

## [R] R 에서 사용되는 기본적인 시각화 그래프-2 — 나무늘보의 개발 블로그

노트북: blog

만든 날짜: 2020-10-03 오후 5:50

URL: <https://continuous-development.tistory.com/41?category=793392>

---



R

# [R] R 에서 사용되는 기본적인 시각화 그래프-2

2020. 7. 28. 22:38 수정 삭제 공개

## #산점도

x축과 y축으로 이루어진 그래프에 두 변수의 값을 점으로 나타낸 그래프이다.  
산점도를 이용하면 두 변수의 관계를 파악하는데 용이하다.

구문

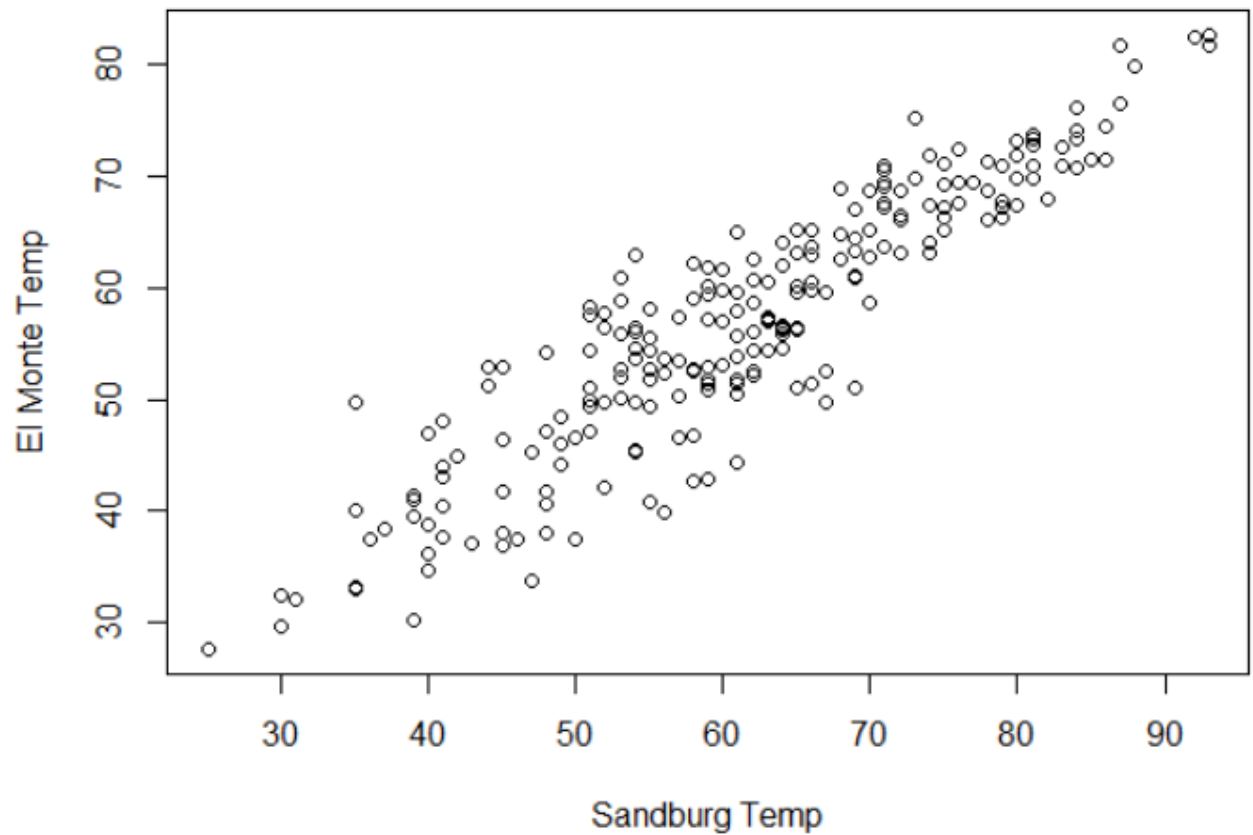
plot( x , y) - 기본구문

```
plot( x , y,  
      xlab="naem1",  
      ylab="name2") # 이런식으로 그래프의 축에 lab을 할 수 있다.
```

## # xlab, ylab

보통 구문은 `plot( x, y )` 로 시작 되고 여기서는 `xlab ,ylab`으로 x 축 lab을 쓰고 y 축 lab을 썼다,

```
18  
19 #축이름(xlab,ylab)  
20 plot(Ozone$V8,Ozone$V9,  
21      xlab = "Sandburg Temp",  
22      ylab = "EI Monte Temp")  
23
```



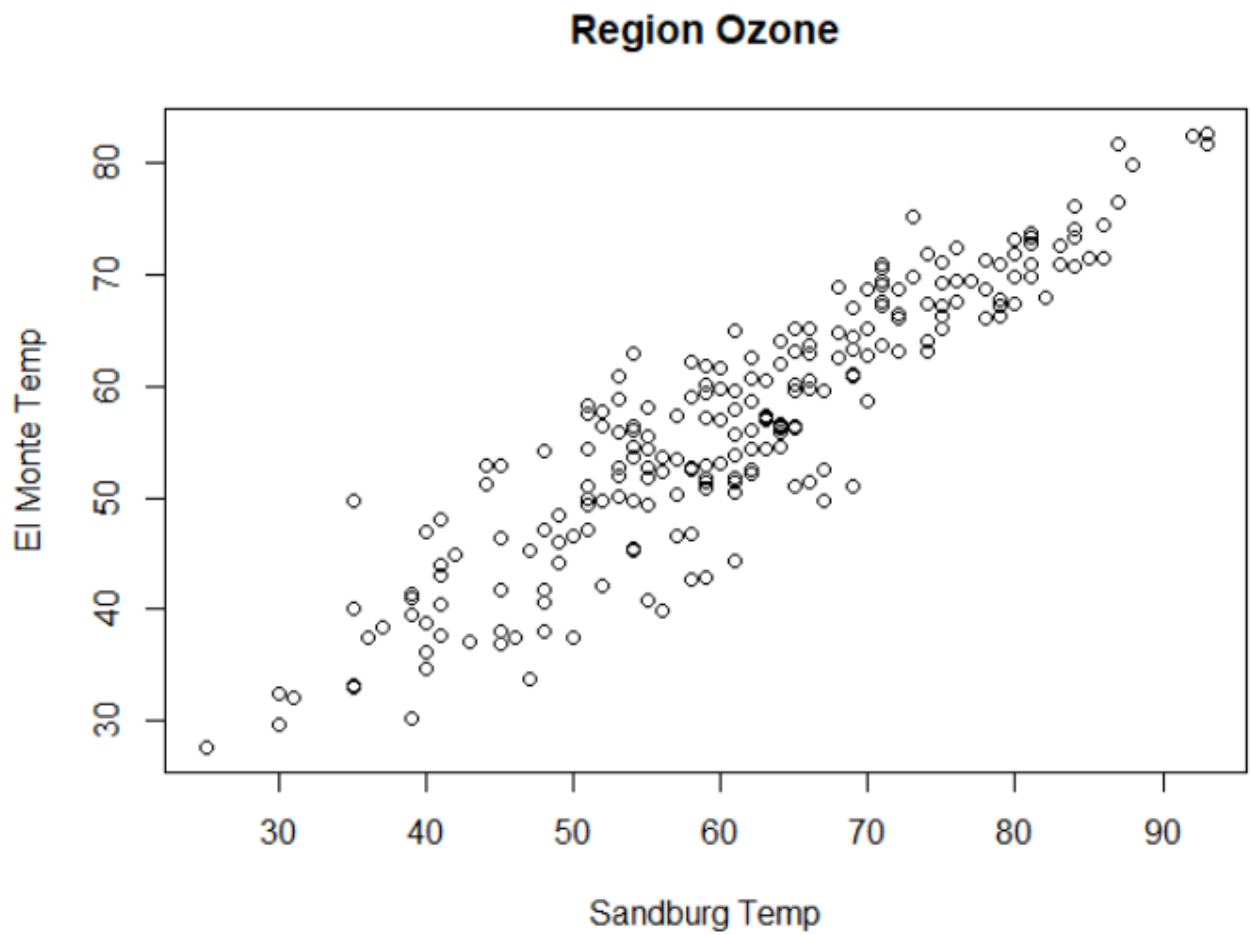
**#main - title**

그래프의 이름을 적는다.

```

23
24 #title(main)
25 plot(Ozone$V8,Ozone$V9,
26       xlab = "Sandburg Temp",
27       ylab = "EI Monte Temp",
28       main = "Region Ozone")

```



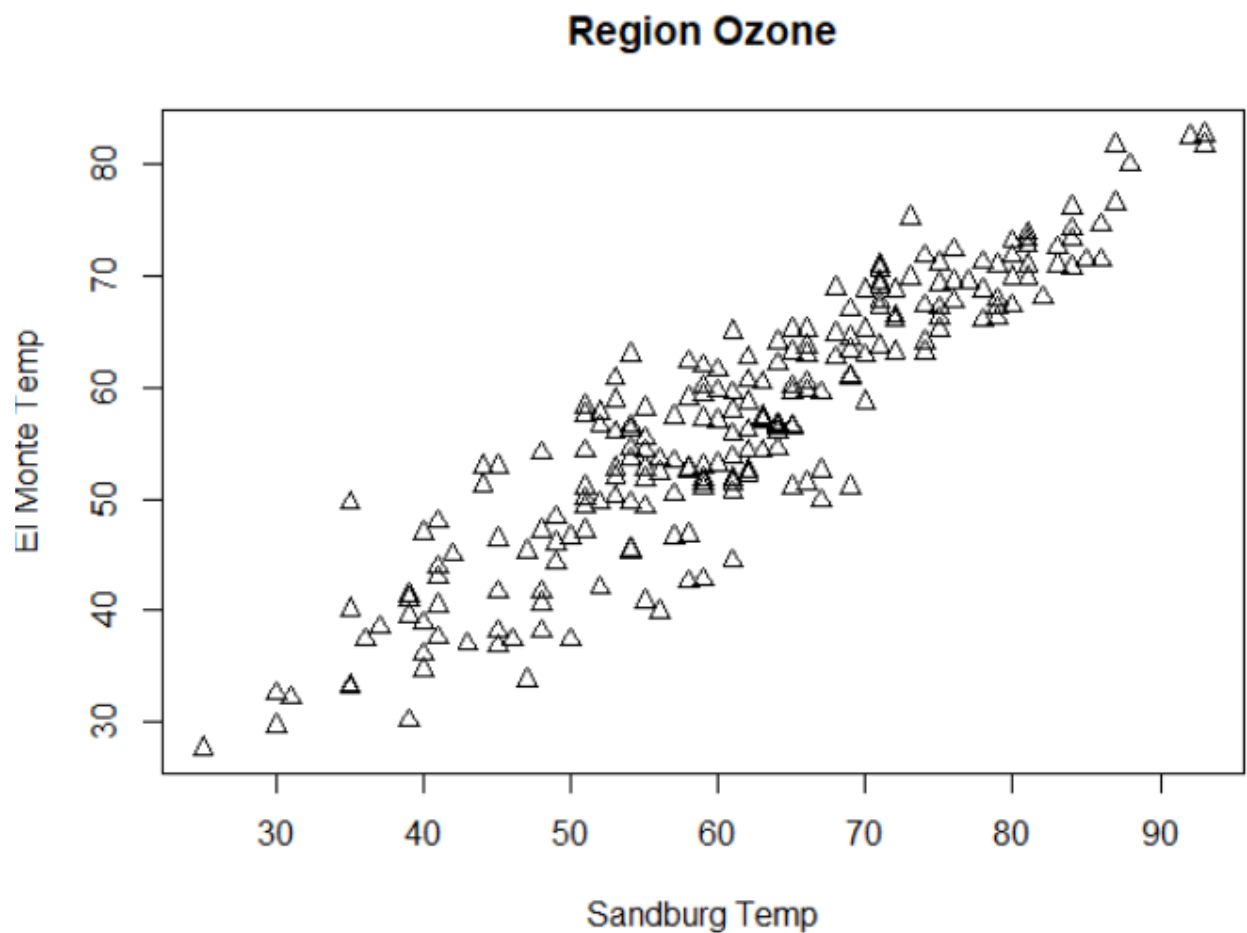
## #pch - 점의종류

pch 를 통해 정의 모양을 바꿀수 있다.

```

30
31 #점의 종류 (pch)
32 plot(Ozone$V8,Ozone$V9,
33       xlab = "Sandburg Temp",
34       ylab = "EI Monte Temp",
35       main = "Region Ozone",
36       pch = 2)
37

```



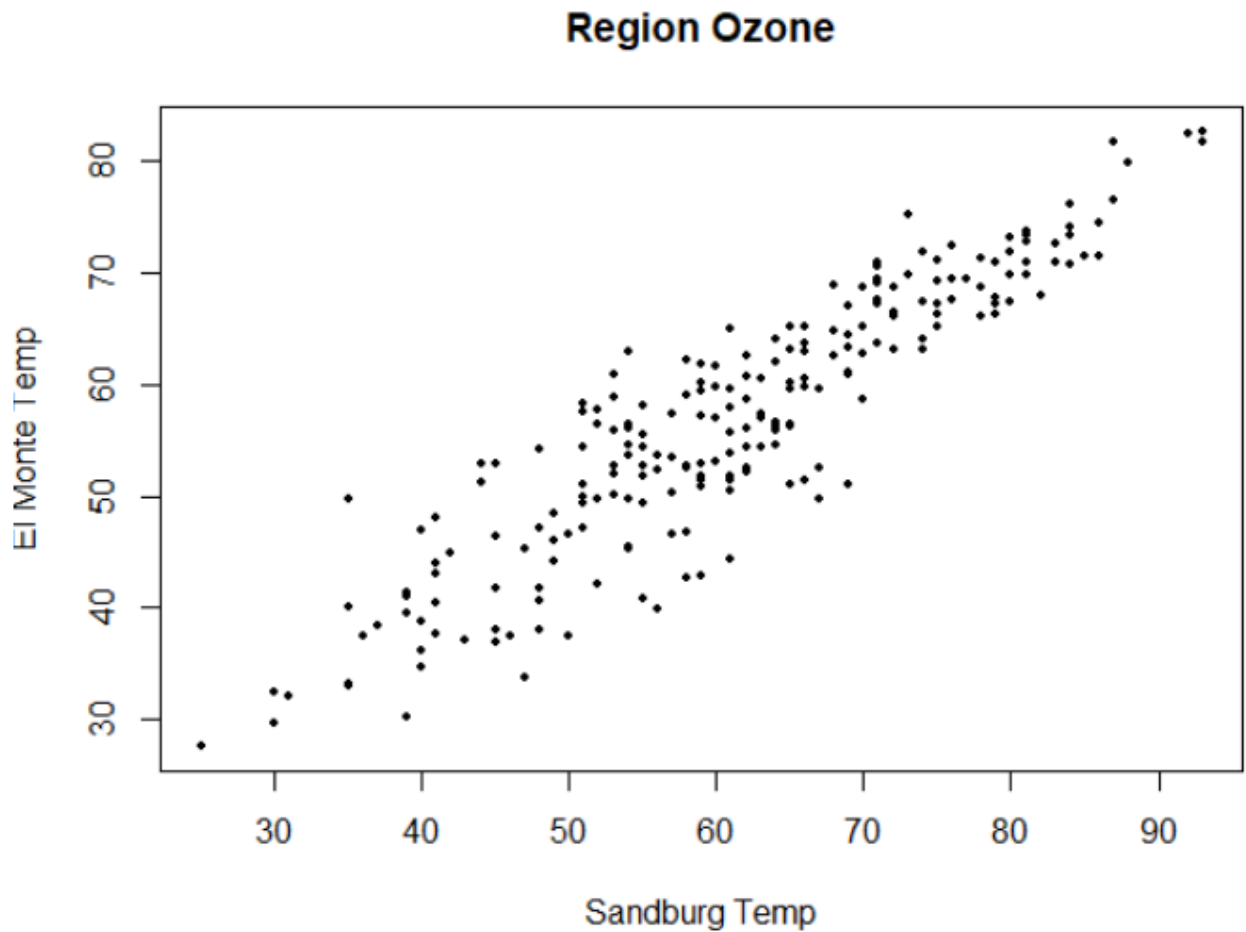
# CEX - 점의 크기

찍히는 점의 크기를 조절한다

```

38
39 #점크기 (cex)
40 plot(Ozone$V8,Ozone$V9,
41       xlab = "Sandburg Temp",
42       ylab = "EI Monte Temp",
43       main = "Region Ozone",
44       pch = 20,
45       cex = .8)

```



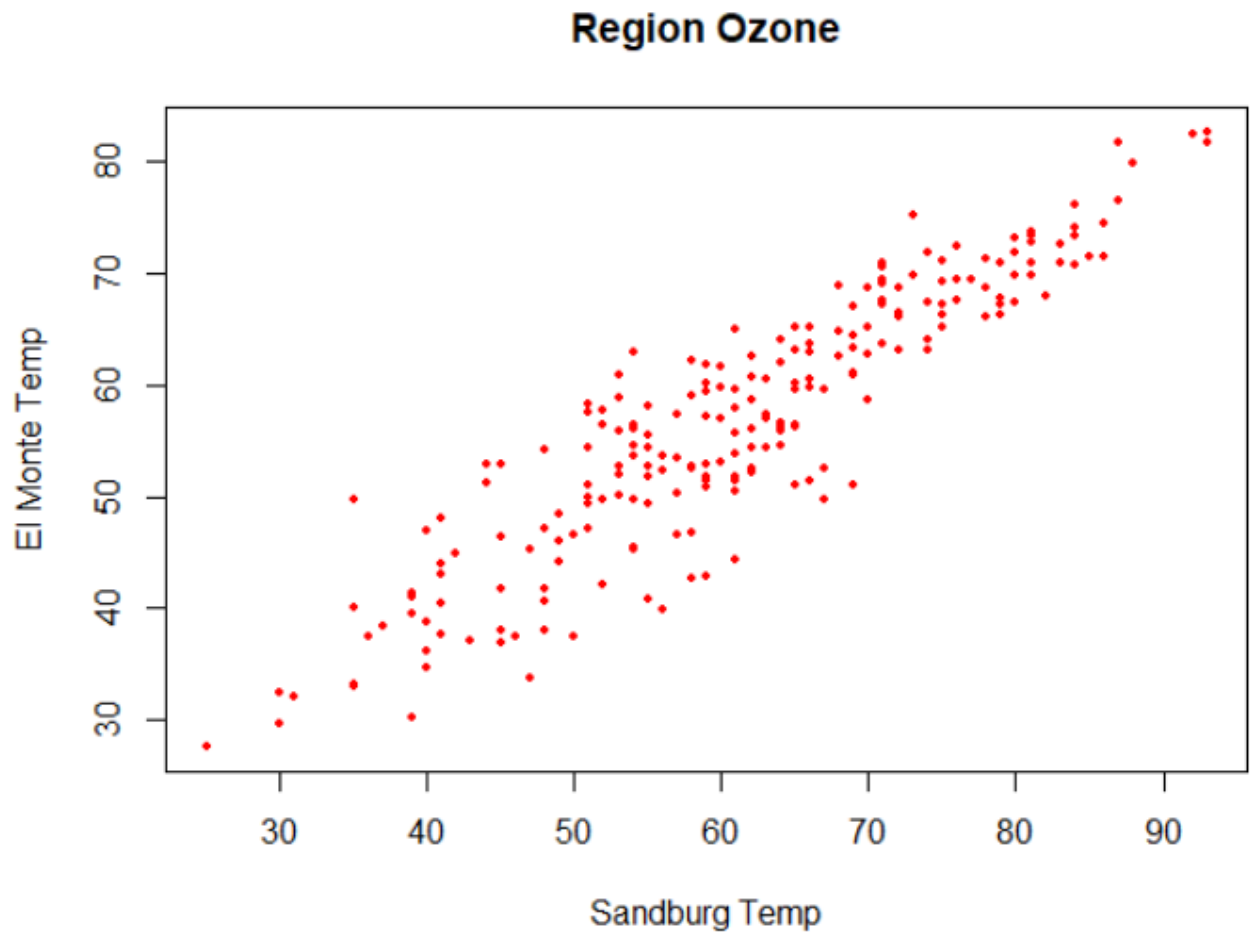
#점크기 (cex)

.8은 0.8이라는 뜻이다.

```

50
51 #점크기 (cex)
52 plot(Ozone$V8,Ozone$V9,
53       xlab = "Sandburg Temp",
54       ylab = "EI Monte Temp",
55       main = "Region Ozone",
56       pch = 20,
57       cex = .8,
58       col = "red")

```



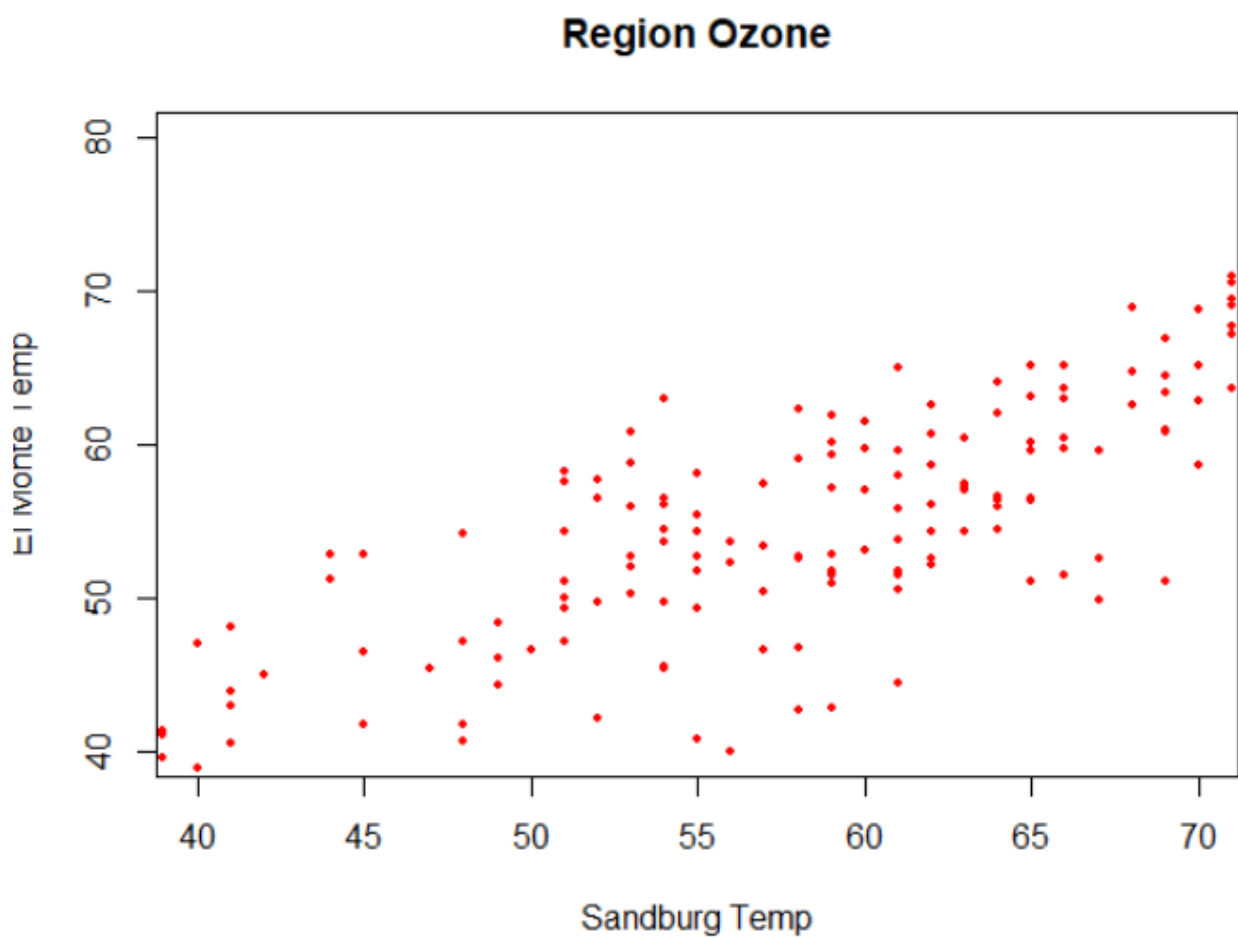
# xlim , ylim -좌표축의 범위

x축의 범위 y축의 범위를 지정한다.

```

59
60 # 좌표축의 범위 (xlim,ylim)
61 range(Ozone$v8, na.rm=T)
62 range(Ozone$v9, na.rm=T)
63
64 plot(Ozone$v8,Ozone$v9,
65       xlab = "Sandburg Temp",
66       ylab = "EI Monte Temp",
67       xlim = c(40,70),
68       ylim = c(40,80),
69       main = "Region Ozone",
70       pch = 20,
71       cex = .8,
72       col = "red")
73

```



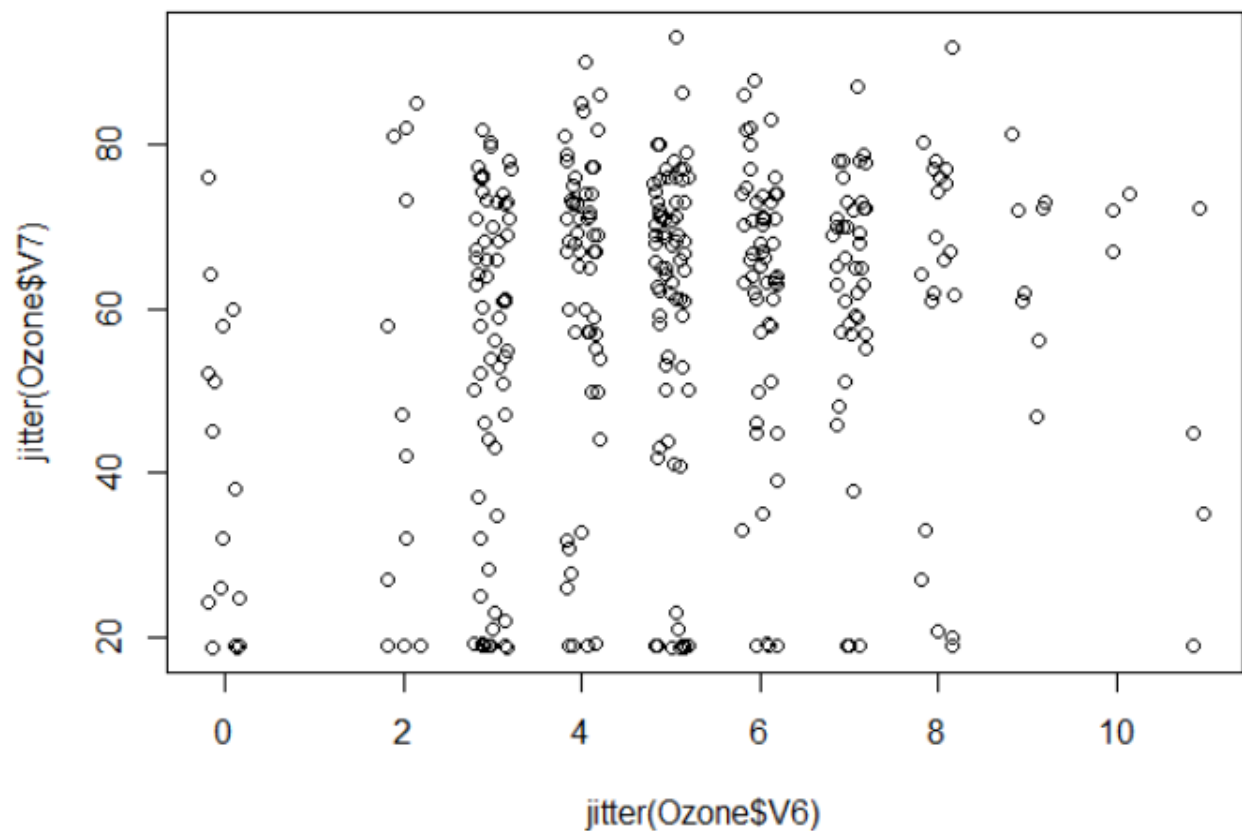
## #jitter - 노이즈 추가

지터Jitter는 데이터 값에 약간의 노이즈를 추가하는 방법을 말한다.  
노이즈를 추가하면 데이터 값이 조금씩 움직여서 같은 값을 가지는  
데이터가 그래프에 여러 번 겹쳐서 표시되는 현상을 막아준다.

구문

```
plot(jitter(x), jitter(y))
```

```
76 # 지터Jitter는 데이터 값에 약간의 노이즈를 추가하는 방법을 말한다.  
77 # 노이즈를 추가하면 데이터 값이 조금씩 움직여서 같은 값을 가지는  
78 # 데이터가 그래프에 여러 번 겹쳐서 표시되는 현상을 막아준다.  
79 plot(jitter(Ozone$V6), jitter(Ozone$V7))
```



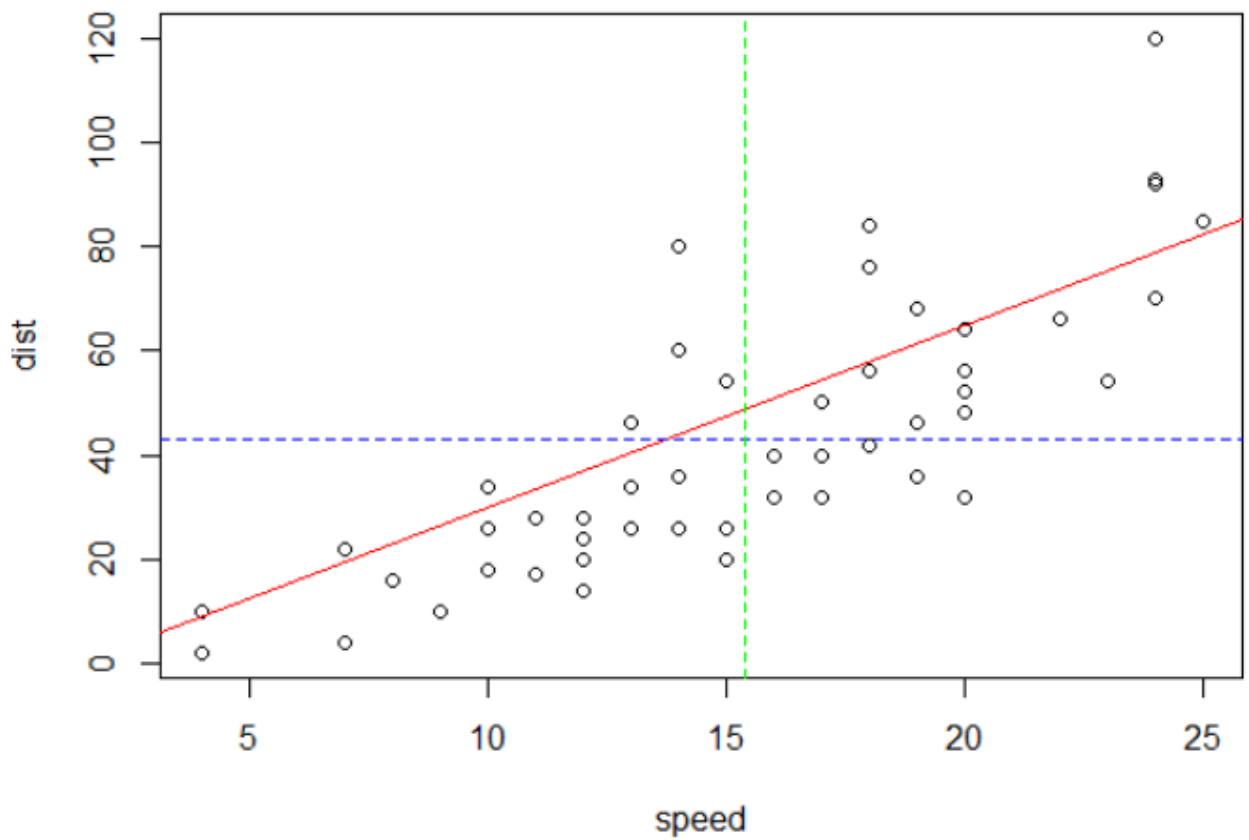


## #abline - 직선 긋기

- abline( )은  $y = ax + b$  형태의 직선이나  $y=h$  형태의 가로로 그은 직선 또는  $x=v$  형태의 세로로 그은 직선을 그래프에 그린다.

옵션 lty - 선 유형 / col - 선 컬러

```
91
92 #직선(abline)
93 # b0 : 절편 , b1 : 기울기
94 #  $y = b0 + b1x + e$ 
95
96 #  $dist = -5 + 3.5 * speed$ 
97 plot(cars)|
98 abline(a=-5 , b=3.5, col = "red" )
99
100 # 그래프에서 speed와 dist 평균까지 abline 표시한다면
101
102 #abline 으로 가로로(h) (cars$dist)평균값을 라인타입 2에 색깔을 파랑색으로 나타낸다.
103 abline(h=mean(cars$dist), lty=2, col="blue")
104
105 #abline 으로 세로로(v) (cars$speed)평균값을 라인타입 2에 색깔을 초록색으로 나타낸다.
106 abline(v=mean(cars$speed), lty=2, col="green")
107
```



## #선형 회귀

선형 회귀는 종속 변수(또는 반응 변수)를 독립 변수(또는 설명 변수)에 의해 설명하는 모델을 다루는 회귀 분석(Regression Analysis)의 한 종류다.

구문

```
#lm(formula, data = )  
#lm(종속변수 ~ 독립변수, data = )  
  
lm(  
  formula, # 종속 변수 ~ 독립 변수 형태로 지정한 포뮬러  
  data    # 포뮬러를 적용할 데이터. 보통 데이터 프레임  
)
```

```

111 #선형회귀(lm)
112
113 #종속변수 - 독립 변수의 변화에 따라 어떻게 변화하는지 알고 싶은 값 / 찾고자하는 값
114 #독립변수 - 연구자가 의도적으로 변화시키는 변수로서 실험이나 모델링에서 사용되는 값
115
116 #lm(formula, data = )
117 #lm(종속변수 ~ 독립변수, data = )
118
119 #dist = -17.579 + 3.932 * 4
120 abs(-1.851)
121 head(cars)
122 |
123 car_model <- lm(dist ~ speed, data = cars)
124 car_model
125
126 #이걸 생성할 경우 평균을 지나는 선이 하나 생긴다.
127 abline(car_model)
128

```

```

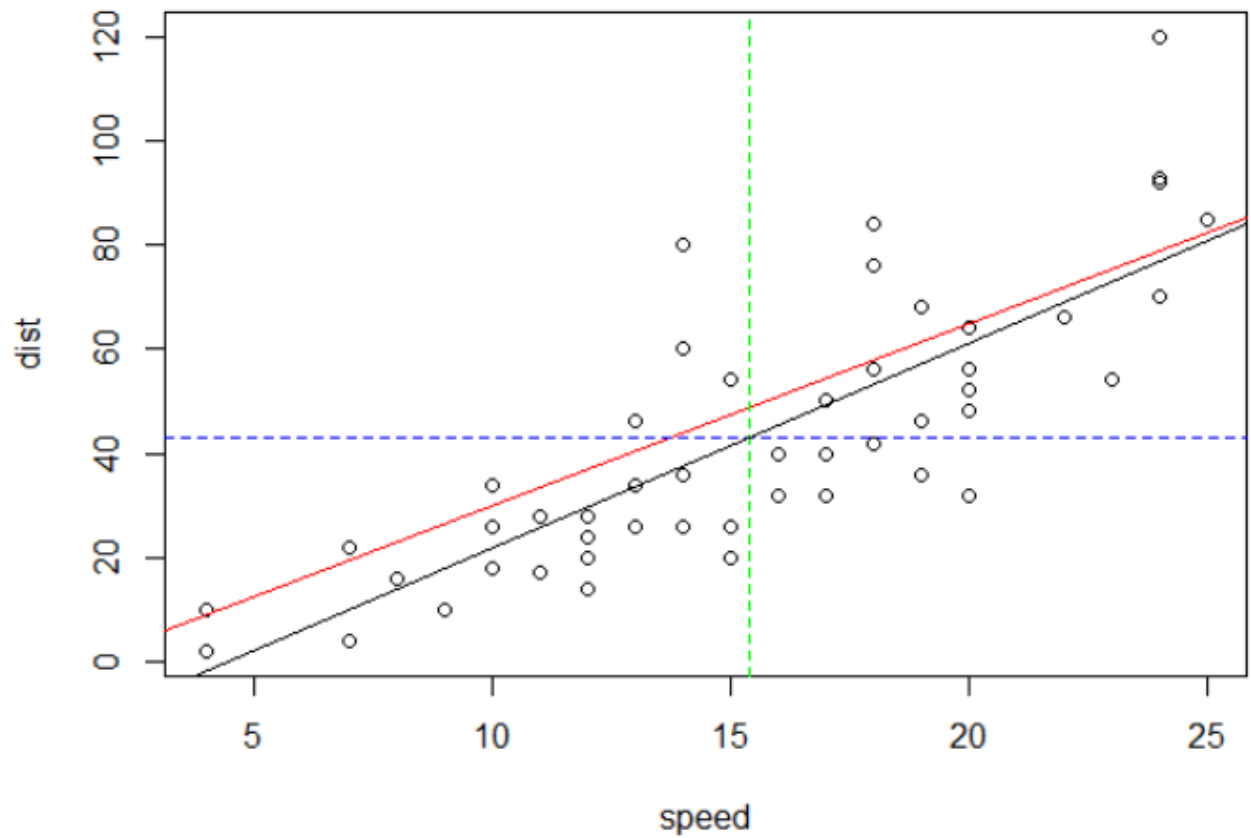
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
> car_model <- lm(dist ~ speed, data = cars)
> car_model

Call:
lm(formula = dist ~ speed, data = cars)

Coefficients:
(Intercept)      speed
   -17.579        3.932

```

dist와 speed 간의 관계 = coefficients



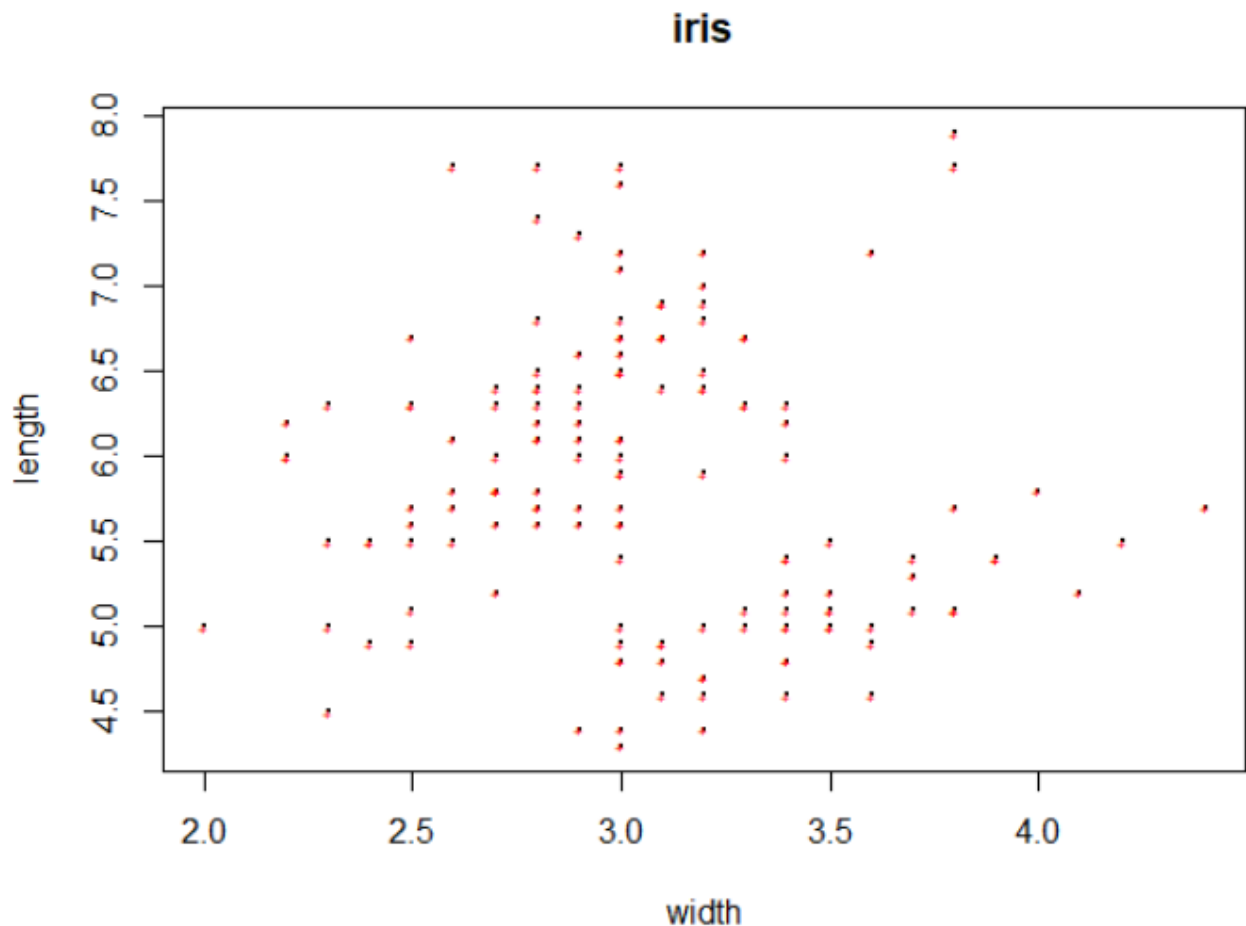
## #points() - 점으로 찍기

`points()`는 점을 그리는 함수다. `plot()`을 연달아 호출하는 경우 매번 새로운 그래프가 그려지는 것과 달리 `points()`는 이미 생성된 `plot`에 점을 추가로 그려준다.

```

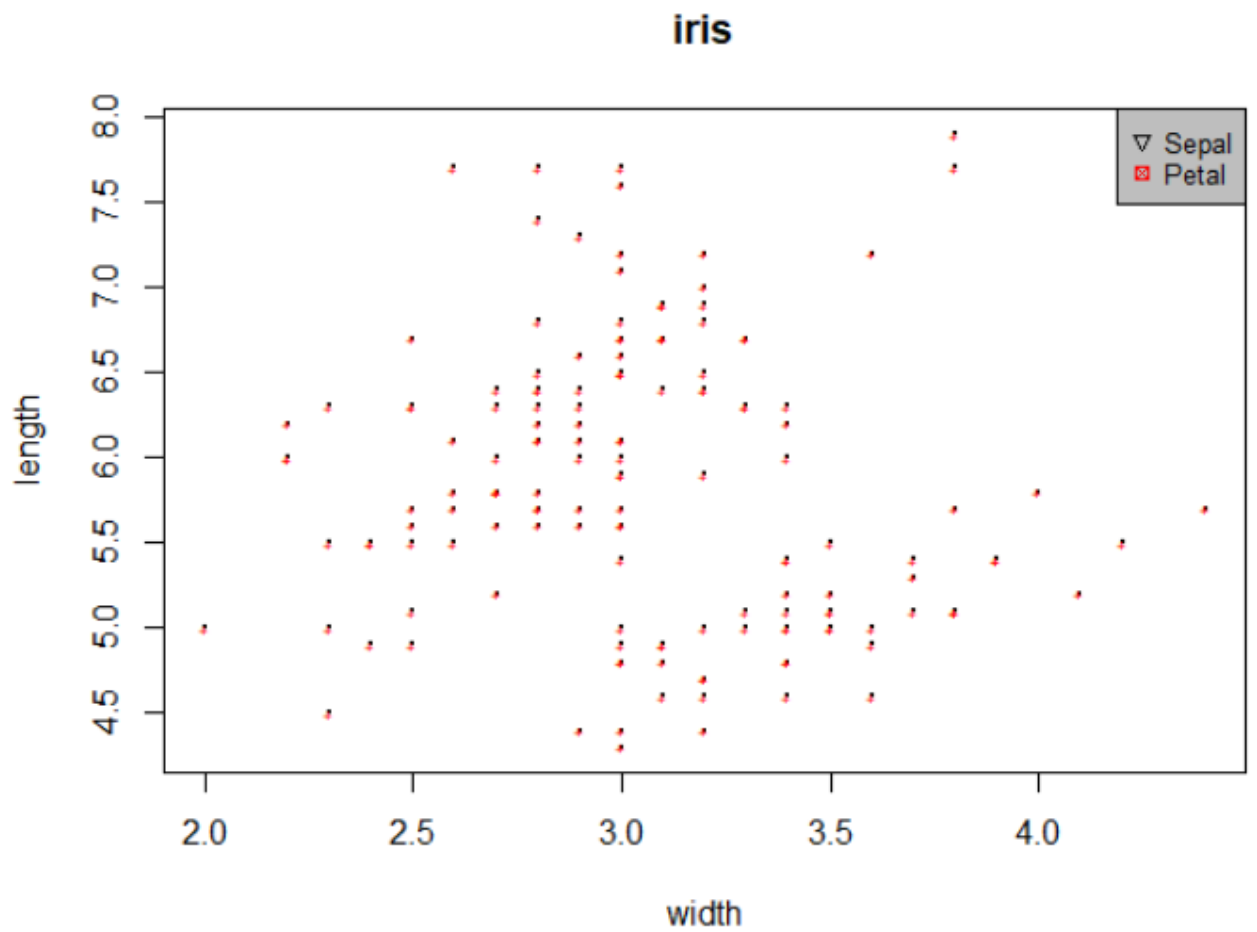
133 #점(points)
134 #iris, Sepal.width, Sepal.Length → plot
135 iris
136 plot(iris$Sepal.Width, iris$Sepal.Length,
137       cex=.5,
138       pch=20,
139       xlab="width",ylab="length",
140       main="iris")
141
142 points(iris$Sepal.Width, iris$Sepal.Length,
143        cex=.5,
144        pch="+",
145        col = "#FF0000")
146
147 #명시적으로 그래프 그리기
148 with(iris,
149     {
150         plot(Sepal.Width, Sepal.Length,
151              cex=.5,
152              pch=20,
153              xlab="width",ylab="length",
154              main="iris")
155         points(Sepal.Width, Sepal.Length,|
156                cex=.5,
157                pch="+",
158                col = "#FF0000")
159     }
160
161 )

```



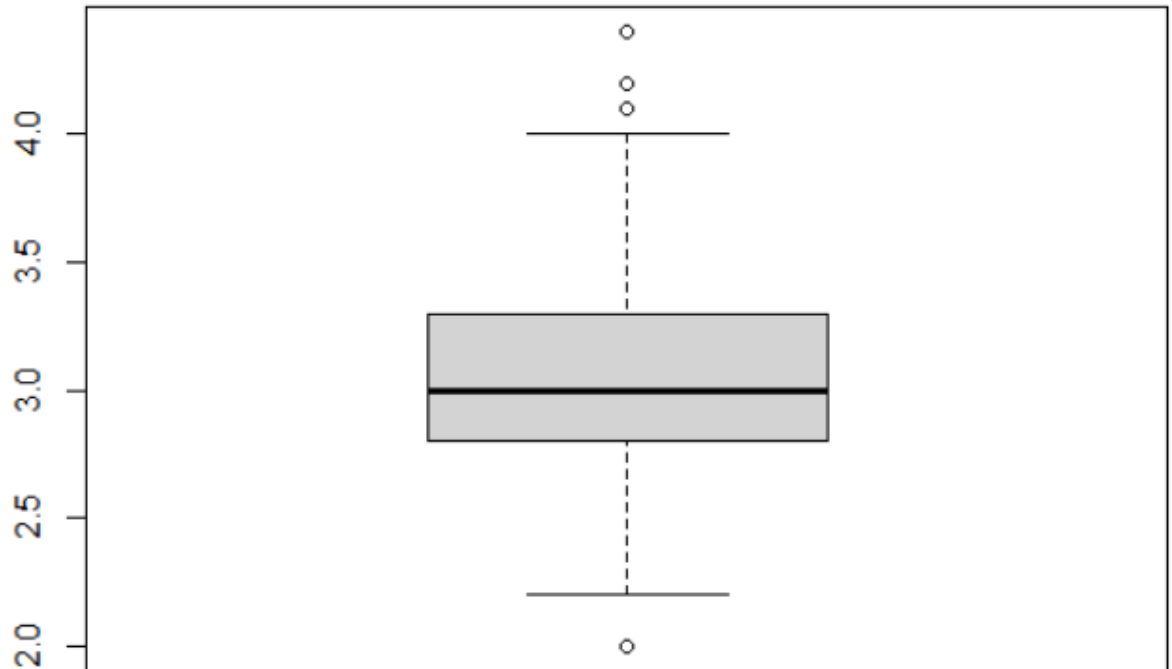
## #legend - 범례 표시

```
#명시적으로 그래프 그리기
with(iris,|
{
  plot(Sepal.Width, Sepal.Length,
       cex=.5,
       pch=20,
       xlab="width", ylab="length",
       main="iris")
  points(Sepal.Width, Sepal.Length,
        cex=.5,
        pch="+",
        col = "#FF0000")
  legend("topright", #범례표시
        legend = c("Sepal", "Petal"),
        pch = c(6,7),
        cex=.8,
        col=c("black", "red"),
        bg = "gray")
}
```



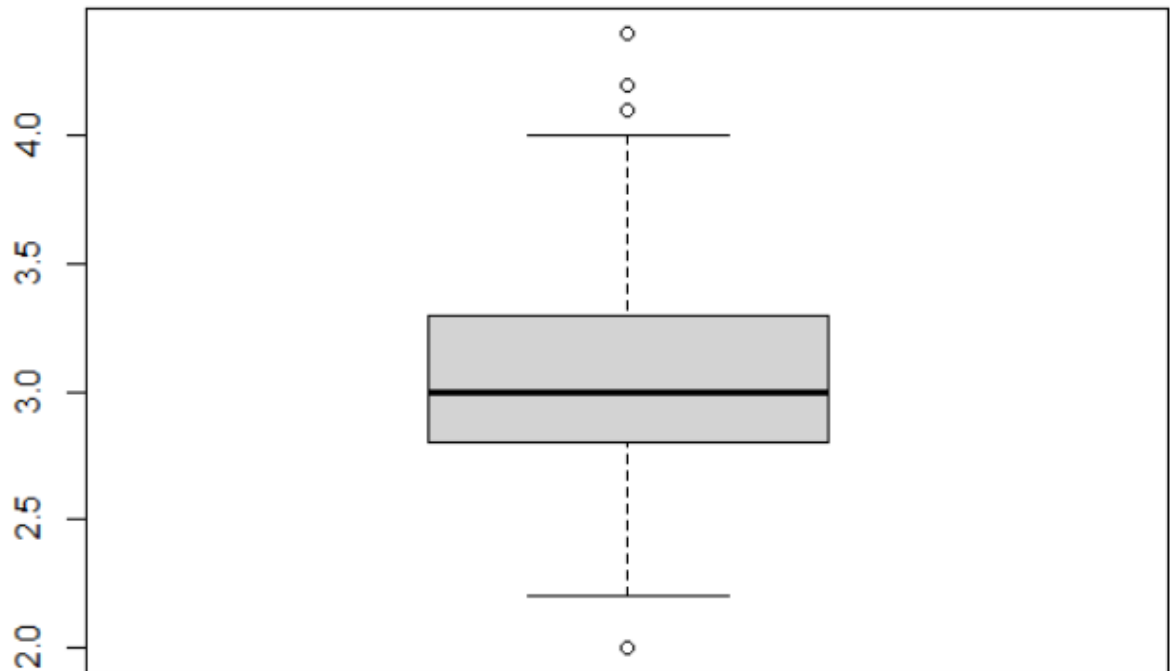
#boxplot - 상자수염그림

```
171 #boxplot - 상자수염그림
172
173 summary(iris$Sepal.Width)
174
175 #양끝에 마지막 선이 넘는건 이상치라고 본다.
176 boxplot(iris$Sepal.Width)
177
```



```
171 #boxplot - 상자수염그림
172
173 summary(iris$Sepal.Width)
174
175 #양끝에 마지막 선이 넘는건 이상치라고 본다.
176 boxplot(iris$Sepal.Width)
177
178 # IQR(3사분위수 - 1사분위수)
179 # 1사분위수 2.8
180 # 3사분위수 3.3
181 # median 3.0
182 # whisker 값을 계산한다면
183 # 중앙값 - 1.5 * IQR(lower whisker)
184 # 중앙값 + 1.5 * IQR(upper whisker)
185 3.0-1.5*(3.300 - 2.800)
186 3.0+1.5*(3.300 - 2.800)
```



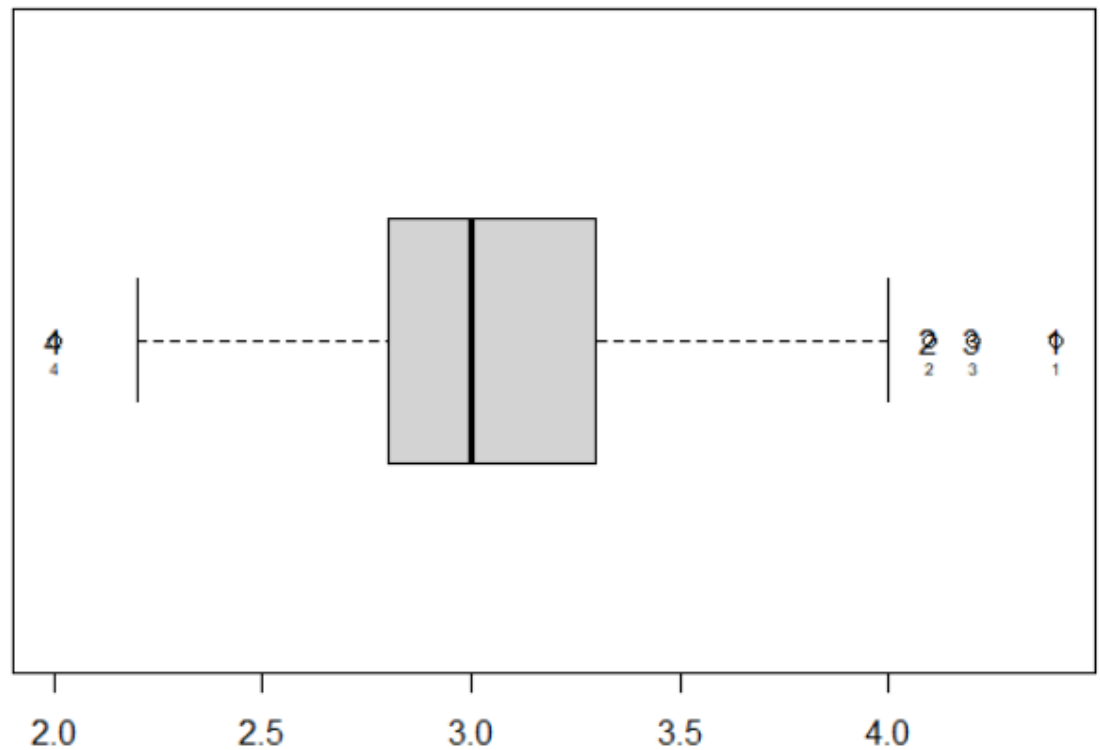


#boxplot horizontal =T - 가로로 바꾸기

```

170
171 #boxplot - 상자수염그림
172
173 summary(iris$Sepal.Width)
174
175 #양끝에 마지막 선이 넘는건 이상치라고 본다.
176 boxplot(iris$Sepal.Width)
177
178 # IQR(3사분위수 - 1사분위수)
179 # 1사분위수 2.8
180 # 3사분위수 3.3
181 # median 3.0
182 # whisker 값을 계산한다면
183 # 중앙값 - 1.5 * IQR(lower whisker)
184 # 중앙값 + 1.5 * IQR(upper whisker)
185 3.0-1.5*(3.300 - 2.800)
186 3.0+1.5*(3.300 - 2.800)
187
188 boxplotStats <- boxplot(iris$Sepal.Width,
189                          horizontal=T)
190
191 boxplotStats
192
193 text(boxplotStats$out, #이산치 값을 텍스트로 뿌려달라.
194       rep(1, NROW(boxplotStats$out)), #반복
195       pos = 1, cex=.5)
196

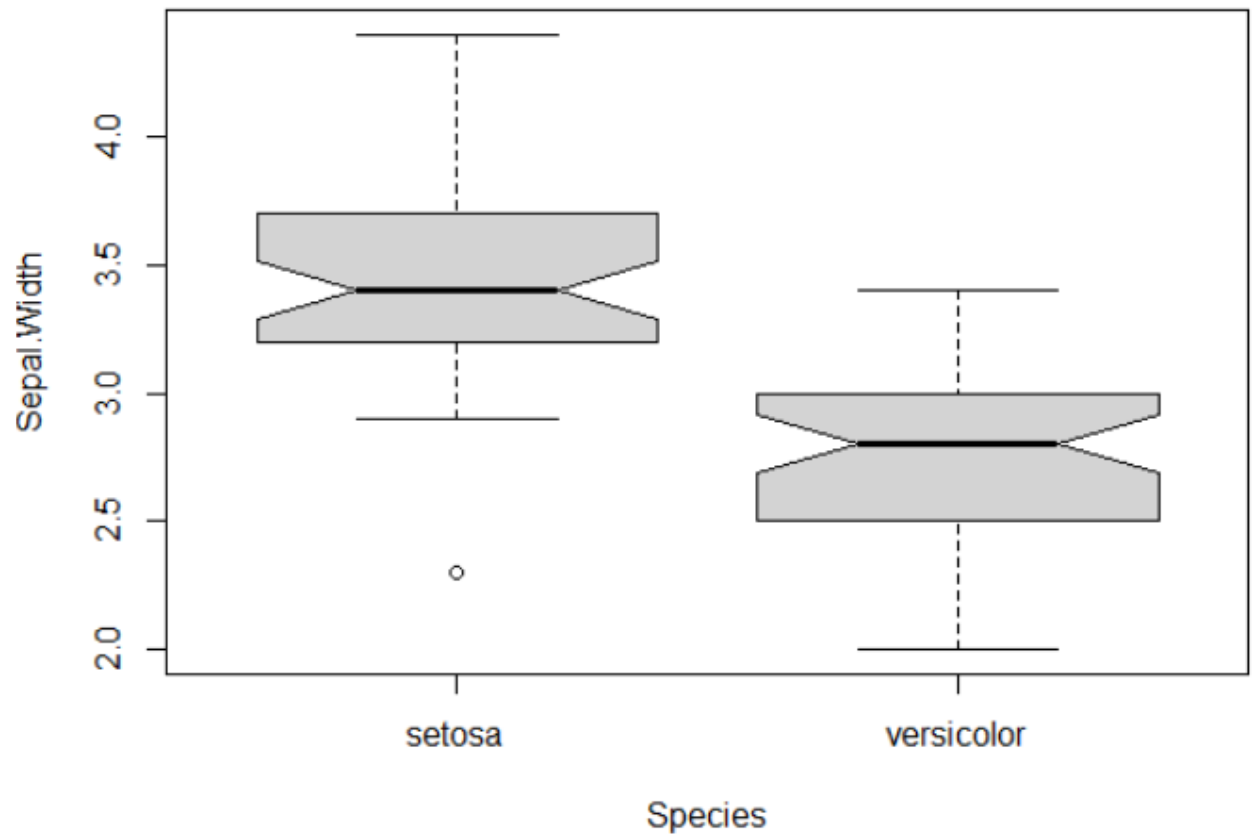
```



boxplot( formula)

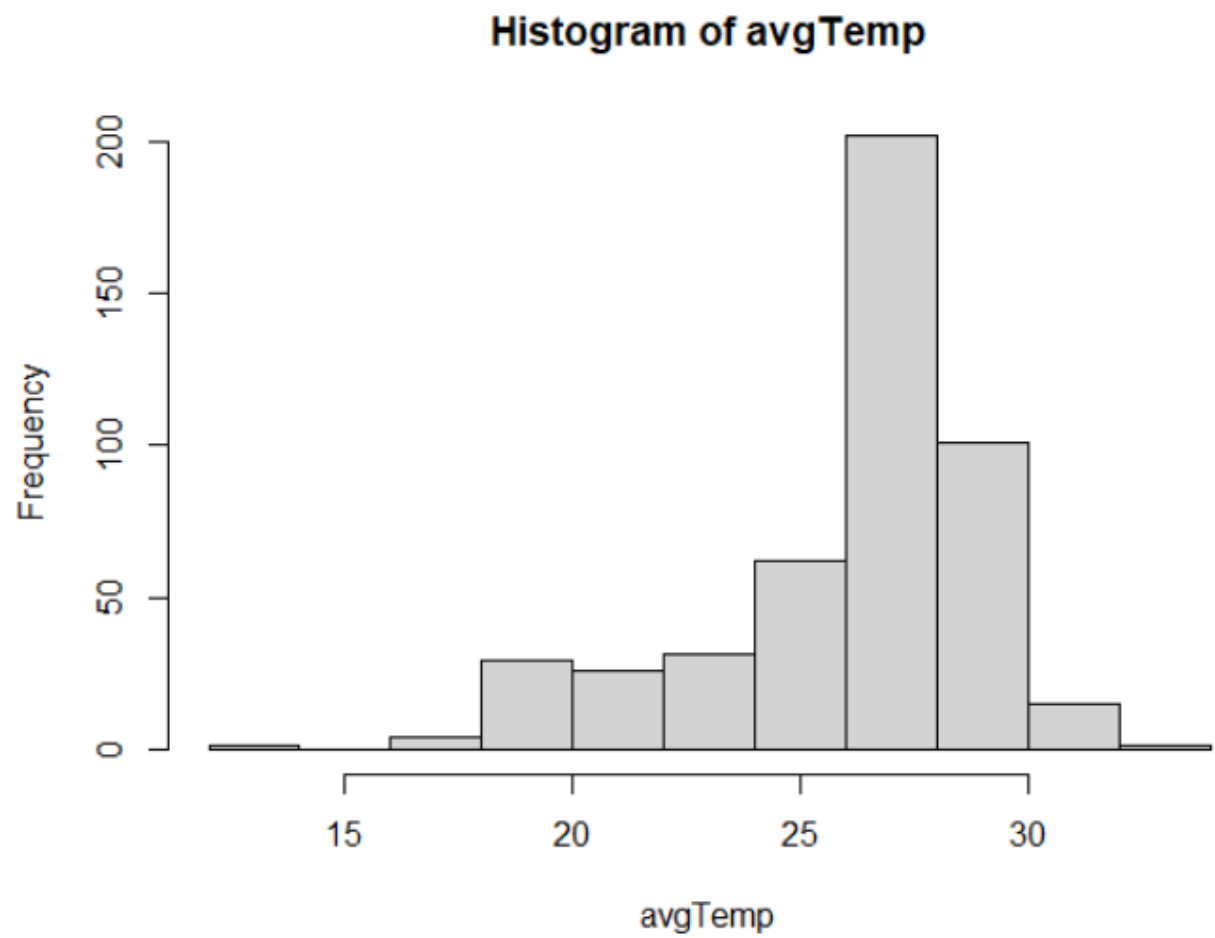
formula, # y ~ grp의 형식으로 y는 분포를 그릴 값, grp는 값들을 그룹 짓는 변수다.

```
198 # iris의 setosa 종과 versicolor 종의
199 # Sepal.Width에 대한 상자 그림을 그려보자
200
201 sv <- subset(iris, Species=="setosa" | Species=="versicolor")
202 str(sv)
203 sv$Species <- factor(sv$Species) #종을 두개만 사용하는데 레벨이 3개면 안되서 factor를 새롭게 만들어서 넣어버린다.
204 levels(sv$Species)
205 boxplot(Sepal.Width ~ Species, data=sv, notch=TRUE) #하나가 아닌 여러개 일때는 포물러 형식을 쓴다.
206
```

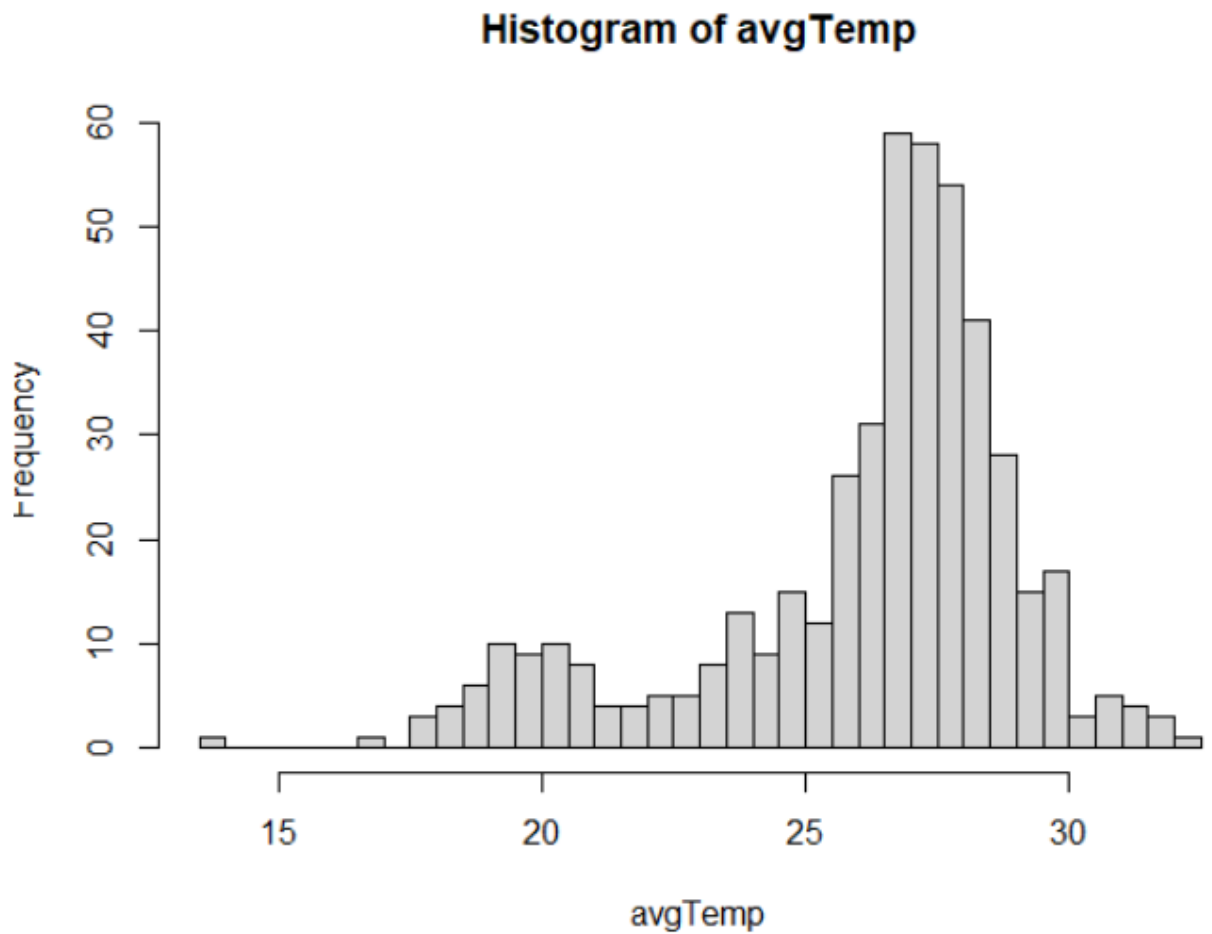


#hist 그래프

```
209 # hist(빈도수 기반 , 밀도 기반)
210 # service_date_visualization_region_weather.csv
211
212 install.packages("readxl")
213 library(readxl)
214 region <- read.csv(file.choose())
215
216 str(region)
217 avgTemp <- region$평균기온
218 hist(avgTemp)
```



```
219  
220 hist(avgTemp, breaks = 30 ) #구간 나누기  
221 |
```



#### 'R' 카테고리의 다른 글

[R] R 을 활용한 데이터 탐색(Exploratory Data Analysis)

[R] R ggplot 사용법 (데이터 시각화 도구)

**[R] R 에서 사용되는 기본적인 시각화 그래프-2**

[R] R 에서 사용되는 기본적인 시각화 그래프

[R] R 데이터 가공을 위한 기본적인 함수

[R] R 사용자 정의 함수(FUNCTION)와 데이터 전처리를 위한 기본적인 함수

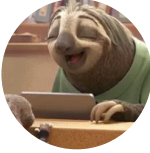
boxplot

hist 그래프

points

산점도

선형회귀



꾸까꾸

혼자 끄적끄적하는 블로그 입니다.

