L07. *ggplot* (1)

Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

SEOULTECH 서울과학기술대학교 데이터사이언스학과

1. I. Preparation
2. II. Scatterplot (산점도)
3. III. Faceting

Section 1

I. Preparation

ggplot2 패키지 준비

```
install.packages("ggplot2")
library(ggplot2)
```

mpg 데이터셋

- ggplot2 패키지에 내장된 데이터셋
- 1999년과 2008년에 생산된 차량들의 연비 데이터
- 데이터셋 diamond, iris와 함께 R에서 가장 많이 예제로 쓰임

More info on *mpg*

```
? mpg # getting help from web
help(mpg) # getting help from help panel
```

*mpg*의 struture

```
str(mpg)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

*mpg*의 변수

변수이름	설명
`manufacturer`	제조사
`model`	차종
`displ`	엔진크기
`year`	생산연도
`cyl`	기통
`trans`	트랜스미션
`drv`	전륜(f)/후륜(r)/사륜(4)
`cty`	도심 마일리지
`hwy`	고속도로 마일리지
`fl`	연료종류 (fuel)
`class`	클래스 (중형, 트럭, SUV,…)

Section 2

II. Scatterplot (산점도)

Motivation

- 엔진의 용량이 크면 연비가 안 좋을까?
- 산점도 (scatterplot)
  - x축: 엔진 크기(`displ`)
  - y축: 고속도로 마일리지(`hwy`)

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

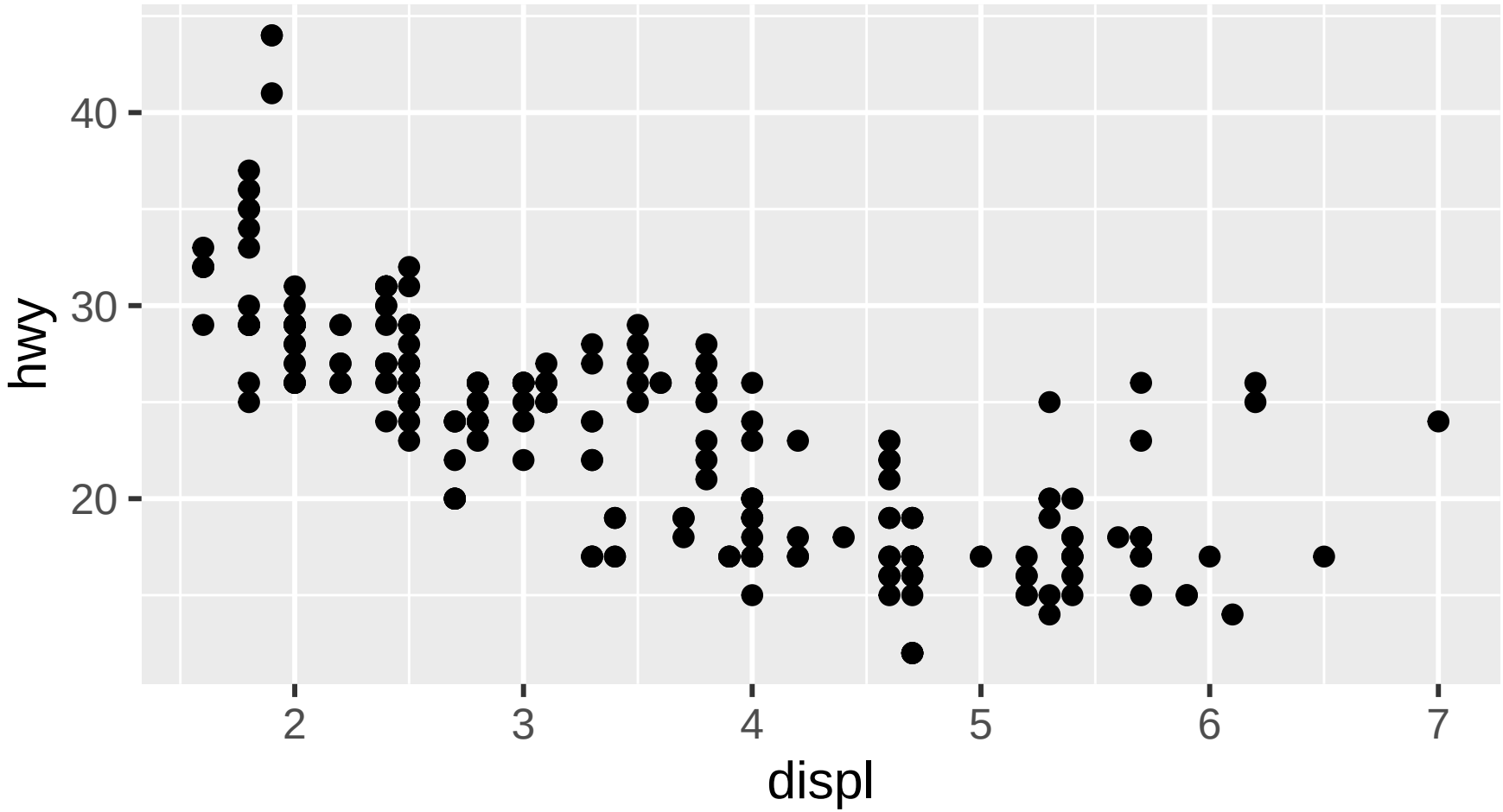


Figure 1: 가장 간단한 산점도

- 엔진의 용량이 크면 연비가 안 좋을까?

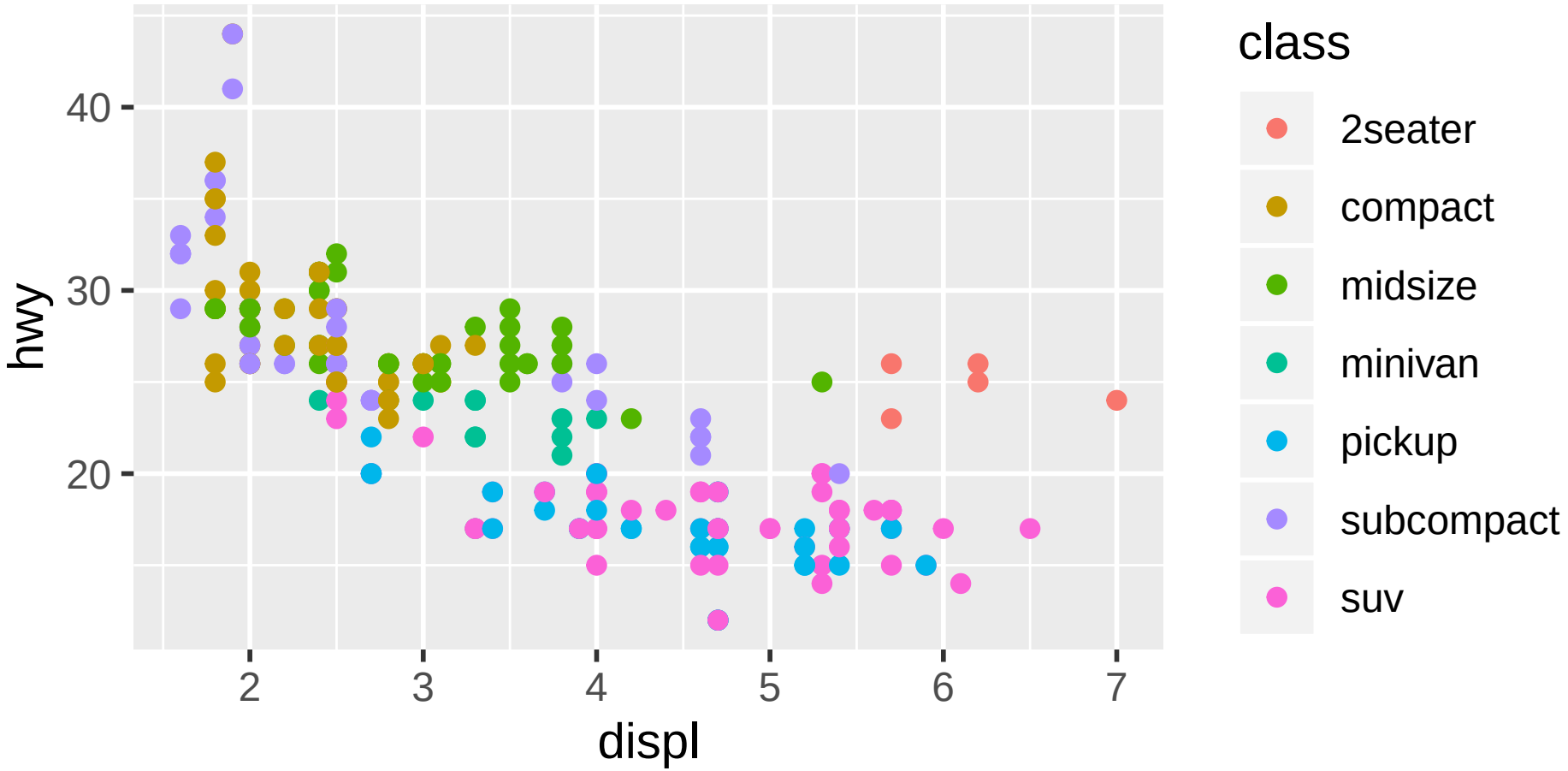
Basic Syntax

```
# The plot above
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))

# The basic ggplot syntax
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

a: *color = class*

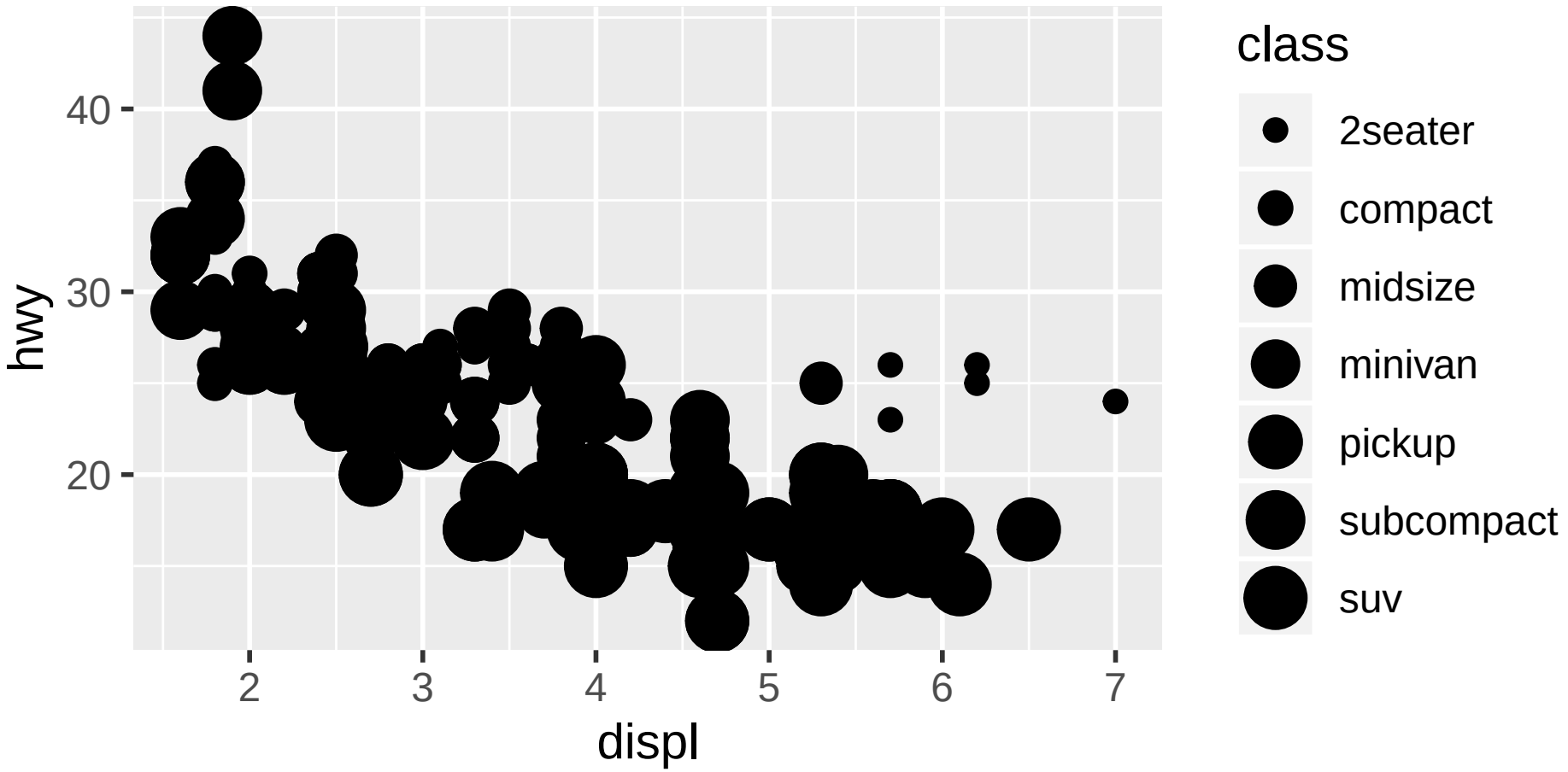
```
a <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
a
```



- `class` 변수의 값에 따라 color가 다른 산점도
- 위의 `ggplot` 객체를 변수 `a`에 저장

b: *size = class*

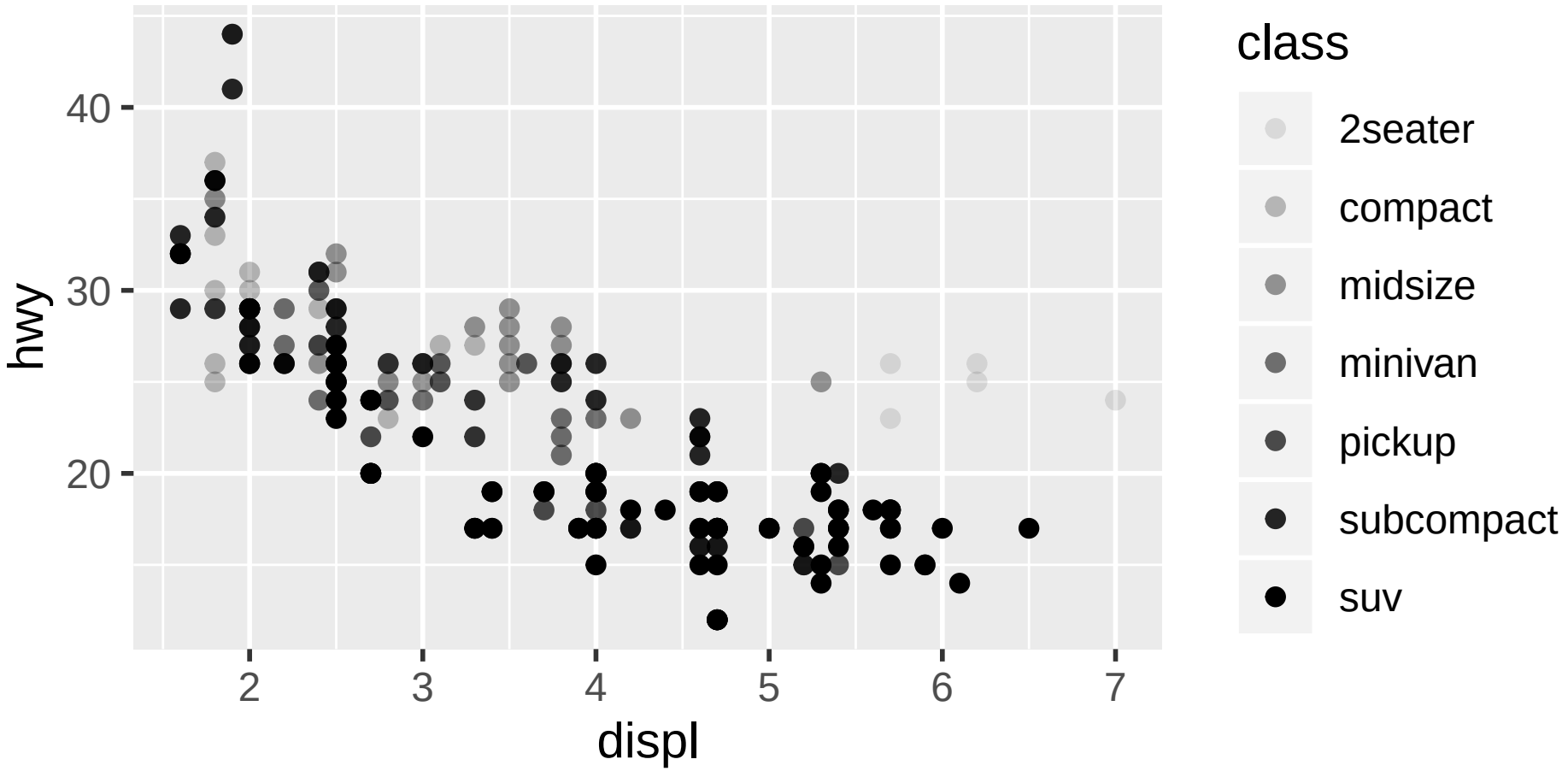
```
b <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
b
```



- `class`변수의 값에 따라 size가 다른 산점도

c: *alpha = class*

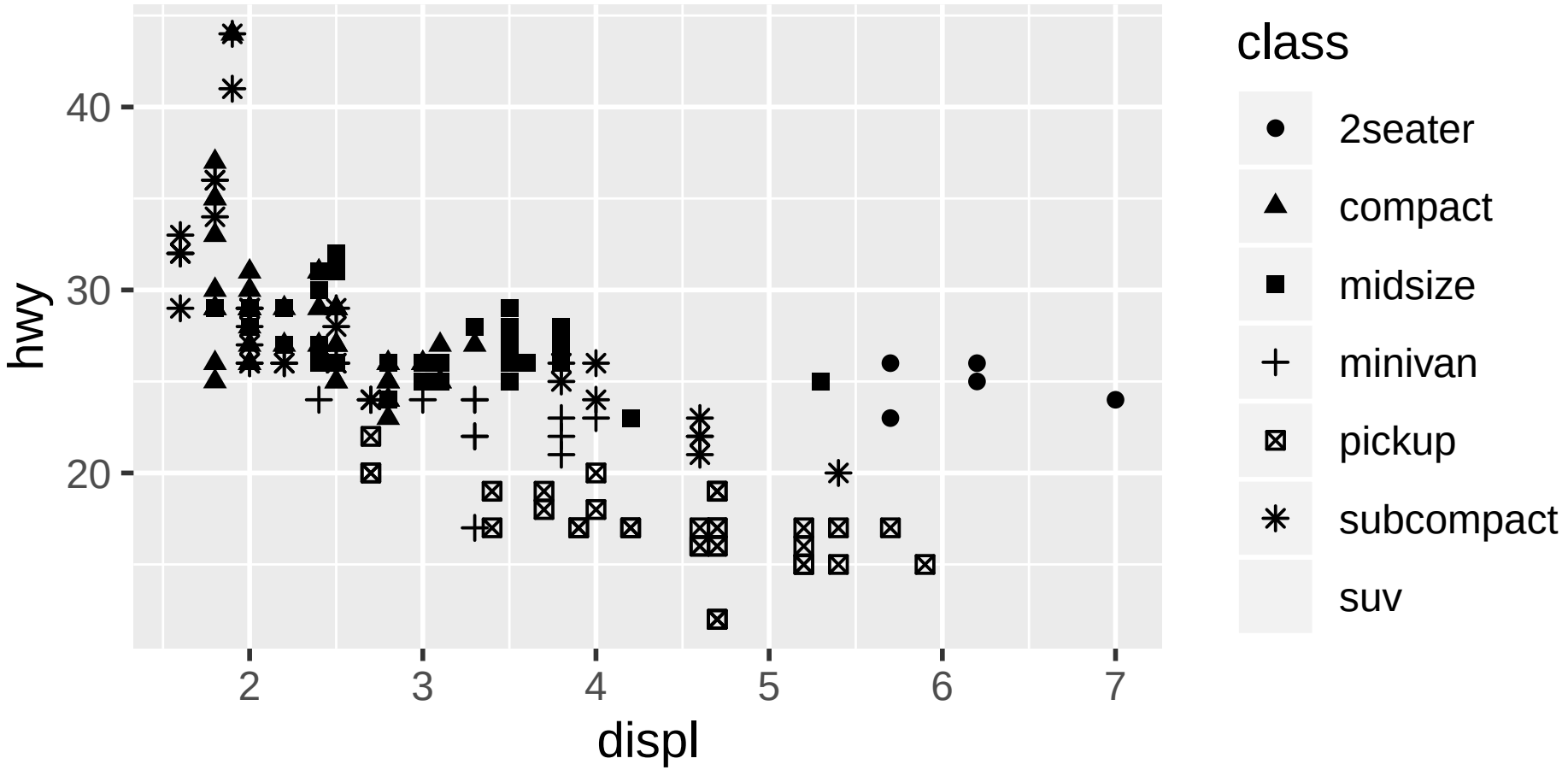
```
c <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
c
```



- `class` 변수의 값에 따라 alpha(진하기)가 다른 산점도

d: *shape = class*

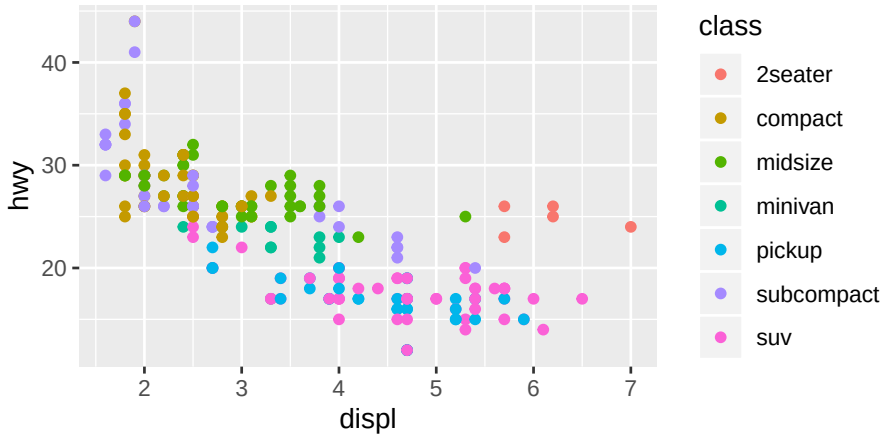
```
d <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
d
```



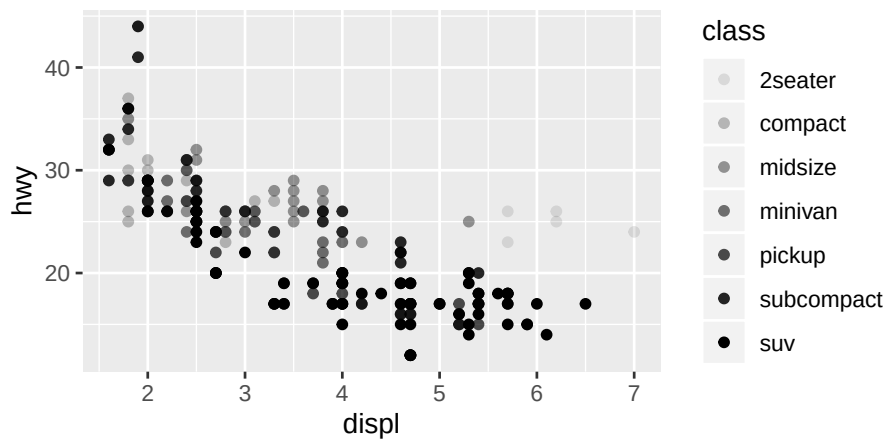
- `class`변수의 값에 따라 shape이 다른 산점도

Discussion on a,b,c,d

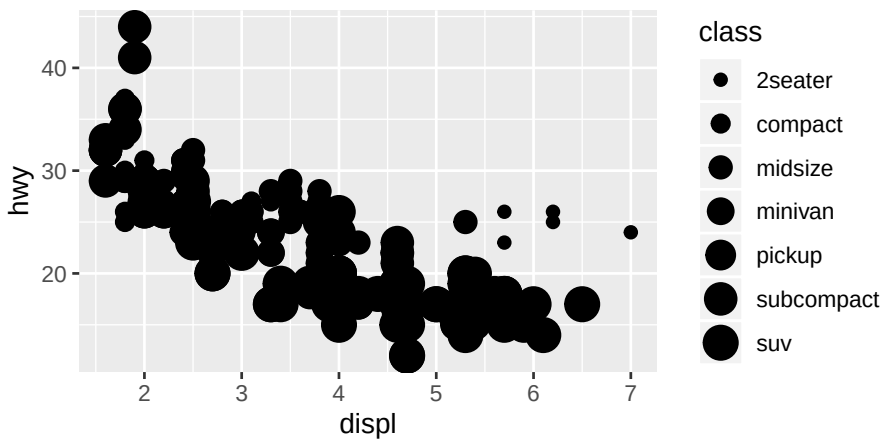
a # color = class



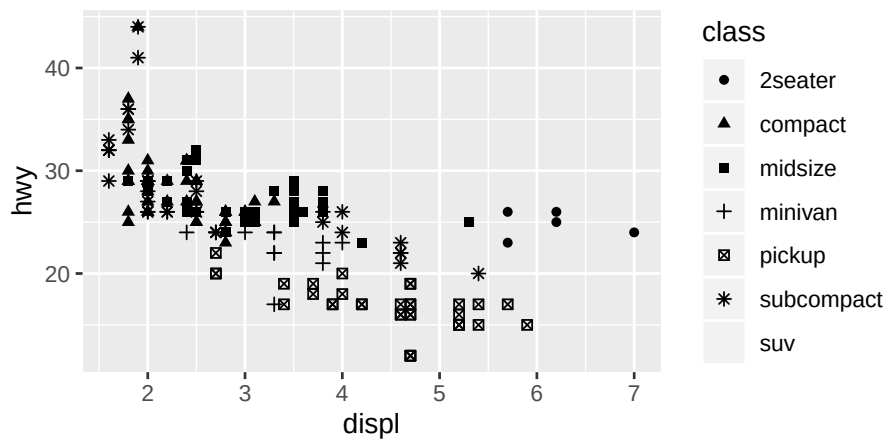
c # alpha=class



b # size = class



d # shape=class



- Which is the best? Why?

`class`는 어떤 변수인가?

- '종류가 다른것'에 해당하지만,
- '크고 작은것'에 대한 특징을 담고 있지 못하다.
- 그러므로 `color` 혹은 `shape`의 variation을 주는 것이 바람직하다.

R에서도 warning message를 제공한다.

- b (`size=class`)의 warning message
  - 'Using size for a discrete variable is not advised.'
- c (`alpha=class`)의 warning message
  - 'Using alpha for a discrete variable is not advised.'

변수의 분류

- 수량적 특성 – 수치로 표현되는가?
  - 질적변수 (qualitative, `factor`) – 서열이 있는가?
    - 비서열 질적변수 (unordered-qualitative, `unordered factor`)
    - 서열 질적변수 (order-qualitative, `ordered factor`)
  - 양적변수 (quantitative, `numeric`) – 변수가 어떤 실수값이나 가질수 있는가?
    - 연속형 (continuous, `float`)
    - 이산형 (discrete, `integer`)
    - Some data are continuous but measured in a discrete way, e.g. age.
- Q. 연속형 변수를 이산형으로 만들고 서열 질적변수로 만드는 예를 생각해보라.
- Q. 연속형 변수에 비해서 서열 질적변수의 장점은 무엇인가?

*color*와 *shape*

- **정확히 얘기하자면, `class`는 비서열 질적변수 (unordered-qualitative, `unordered factor`)이므로 `color` 혹은 `shape`가 가장 적절하다.**
- `color`와 `shape`의 차이
  - 디스플레이나 프린터에서 컬러를 지원한다면 `color`가 preferred.
  - 두 가지를 모두 implement하는 것이 바람직함.
  - 논문낼때 컬러그림을 내면 추가 비용을 받는 저널도 있음.
- 변수가 가질수 있는 값의 갯수 `length(unique(mpg$class))`와 그래픽
  - 인간이 쉽게 인식할 수 있는 `color`의 갯수는 최대 7개 (magic number 이론)
  - R에서 warning 없이 생성하는 `shape`의 갯수는 최대 6개
    - 'The shape palette can deal with a maximum of 6 discrete values because more than 6 becomes difficult to discriminate; you have 7. Consider specifying shapes manually if you must have them.'
  - 독자의 배경지식에 따라서 허용치가 다르겠지만, 4-5개 정도가 적절하다.

*color*와 *shape*

- Q. 변수가 가질수 있는 값의 갯수가 많은 경우에는 어떻게 할 것인가?
  - A. `facet` 기능을 이용한다.
  - A. 비서열 질적 변수는 대부분 further grouping이 가능하다. 이를 통해 간결한 그래픽을 만들어 낼 수 있다.

```
unique(mpg$class)
```

```
## [1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
## [6] "pickup"     "subcompact"
```

- Q. 데이터 포인트가 너무 많아서 복잡해 보인다면 어떻게 할 것인가?
  - A. `facet` 기능을 이용한다.
  - A. 다른 geometric object들, `line`, `smooth`, `path`등을 사용한다. 만약에 이들 geometric object로 잘 요약이 되지 않는다면, 애초에 plot을 그릴 가치가 없는 데이터이다.

size vs *alpha*

- 양적변수(연속형과 이산형 모두)와 서열질적변수에 대해서 사용할 수 있다.
- R의 warning message
  - Original: 'Using size for a discrete variable is not advised.'
  - May-be-better: 'Using size for a 비서열 질적 variable is not advised.'
- 변수의 상한과 하한이 정해진 경우에는 `alpha`를 쓰는 것이 바람직하다.
  - `size`: Unbounded quantity
  - `alpha`: Bounded quantity (농도, 비율, 빈도, 확률등)

Takeaway

- 좋은 데이터 시각화에는 다음과 같은 요소가 고려되어야 한다.
  - 독자의 특성 (배경지식, 관심분야, 연령대 등)
  - 데이터의 특성과 본질
  - 전달하고자 하는 메시지
  - 데이터 시각화의 목적은 **커뮤니케이션**

The ideal scatterplot so far

```{r, out.height='80%', fig.width=4.5, fig.cap="The best so far"}
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=class, shape=class))
```

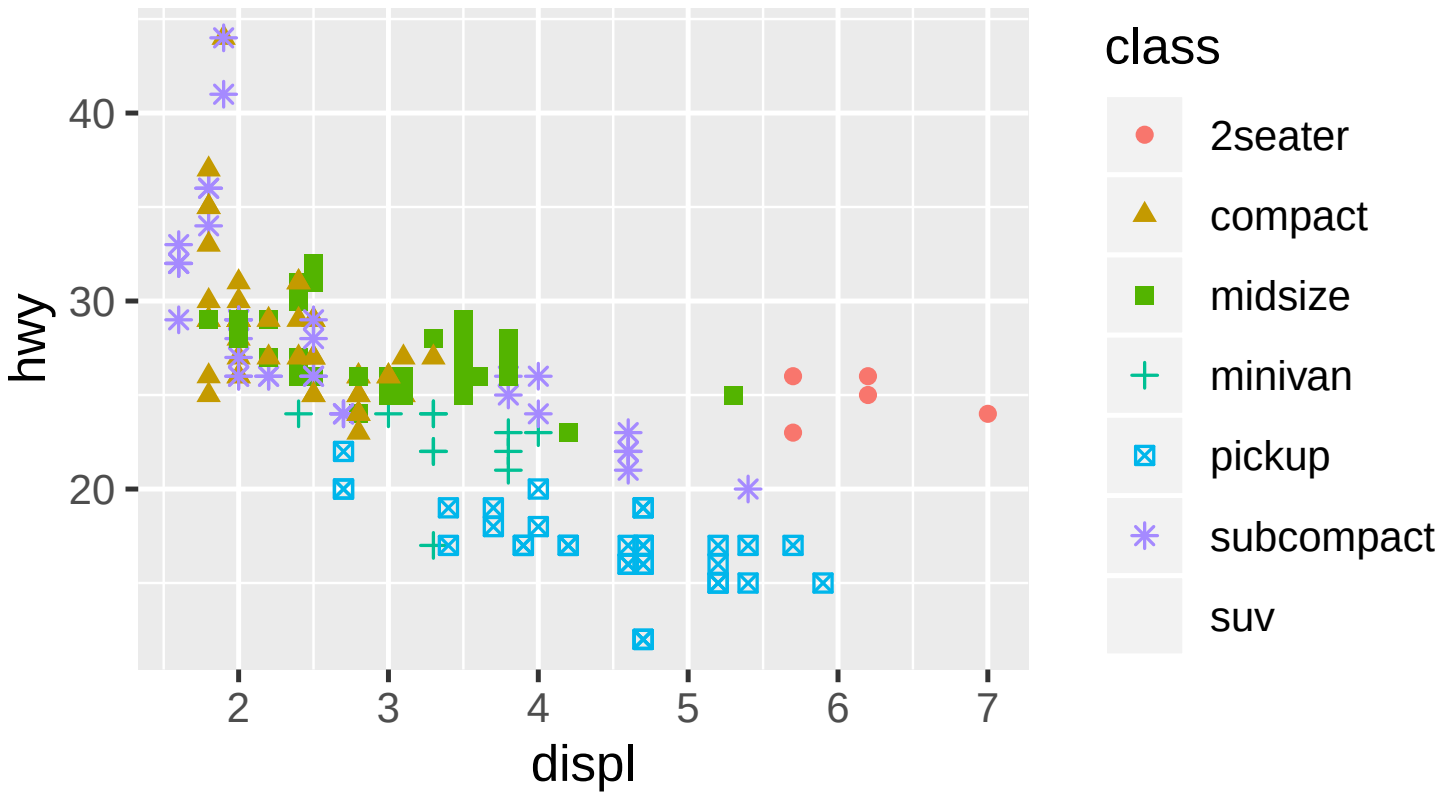


Figure 2: The ideal

Chunk figuration for rmd output

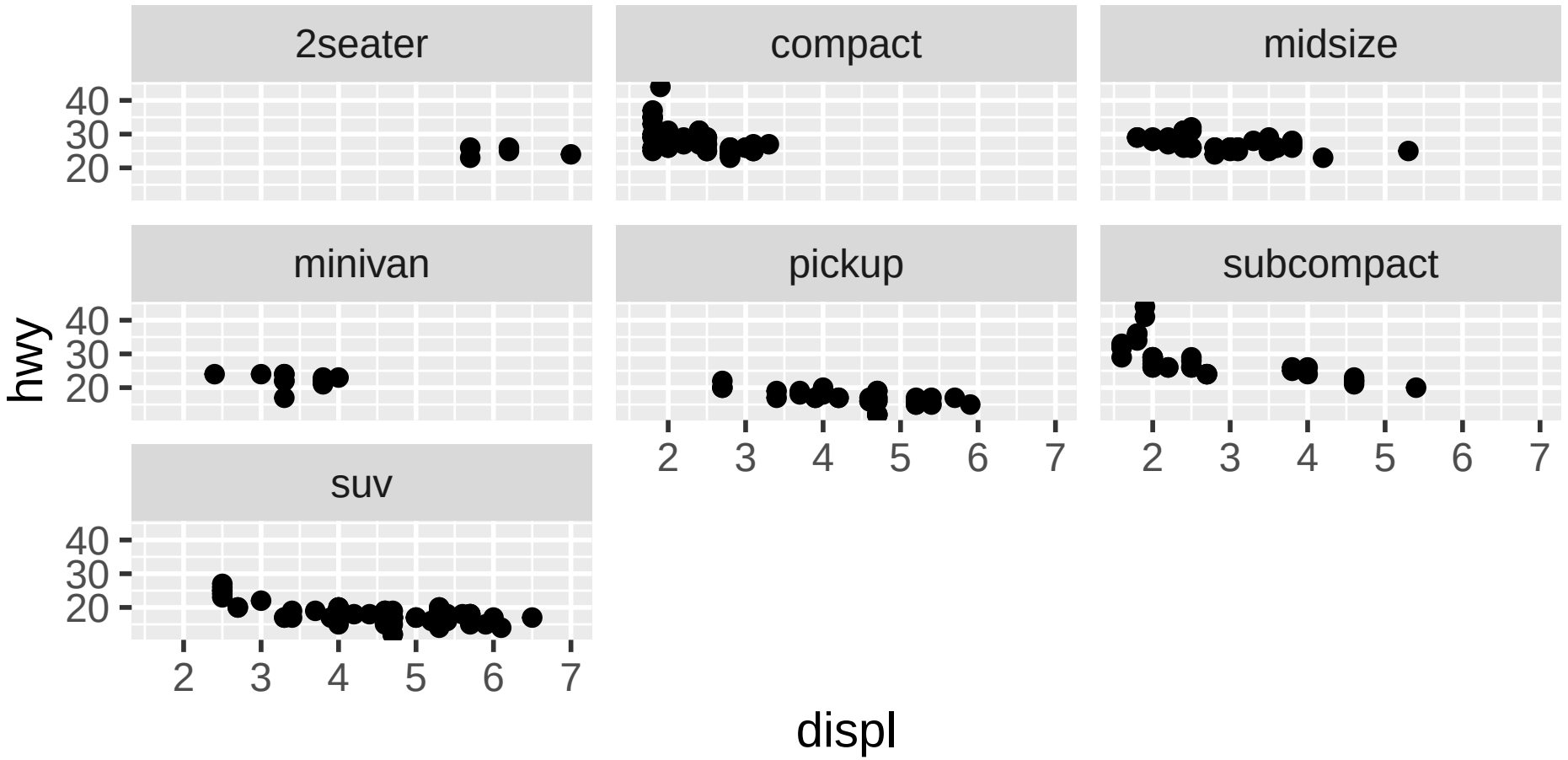
- `fig.width` and `fig.height`
  - The (graphical device) size of R plots in inches.
  - R plots in code chunks are first recorded via a graphical device in **knitr**, and then written out to files.
  - You can also specify the two options together in a single chunk option `fig.dim`, e.g., `fig.dim = c(6, 4)` means `fig.width = 6` and `fig.height = 4`.
- `out.width` and `out.height`
  - The output size of R plots in the output document. These options may scale images.
  - You can use percentages, e.g., `out.width = '80%'` means 80% of the page width.
- `fig.align`
  - The alignment of plots. It can be `'left'`, `'center'`, or `'right'`.
- `fig.cap`
  - The figure caption.

Section 3

III. Faceting

facet_wrap() to increment one dimension

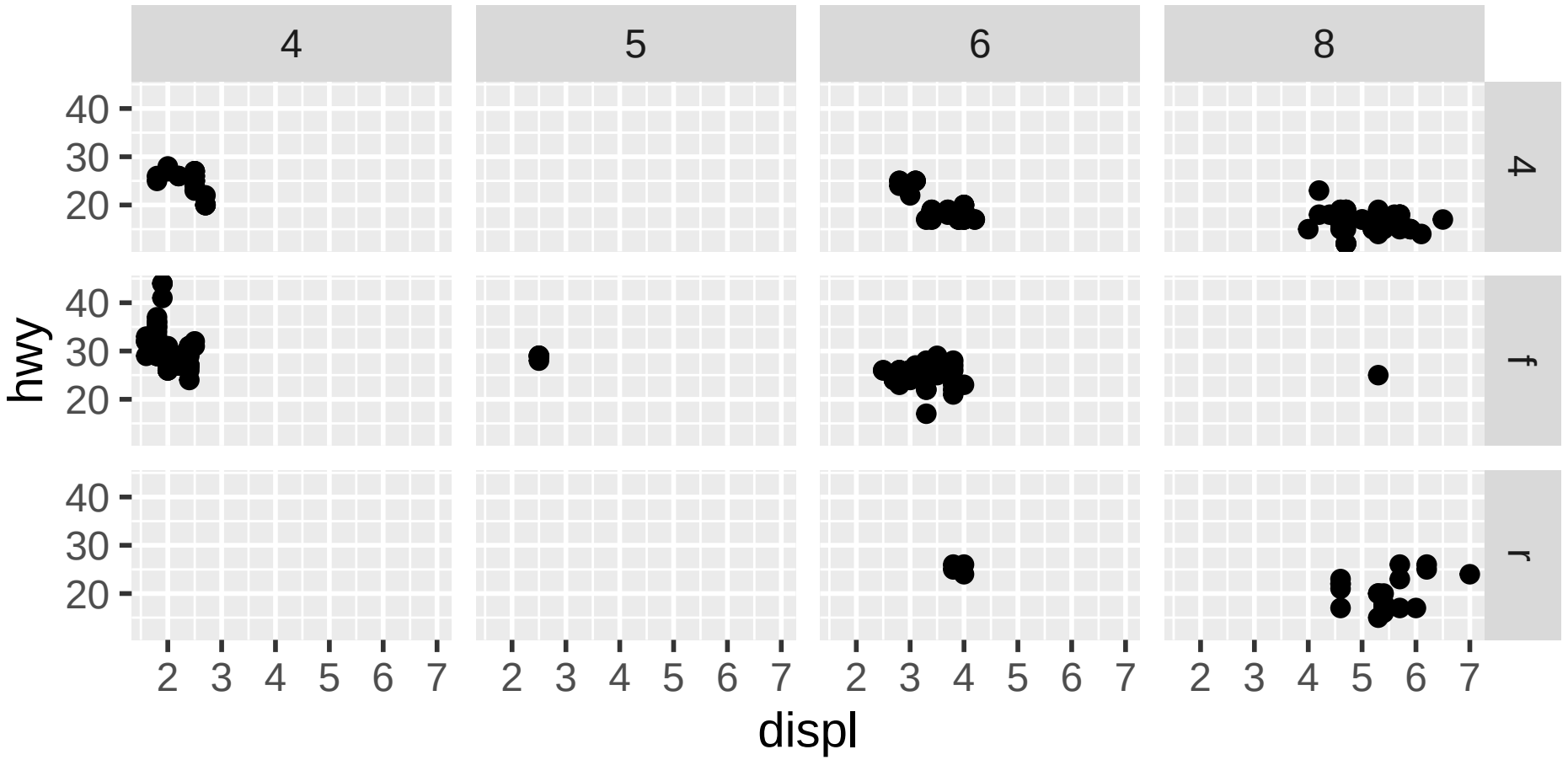
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class)
```



- `class`의 값에 따라서 분할하여 배열

facet_grid() to increment two dimensions

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



- drv를 y축(세로)으로 cyl을 x축(가로)으로 grid 배열

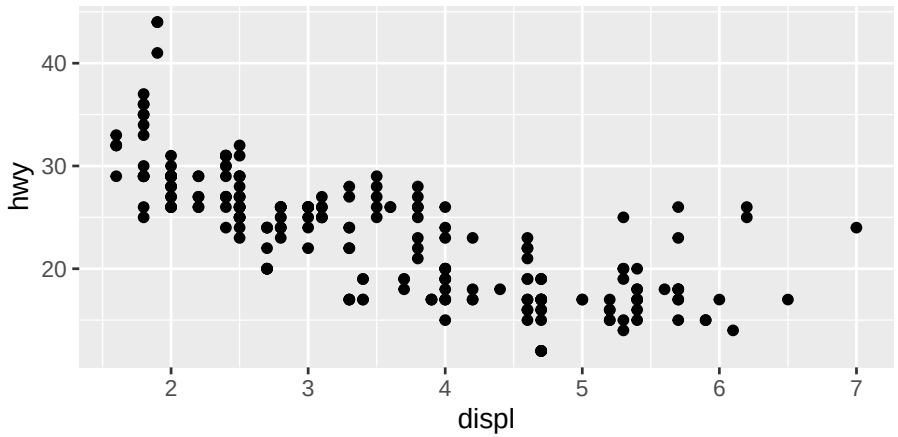
Yes or No faceting?

- Yes faceting
  1. Data가 충분히 많을 때
  2. Categorical 변수의 값에 대해 각각의 distribution을 보고 싶을 때
- No faceting
  1. 근접 비교를 하고 싶을 때
  2. Reader들의 사전 지식 수준이 높을때
- Presentation flow
  1. No facet → Yes facet 순서로
  2. 큰그림 → 작은 그림
  3. Top-down approach
  4. **`ggplot2`** supports such subsequential implementation just like addition!

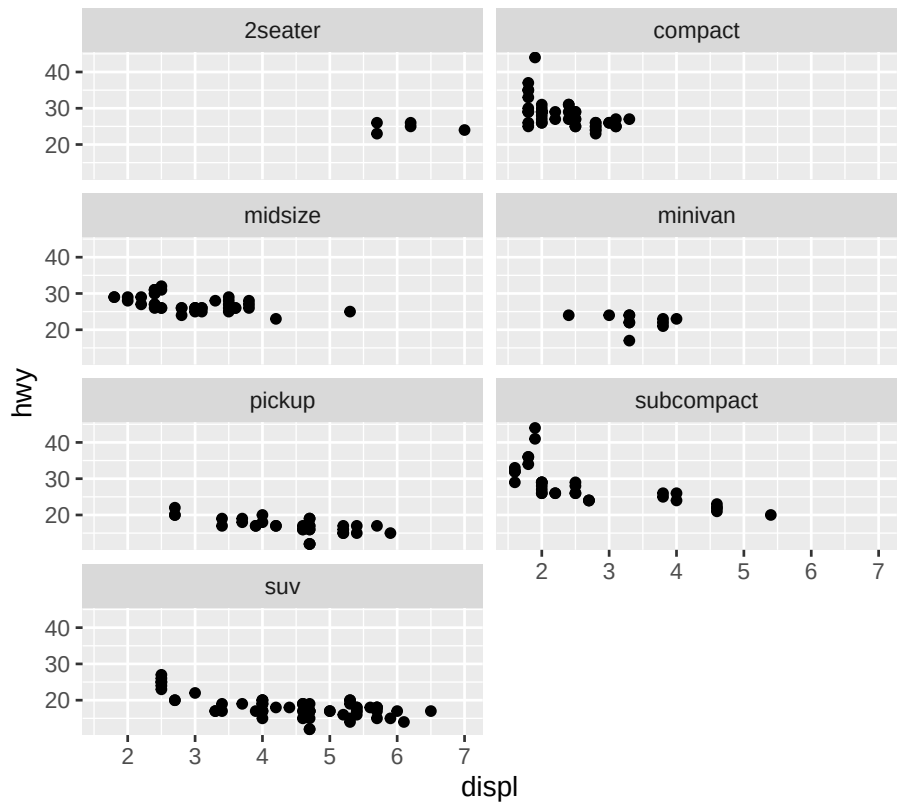
Implementation (1)

```
fig1 <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

```
fig1
```



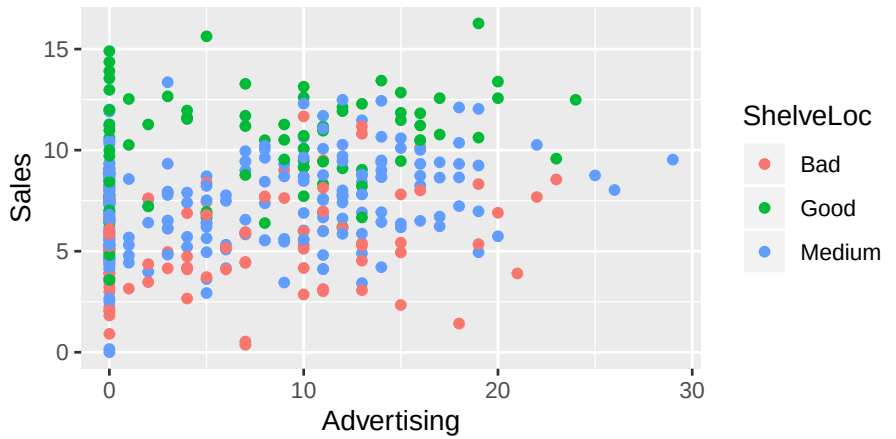

```
# Add a feature to an existing ggplot
#   object by "SIMPLY ADDING"
#   the code for the feature!
fig1 + facet_wrap(~ class, ncol=2)
```



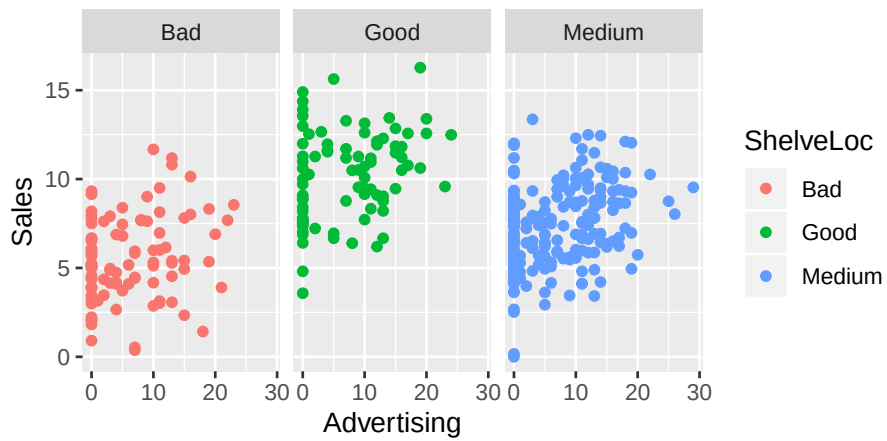
Implementation (2)

```
library(ISLR)
fig2 <- ggplot(data = Carseats, aes(x = Advertising, y = Sales)) +
  geom_point(aes(color = ShelveLoc))
```

```
fig2
```



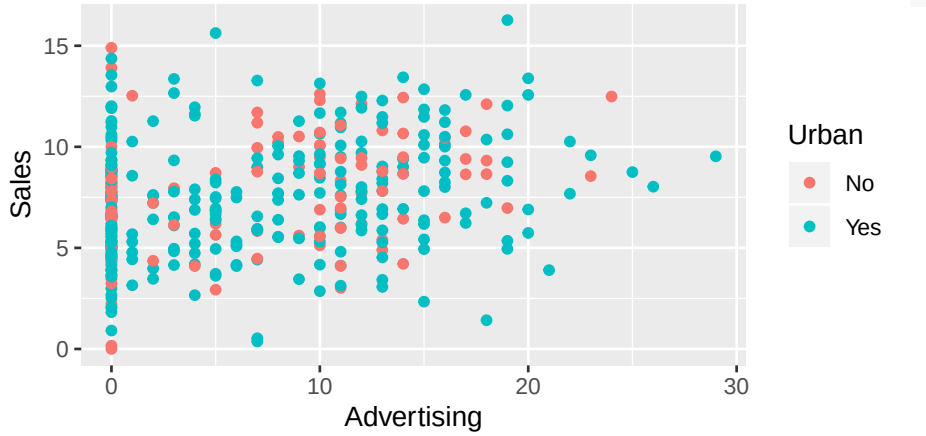

```
# Add a feature to an existing ggplot
#   object by "SIMPLY ADDING"
#   the code for the feature!
fig3 <- fig2 + facet_wrap(~ ShelveLoc)
fig3
```



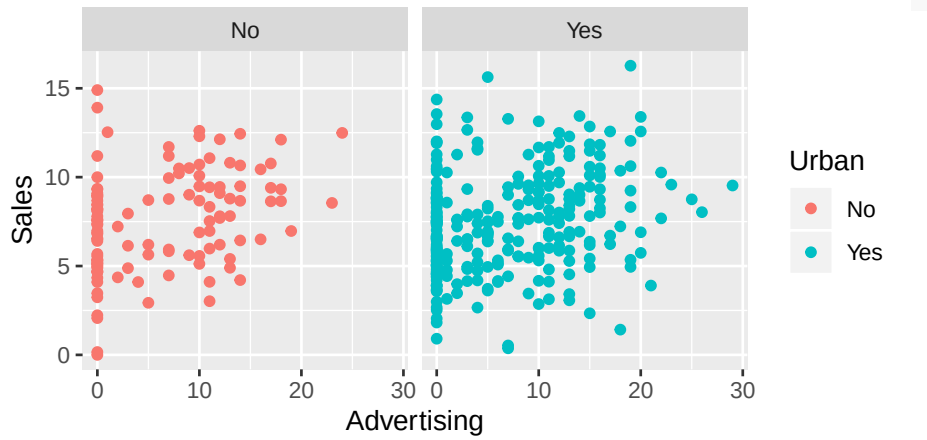
Implementation (3)

```
fig4 <- ggplot(data = Carseats, aes(x = Advertising, y = Sales)) +
  geom_point(aes(color = Urban))
```

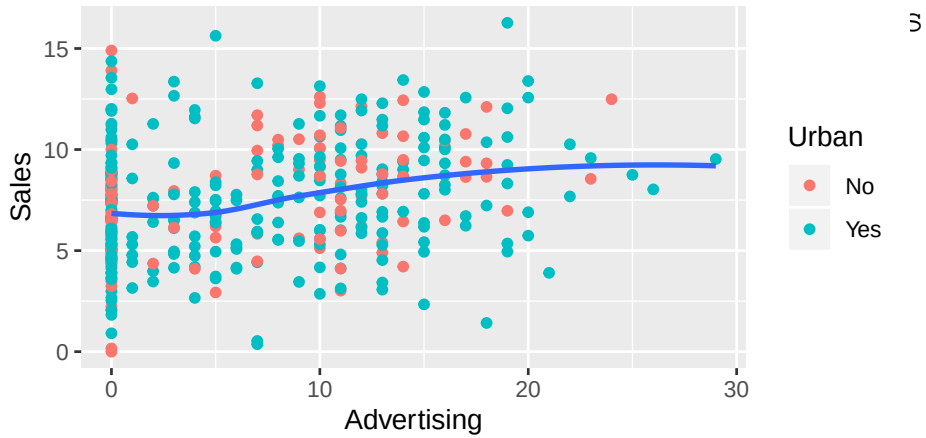
```
fig4
```



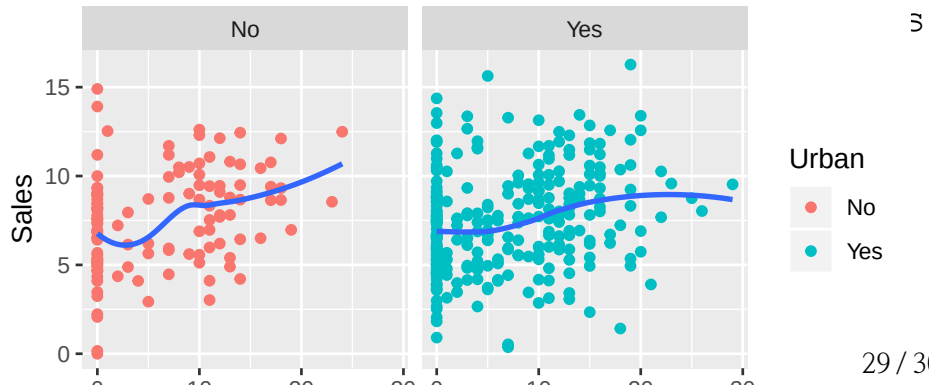

```
fig4 + facet_wrap(~ Urban)
```




```
fig4 + geom_smooth(se=FALSE)
```




```
fig4 + facet_wrap(~ Urban) +
  geom_smooth(se=FALSE)
```



```
## The simple graph has brought more information to the data analyst's mind
##  than any other device. - John Tukey
```