

## R07. ggplot (1)

### Basic commands

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



서울과학기술대학교 데이터사이언스학과

## 1 I. Getting started with *ggplot2* and *mpg*

## 2 II. Scatterplot for *displ* and *hwy*

## 3 III. Faceting (분할 배열)

## I. Getting started with *ggplot2* and *mpg*

## *ggplot2* 라이브러리

### About

- Data visualization package for R, created by Hadley Wickham in 2005.
- *ggplot2* is an implementation of Leland Wilkinson's Grammar of Graphics, whose scheme breaks up graphs into semantic components such as scales and layers.

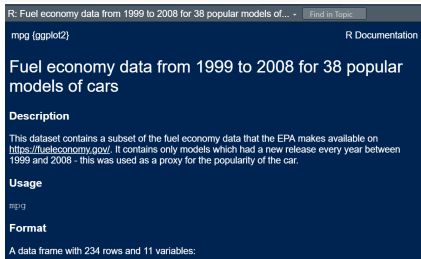
```
install.packages("ggplot2")  
library(ggplot2)
```

## *mpg* 데이터셋

### About

- *ggplot2* 패키지에 내장된 데이터셋
- 1999년과 2008년에 생산된 38개 차량모델들의 연비 데이터
- 데이터셋 *diamond*, *iris*와 함께 R에서 가장 많이 예제로 쓰임
- 아래의 명령어로 built-in documentation 조회 가능

```
? mpg  
help(mpg)
```



R: Fuel economy data from 1999 to 2008 for 38 popular models of...

`mpg` (ggplot2) R Documentation

### Fuel economy data from 1999 to 2008 for 38 popular models of cars

**Description**

This dataset contains a subset of the fuel economy data that the EPA makes available on <https://fueleconomy.gov/>. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

**Usage**

```
mpg
```

**Format**

A data frame with 234 rows and 11 variables:

## Overview

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
## $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
## $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
## $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
## $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
## $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
## $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
## $ drv         : chr [1:234] "f" "f" "f" "f" ...
## $ cty         : int [1:234] 18 21 20 21 16 18 18 16 20 ...
## $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
## $ fl         : chr [1:234] "p" "p" "p" "p" ...
## $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

## Variables

Name	Description	Type
manufacturer	제조사	chr
model	차종	chr
displ	엔진크기	num
year	생산연도	int
cyl	기통	int
trans	트랜스미션	chr
drv	전륜(f)/후륜(r)/사륜(4)	chr
cty	도심 연비	int
hwy	고속도로 연비	int
fl	연료종류 (fuel)	chr
class	타입 (세단, 트럭, SUV, ...)	chr

## II. Scatterplot for *displ* and *hwy*



## Motivation

- Q. 엔진의 용량과 연비는 어떤 관계가 있는가?
- 엔진용량(*displ*)과 고속도로연비(*hwy*)는 각각 *num*(수치형)과 *int*(정수형)의 데이터 타입을 가지고 있다.
- 둘 다 수치변수(사칙연산이 가능함)이므로 산점도를 그려서 조사해보자.
  - 원인에 해당하는 엔진크기(*displ*)를 X축
  - 결과에 해당하는 고속도로연비(*hwy*)를 Y축

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

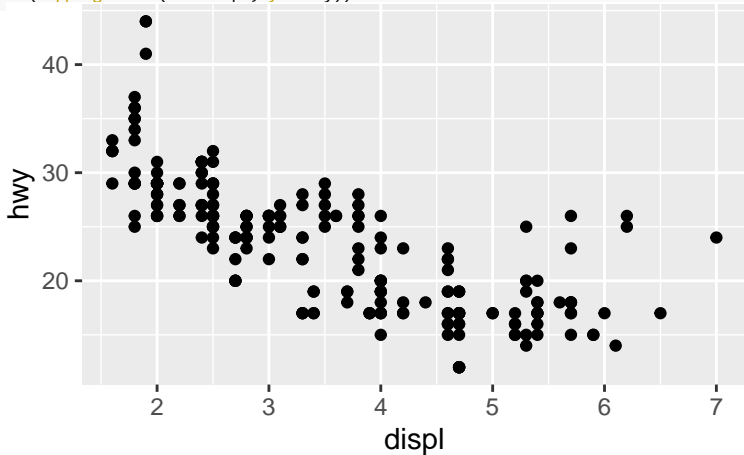


그림 1: 엔진의 크기와 연비의 관계

- 위의 그림으로 보아 엔진의 용량과 연비는 어떤 관계가 있는가?

## Basic Syntax

- 위의 플랏을 그리기 위해 아래의 명령어가 사용되었다.

*# The plot above*

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

- 가장 minimal한 ggplot2의 명령어는 아래와 같다.
  - <DATA> - 데이터프레임을 지정한다.
  - <GEOM\_FUNCTION> - 데이터를 어떤 모양으로 표현할 것인지를 결정한다.
    - ex) point, line, curve, bar, histogram, text, label, and so on
  - <MAPPING> - 변수를 시각요소(aesthetic)에 대응시킨다.
    - ex) position(x, y), size, line width, color, line type, and so on.

*# The basic ggplot syntax*

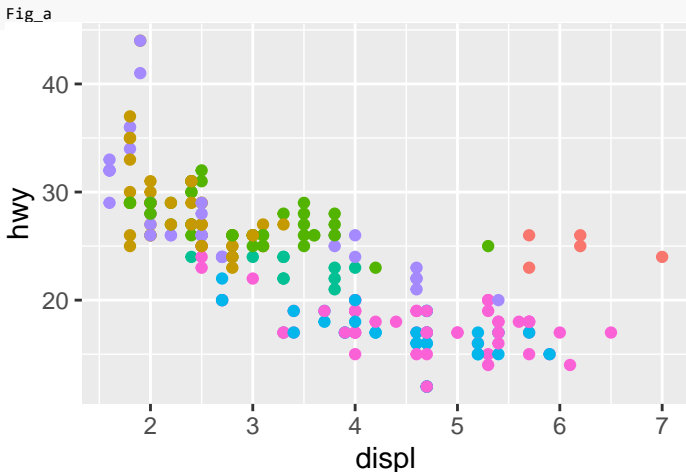
```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

## Adding the variable *class* to the scatterplot.

- 앞의 산점도에 *class* 변수(차량의 타입)를 추가하여 다시 그려보자.
- 몇 가지의 방법의 aesthetic mapping을 생각할 수 있다.
  1. Fig\_a: 점의 색깔을 *class* 값에 따라서 다르게 한다. (*color=class*)
  2. Fig\_b: 점의 크기를 *class* 값에 따라서 다르게 한다. (*size=class*)
  3. Fig\_c: 점의 진하기를 *class* 값에 따라서 다르게 한다. (*alpha=class*)
  4. Fig\_d: 점의 모양을 *class* 값에 따라서 다르게 한다. (*shape=class*)

## 1. Fig\_a: *color = class*

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

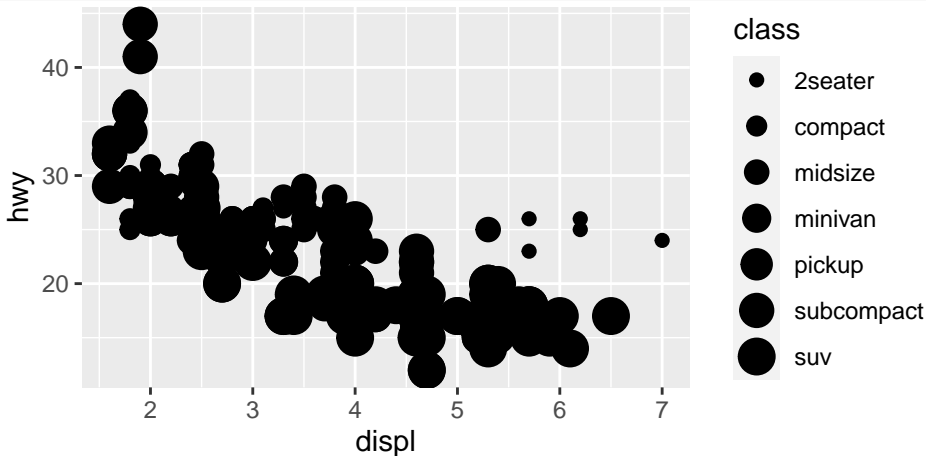


class

- 2seater
- compact
- midsize
- minivan
- pickup
- subcompact
- suv

## 2. Fig\_b: *size = class*

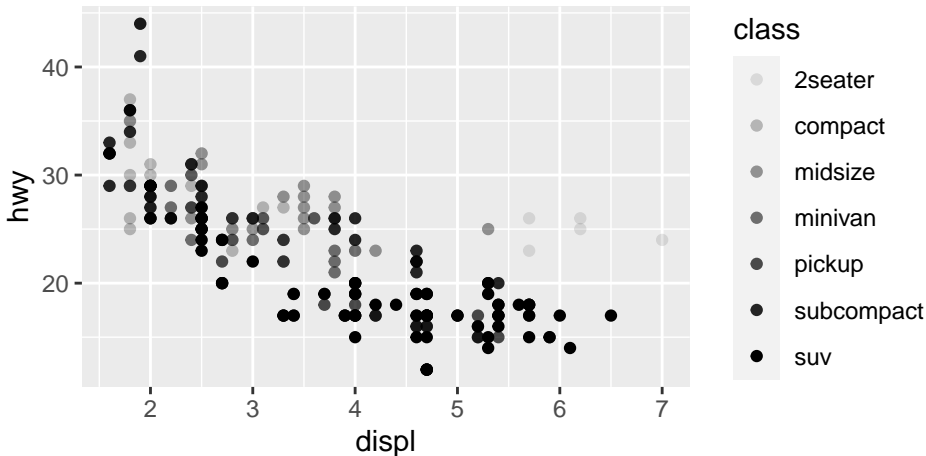
```
Fig_b <- ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, size = class))  
Fig_b
```



### 3. Fig\_c: *alpha* = *class*

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

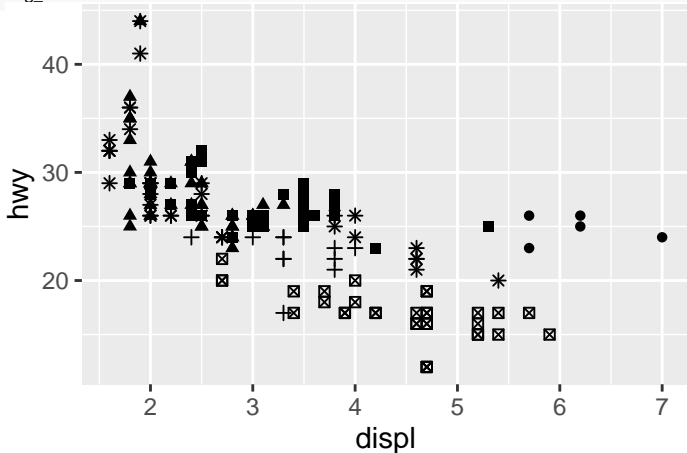
Fig\_c



## 4. Fig\_d: *shape = class*

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

Fig\_d



class

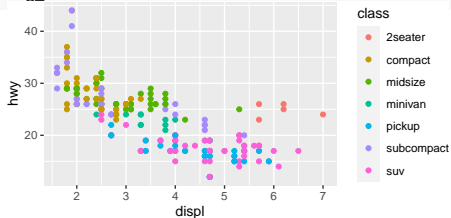
- 2seater
- ▲ compact
- midsize
- + minivan
- ⊠ pickup
- \* subcompact
- ◇ suv



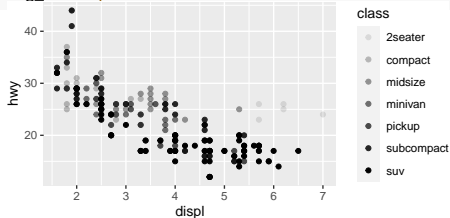
## Discussion on *Fig\_a*, *Fig\_b*, *Fig\_c*, and *Fig\_d*

- What is the best and why?

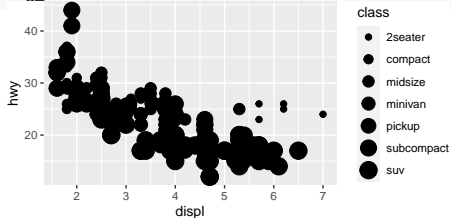
*Fig\_a* # *color = class*



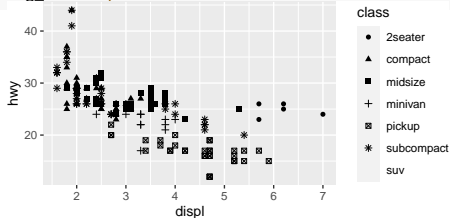
*Fig\_c* # *alpha=class*



*Fig\_b* # *size = class*



*Fig\_d* # *shape=class*



## Proper vs Improper aesthetic mapping

- 앞의 그림에는 아래와 같은 aesthetic mapping이 적용되었다.
  1. Fig\_a: 점의 색깔(color)이 *class* 값에 따라서 **달라진다**. (*color=class*)
  2. Fig\_b: 점의 크기(size)가 *class* 값에 따라서 **크고 작다**. (*size=class*)
  3. Fig\_c: 점의 진하기(alpha)가 *class* 값에 따라서 **진하고 옅다**. (*alpha=class*)
  4. Fig\_d: 점의 모양(shape)이 *class* 값에 따라서 **달라진다** (*shape=class*)
- *class*의 변수 값은 i) 달라지고, ii) 크고 작고, iii) 진하고 옅은 것에 연관시키는 것이 적합한가?
- *class* 변수는
  - ordered/unordered variable?
  - discrete/continuous variable?

## Basic rule for aesthetic mapping

	ordered/unordered	discrete/conti.	others
color	unordered	discrete	color display
size	ordered	.	.
alpha	ordered	.	bounded
shape	unordered	discrete	b/w display

- color나 shape를 사용할 때에는 **magic number 7**에 주의해야 한다. 아래의 대안을 고려할 수 있다.
  - further grouping
  - faceting
  - 다른 geometric object (line, smooth, path)
- alpha is more suitable than size if bounded.
  - 흔히 0과 1 사이의 값에 해당됨
  - 농도, 비율, 빈도, 확률, Intensity.

## R에서 제공하는 warning messages

- Fig\_b (`size=class`)
  - “Using size for a discrete variable is not advised.”
- Fig\_c (`alpha=class`)
  - “Using alpha for a discrete variable is not advised.”
- Fig\_d (`shape=class`)
  - 1: 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.
  - 2: Removed 62 rows containing missing values (`geom_point`).

## The best so far?

```
```{r, out.height='60%', fig.asp=0.6, fig.width=5, fig.cap="The best so far?"}  
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color=class, shape=class))  
```
```

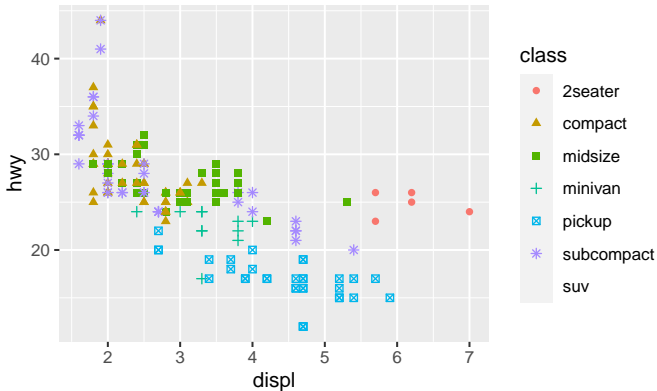


그림 2: The best so far?

## Chunk figuration for rmd output

- **fig.width** and **fig.height**
  - The size of R plots in inches.
  - Alternatively, 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.
  - e.g., `out.width = '80%'` means 80% of the page width.
- **fig.asp**
  - 1.6 or 0.625 for “golden ratio”.
- **fig.align**
  - The alignment of plots, which can be 'left', 'center', or 'right'.
- **fig.cap**
  - The figure caption

## Review

```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))  
ggplot(data = <DATA>) + <GEOM_FUNCTION>(mapping = aes(<MAPPING>))
```

```
Fig_a <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = class))  
Fig_b <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = class))  
Fig_c <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, alpha = class))  
Fig_d <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = class))
```

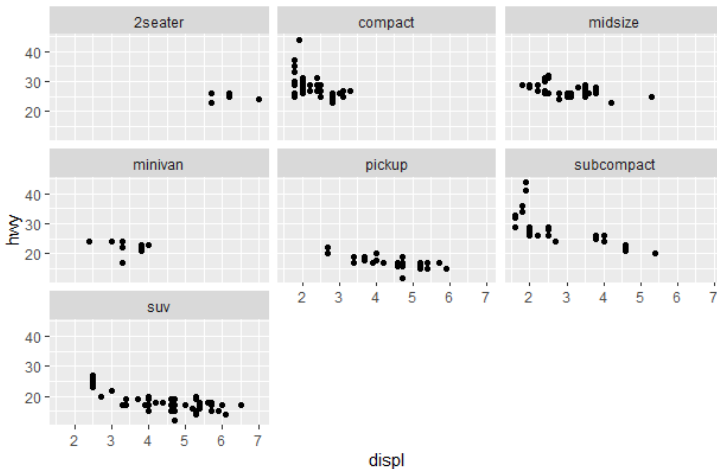
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color=class, shape=class))
```

### III. Faceting (분할 배열)



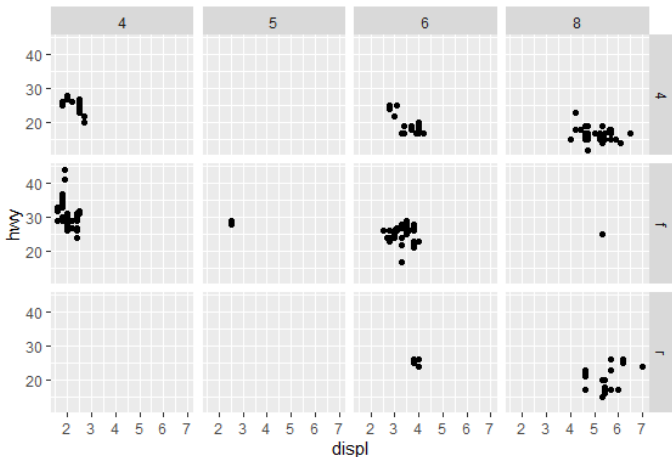
## *facet\_wrap()* to add one dimension

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



## *facet\_grid()* to increment two dimensions

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



- *drv*를 y축(세로)으로 *cyl*을 x축(가로)으로 grid 배열

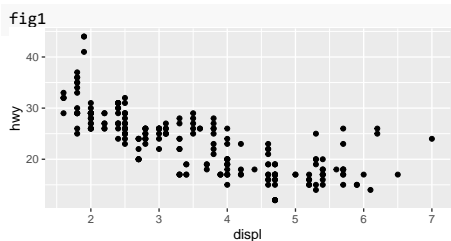
## Yes or No to faceting?

- Yes to 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. 전체 그림 → Zooming in
  2. No facet → Yes facet
  3. Top-down approach

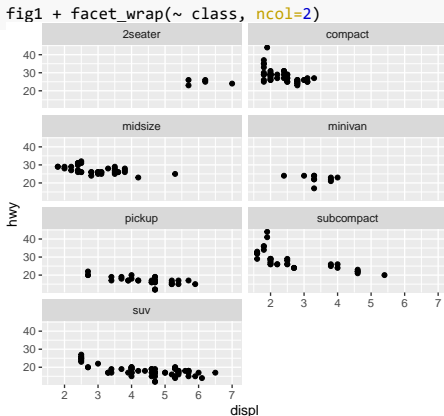
## Additive Implementation

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

### ● Original



### ● Add facet\_wrap()

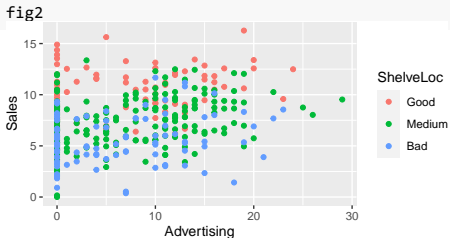


### ● 그림의 특징을 더하기 위해 코드를 더하는 방식으로 구현

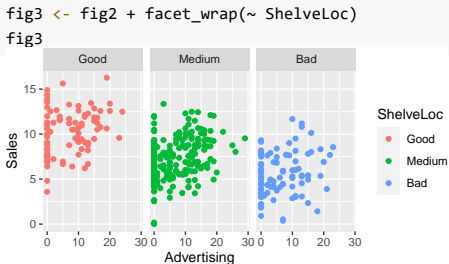
## Rearranging factor variable

```
library(ISLR)
fig2 <- ggplot(Carseats %>%
  mutate(ShelveLoc = factor(ShelveLoc, levels = c("Good", "Medium", "Bad"))) %>%
  arrange(ShelveLoc)) +
  geom_point(aes(x = Advertising, y = Sales, color = ShelveLoc))
```

### ● Original



### ● Add facet\_wrap()



## Versatile additivity

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

fig4

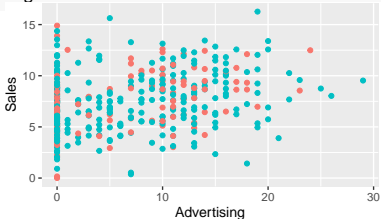


fig4 + facet\_wrap(~ Urban)

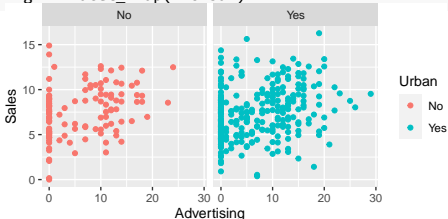


fig4 + geom\_smooth(se=FALSE)

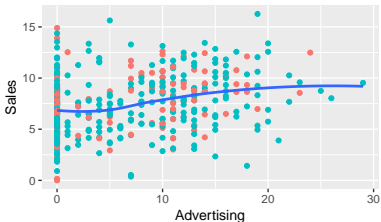
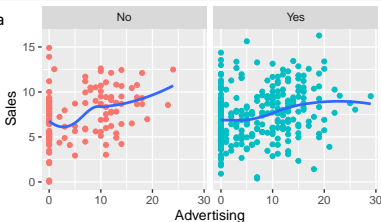


fig4 + facet\_wrap(~ Urban) + geom\_smooth(se=FALSE)

ula



ula

## Review

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_wrap(~ class)  
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_grid(drv ~ cyl)  
  
fig1 <- ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))  
fig1 + facet_wrap(~ class, ncol=2)  
fig2 <- ggplot(Carseats %>%  
  mutate(ShelveLoc = factor(ShelveLoc, levels = c("Good", "Medium", "Bad"))) %>%  
  arrange(ShelveLoc)) +  
  geom_point(aes(x = Advertising, y = Sales, color = ShelveLoc))  
fig3 <- fig2 + facet_wrap(~ ShelveLoc)  
fig4 <- ggplot(data = Carseats, aes(x = Advertising, y = Sales)) + geom_point(aes(color = Urban))  
fig4 + geom_smooth(se=FALSE)  
fig4 + facet_wrap(~ Urban)  
fig4 + facet_wrap(~ Urban) + geom_smooth(se=FALSE)
```

## Suggested Readings

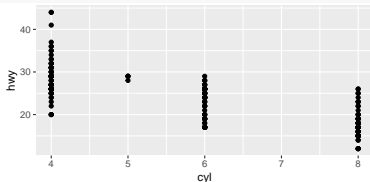
- R for Data Science
  - <https://r4ds.had.co.nz>
- [R4DS] Ch1. Data Visualization with *ggplot2*
  - <https://r4ds.had.co.nz/data-visualisation.html>



## Exercise

- In Ch1@R4DS, make a scatterplot of *hwy* vs *cyl* as below.

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



- Analyze the vis in terms of the three step framework.
- Discuss this vis critically. Why is this vis poor?
- Describe the better alternative graphic in your mind by the three step framework.
- Provide a hand drawing of the ideal plot accordingly.