

L08. *ggplot* (2)

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



서울과학기술대학교 데이터사이언스학과

- ① I. 중첩
- ② II. Grammar of Graphics
- ③ Miscellaneous

Section 1

I. 중첩

geom_point() vs *geom_smooth()*

```
fig1 <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))
fig2 <- ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ, y = hwy))
```

fig1

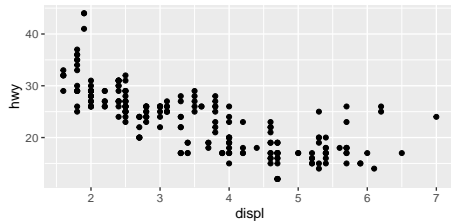
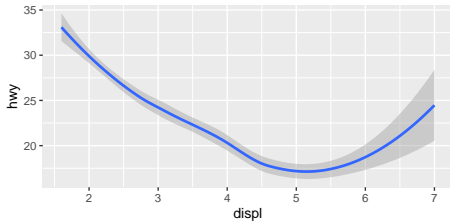


fig2



- *geom_point()*: 점으로 표시
- *geom_smooth()*: curve로 fitting하여 표시

mapping의 상속

```
fig3 <- ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
fig4 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth()
```

fig3

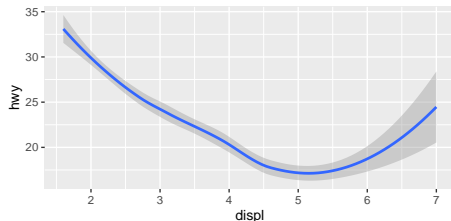
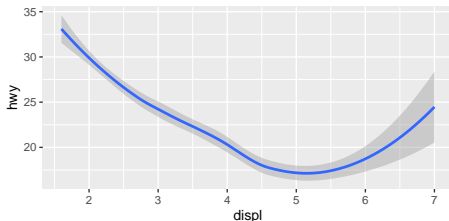


fig4



- 명령어의 variation이 있지만 결과는 동일함
- fig4에서는 ggplot() 함수에 포함된 mapping이라는 객체가 아래의 geom_smooth() 함수의 인수로 상속되었음
- fig4의 geom_smooth()에서 mapping이 없음
- → 바로 앞의 ggplot()의 값을 사용

`geom_point() + geom_smooth()`

```
fig5 <- ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))  
fig6 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point() + geom_smooth()
```

fig5

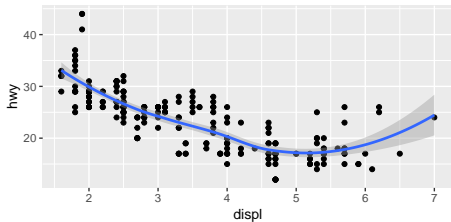
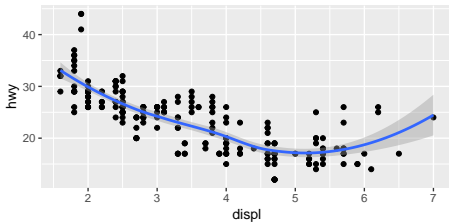


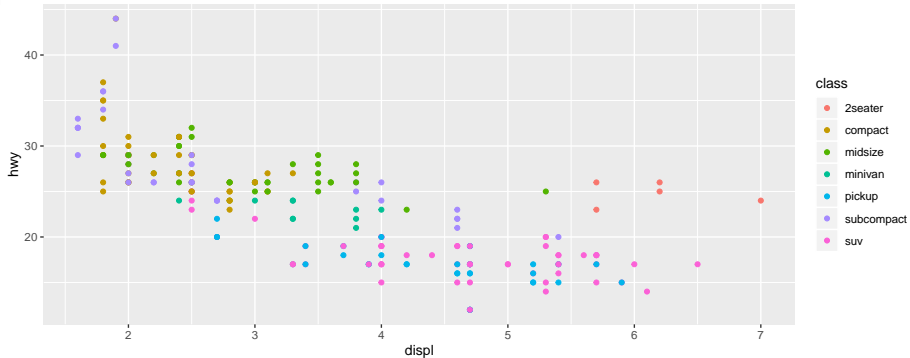
fig6



- fig5와 fig6는 동일한 객체이지만,
- fig5 객체는
 - ▶ mapping을 상속 기능을 이용해서 코드가 간결함
 - ▶ geometric object를 추가하기 용이함

Geometric object 별로 다른 종류의 *mapping*을 추가 (1)

```
fig7 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))
fig7
```

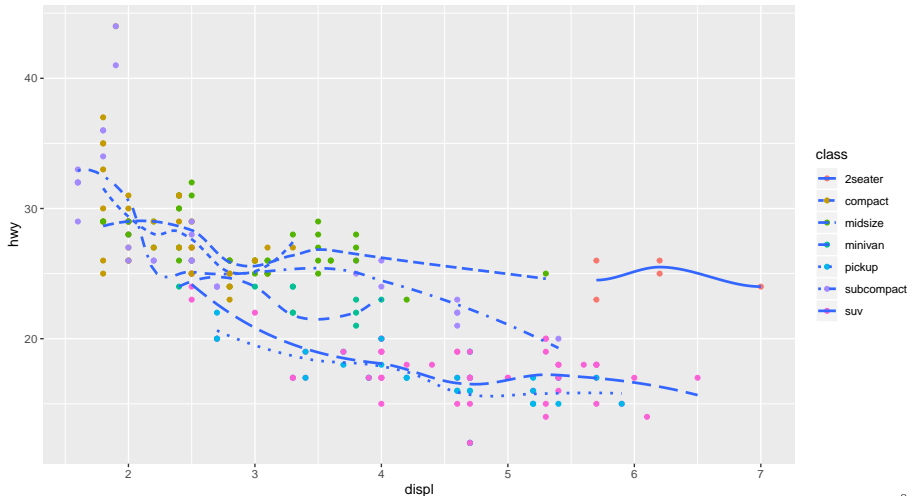


- `geom_point()` 의 mapping

- ▶ 부모로부터 상속받은 `aes(x = displ, y = hwy)`
- ▶ 스스로 가지고 있는 `aes(color=class)`

Geometric object 별로 다른 종류의 *mapping*을 추가 (2)

```
fig8 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color=class)) +
  geom_smooth(aes(linetype = class), se = FALSE)
fig8
```




```
fig7 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))
fig8 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(aes(linetype = class), se = FALSE)
```

- `geom_point()` 의 mapping

- ▶ 부모로부터 상속받은 `aes(x = displ, y = hwy)`
- ▶ 스스로 가지고 있는 `aes(color=class)`
- ▶ 형제(`geom_smooth()`)의 `aes(linetype=class)`는 가지지 않는다.
- ▶ `aes(linetype=class)`는 `geom_point()`와 무시당한다.

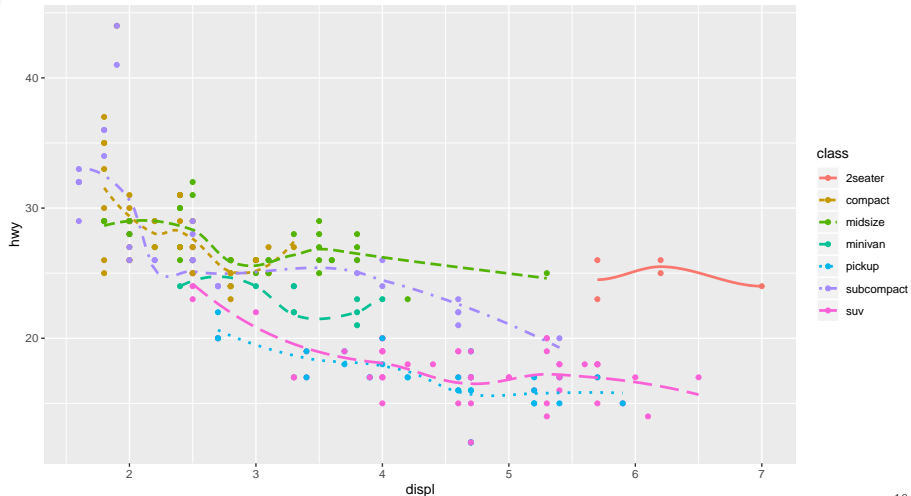
- `geom_smooth()` 의 mapping

- ▶ 부모로부터 상속받은 `aes(x = displ, y = hwy)`
- ▶ 스스로 가지고 있는 `aes(linetype=class)`
- ▶ 형제(`geom_point()`)의 `aes(color=class)`는 가지지 않는다.
- ▶ `aes(color=class)`는 `geom_smooth()`에 사용될 수 있으며, 실제로는 더 보기 좋다!
- ▶ → 그렇다면 아래와 같이 `color=class`는 형제 모두가 사용하므로 부모가 가지는 것이 바람직하다.

```
fig9 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=class)) +
  geom_point() +
  geom_smooth(aes(linetype = class), se = FALSE)
```

Perhaps the best one so far

```
fig9 <- ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=class)) +  
  geom_point() +  
  geom_smooth(aes(linetype = class), se = FALSE)  
fig9
```



Section 2

II. Grammar of Graphics

GG = Grammar of Graphics

- Motivation

- ① 그래픽에 관련된 함수와 패키지는 특수 경우의 모음일 뿐일까?
- ② 체계적인 문법이 있다면 체계적인 그래픽 구성이 가능하다!

- Features

- ① 독립적이고 더할 수 있는 구성 요소들로 그래픽을 구성하여 iterative updating이 가능
- ② 연구자의 생각의 흐름 = 그래픽을 발전시켜가는 코딩 과정
- ③ 스토리텔링 방식으로 효과적인 interactive graphics 구성

Revisit 'Implementation (3)' from the previous lecture note

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

fig

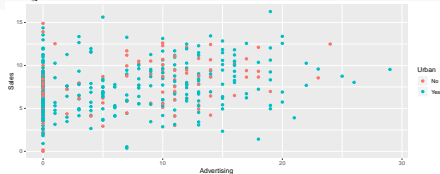


fig + facet_wrap(~ Urban)

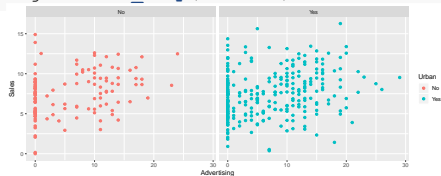
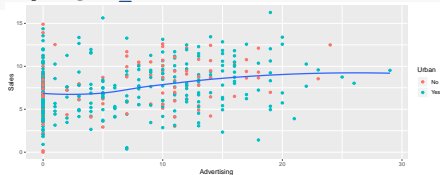
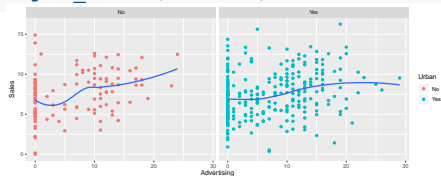


fig + geom_smooth(se=FALSE)

fig + facet_wrap(~ Urban) +
geom_smooth(se=FALSE)

Interactive graphics?

- Link: The interactive graphic for this example
- Will cover in the second half of this course.

그래픽의 구성 요소

- Skeleton of the syntax

```
ggplot(data = <DATA>) + <GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

- A minimal example

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

- ① Data: tidy dataset (각 컬럼이 변수를 의미)
- ② Aesthetics mapping
 - ▶ aes() - 각 변수를 aesthetic attribute에 어떻게 mapping할지 결정
 - ▶ x, y, position, size, color, shape, linetype,...
- ③ Geometric object
 - ▶ 실제로 플랏에 보게될 것을 결정 (points, lines, polygons, and etc.)
 - ▶ 차트의 종류가 결정됨 (산점도, 막대그래프 등)

Behind the scene

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

data = mpg

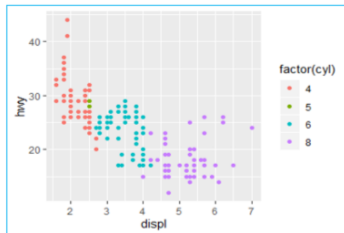
```
> mpg[,c("displ", "hwy", "cyl")] %>% head(8)
# A tibble: 8 x 3
  displ hwy  cyl
<dbl> <int> <int>
1  1.8  29    4
2  1.8  29    4
3  2.0  31    4
4  2.0  30    4
5  2.8  26    6
6  2.8  26    6
7  3.1  27    6
8  1.8  26    4
```

```
mapping = aes(x = displ, y = hwy, color = factor(cyl))
```

x	y	colour
1.8	29	4
1.8	29	4
2.0	31	4
2.0	30	4
2.8	26	6
2.8	26	6
3.1	27	6
1.8	26	4

geom_point()

x	y	colour	size	shape
0.037	0.531	#F8766D	1	19
0.037	0.531	#F8766D	1	19
0.074	0.594	#F8766D	1	19
0.074	0.562	#F8766D	1	19
0.222	0.438	#00BFC4	1	19
0.222	0.438	#00BFC4	1	19
0.278	0.469	#00BFC4	1	19
0.037	0.438	#F8766D	1	19



Section 3

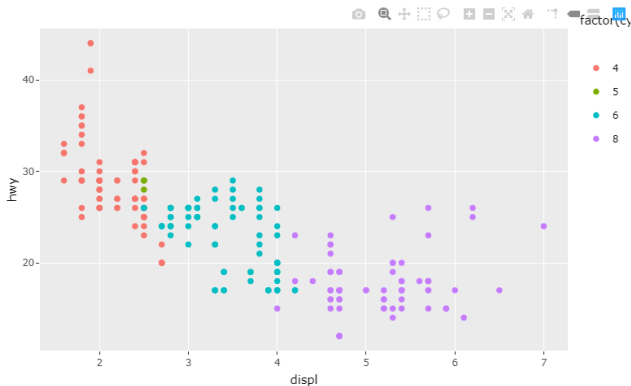
Miscellaneous

Export to a image file (*.eps*, *.png*, *.pdf*, *.tiff*, etc.)

```
ggsave(filename = "fig.png", plot = fig, dpi = 300,
        width = 8, height = 5) # cf) golden ratio: 1.62
```

Transform to *htmlwidget* (interactive graphic)

```
library(plotly)
ggplotly(fig)
```



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