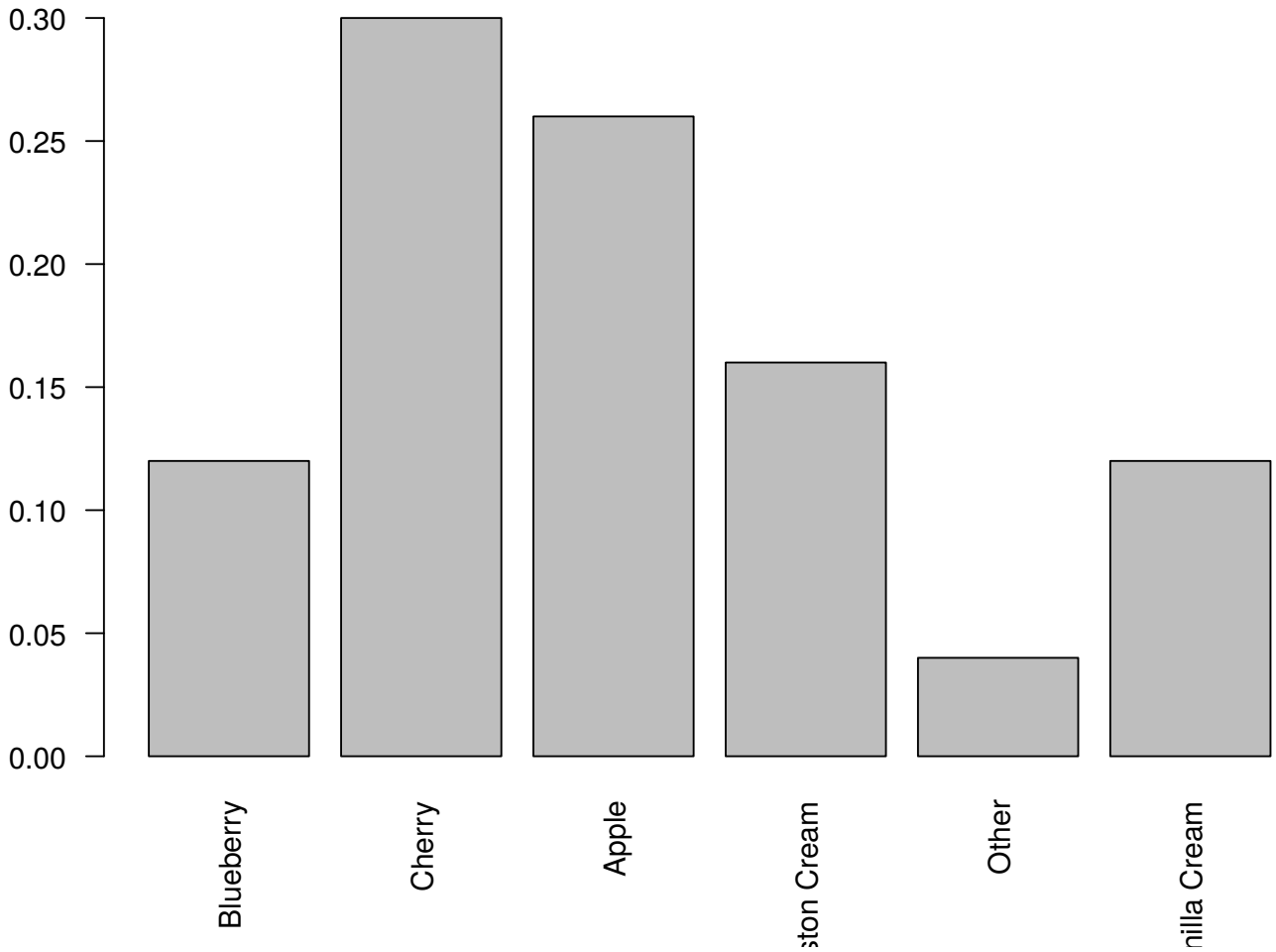
 그림 범례(legend) 표시

```
plot(~Sepal.Length+Sepal.Width, data=iris, pch=rep(15:17, each=50),
     col=c("red", "blue", "green")[iris$Species], cex=1.5)
legend("topright", legend = levels(iris$Species), pch=15:17,
       col=c("red", "blue", "green"), cex=1.2, bty="n")
```

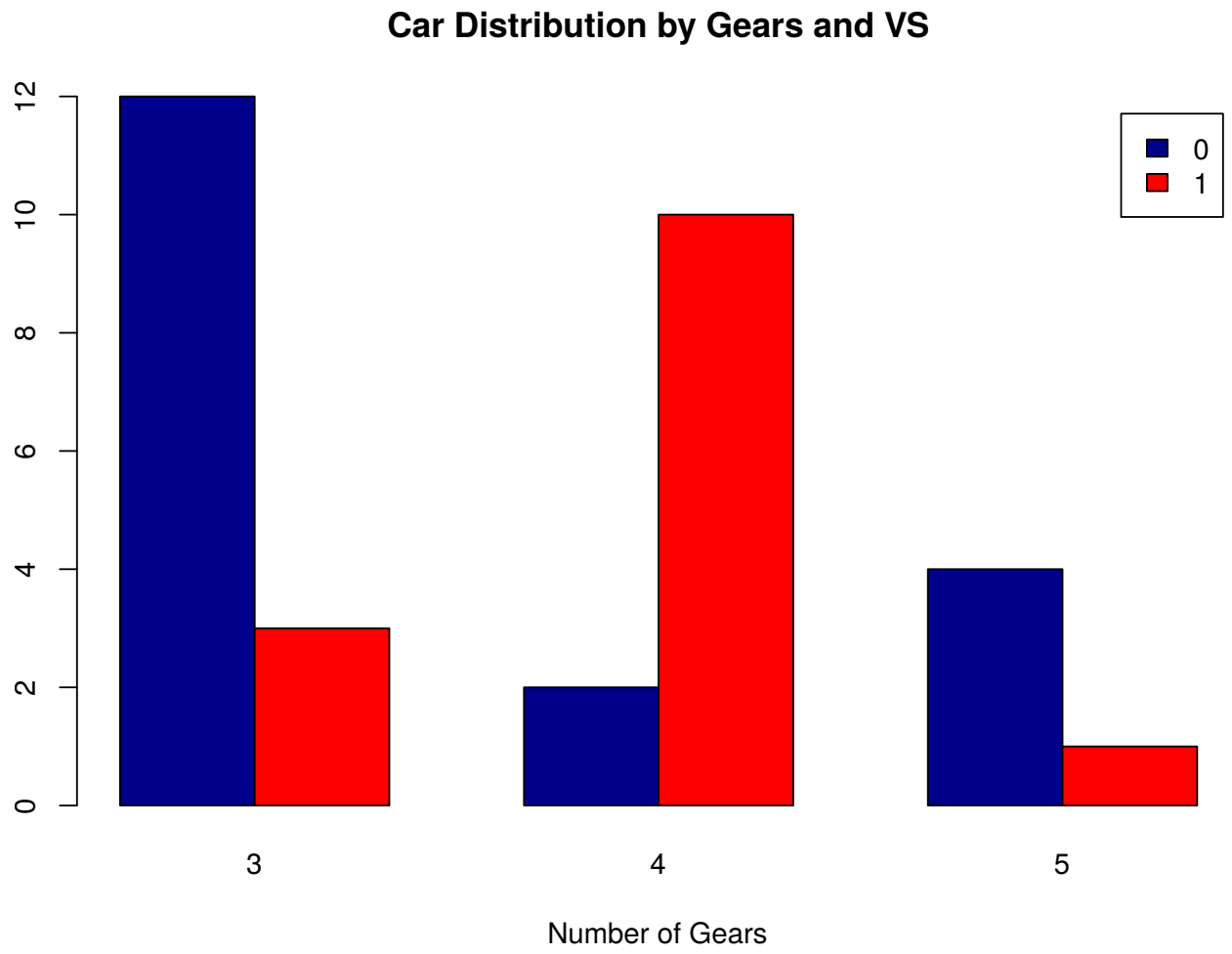


1.2 막대그래프: barplot 함수

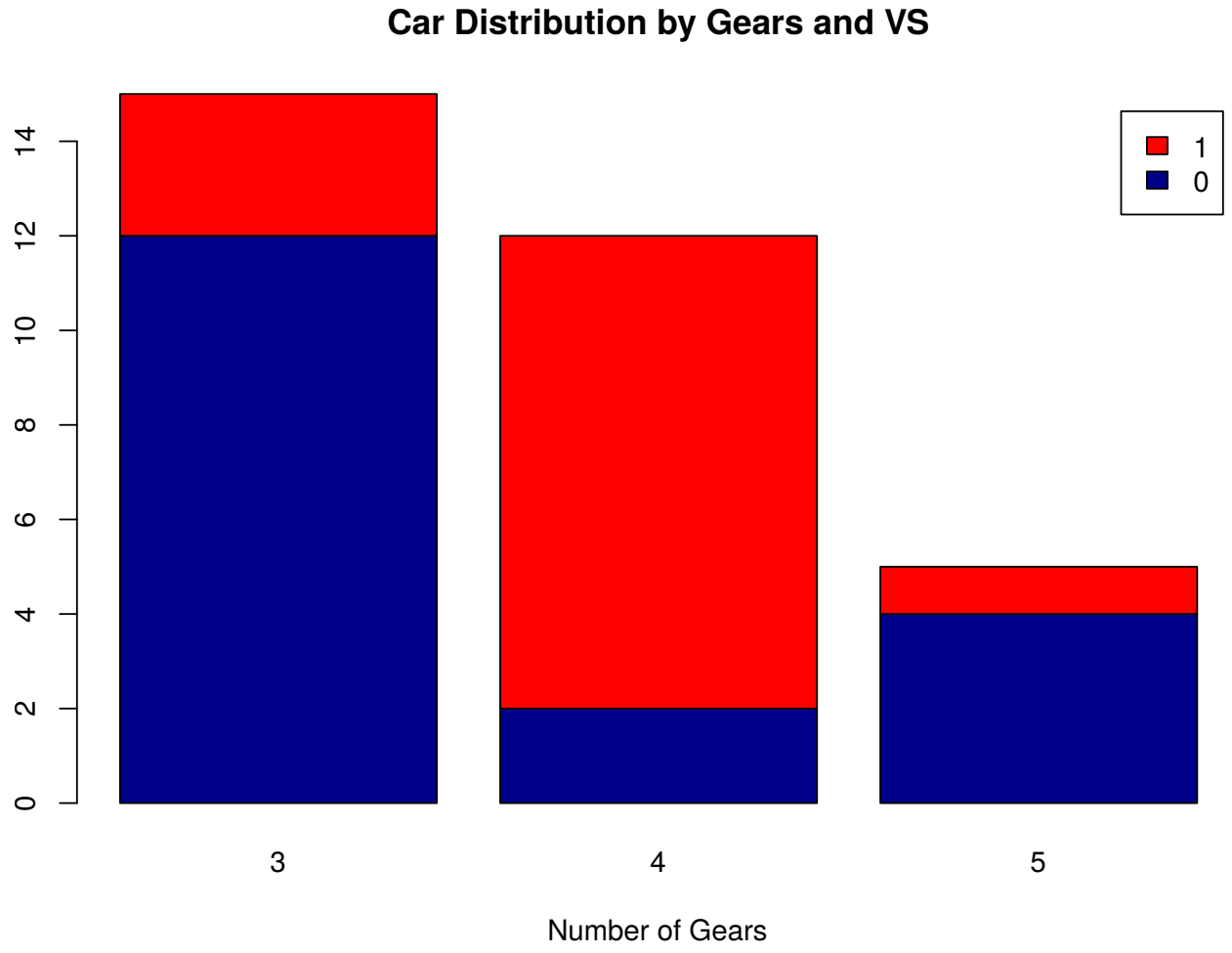
```
pie.sales <- c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
names(pie.sales) <- c("Blueberry", "Cherry", "Apple", "Boston Cream",
"Other", "Vanilla Cream")
barplot(pie.sales, las=2) #las=2: x-tick
```


0.30
0.25
0.20
0.15
0.10
0.05
0.00
Blueberry
Cherry
Apple
ston Cream
Other
nilla Cream

```r
counts <- table(mtcars$vs, mtcars$gear)
barplot(counts, main="Car Distribution by Gears and VS",
  xlab="Number of Gears", col=c("darkblue","red"),
    legend = rownames(counts), beside=TRUE)
```

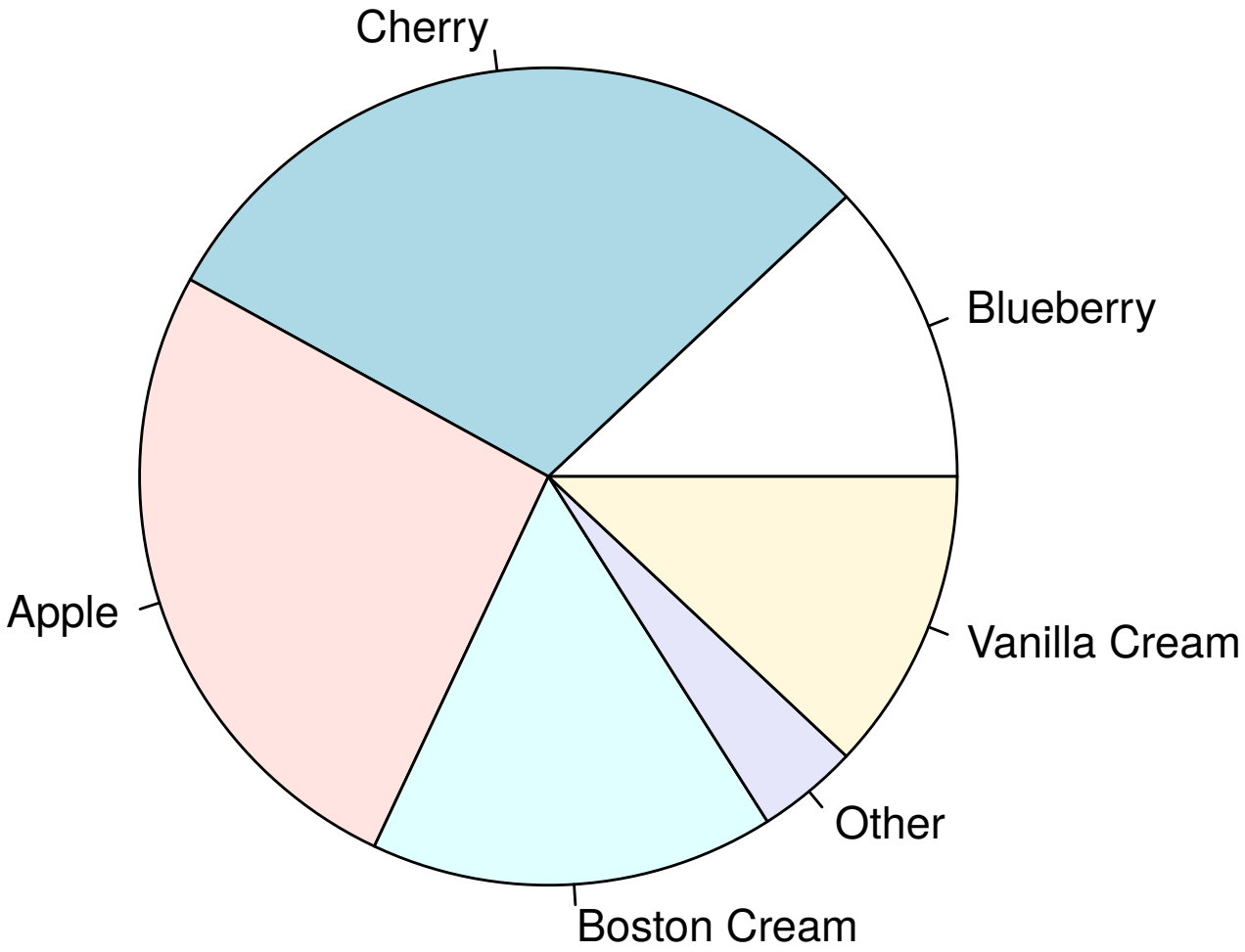


```r
barplot(counts, main="Car Distribution by Gears and VS",
  xlab="Number of Gears", col=c("darkblue","red"),
    legend = rownames(counts))
```



1.3 원그래프: pie 함수

```
pie(pie.sales)
```



1.4 점도표 : dotchart 함수

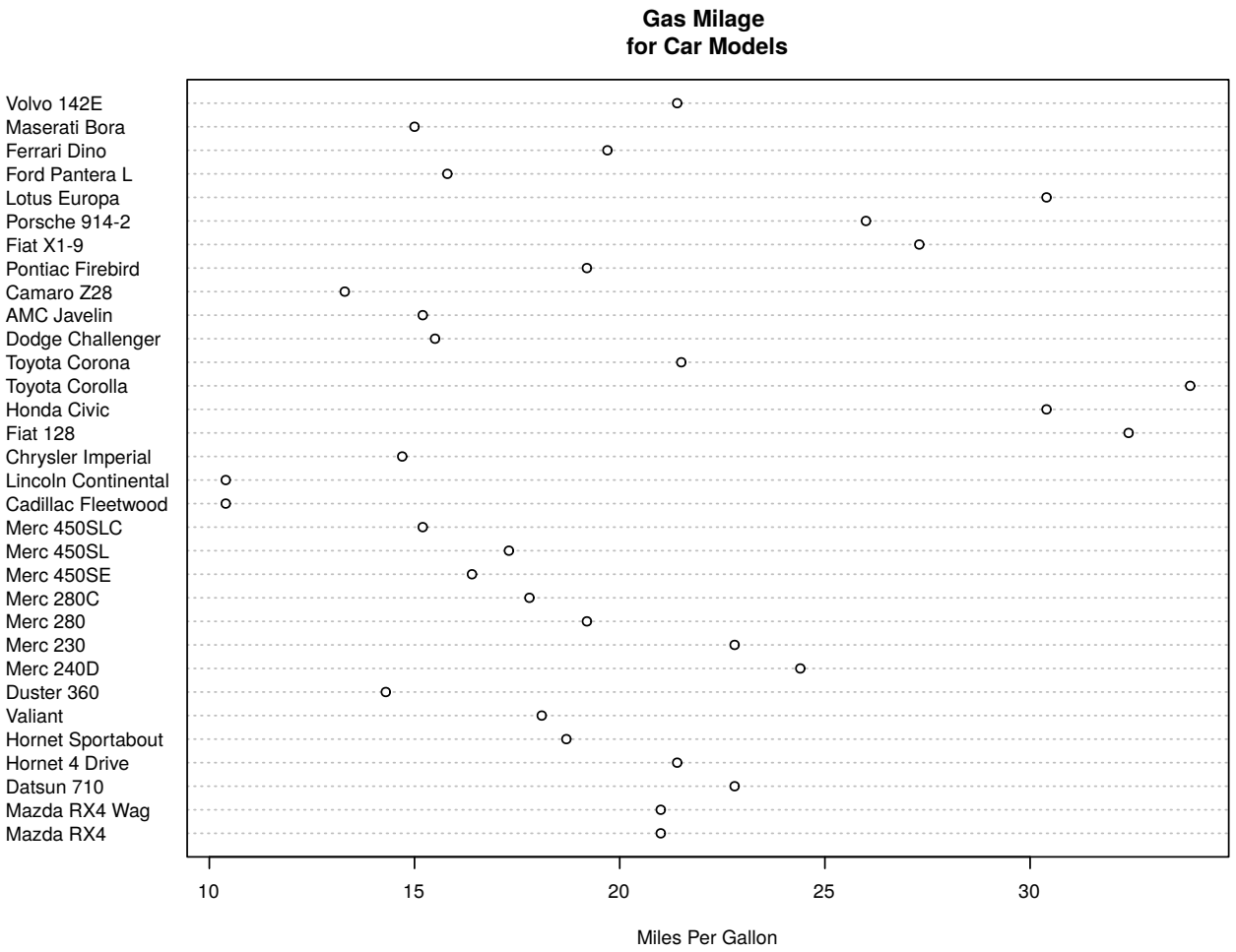
 mtcar data

```
head(mtcars)
```

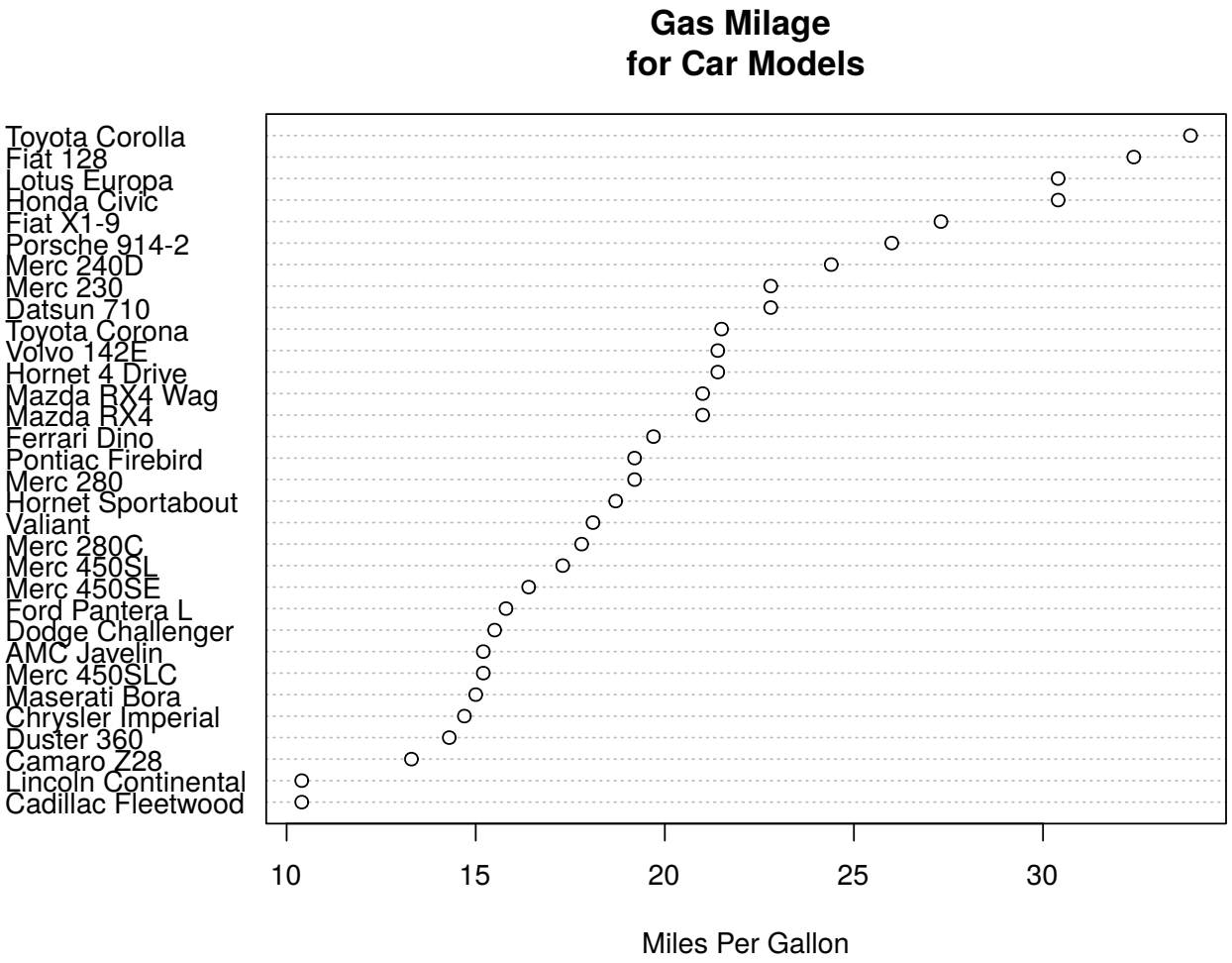
```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 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
```

 dotchart

```
dotchart(mtcars$mpg,labels=row.names(mtcars),
  cex=0.7,
  main="Gas Milage \nfor Car Models",
  xlab="Miles Per Gallon")
```


Gas Milage
for Car Models
Volvo 142E
Maserati Bora
Ferrari Dino
Ford Pantera L
Lotus Europa
Porsche 914-2
Fiat X1-9
Pontiac Firebird
Camaro Z28
AMC Javelin
Dodge Challenger
Toyota Corona
Toyota Corolla
Honda Civic
Fiat 128
Chrysler Imperial
Lincoln Continental
Cadillac Fleetwood
Merc 450SLC
Merc 450SL
Merc 450SE
Merc 280C
Merc 280
Merc 230
Merc 240D
Duster 360
Valiant
Hornet Sportabout
Hornet 4 Drive
Datsun 710
Mazda RX4 Wag
Mazda RX4
10
15
20
25
30
Miles Per Gallon

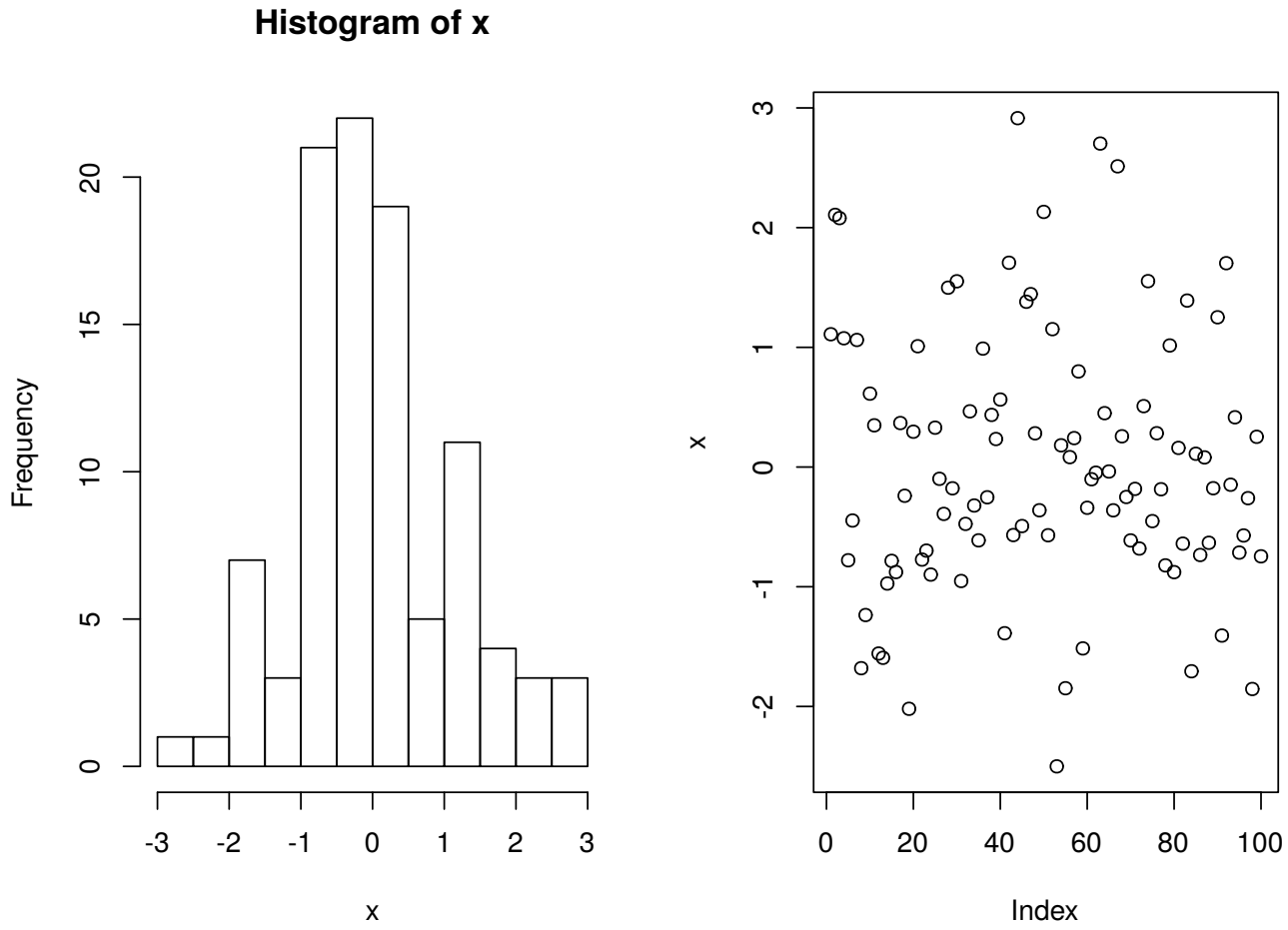
```r
idx <- order(mtcars$mpg)
dotchart(mtcars$mpg[idx],labels=row.names(mtcars)[idx], cex=1,
  main="Gas Milage \nfor Car Models", xlab="Miles Per Gallon")
```



1.5 par() 함수를 이용한 그래프 옵션 지정

1. 한 화면에 여러개의 그래프 표시 : mfrow

```
x<-rnorm(100)
par(mfrow=c(1,2))
hist(x); plot(x)
```


Histogram of x
Frequency
0
5
10
15
20
-3
-2
-1
0
1
2
3
x
x
-2
-1
0
1
2
3
0
20
40
60
80
100
Index

1.6 히스토그램(Histogram)

 cars 데이터: 자동차의 속도(speed)와 정지시까지 거리(dist)

```
data(cars)
head(cars, 3)
```

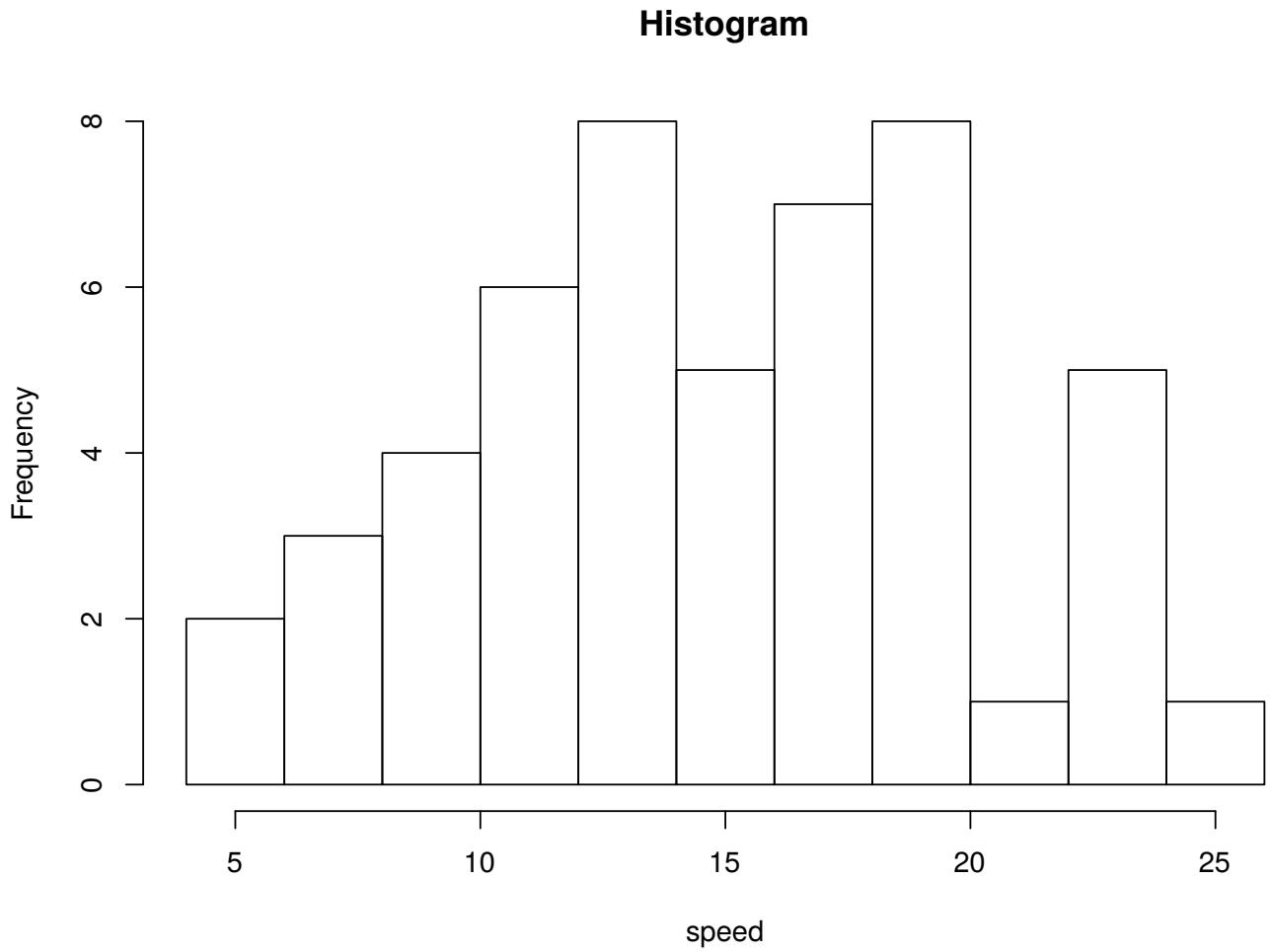
```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
```

```
tail(cars, 3)
```

```
##    speed dist
## 48    24   93
## 49    24  120
## 50    25   85
```

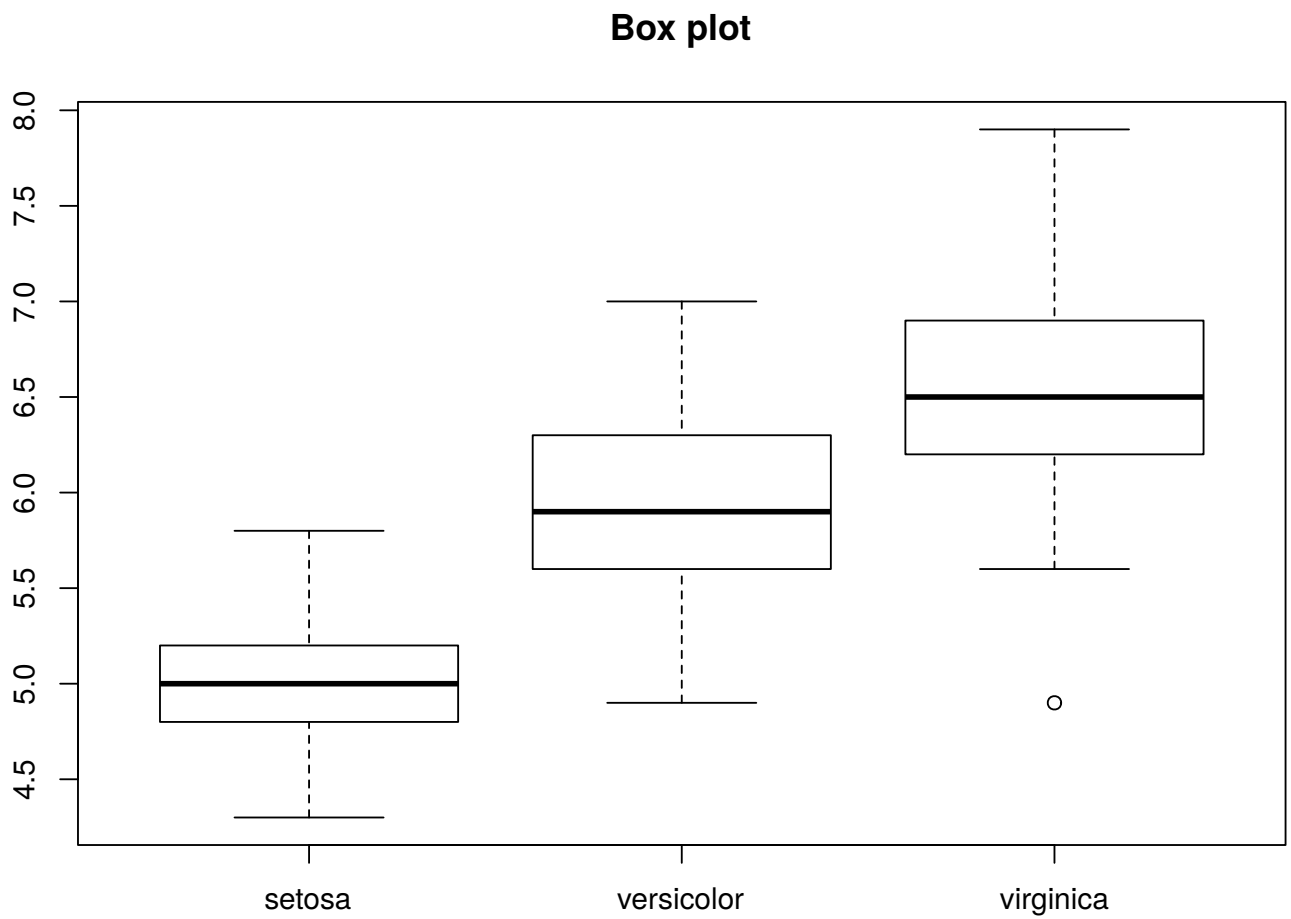
 hist를 이용한 히스토그램(Histogram)

```r
hist(cars$speed, nclass=8, main="Histogram", xlab="speed")
```



1.7 상자그림(box plot): boxplot

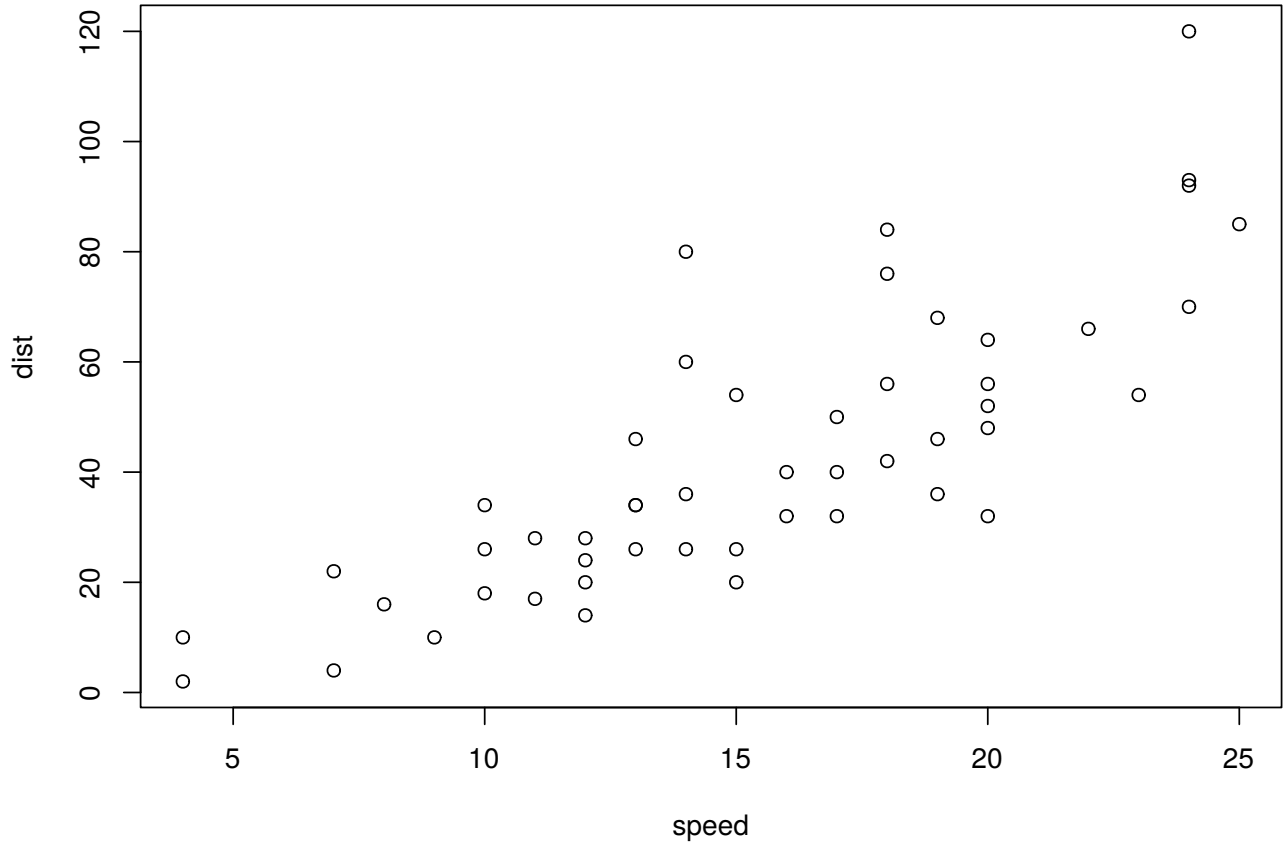
```
boxplot(Sepal.Length~Species, data=iris, main="Box plot")
```



1.8 산점도(scatter plot)

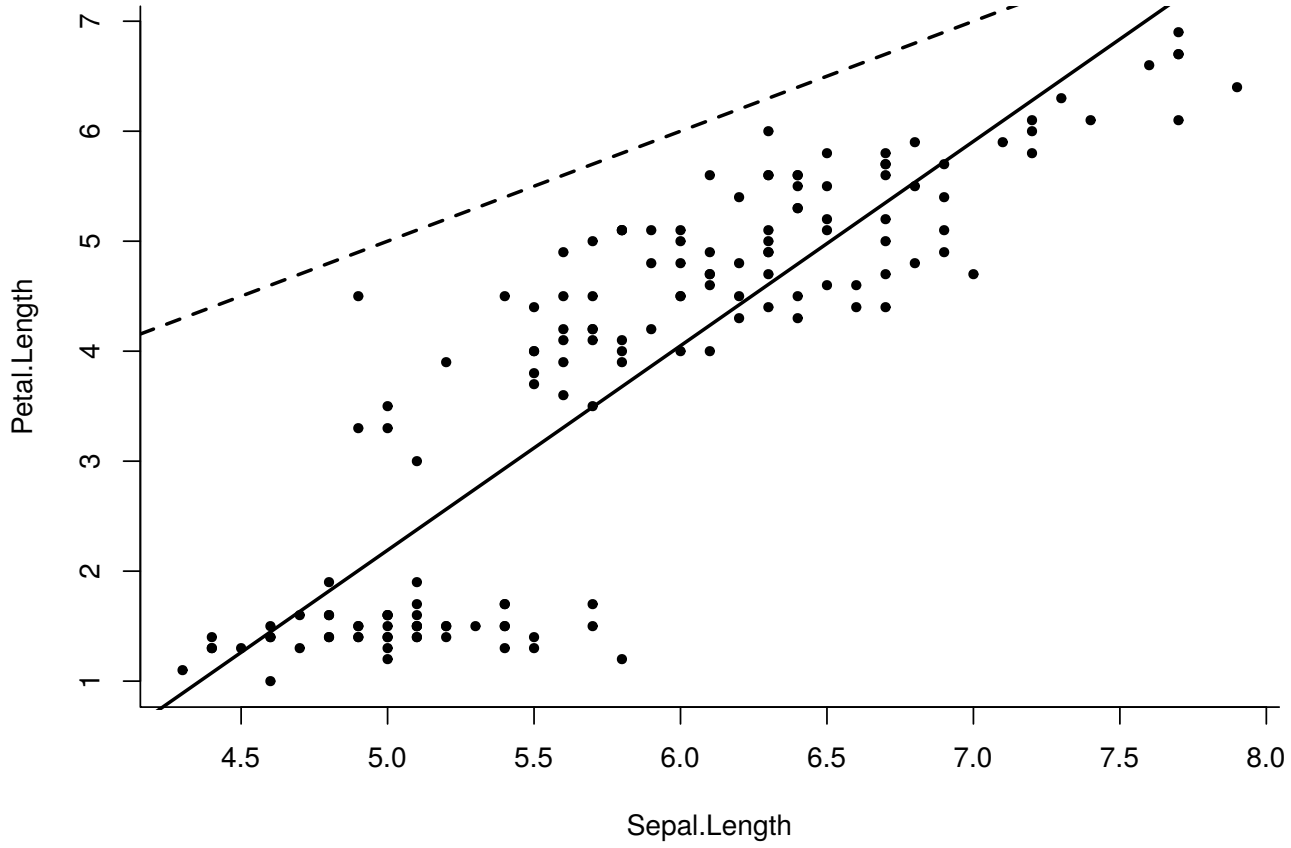
 산점도 : plot()

```
plot(cars)
```



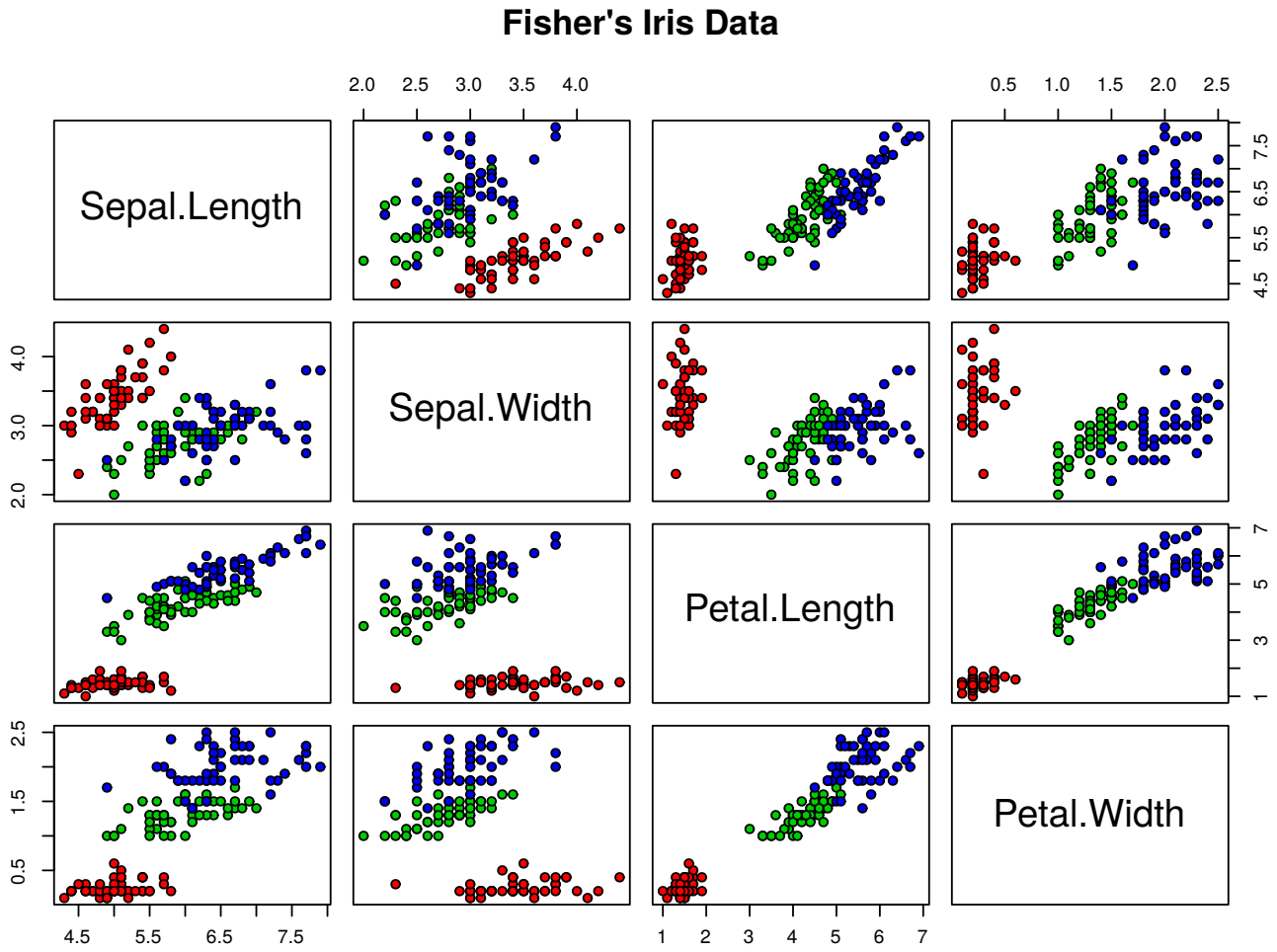
 산점도 위에 직선 추가 : abline

```
plot(Petal.Length ~ Sepal.Length, data=iris, bty="l",pch=20)
abline(a=0,b=1,lty=2,lwd=2)
abline(lm(Petal.Length ~ Sepal.Length, data=iris),lty=1,lwd=2)
```



 여러개의 산점도를 동시에 그리기 : pair()

```
pairs(iris[,1:4], main = "Fisher's Iris Data",
 pch = 21,bg = c("red","green3","blue")[unclass(iris$Species)])
```



2 3차원 도표

 trees 자료

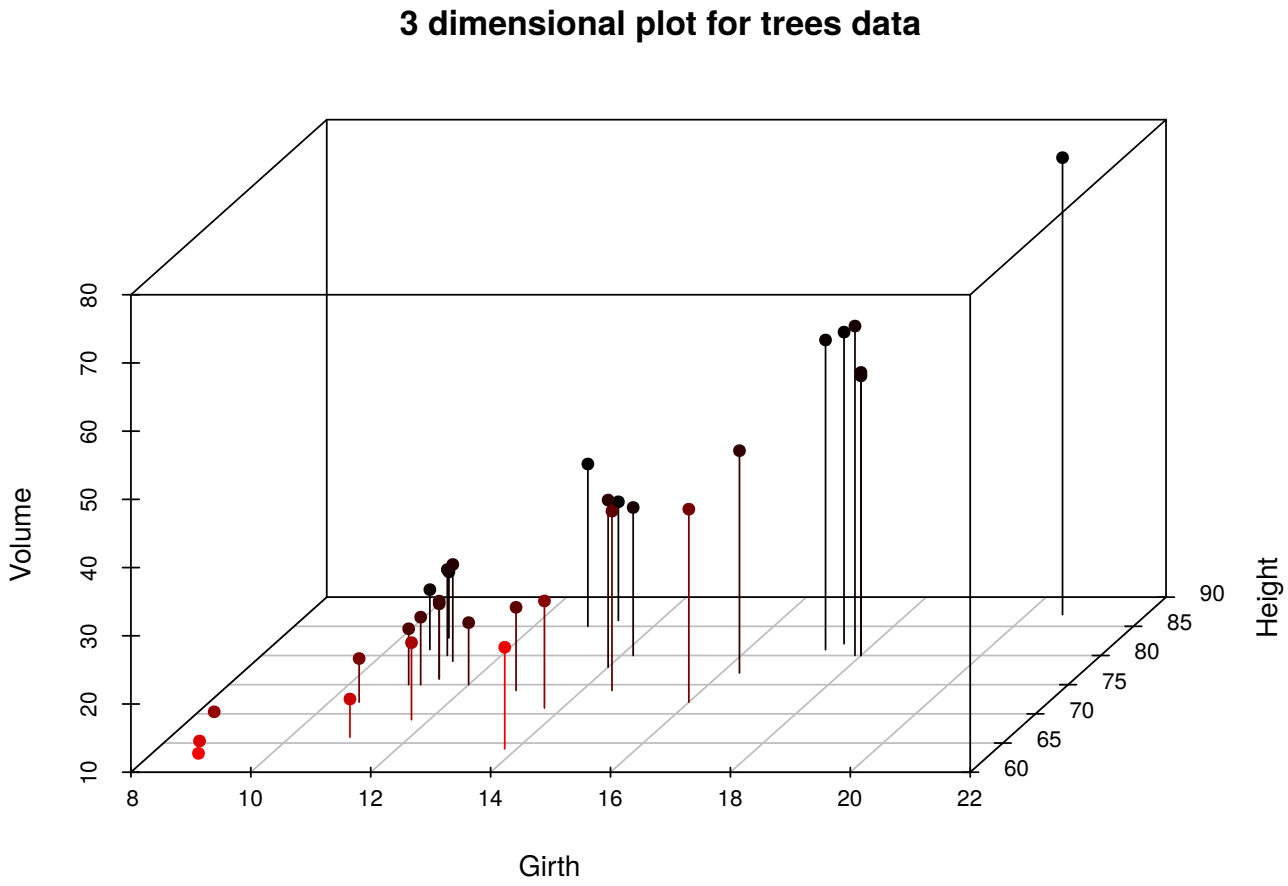
```
  A data frame with 31 observations on 3 variables.
  [,1]  Girth    numeric     Tree diameter in inches
  [,2]  Height   numeric     Height in ft
  [,3]  Volume   numeric     Volume of timber in cubic ft
```

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

2.1 3차원 산점도 : scatterplot3d::scatterplot3d

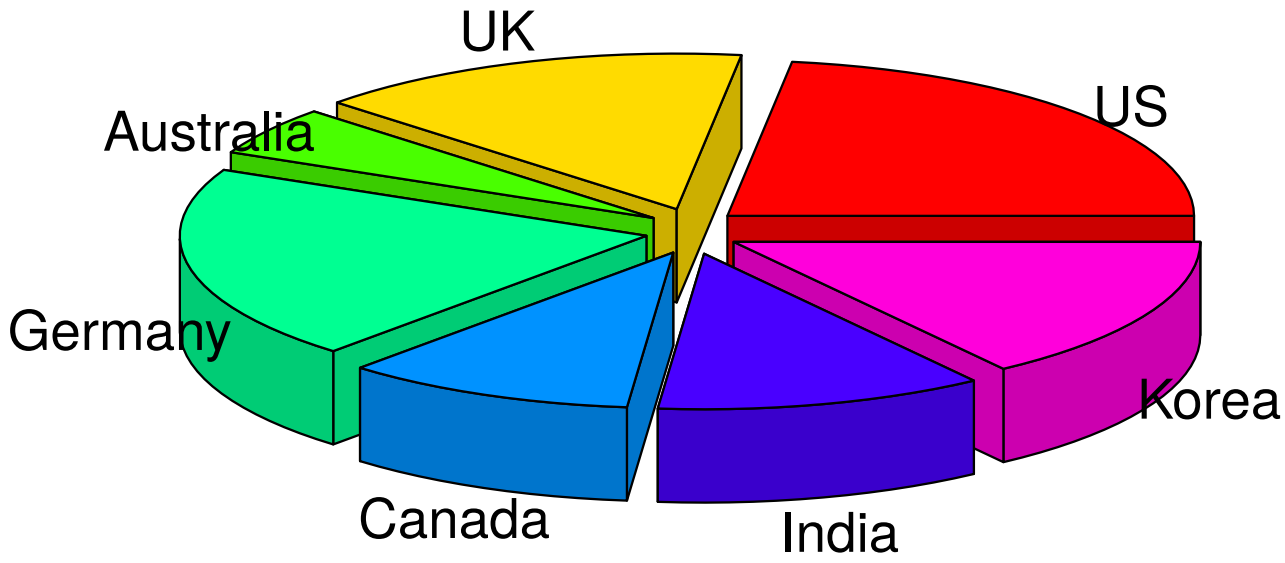
```
require(scatterplot3d)
scatterplot3d(trees, type="h", highlight.3d=TRUE,
      angle=55, scale.y=0.7, pch=16, main="3 dimensional plot for trees data")
```



2.2 3D-파이차트 범주형 자료: plotrix::pie3D

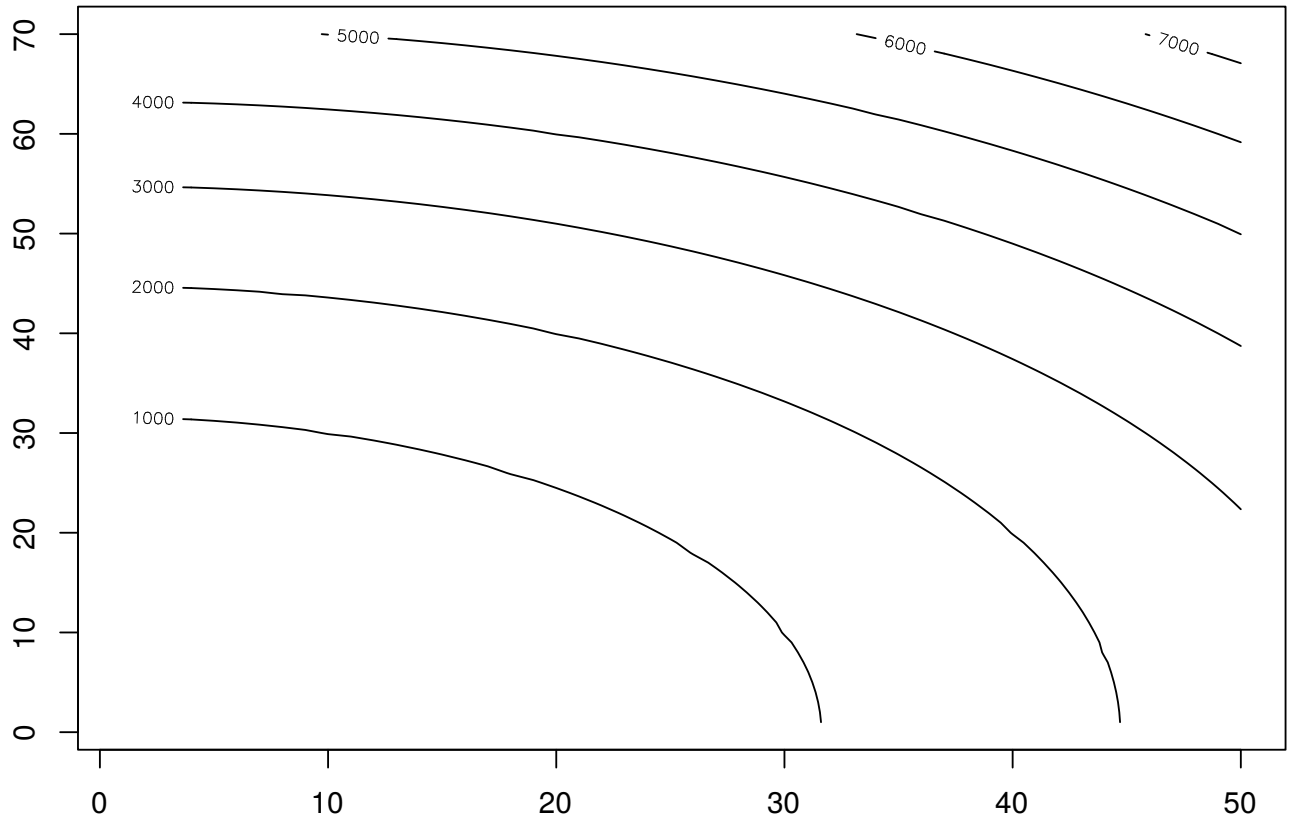
```
slices <- c(18, 12, 4, 16, 8, 9, 12)
lbels <- c("US", "UK", "Australia", "Germany", "Canada", "India", "Korea")
library(plotrix)
pie3D(slices,labels=lbels,explode=0.1, main="3D Pie Chart", mar=c(4,0,3,0))
```

3D Pie Chart

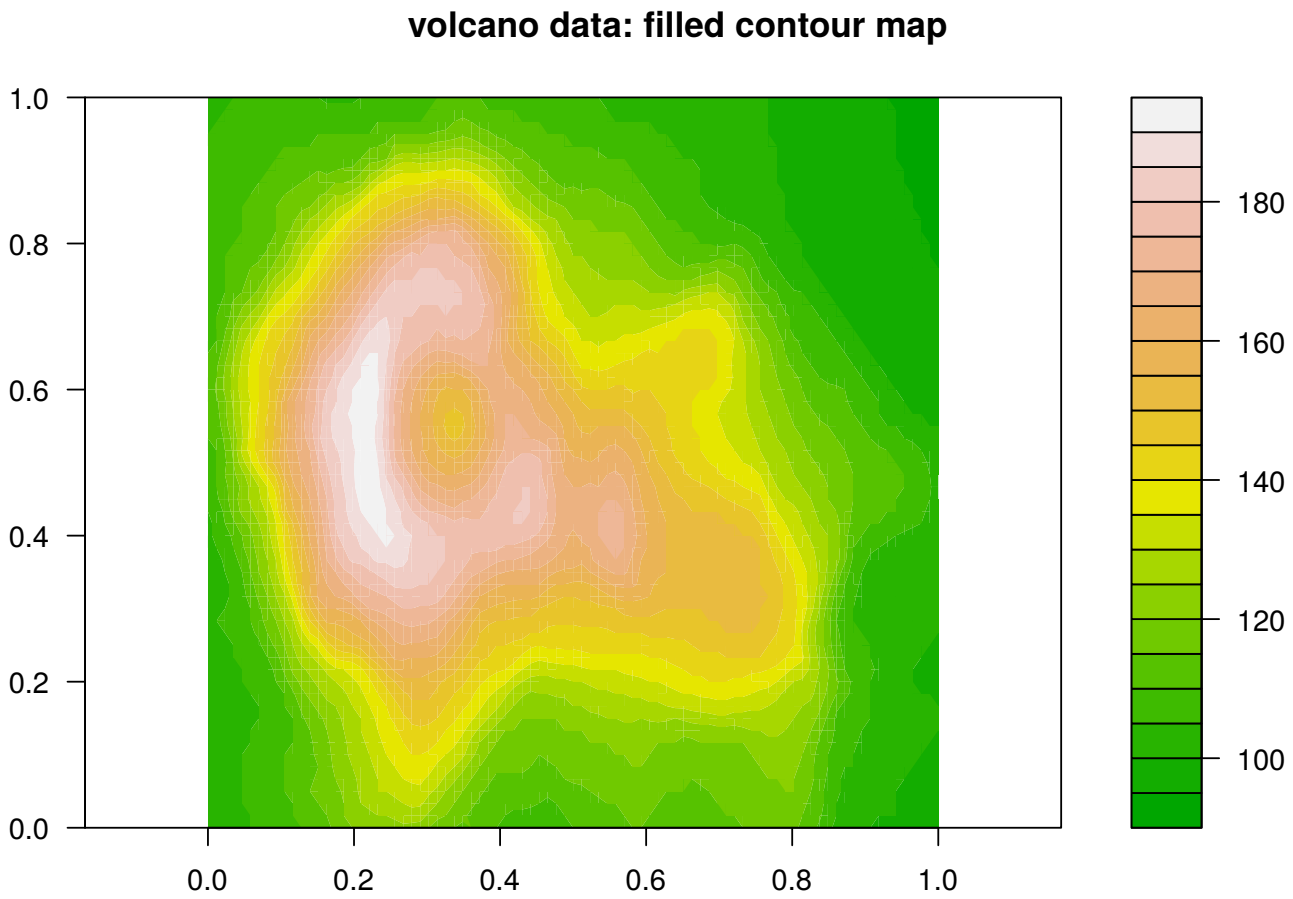


2.3 등고선 그림

```
x <- 1:50; y <- 1:70
z <- matrix(expand.grid(x,y)$Var1^2 + expand.grid(x,y)$Var2^2,50,70)
contour(x,y,z)
```



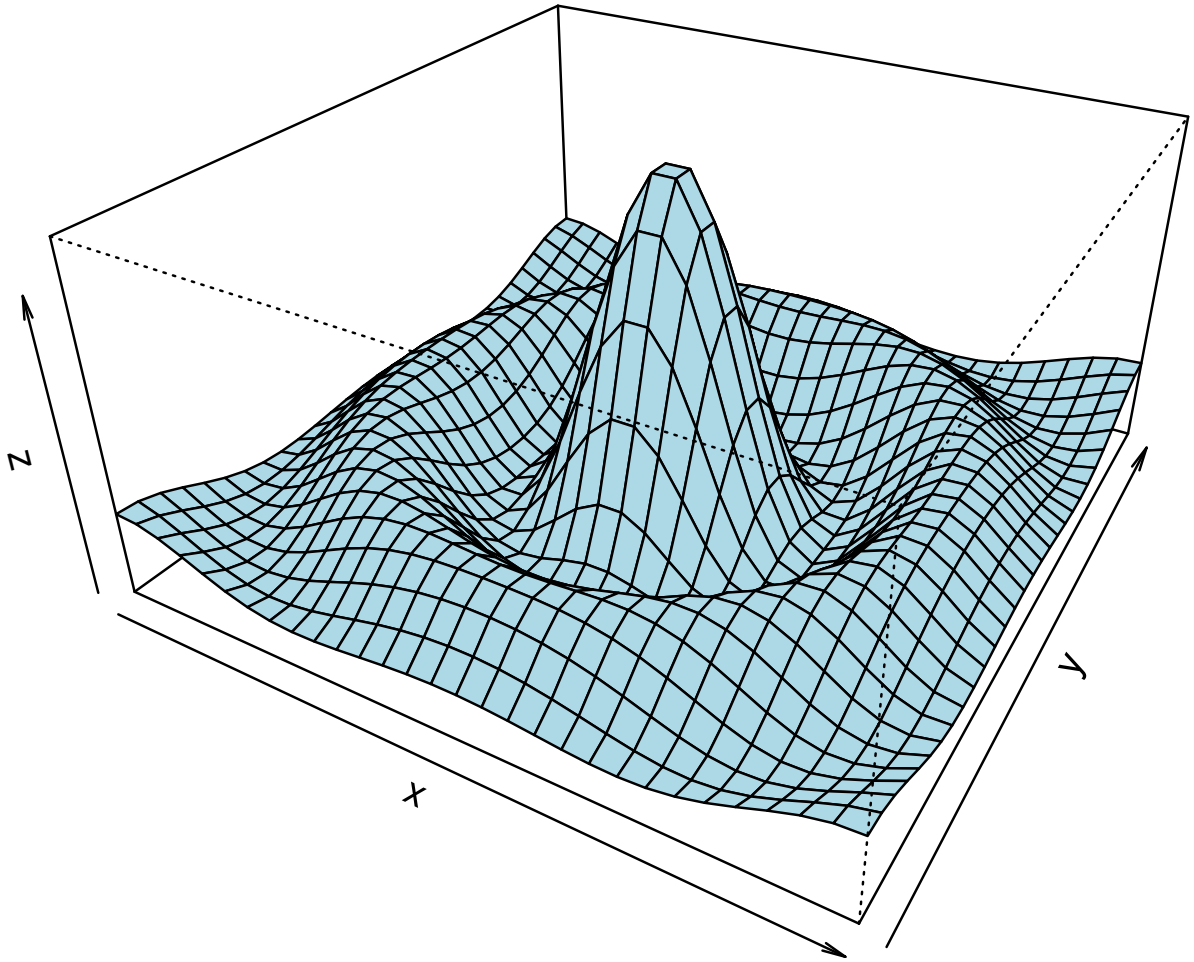
```r
filled.contour(volcano, color.palette = terrain.colors, asp = 1)
title(main = "volcano data: filled contour map")
```



2.4 투시도 (perspective plot)

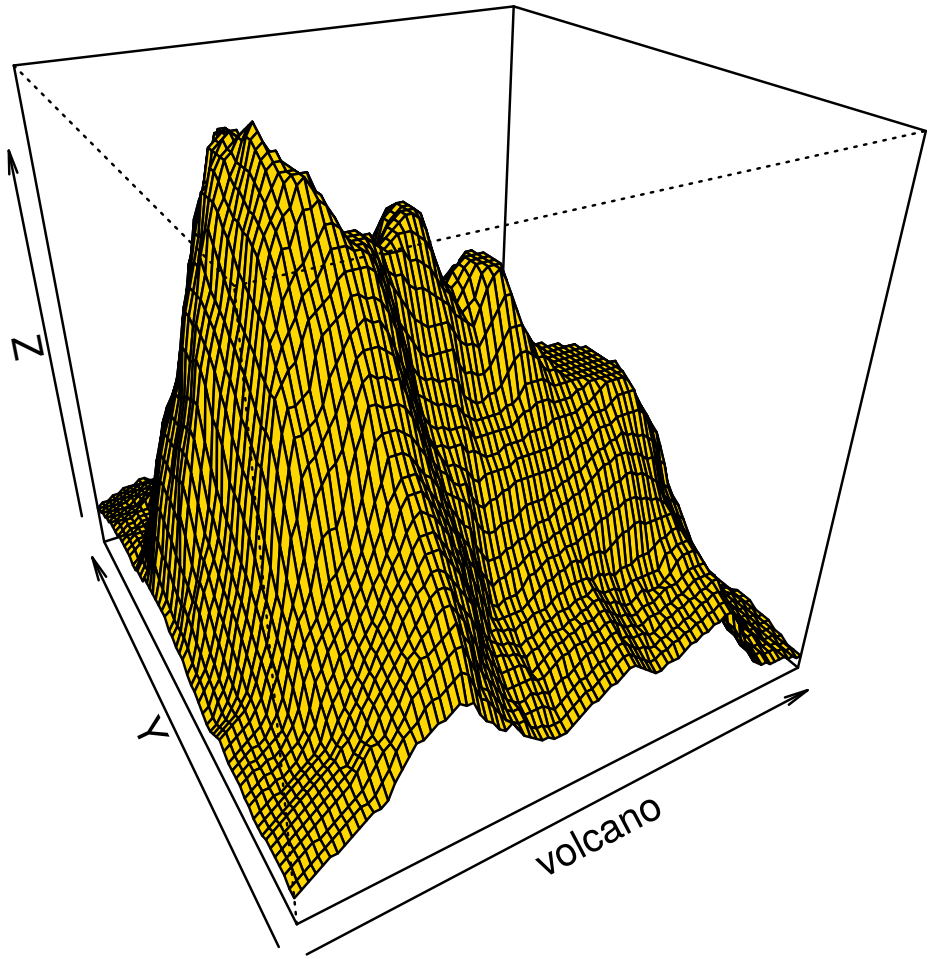
 persp: 투시도 예제

```
x <- seq(-10, 10, length= 30); y <- x
f <- function(x, y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
z <- outer(x, y, f)
z[is.na(z)] <- 1
op <- par(bg = "white")
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
```


z
x
y

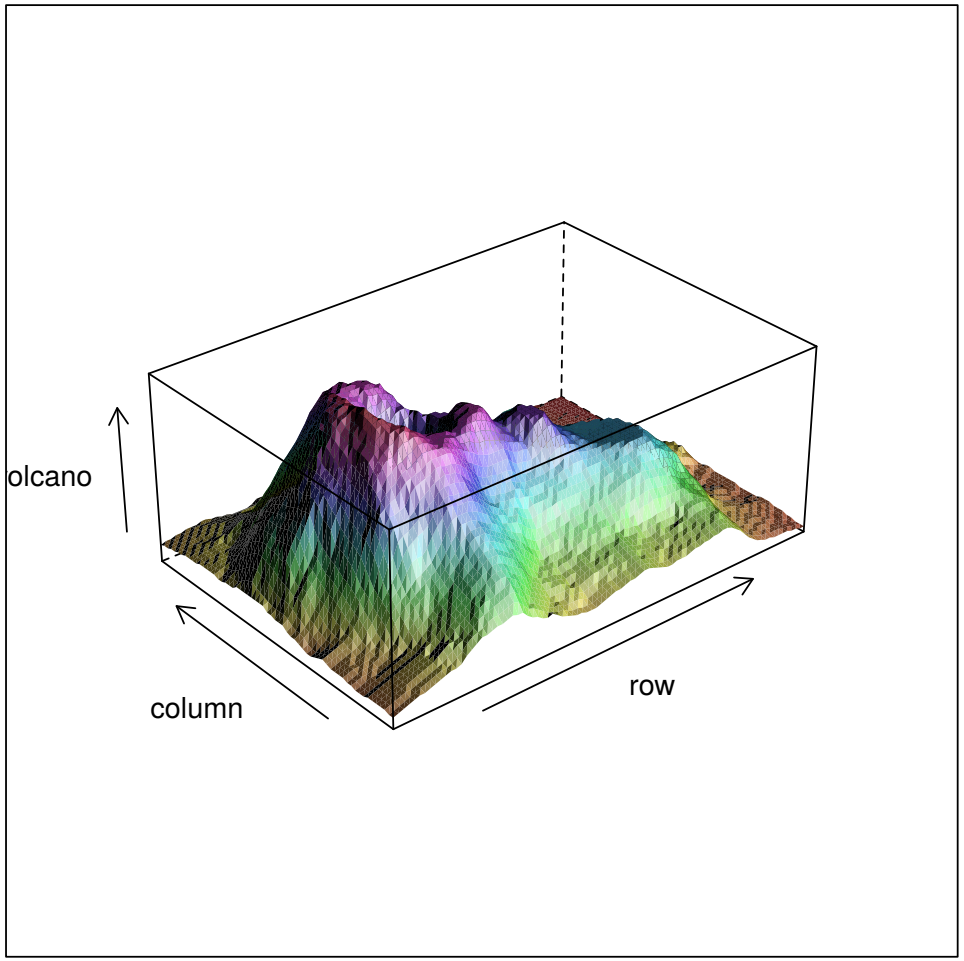
 persp: 화산자료의 투시도

```
persp(volcano, col="gold",phi = 30, theta = -30)
```



 lattice::wireframe: 화산자료의 투시도

```
library(lattice)
wireframe(volcano, shade = TRUE,  aspect = c(61/87, 0.4),
          light.source = c(10,0,10))
```



2.5 3D Visualization Using OpenGL

```
#install.packages("rgl")
library(rgl)
x <- 1:5/10
y <- 1:5
z <- x %o% y
z <- z + .2*z*runif(25) - .1*z
```

```
#install.packages("plot3d")
persp3d(x, y, z, col="skyblue")
open3d() #open new device
x <- sort(rnorm(1000))
y <- rnorm(1000)
z <- rnorm(1000) + atan2(x, y)
plot3d(x, y, z, col = rainbow(1000))
```

3 R 그래픽스 기타

3.1 R로 생성한 이미지를 파일로 저장하기

- 이미지파일 형식: pdf, png, jpeg, ⋯
- R functions: `pdf()`, `png()`,`dev.off()`, `...`

```
# 그래프를 그릴 그래프 장치(device)를 지정
pdf(file="test.pdf", width=4, height=4)
# 그래프 드로잉
plot(iris, main="save R graph to image file")
# 이미지 생성 후, 디바이스 종료
dev.off()
# 이미지저장 폴더에서 이미지 파일이 생성되었는지 확인
```

3.2 글자 모양 (fonts)

- 폰트패키지 설치

```r
# 폰트패키지 설치
# install.packages("extrafont")
# 폰트패키지 로드
library(extrafont)
# 패키지의 폰트 저장 및 폰트 로드 (설치후 한번만 할 것)
# font_import()
# loadfonts()
# 설치된 폰트 페밀리 리스트 보기
fonts() # fonttable()
```

```
##   [1] "Padauk"                  "Padauk Book"
##   [3] "Amiri"                   "Amiri Quran"
##   [5] "Arimo"                   "Baekmuk Batang"
##   [7] "Baekmuk Dotum"           "Baekmuk Gulim"
##   [9] "Baekmuk Headline"        "AR PL SungtiL GB"
##  [11] "Caladea"                 "Carlito"
##  [13] "Comfortaa"               "Cousine"
##  [15] "DejaVu Sans"             "DejaVu Sans Light"
##  [17] "DejaVu Sans Condensed"   "DejaVu Sans Mono"
##  [19] "DejaVu Serif"            "DejaVu Serif Condensed"
##  [21] "EB Garamond 08"          "EB Garamond 08 SC"
```

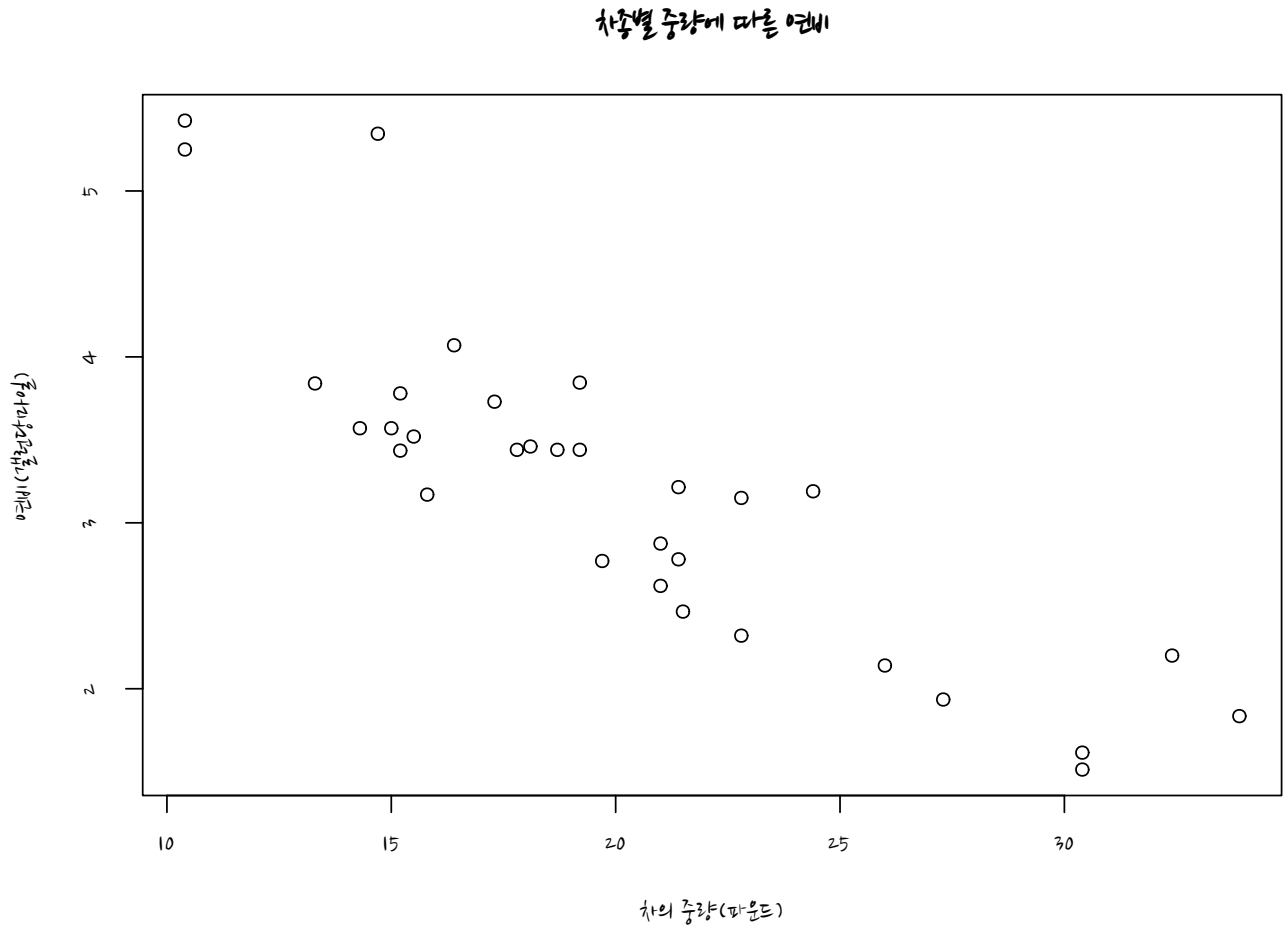
```
##  [23] "EB Garamond 12"             "EB Garamond Initials Fill1"
##  [25] "EB Garamond Initials Fill2" "FontAwesome"
##  [27] "FreeMono"                   "FreeSans"
##  [29] "FreeSerif"                  "AR PL KaitiM GB"
##  [31] "Gentium"                    "GentiumAlt"
##  [33] "Gentium Basic"              "Gentium Book Basic"
##  [35] "Gentium Plus"               "Gentium Plus Compact"
##  [37] "HCR Batang"                 "HCR Dotum"
##  [39] "IPAGothic"                  "IPAMincho"
##  [41] "IPAPGothic"                 "IPAPMincho"
##  [43] "IPAexGothic"                "IPAexMincho"
##  [45] "Junicode"                   "Lato Black"
##  [47] "Lato"                       "Lato Hairline"
##  [49] "Lato Heavy"                 "Lato Light"
##  [51] "Lato Medium"                "Lato Semibold"
##  [53] "Lato Thin"                  "Nanum Brush Script"
##  [55] "NanumGothic"                "NanumGothicExtraBold"
##  [57] "NanumMyeongjo"              "NanumMyeongjoExtraBold"
##  [59] "Nanum Pen Script"           "Noto Kufi Arabic"
##  [61] "Noto Naskh Arabic"          "Noto Naskh Arabic UI"
##  [63] "Noto Sans"                  "Noto Sans Armenian"
##  [65] "Noto Sans Bengali"          "Noto Sans Bengali UI"
```

```
##  [67] "Noto Sans Cham"            "Noto Sans Devanagari"
##  [69] "Noto Sans Devanagari UI"   "Noto Sans Ethiopic"
##  [71] "Noto Sans Georgian"        "Noto Sans Gujarati"
##  [73] "Noto Sans Gujarati UI"     "Noto Sans Gurmukhi"
##  [75] "Noto Sans Gurmukhi UI"     "Noto Sans Hebrew"
##  [77] "Noto Sans Kannada"         "Noto Sans Kannada UI"
##  [79] "Noto Sans Khmer"           "Noto Sans Khmer UI"
##  [81] "Noto Sans Lao"             "Noto Sans Lao UI"
##  [83] "Noto Sans Malayalam"       "Noto Sans Malayalam UI"
##  [85] "Noto Sans Myanmar"         "Noto Sans Myanmar UI"
##  [87] "Noto Sans Oriya"           "Noto Sans Oriya UI"
##  [89] "Noto Sans Sinhala"         "Noto Sans Tamil"
##  [91] "Noto Sans Tamil UI"        "Noto Sans Telugu"
##  [93] "Noto Sans Telugu UI"       "Noto Sans Thaana"
##  [95] "Noto Sans Thai"            "Noto Sans Thai UI"
##  [97] "Noto Sans Tibetan"         "Noto Sans UI"
##  [99] "Noto Serif"                "Noto Serif Armenian"
## [101] "Noto Serif Bengali"        "Noto Serif Georgian"
## [103] "Noto Serif Gujarati"       "Noto Serif Kannada"
## [105] "Noto Serif Khmer"          "Noto Serif Lao"
## [107] "Noto Serif Malayalam"      "Noto Serif Tamil"
## [109] "Noto Serif Telugu"         "Noto Serif Thai"
```

```
## [111] "Roboto"                    "Roboto Condensed"
## [113] "AR PL Mingti2L Big5"       "Tinos"
## [115] "UnBatang"                  "UnBom"
## [117] "UnDinaru"                  "UnDotum"
## [119] "UnGraphic"                 "UnGungseo"
## [121] "UnJamoBatang"              "UnJamoDotum"
## [123] "UnJamoNovel"               "UnJamoSora"
## [125] "UnPen"                     "UnPenheulim"
## [127] "UnPilgi"                   "UnPilgia"
## [129] "UnShinmun"                 "UnTaza"
## [131] "UnVada"                    "UnYetgul"
## [133] "AR PL KaitiM Big5"
```

 글자 모양 (fonts) 지정

```
par(family="Nanum Brush Script")
plot(mtcars$mpg, mtcars$wt, main = "차종별 중량에 따른 연비",
     xlab = "차의 중량(파운드)",  ylab = "연비(갤런당마일)")
```


차종별 중량에 따른 연비
연비(갤런당 마일)
2
3
4
5
10
15
20
25
30
차의 중량(파운드)