시각화

박찬영

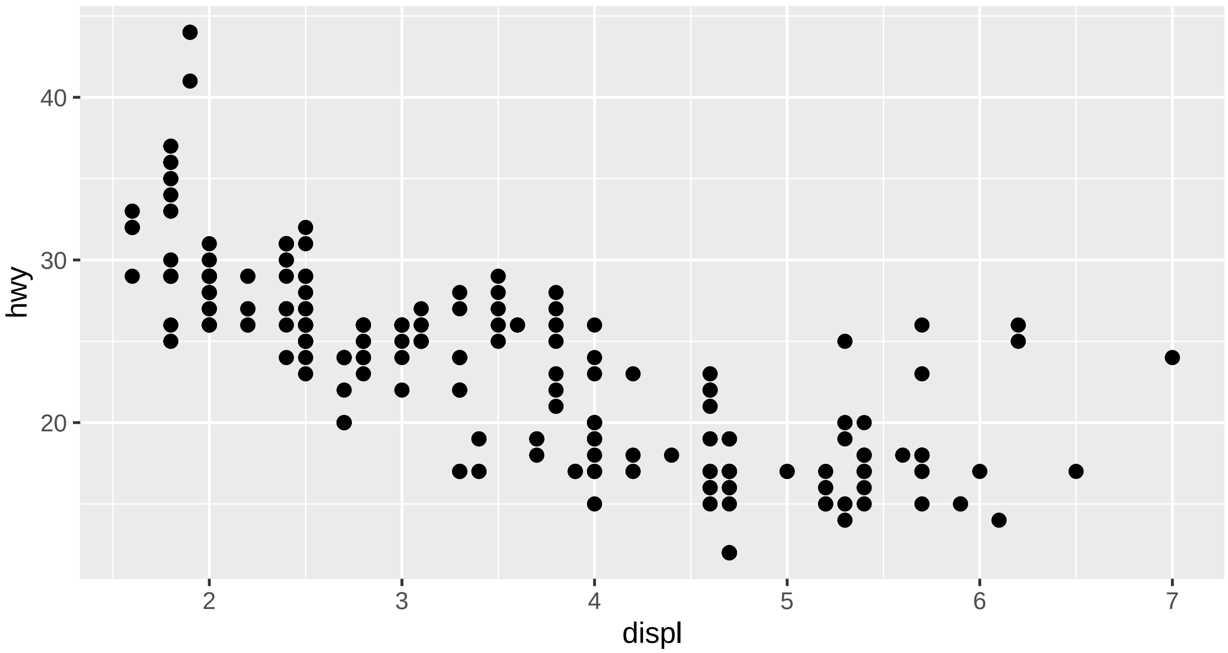
2024-08-19

tidyverse library를 사용합니다.

ggplot을 써봅시다.

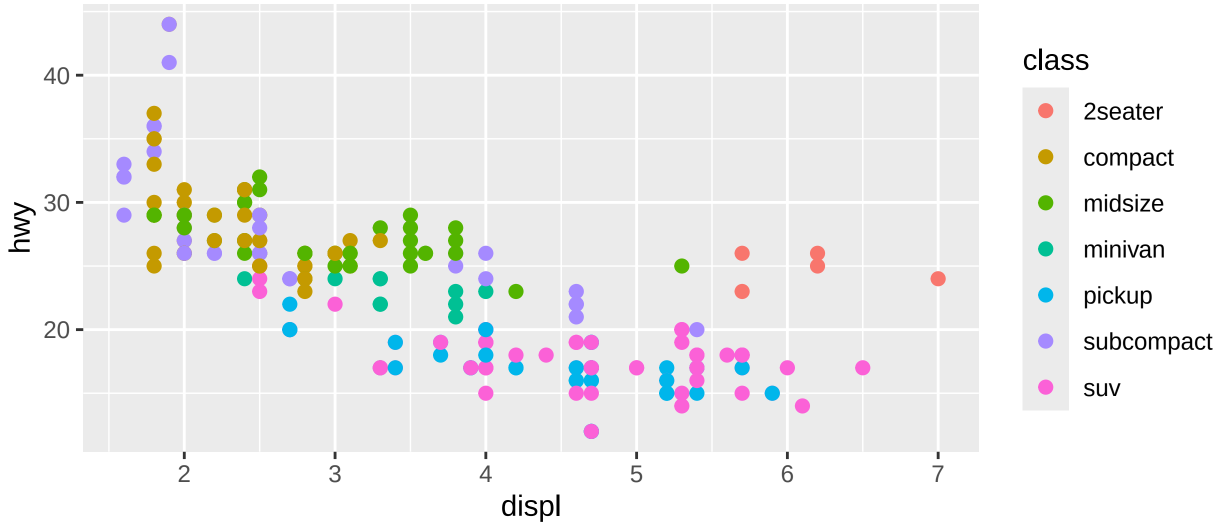
기본적인 그래프를 그리는 방법

```
ggplot(data = mpg) + #플롯창을 띄운다는 느낌
    geom_point(mapping = aes(x=displ, y=hwy), size=2) +
    #aes 함수는 데이터, 축에 대한 설정, 점크기는 size
    theme(aspect.ratio = 1/2) #플롯 비율
```



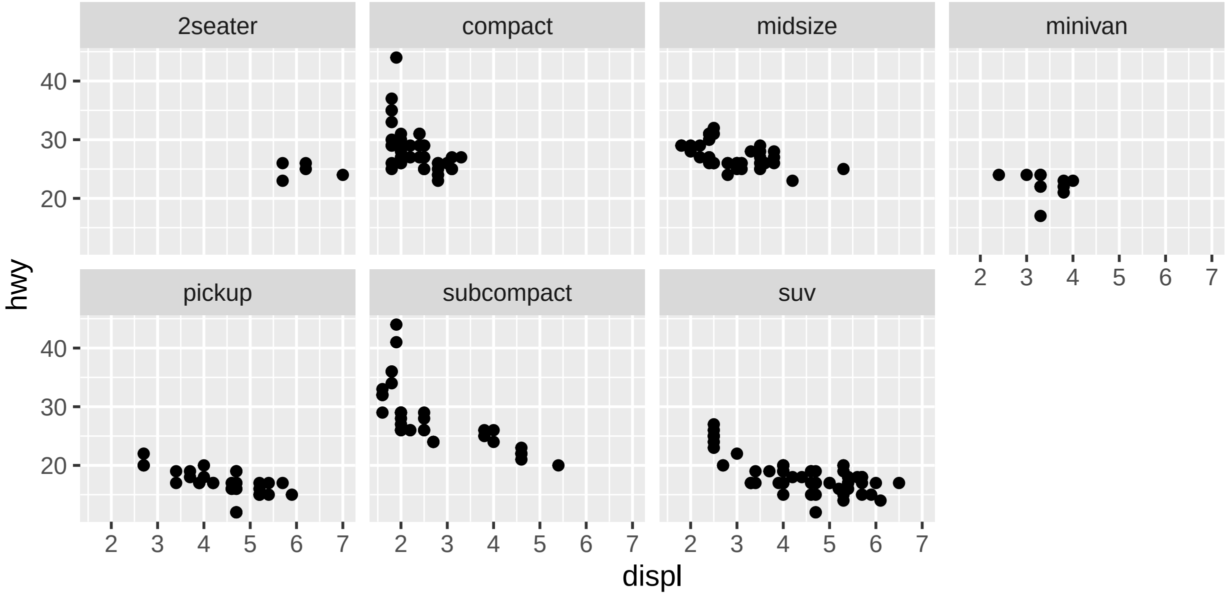
범주 데이터를 활용

```
ggplot(data = mpg) +
    geom_point(mapping = aes(x=displ, y=hwy, color=class), size=2) +
    #color, size, alpha, shape는 범주형 데이터를 반영해줌
    theme(aspect.ratio = 1/2)
```

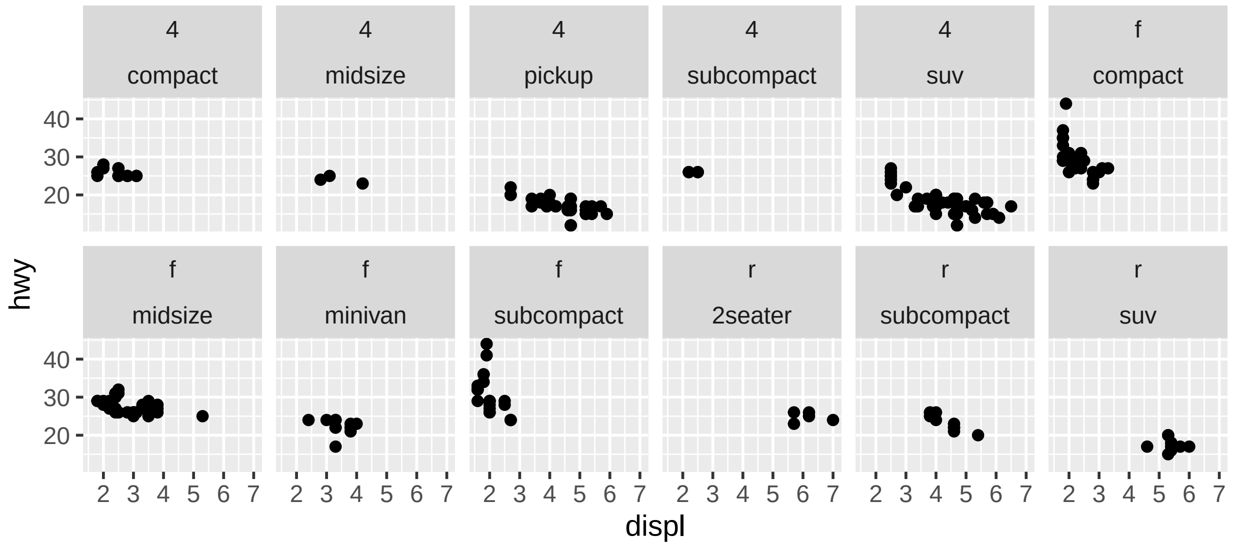


서브플롯을 그려보아요

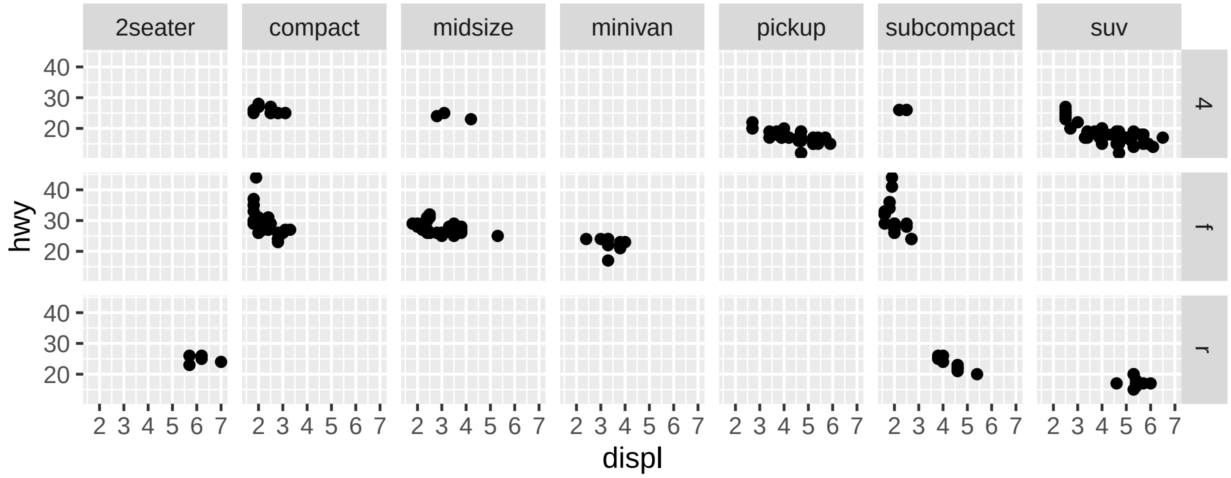
```
ggplot(data = mpg) +
    geom_point(mapping = aes(x=displ, y=hwy)) +
    facet_wrap(~ class, nrow=2) +
    #facet은 범주데이터에 따라 서브플롯을 그려줌
    theme(aspect.ratio = 3/4)
```



```
ggplot(data = mpg) +
    geom_point(mapping = aes(x=displ, y=hwy)) +
    facet_wrap(drv ~ class, nrow=2) +
    #drv ~ class 하면 각 drv에 따른 class로 보여줌
    theme(aspect.ratio = 3/4)
```



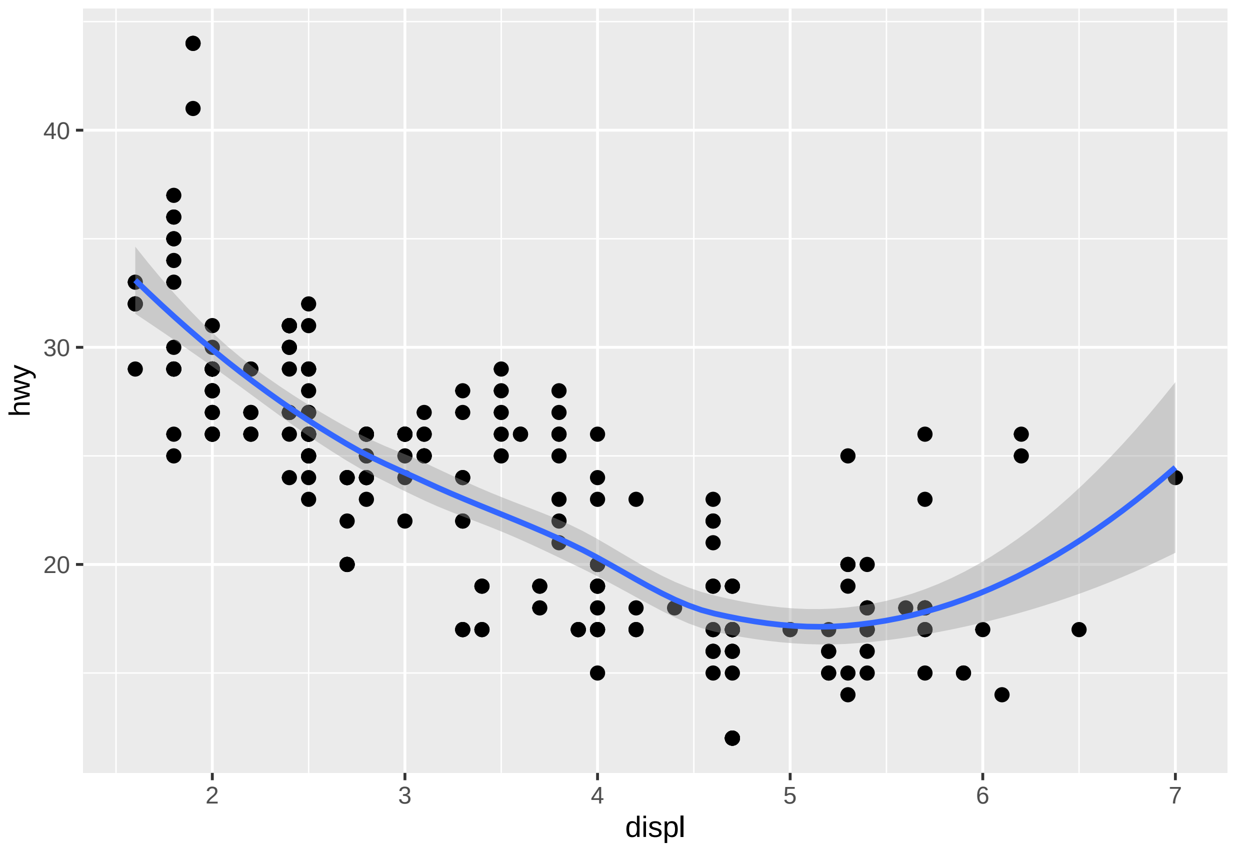
```
ggplot(data = mpg) +
    geom_point(mapping = aes(x=displ, y=hwy)) +
    facet_grid(drv ~ class) +
    #grid 하면 이렇게 뜸
    theme(aspect.ratio = 3/4)
```



스무스한 선을 그리기

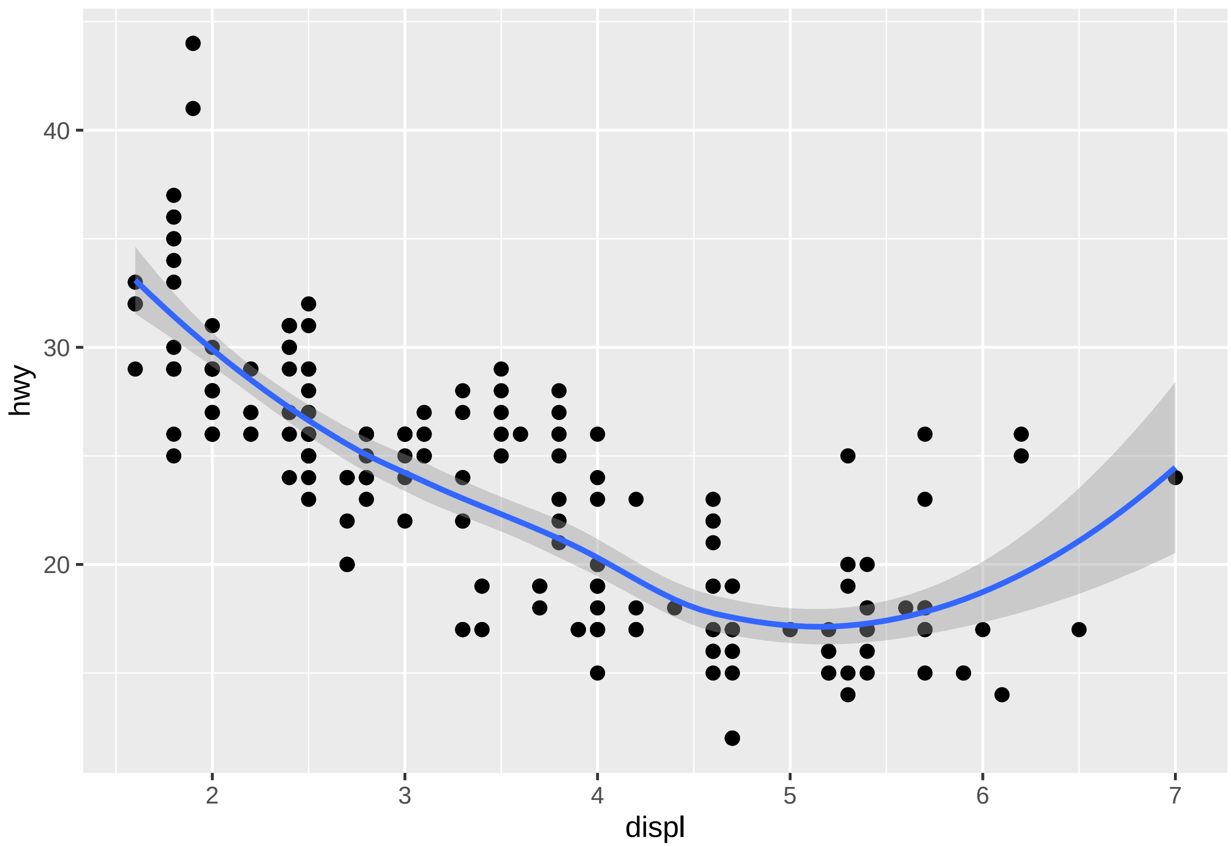
```
ggplot(data = mpg) +
    geom_point(mapping = aes(x=displ, y=hwy), size=2) +
    geom_smooth(aes(displ, hwy)) #스무스한 선을 만듬
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



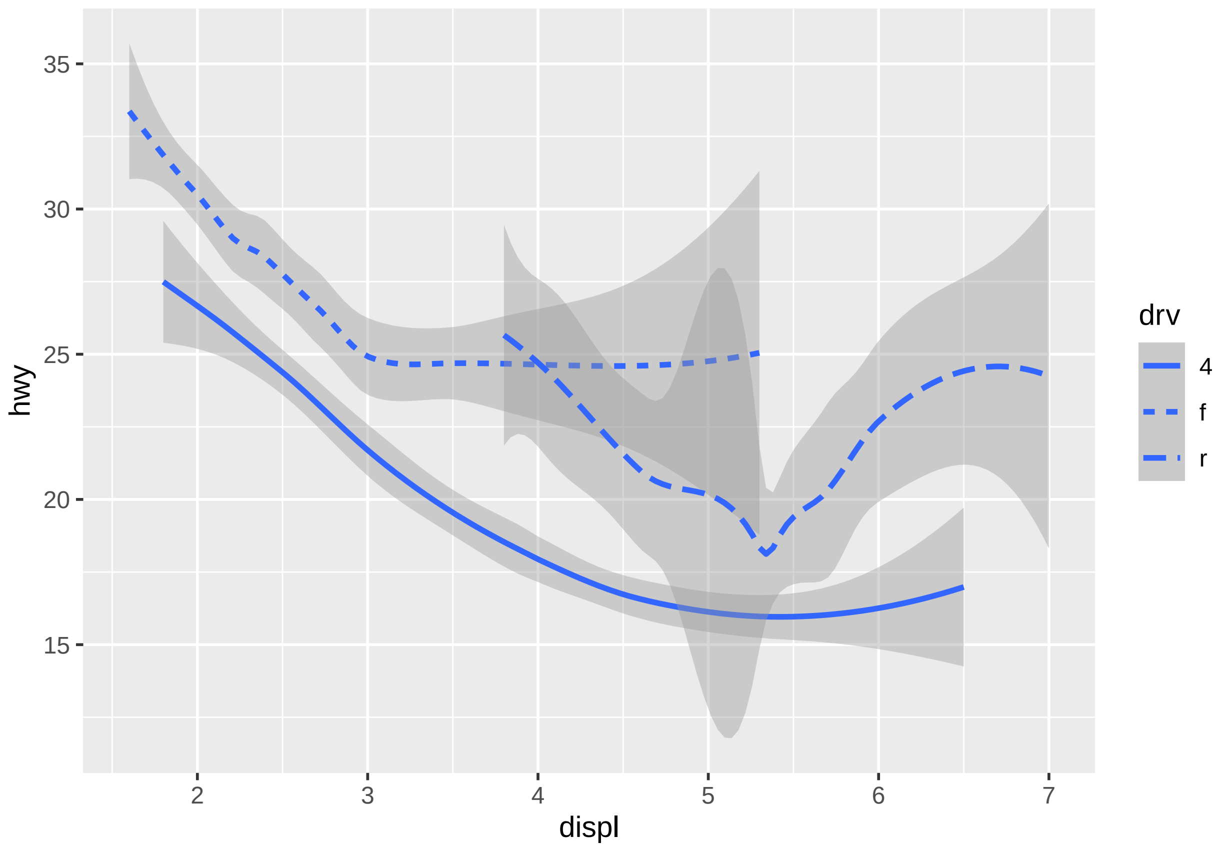
```
ggplot(data = mpg, aes(displ, hwy)) +
    geom_point(size=2) + geom_smooth() #간결한 코드
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



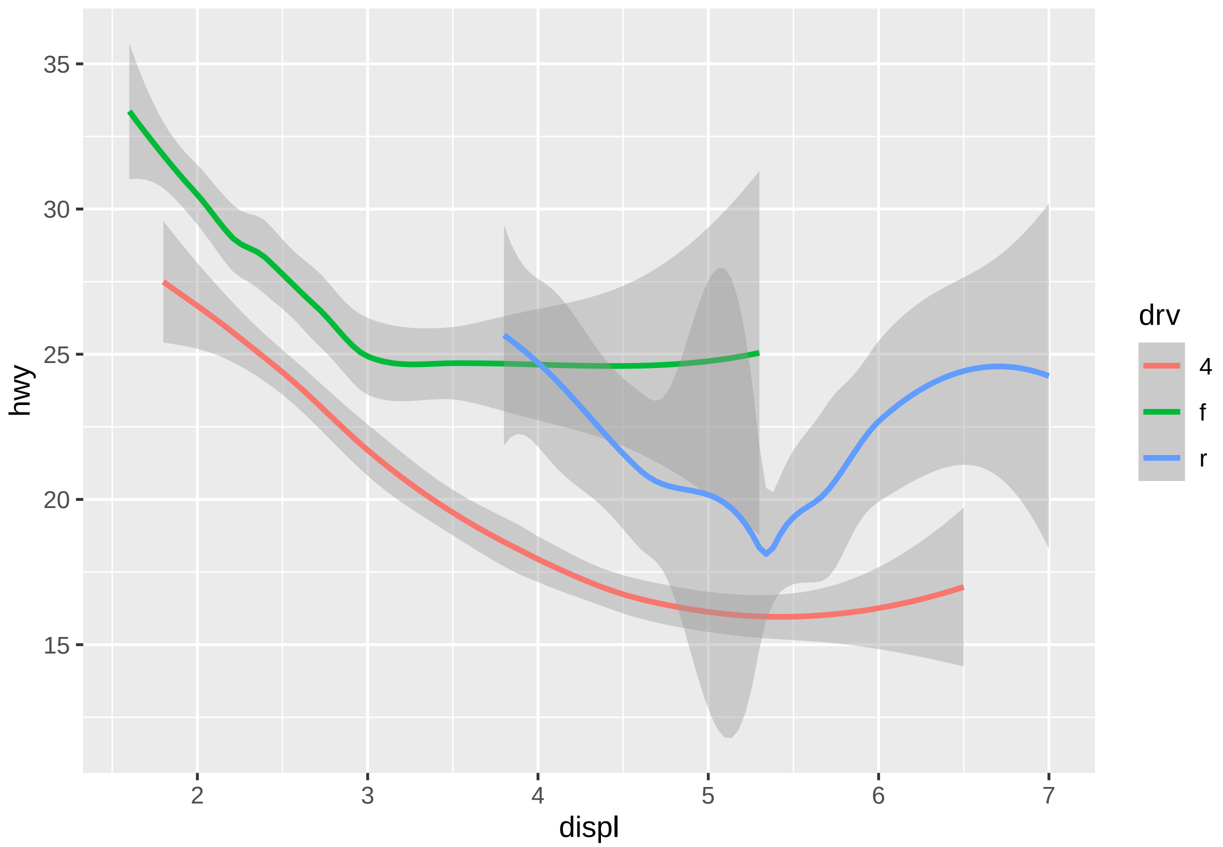
```
ggplot(data = mpg) +
    geom_smooth(aes(displ, hwy, linetype=drv)) #drv에 따라서..
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



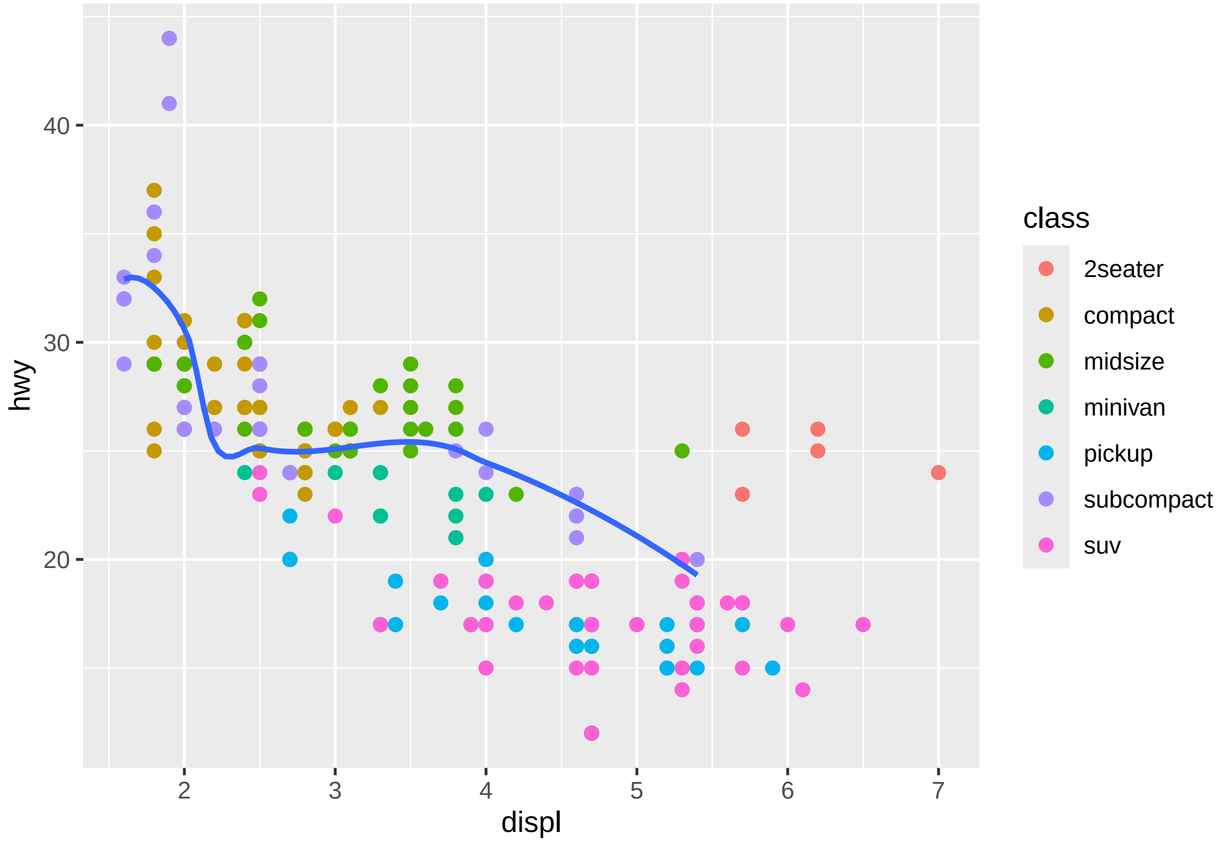
```
ggplot(data = mpg) +
    geom_smooth(aes(displ, hwy, color=drv)) #당연히 얘들도 됨
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
ggplot(data = mpg, aes(displ, hwy)) +
    geom_point(aes(color=class),size=2) +
    geom_smooth(data=filter(mpg, class=="subcompact"), se=FALSE)
```

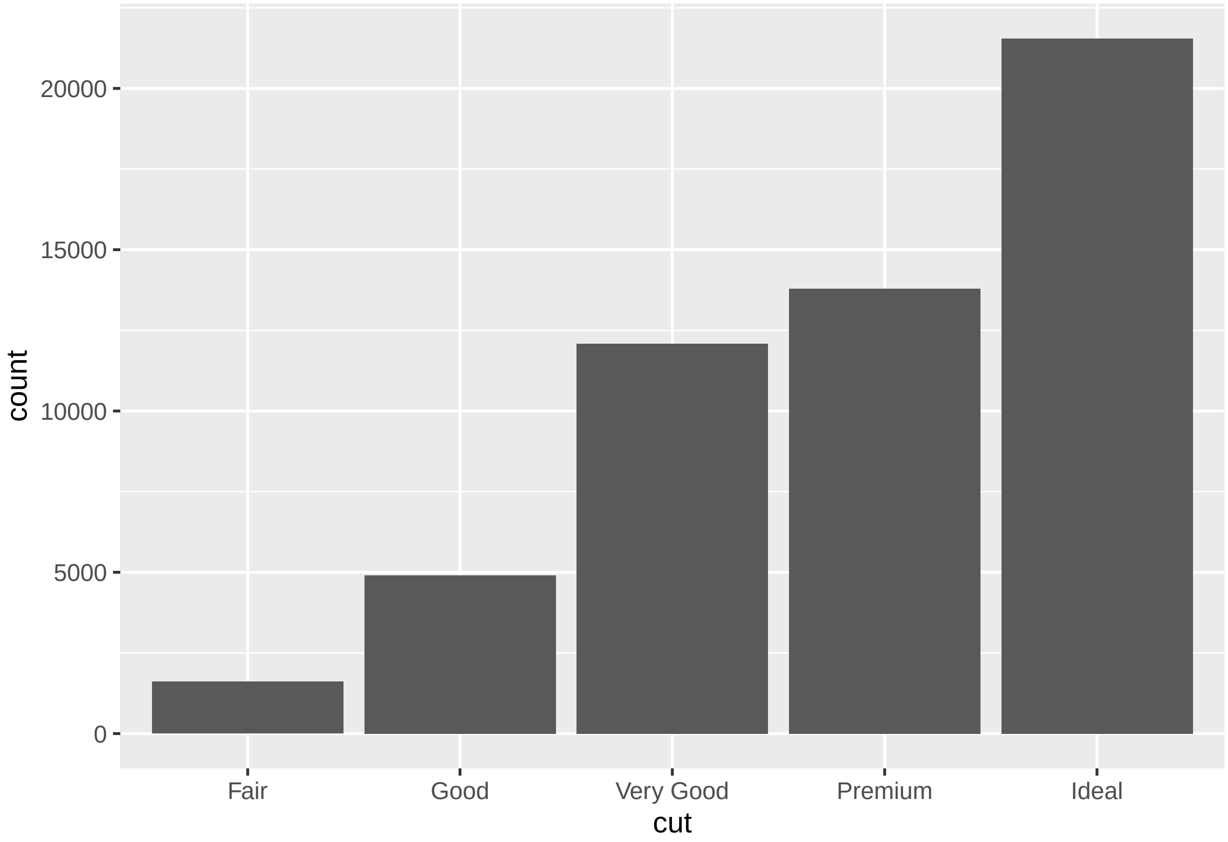
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



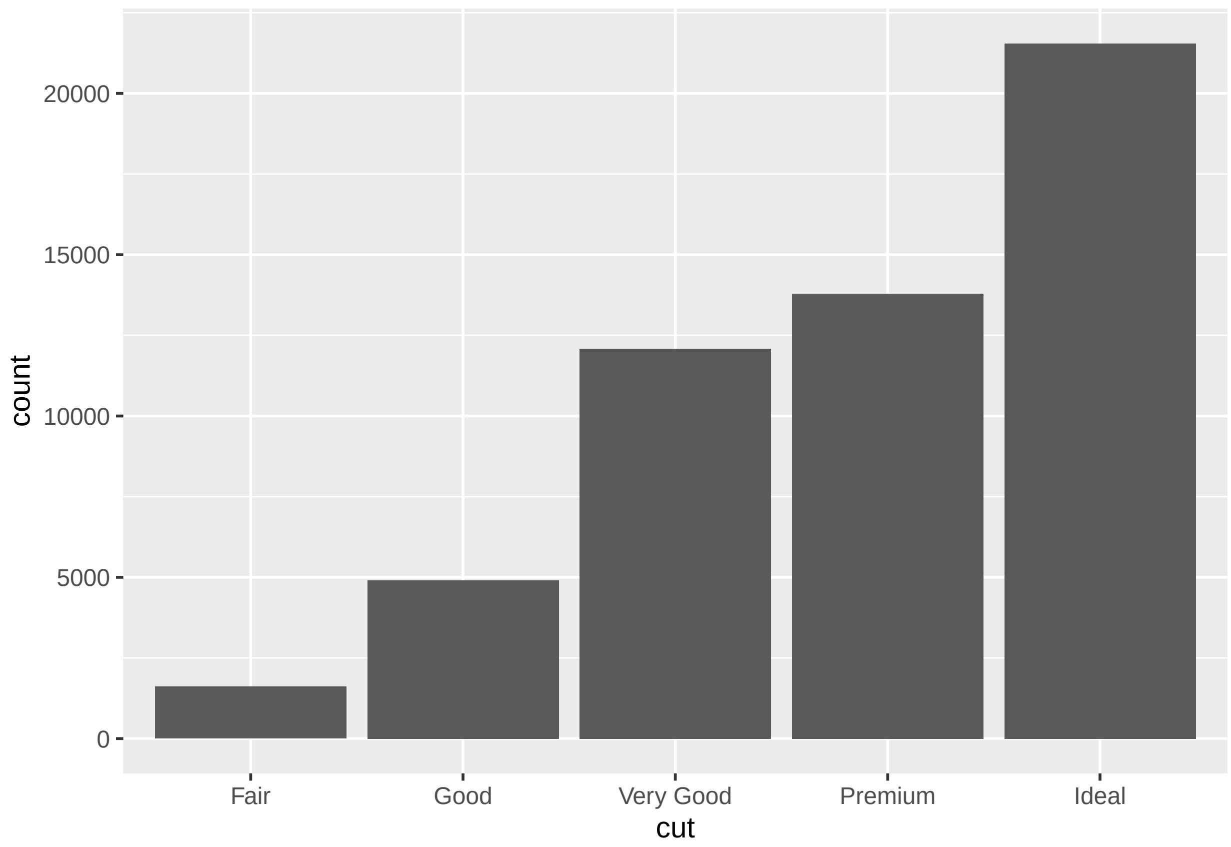
```
    #filter를 이용해 class가 subcompact인 놈들만 스무스를 그림
    #se (스무스 배경)은 FALSE
```

## 히스토그램을 그려보아요

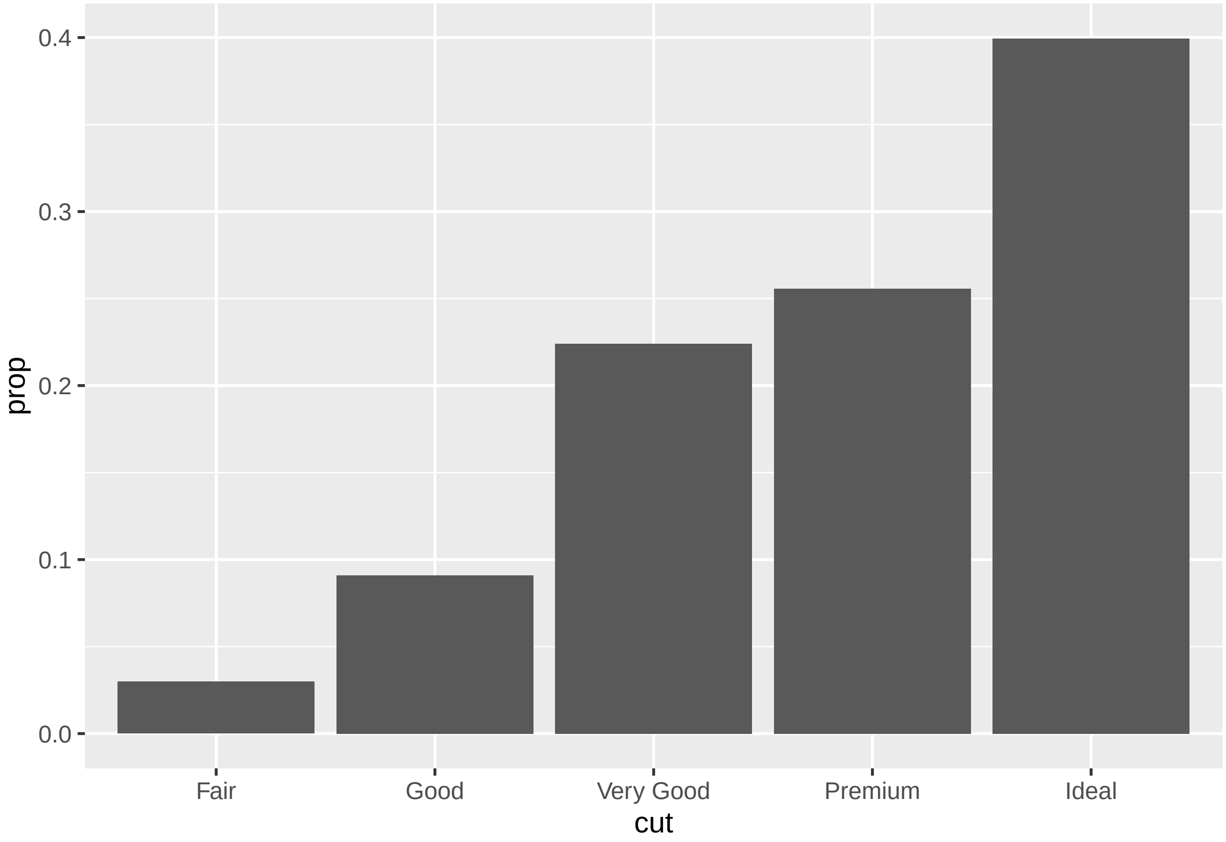
```
ggplot(data= diamonds) +
    geom_bar(aes(x=cut))
```



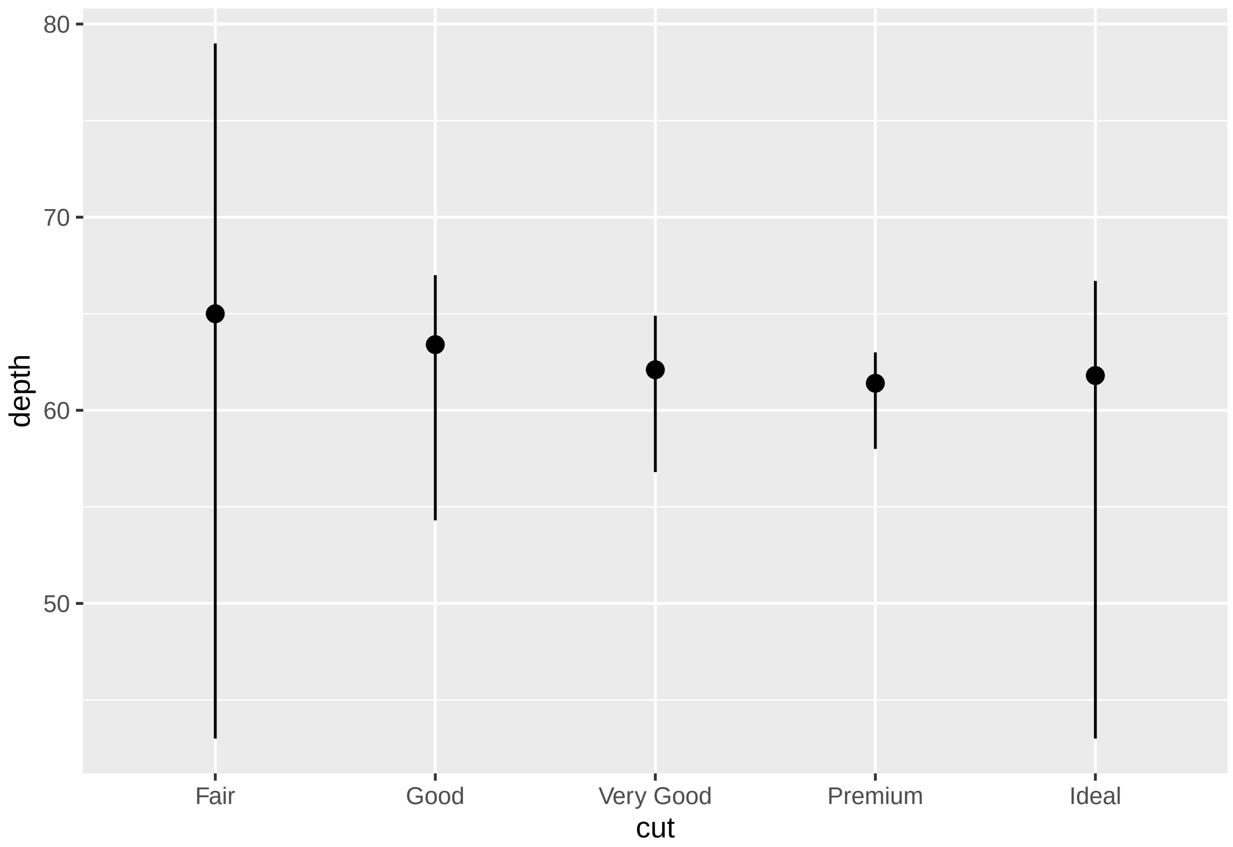
```
#geom_bar는 stat_count를 사용

ggplot(data=diamonds) +
    stat_count(aes(x=cut))
```



```r
ggplot(data=diamonds) +
    geom_bar(aes(x=cut, y=after_stat(prop), group=1))
```



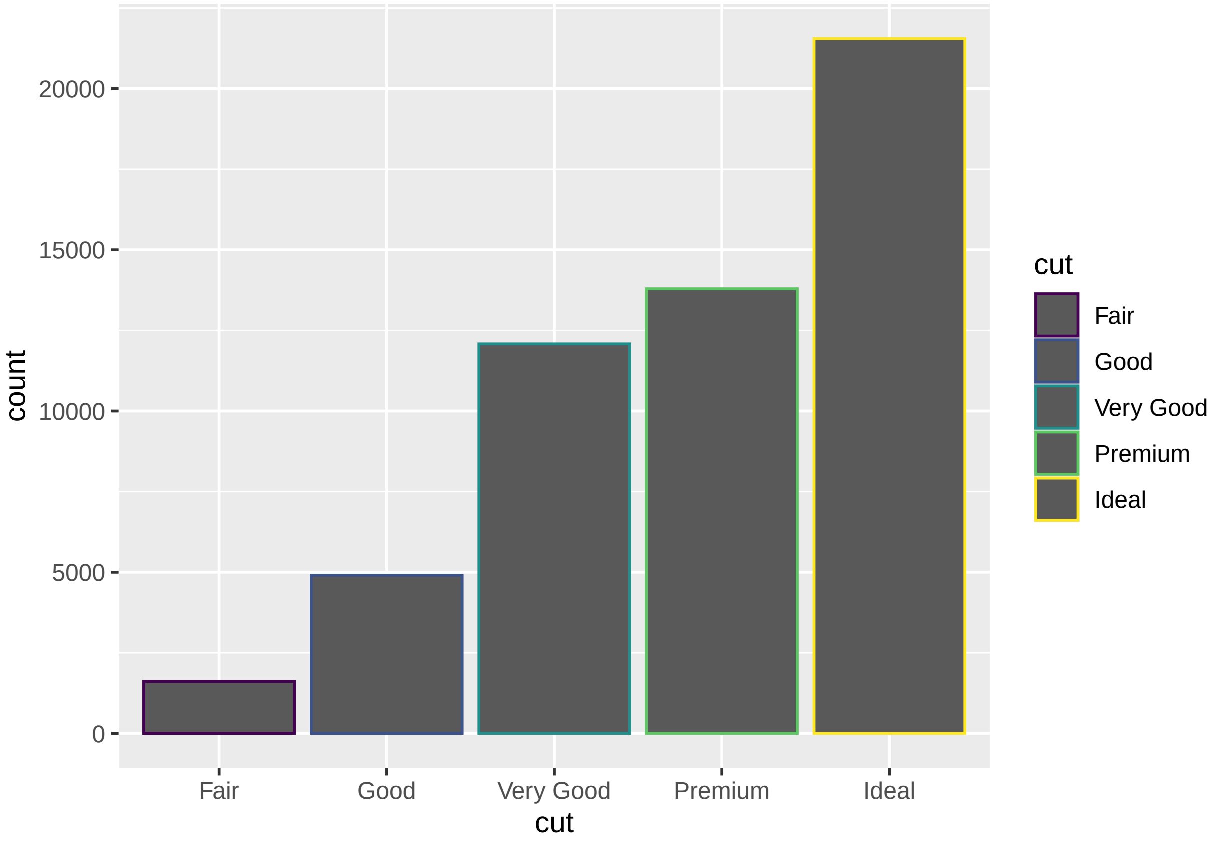
```
#상대도수로 만들기

ggplot(data=diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
    fun.min = min,
    fun.max = max,
    fun = median
  )
```



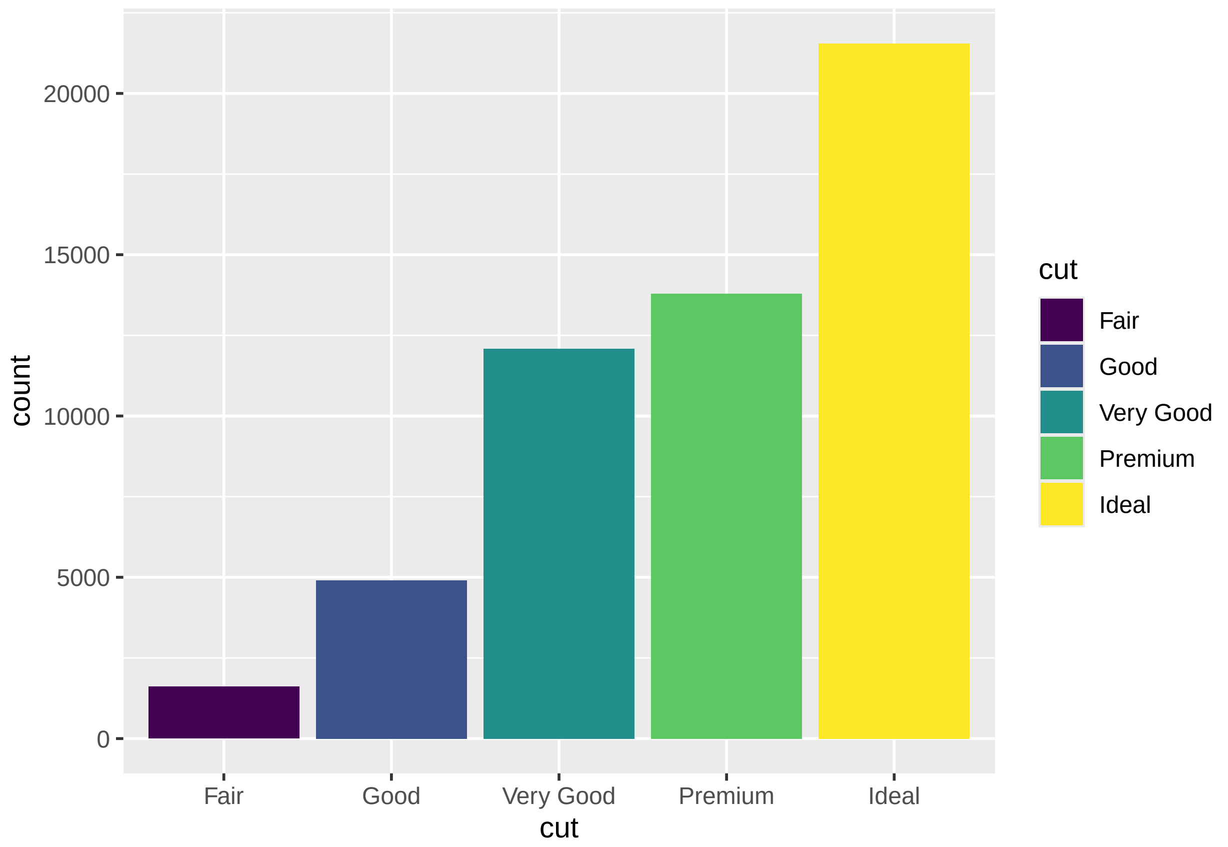
```
#사용자 지정 통계라네요, cut에따른 depth의 박스플롯같은걸 그려줍니다.
```

히스토그램 꾸미기

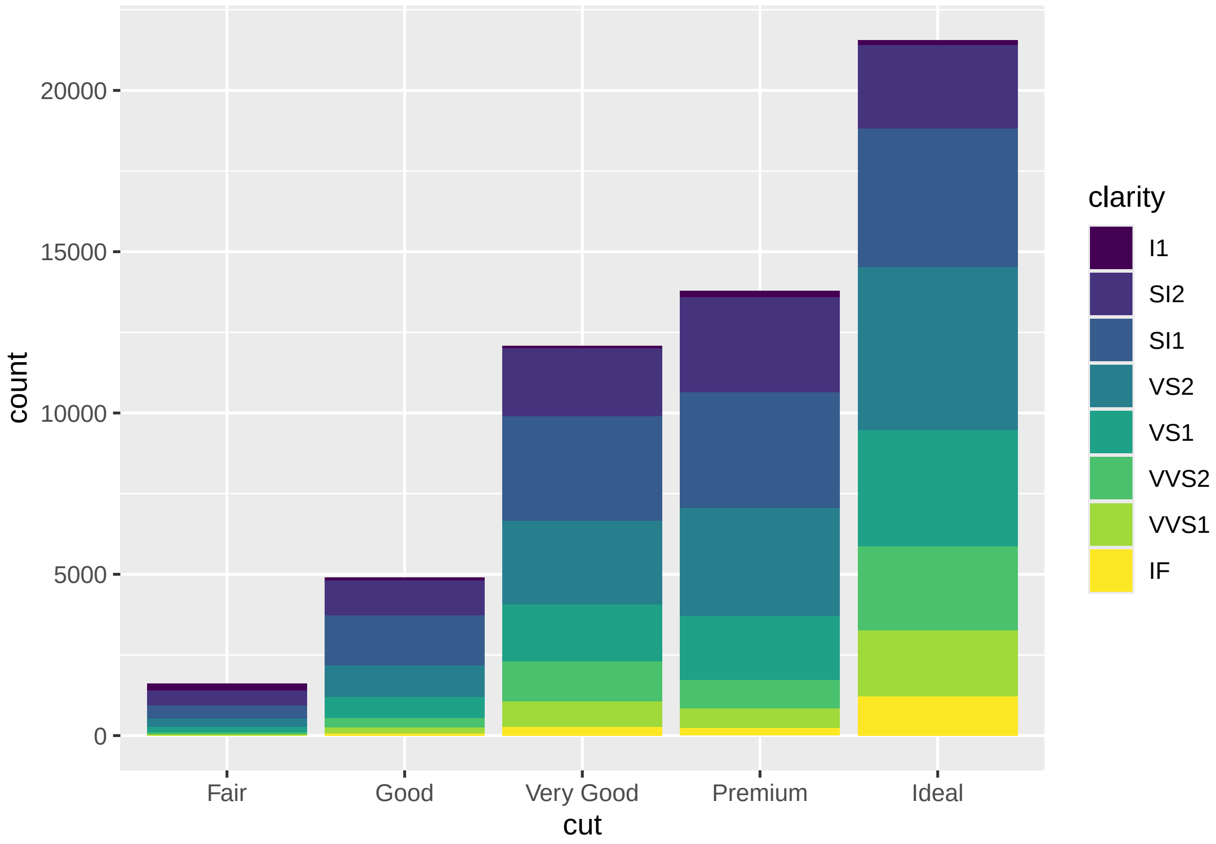
```
ggplot(data=diamonds) +
    geom_bar(aes(cut, colour=cut))
```



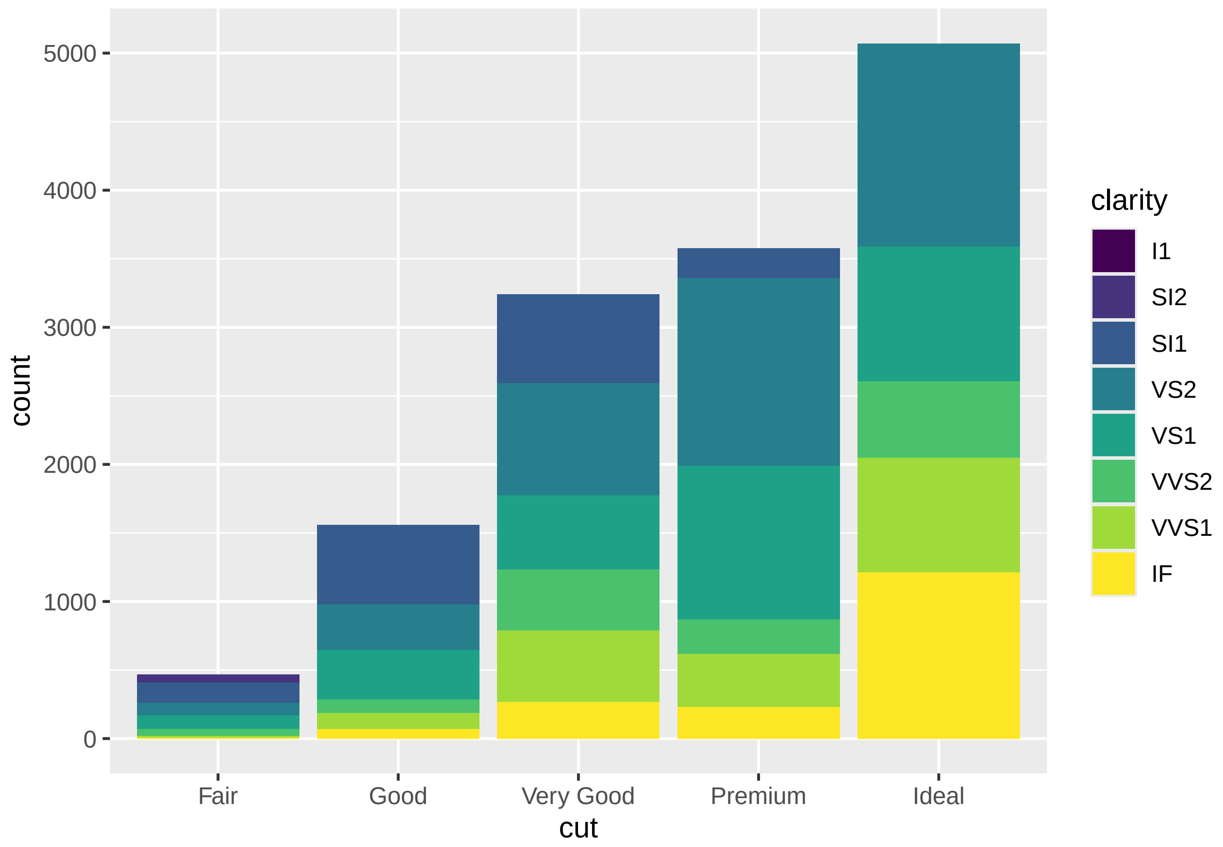
```
#테두리 칠하기
ggplot(data=diamonds) +
    geom_bar(aes(cut, fill=cut))
```



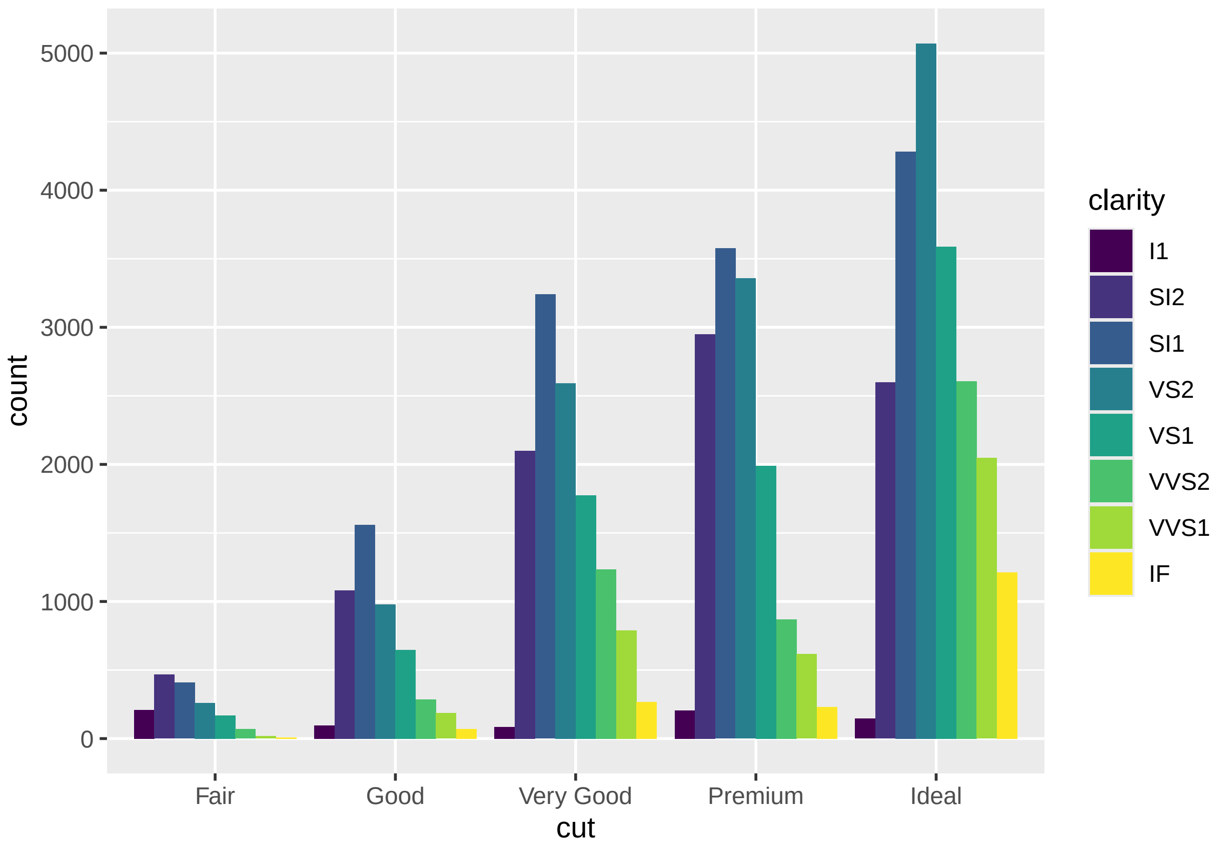
```
#색칠하기

ggplot(data=diamonds) +
    geom_bar(aes(cut, fill=clarity))
```



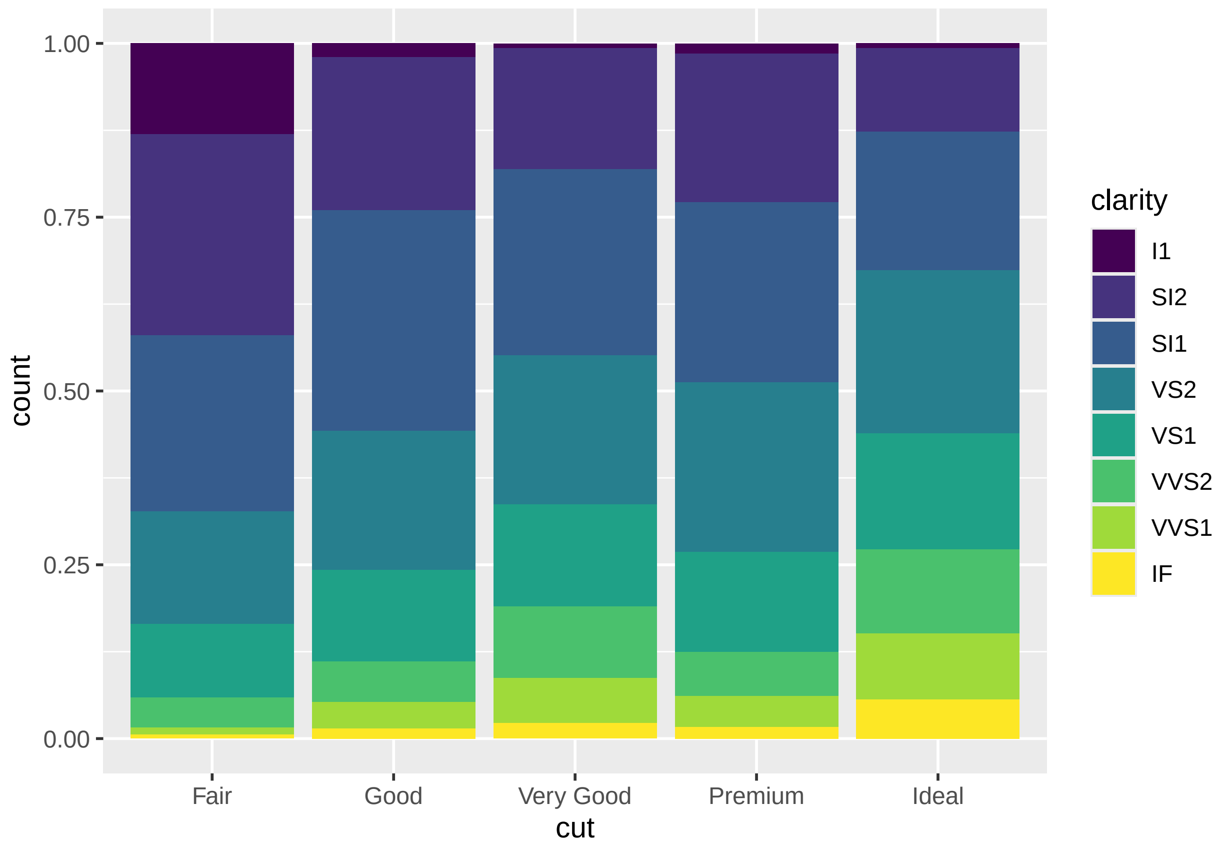
```
#clarity에 따른 색칠하기도 가능, 위에서 했던 느낌

ggplot(data=diamonds) +
    geom_bar(aes(cut, fill=clarity), position="identity")
```



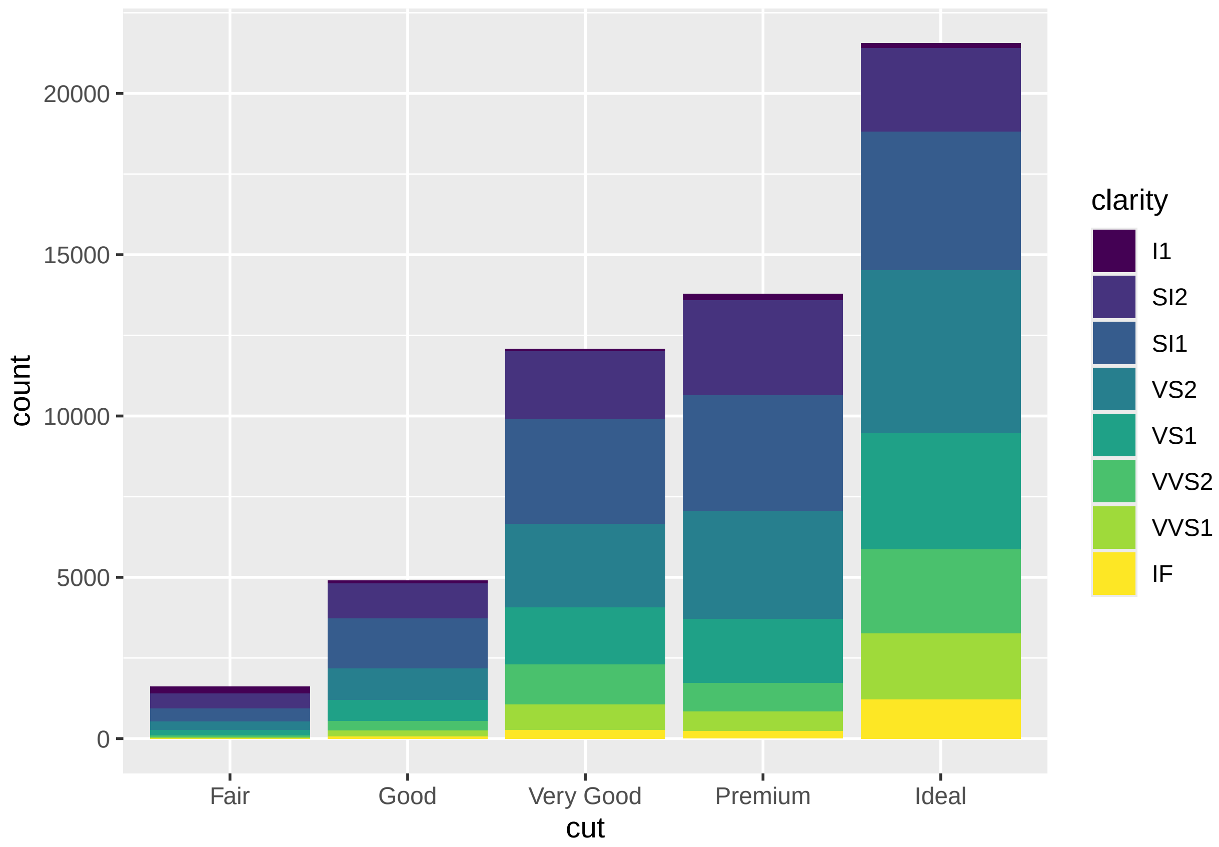
```
#포지션의 인수를 사용, identity는 뭔지 모름

ggplot(data=diamonds) +
    geom_bar(aes(cut, fill=clarity),position="dodge")
```



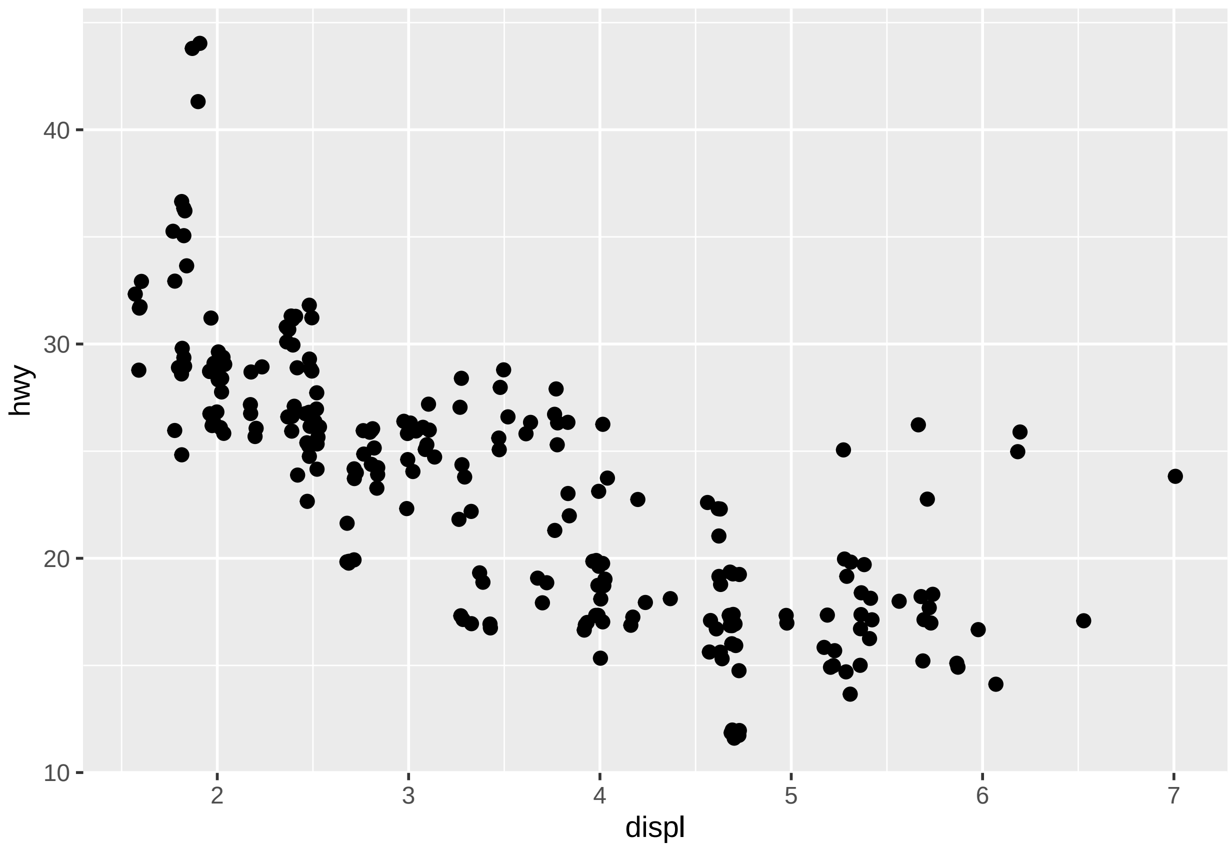
```
#각각 보여주기

ggplot(data=diamonds) +
    geom_bar(aes(cut, fill=clarity),position="fill")
```



```
#높이 정규화

ggplot(data=diamonds) +
    geom_bar(aes(cut, fill=clarity),position="stack")
```



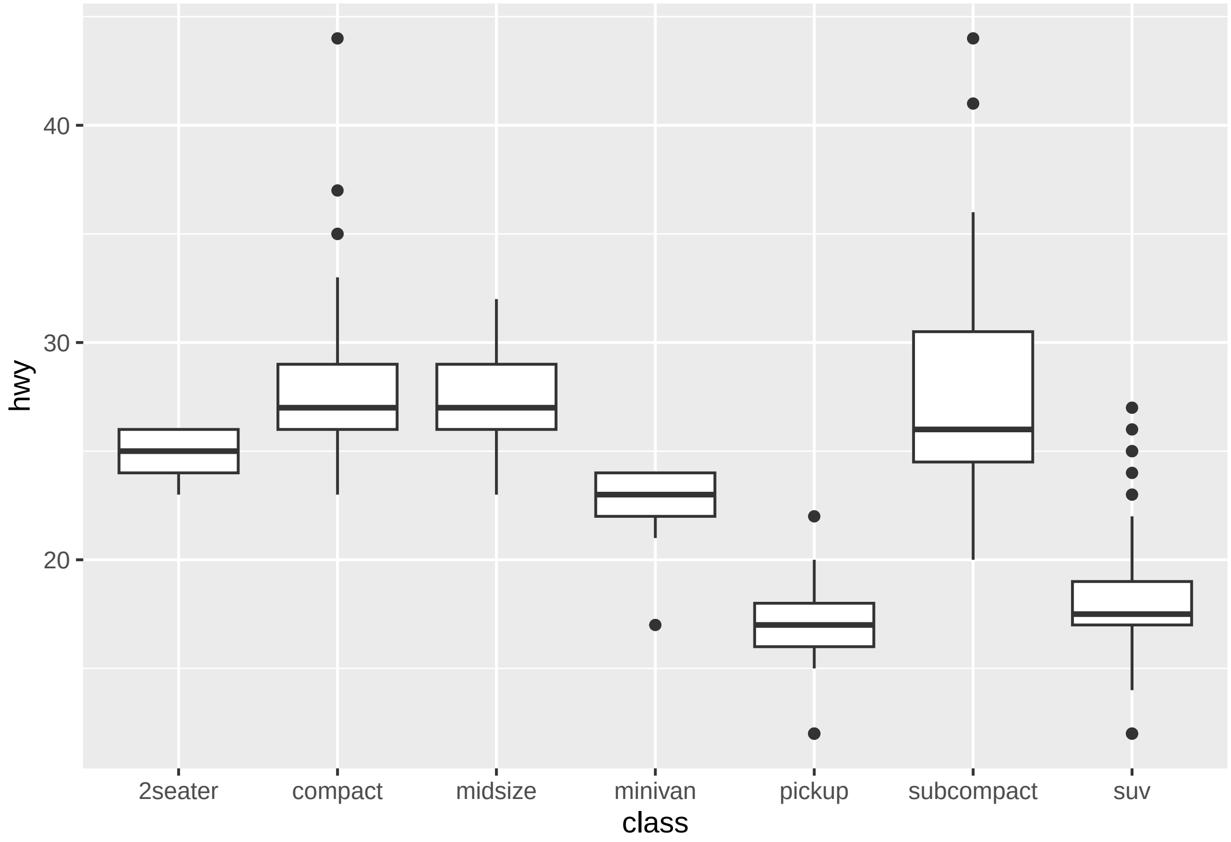
```
#기본


ggplot(data = mpg) +
    geom_point(aes(displ, hwy), position = "jitter", size=2)
```



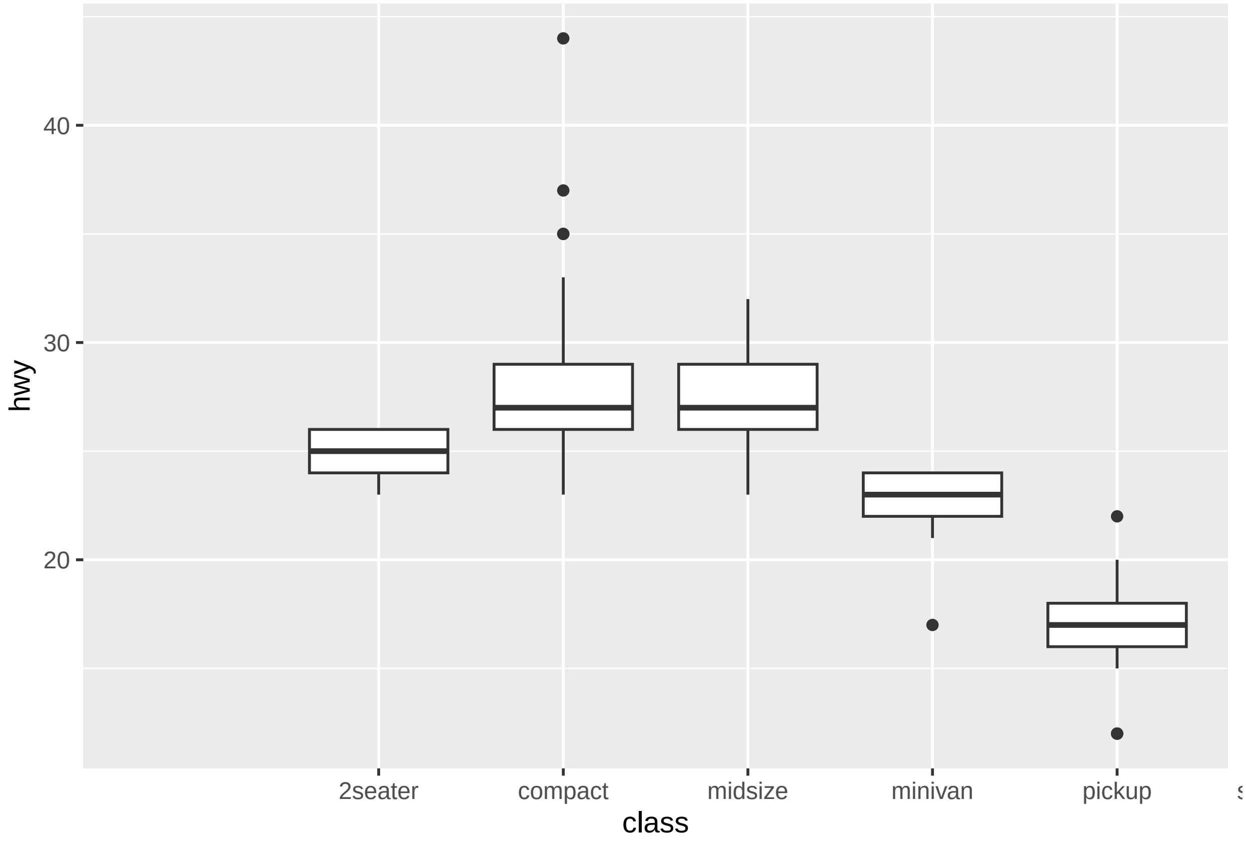
```
#포인트에서도 가능
```

## 박스플롯을 그려보아요

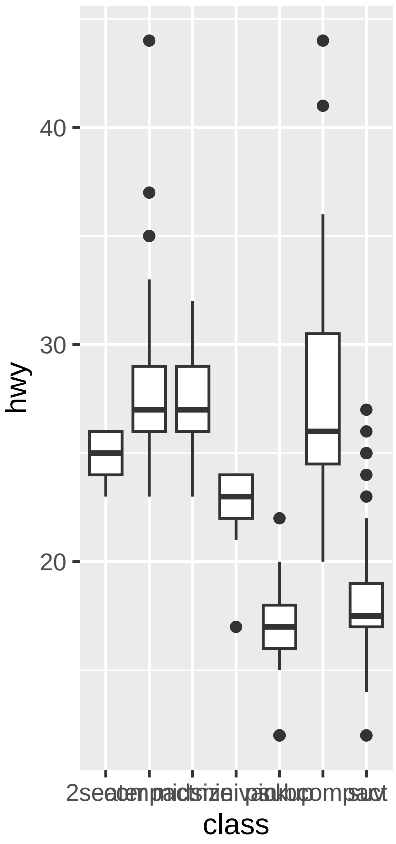
```
ggplot(data=mpg) +
    geom_boxplot(aes(x=class, y=hwy))
```



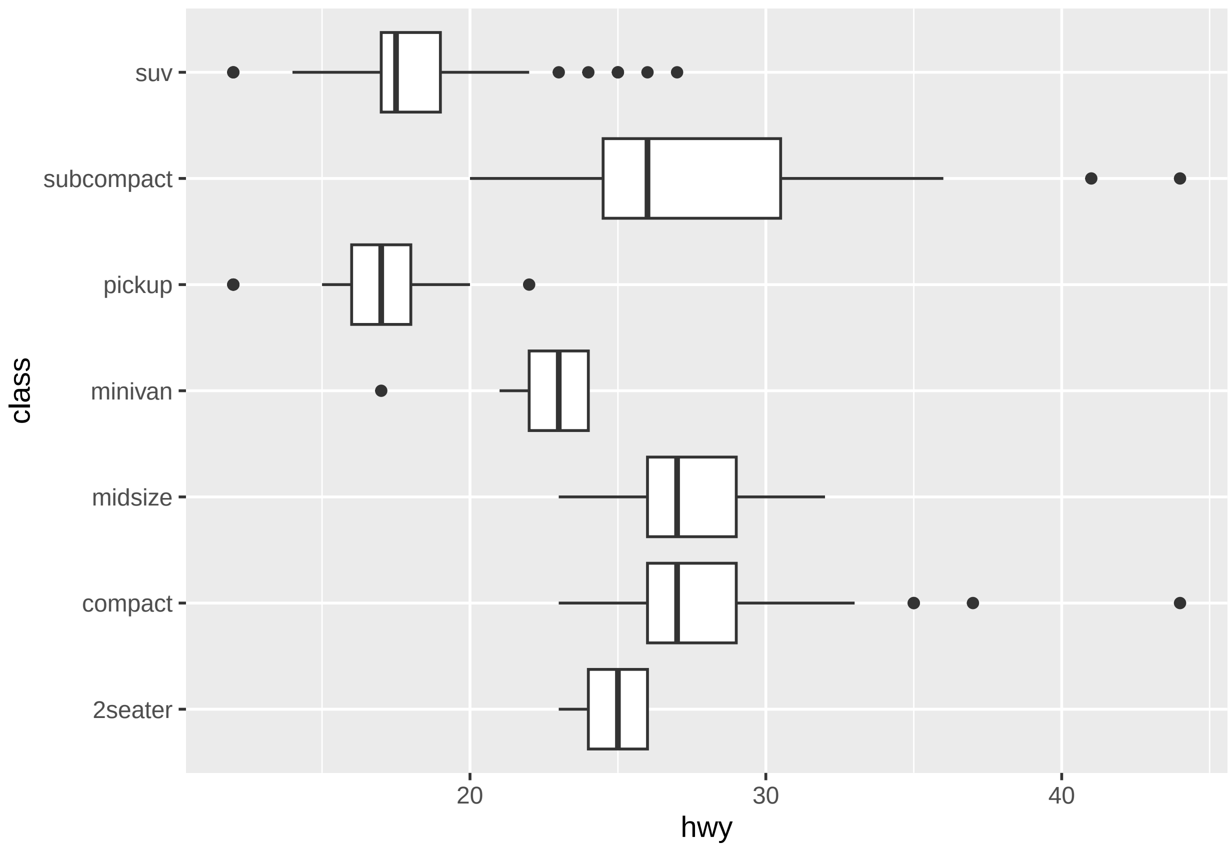
```r
ggplot(data=mpg) +
    geom_boxplot(aes(x=class, y=hwy)) +
    coord_cartesian(xlim=c(0,5))
```



```
#coord는 좌표평면을 수정

ggplot(data=mpg) +
    geom_boxplot(aes(x=class, y=hwy)) +
    coord_fixed(ratio=1/2)
```



```
#비율  조정

ggplot(data=mpg) +
    geom_boxplot(aes(x=class, y=hwy)) +
    coord_flip()
```



```
#돌리기!
```

## 지도를 그려보아요

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.4.1
```

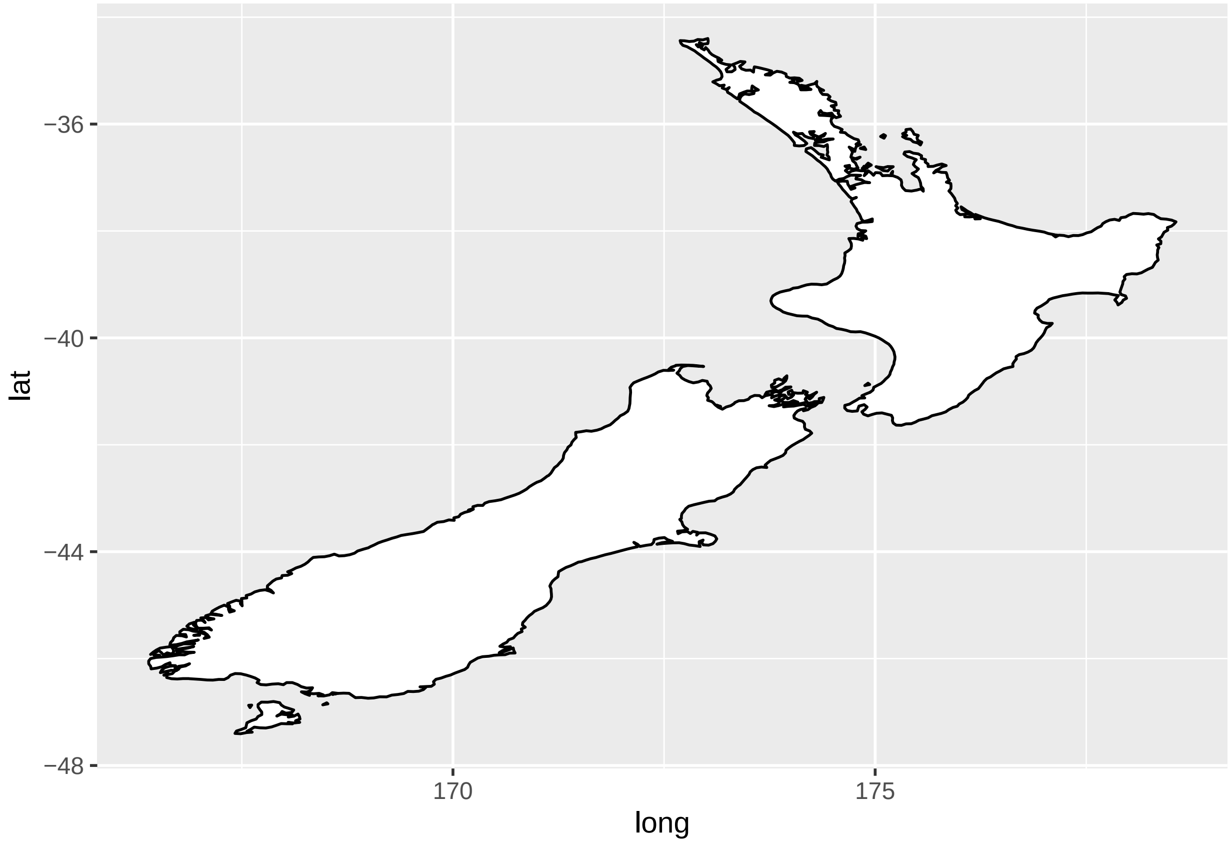
```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

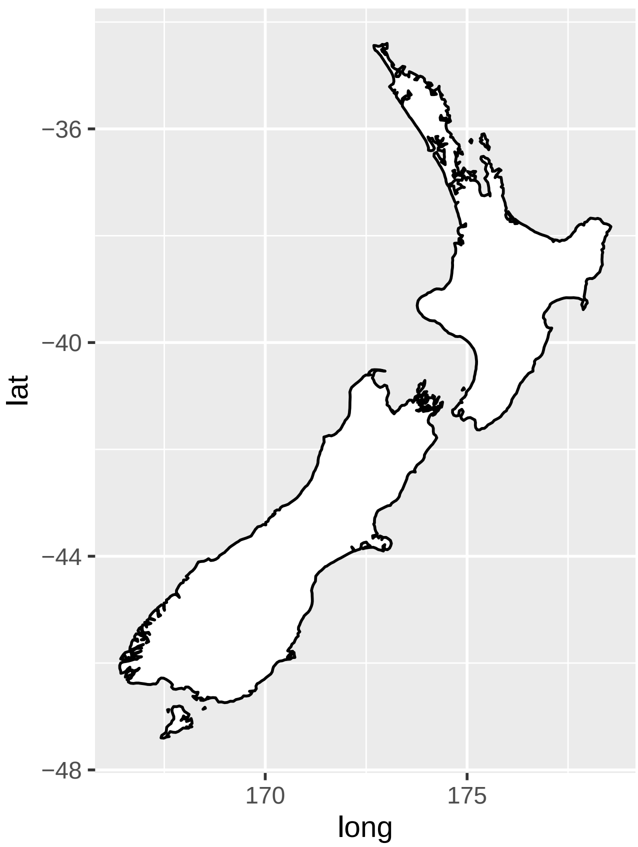
```
nz <- map_data("nz")

head(nz) #지도 데이터
```

```
##       long       lat group order       region subregion
## 1 172.7433 -34.44215     1     1 North.Island      <NA>
## 2 172.7983 -34.45562     1     2 North.Island      <NA>
## 3 172.8528 -34.44846     1     3 North.Island      <NA>
## 4 172.8986 -34.41786     1     4 North.Island       <NA>
## 5 172.9593 -34.42503     1     5 North.Island       <NA>
## 6 173.0184 -34.39895     1     6 North.Island       <NA>
```

```
ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", colour = "black")
```

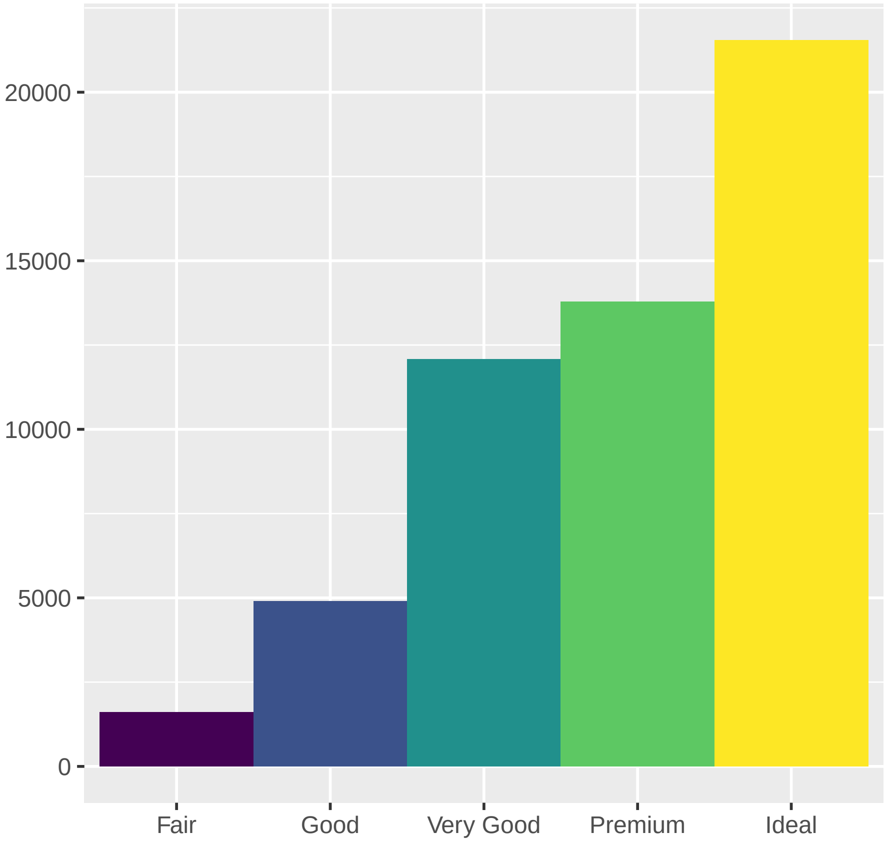


```
#지도 그리기

ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", colour = "black") +
  coord_quickmap() #축적으로 배치
```

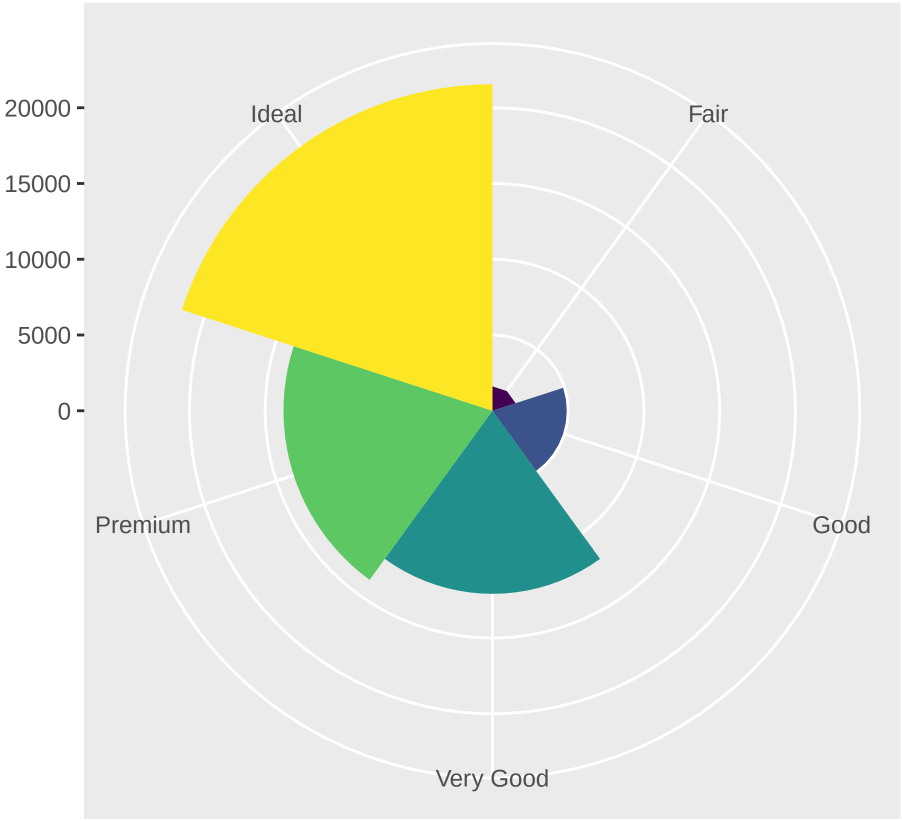


## 또 다른 느낌

```
bar <- ggplot(data = diamonds) +
  geom_bar(
    mapping = aes(x = cut, fill = cut),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)

bar
```



```
bar+coord_polar() #극좌표 표현
```



## ggplot의 플롯

ggplot(data =) + (mapping = aes(), stat = , position =) + +

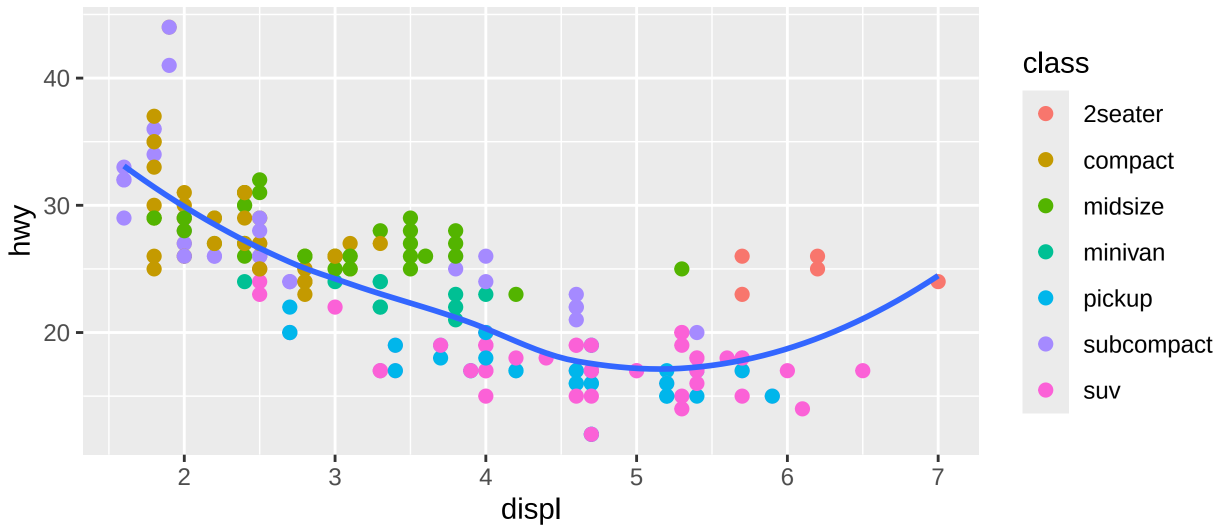
이러합니다

디테일을 잡읍시다

labs를 추가하면 제목 이름 범주등등을 추가할 수 있습니다.

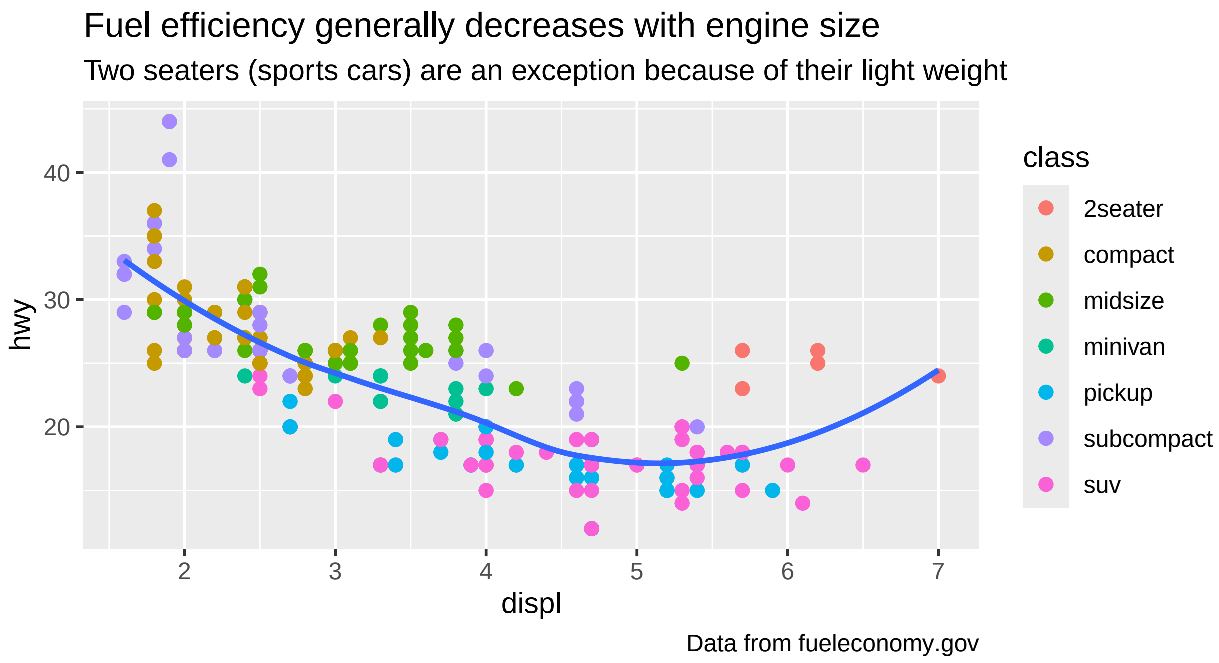
```
grp= ggplot(mpg, aes(displ, hwy)) +
     geom_point(aes(color = class),size= 2) +
     geom_smooth(se = FALSE) +
     theme(aspect.ratio = 1/2)

grp
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



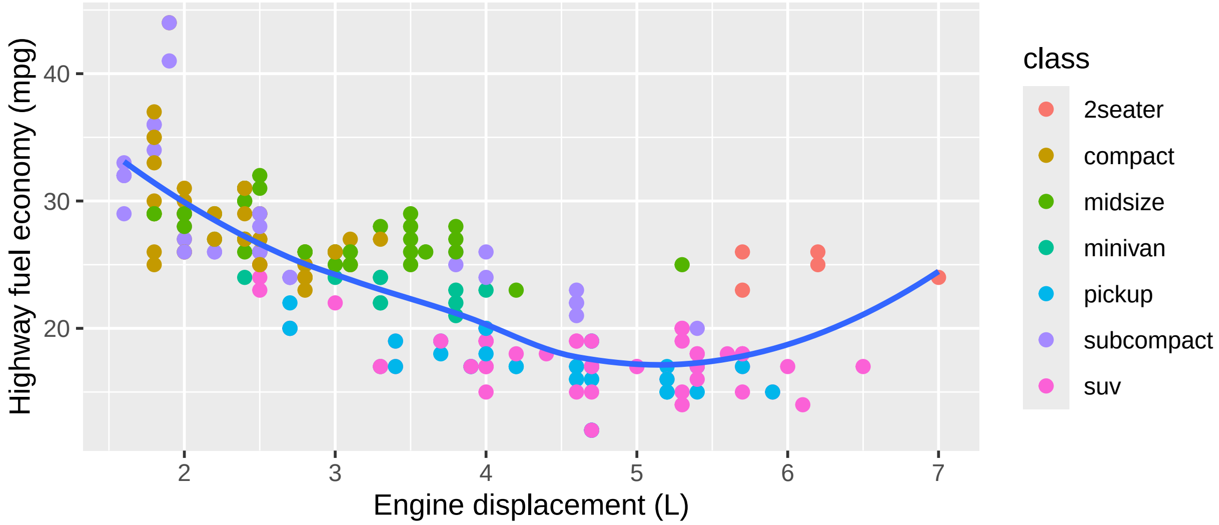
```
grp + labs(
  title = "Fuel efficiency generally decreases with engine size",
  subtitle = "Two seaters (sports cars) are an exception because of their light weight",
  caption = "Data from fueleconomy.gov"
)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

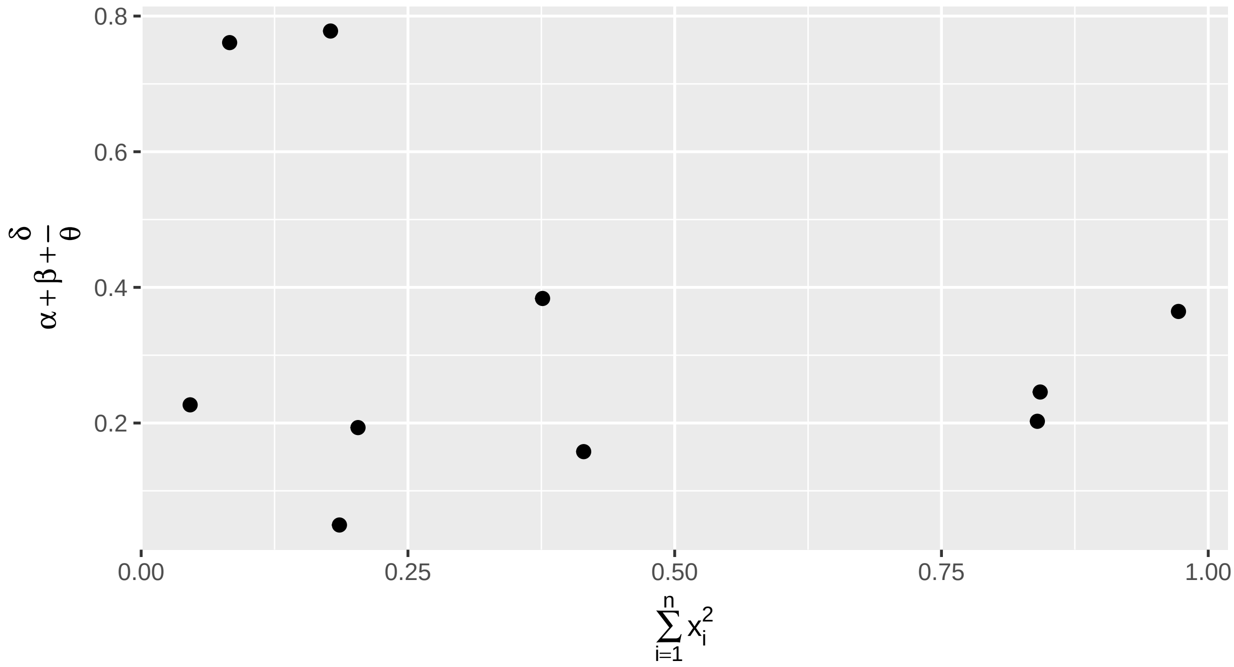


```
#타이틀 서브타이틀 캡션 추가

grp+ labs( x = "Engine displacement (L)",
  y = "Highway fuel economy (mpg)")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



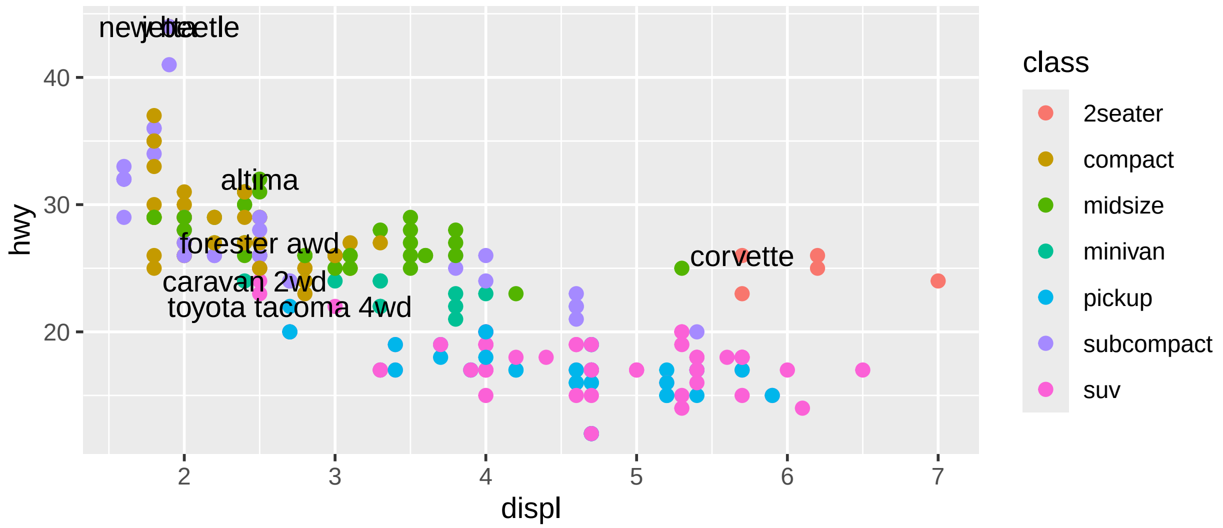
```
#x, y 레이블 추가

df <- tibble(x = runif(10), y = runif(10))
ggplot(df, aes(x, y)) + geom_point(size=2) +
  labs(
    x = quote(sum(x[i] ^ 2, i == 1, n)),
    y = quote(alpha + beta + frac(delta, theta))
  ) +
  theme(aspect.ratio = 1/2)
```



```
#quote 사용으로 수학 방정식도 레이블로 넣을 수 있음

#mpg 데이터셋의 class에서 연비 좋은놈들만 남겨보자
best_in_class = mpg %>% group_by(class) %>% filter(row_number(desc(hwy))==1)

best_in_class #이놈들이 연비 좋은놈들
```

```
## # A tibble: 7 x 11
## # Groups:   class [7]
##   manufacturer model        displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr>        <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 chevrolet    corvette       5.7  1999     8 manu~ r        16    26 p     2sea~
## 2 dodge        caravan 2wd    2.4  1999     4 auto~ f        18    24 r     mini~
## 3 nissan       altima         2.5  2008     4 manu~ f        23    32 r     mids~
## 4 subaru       forester a~    2.5  2008     4 manu~ 4        20    27 r     suv
## 5 toyota       toyota tac~    2.7  2008     4 manu~ 4        17    22 r     pick~
## 6 volkswagen   jetta          1.9  1999     4 manu~ f        33    44 d     comp~
## 7 volkswagen   new beetle     1.9  1999     4 manu~ f        35    44 d     subc~
```

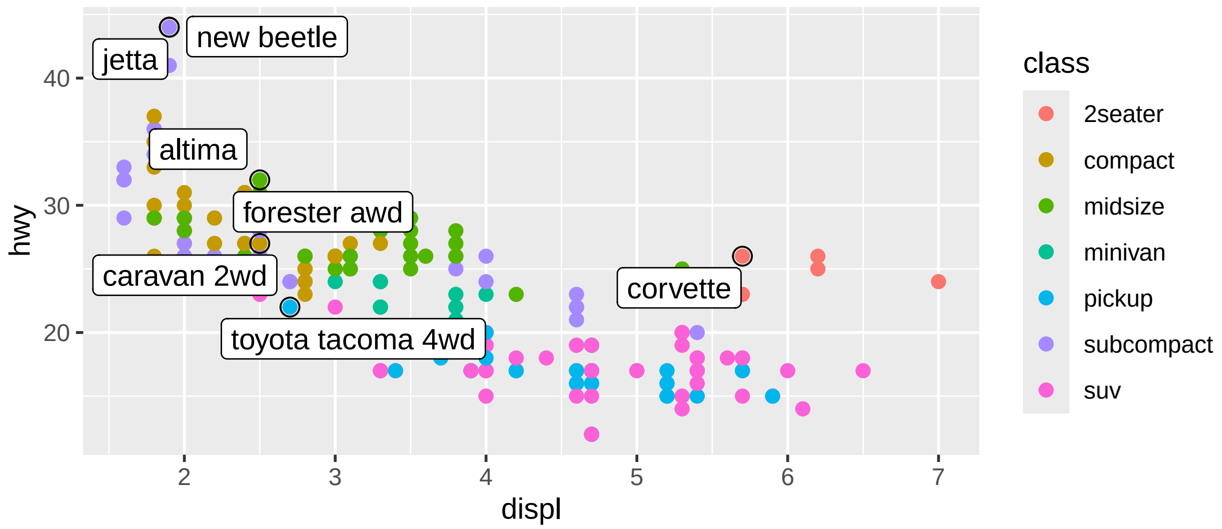
```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(colour = class),size=2) +
  geom_text(aes(label = model), data = best_in_class) +
  theme(aspect.ratio = 1/2)
```



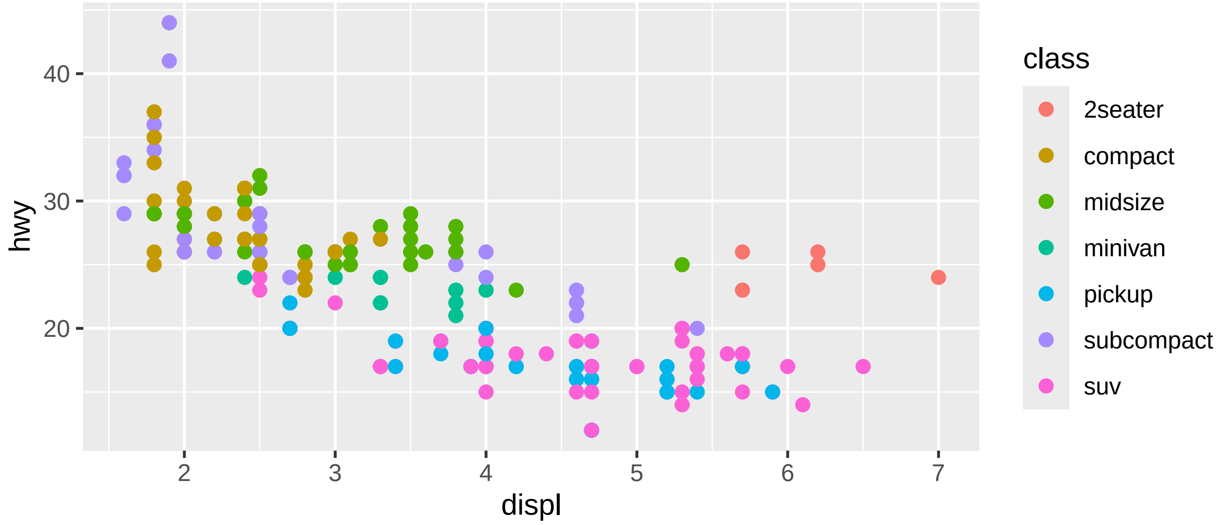
```r
#best_in_class를 이용해 원하는 데이터에 텍스트 추가하기


library("ggrepel")
```

```
## Warning: package 'ggrepel' was built under R version 4.4.1
```

```r
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class), size=2) +
  geom_point(size = 3, shape = 1, data = best_in_class) +
  ggrepel::geom_label_repel(aes(label = model), data = best_in_class) +
  theme(aspect.ratio = 1/2)
```

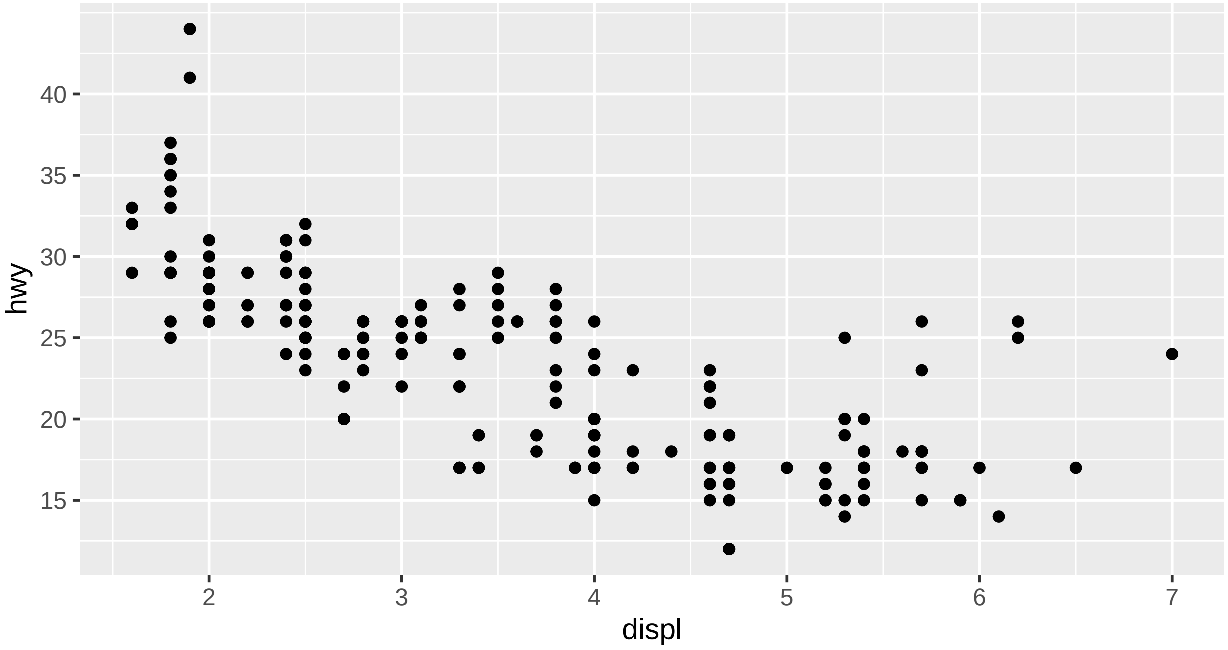


```r
#ggrepel 라이브러리로 보기 좋게 만들 수 있음



ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(colour = class),size=2) +
  scale_x_continuous() +
  scale_y_continuous() +
  scale_colour_discrete() +
  theme(aspect.ratio = 1/2)
```

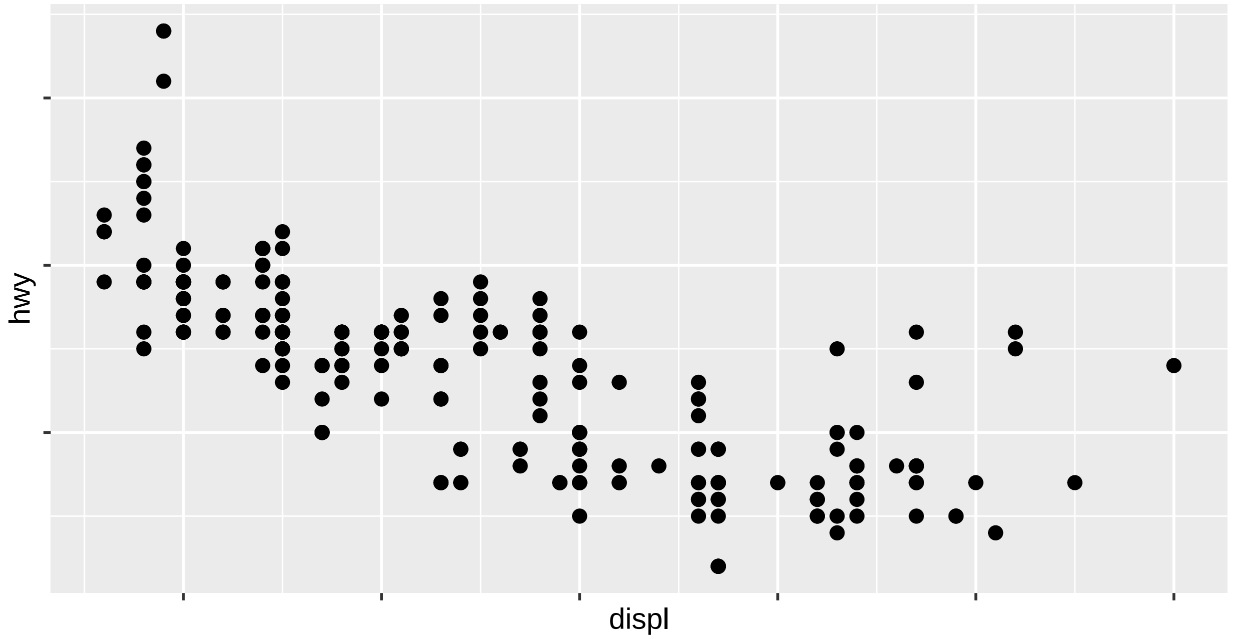


```
#스케일

ggplot(mpg, aes(displ, hwy)) +
  geom_point(siez=2) +
  scale_y_continuous(breaks = seq(15, 40, by = 5)) +
  theme(aspect.ratio = 1/2)
```

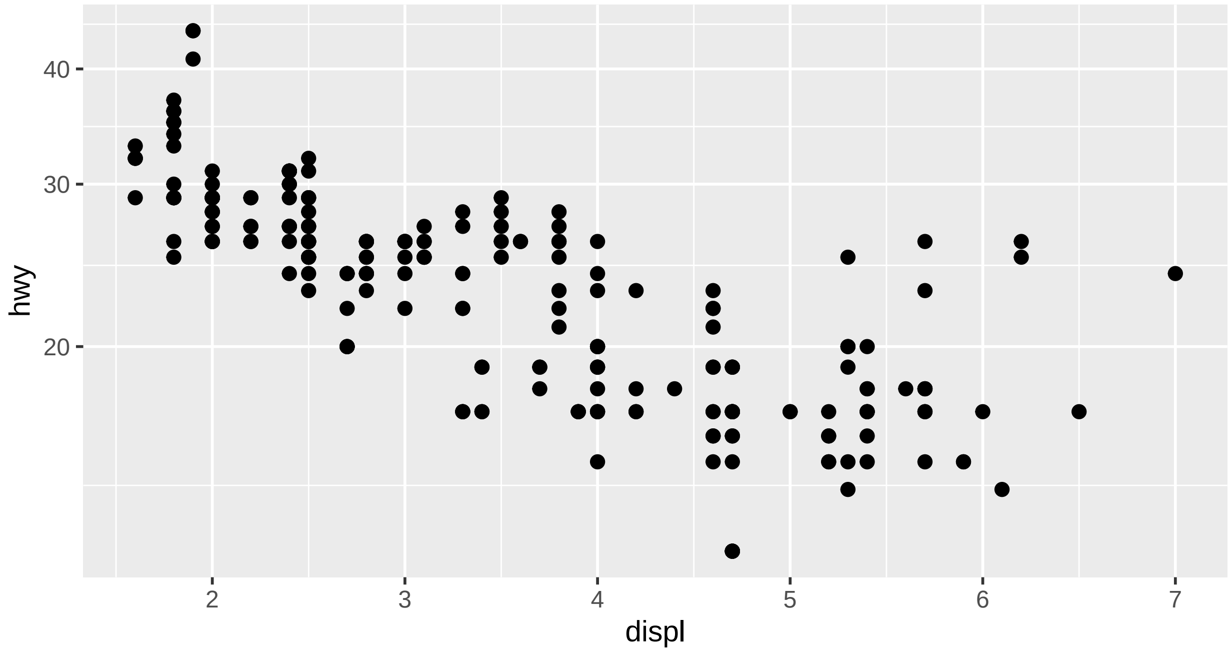
```
## Warning in geom_point(siez = 2): Ignoring unknown parameters: `siez`
```



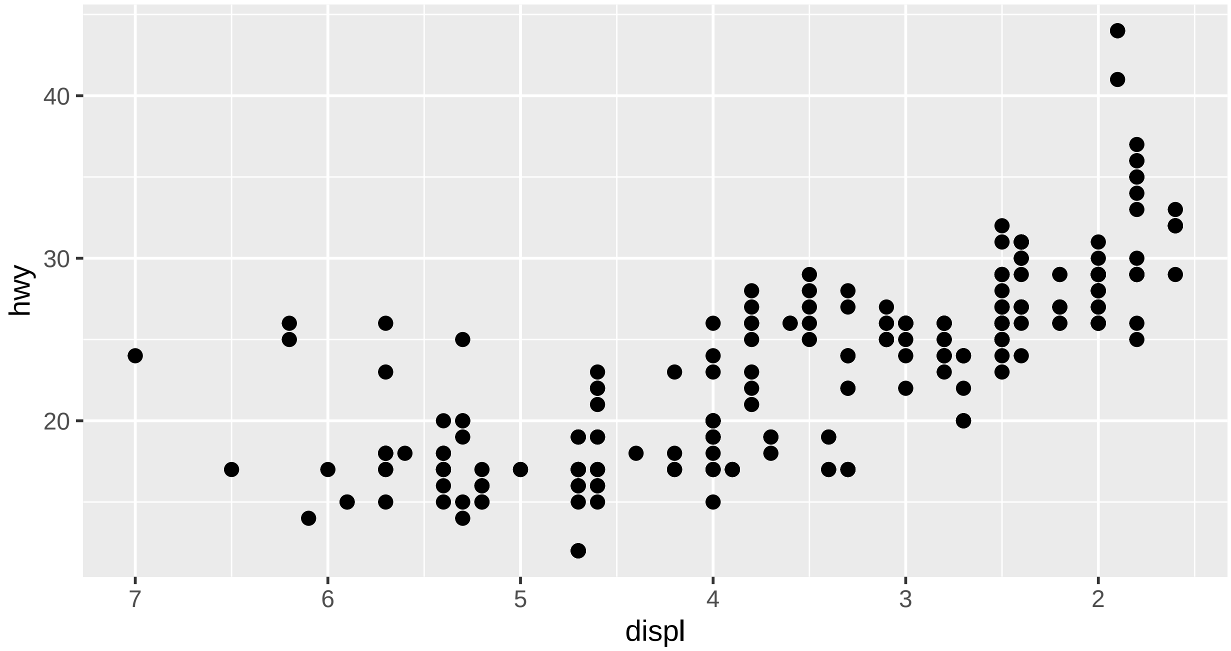
```
#축 스케일 쪼개기

ggplot(mpg, aes(displ, hwy)) +
  geom_point(size=2) +
  scale_x_continuous(labels = NULL) +
  scale_y_continuous(labels = NULL) +
  theme(aspect.ratio = 1/2)
```



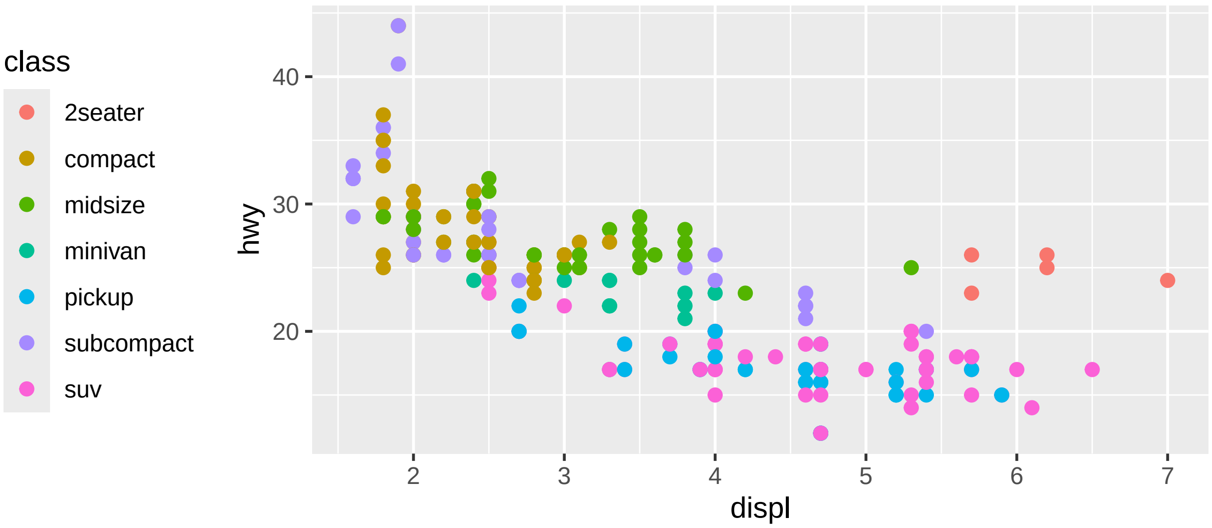
```
#축 레이블 없애기

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(size=2) +
  scale_y_log10() +
  theme(aspect.ratio = 1/2)
```



```
#로그  스케일

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(size=2) +
  scale_x_reverse() +
  theme(aspect.ratio = 1/2)
```

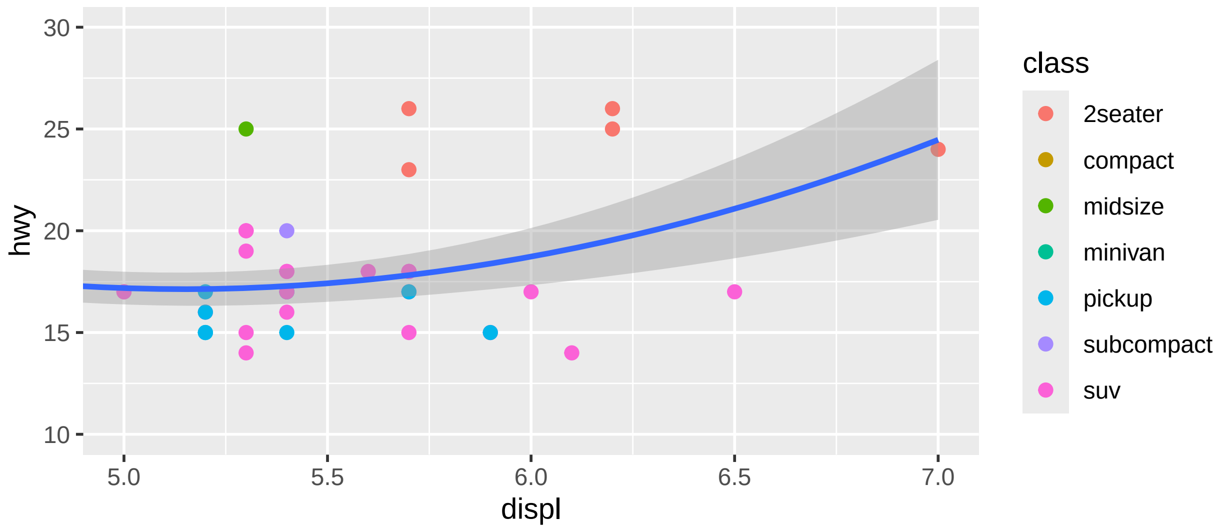


```
#x축 반전

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color=class), size=2) +
  theme(legend.position = "left", aspect.ratio = 1/2)
```



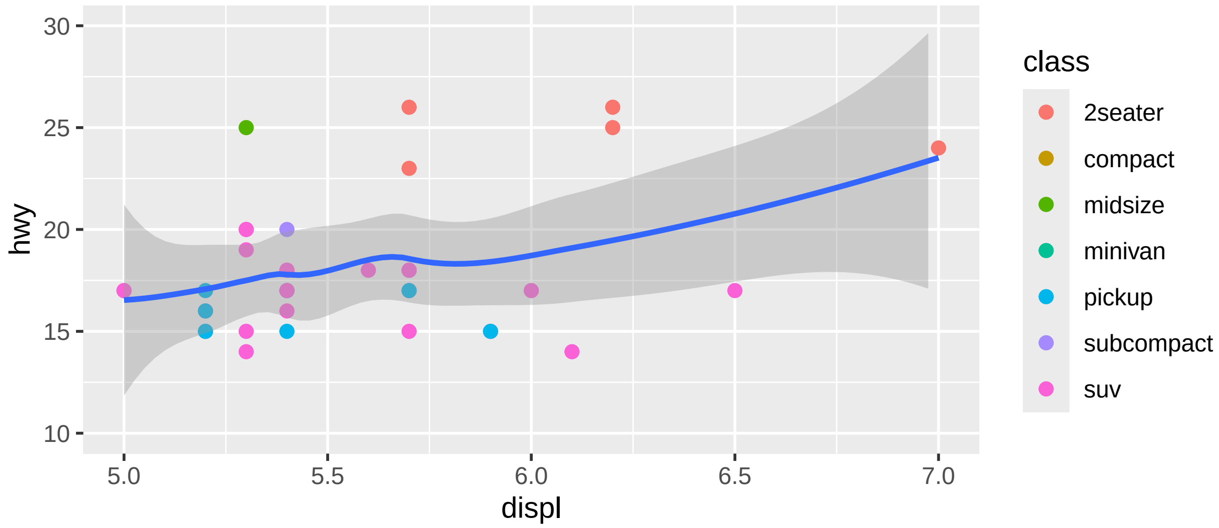
```
#범주  옮기기  left right top bottom none



ggplot(data=mpg, aes(displ, hwy)) +
    geom_point(aes(color=class),size=2) +
    geom_smooth() +
    coord_cartesian(xlim=c(5,7), ylim=c(10, 30)) +
    theme(aspect.ratio = 1/2)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```r
#얘는 그림을 자름

ggplot(data=mpg, aes(displ, hwy)) +
    geom_point(aes(color=class),size=2) +
    geom_smooth() +
    theme(aspect.ratio = 1/2) +
    xlim(5,7) + ylim(10,30)
```

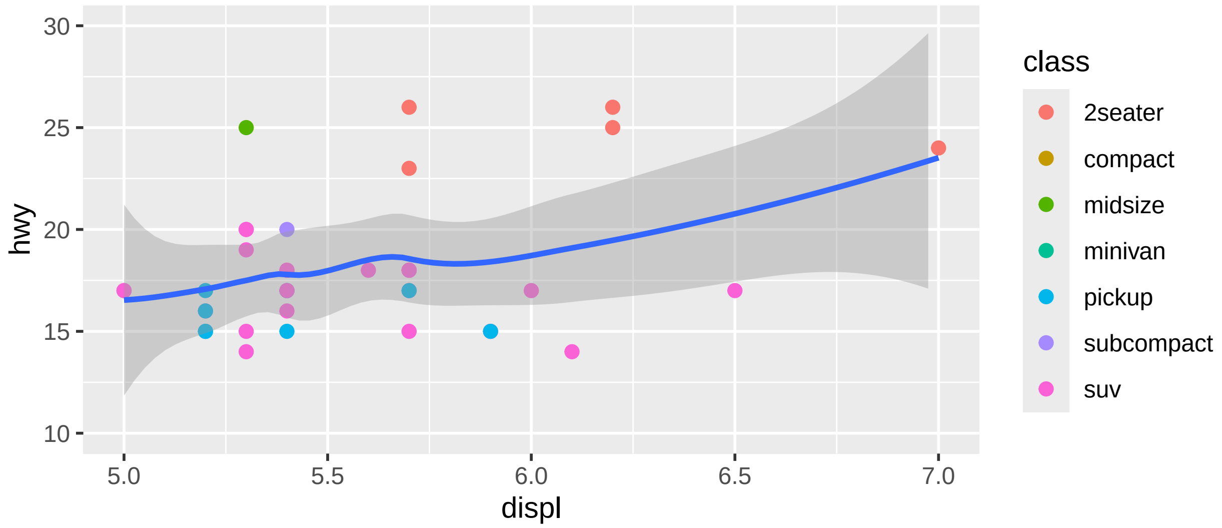
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 196 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 196 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
#얘는 데이터를 자름

ggplot(data=mpg, aes(displ, hwy)) +
    geom_point(aes(color=class),size=2) +
    geom_smooth() +
    theme(aspect.ratio = 1/2) +
    scale_x_continuous(limits = c(5, 7)) +
    scale_y_continuous(limits = c(10, 30))
```

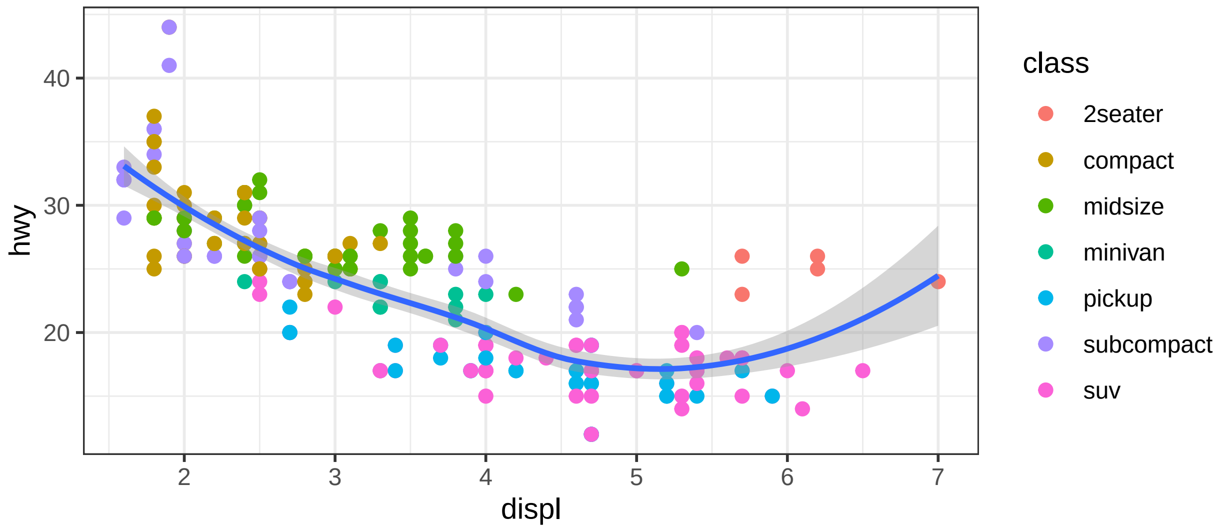
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 196 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 196 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
#이것도 데이터를 자름

#물론 데이터를 직접 잘라도됨.ㅋㅋ

ggplot(data=mpg, aes(displ, hwy)) +
    geom_point(aes(color=class),size=2) +
    geom_smooth() +
    theme_bw() +
    theme(aspect.ratio = 1/2)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
#테마로 꾸미기


ggsave("my plot.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
#저장가능
```