

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

노트북: blog

만든 날짜: 2020-10-03 오후 5:49

URL: <https://continuous-development.tistory.com/40?category=793392>

R

[R] R 에서 사용되는 기본적인 시각화 그래프

2020. 7. 28. 02:51 수정 삭제 공개

#시각화

변수 구분(이산 VS 연속)

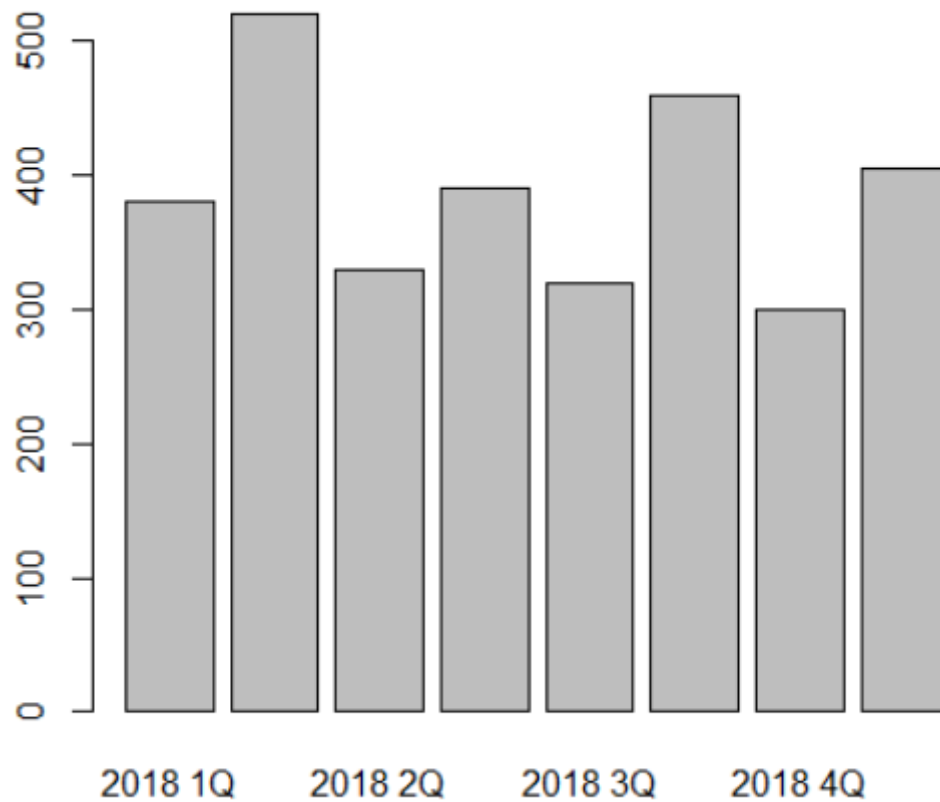
이산형 변수 : 변수가 가질 수 있는 값이 끊어진 변수

-명목변수 : 변수에 들어있는 값이 의미가 없는 것 / -순위 변수 : 변수에 들어 있는 값이 순위를 가지는 변수

막대 , 점 , 파이

##barplot - 기본적인 막대차트

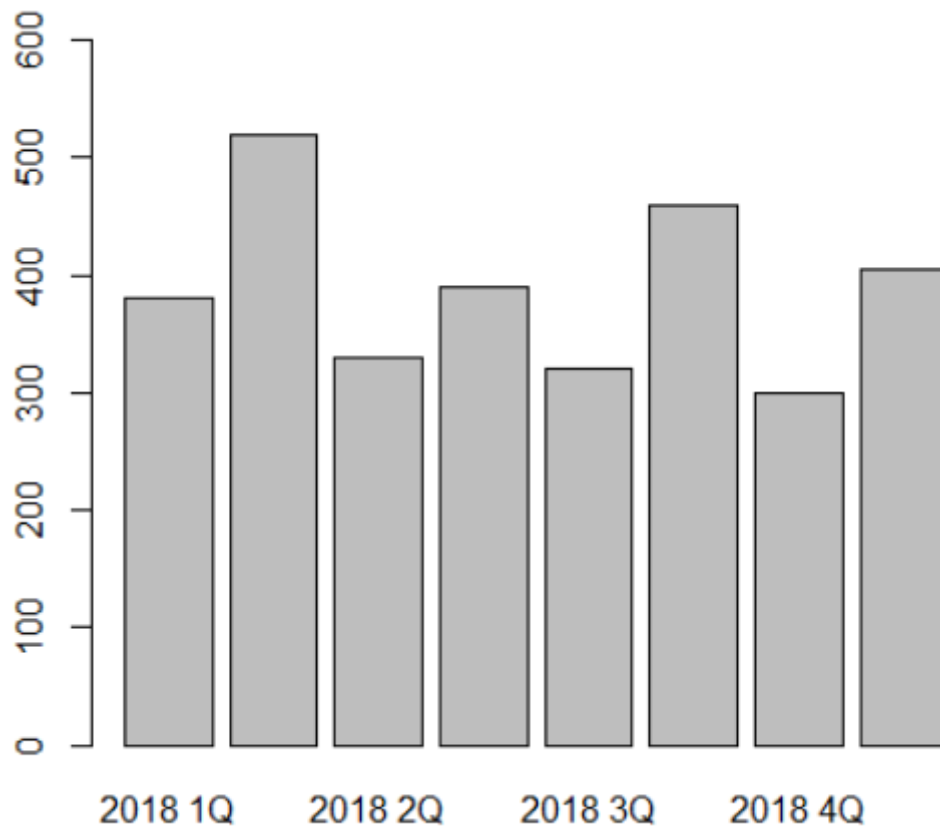
```
> ##시각화
> # 변수 구분( 이산 VS 연속)
> # 이산형 변수 : 변수가 가질 수 있는 값이 끊어진 변수
> # -명목변수 : 변수에 들어있는 값이 의미가 없는 것 / -순위 변수 : 변수에 들어 있는 값이 순위를 가지는 변수
> # 막대 , 점 , 파이
> char_data <-c(380,520,330,390,320,460,300,405)
> names(char_data) <-c("2018 1Q","2019 1Q", "2018 2Q","2019 2Q","2018 3Q","2019 3Q","2018 4Q","2019 4Q")
> char_data
2018 1Q 2019 1Q 2018 2Q 2019 2Q 2018 3Q 2019 3Q 2018 4Q 2019 4Q
    380    520    330    390    320    460    300    405
> range(char_data)
[1] 300 520
> max(char_data)
[1] 520
> length(char_data)
[1] 8
>
> # 막대차트:: barplot()
> barplot(char_data)
> |
```



#ylim 옵션- 차트의 범위 조절

아래와 같은 경우 y축을 0~600까지로 limit해 범위를 잡았다.

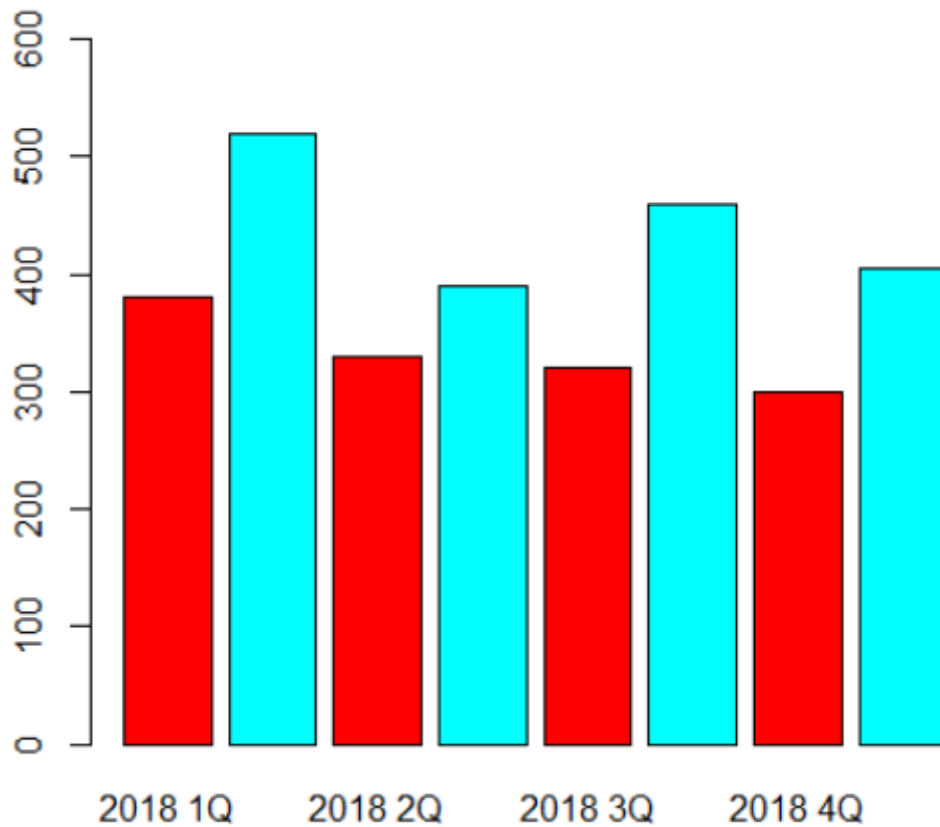
```
398 # 막대차트 :: barplot()
399 barplot(char_data)
400
401 barplot(char_data, #ylim으로 범위를 조절 할 수 있다.
402           ylim=c(0,600))
403
```



#col 옵션- 컬러 옵션이다.

rainbow로 할 경우 2가지 색을 나타낸다. 이 숫자가 늘어날수록 색이 다양해진다.

```
404 barplot(char_data,  
405           ylim=c(0,600),  
406           col = rainbow(2)) #col 으로 컬러를 조절 할 수 있다.  
407
```

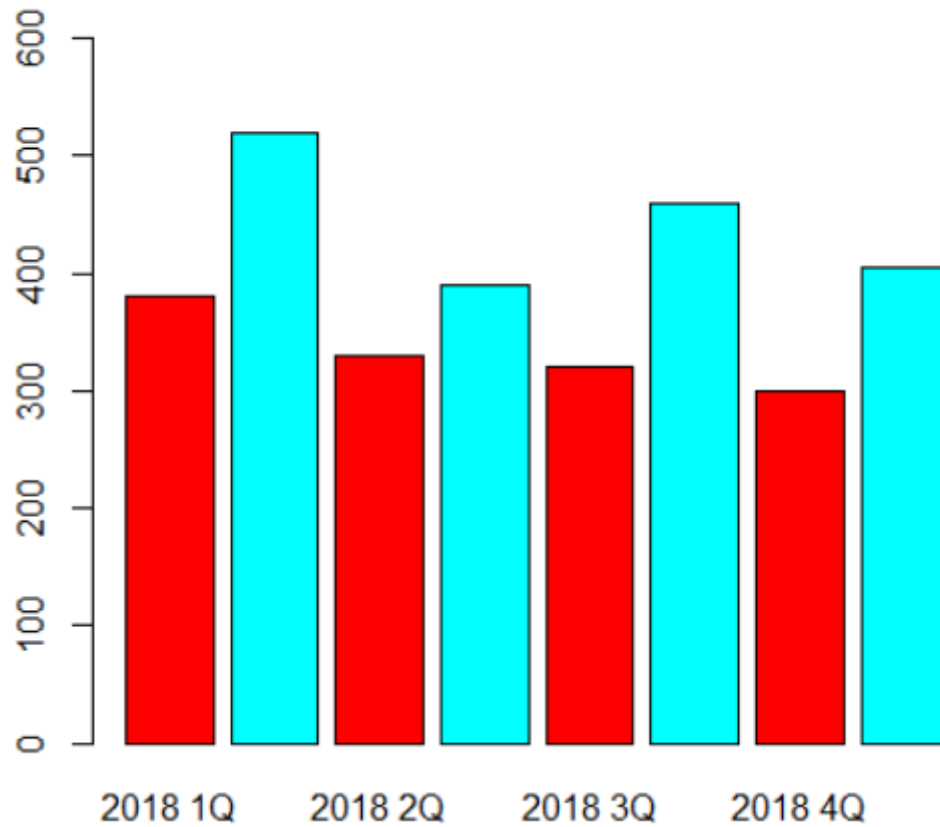


#main 옵션- 그래프 명을 나타낸다.

그래프 중간에 있는 그래프 명을 작성했다

```
407  
408 barplot(char_data,  
409         ylim=c(0,600),  
410         col = rainbow(2),  
411         main = "2018 VS 2019 분기매출") #그래프 명을 만든다.  
412
```

2018 VS 2019 분기매출

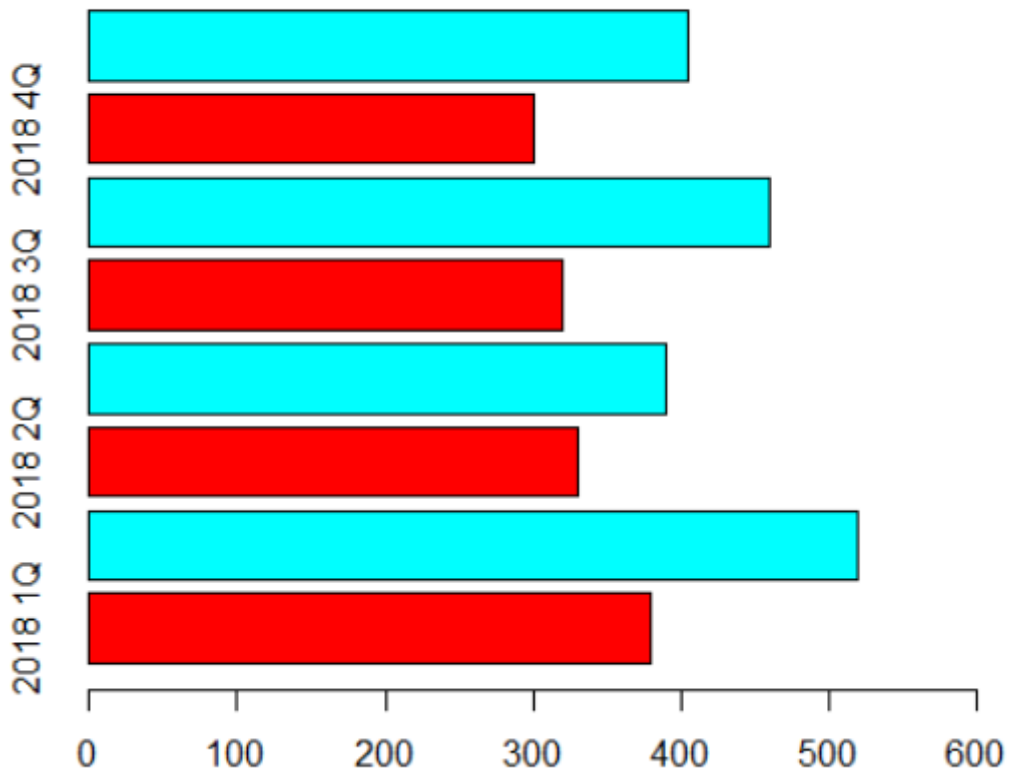


#horiz 옵션- 세로형 그래프를 가로축으로 바꿔준다..

그래프를 바꿔줄때는 ylim도 xlim으로 같이 바꿔줘야한다. 이건 y축의 limit을 정해준다.

```
412  
413 barplot(char_data,  
414         xlim=c(0,600), # 가로막대일때는 이 부분도 수정해줘야 한다.  
415         col = rainbow(2),  
416         main = "2018 VS 2019 분기매출",  
417         horiz = T) #가로 막대로 만든다.  
418
```

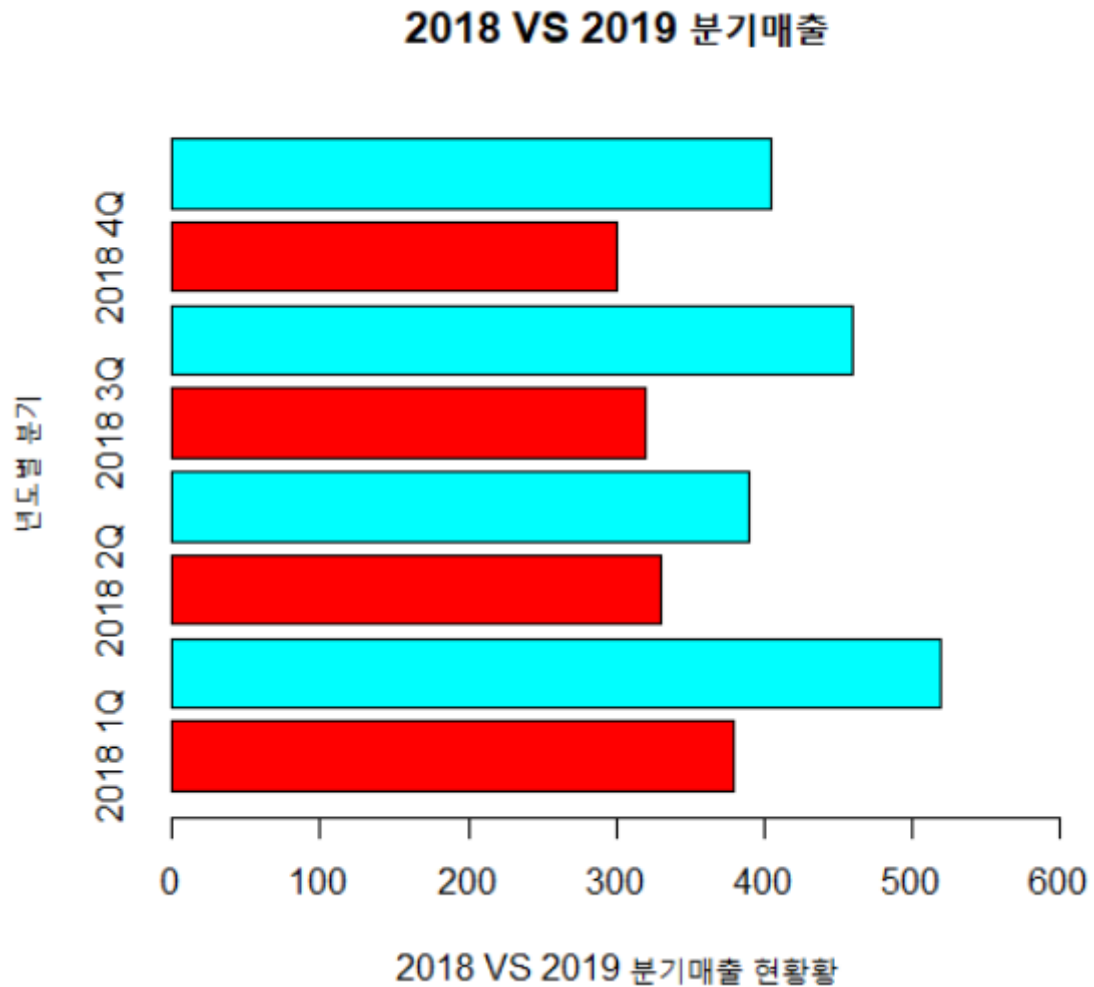
2018 VS 2019 분기매출



#label 옵션 - 라벨을 작성해준다.

이것 또한 x축 y축에 대한 라벨이 있다.

```
419 barplot(char_data,  
420         xlim=c(0,600), # 가로막대일때는 이 부분도 수정해줘야 한다.  
421         col = rainbow(2),  
422         main = "2018 VS 2019 분기매출",  
423         horiz = T,  
424         ylab = "년도별 분기", #label 을 만들 수 있다.  
425         xlab = "2018 VS 2019 분기매출 현황") #label 을 만들 수 있다.
```



##dotchart - dotplot를 그리는 함수

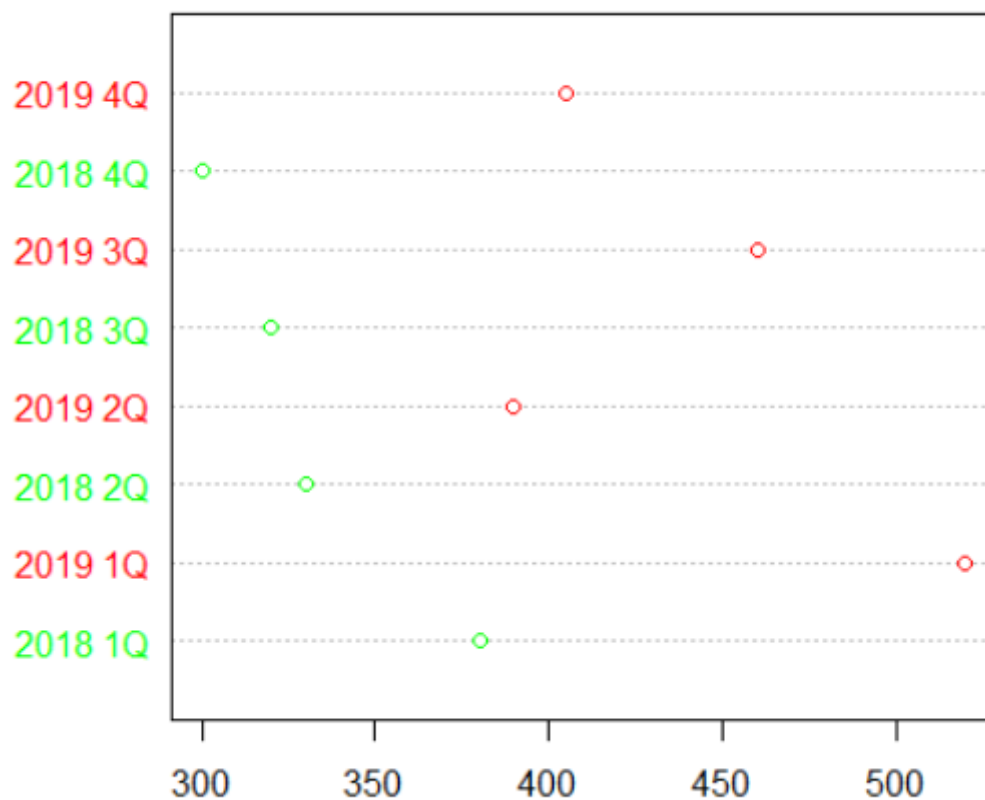
#col 옵션- 컬러 옵션이다.

col로 에서 컬러를 지정해준다

```

428 # dot chart
429 ?dotchart
430
431 dotchart(char_data)
432 dotchart(char_data,
433           col = c("green", "red"))
434

```

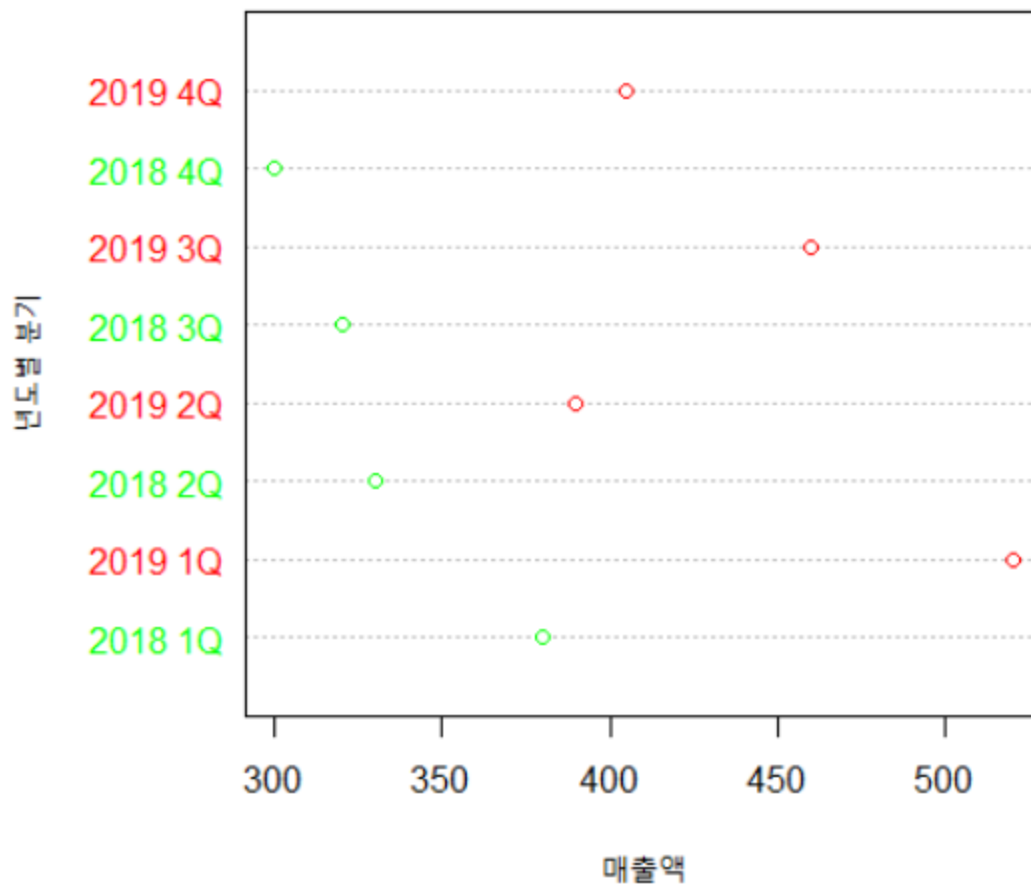


#lab 옵션- 라벨 옵션이다.


```

435
436 dotchart(char_data,
437           col = c("green", "red"),
438           xlab = "매출액",
439           ylab = "년도별 분기")
440 |

```



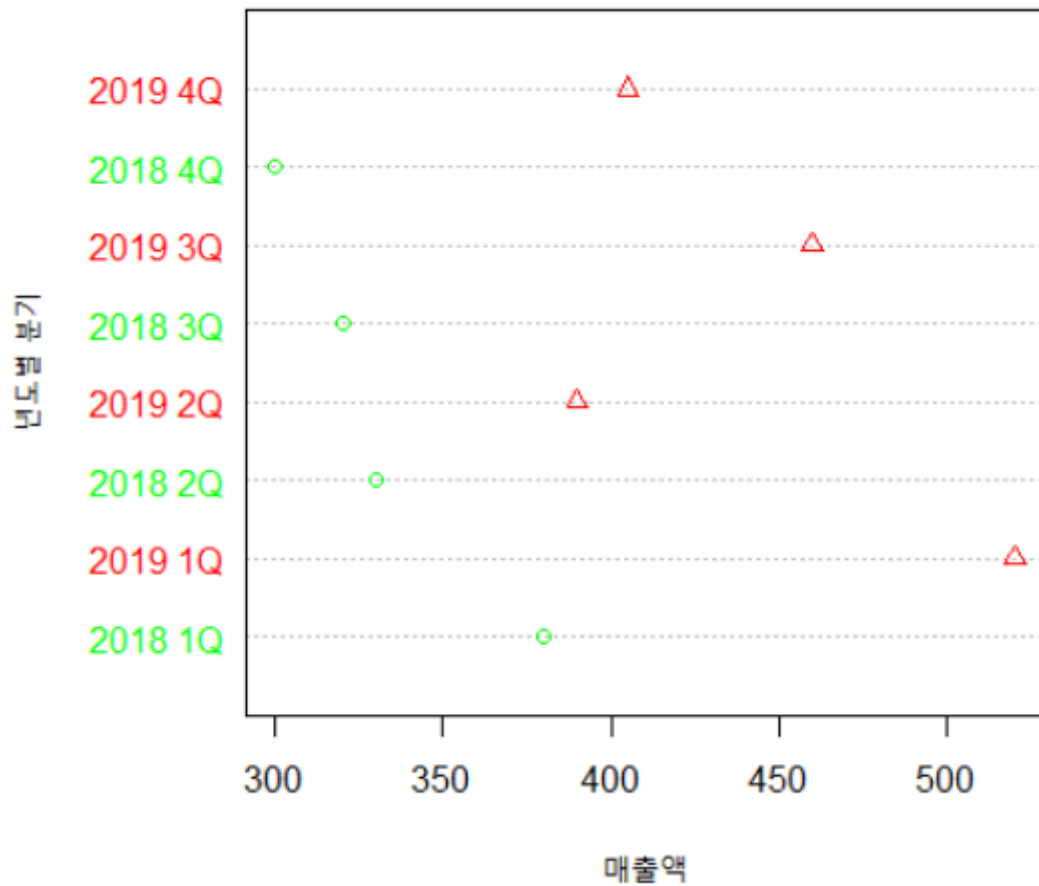
#pch 옵션- 찍히는 모양을 정해준다.

아래와 같은 경우 pch가 2개여서 두가지를 지정해줬고 이 값에 따라서 찍히는 모양이 달라진다.

```

440
441 dotchart(char_data,
442           col = c("green", "red"),
443           xlab = "매출액",
444           ylab = "년도별 분기",
445           pch = 1:2) # 번호별로 모양이 정해져있어서 아래와 같이 삼각형모양으로 나온다
446

```



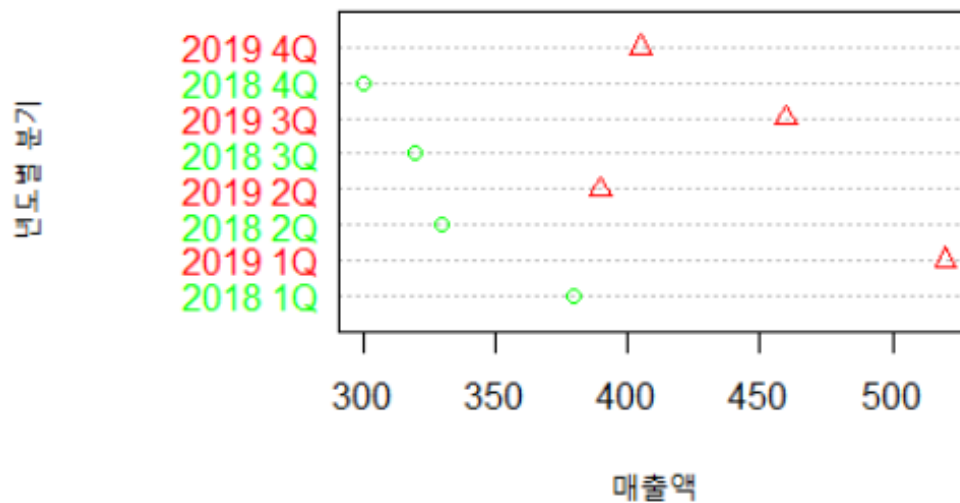
#cex옵션- 찍히는 모양의 크기를 정해준다.

이 값에 따라서 크기가 달라진다.

```

447
448 dotchart(char_data,
449           col = c("green", "red"),|
450           xlab = "매출액",
451           ylab = "년도별 분기",
452           pch = 1:2,
453           cex = 1) # 그래프에 찍히는 모양의 크기를 정해준다.
454

```



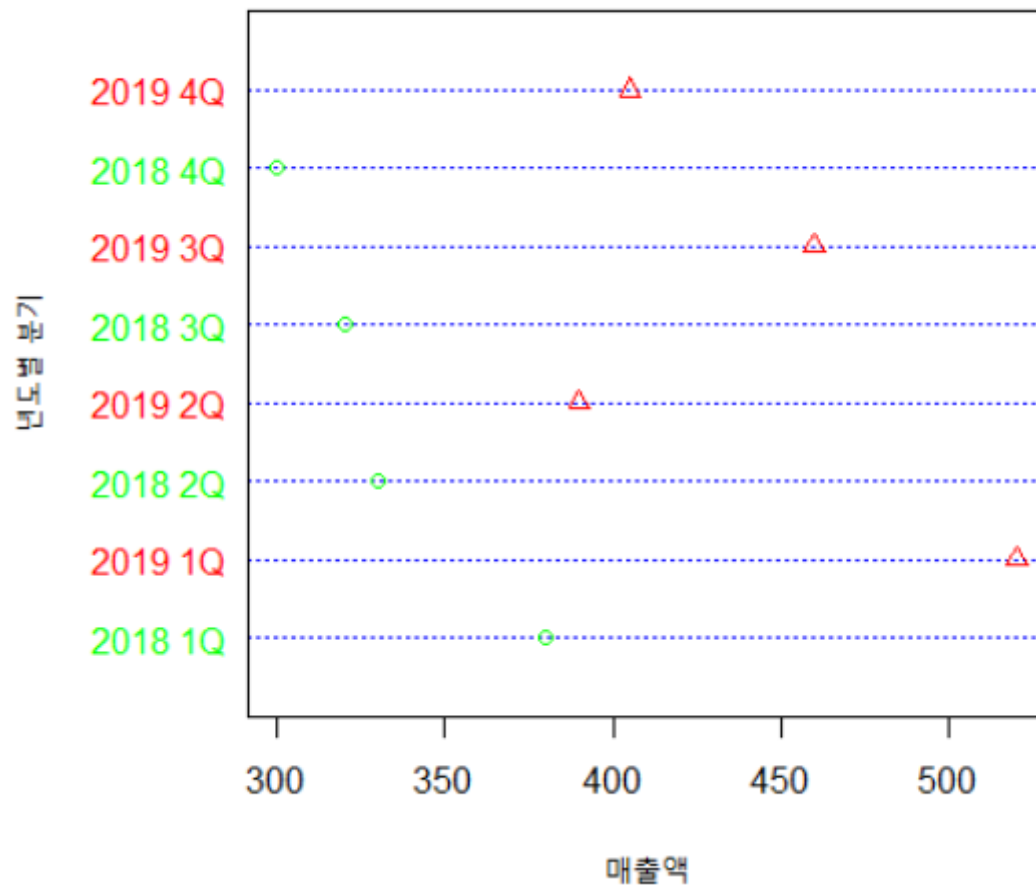
#lcolor 옵션 - 점선의 색을 나타낸다.

간격에 따른 점선의 색을 바꿔준다.

```

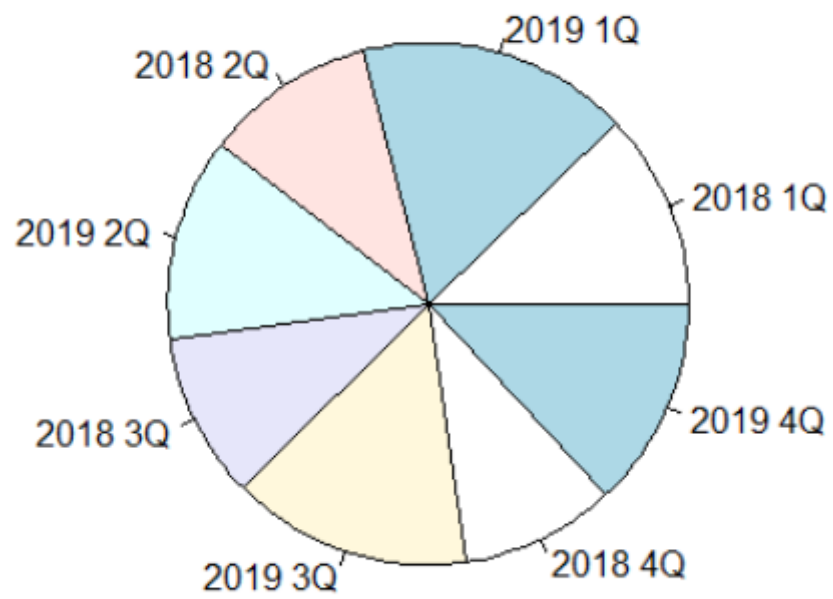
455 dotchart(char_data,
456           col = c("green", "red"),
457           xlab = "매출액",
458           ylab = "년도별 분기",
459           pch = 1:2,
460           cex = 1,
461           lcolor = "blue") # 그래프의 간격의 점선 색깔을 정한다.
462
463

```

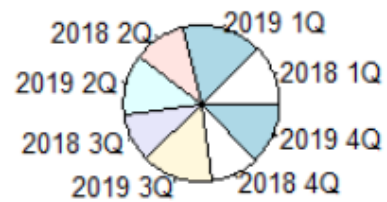
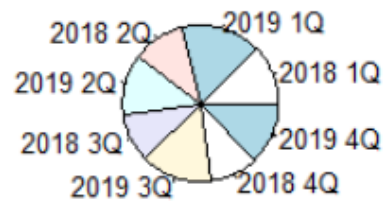
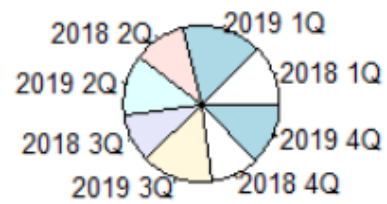
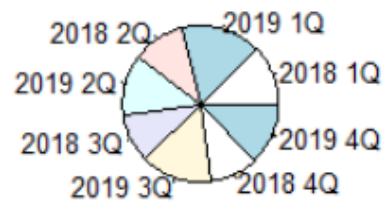


##PIE()- PIE 차트를 그린다..

```
463  
464 # pie chart  
465 pie(char_data)  
466
```

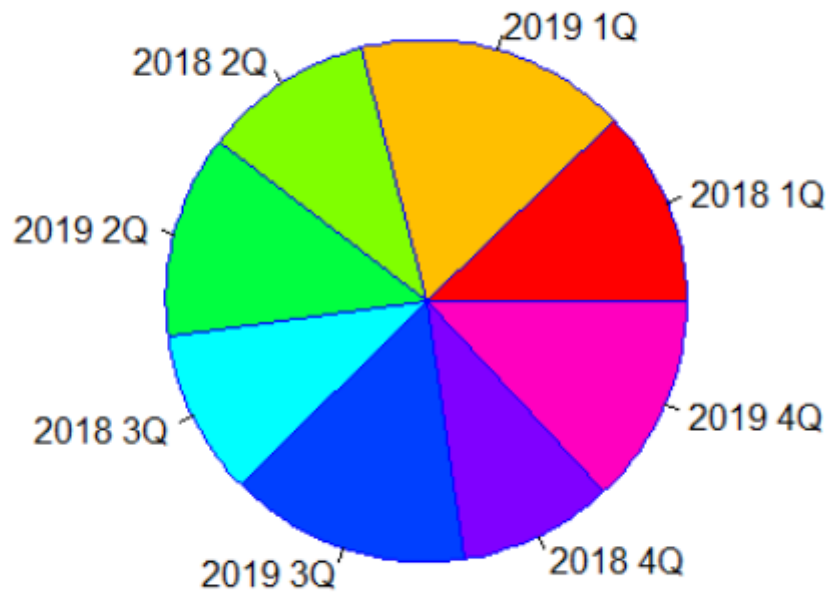


```
467 par(mfrow=c(2,2)) # 2행 2열에 그래프를 그린다
468 pie(char_data)
```



#col 옵션 - 색을 나타낸다.

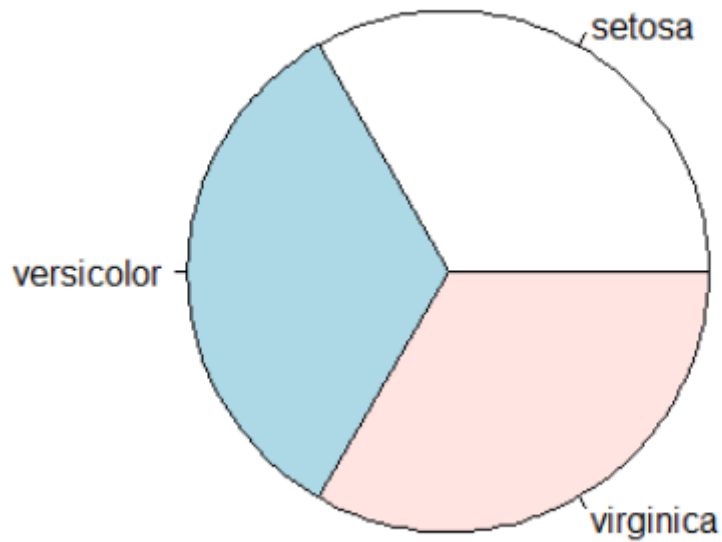
```
471 pie(char_data,
472     border = 'blue', #파이차트에 들어가는 선의 색깔
473     col     = rainbow(8))
```



#범주형 데이터를 파이 차트로

범주형 데이터를 파이차트로 나타낼수 있다. 만약 범주형 데이터가 char 일 경우 타입을 바꿔서 넣어준다.

```
480  
481 data(iris)  
482 class(table(iris$Species))  
483 pie(table(iris$Species)) #기존에는 범주형이라 들어가지 않는다. 그래서 타입을 바꿔준다  
484
```

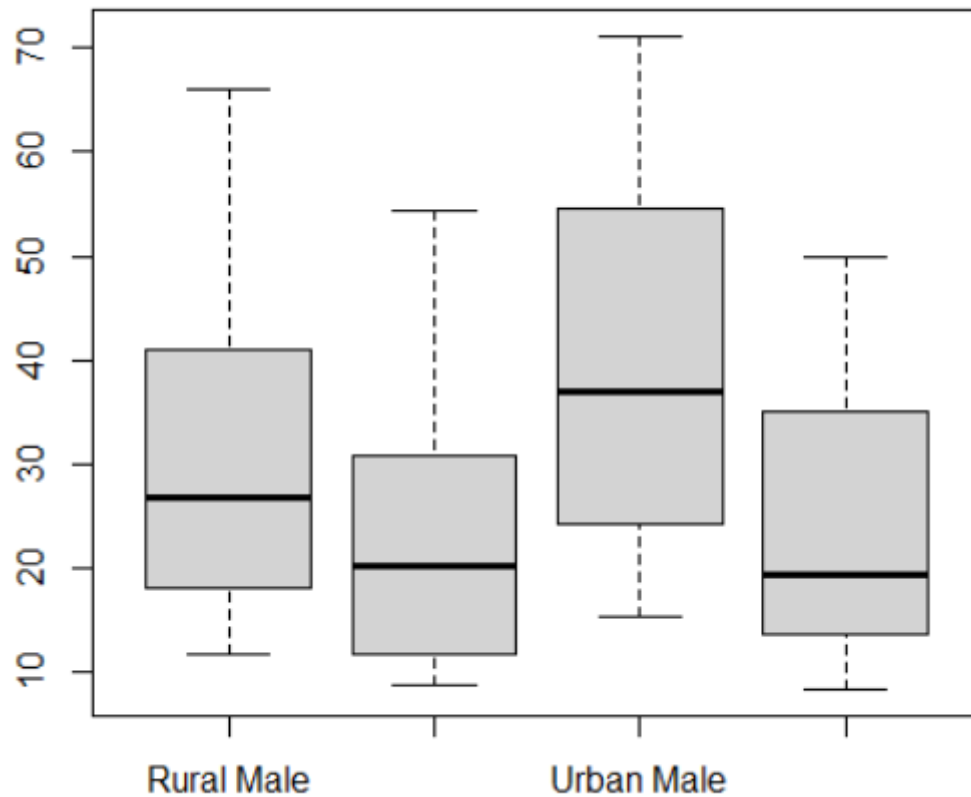


##기타 그래프들

#boxplot() - 상자 수염 그림

상자 그림은 데이터의 분포를 보여주는 그림으로 가운데 상자는 제1사분위수, 중앙값, 제3사분위수⁹를 보여준다. 그 위아래로 있는 값은 제일 작은 값과 큰값으로 이상치를 나타낸다.

```
500
501 # 박스 플롯'은 '상자 수염 그림'(Box-and-Whisker Plot) '상자 그림'으로 불리워진다.
502 # 이 박스플롯을 가로로 돌린다음에 정규분포도랑 비교한다.
503 # 보통 이상치를 제거 하기 위해 시각화해서 보는 방법이 boxplot이다. 표준화와 정규화를 위해 사용
504 boxplot(VADeaths)
505
```

#hist() - 히스토그램

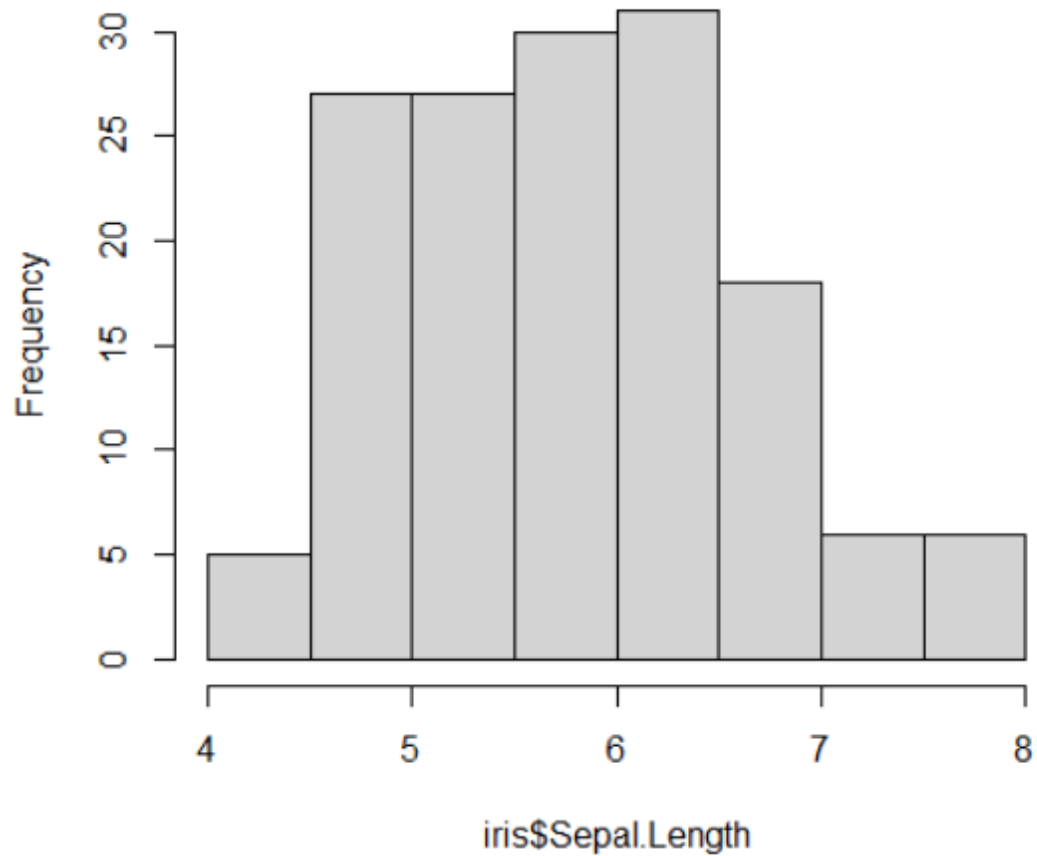
히스토그램은 값의 범위마다 빈도를 표시한 그래프다.

```

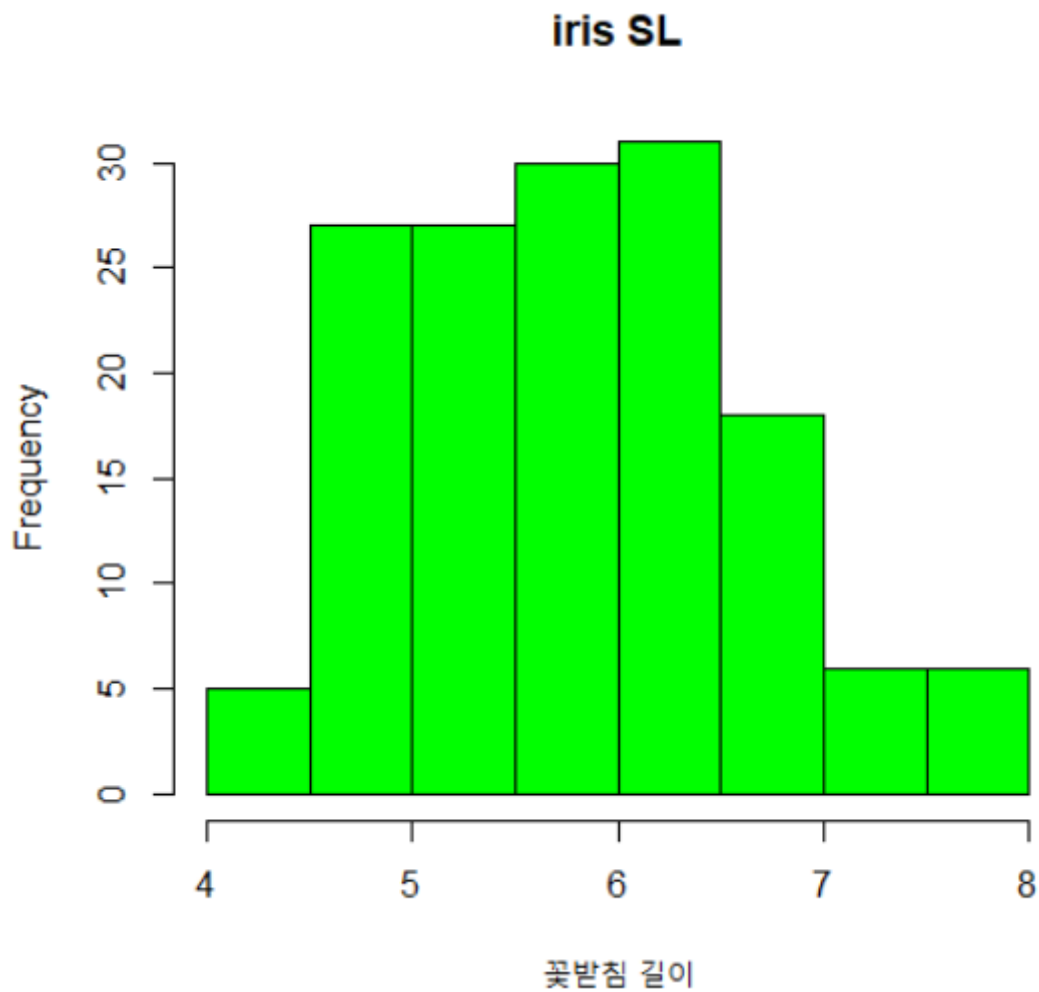
507 #히스토그램
508 data(iris)
509 iris
510
511 # attach() - 이 명령어를 사용하면 붙여있을 동안 데이터명$ 을 안쓰고 그냥 변수명을 써도 된다. ,
512 # detach() - 이 명령어로 attach를 없앤다.
513 attach(iris)
514 mean(Sepal.Length)
515
516 summary(iris)
517 hist(iris$Sepal.Length)

```

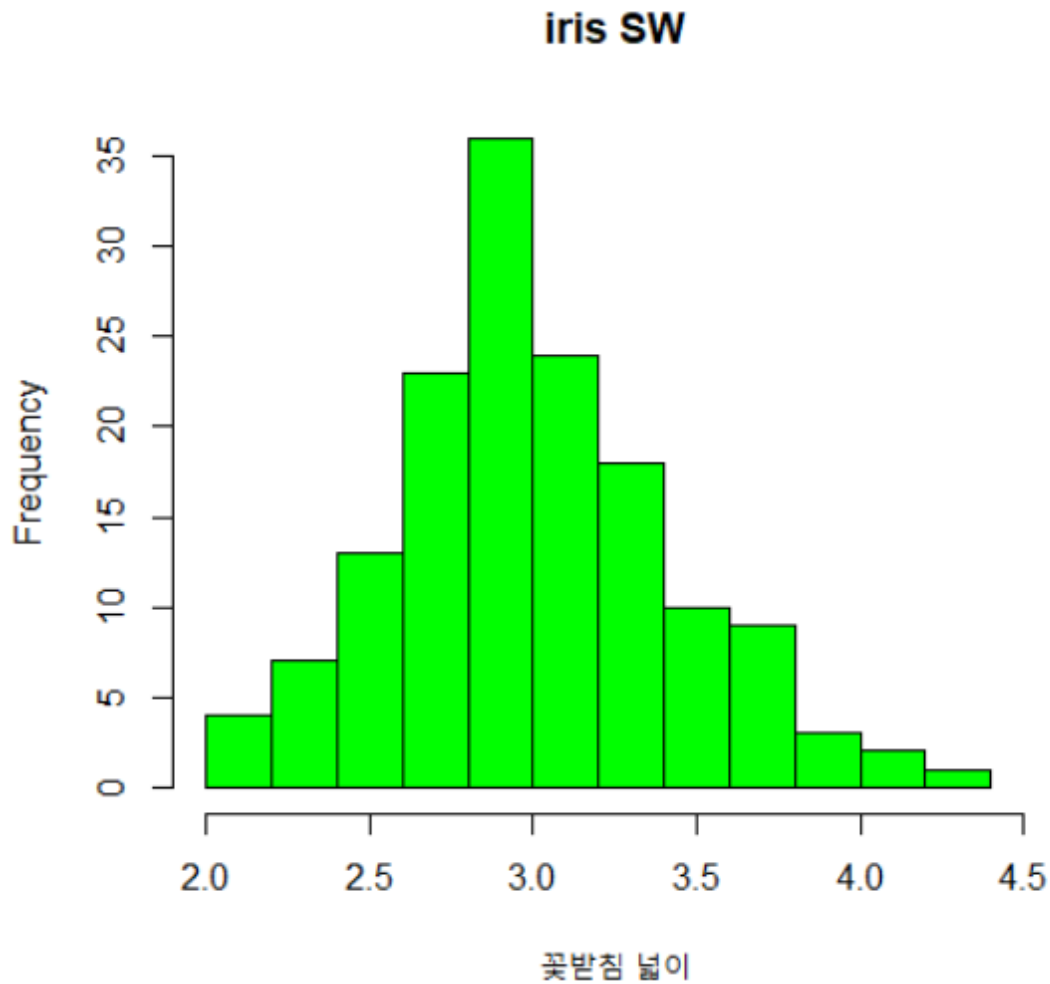
Histogram of iris\$Sepal.Length



```
519 hist(iris$Sepal.Length,  
520       xlab = "꽃받침 길이",  
521       col  = "green",  
522       main = "iris SL")  
523
```



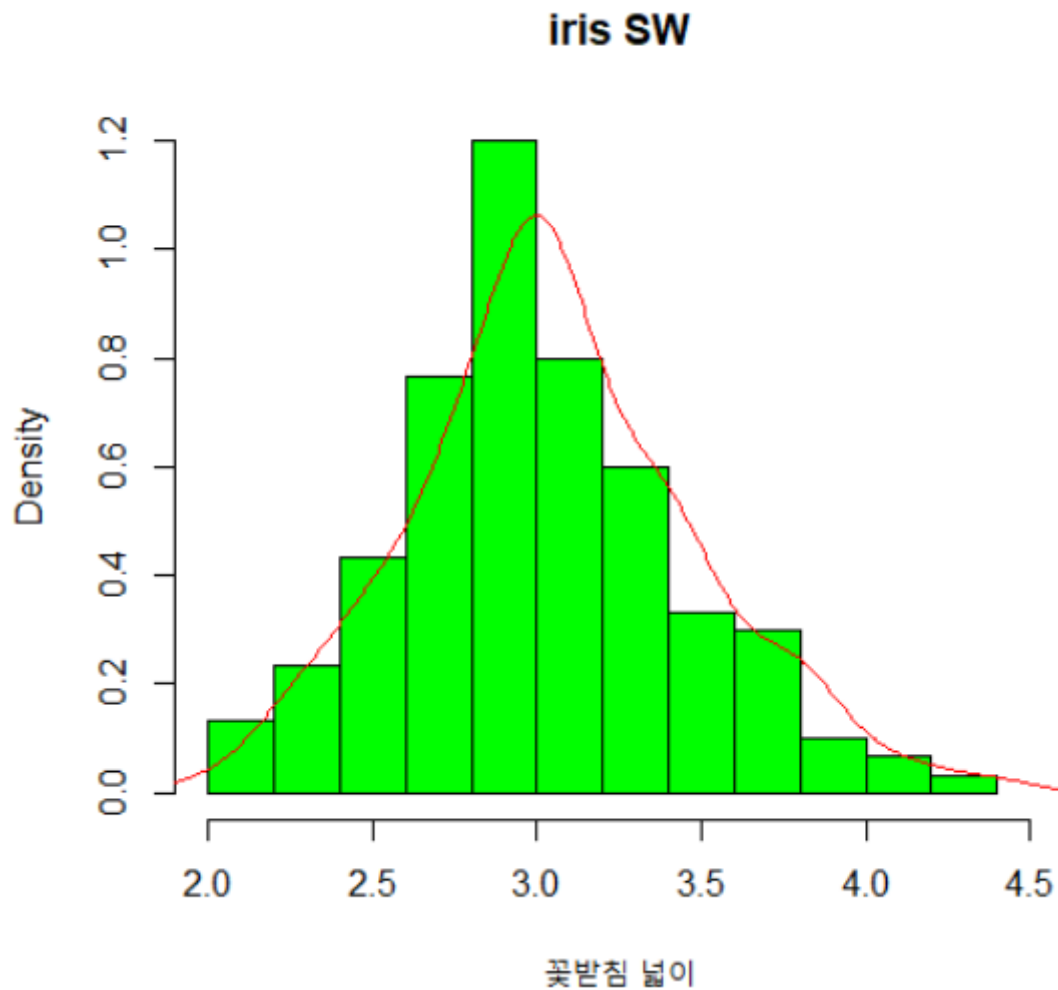
```
524  
525 hist(iris$Sepal.Width,  
526       xlab = "꽃받침 넓이",  
527       col = "green",  
528       main = "iris SW",  
529       xlim = c(2.0,4.5)) # xlimit
```



#density() - 밀도 그림

`density()`로 그리는 밀도 그림은 막대의 너비를 가정하지 않고 모든 점에서 데이터의 밀도를 추정하는 커널 밀도 추정(kernel density estimation) 방식을 사용

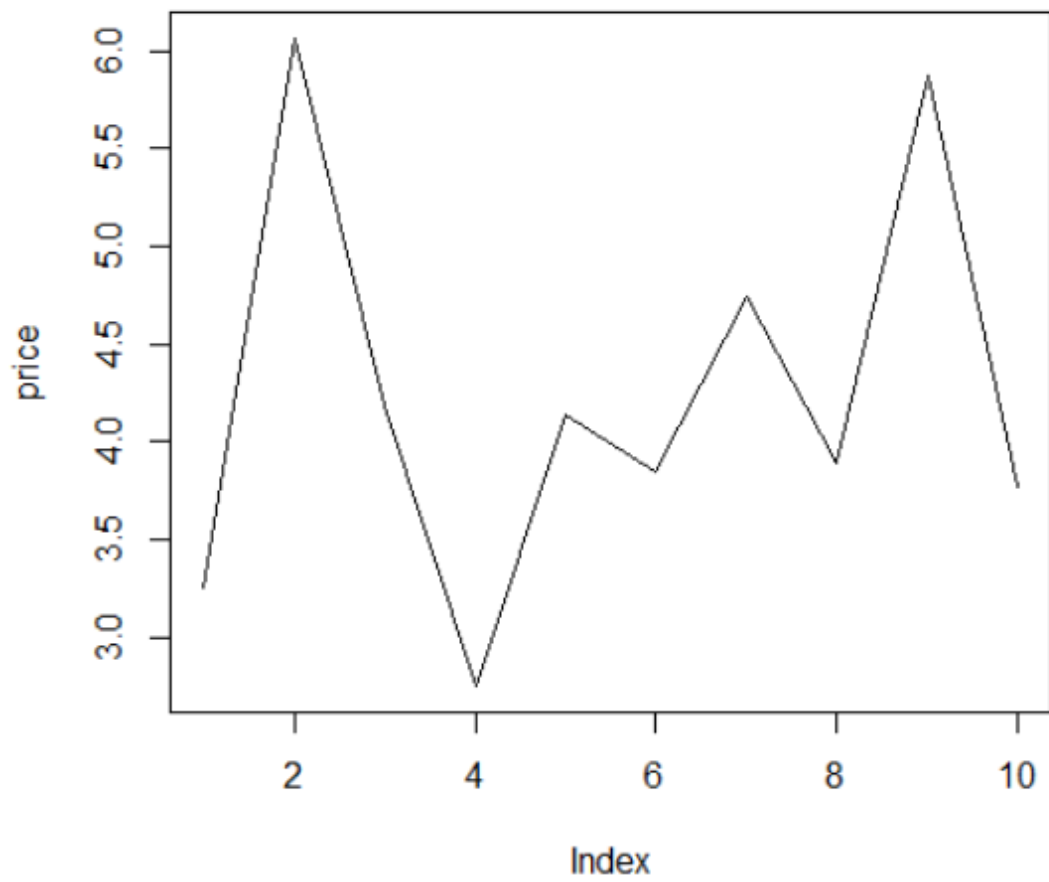
```
lines(density(iris$Sepal.Width),col="red")
```



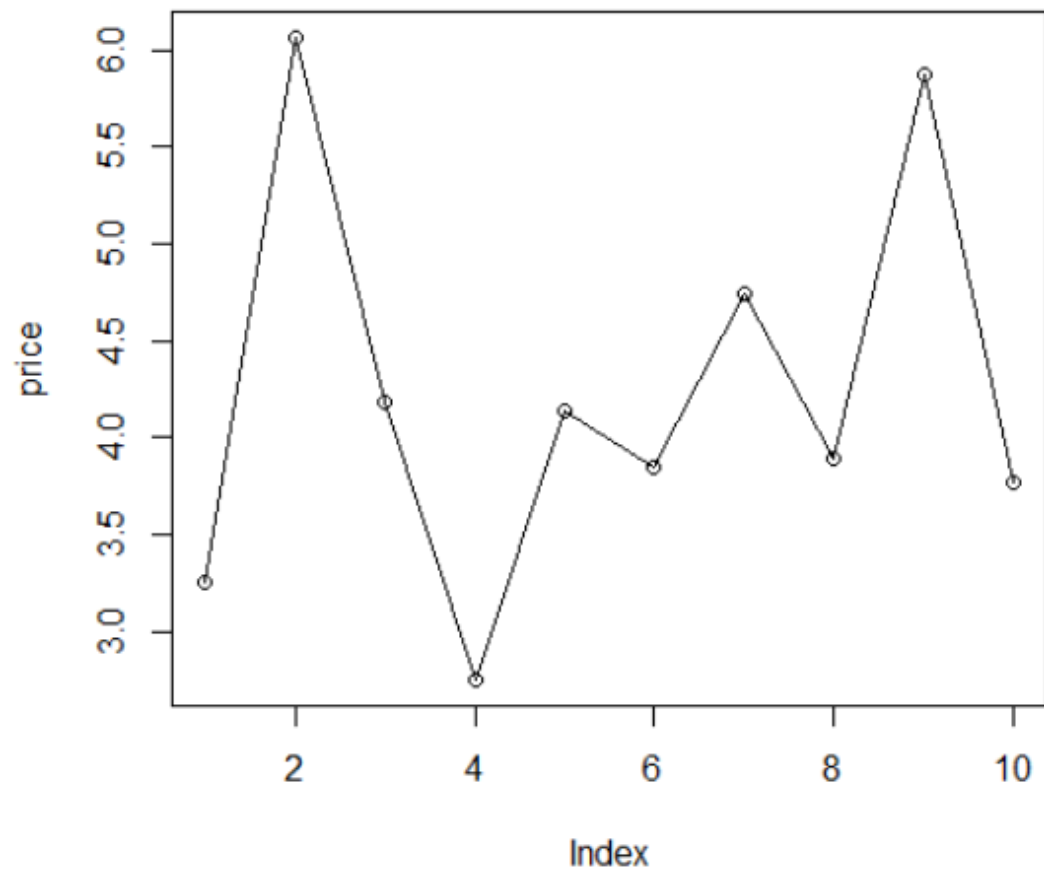
#plot () -산점도

- 주어진 데이터를 점으로 표시해 흩뿌리듯이 시각화한 그림

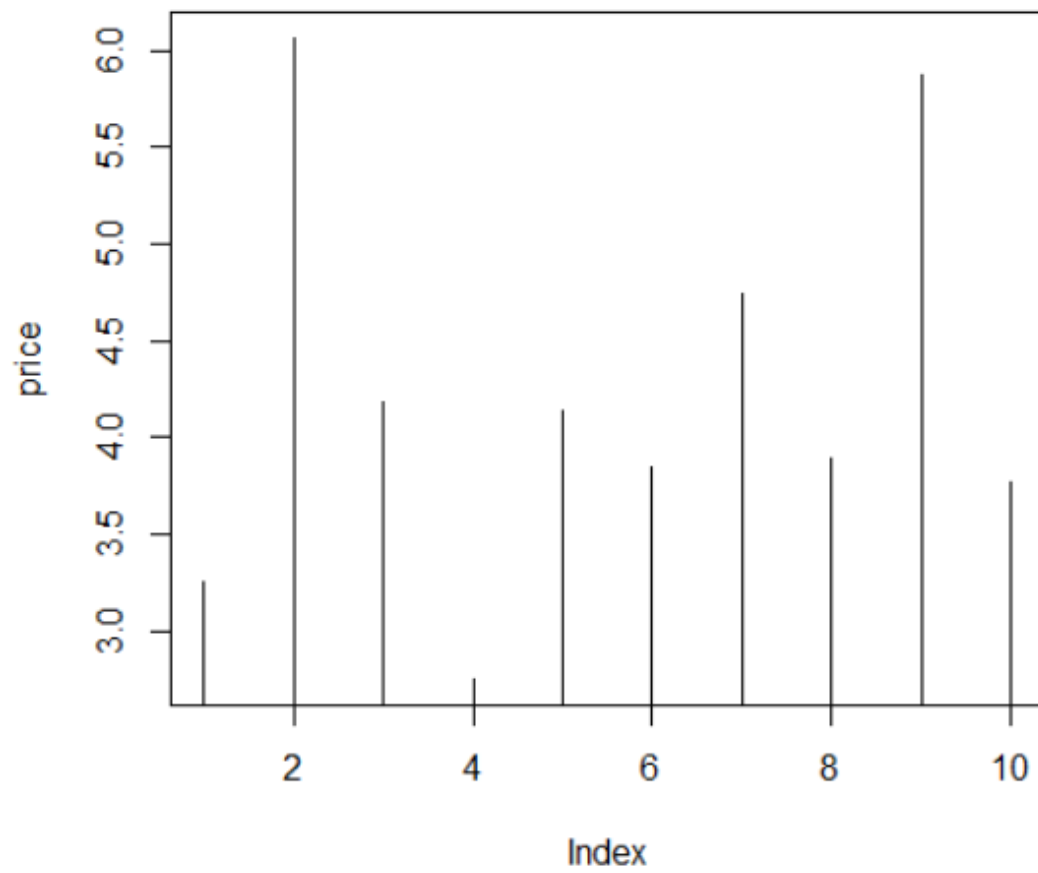
```
551  
552 price <- runif(10, min =2 , max =8)  
553 plot(price, type ="l") # 일반 직선 산점도  
554
```



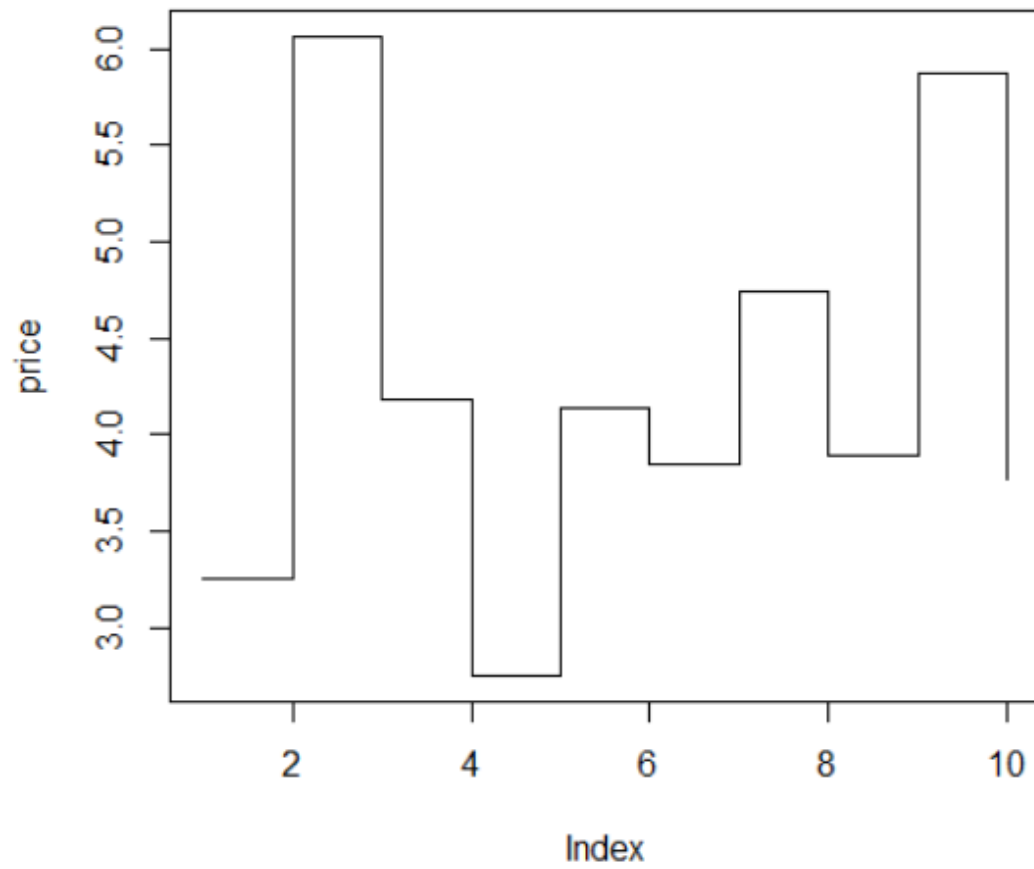
```
555 plot(price, type = "o") # 데이터 부분에 o 표시를 하는 타입
```



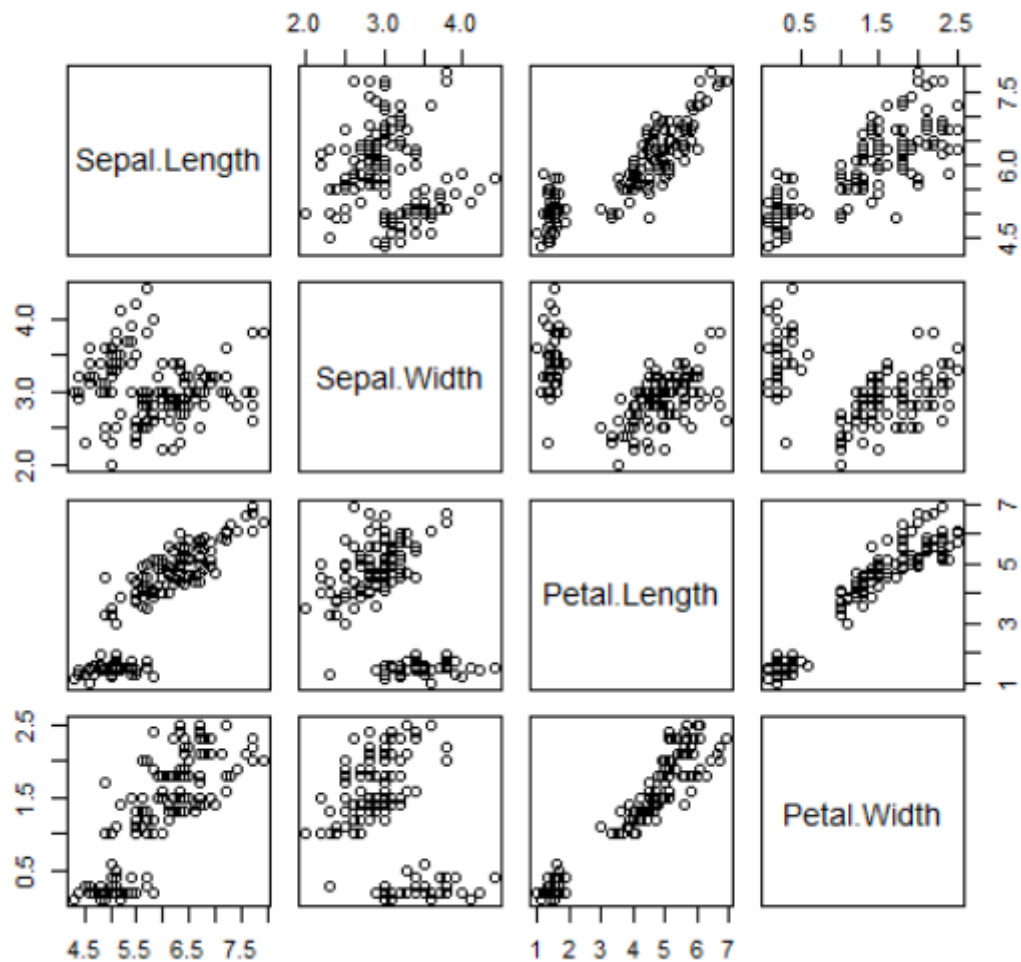
```
556  
557 plot(price, type = "h") # 직선으로 나타내는 타입  
558
```



```
558  
559 plot(price, type = "s") # 선을 이어 직각으로 나타내는 옵션  
560
```

```
565  
566 # iris - scatter matrix(산점도 매트릭스)  
567 # paris()  
568 |  
569 iris  
570 pairs(iris[1:4])  
571
```



scatterplot3d - 3차원 산점도

산점도를 3차원으로 나타낸 그래프이다. 이 같은 경우에는 종으로 데이터가 분류되어있어야 한다.

```

# 3차원 산점도
install.packages("scatterplot3d")
library(scatterplot3d)
# 3D로 그릴려면 종이 분류되어 있어야된다. filter , subset

# 3가지 방법
# iris_setosa <- filter(iris, Species == "setosa")
# iris__setosa <- iris[iris$Species == "setosa"]
# iris_setosa<-subset(iris, iris$Species == "setosa")

iris_setosa<-subset(iris, iris$Species == "setosa")
iris_versicolor<-subset(iris, iris$Species == "versicolor")
iris_virginica<-subset(iris, iris$Species == "virginica")

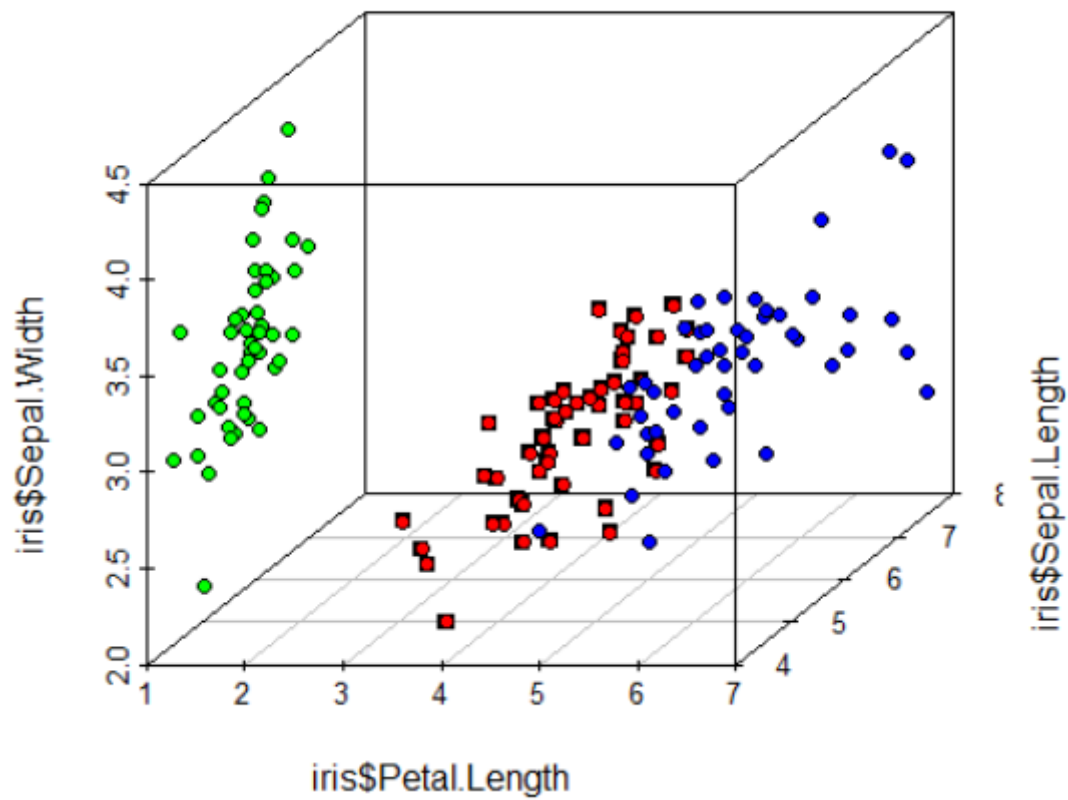
iris3D <- scatterplot3d(iris$Petal.Length,
                        iris$Sepal.Length,
                        iris$Sepal.Width, type = 'n')

iris3D$points3d(iris_setosa$Petal.Length,
                iris_setosa$Sepal.Length,
                iris_setosa$Sepal.Width,
                bg = "green",
                pch = 21)

iris3D$points3d(iris_versicolor$Petal.Length,
                iris_versicolor$Sepal.Length,
                iris_versicolor$Sepal.Width,
                bg = "red",
                pch = 21)

iris3D$points3d(iris_virginica$Petal.Length,
                iris_virginica$Sepal.Length,
                iris_virginica$Sepal.Width,
                bg = "blue",
                pch = 21)

```



'R' 카테고리의 다른 글

[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

[R] R로 만드는 제어문 (if, else if, for)과 예제

R boxplot

R boxplot 함수

R dotchart

R dotchart 함수

R hist

R hist 함수

R PIE

R PIE 함수

R plot

R plot 함수



꾸까꾸

혼자 끄적끄적하는 블로그 입니다.