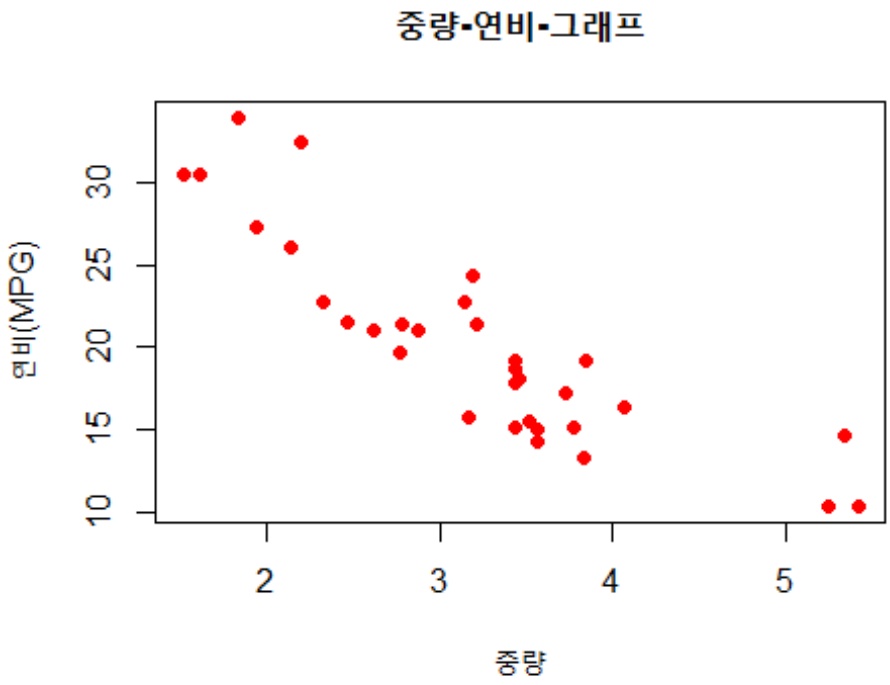
데이기반프로그래밍(2)

신봉균 20191624

2023-04-05

코드 6-1

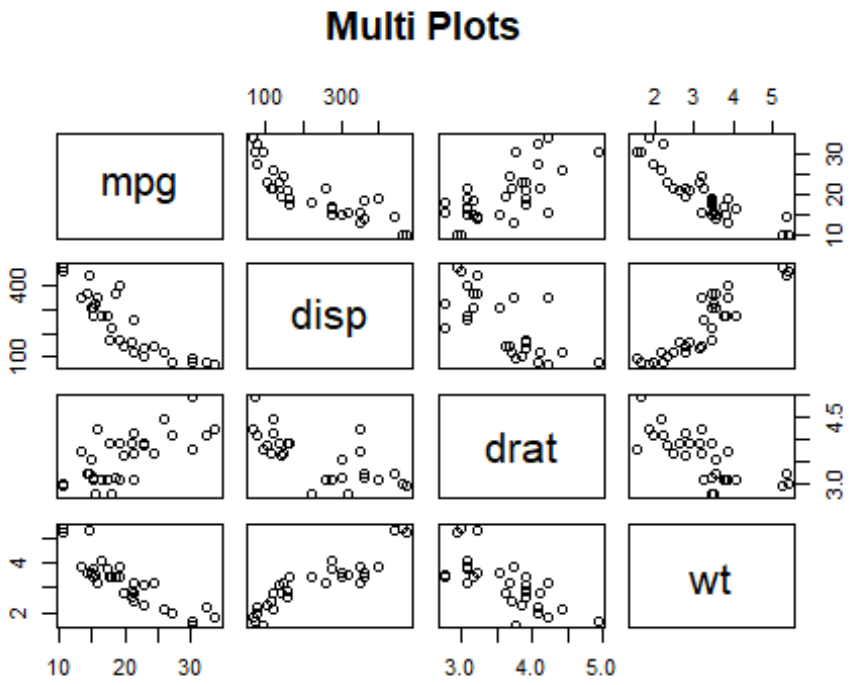
```
wt =mtcars$wt  #중량자료
mpg = mtcars$mpg  #연비자료
plot(wt, mpg,   #2 개 변수 x 축 y 축
     main= '중량-연비-그래프',
     xlab='중량',
     ylab='연비(MPG)',
     col= 'red',    #포인트의 색갈
     pch=19)  #포인트의 종류
```



산점도는 두 변수의 데이터 분포를 나타내는 것이기 때문에 두 개의 변수에 대한 자료가 필요하다. **wt** 와 **mpg** 에 각각 중량과 연비자료를 저장한 후에 **plot()**함수를 이용하여 산점도를 나타낸다. **plot()**함수의 첫 번째, 두 번째 매개변수가 산점도를 작성하고자 하는 2 개의 변수 **wt** 와 **mpg** 인데, **wt** 가 그래프에서 x 축, **mpg** 는 y 축이 된다. **plot()** 함수의 매개변수 **pch** 는 점의 모양을 지정하기 위한 것으로 **pch** 값에 따른 점의 모양은 **[그림 6-2]**와 같다.

코드 6-2

```
vars= c('mpg','disp','drat','wt') #대상 변수
target = mtcars[,vars]
head(target)

##                    mpg disp drat    wt
## Mazda RX4         21.0  160 3.90 2.620
## Mazda RX4 Wag     21.0  160 3.90 2.875
## Datsun 710        22.8  108 3.85 2.320
## Hornet 4 Drive    21.4  258 3.08 3.215
## Hornet Sportabout 18.7  360 3.15 3.440
## Valiant           18.1  225 2.76 3.460

pairs(target,
      main= 'Multi Plots')
```



mtcars 데이터셋에는 11 개의 변수가 있는데 이 중 4 개의 변수인 **mpg**(연비), **disp**(배기량), **drat**(리어액슬기어비), **wt**(중량)만 선택하여 **target** 데이터셋을 새로 만들었다.

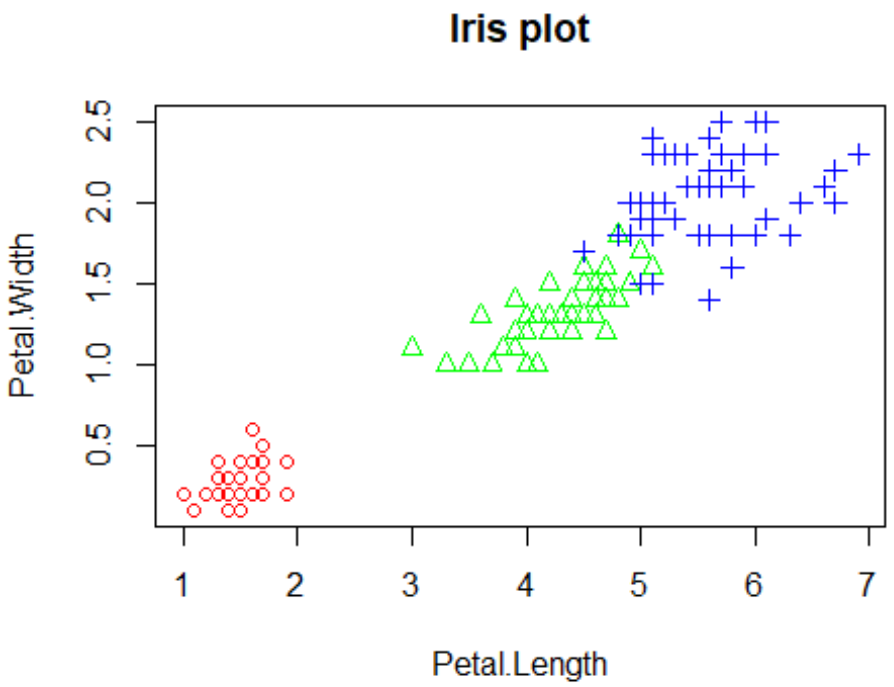
pairs()함수를 이용하여 4 개의 변수에 대한 다중 산점도를 함. 4 개의 변수가 대각선에 표기되어있다. 대각석을 기준으로 오른쪽 위와 왼쪽 아래와 대칭을 이루는 구조이다. 이와 같이 다중 산점도는 여러 변수들 간의 추세를 한눈에 파악할 수 있어서 편리하다.

코드 6-3

```
iris.2= iris[,3:4]    #데이터 준비
point = as.numeric((iris$Species)) #점의 모양
point
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2
##  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3
3 3 3
## [112] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3
## [149] 3 3
```

```
color= c('red','green','blue')    #점의 색상
plot(iris.2,
     main='Iris plot',
     pch=c(point),
     col= color[point])
```



iris[,3:4] iris 데이터의 모든 row, col 은 3 번째 4 번째 데이터만 불러온다.

as.numeric(iris*Species*) ** *는팩터타입으로되어있는* ** *iris***Species** 를 숫자로 바꾸는 함수이다. 그 결과 setosa, versicolor, virginica 품종이 각각 1,2,3,으로 변환되었다.

plot()함수를 이용하여 산점도를 작성할때 매개변수를 **pch** 는 품종을 나타내는 **point** 벡터에서 선택하고, 점의 색은 color 벡터에 있는 값에서 선택한다.

위 데이터를 분석하면 꽃잎의 길이가 커지면 꽃잎의 폭이 커지는 것을 알 수 있다.

virginica 의 품종은 다른 두품중에 비해 꽃잎의 길이와 폭이 제일 크다는 것도 알 수 있다.

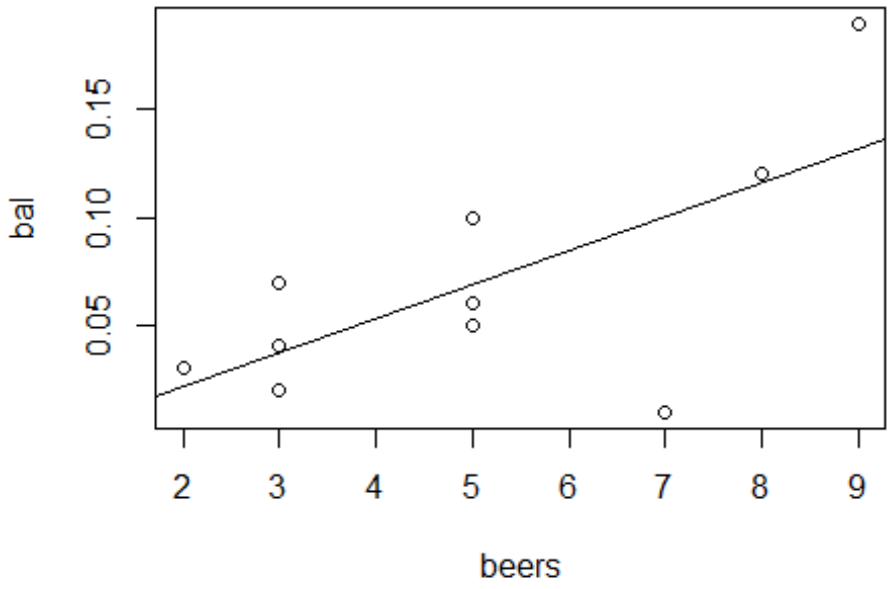
코드 6-4

```
beers= c(5,2,9,8,3,7,3,5,3,5) #자료입력
bal= c(0.1,0.03,0.19,0.12,0.04,0.0095,0.07,0.06,0.02,0.05) #자료입력
```

```r
tbl = data.frame(beers, bal) #데이터 프레임 생성
tbl
```

```
##    beers    bal
## 1      5 0.1000
## 2      2 0.0300
## 3      9 0.1900
## 4      8 0.1200
## 5      3 0.0400
## 6      7 0.0095
## 7      3 0.0700
## 8      5 0.0600
## 9      3 0.0200
## 10     5 0.0500
```

```r
plot(bal~beers, data=tbl)  #회귀식 도출
res= lm(bal~beers, data= tbl) #회귀선 그리기
abline(res)
```




```r
cor(beers,bal)
```

```
## [1] 0.6797025
```

beers 는 맥주를 마신 정도이고 **bal** 혈중 알콜농도를 나타내며 **plot()**함수를 사용하여 **혈중알콜농도=f(맥주를마신 정도)** 즉, 맥주 마신 정도별 혈중알콜농도를 나타내는 그림을 그렸으며 위와 같은 그림이 나왔다. 거기에 **lm(), abline()**함수를 사용하여 회귀선을 그려 넣었다. lm()은 회귀식 도출 abline()은 회귀선을 그리는 역할을 한다.

cor() 함수와 tbl 데이터 프레임 안에있는 beers 와 bal 를 이용하여 **corelation coeficient** 구했으며 결과는 약 0.68 이 나왔으며 이와 같은 숫자는 강한 양의 상관관계를 가지고 있다고 해석한다.

코드 6-5

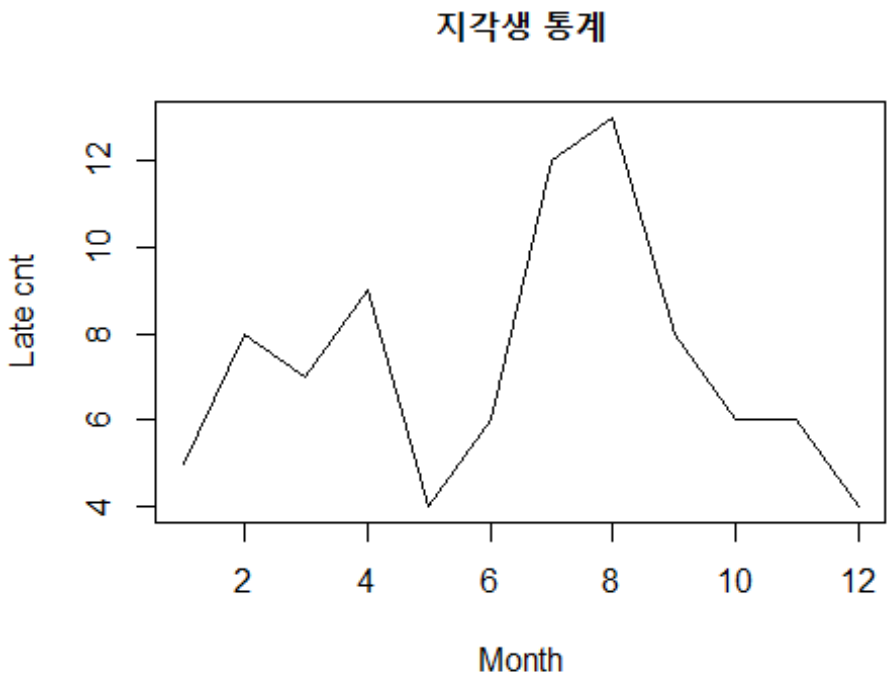
```
cor(iris[,1:4]) #4 개 변수 간 상관성 분석
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

실행 결과를 보면 4 개의 변수가 x 축,y 축 방향으로 나열되어 있고, 두 변수가 만나는 지점에 두 변수의 상관계수가 표시되어 있다. petal.legnth 와 petal.width 와의 상관관계 가장 강하다.

코드 6-6

```
month= 1:12  #자료 입력
late= c(5,8,7,9,4,6,12,13,8,6,6,4) #자료 입력
plot(month,      #xdata
     late,         #ydata
     main='지각생 통계', #제목
     type= 'l',
     lty=1,             #선의 종류 (line type)선택
     lwd=1,             #선의 굵기 선택
     xlab= 'Month',
     ylab= 'Late cnt')
```



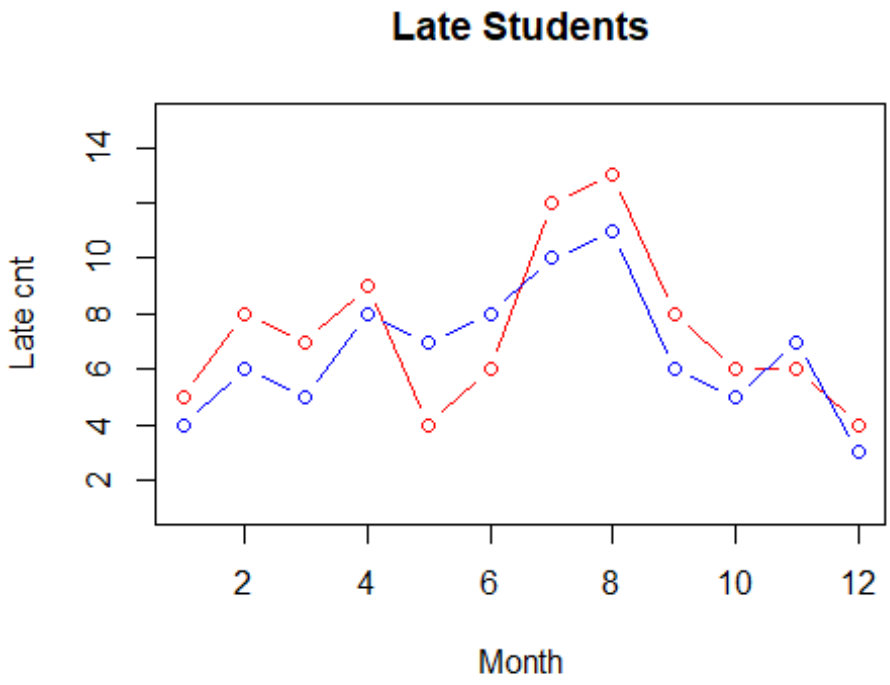
선그래프를 작성하는 함수는 산점도를 작성할 때 사용한 **plot()**함수이다. **plot()**함수에서 매개변수 **type** 의 값을 **l** 선 그래프가 작성된다. **type** 의 값은 숫자가 아니라 알파벳이다.

그래프의 결과를 보면 지각생 수가 5 월에 급감했다가 7,8 월에는 급증하는 것을 알 수 있다.

코드 6-7

```
month= 1:12
late1= c(5,8,7,9,4,6,12,13,8,6,6,4)
late2= c(4,6,5,8,7,8,10,11,6,5,7,3)
plot(month,
     late1, #x data
     main= 'Late Students', #y data
     type='b',  #그래프의 종류 선택
     lty=1,    #선의 종류 선택
     col= 'red', #선의 색 선택
     xlab= 'Month',  #x 축 레이블
     ylab = 'Late cnt', #y 축 레이블
     ylim = c(1,15)) #y 축의 범위 제한
```

```
lines(month,
      late2,
      type = 'b',
      col='blue')
```

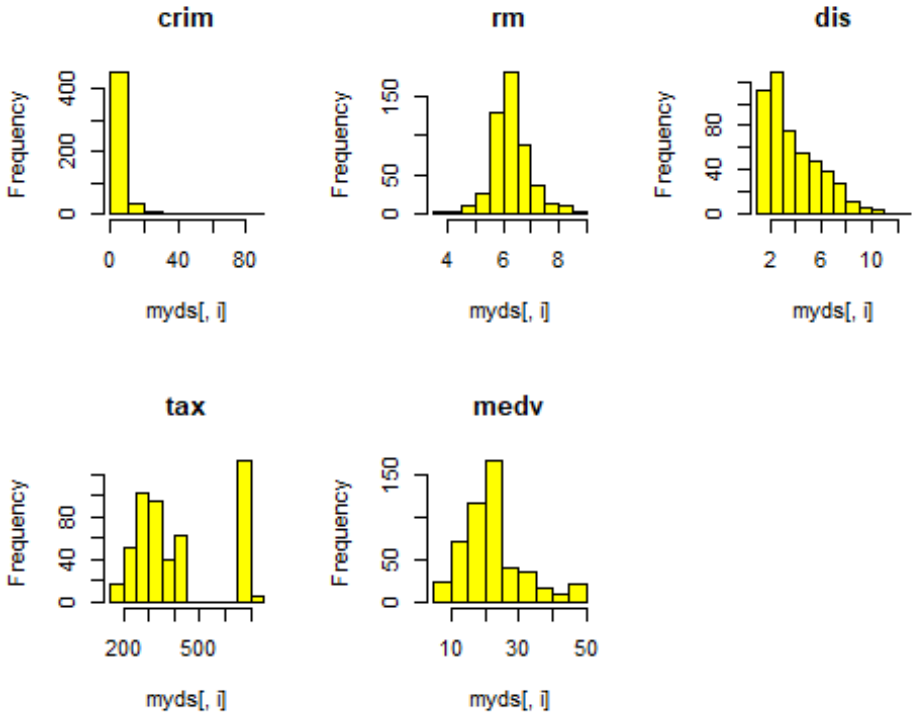


lines()함수는 **plot()**함수로 작성한 그래프 위에 선을 겹쳐서 그리는 역할을 한다.

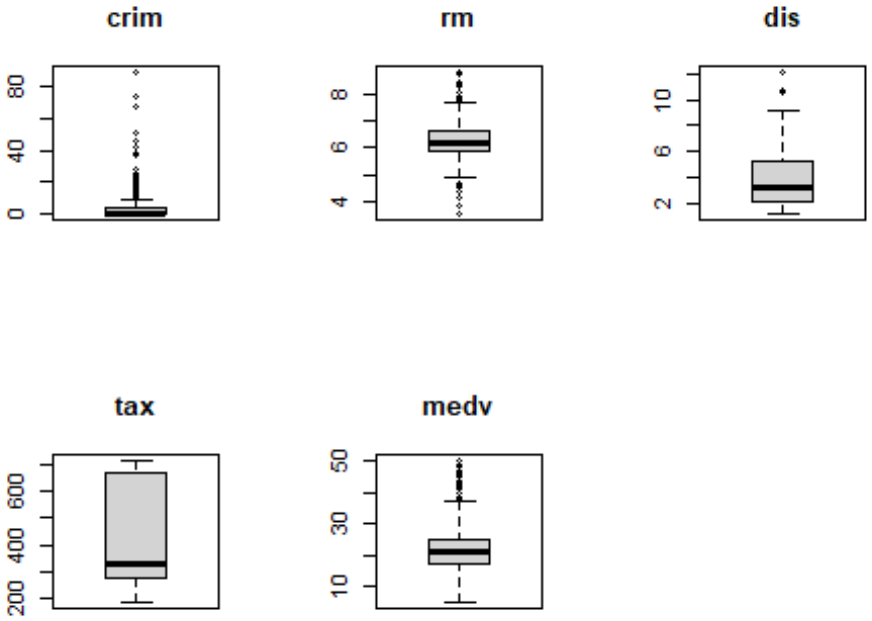
코드 6-8

```
## (1) Prepare Data----------------------
library(mlbench)

## Warning: 패키지 'mlbench'는 R 버전 4.2.3 에서 작성되었습니다

data('BostonHousing')
myds= BostonHousing[,c('crim','rm','dis','tax','medv')]

##(2) Add new column-----------
grp = c()
for (i in 1:nrow(myds)) {              #myds$medv 값에 따라 그룹 분류
  if (myds$medv[i] >= 25.0){
    grp[i] = 'H'
  } else if (myds$medv[i] <= 17.0) {
    grp[i] = 'L'
  } else {
```

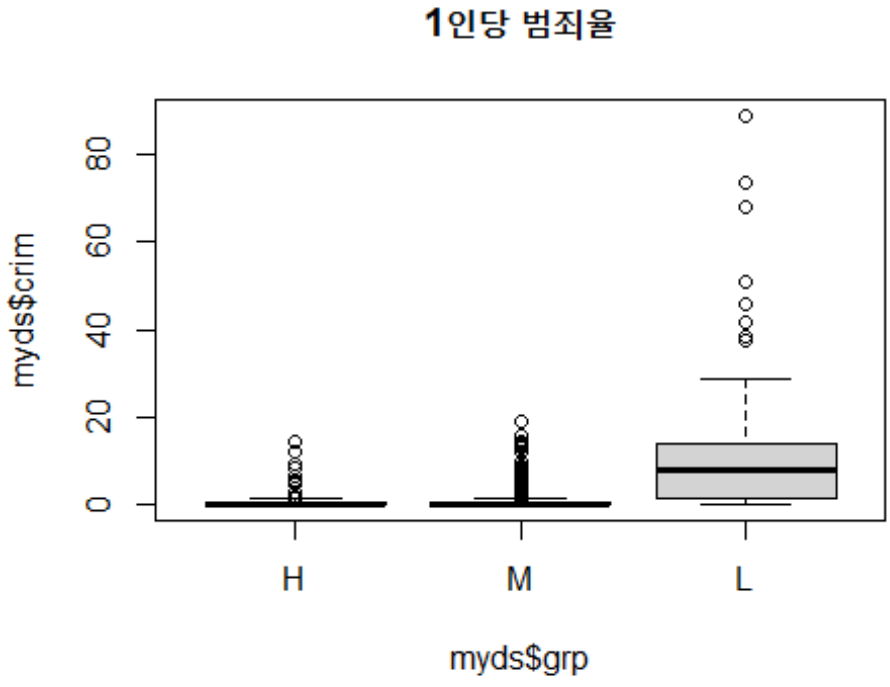
```
    grp[i] = 'M'
  }
}
grp = factor(grp)                          #문자벡터를 팩터 타입으로 변경
grp = factor(grp, levels= c('H','M','L')) #레벨의 순서를 H,L,M -> H,M,L

myds = data.frame(myds, grp)                        #myds 에 grp 칼럼추가

## (3) Add new column----------------------
str(myds)

## 'data.frame':    506 obs. of  6 variables:
##  $ crim: num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ rm  : num  6.58 6.42 7.18 7 7.15 ...
##  $ dis : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ tax : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ medv: num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
##  $ grp : Factor w/ 3 levels "H","M","L": 2 2 1 1 1 1 2 1 3 2 ...

head(myds)

##      crim    rm    dis tax medv grp
## 1 0.00632 6.575 4.0900 296 24.0   M
## 2 0.02731 6.421 4.9671 242 21.6   M
## 3 0.02729 7.185 4.9671 242 34.7   H
## 4 0.03237 6.998 6.0622 222 33.4   H
## 5 0.06905 7.147 6.0622 222 36.2   H
## 6 0.02985 6.430 6.0622 222 28.7   H

table(myds$grp)          #주택 가격 그룹별 분포

##
##   H   M   L
## 132 247 127

## (4) histogram--------------------
par(mfrow= c(2,3))    #2x3 가상화면 분할
for (i in 1:5) {
  hist(myds[,i], main=colnames(myds)[i], col='yellow')
}
par(mfrow=c(1,1)) #2x3 가상화면 분할 해제제
```



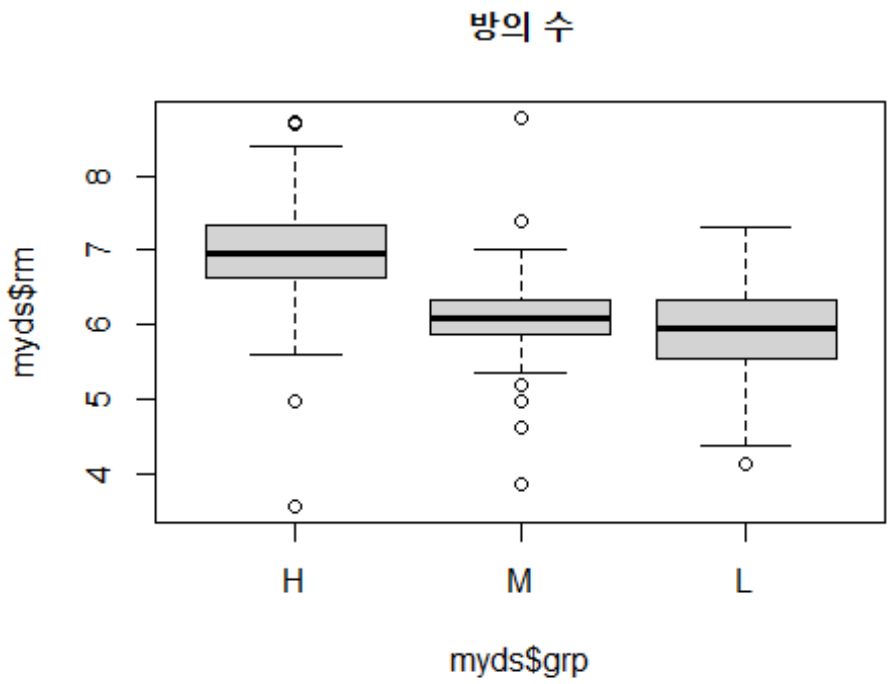

```
## (5)boxplot -----------------
par(mfrow=c(2,3)) #2x3 가상화면 분할
for(i in 1:5){
  boxplot(myds[,i], main=colnames(myds)[i])
}
par(mfrow=c(1,1)) #2x3 가상화면 분할 해제
```



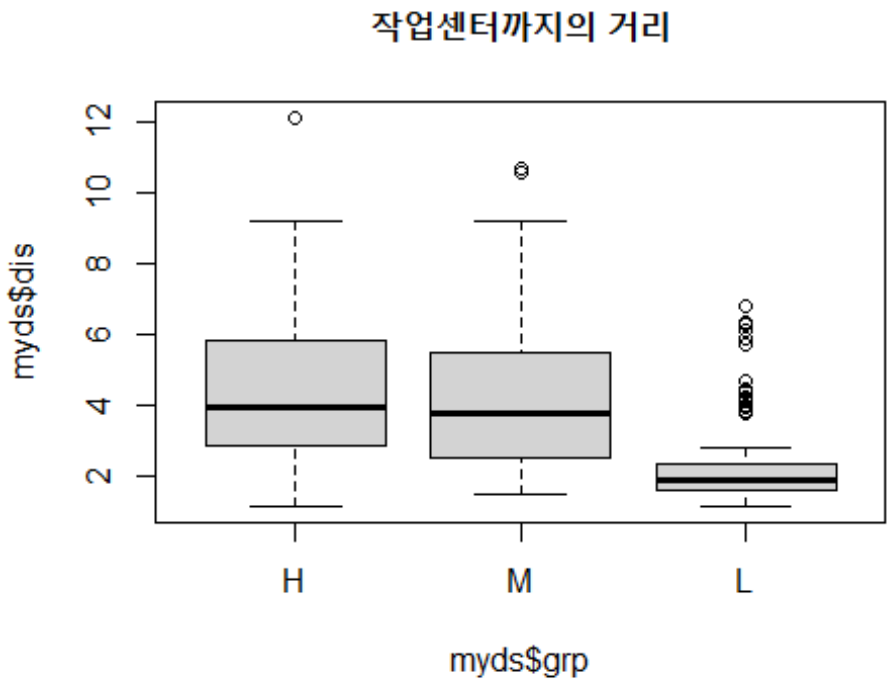

```
## (6) boxplot by group------------------
boxplot(myds$crim~myds$grp, main='1 인당 범죄율')
```



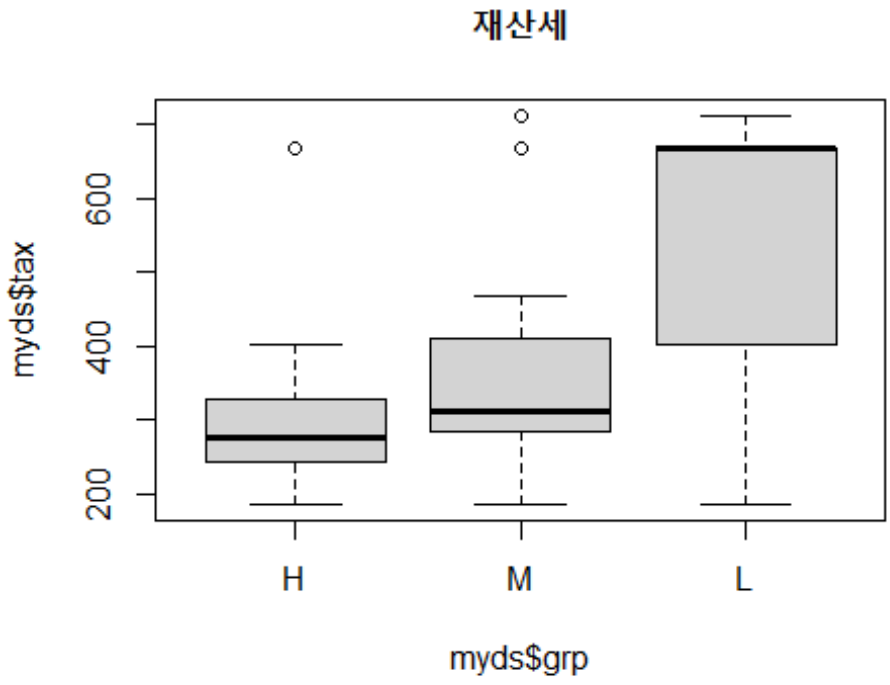
```
boxplot(myds$rm~myds$grp, main='방의 수')
```



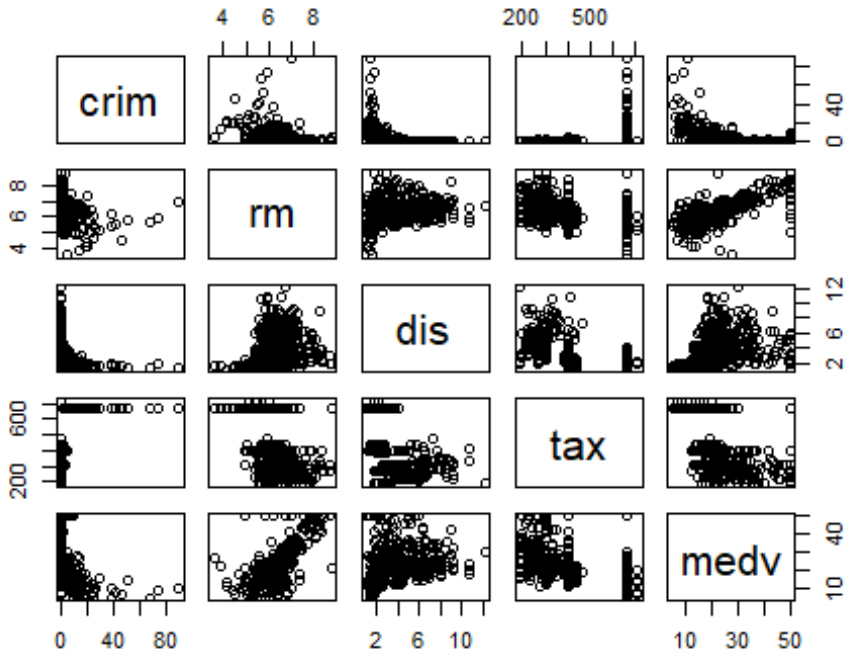

```
boxplot(myds$dis ~ myds$grp, main='작업센터까지의 거리')
```



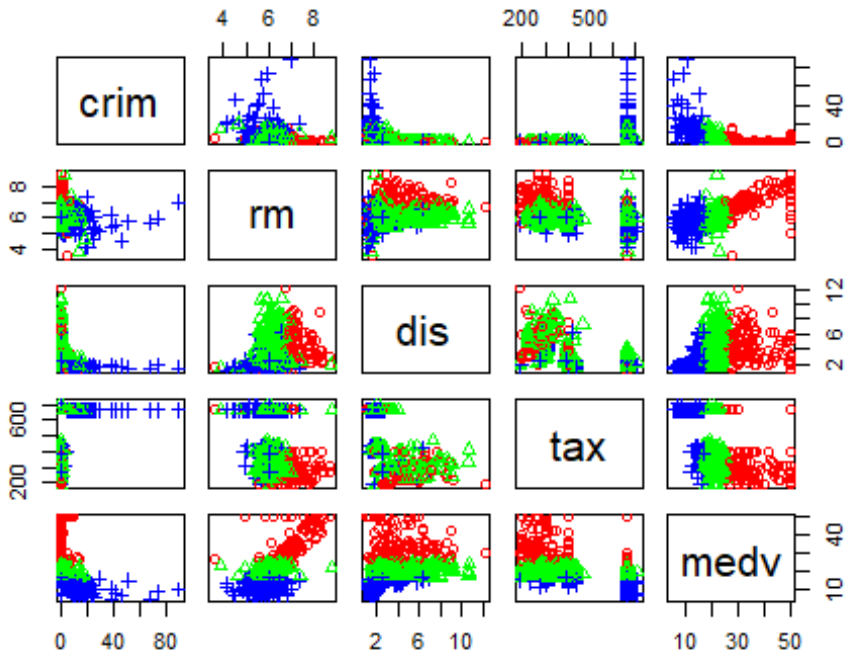

```
boxplot(myds$tax~myds$grp, main='재산세')
```



```r
## (7) scatter plot-----------------------
pairs(myds[,-6])
```




```r
## (8) scatter plot with group--------------
point = as.integer(myds$grp)      #점의 모양 지정
color = c('red','green','blue')   #점의 색 지정
pairs(myds[,-6],pch= point, col=color[point])
```




```
## (9) correlation coefficient-----------------
cor(myds[,-6])
```

```
##             crim         rm        dis        tax       medv
## crim  1.0000000 -0.2192467 -0.3796701  0.5827643 -0.3883046
## rm   -0.2192467  1.0000000  0.2052462 -0.2920478  0.6953599
## dis  -0.3796701  0.2052462  1.0000000 -0.5344316  0.2499287
## tax   0.5827643 -0.2920478 -0.5344316  1.0000000 -0.4685359
## medv -0.3883046  0.6953599  0.2499287 -0.4685359  1.0000000
```