Homework3

201982188 통계학과 박현주

2020/9/25

Chap3. Graphics with ggplot2

```
library(tidyverse)
library(ggplot2)
```

1. Basics - pipes

[Your turn 01]

1. Load data diamonds

```
data("diamonds")
```

2. Find the summary of the data frame diamond and extract the first ten rows of the diamonds data using pipes

```
diamonds %>% head(10)
```

```
# A tibble: 10 x 10
   carat cut       color clarity depth table price     x     y     z
   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
 7 0.24  Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
 8 0.26  Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
 9 0.22  Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
10 0.23  Very Good H     VS1      59.4    61   338  4     4.05  2.39
```

```
diamonds %>% str
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':   53940 obs. of  10 variables:
 $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
 $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

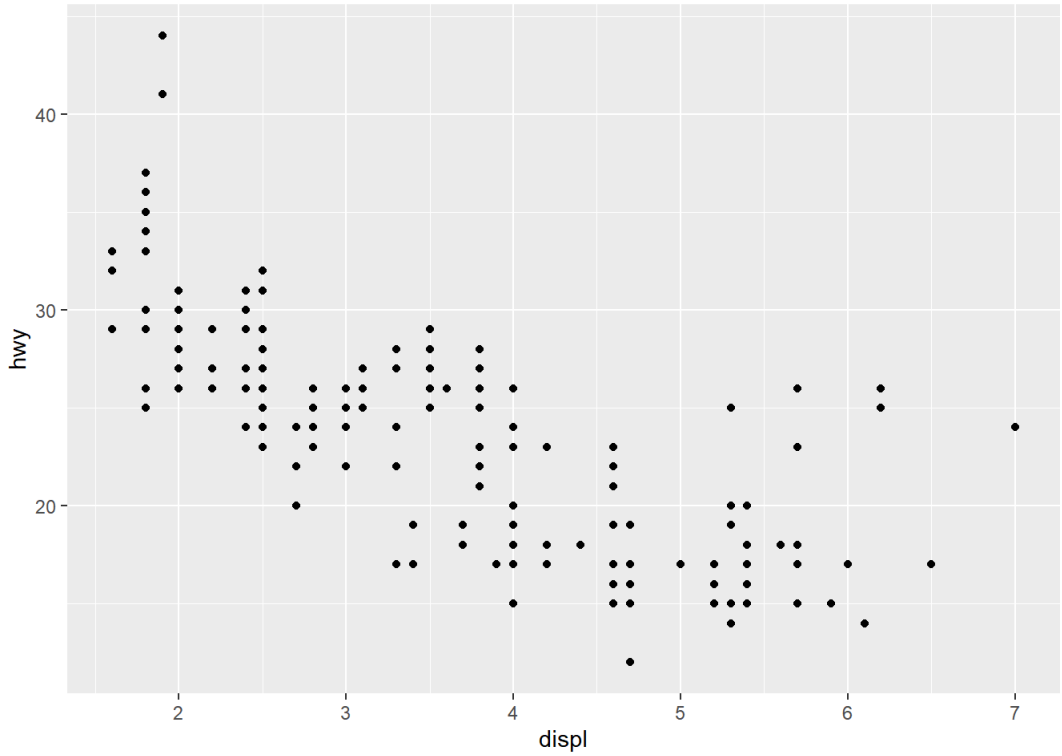
2. mpg data (1)

[example]

```
mpg %>% head(5)
```

```
# A tibble: 5 x 11
  manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
  <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
```

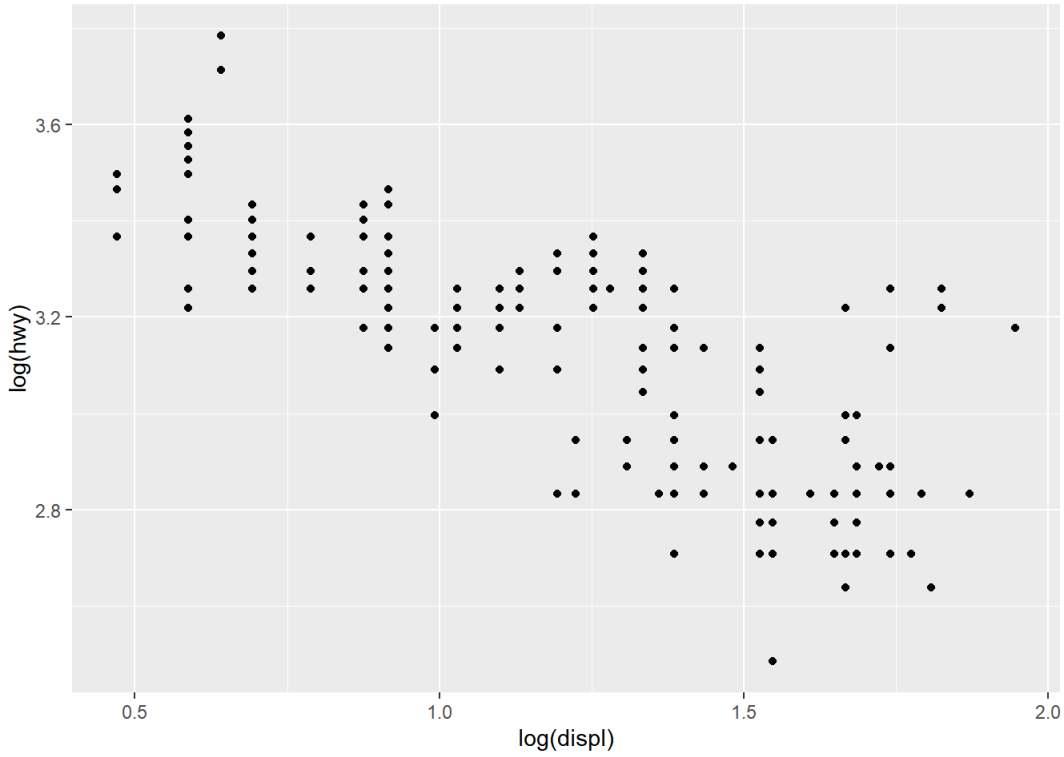
```
mpg %>% ggplot(mapping = aes(x=displ, y=hwy)) + geom_point()
```



displ이 커질수록 hwy이 작아지고 있다.

- log transformation

```
mpg %>% ggplot(mapping = aes(x=log(displ), y=log(hwy))) + geom_point()
```



[Your turn 02]

1. ggplot(mpg)

```
ggplot(mpg)
```

아무것도 안보인다.

2. How many rows are in mpg? How many columns?

```
dim(mpg);glimpse(mpg)
```

```
[1] 234  11
```

```
Observations: 234
Variables: 11
$ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi"...
$ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro"...
$ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0,...
$ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, ...
$ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, ...
$ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "a...
$ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4",...
$ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17...
$ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25...
$ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
$ class        <chr> "compact", "compact", "compact", "compact", "compact",...
```

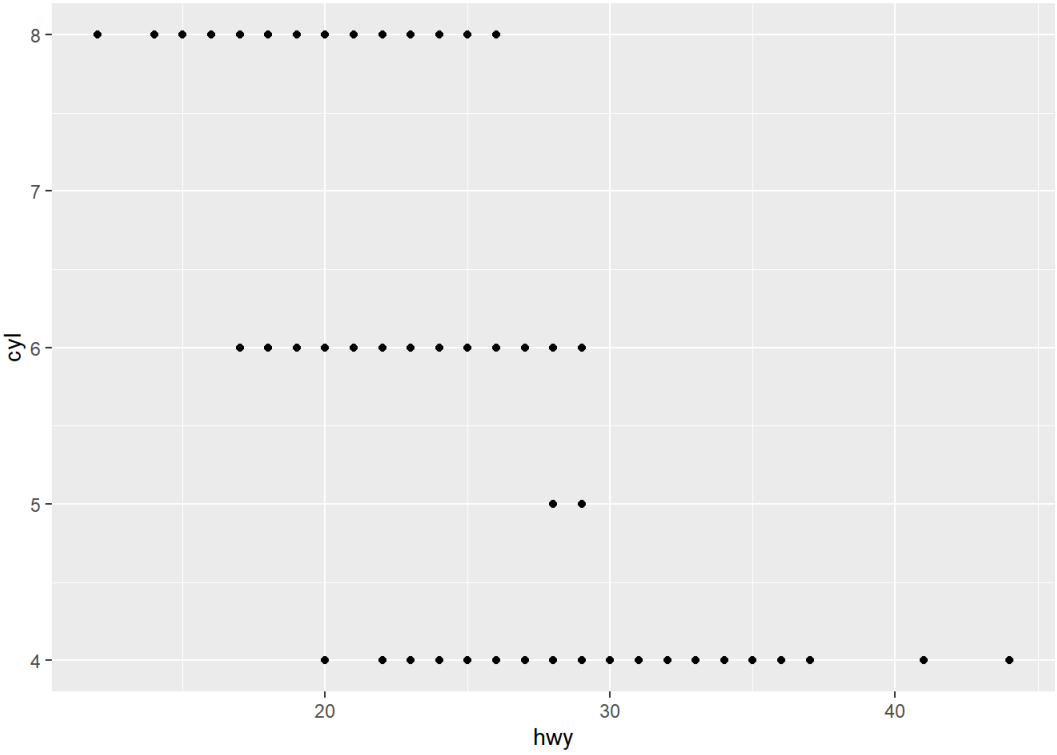
234개의 관측치와 11개의 변수를 가지고 있다.

3. What does the drv variable describe? Read the help for ?mpg to find out.

```
?mpg
```

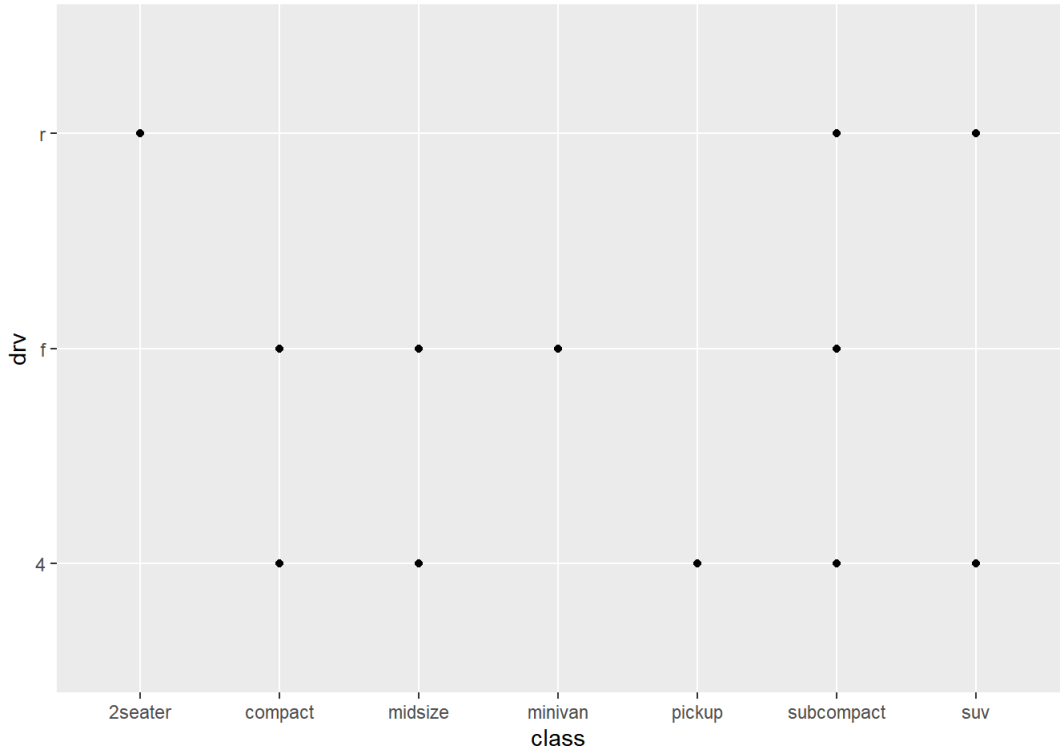
4. Make a scatterplot of hwy vs cyl.

```
mpg %>% ggplot(aes(x=hwy, y=cyl)) + geom_point()
```



5. What happens if you make a scatterplot of class vs drv? Why is the plot not useful?

```
mpg %>% ggplot(aes(x=class, y=drv)) + geom_point()
```



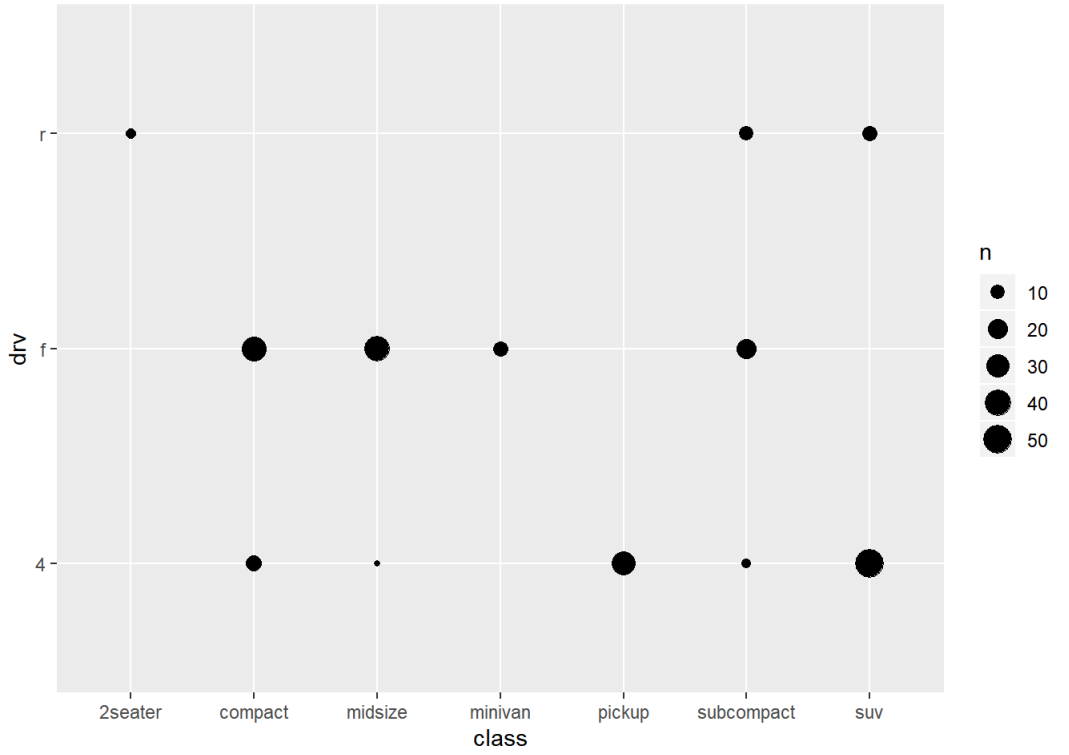
두 variable 모두 factor이기 때문에 의미가 없다.

- [5번의 대안]

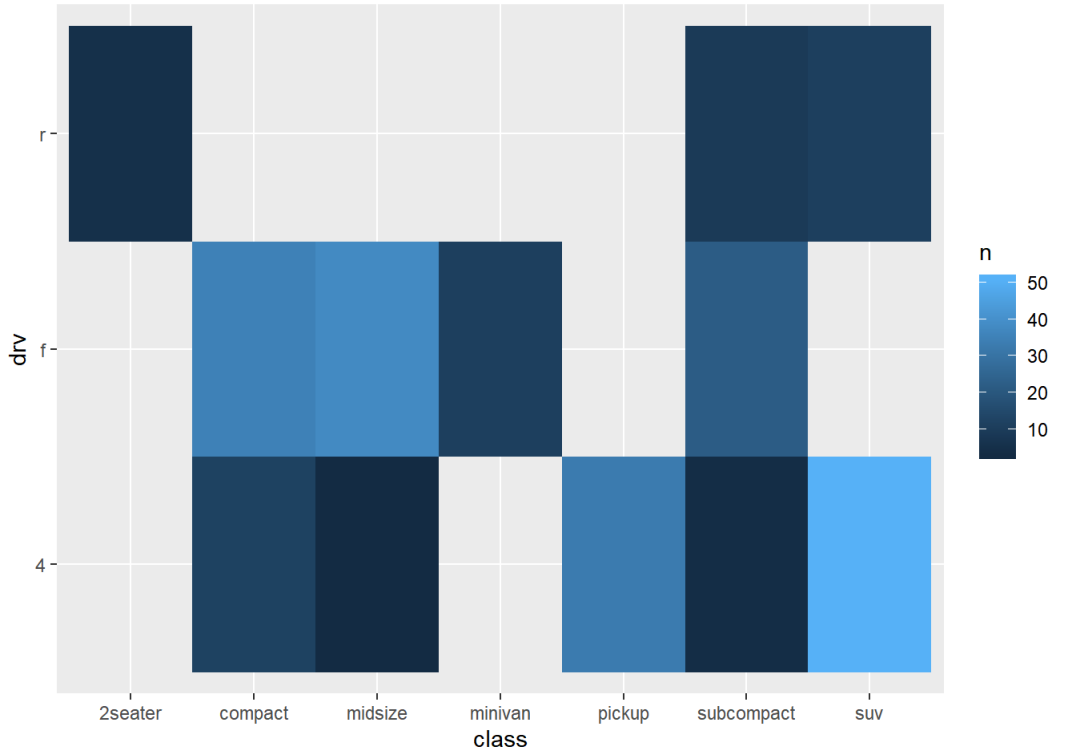
```
count(mpg, drv, class) # facotr인 경우 각각 몇개가 있는 지 확인가능하다.
```

```
# A tibble: 12 x 3
   drv   class          n
   <chr> <chr>      <int>
 1 4     compact       12
 2 4     midsize        3
 3 4     pickup        33
 4 4     subcompact     4
 5 4     suv           51
 6 f     compact       35
 7 f     midsize       38
 8 f     minivan       11
 9 f     subcompact    22
10 r     2seater        5
11 r     subcompact     9
12 r     suv           11
```

```
mpg %>% ggplot(aes(x=class, y=drv)) + geom_count()
```




```
mpg %>% count(class, drv) %>% #n이 생성된다.(새로운 변수 생성)
  ggplot(aes(x=class, y = drv)) +
  geom_tile(aes(fill = n))
```

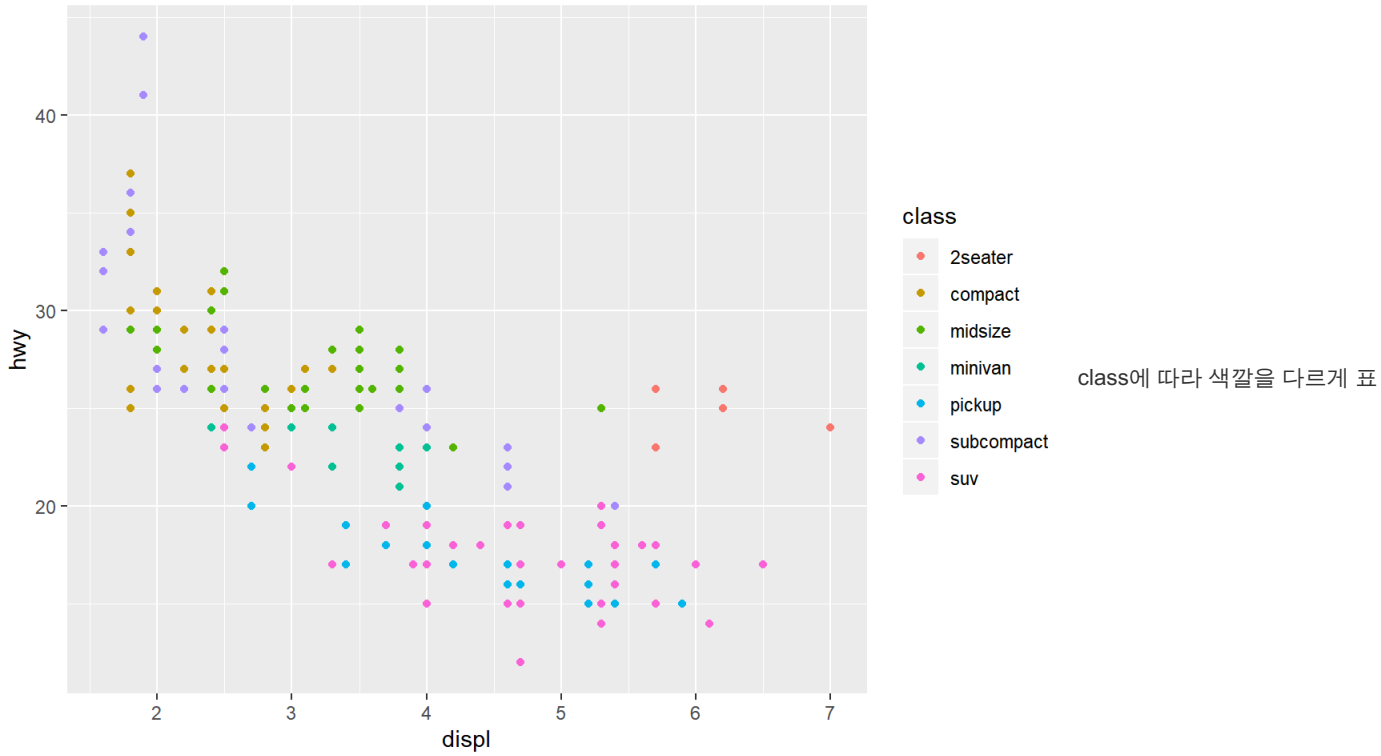


tile모양의 그래프가 된다. 점을 찍는 것이 아니라 색깔로 보여준다. 그리고 n에 따라 색을 채워준다.(n이 커지면 색이 진해진다.)

3. mpg data (2)

- color

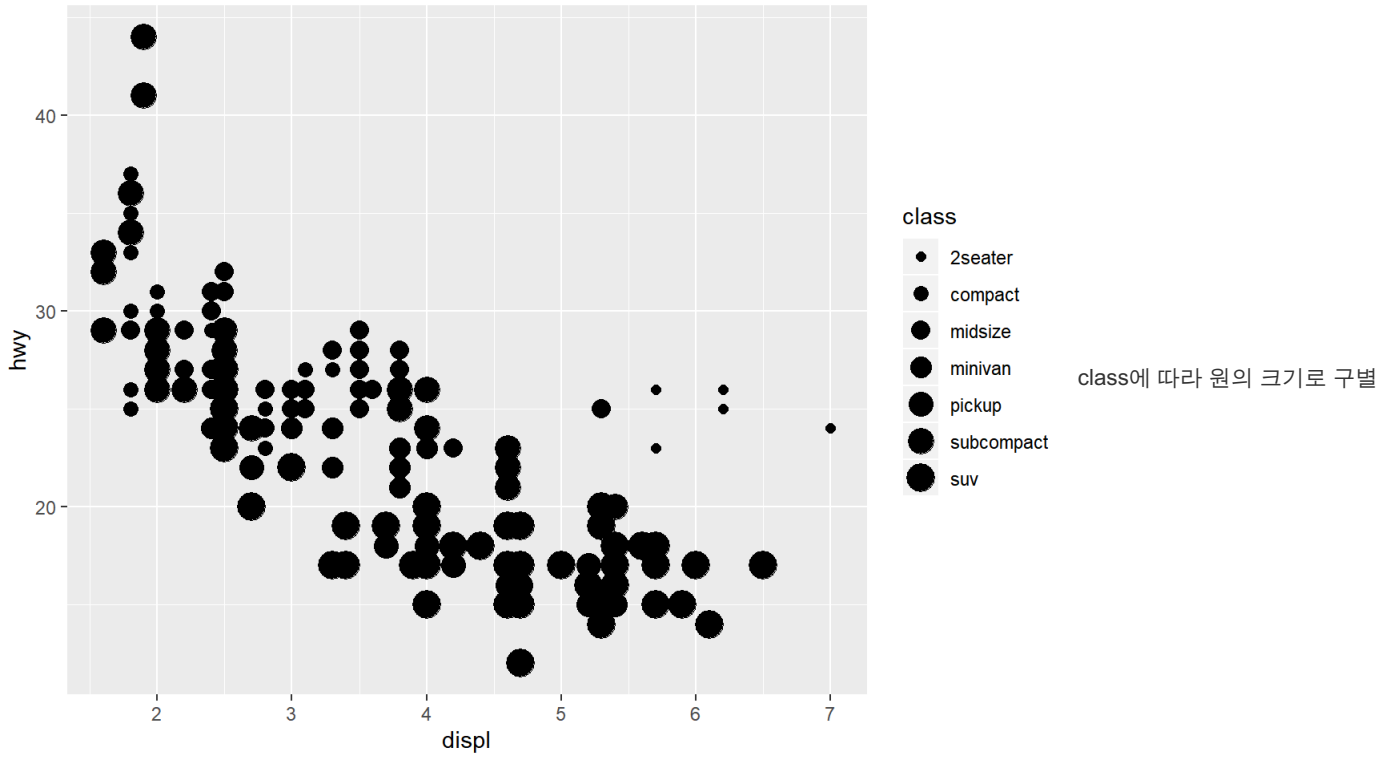
```
mpg %>% ggplot(aes(x=displ, y=hwy, color=class)) + geom_point() #Coloring variables
```



class에 따라 색깔을 다르게 표현한다.

- size

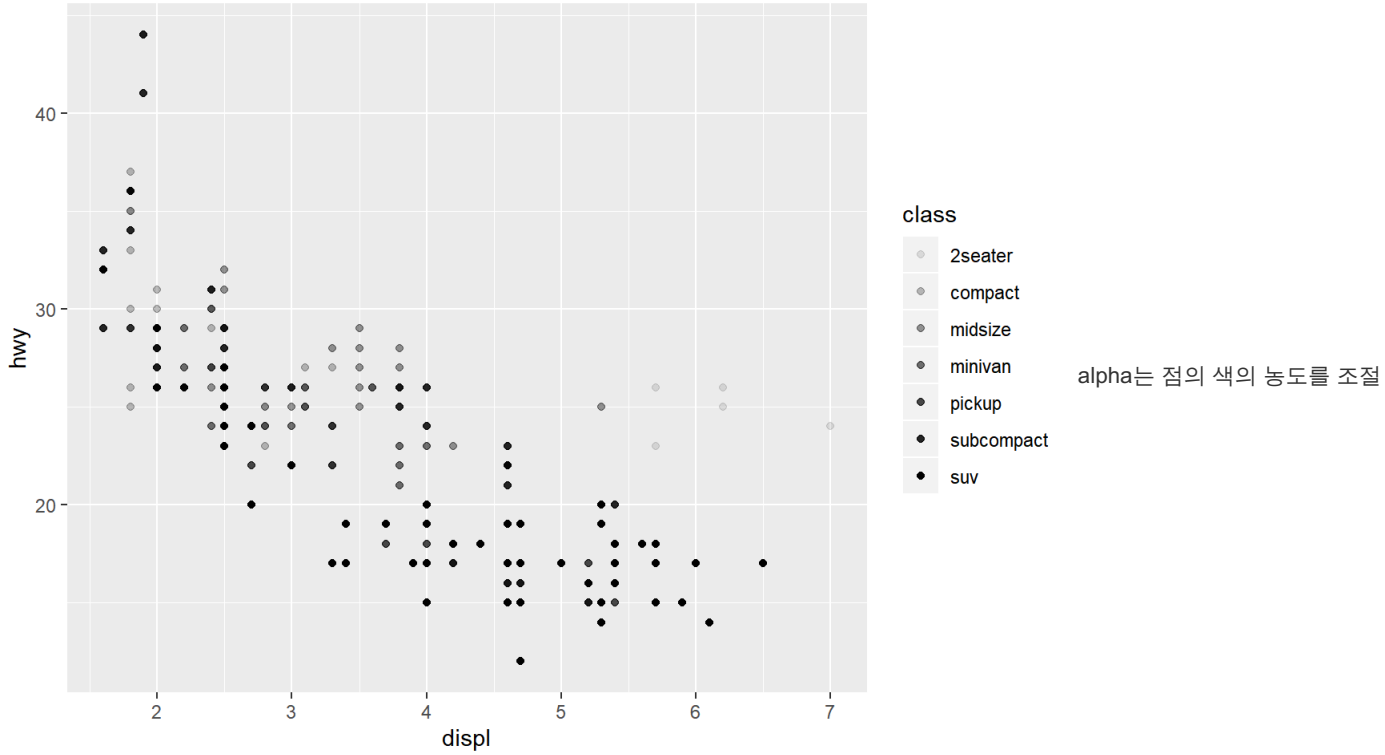
```
mpg %>% ggplot(aes(x=displ, y=hwy, size=class)) + geom_point() #Sizing variables
```



class에 따라 원의 크기로 구별을 해 보여즈는데 보기가 힘들다.

- alpha

```
mpg %>% ggplot(aes(x=displ, y=hwy, alpha=class)) + geom_point() #Controls the transparency of the points
```

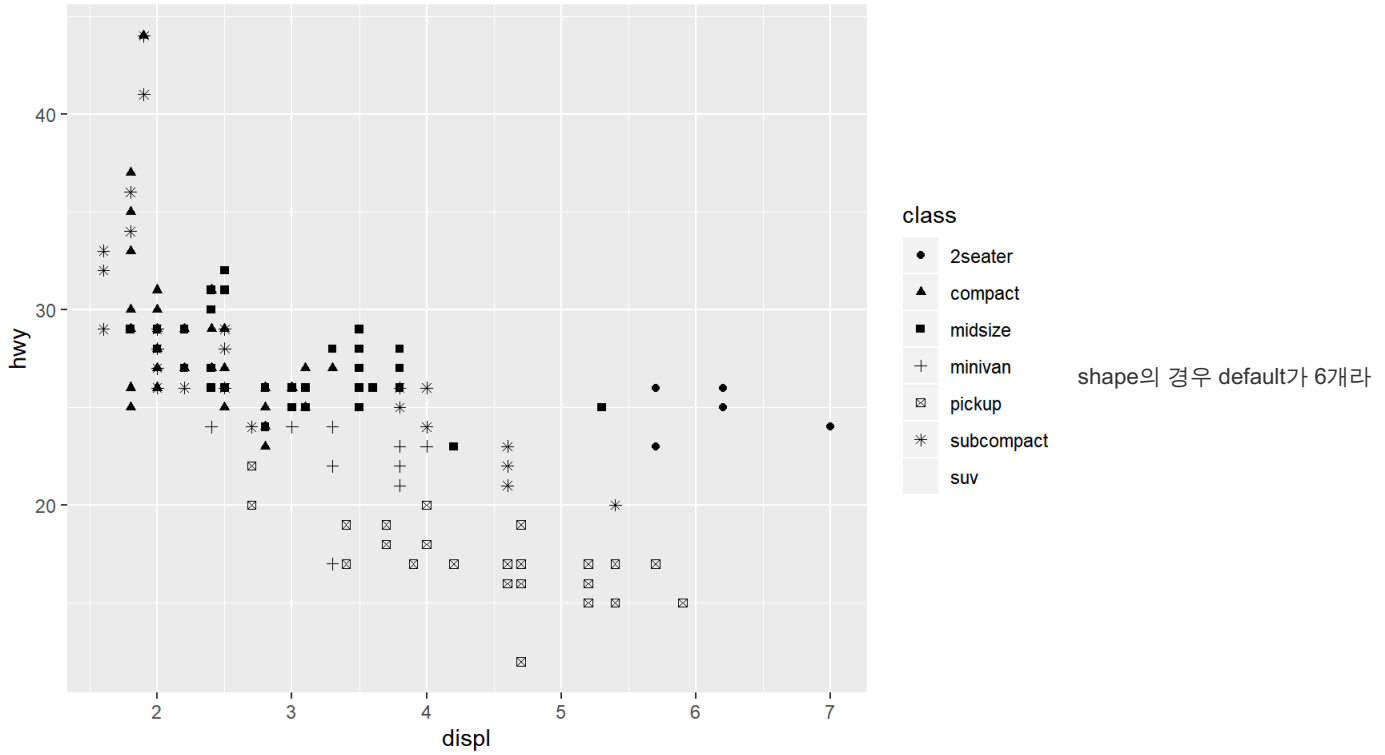


alpha는 점의 색의 농도를 조절

할 수 있어보인다. class를 알파값에 따라 분류해 놓은 그림을 볼 수 있다.

- shape

```
mpg %>% ggplot(aes(x=displ, y=hwy, shape=class)) + geom_point() #Controls the shapes of the points
```



shape의 경우 default가 6개라

서 suv가 삭제됐다. 따로 shaple fucnction을 사용해준다.

[Your turn 03]

1. What goes wrong with this code?

```
#no code
```

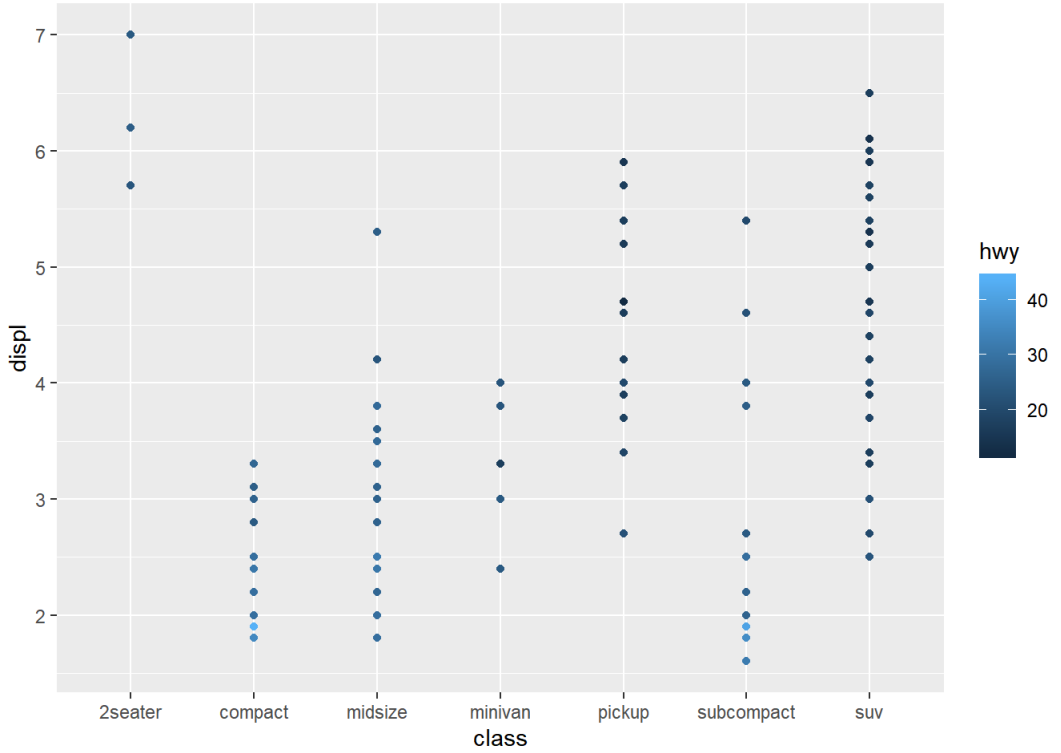
2 Which variables in mpg are categorical? Which variables are continuous?

```
mpg %>% glimpse()
```

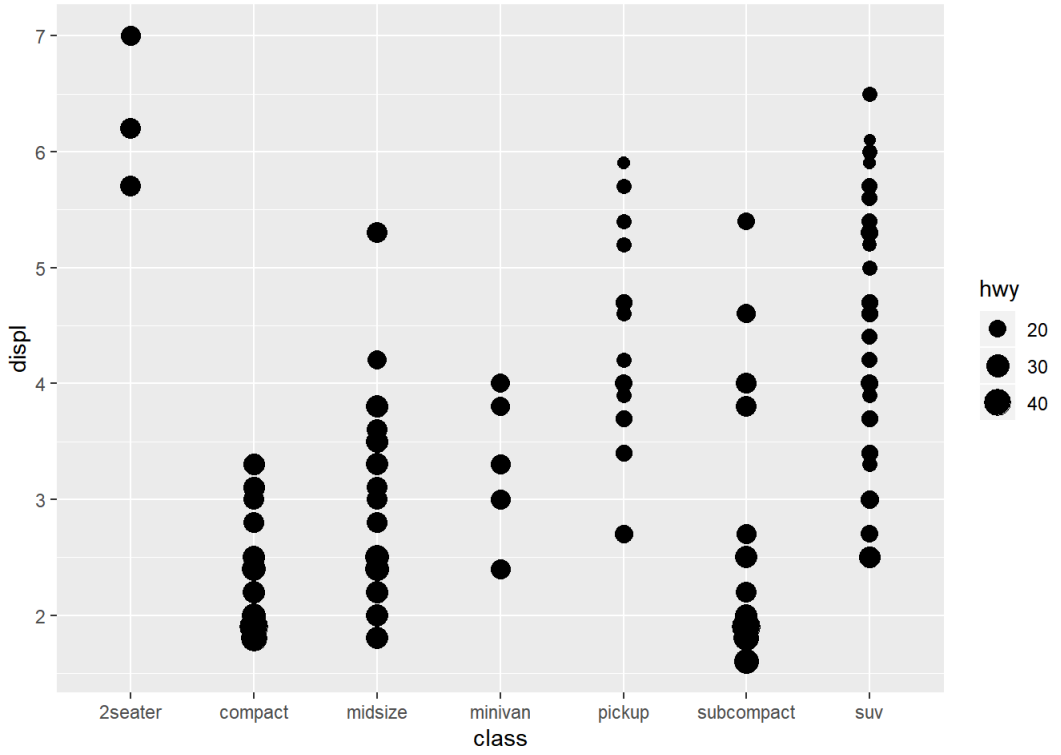
```
Observations: 234
Variables: 11
$ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi"...
$ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro"...
$ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0,...
$ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, ...
$ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, ...
$ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "a...
$ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4",...
$ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17...
$ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25...
$ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
$ class        <chr> "compact", "compact", "compact", "compact", "compact",...
```

3. Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

```
mpg %>% ggplot(aes(x=class, y=displ, color = hwy)) + geom_point()
```




```
mpg %>% ggplot(aes(x=class, y=displ, size = hwy)) + geom_point()
```

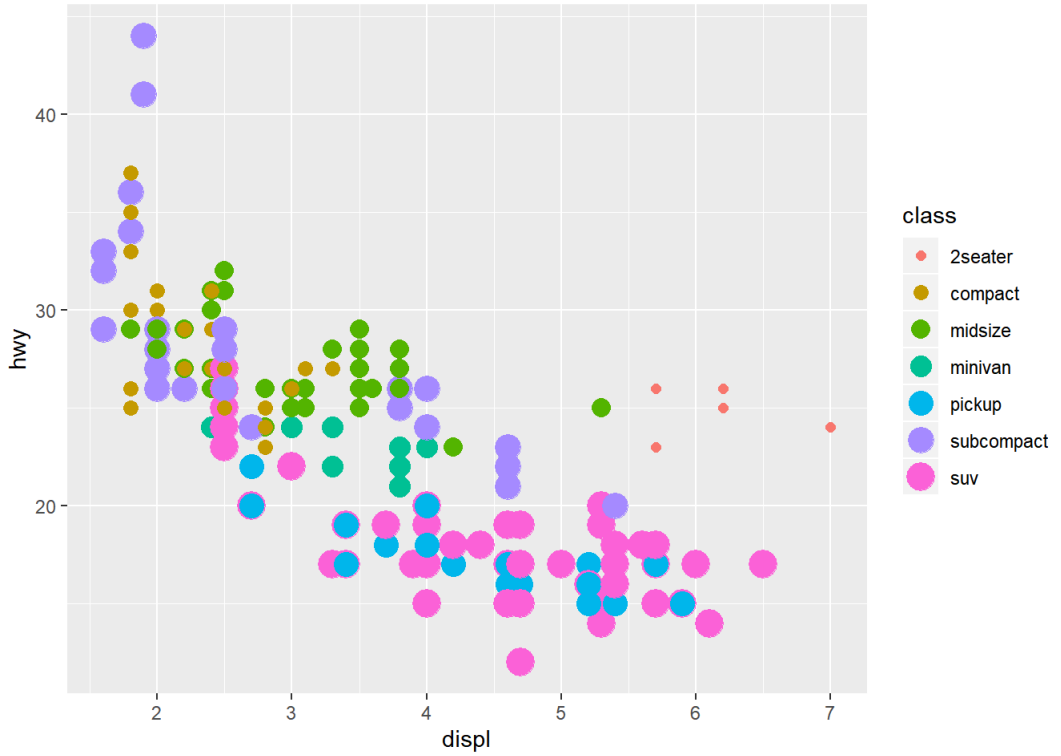



```
#mpg %>% ggplot(aes(x=class, y=displ, shape = hwy)) + geom_point()
```

shape의 경우 continuous variable can not be mapped to shape

4. What happens if you map the same variable to multiple aesthetics?

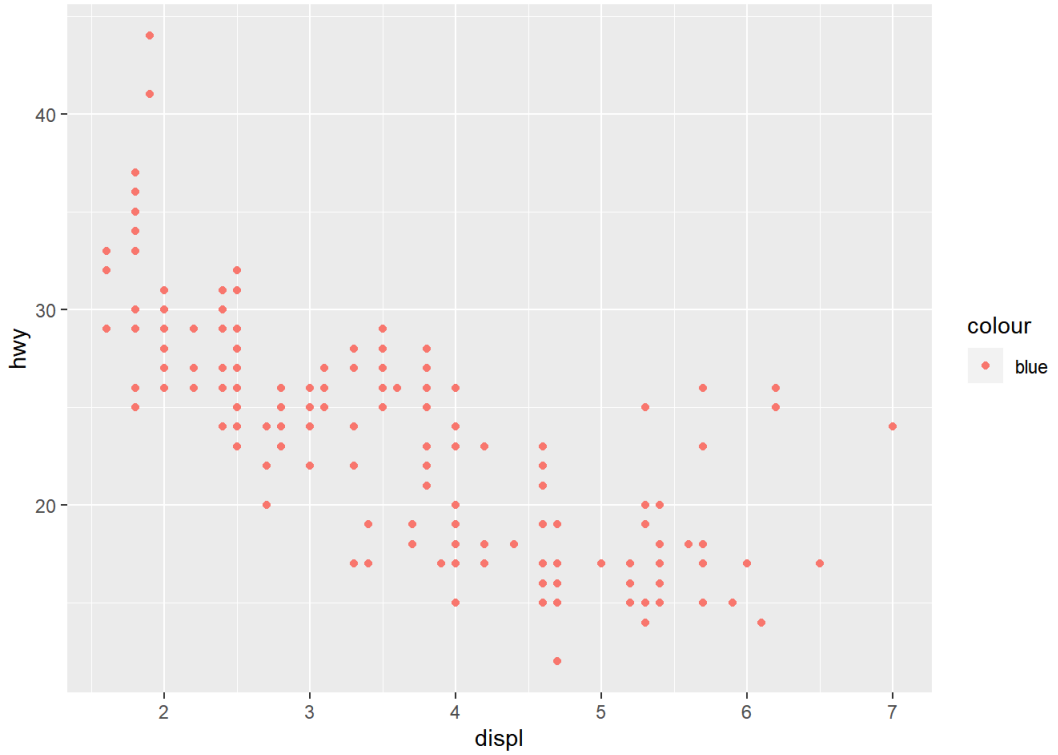
```
mpg %>% ggplot(aes(x=displ, y=hwy, color = class, size=class)) + geom_point()
```



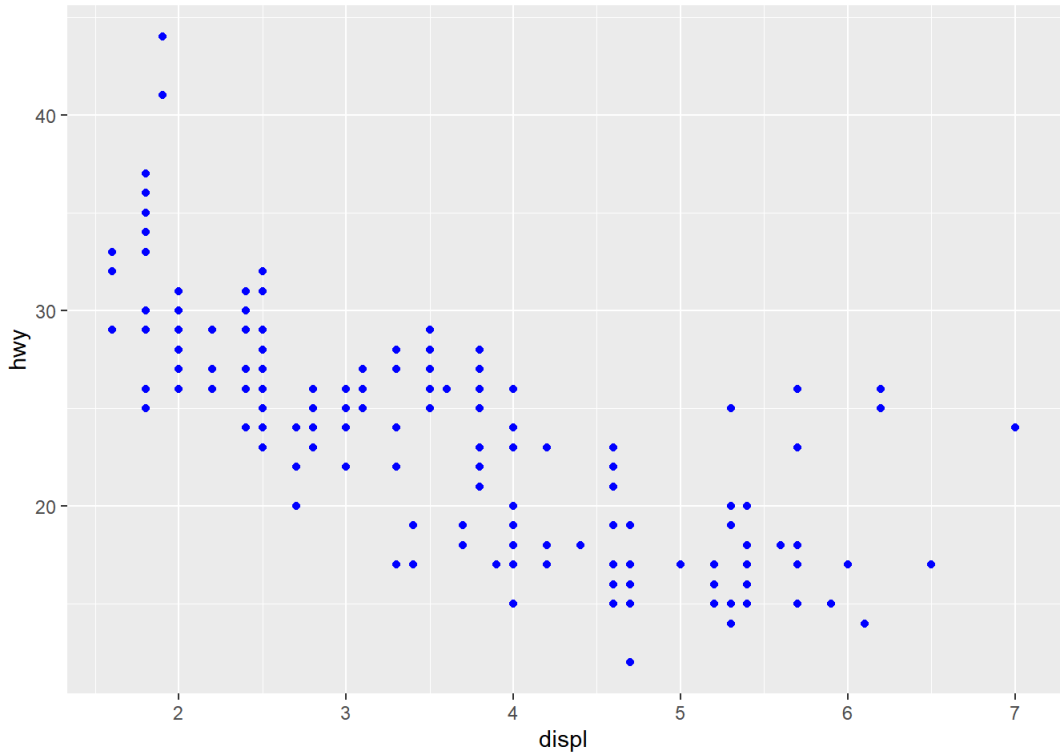
둘이 동시에 적용이 된다.

5.What does the stroke aesthetic do? What shapes does it work with?

```
mpg %>% ggplot(aes(x=displ, y=hwy, color='blue')) + geom_point()
```




```
mpg %>% ggplot(aes(x=displ, y=hwy)) + geom_point( color='blue')
```

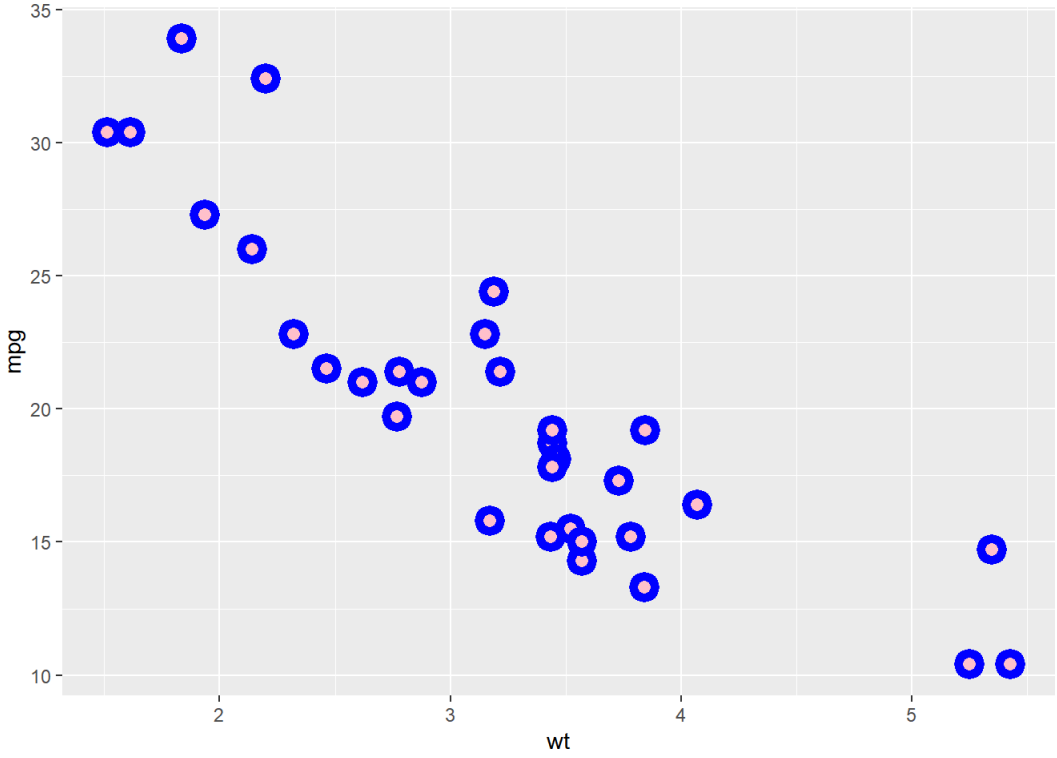


점의 겉면의 두께를 조절해준

다. 주의할 점은 aesthetic요소를 안에 적으면 하나의 변수라고 인식한다. 하지만 밖에 벅으면 color에 대한 색이라고 인지한다.

- stoke : 점의 겉면의 두께

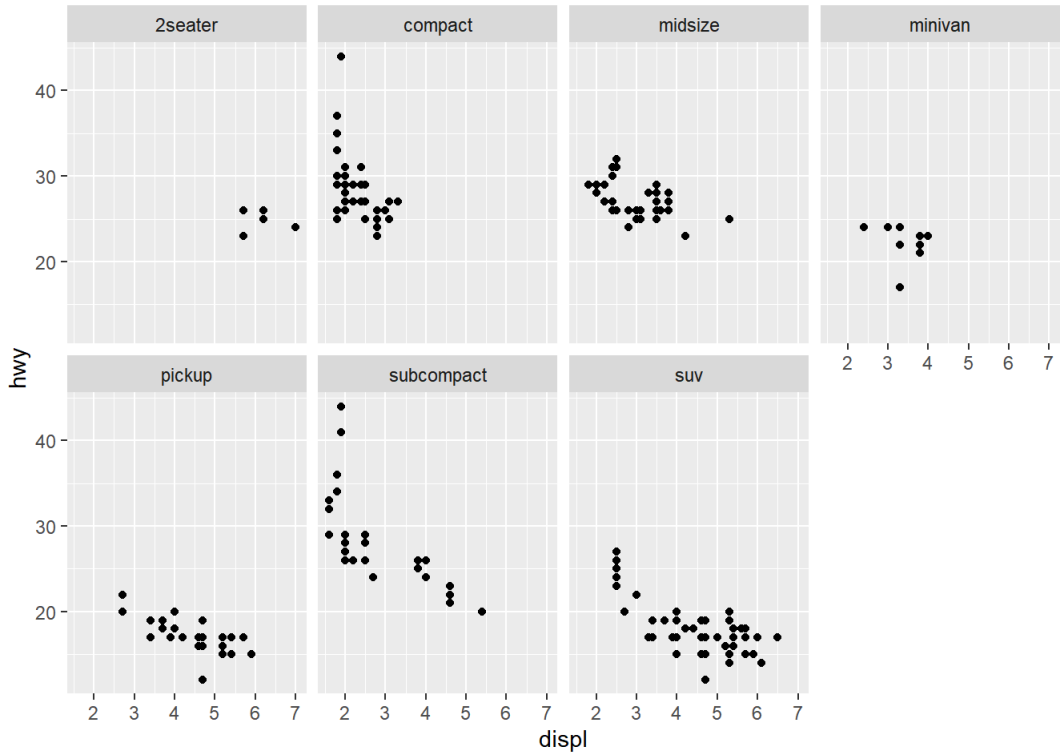
```
mtcars %>% ggplot(aes(wt, mpg))+geom_point(shape=21, color='blue',size=3,stroke=3,fill='pink')
```


35
30
25
20
15
10
mpg
2
3
4
5
wt

4. facet_wrap and facet_grid

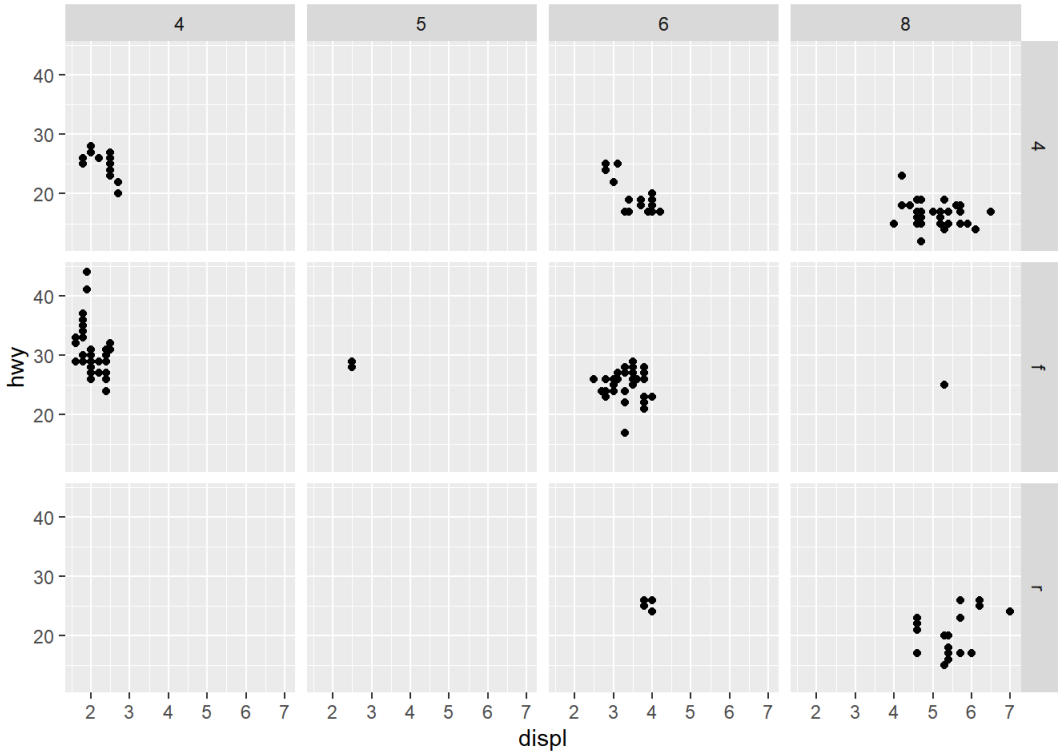
4-1. single variable

```
mpg %>% ggplot(aes(x=displ, y=hwy)) + geom_point()+
  facet_wrap(~class,nrow=2)
```



4-1. two varibale

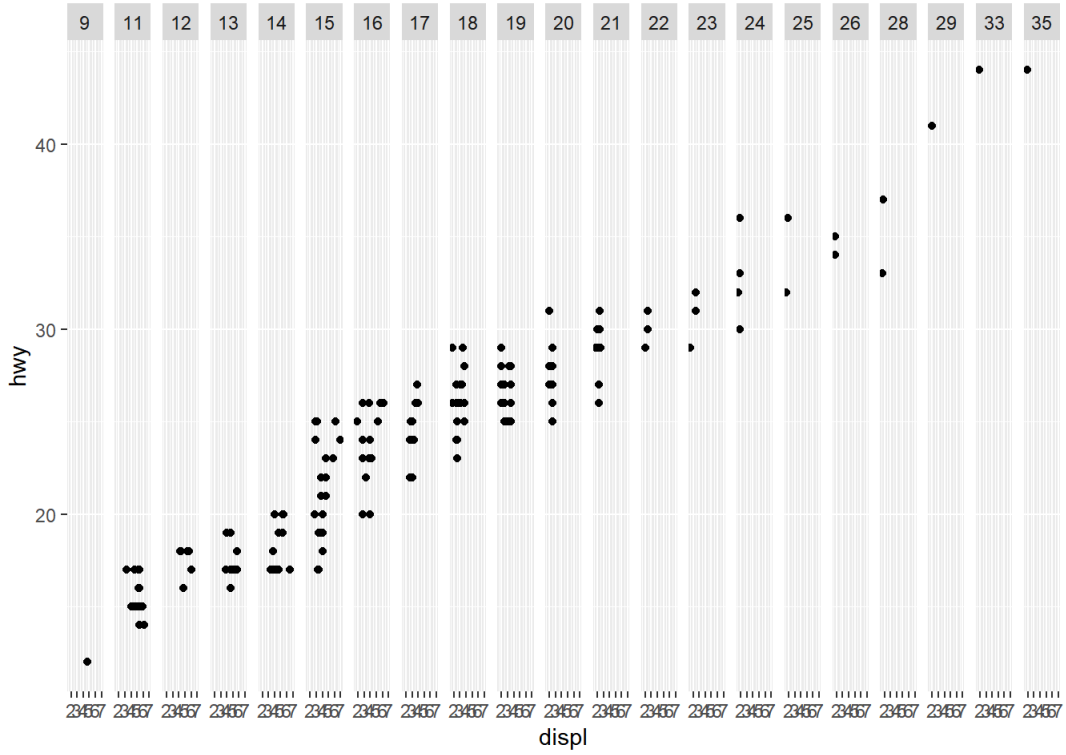
```
mpg %>% ggplot(mapping = aes(x=displ, y=hwy)) + geom_point() +
  facet_grid(drv ~ cyl)
```


4
5
6
8
4
f
r
hwy
displ
40
30
20
2
3
4
5
6
7

[Your turn 04]

1. What happens if you facet on a continuous variable?

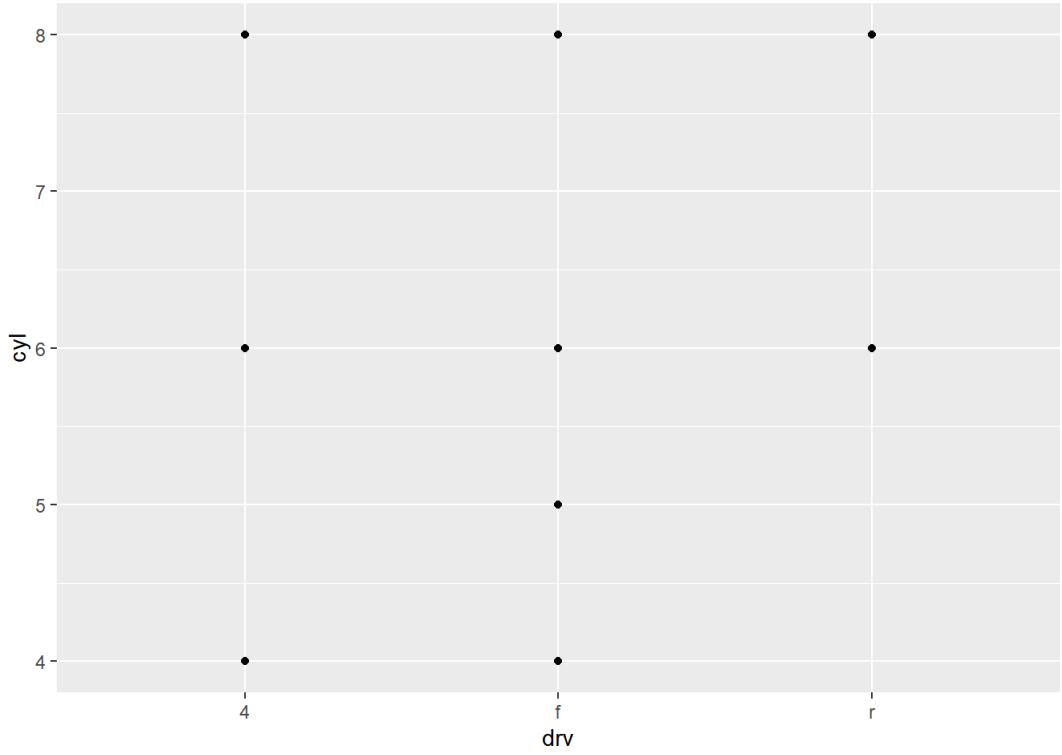
```
mpg %>% ggplot(aes(x=displ, y=hwy)) + geom_point()+
  facet_wrap(~cty,nrow=1)
```



아무 의미가 없다.

2. What do the empty cells in plot with facet_grid(drv ~ cyl) mean? How do they relate to this plot?

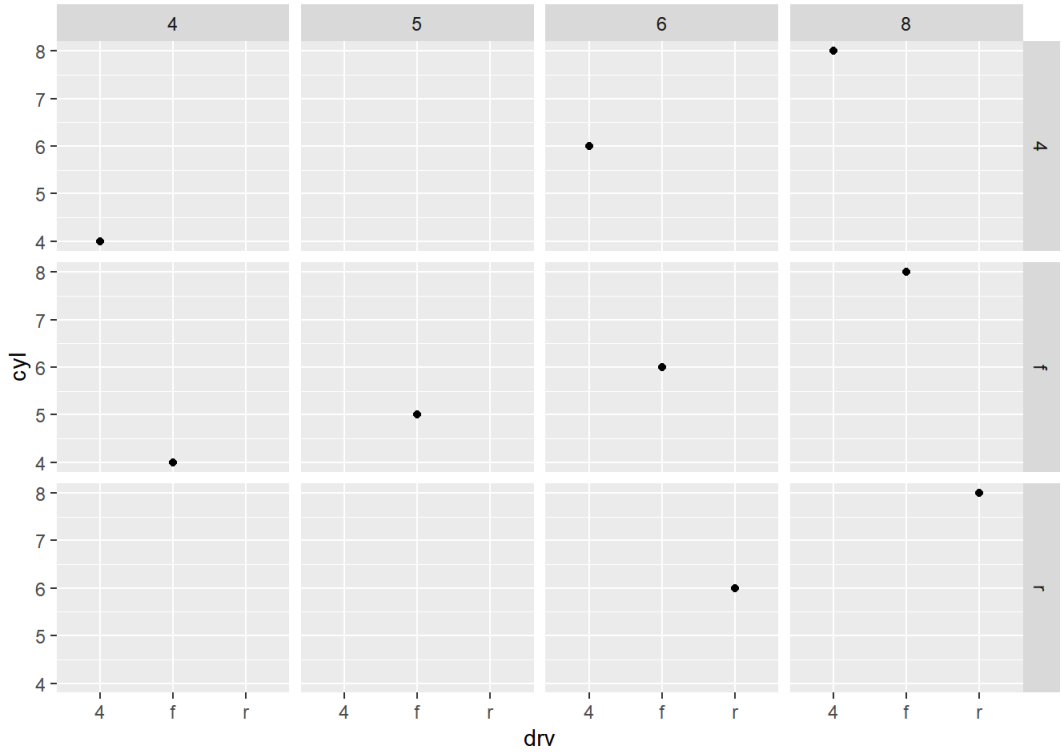
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl))
```



그 부분에 포인트가 없기때문

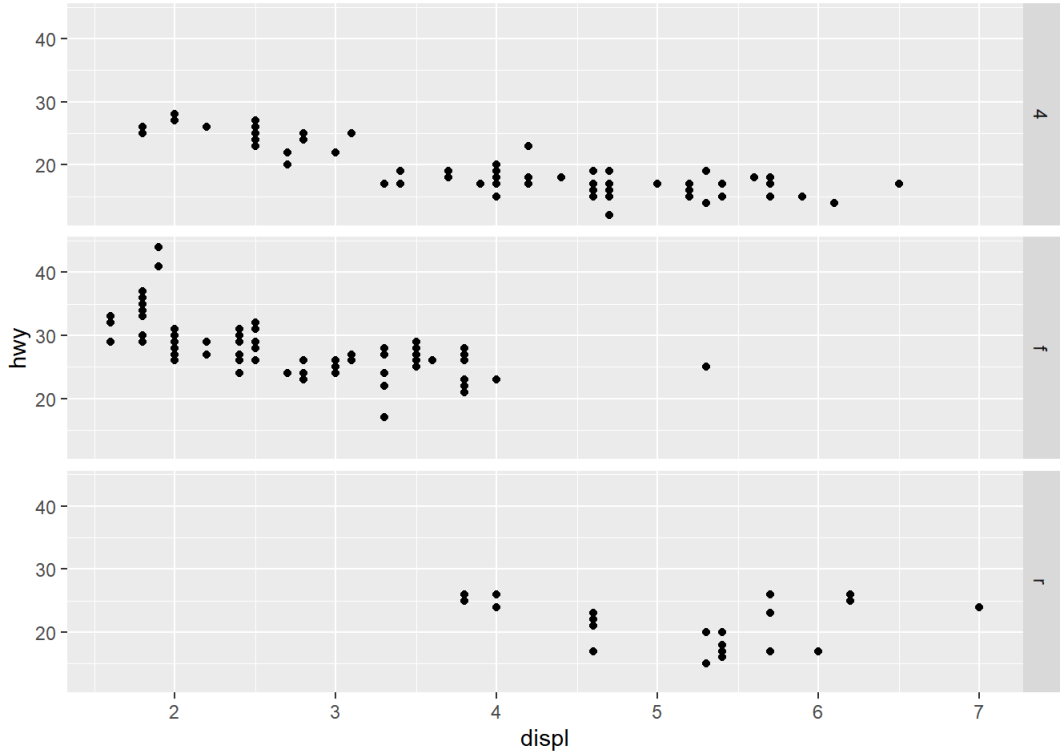
에 empty cell로 나타난다.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl)) +
  facet_grid(drv ~ cyl)
```



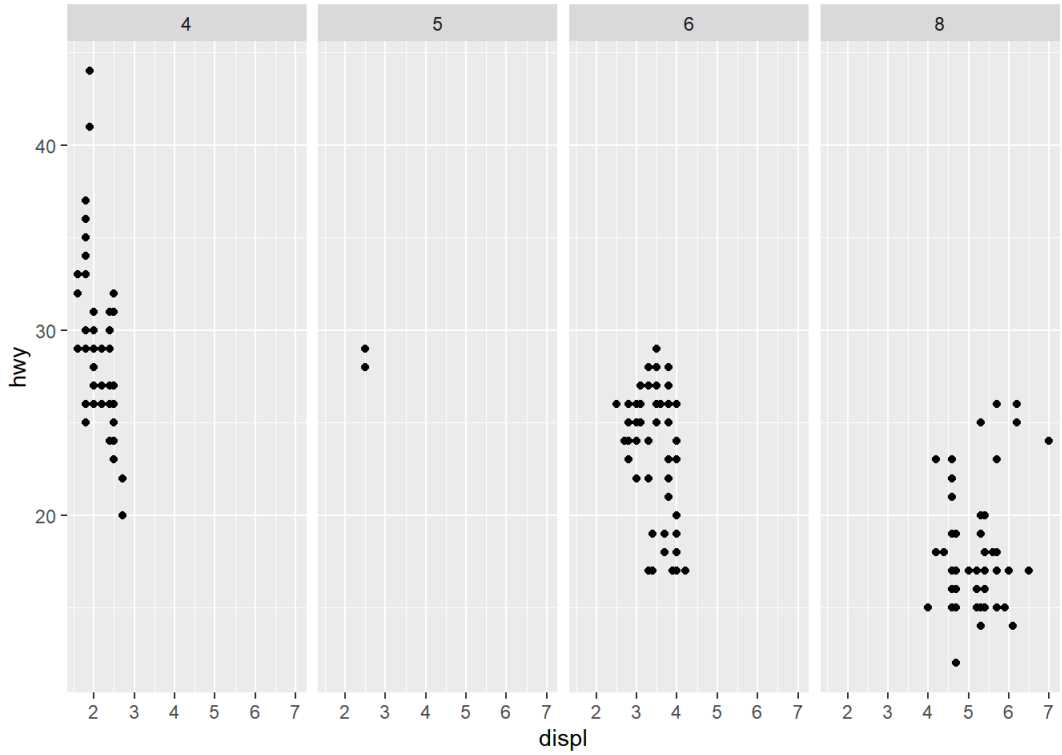
3. What plots does the following code make? What does '.' do?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)
```



drv가 행에 대해서 나눠진다.(x축에 대해 나눠라.)

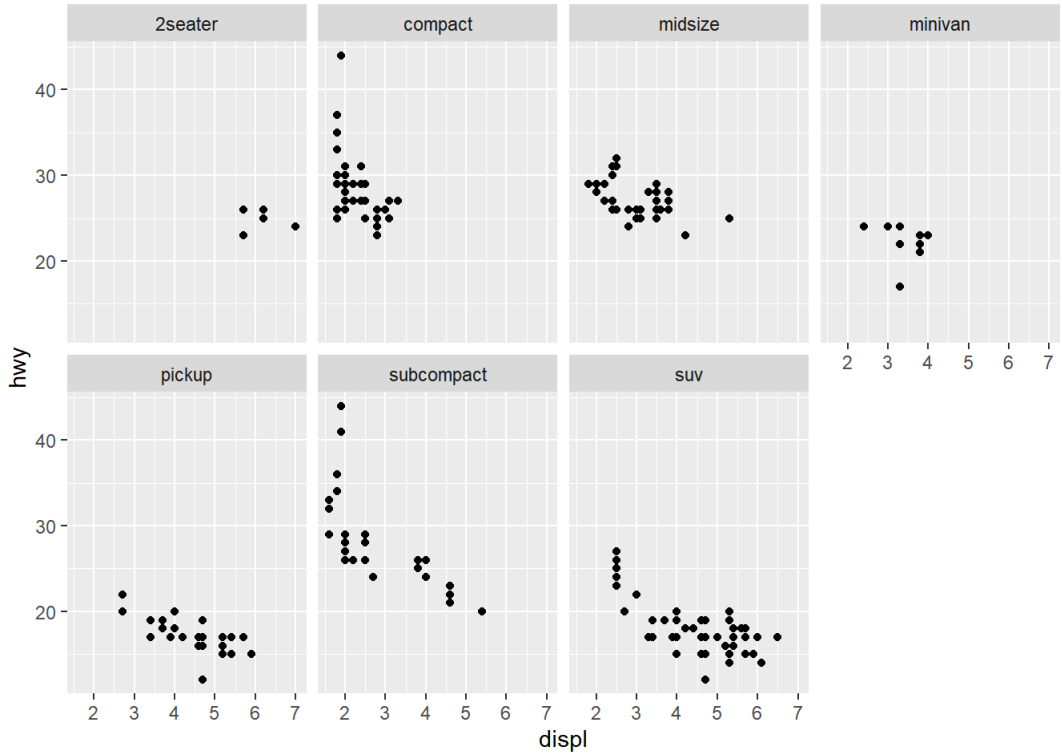
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)
```



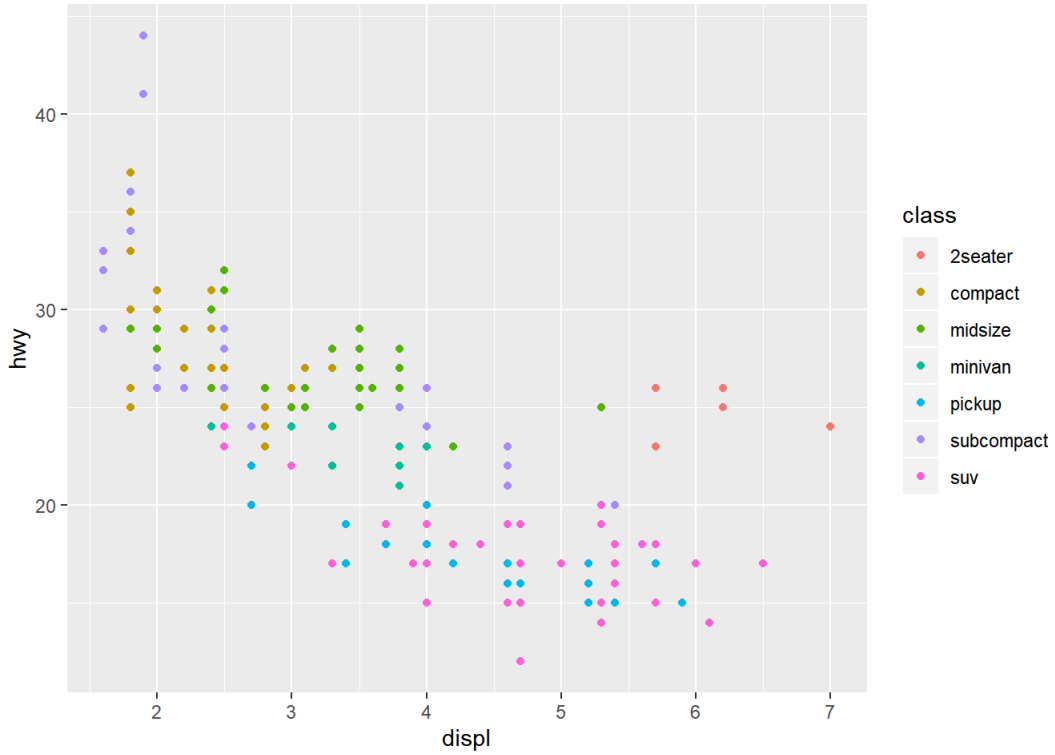
cyl가 열에 대해서 나눠진다.(y축에 대해 나눠라.)

4. Take the first faceted plot in this section:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



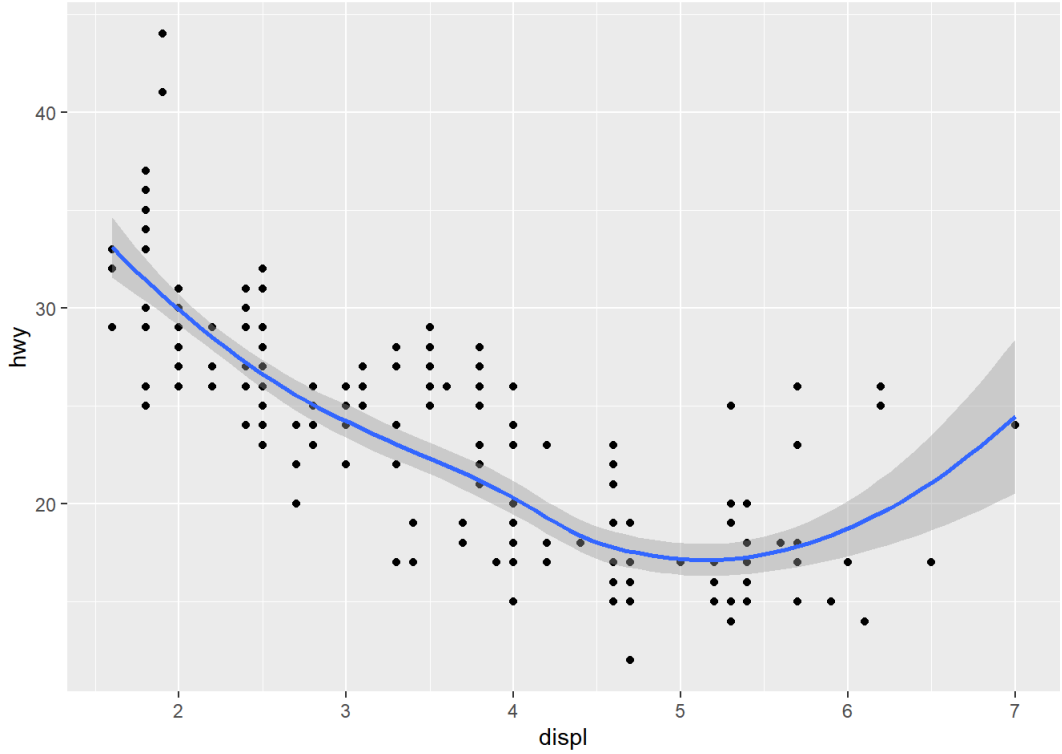

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=class))
```


hwy
40
30
20
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv

5. geom

- multi graph

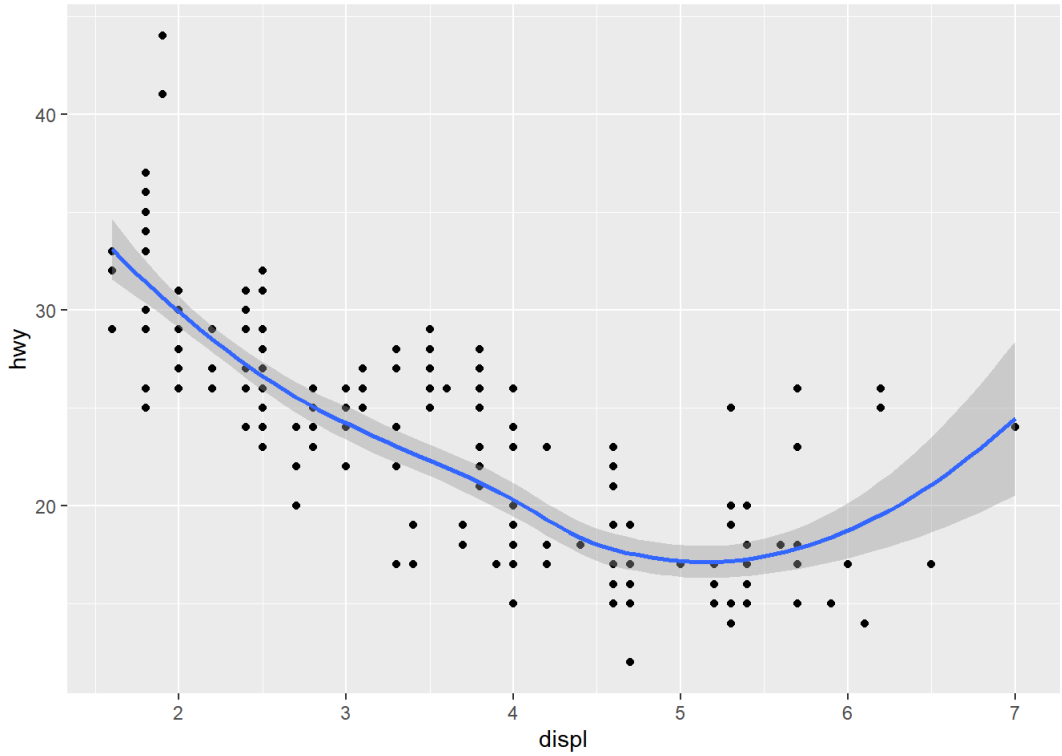
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



동시에 smooth라인과 산점도

를 함께 그려줄 수도 있다.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```

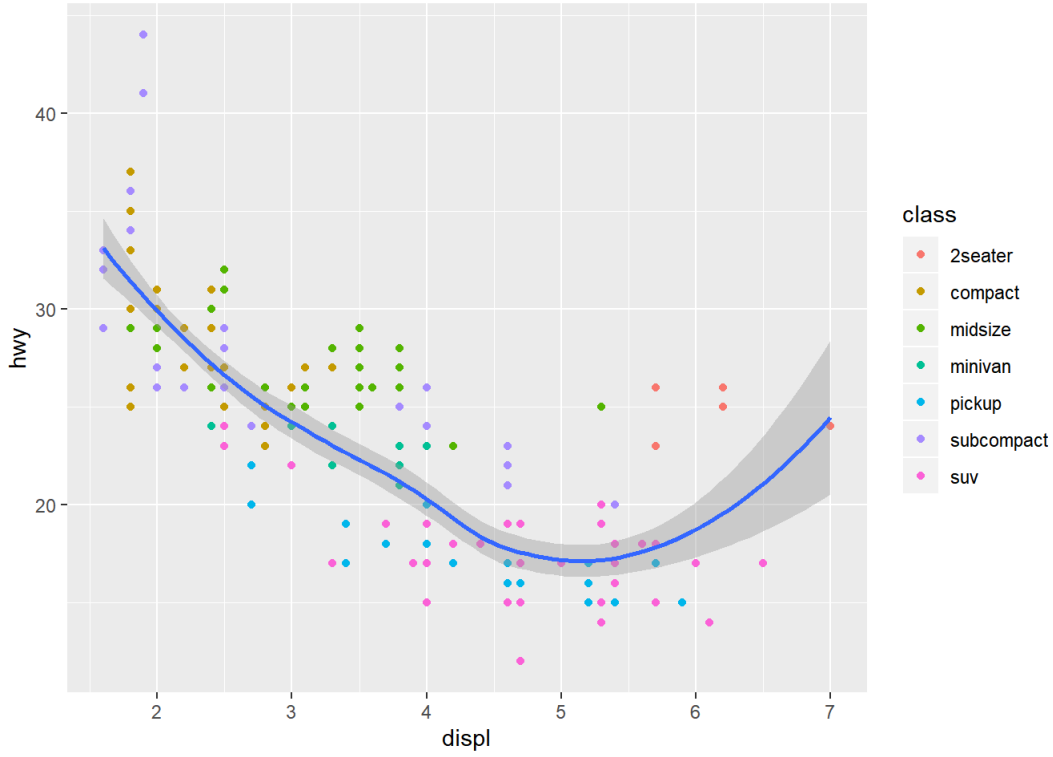


각 geom에 기반이 되는 정보

를 적어도 되지만 안에 요소가 같다면 global option을 의미하는 ggplot안에 한번만 적어도 된다. 위 그림은 이전 그림과 같다.

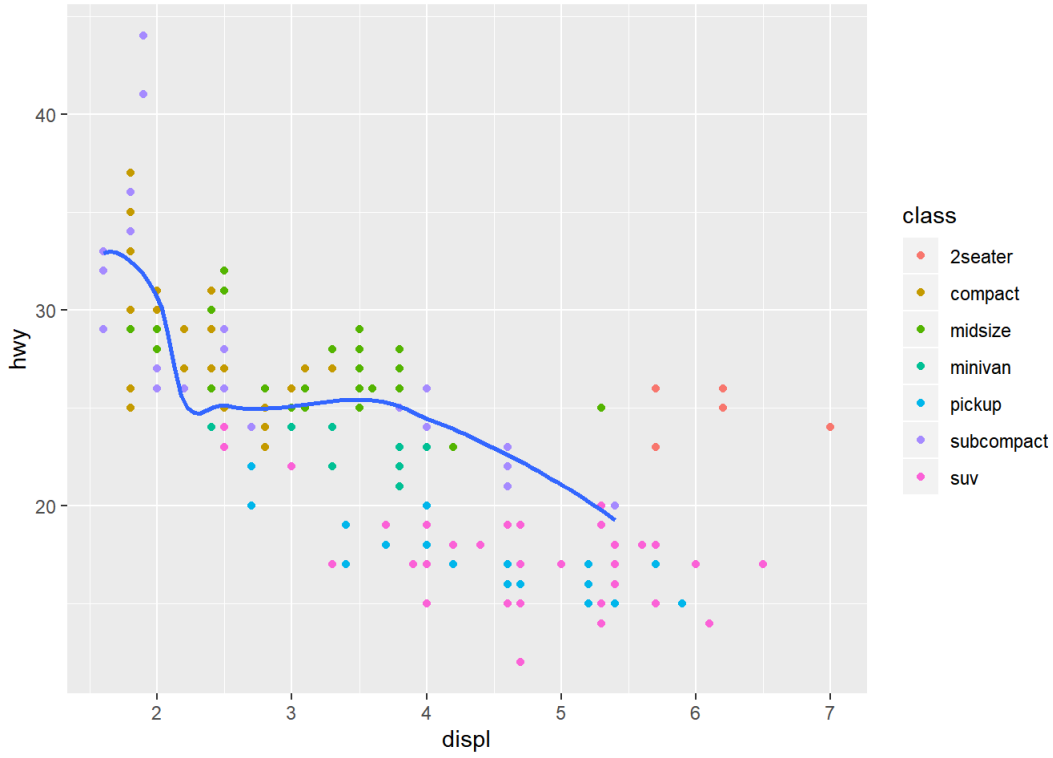
- class에 따라 color를 다르게 주고 smooth + scatter

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color=class)) +
  geom_smooth()
```



- class에 따라 color를 다르게 주고 smooth(class의 subset) + scatter

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth(data = filter(mpg, class == "subcompact"), se = F)
```



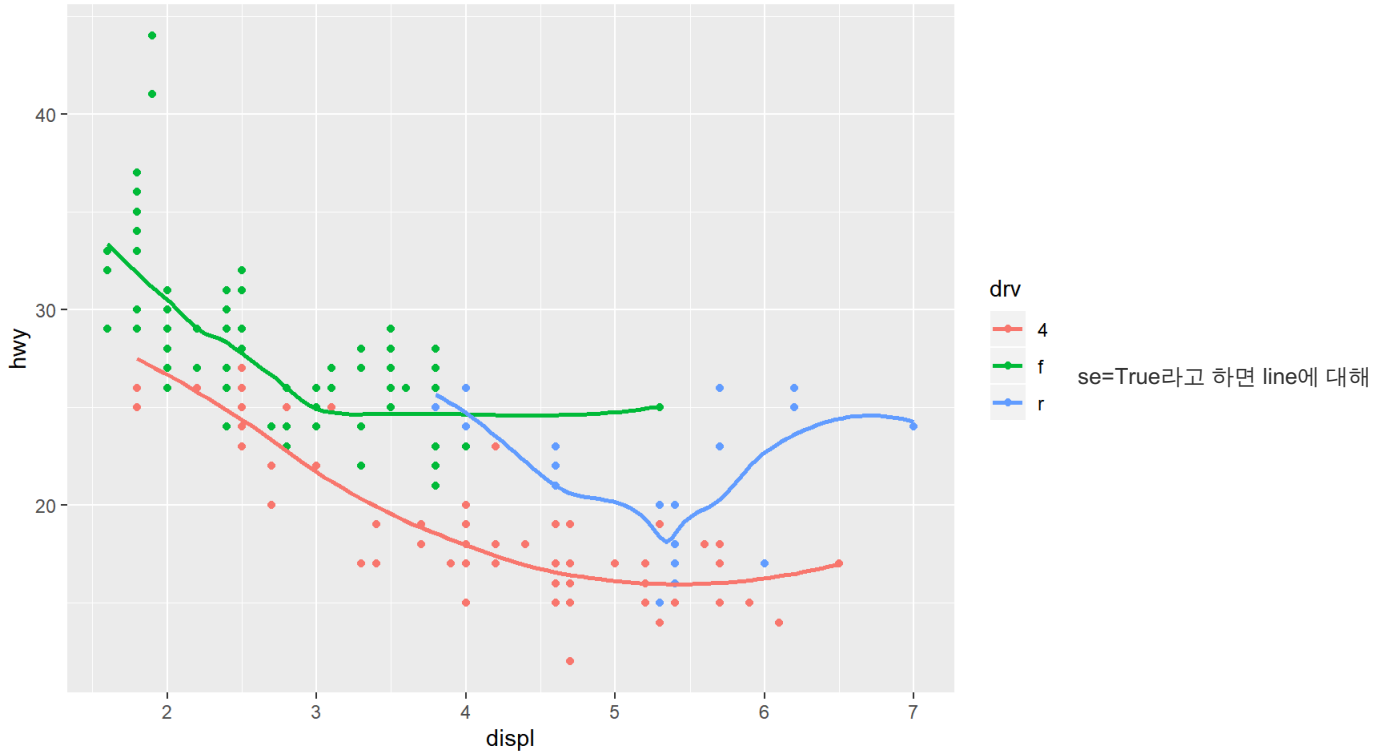
smooth를 class가

"subcompact"인 경우만 필터링하면 위와 같이 그릴 수 있다.

[Your turn 05]

2. Run this code in your head and predict what the output will look like. Then, run the code in R and check your predictions.

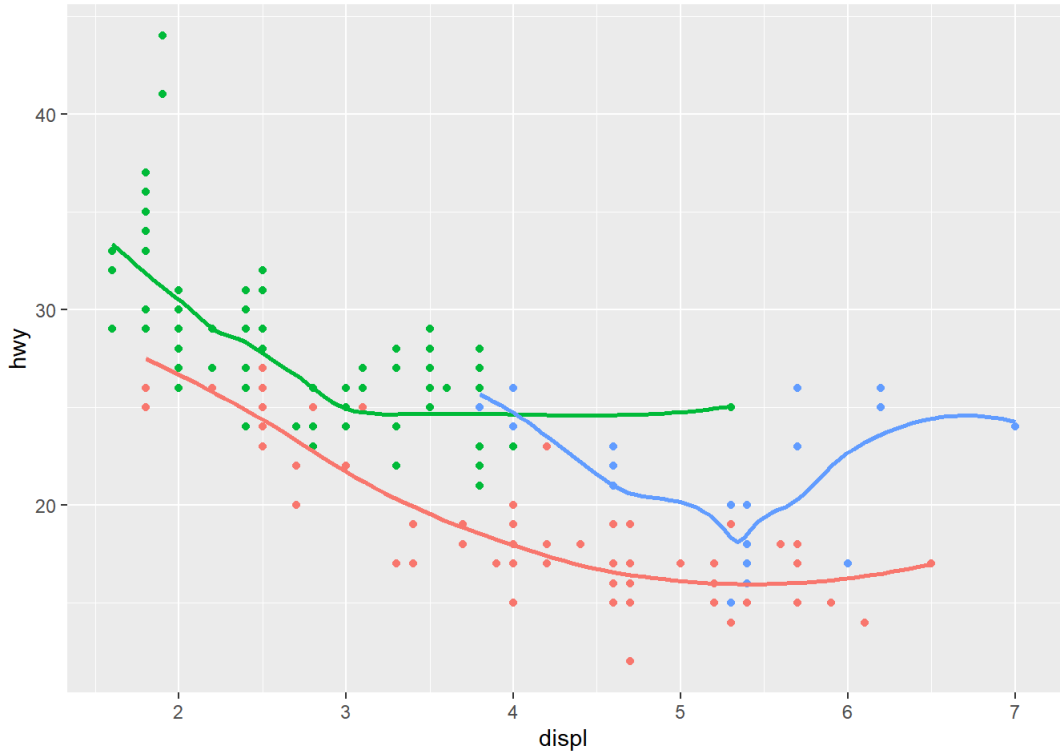
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy,
                                 colour = drv)) +
  geom_point() +
  geom_smooth(se = FALSE)
```



se=True라고 하면 line에 대해 표준편차만큼 띠가 생긴다. se = False라고 하면 그 띠가 생기지 않고 선만 보여준다.

3. What does show.legend = FALSE do? What happens if you remove it? Why do you think I used it earlier in the chapter?

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy,
                                 colour = drv)) +
  geom_point(show.legend = FALSE) +
  geom_smooth(se = FALSE,show.legend = FALSE)
```



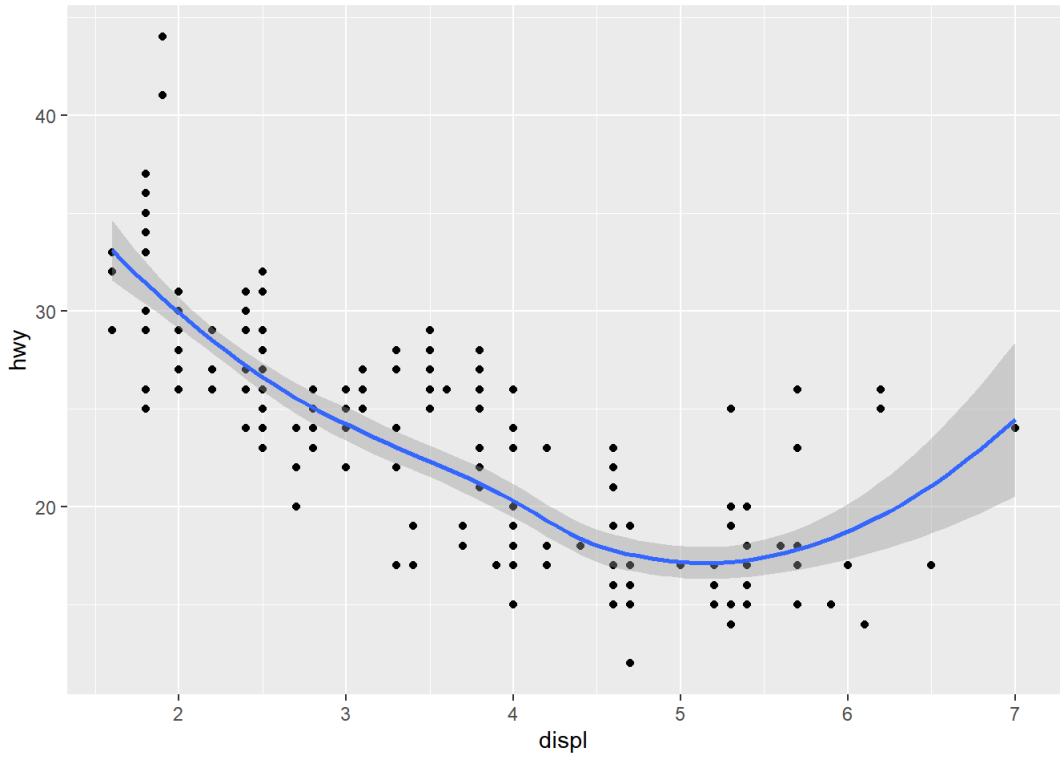
범례값이 보이지 않는다.

4. What does the se argument to geom_smooth() do?

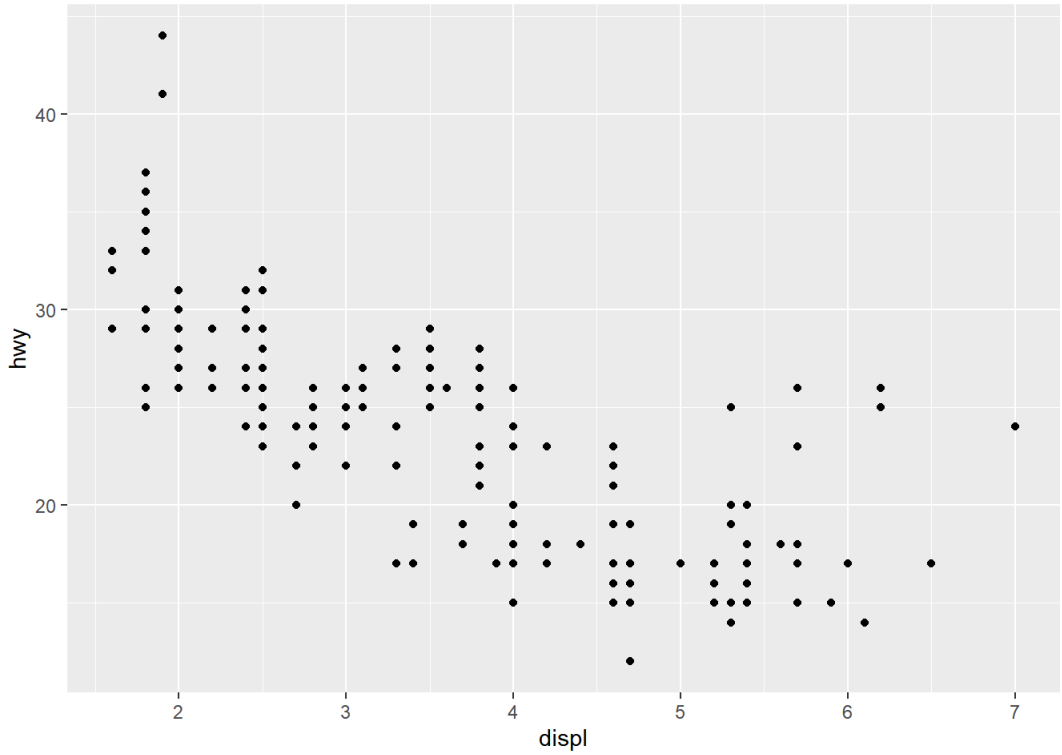
se=True라고 하면 line에 대해 표준편차만큼 띠가 생긴다.

5. Will these two graphs look different? Why/why not?

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```




```
ggplot() +
  geom_point(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth(se=F)
```



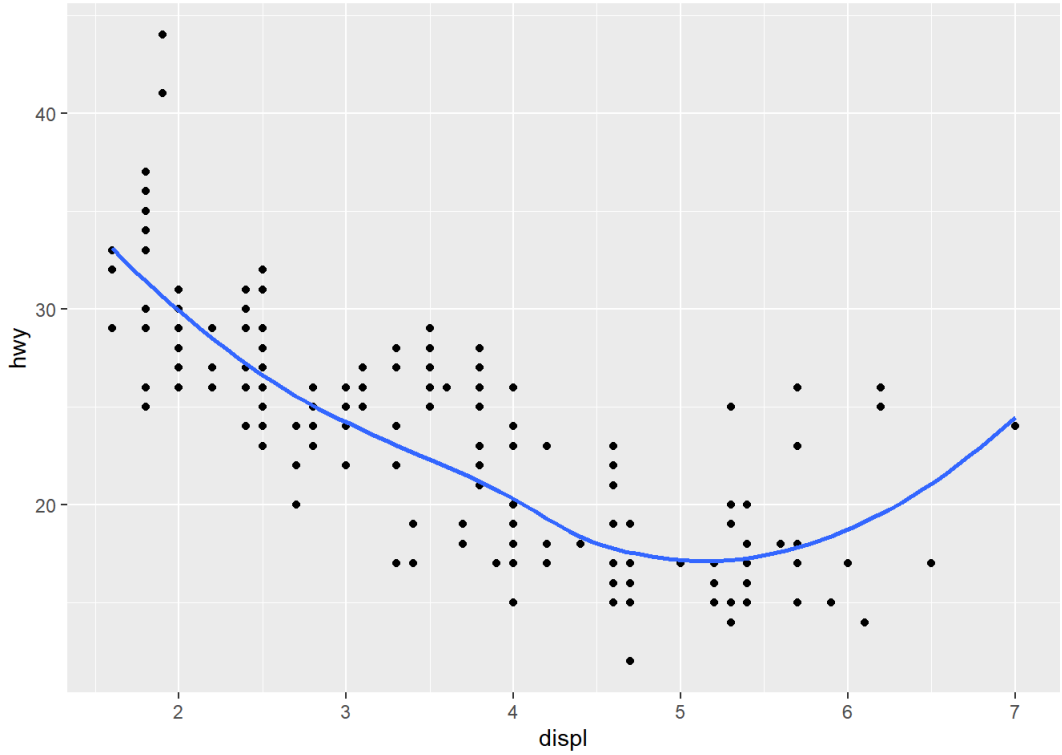
두번째 그래프는 geom_point

에만 x축과 y축의 정보가 입력돼있고 geom_smooth에 대해서는 아무런 정보가 없다. 따라서 line은 그려지지 않는다. smooth에도 동일하게 정보를 입력한다면 se=FALSE이기 때문에 표준편차에 대한 띠는 그려지지 않는다.

7. Use variables displ, hwy, drv and recreate following plots.

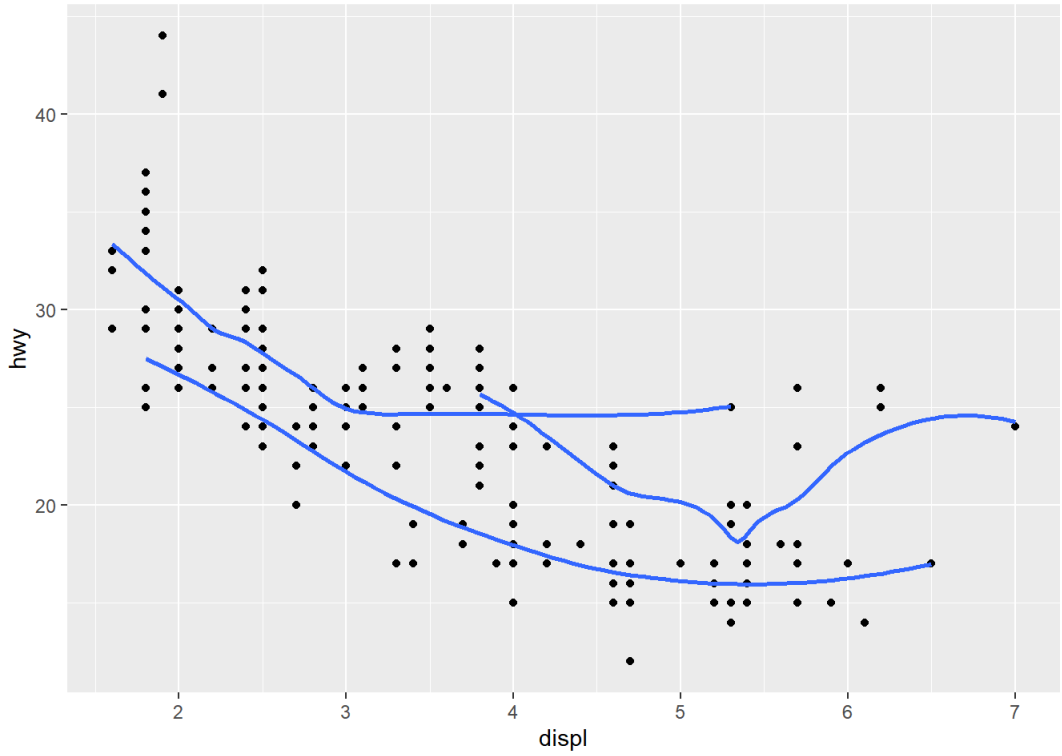
7-1.

```
mpg %>% ggplot(aes(x=displ, y=hwy))+
  geom_point()+
  geom_smooth(se=F)
```



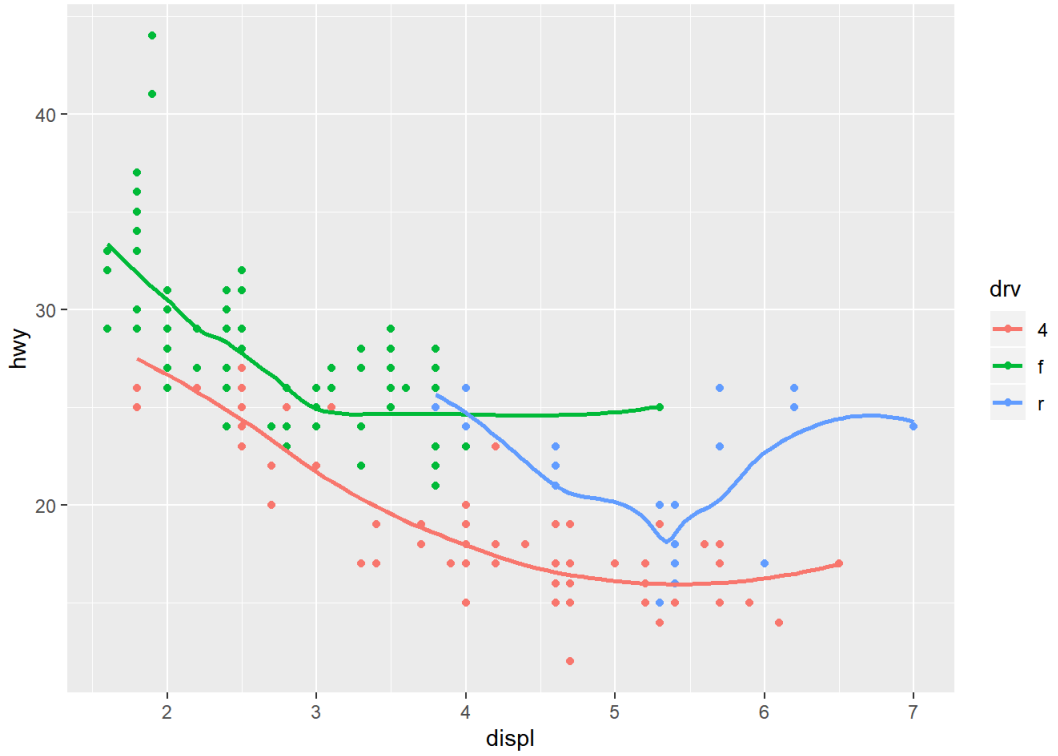
7-2.

```
mpg %>% ggplot(aes(x=displ, y=hwy))+
  geom_point()+
  geom_smooth(aes(group=drv),se=F)
```



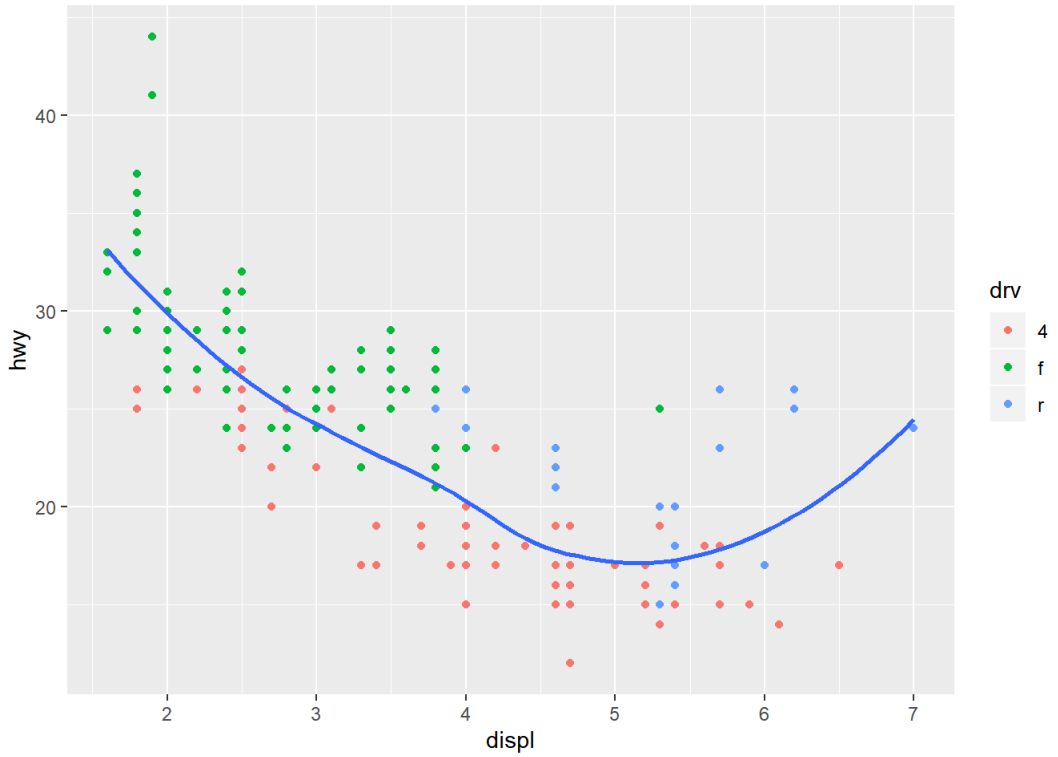
7-3.

```
mpg %>% ggplot(aes(x=displ, y=hwy))+
  geom_point(aes(color=drv))+
  geom_smooth(aes(color=drv),se=F)
```



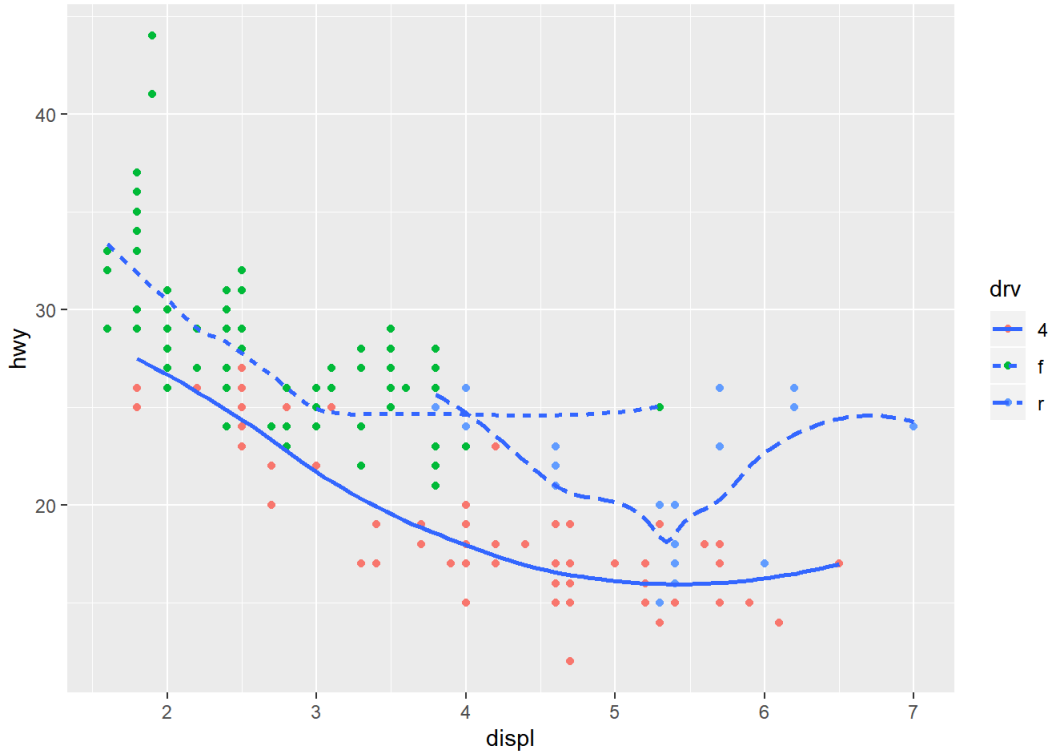
7-4.

```
mpg %>% ggplot(aes(x=displ, y=hwy))+
  geom_point(aes(color=drv))+
  geom_smooth(se=F)
```



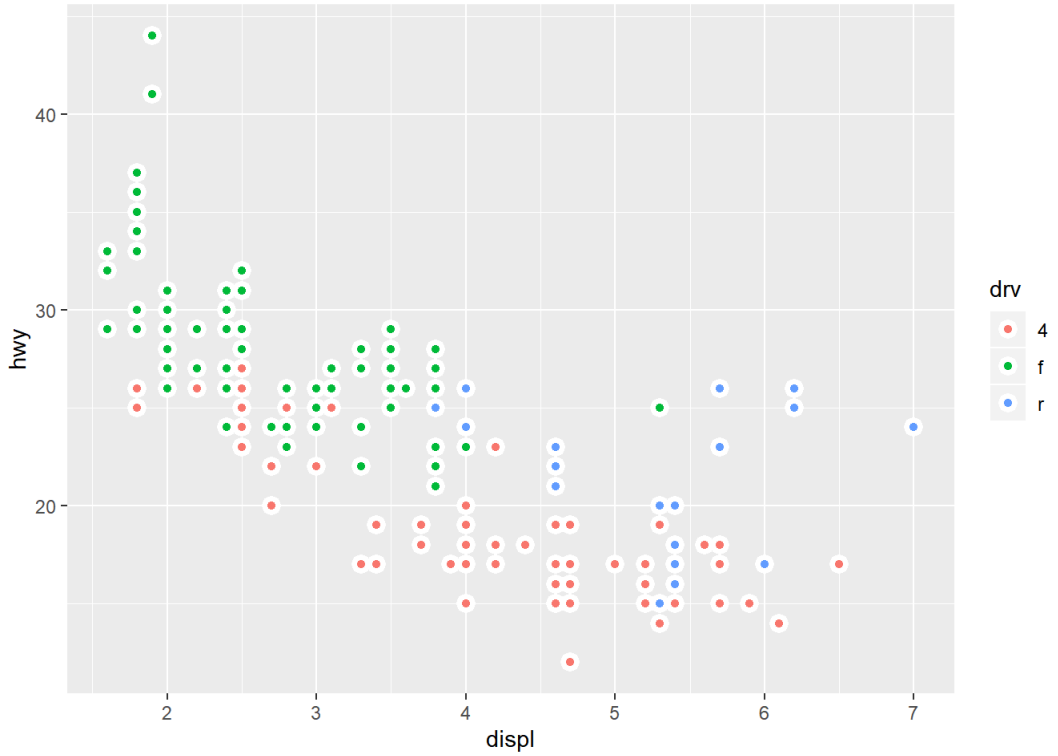
7-5.

```
mpg %>% ggplot(aes(x=displ, y=hwy))+
  geom_point(aes(color=drv))+
  geom_smooth(aes(linetype=drv),se=F)
```



7-6.

```
mpg %>% ggplot(aes(x=displ, y=hwy,fill=drv))+
  geom_point(shape=21,color='white',size=1.9,stroke=2)
```

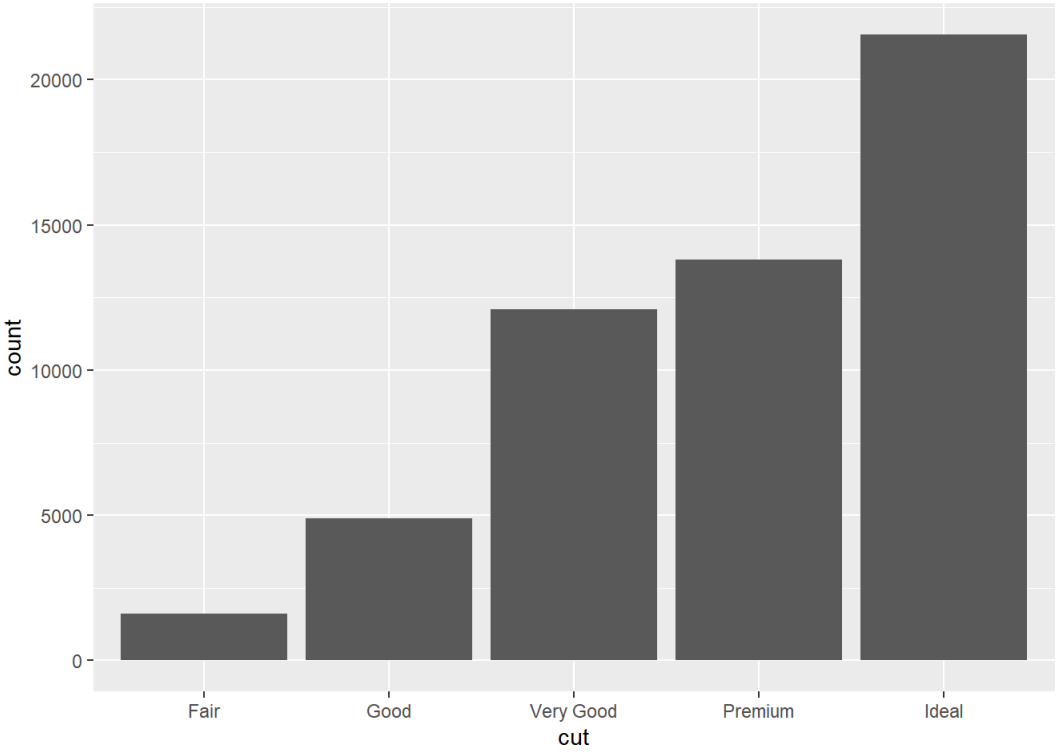


6. Statiscal transformation

[example]

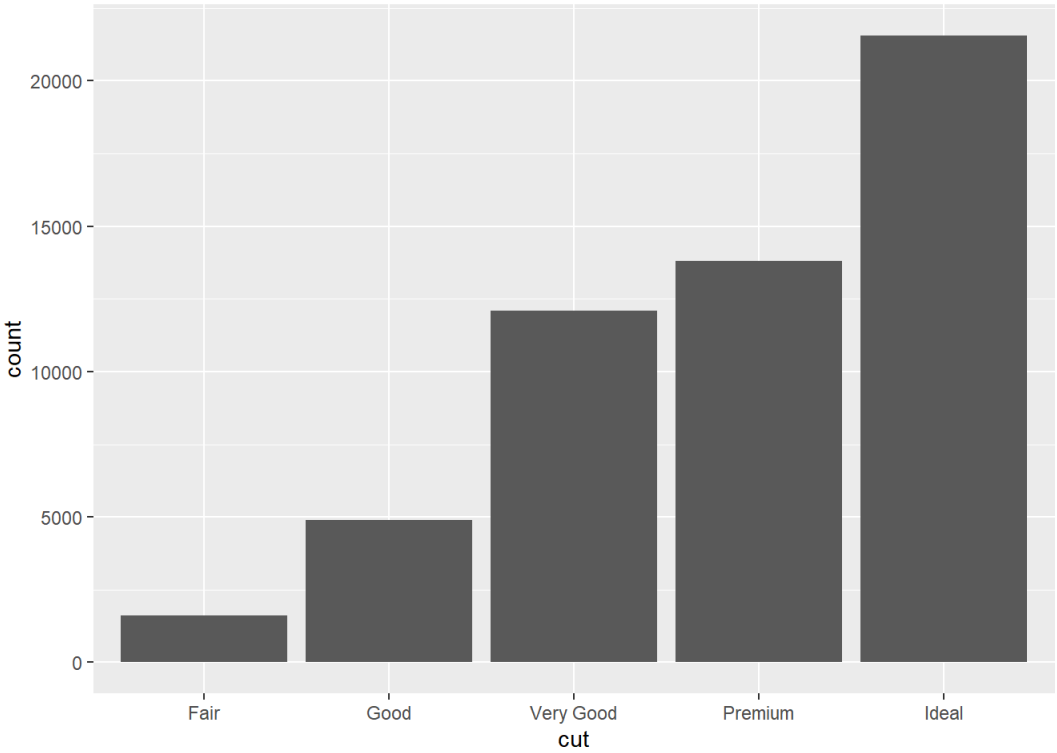
- barchart

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



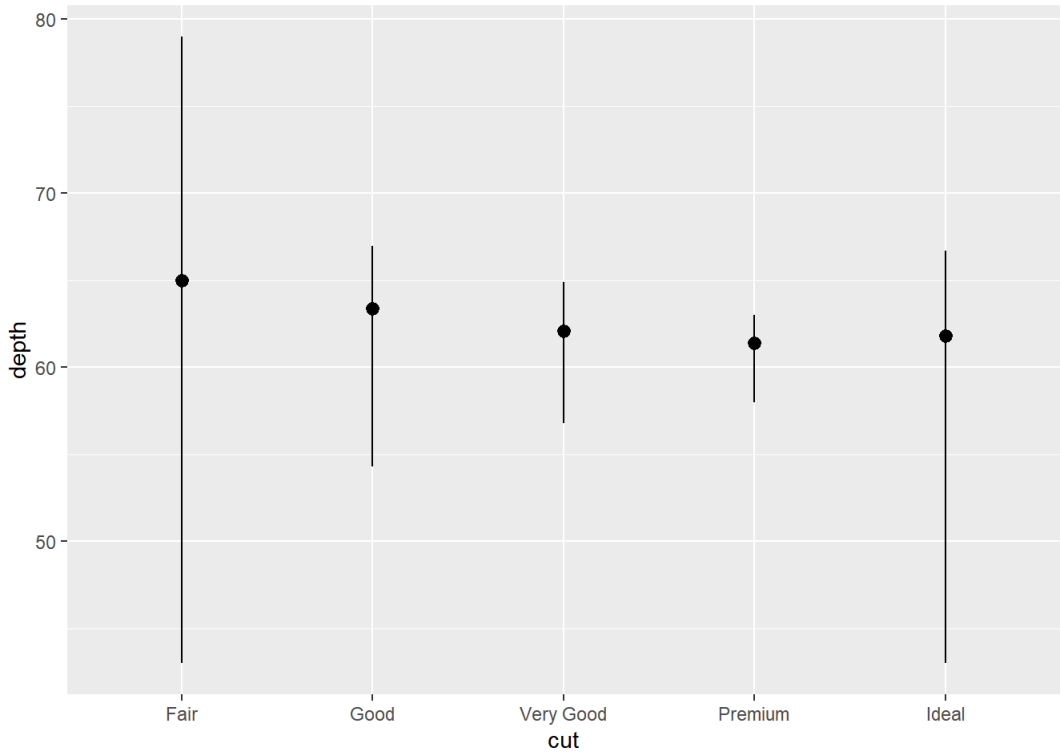
- stat_count

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



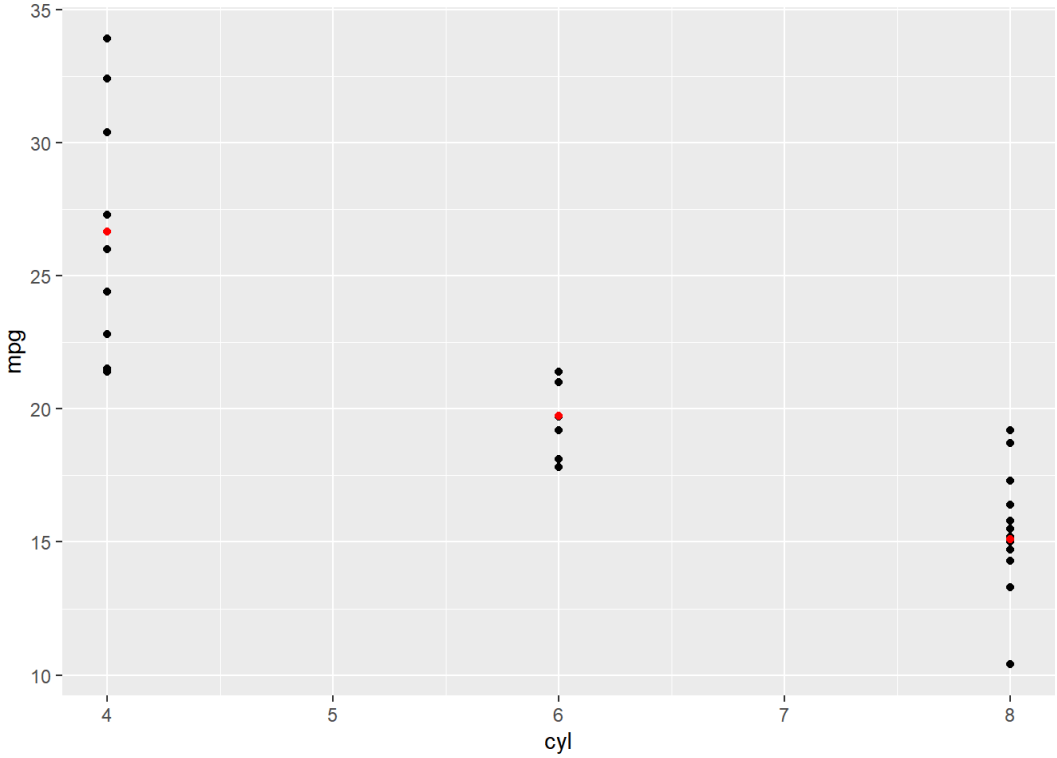
- stat_summary (1)

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



- stat_sumamry (2)

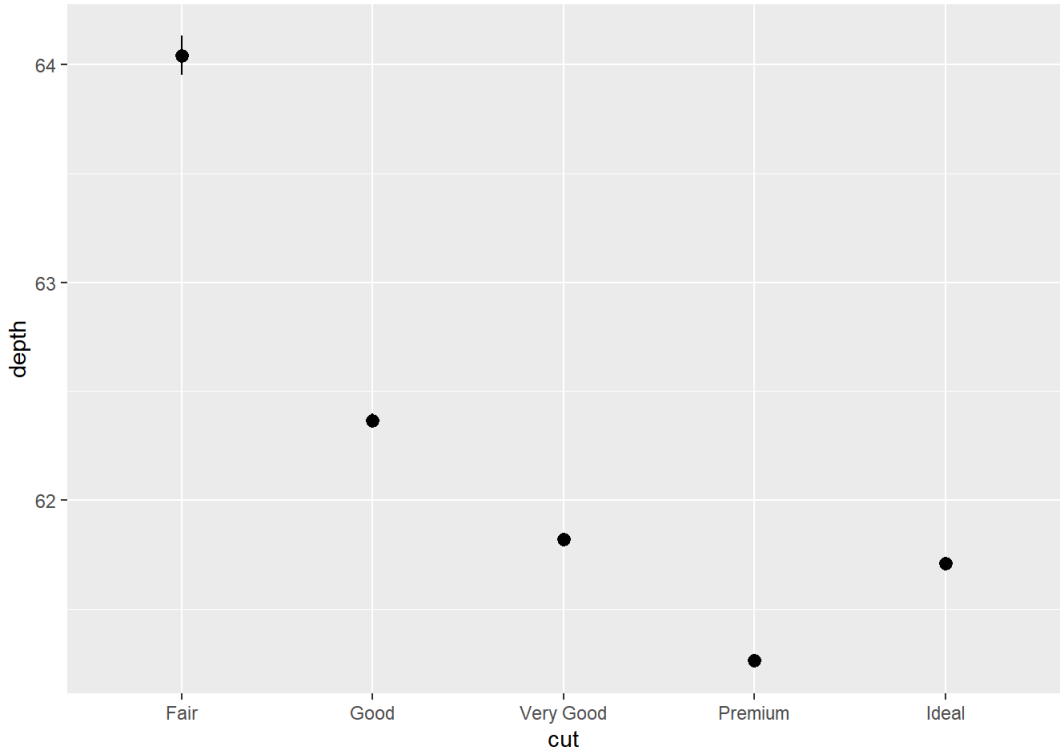
```
d <- ggplot(data=mtcars , aes(x=cyl, y=mpg))+geom_point()
d + stat_summary(
  fun.y = 'mean',
  colour='red',
  geom = 'point'
)
```


35
30
25
20
15
10
mpg
4
5
6
7
8
cyl

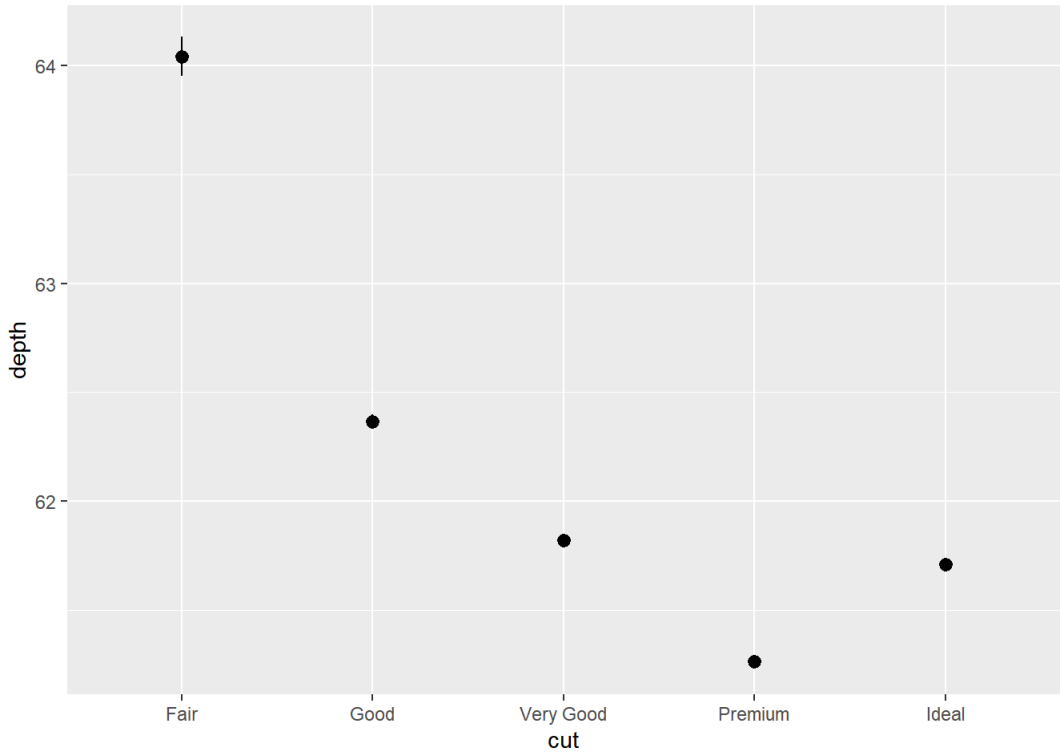
[Your turn 06]

1. What is the default geom associated with stat_summary()? How could you rewrite the previous plot to use that geom function instead of the stat function? Use diamonds data and plot cut vs. depth.

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



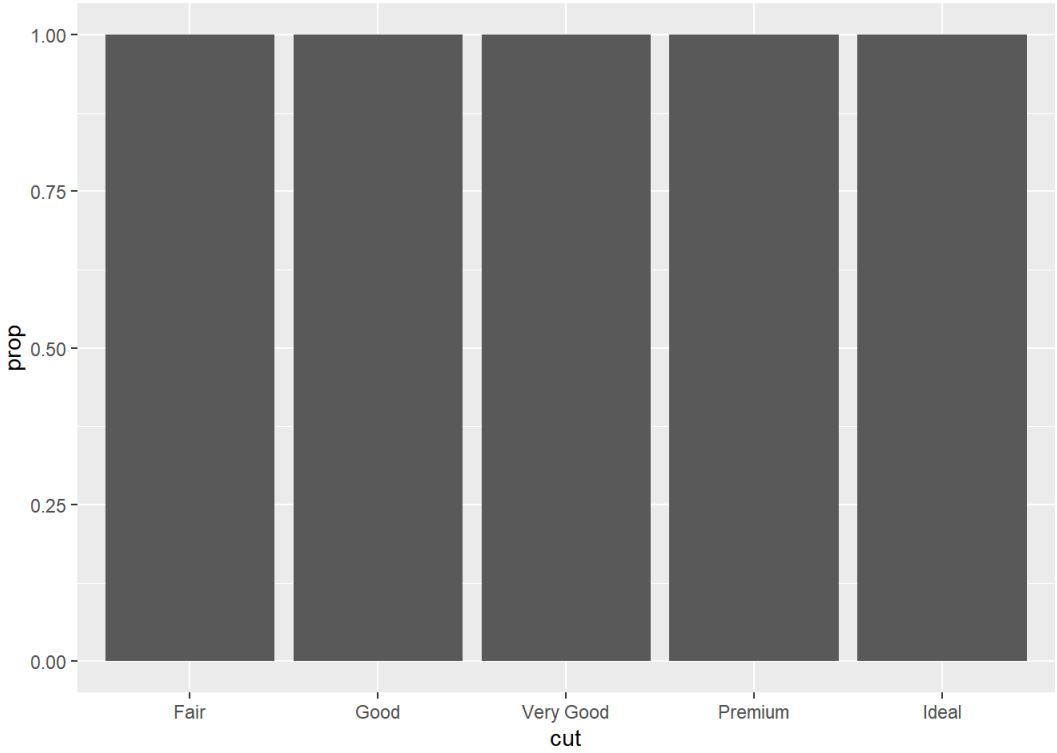

```
diamonds %>% ggplot(aes(x=cut, y=depth)) +
  geom_pointrange(stat='summary')
```



위 두 가지 코드의 결과는 같다.

3. Bar charts for proportion

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop..))
```

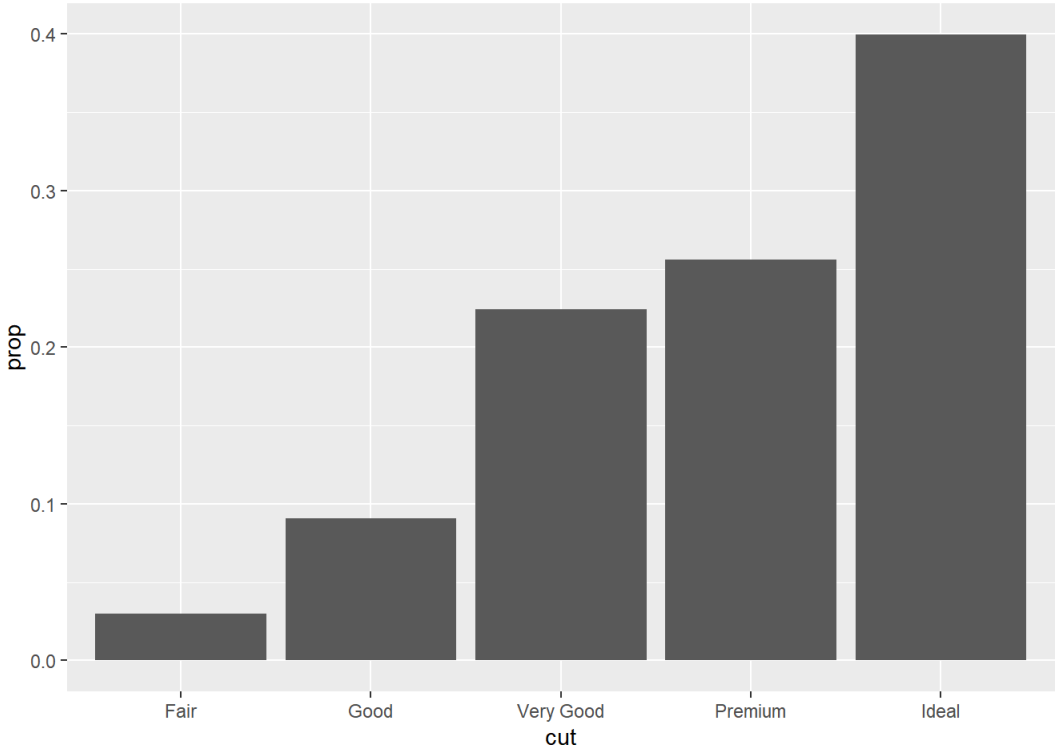


proportion에 대해서 그림을 위

와 같이 그려진다. 이 문제점을 해결하기 위해서 group = 1을 넣어주면 원하는 결과를 얻을 수 있다.

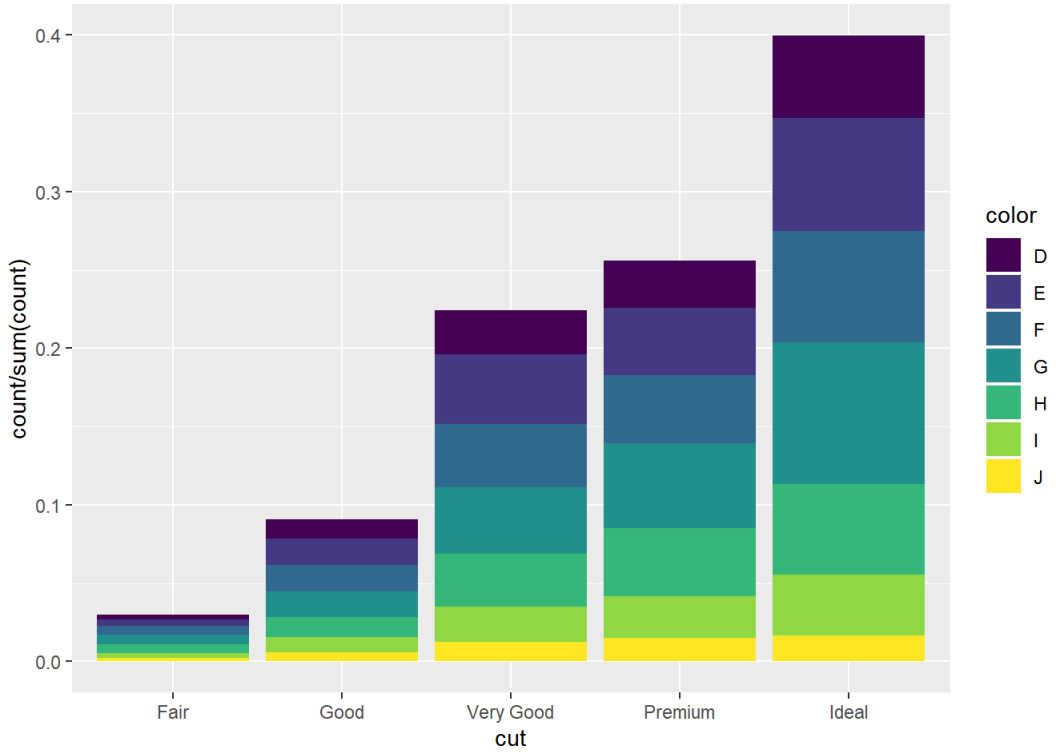
- [3번 해결 방법] : group = 1

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



4. With the fill aesthetic, the heights of the bars need to be normalized.

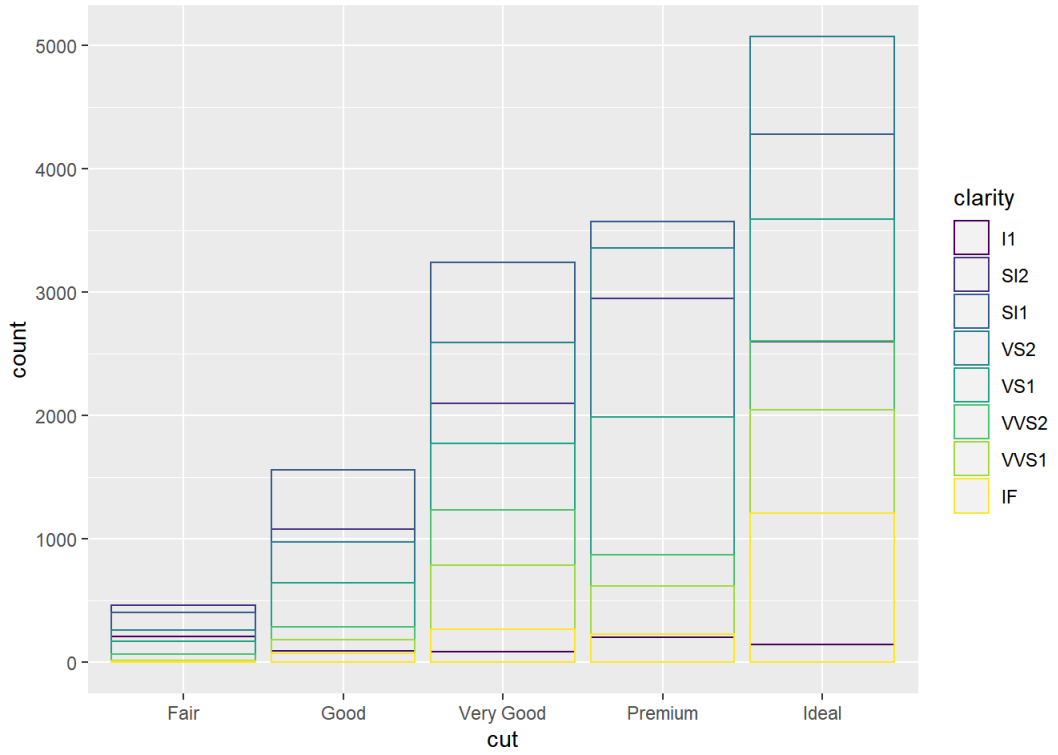
```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y =..count.. / sum(..count..) ,fill=color))
```



- [참고]

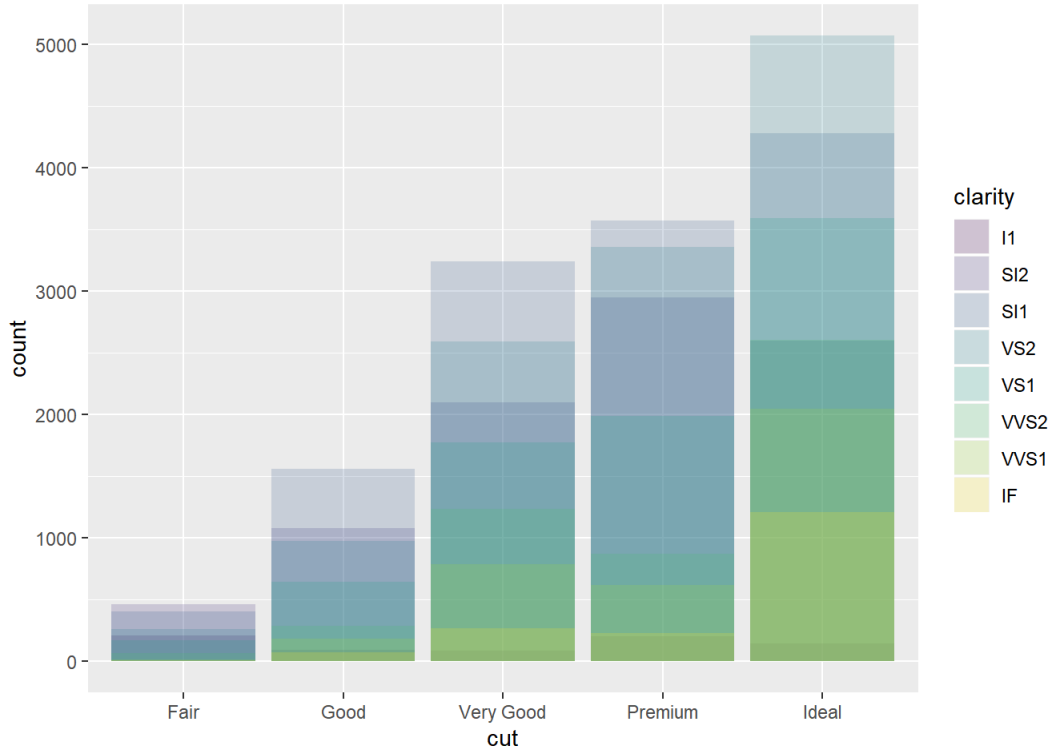
1. color : 도형의 겉 색깔을 의미한다.
2. fill : 도형의 안 색깔을 의미한다. + (position = “identity”)

```
ggplot(data = diamonds,
       mapping = aes(x = cut, colour = clarity)) +
  geom_bar(fill = NA, position = "identity") #도형 안 색깔이 칠해지지 않는다.
```



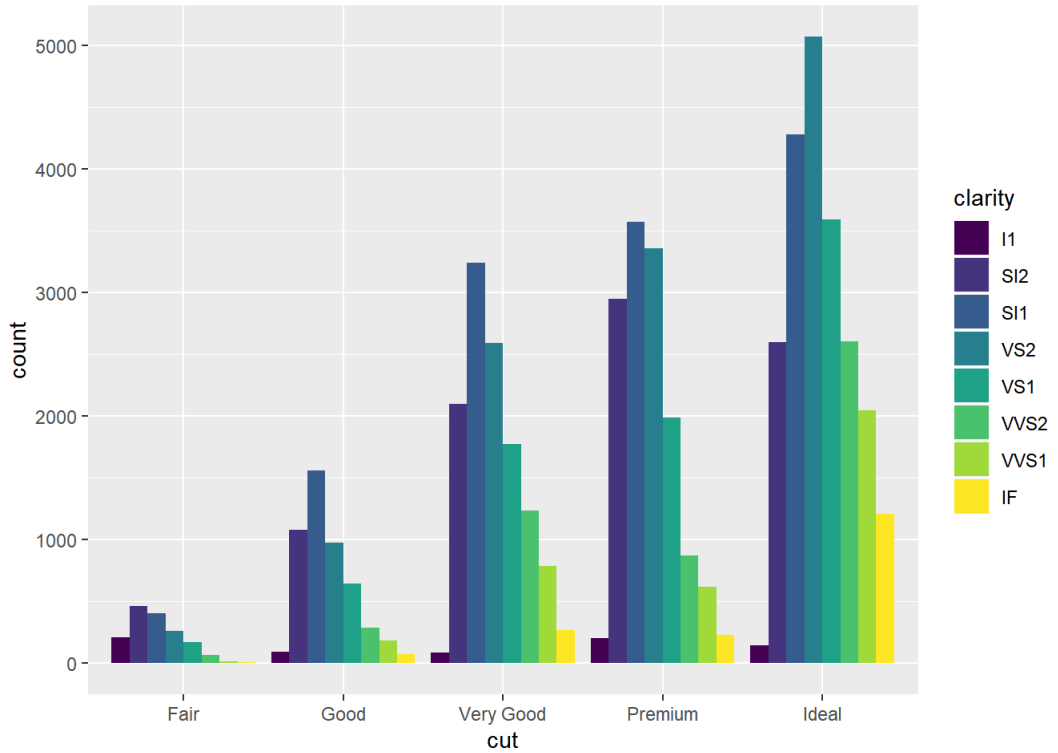
3. alpha : color를 희미하게 해준다.

```
ggplot(data = diamonds,
       mapping = aes(x = cut,fill = clarity)) +
  geom_bar(alpha = 1/5, position = "identity")
```



4. dodge : 옆으로의 분포를 보여준다.

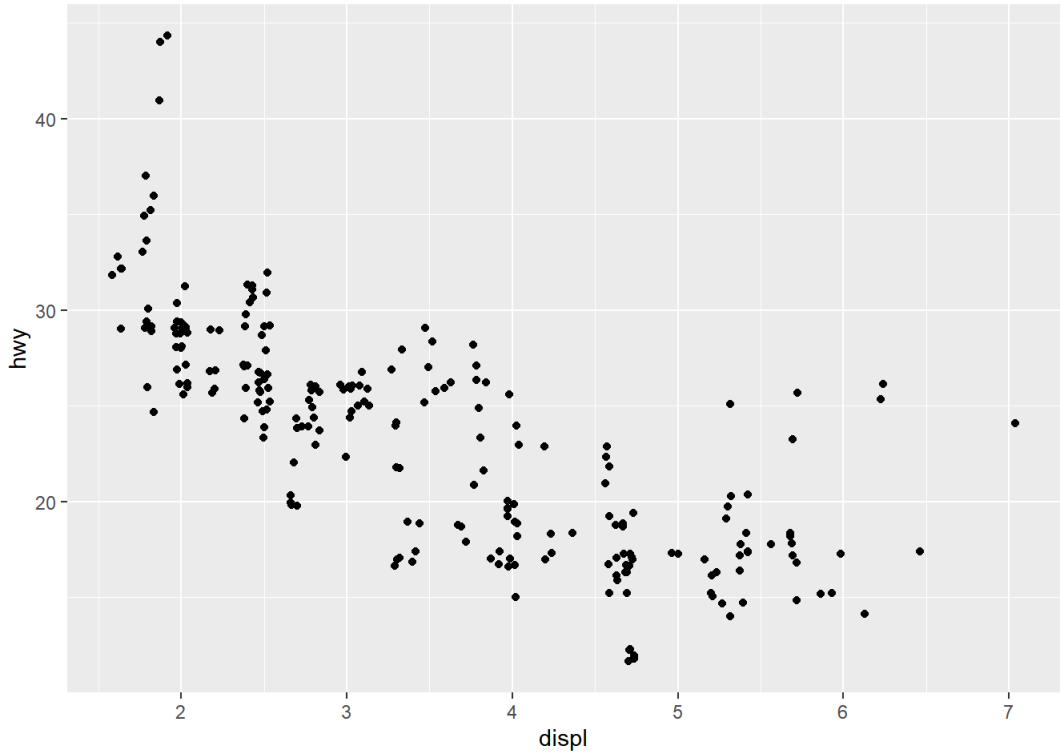
```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity),
           position = "dodge")
```



5. jitter: 겹치는 애들을 떨어져서 보여준다.

흔들어주는 효과! 좀 더 분포를 자세히 보겠다는 뜻이다. 랜덤 에러를 더해주는 방식으로 흐트려준다.

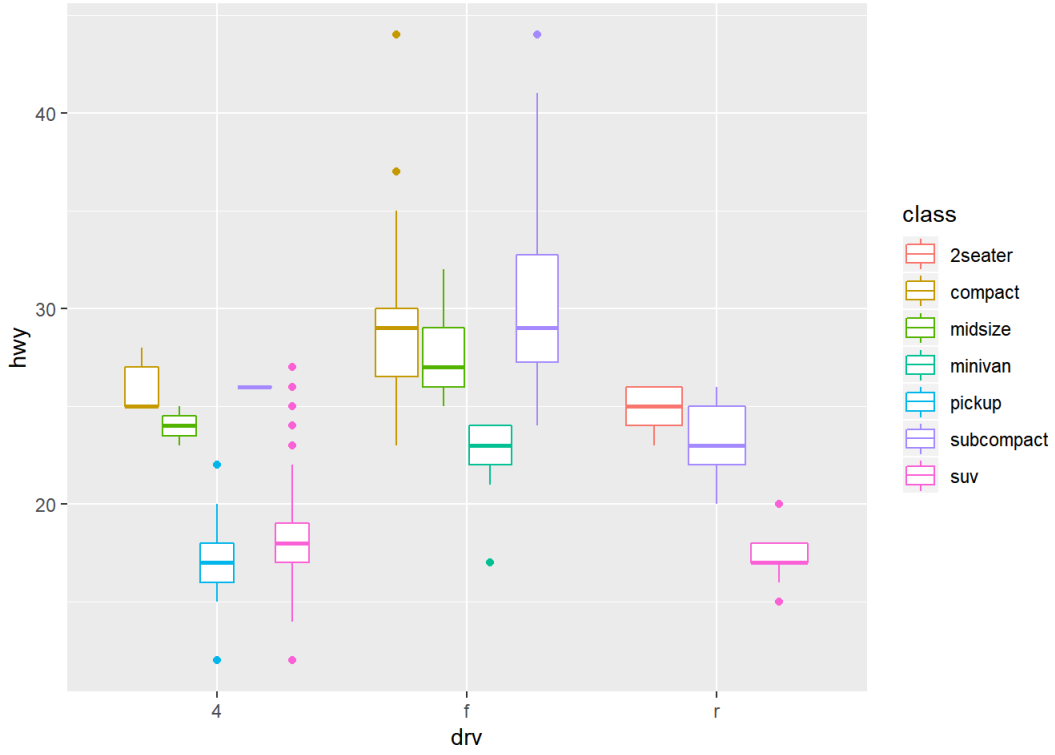
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy),
             position = "jitter")
```


hwy
40
30
20
2
3
4
5
6
7
displ

[Your turn 07]

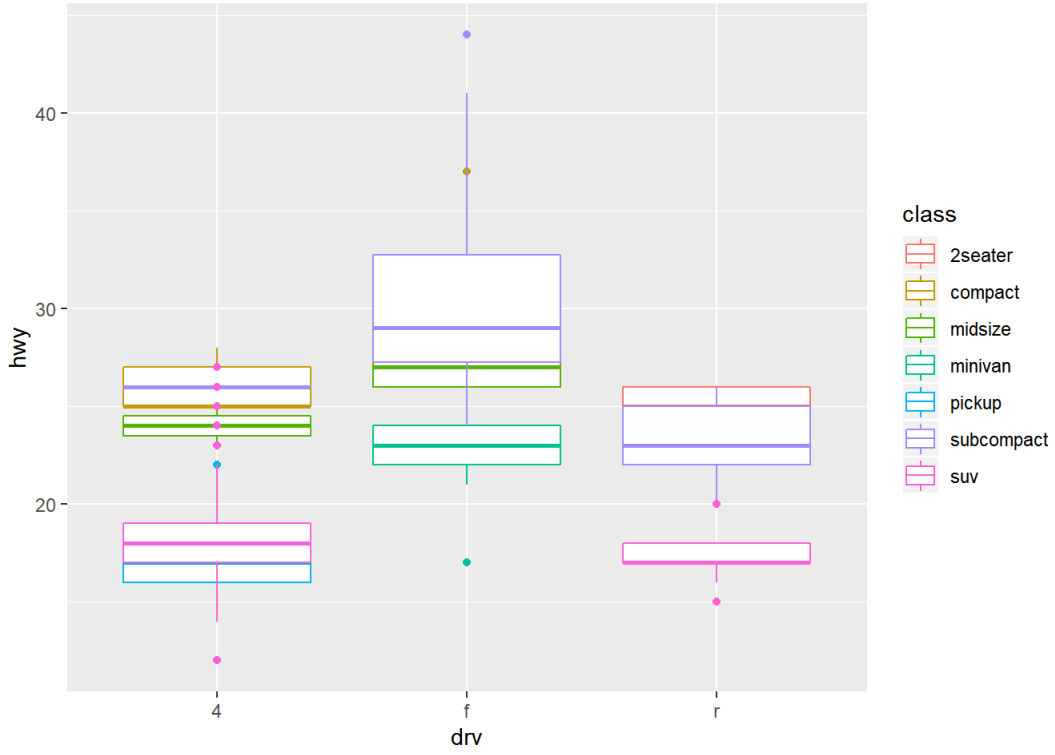
1. What's the default position adjustment for geom_boxplot()? Create a visualization of the mpg dataset that demonstrates it.

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



2. What's the default position adjustment for geom_boxplot()? Create a visualization of the mpg dataset that demonstrates it.

```
ggplot(data = mpg, aes(x = drv, y = hwy, colour = class)) +
  geom_boxplot(position = 'identity')
```

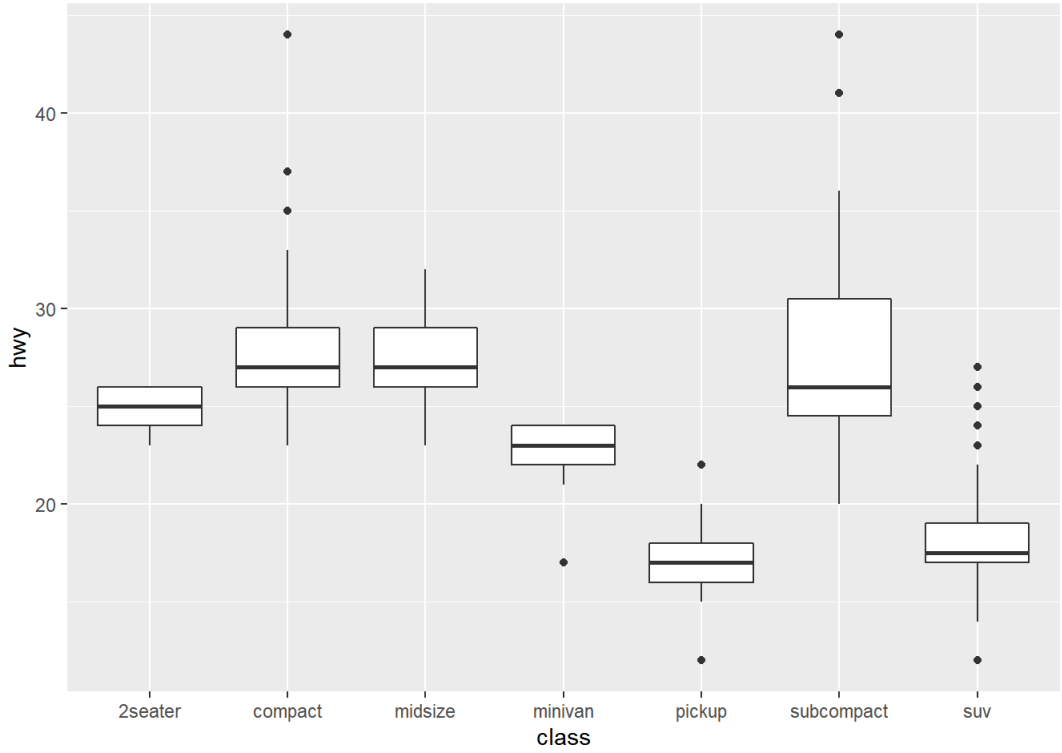


7. Coordinate systems

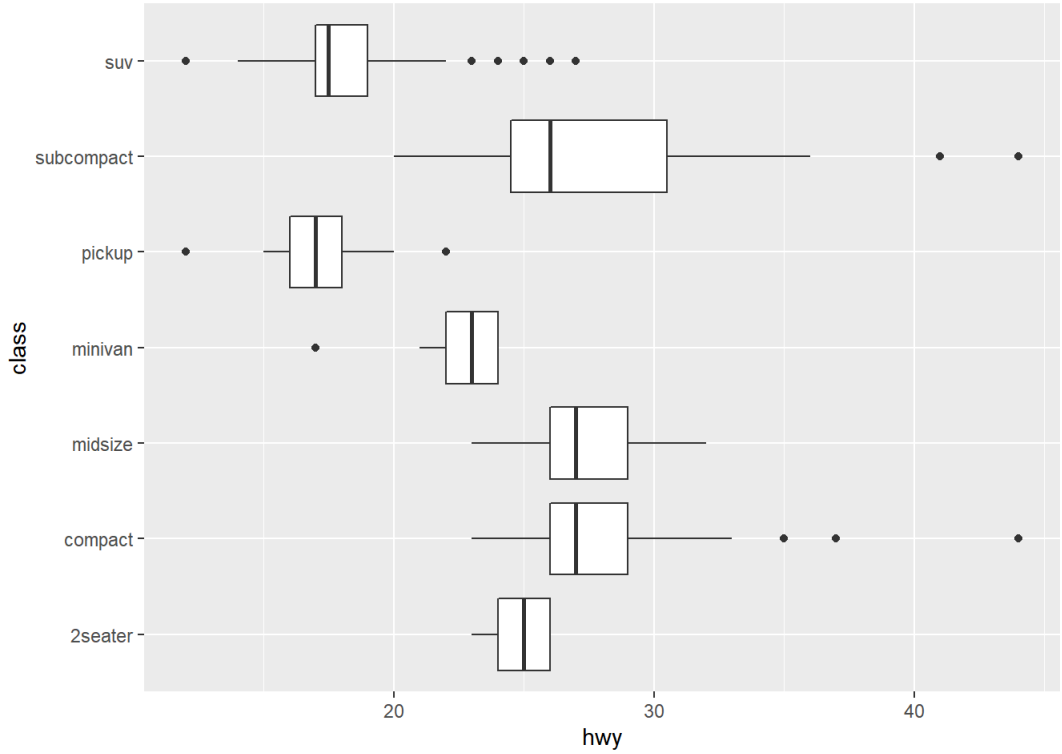
[example]

- cord_filp : x축과 y축을 바꾼다.

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



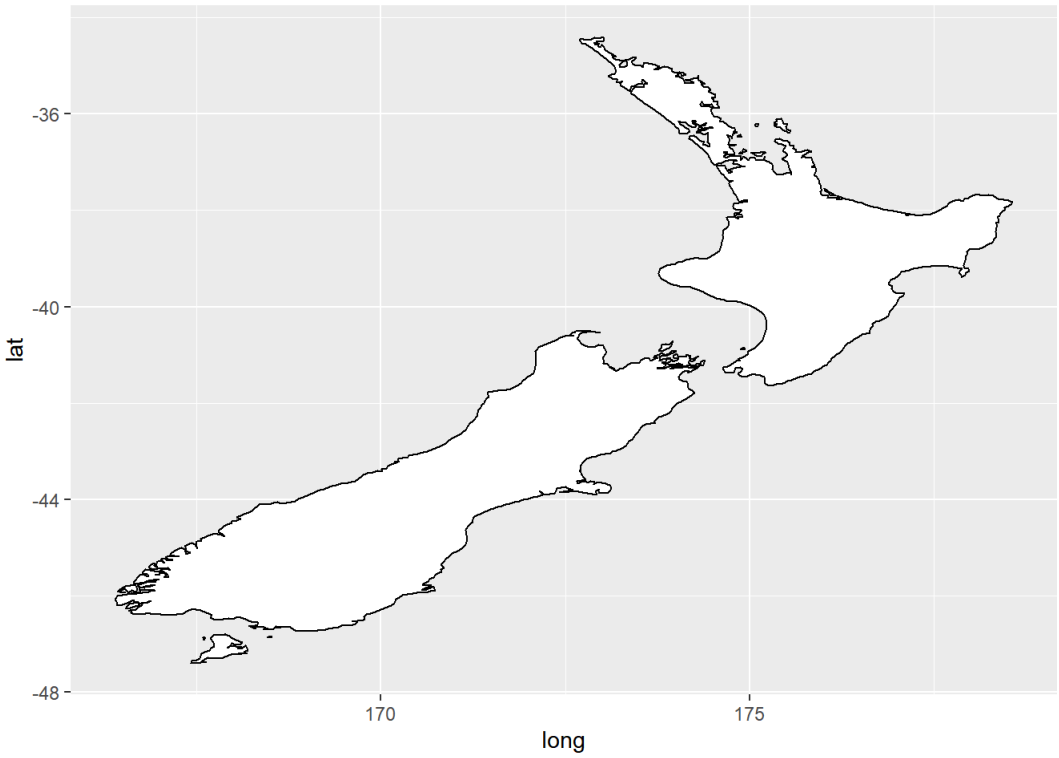

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot()+
  coord_flip()
```



- maps packages

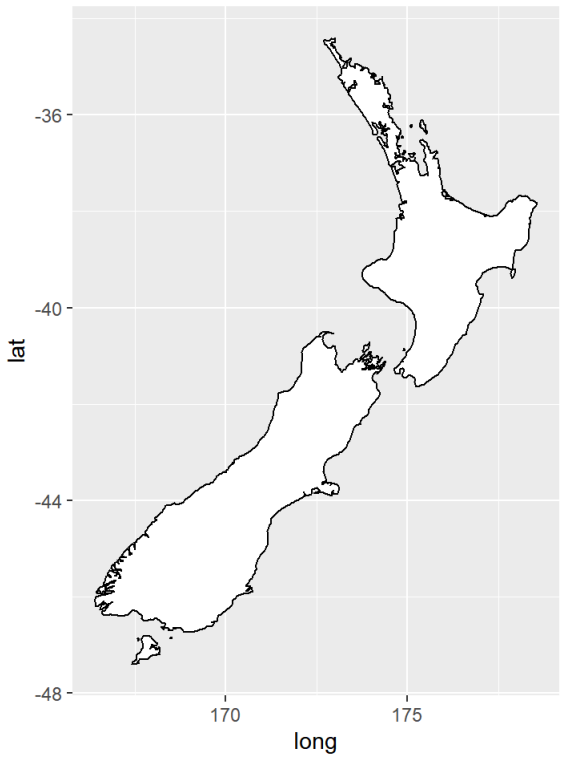
```
library(maps)
nz <- map_data("nz")

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



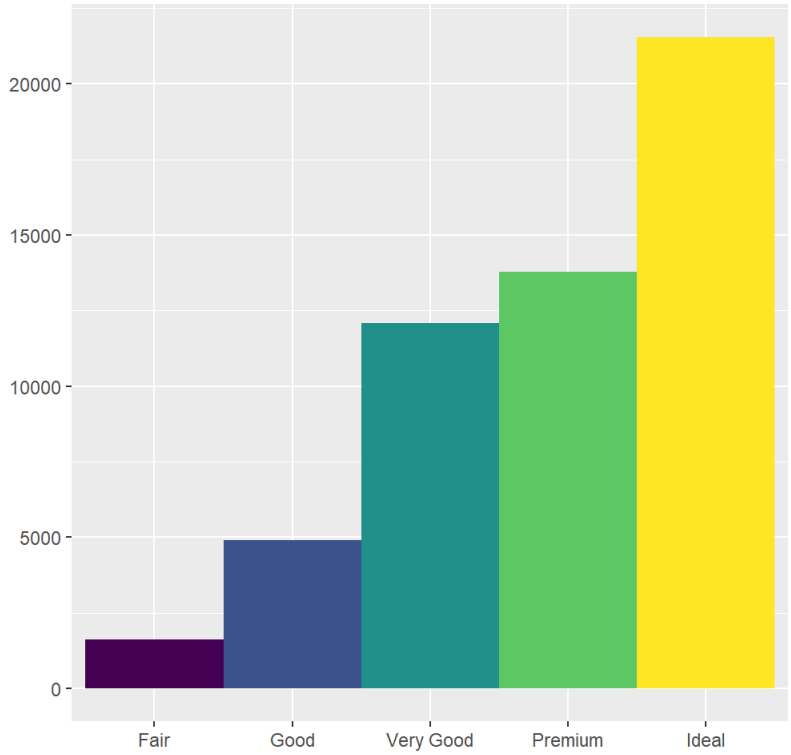
- coord_quickmap(): 눈에 익숙한 형태로 그려준다.

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

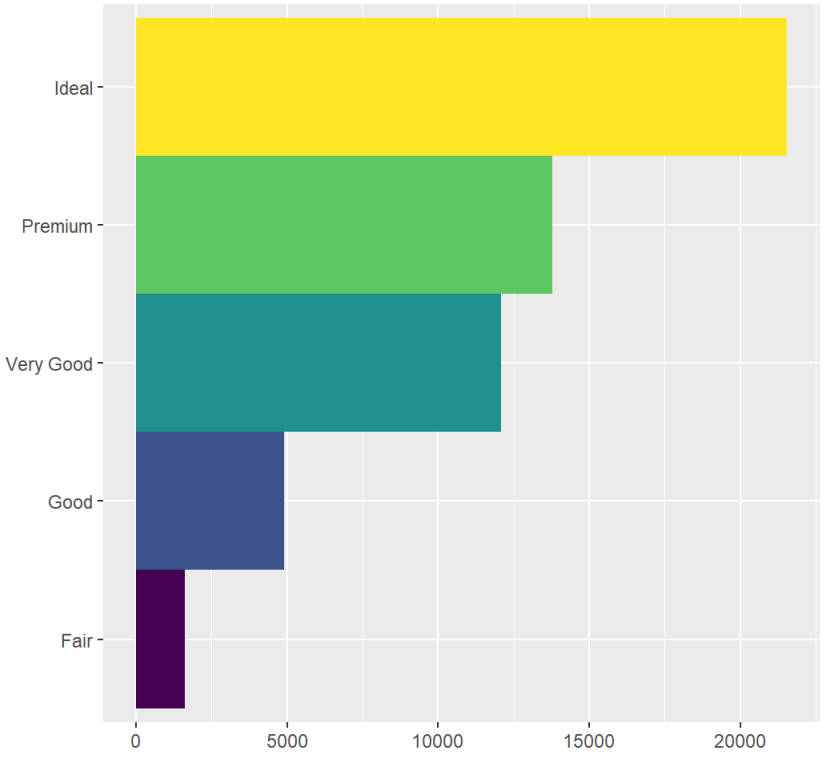


- coord_polar():크기에 따른 부채꼴 그림 그려준다.

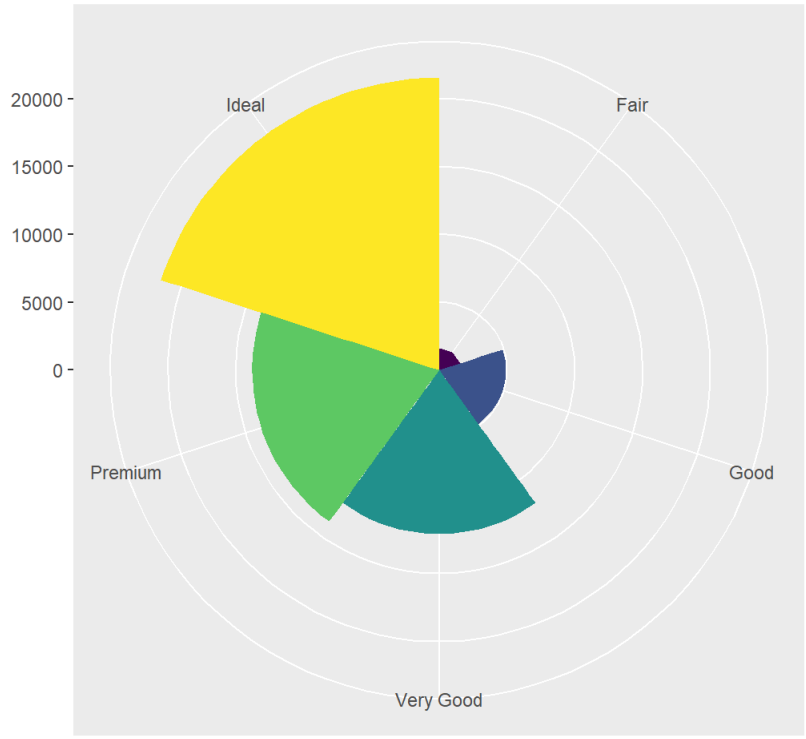
```
bar <- ggplot(data = diamonds) +
  geom_bar(
    mapping = aes(x = cut, fill = cut),
    show.legend = FALSE,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)
bar
```




```
bar + coord_flip()
```



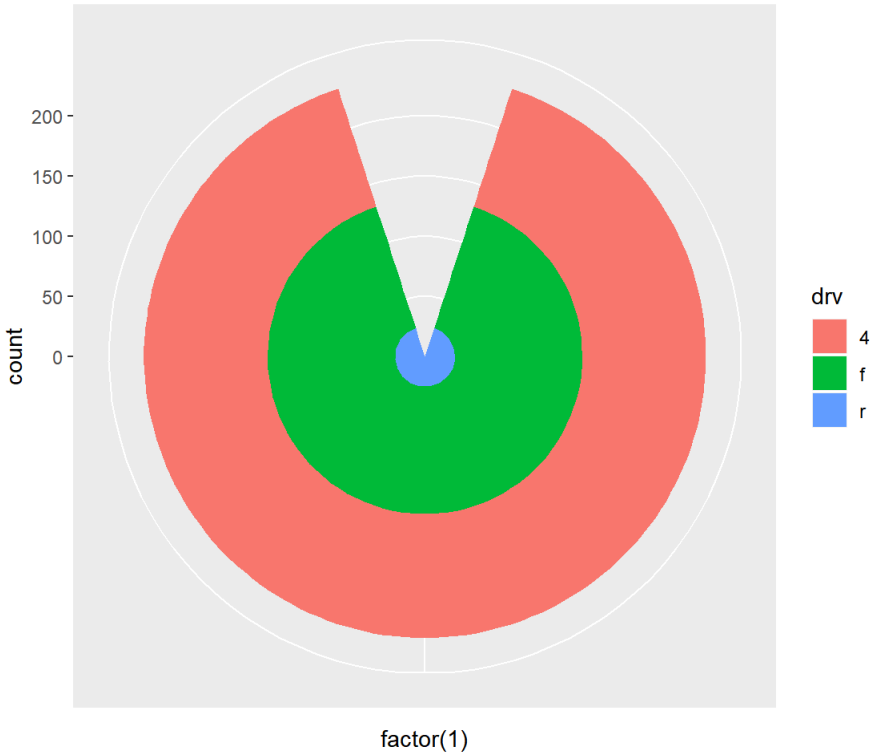

```
bar + coord_polar()
```


20000
15000
10000
5000
0
Ideal
Fair
Premium
Good
Very Good

[Your turn 08]

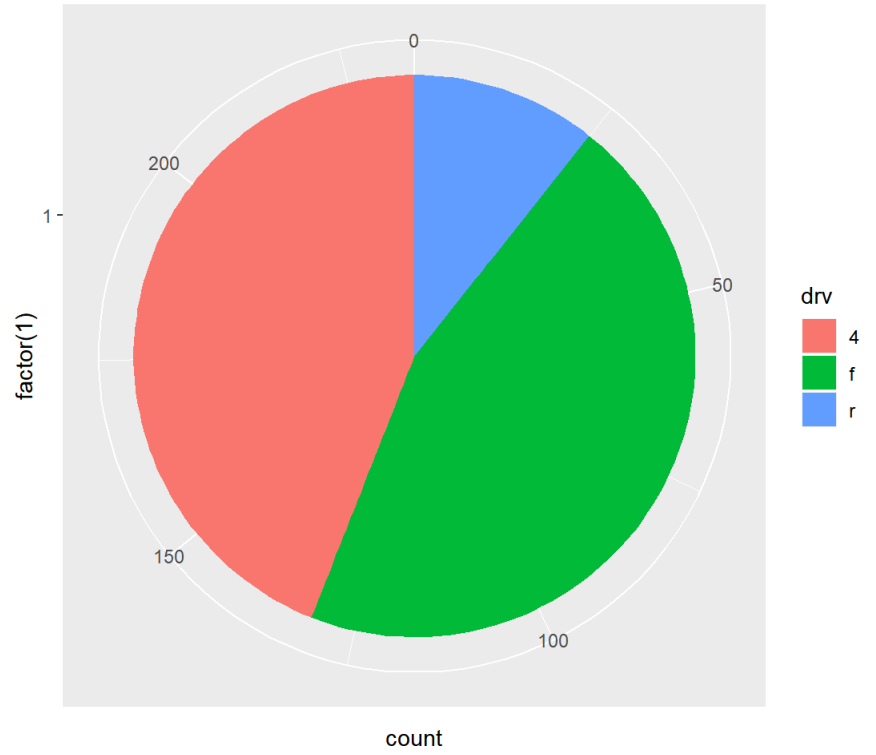
1. Turn a stacked bar chart into a pie chart using coord_polar().

```
mpg %>% ggplot(aes(x = factor(1), fill = drv))+
  geom_bar()+
  coord_polar()
```



2. Now add coord_polar(theta="y") to create pie chart.

```
ggplot(mpg, aes(x = factor(1), fill = drv)) +
geom_bar(width = 1) +
coord_polar(theta = "y")
```



3. The argument theta = "y" maps y to the angle of each section. If coord_polar() is specified without theta =

“y”, then the resulting plot is called a bulls-eye chart.

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot() +
  coord_flip() +
  labs(y = "Highway MPG",
       x = "Class",
       title = "Highway MPG by car class",
       subtitle = "1999-2008",
       caption = "Source: http://fueleconomy.gov")
```