Part 3. 데이터 시각화 (데이터 분석 전문가 양성과정)

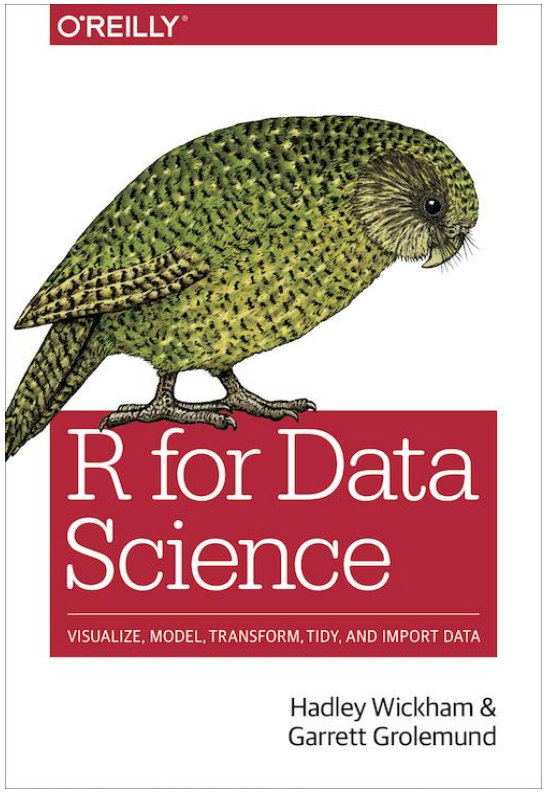
02

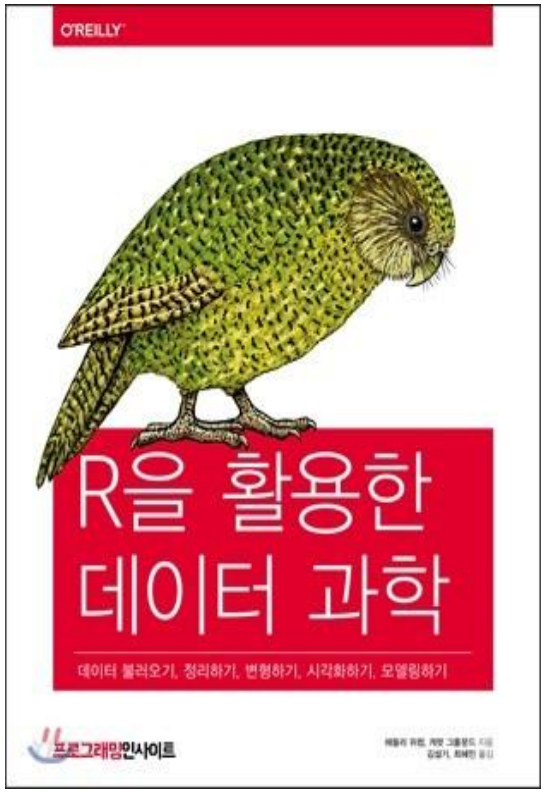
데이터 시각화 실습

경북대학교 배준현 교수
(joonion@knu.ac.kr)

02. 데이터 시각화 실습

- R for Data Science
  - **영문 버전**: https://r4ds.had.co.nz/
  - **한글 버전**: https://bookdown.org/sulgi/r4ds/



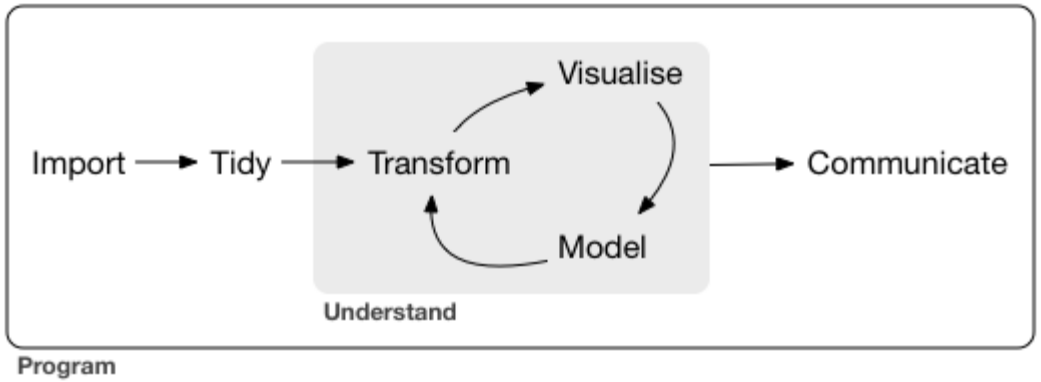


02. 데이터 시각화 실습

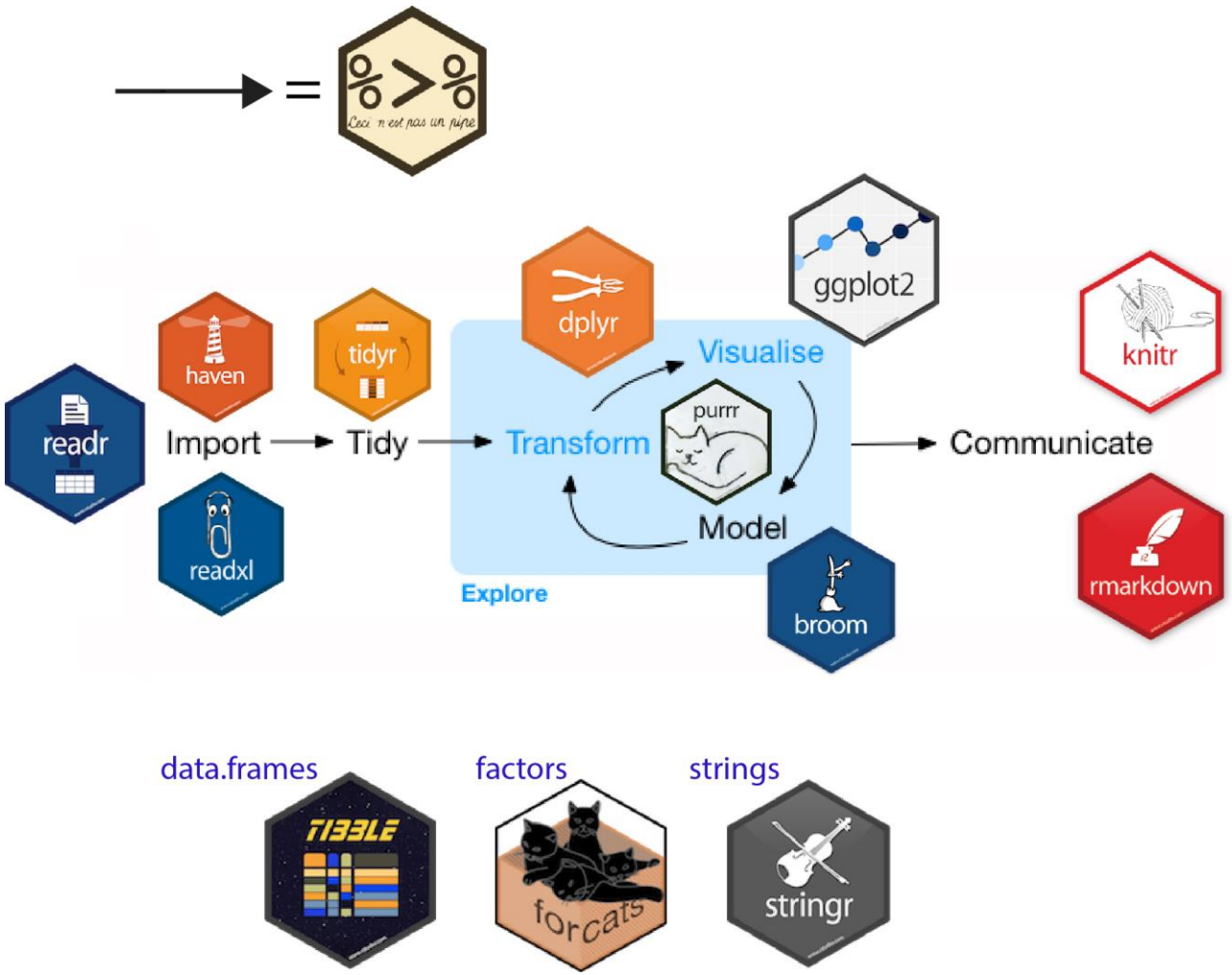
- tidyverse와 ggplot2
  - tidyverse: 해들리 위컴이 모아 놓은 데이터 과학을 위한 R 패키지들
  - ggplot2: tidyverse의 세계에서 시각화를 담당하는 패키지

```
> install.packages("tidyverse")
> library(tidyverse)
-- Attaching packages ----------- tidyverse 1.3.1 --
√ ggplot2 3.3.5     √ purrr   0.3.4
√ tibble  3.1.6     √ dplyr   1.0.7
√ tidyr   1.1.4     √ stringr 1.4.0
√ readr   2.1.1     √ forcats 0.5.1
-- Conflicts -------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

02. 데이터 시각화 실습


Visualise
Import
Tidy
Transform
Communicate
Model
Understand
Program

02. 데이터 시각화 실습


%>%
Ceci n'est pas un pipe
ggplot2
dplyr
tidyr
haven
readr
readxl
Import
Tidy
Visualise
purrr
Transform
Model
Explore
broom
knitr
Communicate
rmarkdown
data.frames
tibble
factors
forcats
strings
stringr

02. 데이터 시각화 실습

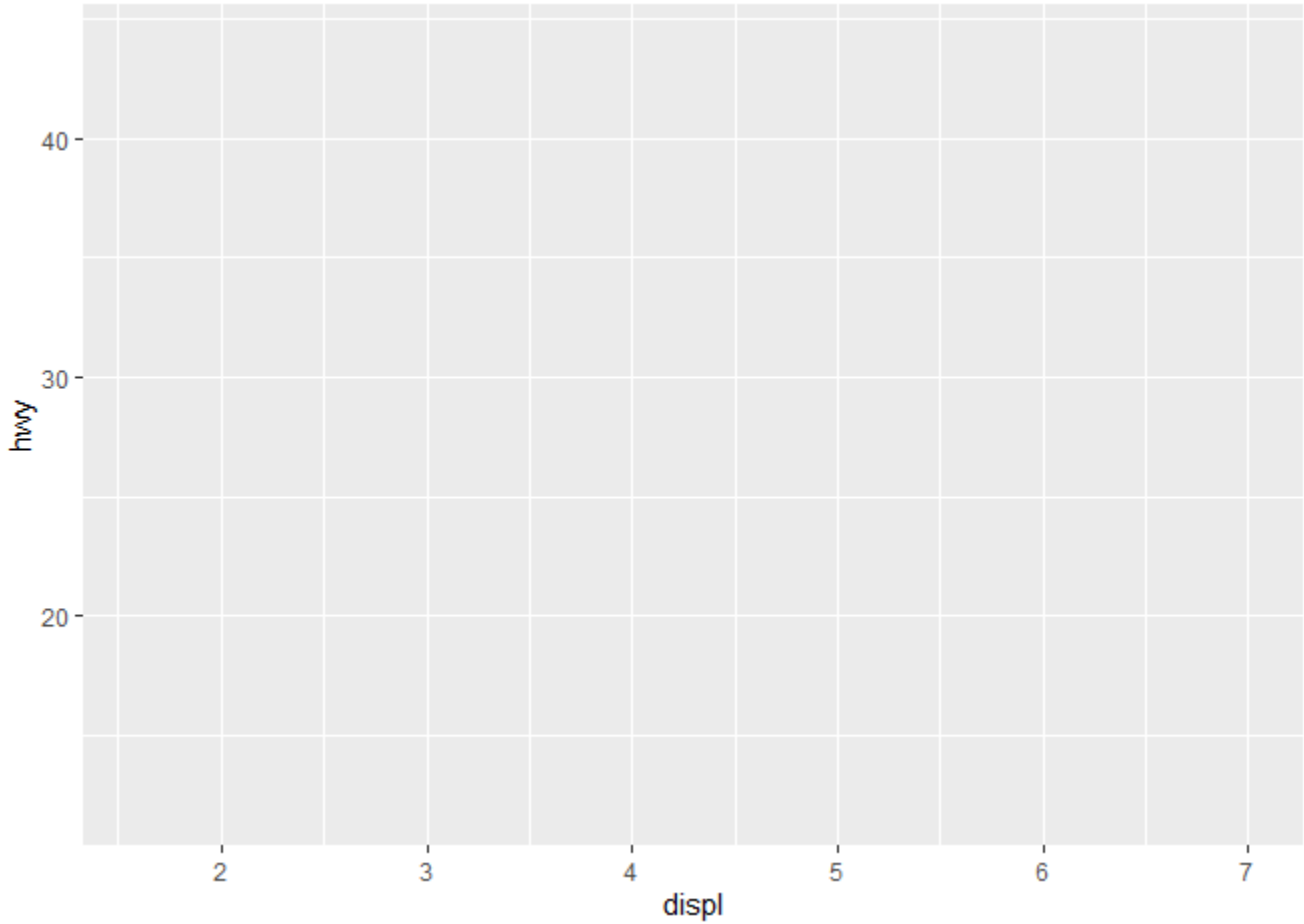
```
> str(mpg)
tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
 $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
 $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
 $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
 $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 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 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" ...
```

02. 데이터 시각화 실습

```
> head(mpg)
# A tibble: 6 x 11
  manufacturer model displ  year   cyl trans drv     cty   hwy fl
  <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr>
1 audi         a4      1.8  1999     4 auto~ f        18    29 p
2 audi         a4      1.8  1999     4 manu~ f        21    29 p
3 audi         a4      2    2008     4 manu~ f        20    31 p
4 audi         a4      2    2008     4 auto~ f        21    30 p
5 audi         a4      2.8  1999     6 auto~ f        16    26 p
6 audi         a4      2.8  1999     6 manu~ f        18    26 p
# ... with 1 more variable: class <chr>
```

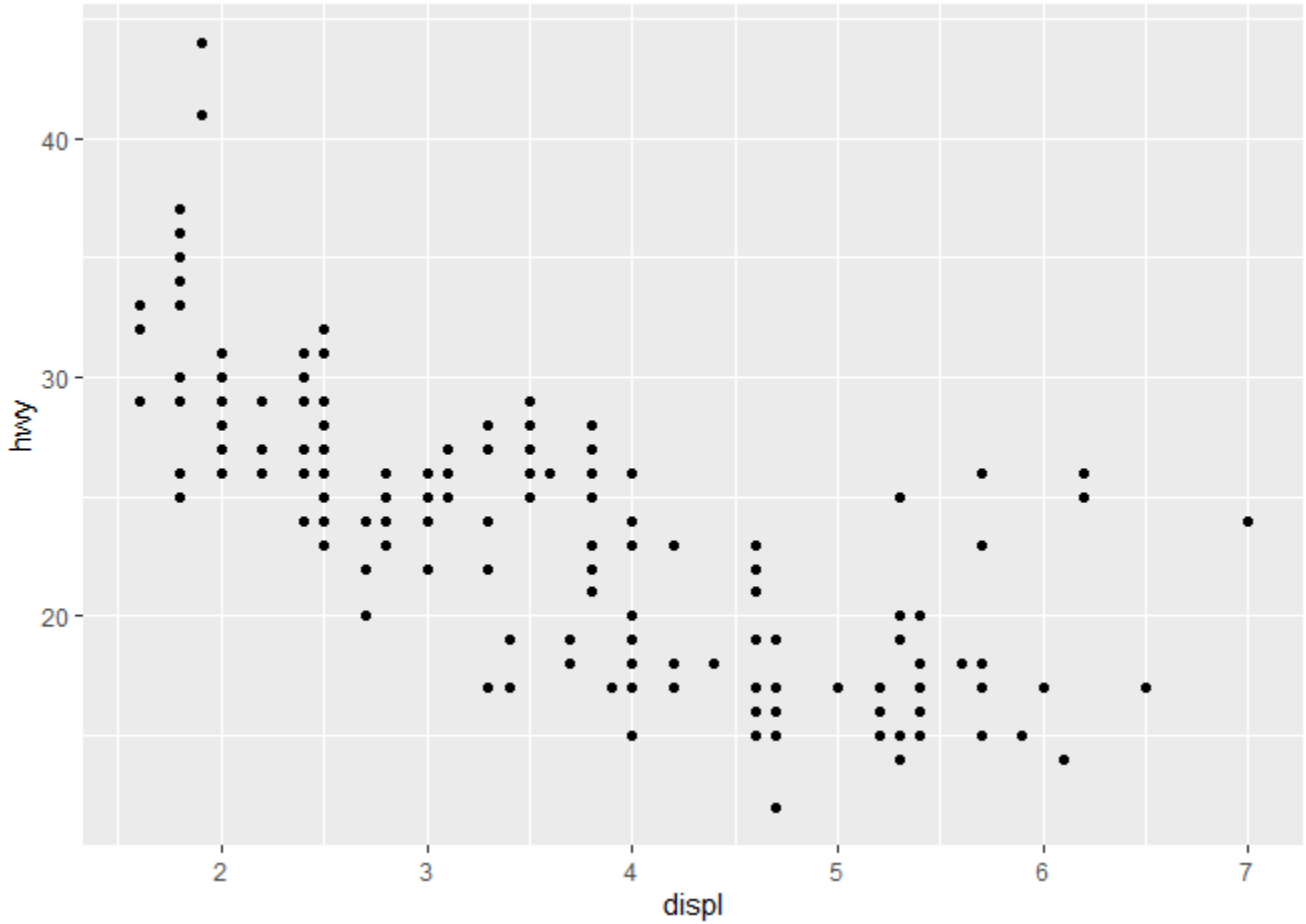
02. 데이터 시각화 실습

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



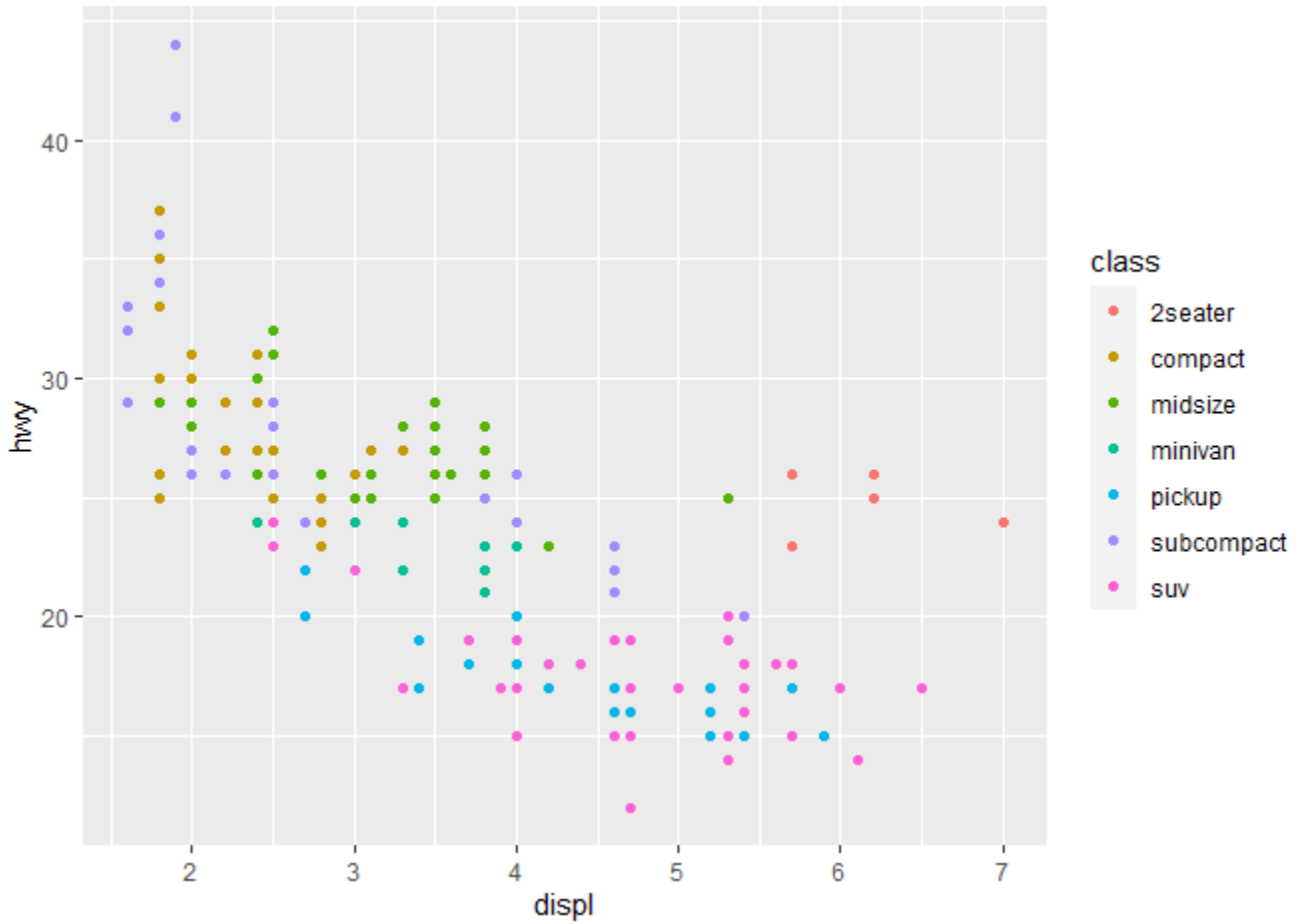
02. 데이터 시각화 실습

```
p + geom_point()
```



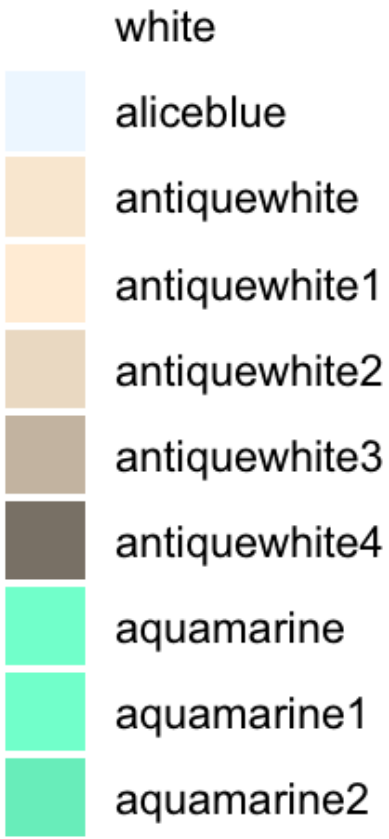
02. 데이터 시각화 실습

```
p + geom_point(mapping = aes(color = class))
```



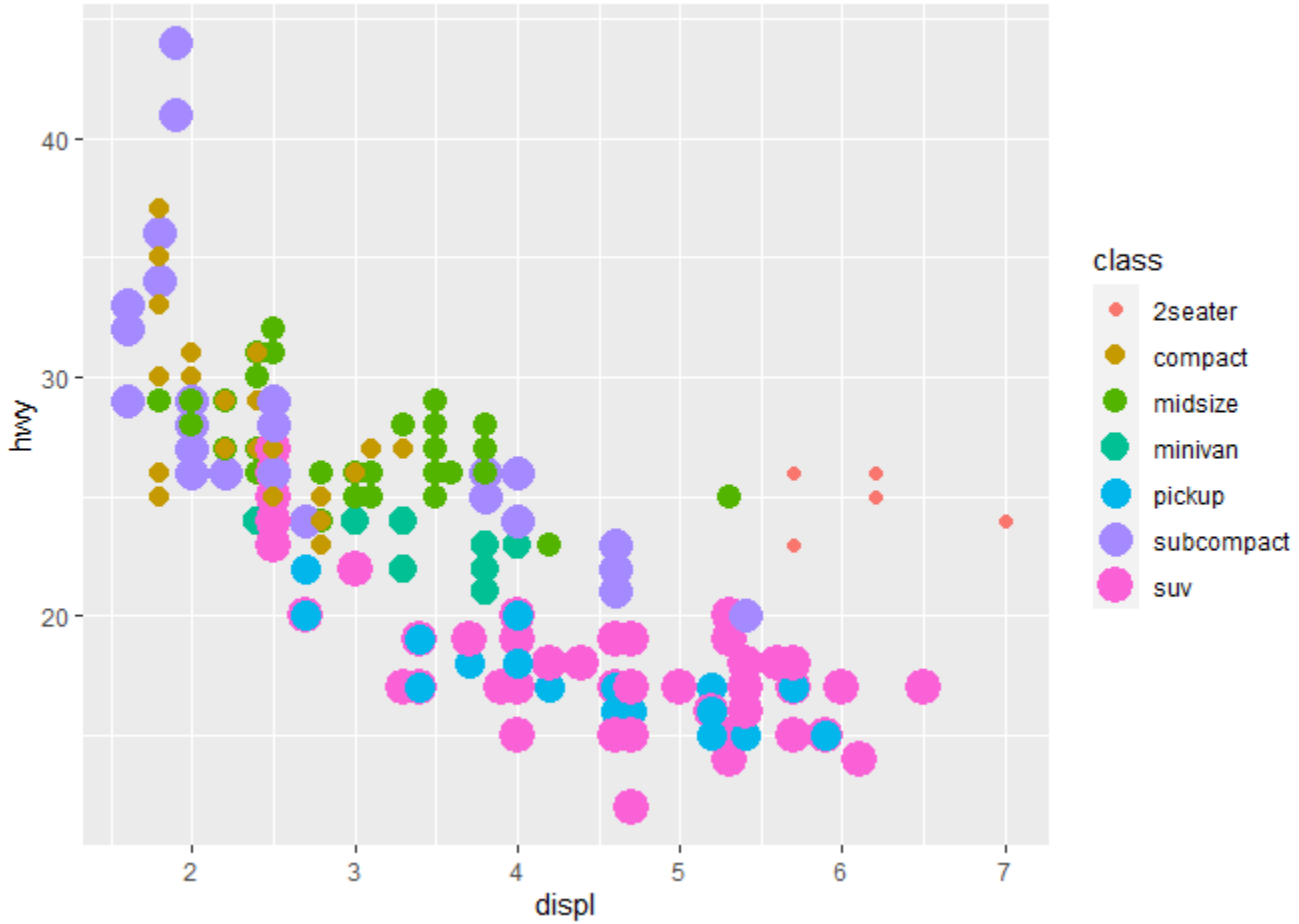
02. 데이터 시각화 실습

```
> head(colors(), 10)
 [1] "white"          "aliceblue"      "antiquewhite"   "antiquewhite1"
 [5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
 [9] "aquamarine1"   "aquamarine2"
```



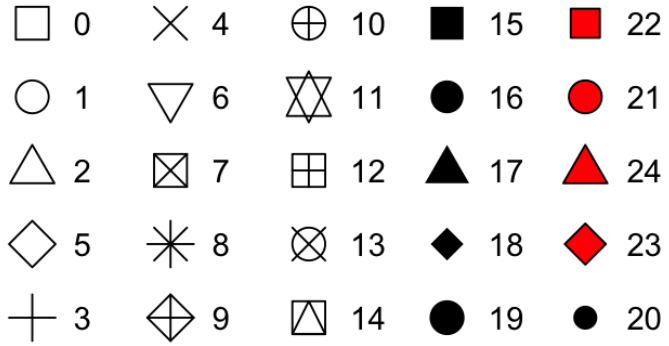
02. 데이터 시각화 실습

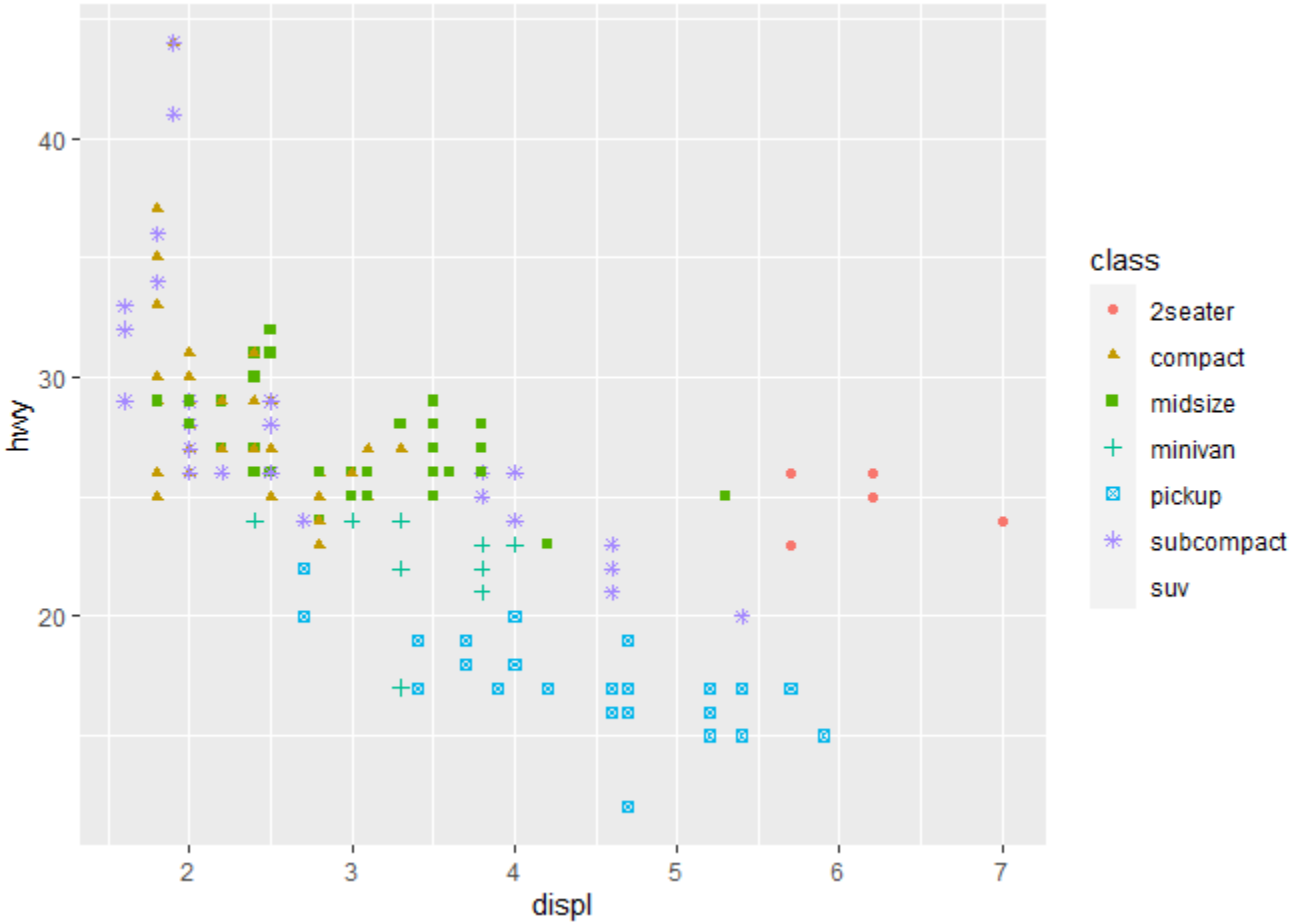
```
p + geom_point(mapping = aes(color = class,
                             size = class))
```



02. 데이터 시각화 실습

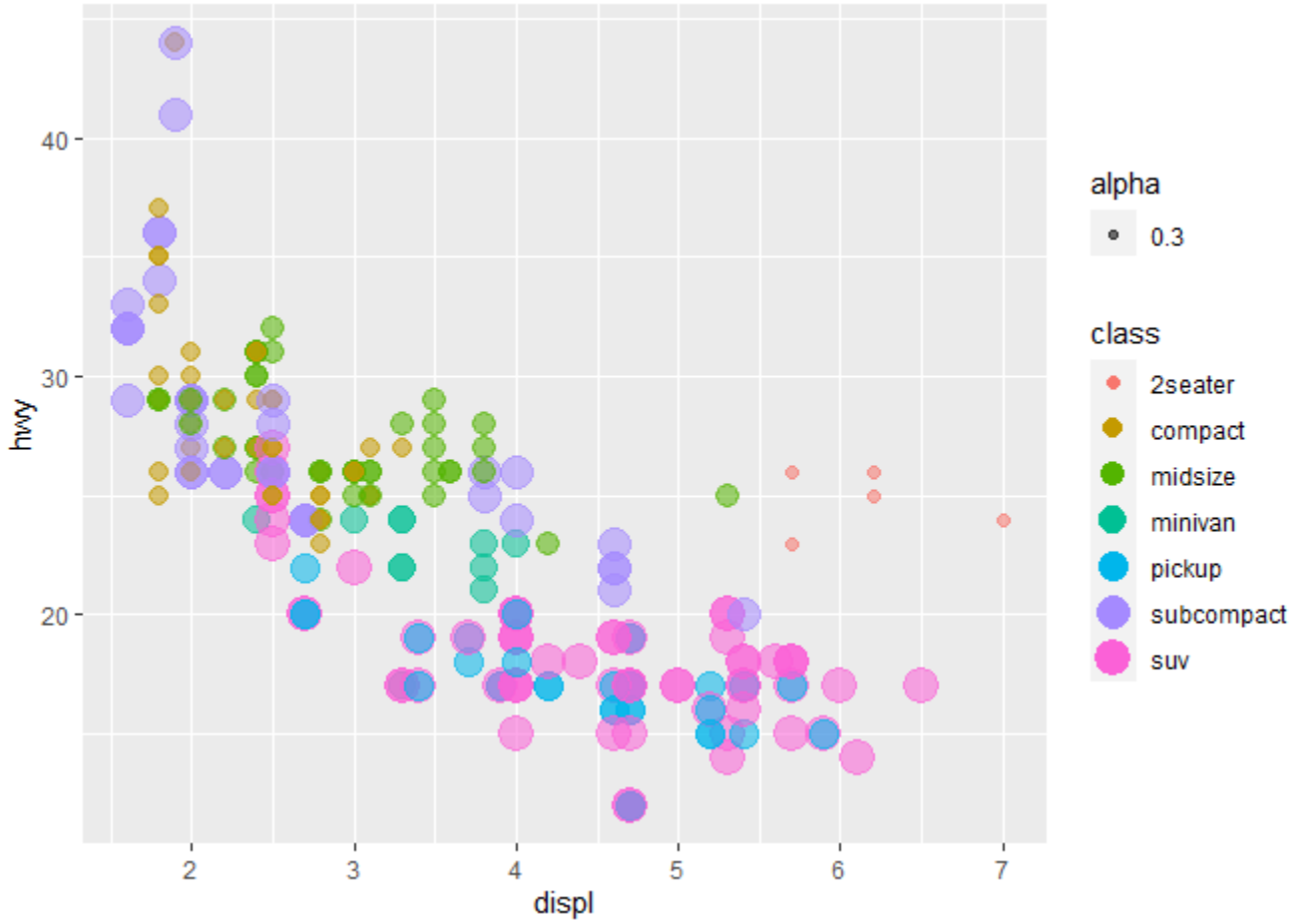
```
p + geom_point(mapping = aes(color = class,
                             shape = class))
```





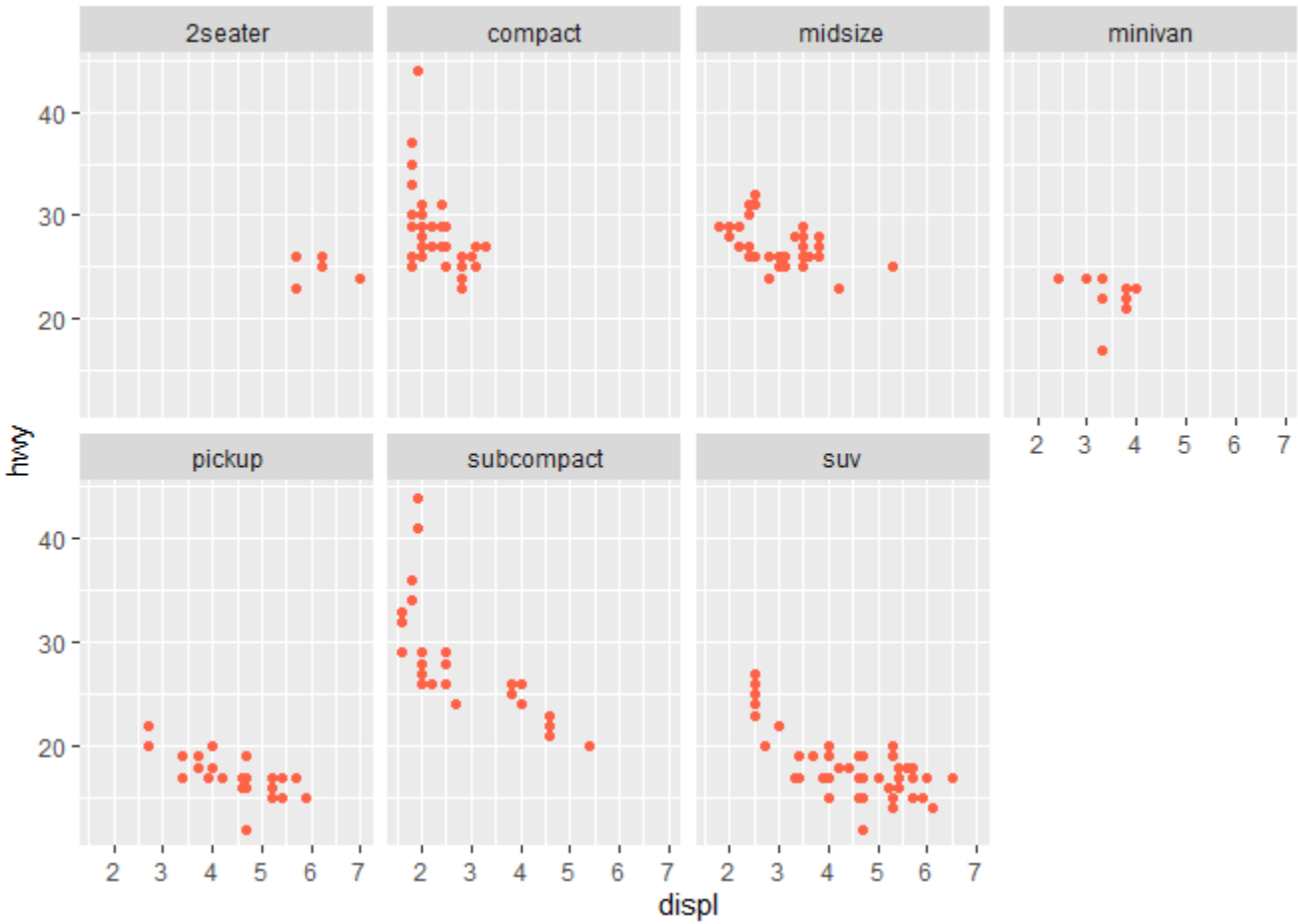
02. 데이터 시각화 실습

```
p + geom_point(mapping = aes(color = class, size = class,
                             alpha = 0.3))
```



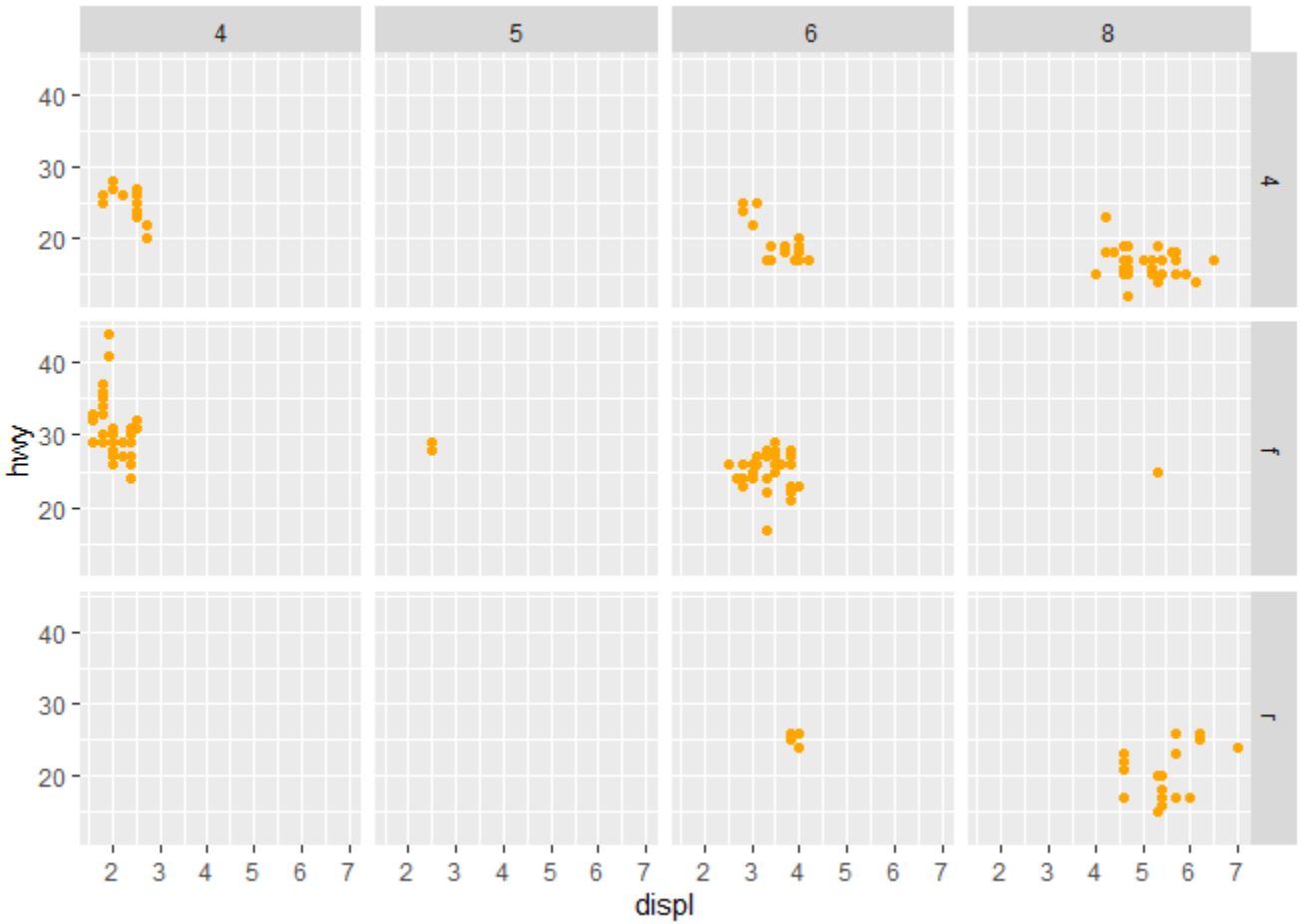
02. 데이터 시각화 실습

```
p + geom_point(color="tomato") +
    facet_wrap(~ class, nrow = 2)
```



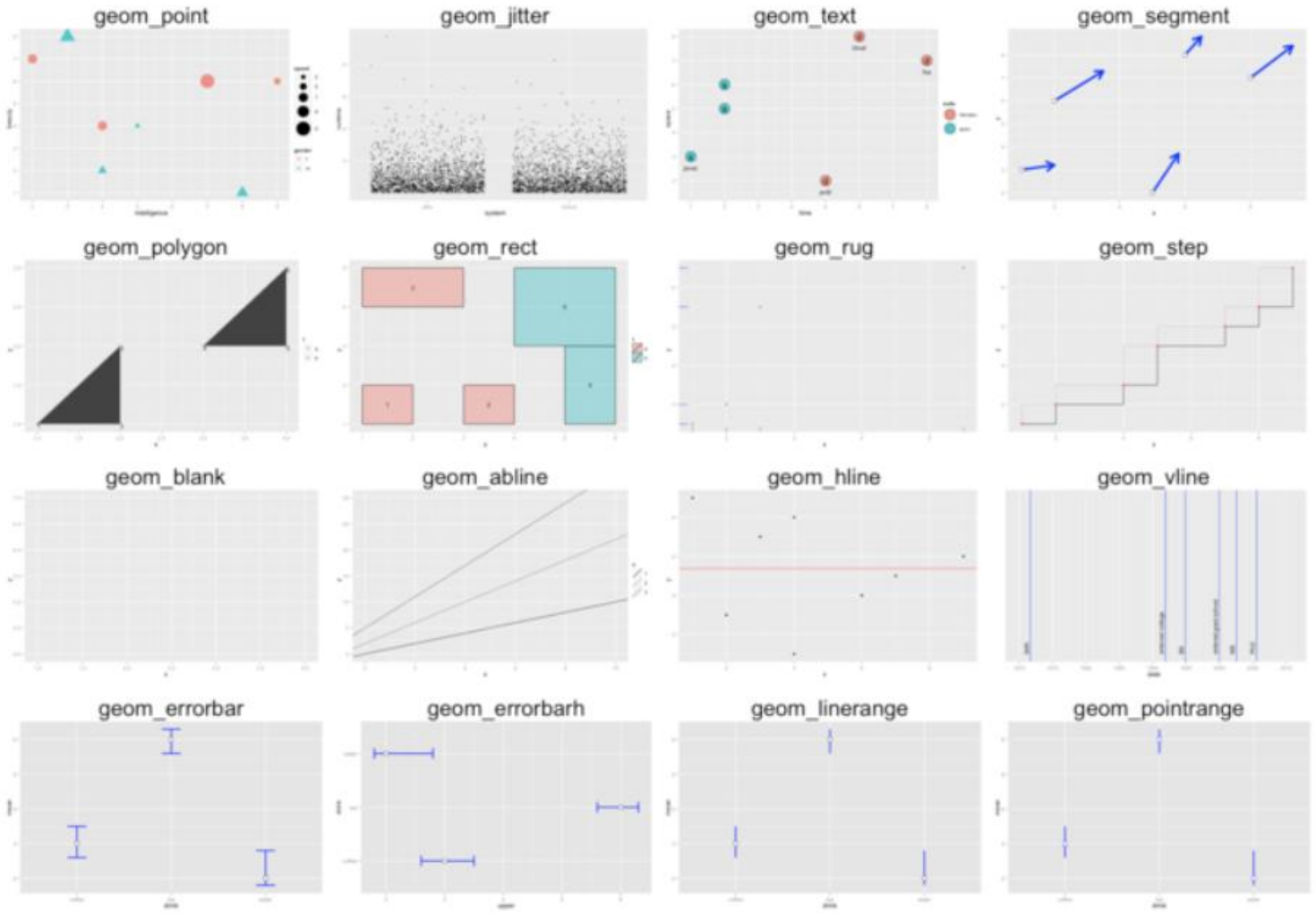
02. 데이터 시각화 실습

```
p + geom_point(color="tomato") +
    facet_grid(drv ~ cyl)
```



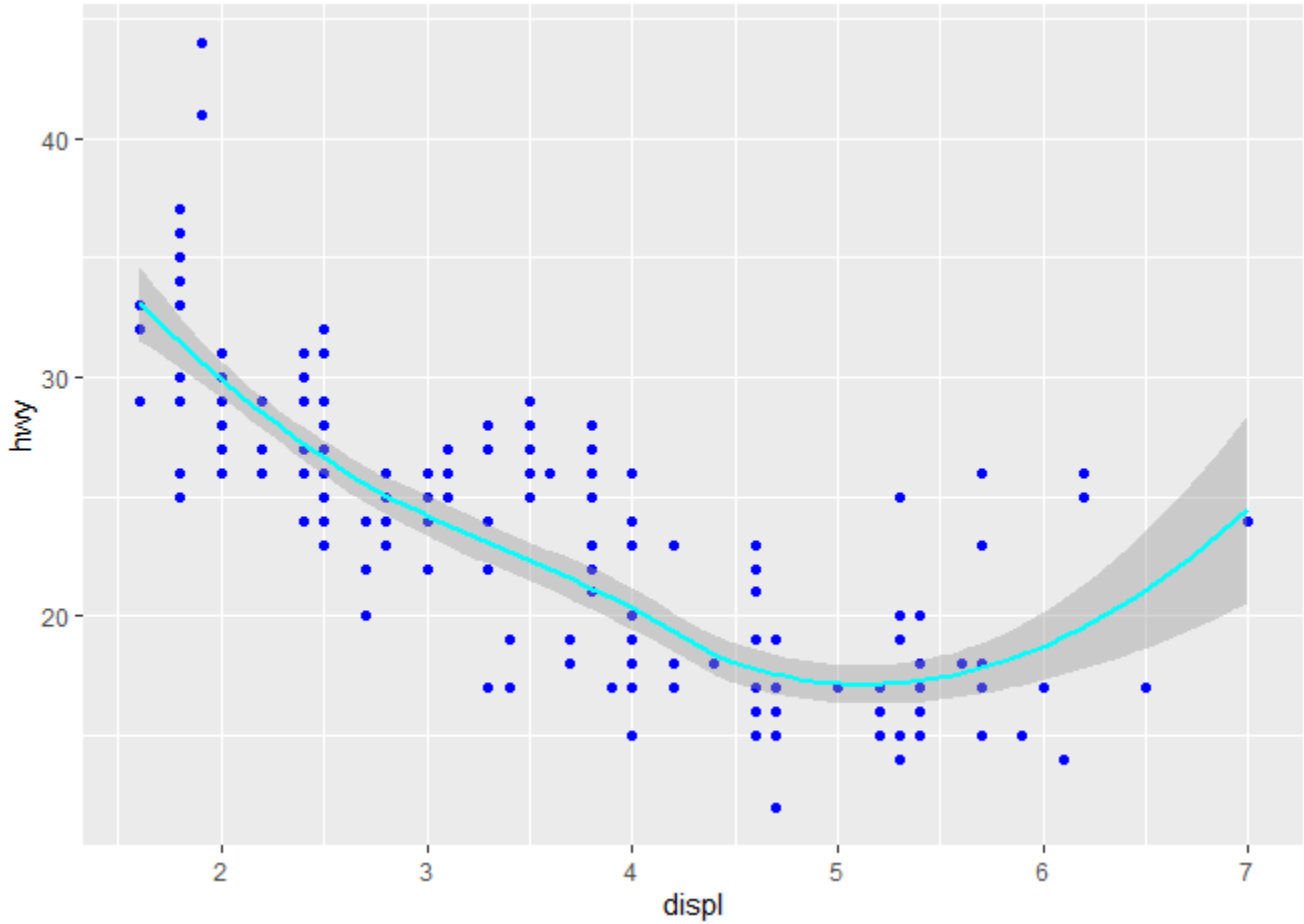
02. 데이터 시각화 실습

- geom: *geometric object*



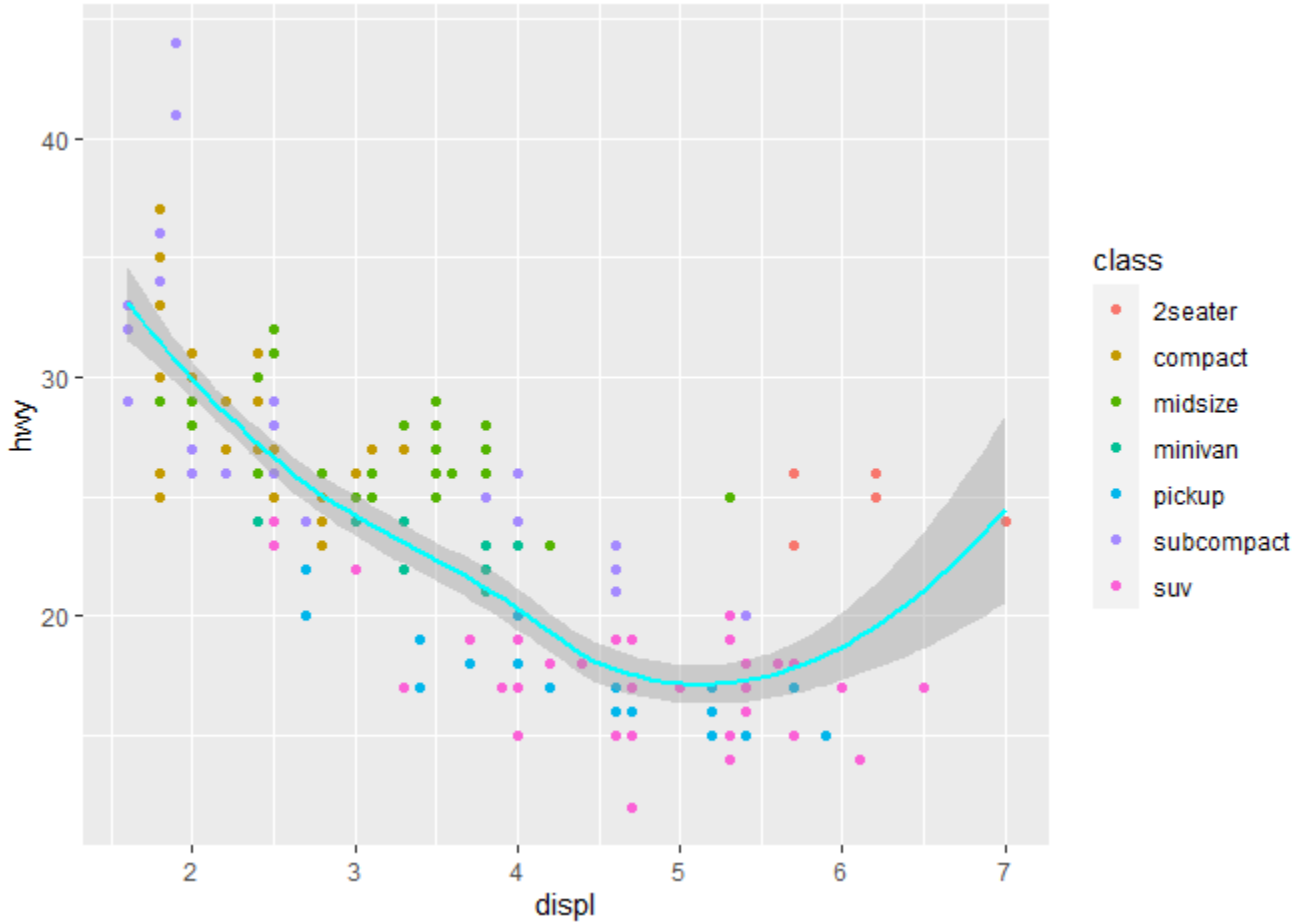
02. 데이터 시각화 실습

```
p + geom_point(color="blue") +
    geom_smooth(color="cyan")
```



02. 데이터 시각화 실습

```
p + geom_point(mapping = aes(color = class)) +
    geom_smooth(color = "cyan")
```



02. 데이터 시각화 실습

```
> library(ggplot2)
> data("diamonds")
> str(diamonds)
tibble [53,940 x 10] (S3: tbl_df/tbl/data.frame)
 $ carat  : num [1:53940] 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 [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

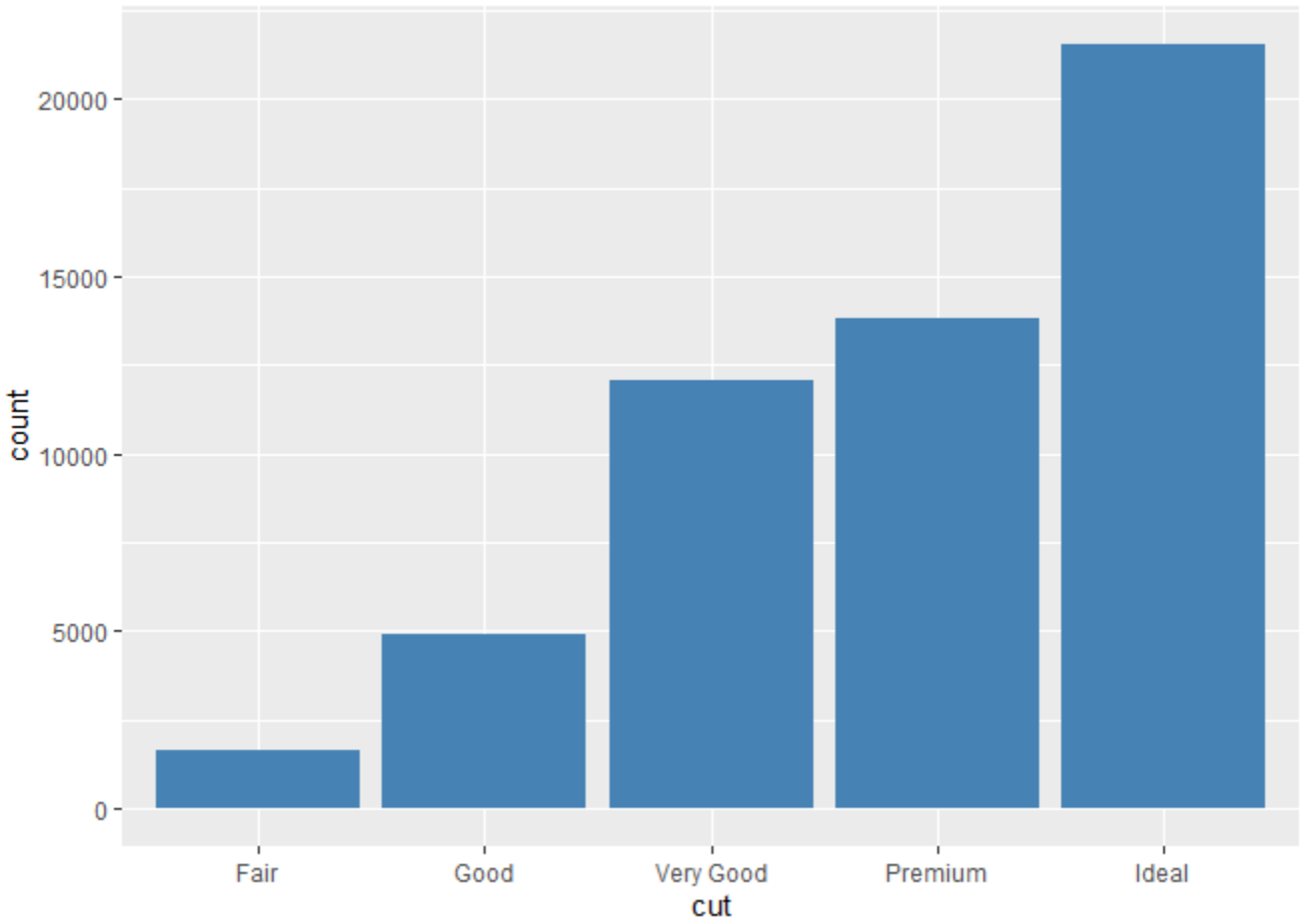
02. 데이터 시각화 실습

```
> head(diamonds)
# A tibble: 6 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  0.29 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
```

02. 데이터 시각화 실습

```
p <- ggplot(data = diamonds)
p + geom_bar(mapping = aes(x = cut),
             fill = "steelblue")
```

```
p + stat_count(mapping = aes(x = cut),
               fill = "steelblue")
```



02. 데이터 시각화 실습

- stat: *statistical transformation*

1. **`geom_bar()`** begins with the **`diamonds`** data set

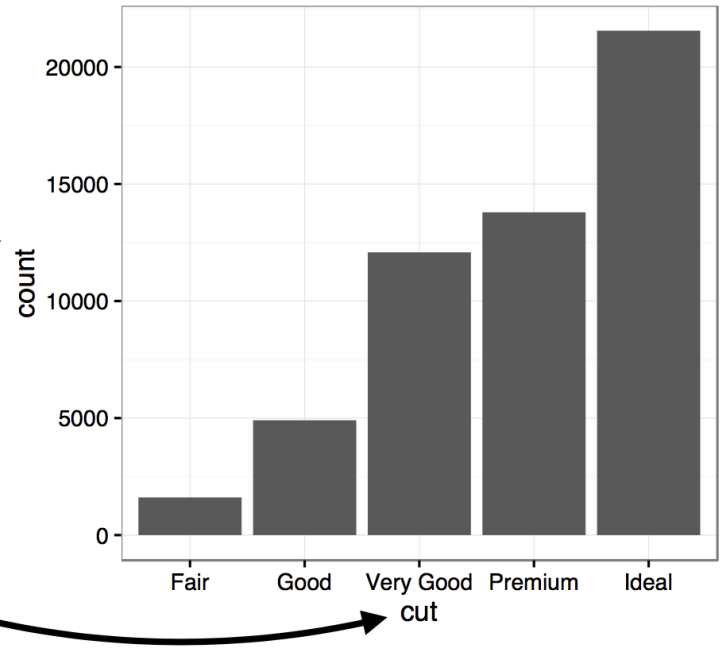
2. **`geom_bar()`** transforms the data with the "count" stat, which returns a data set of cut values and counts.

3. **`geom_bar()`** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

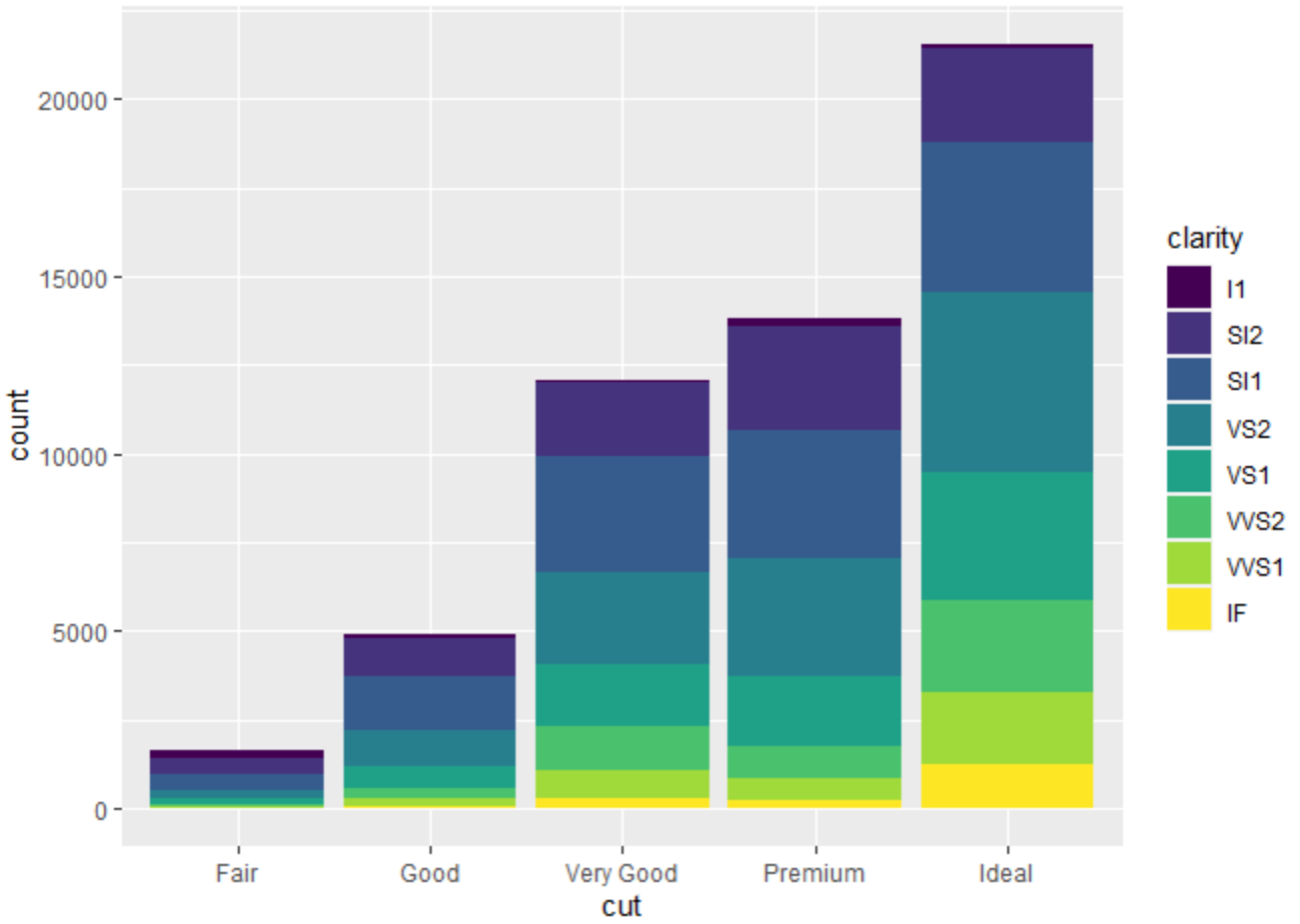
stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1



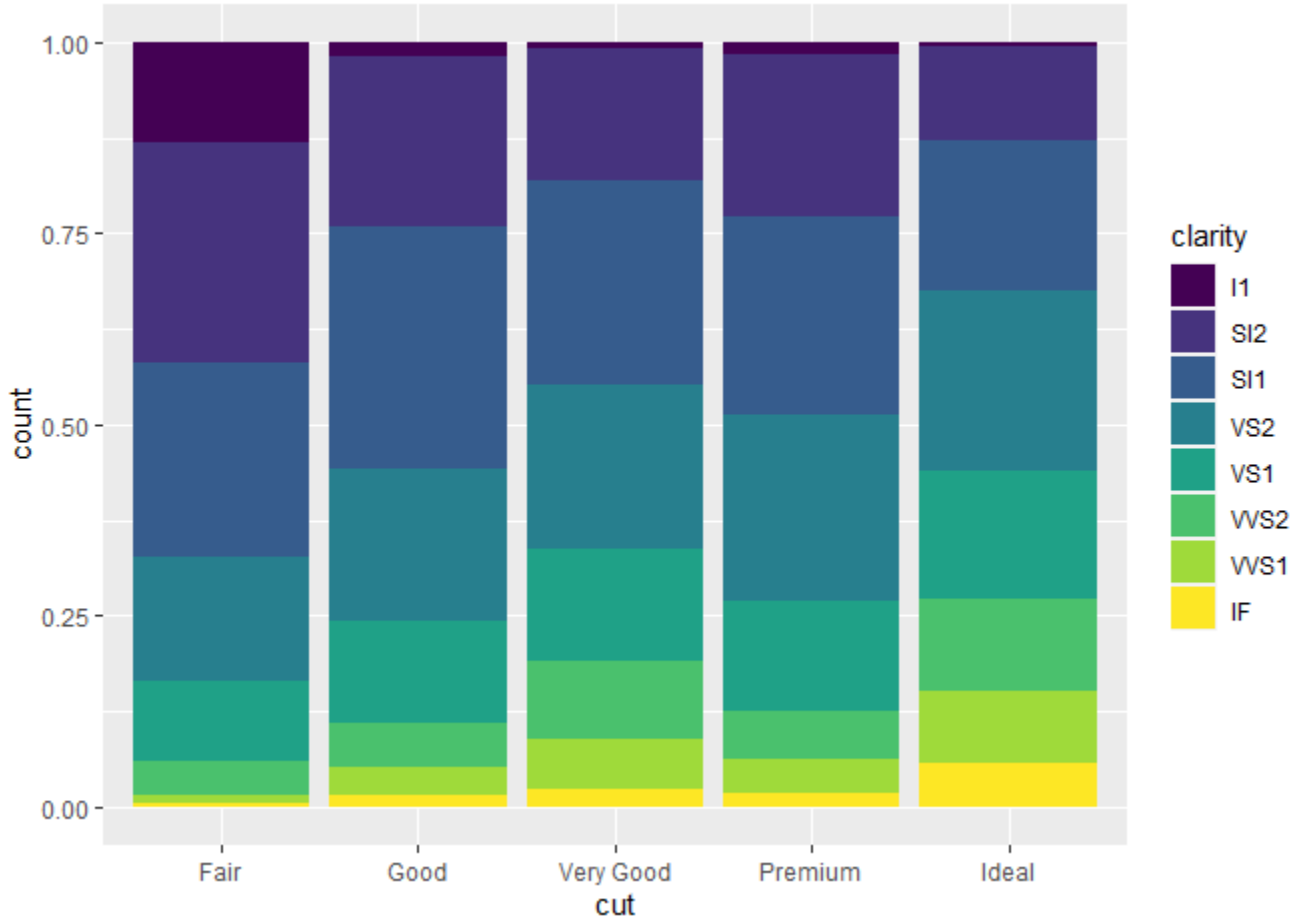
02. 데이터 시각화 실습

```
p + geom_bar(mapping = aes(x = cut,
                           fill = clarity))
```



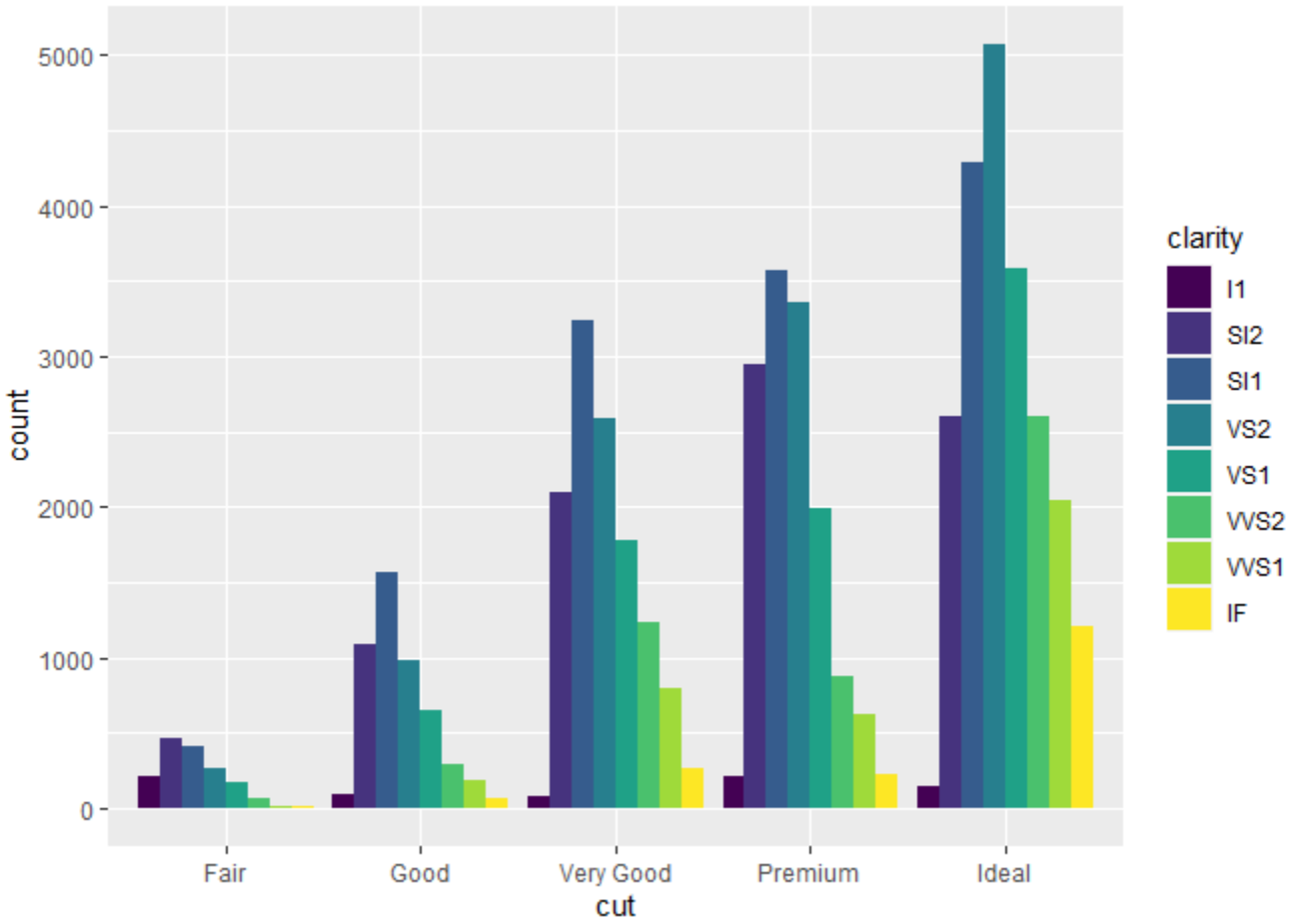
02. 데이터 시각화 실습

```
p + geom_bar(mapping = aes(x = cut, fill = clarity),
             position = "fill")
```



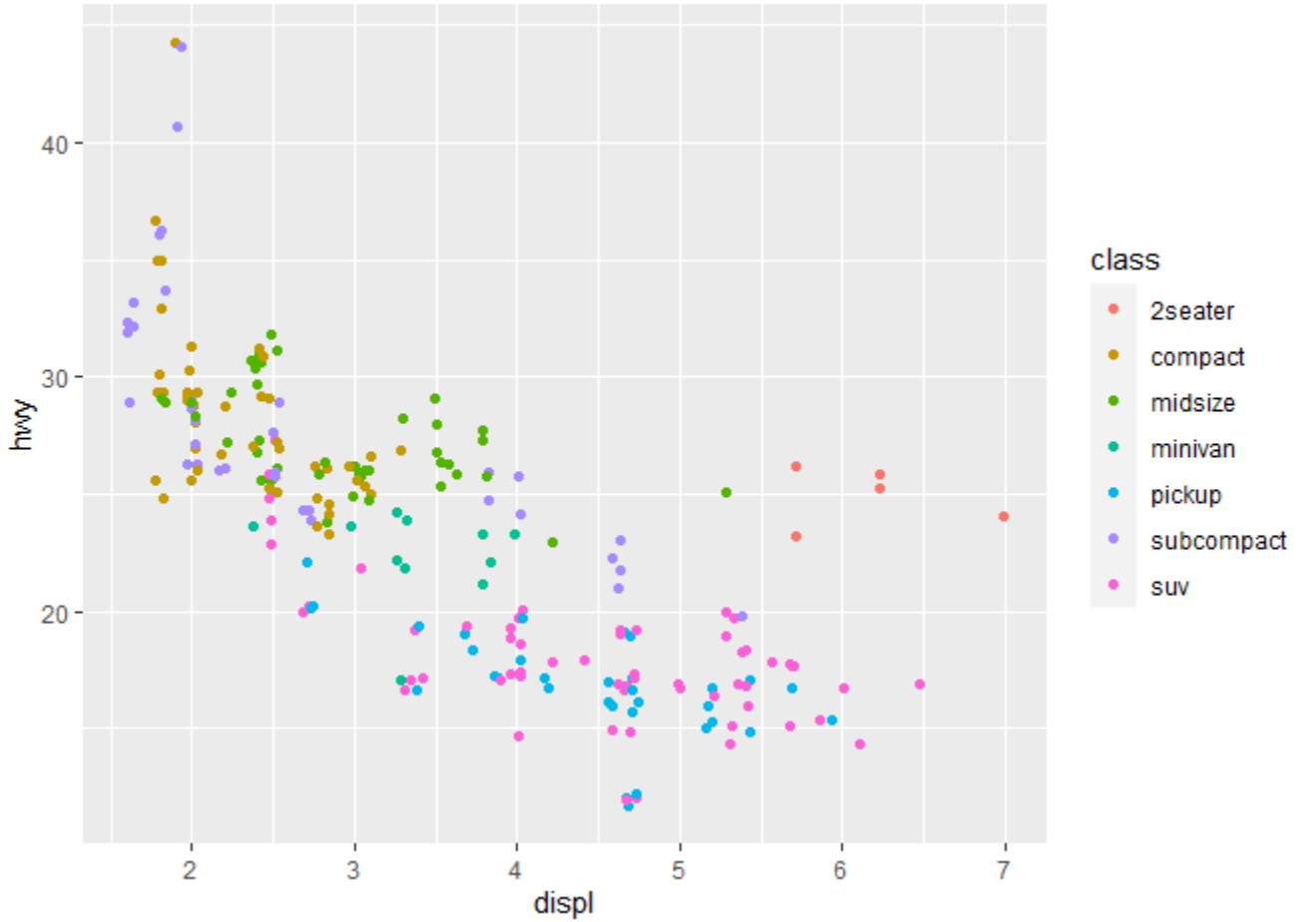
02. 데이터 시각화 실습

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



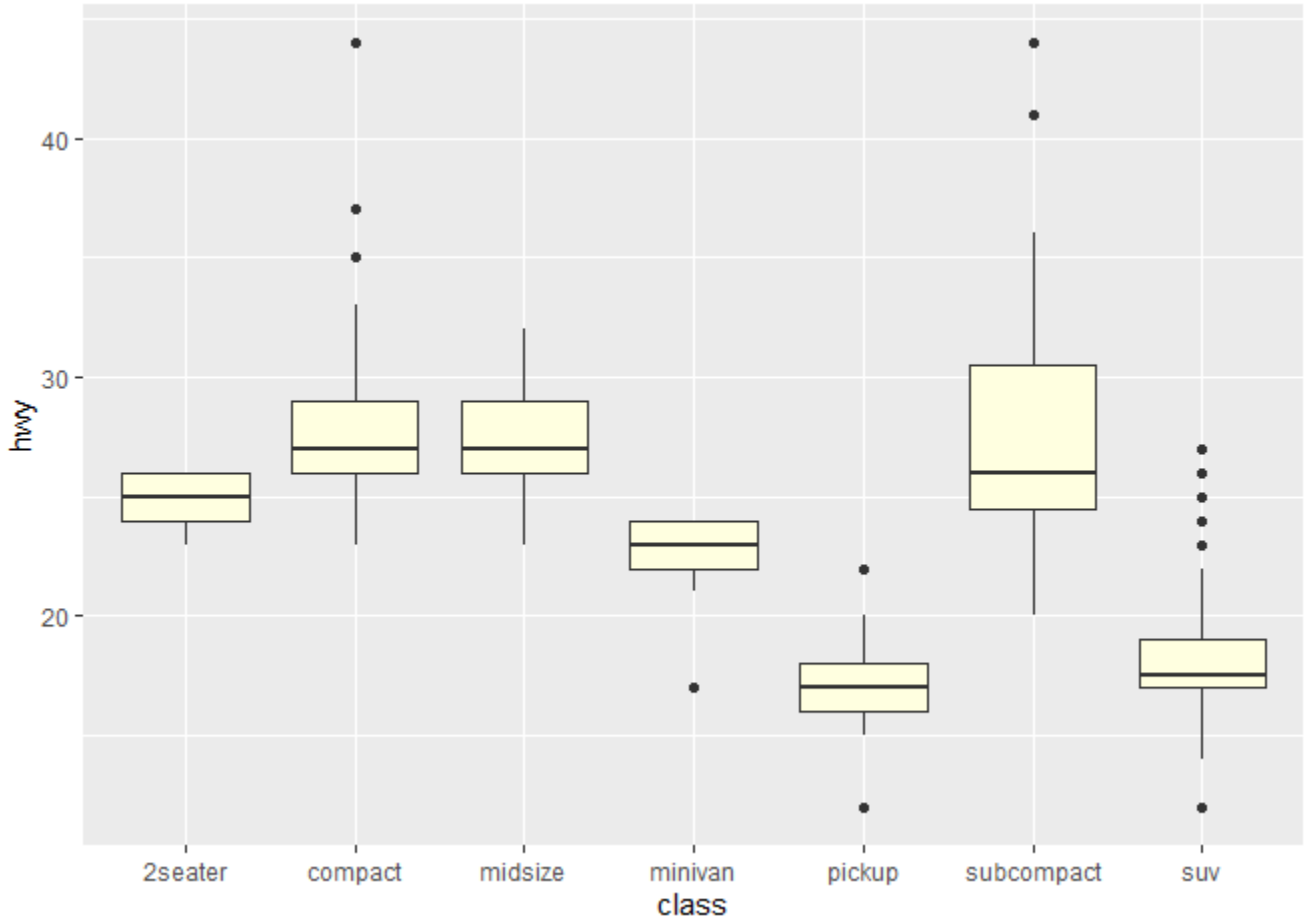
02. 데이터 시각화 실습

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



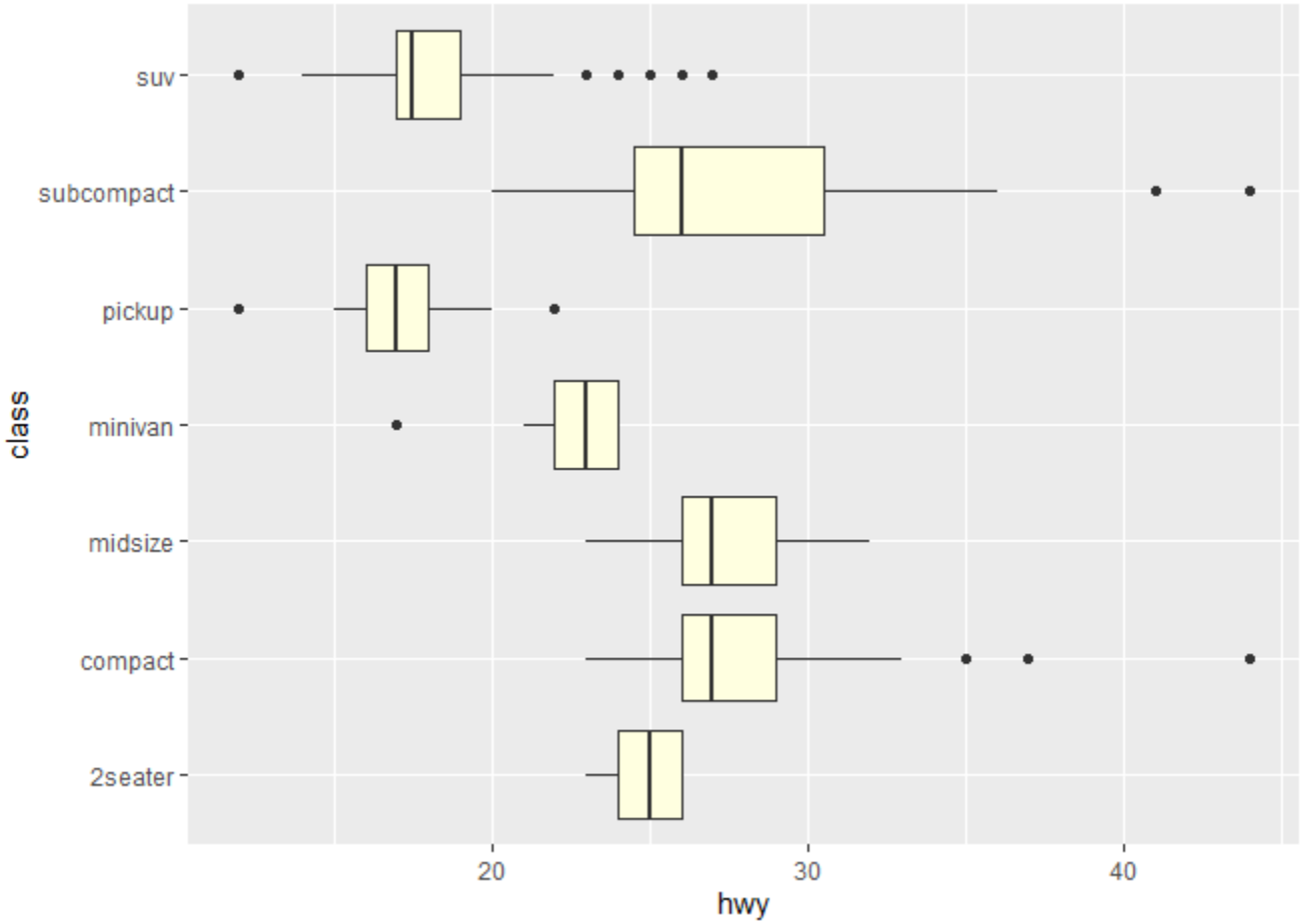
02. 데이터 시각화 실습

```
p <- ggplot(data = mpg,
            mapping = aes(x = class, y = hwy))
p + geom_boxplot(fill = "lightyellow")
```



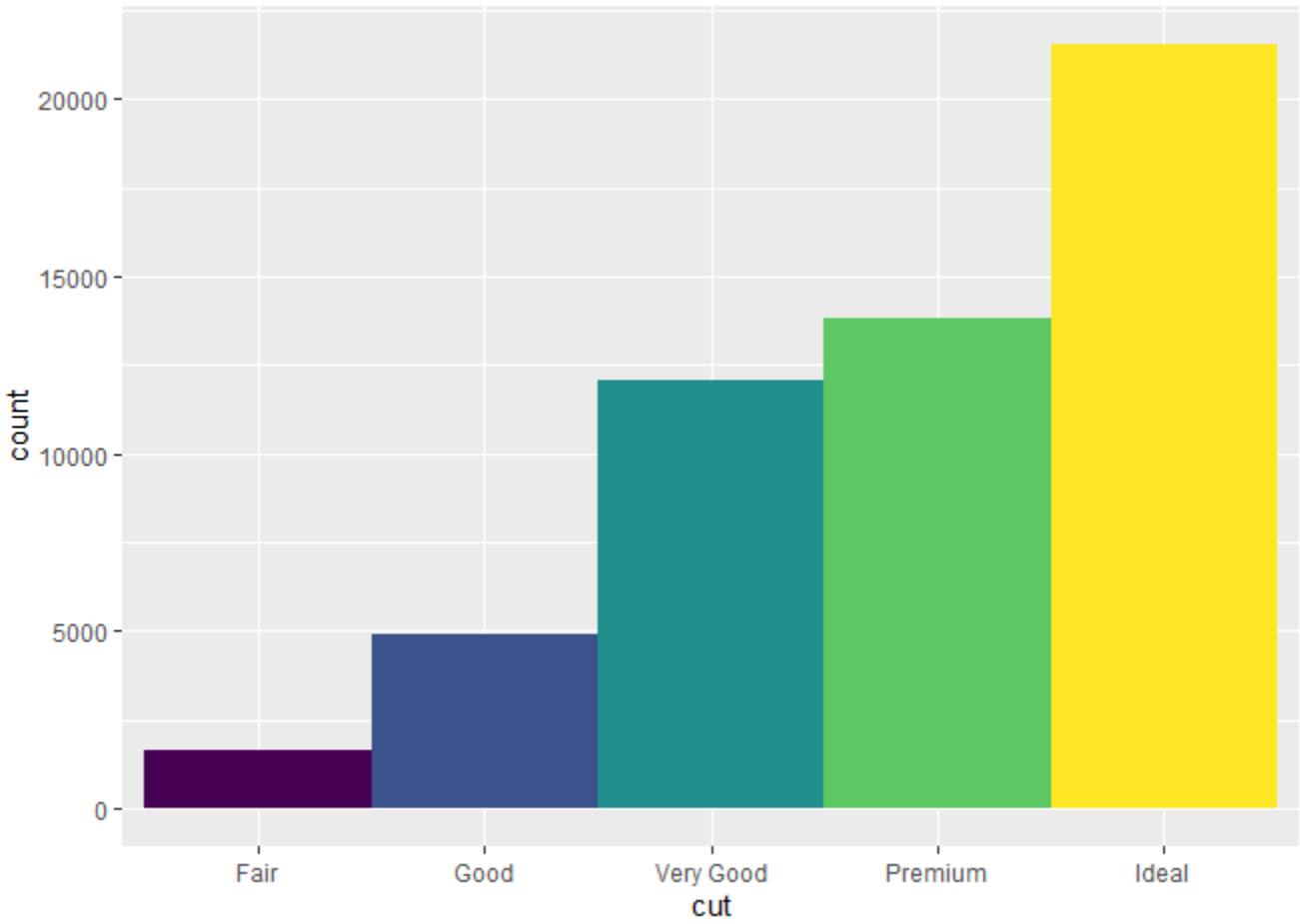
02. 데이터 시각화 실습

```
p + geom_boxplot(fill = "lightyellow") +
    coord_flip()
```



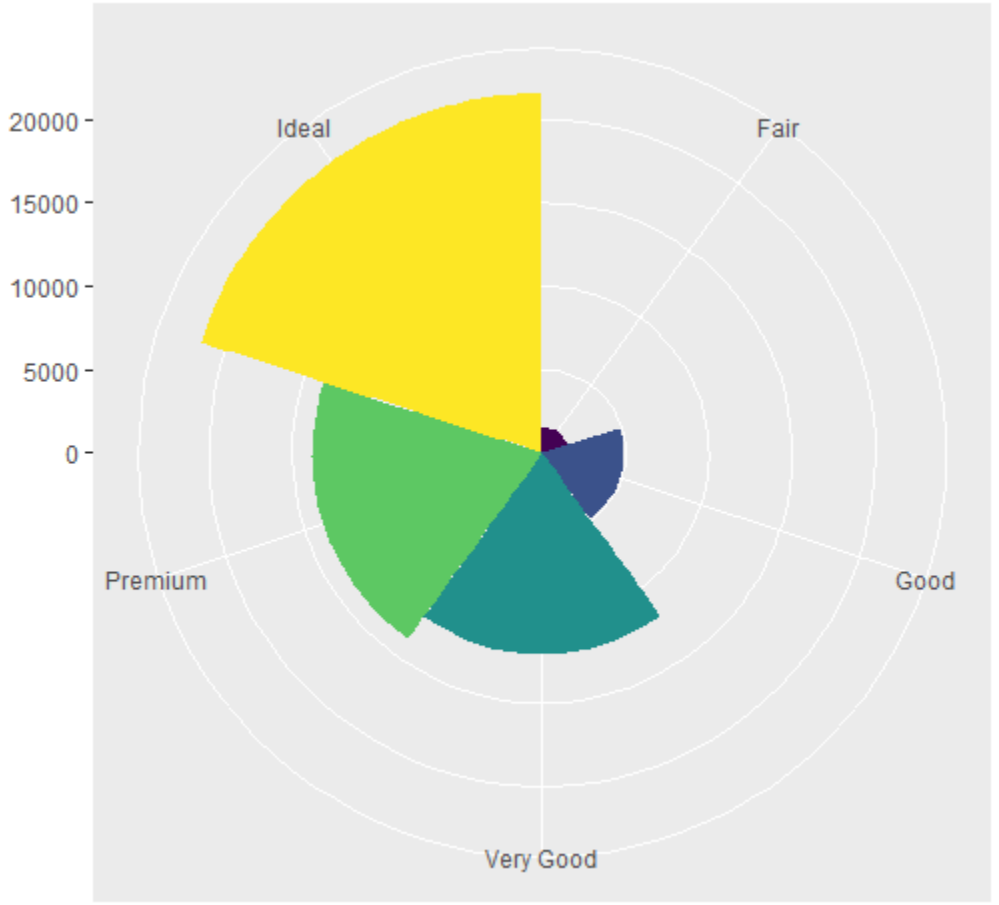
02. 데이터 시각화 실습

```
p <- ggplot(data = diamonds,
            mapping = aes(x = cut, fill = cut))
p + geom_bar(show.legend = F, width = 1)
```



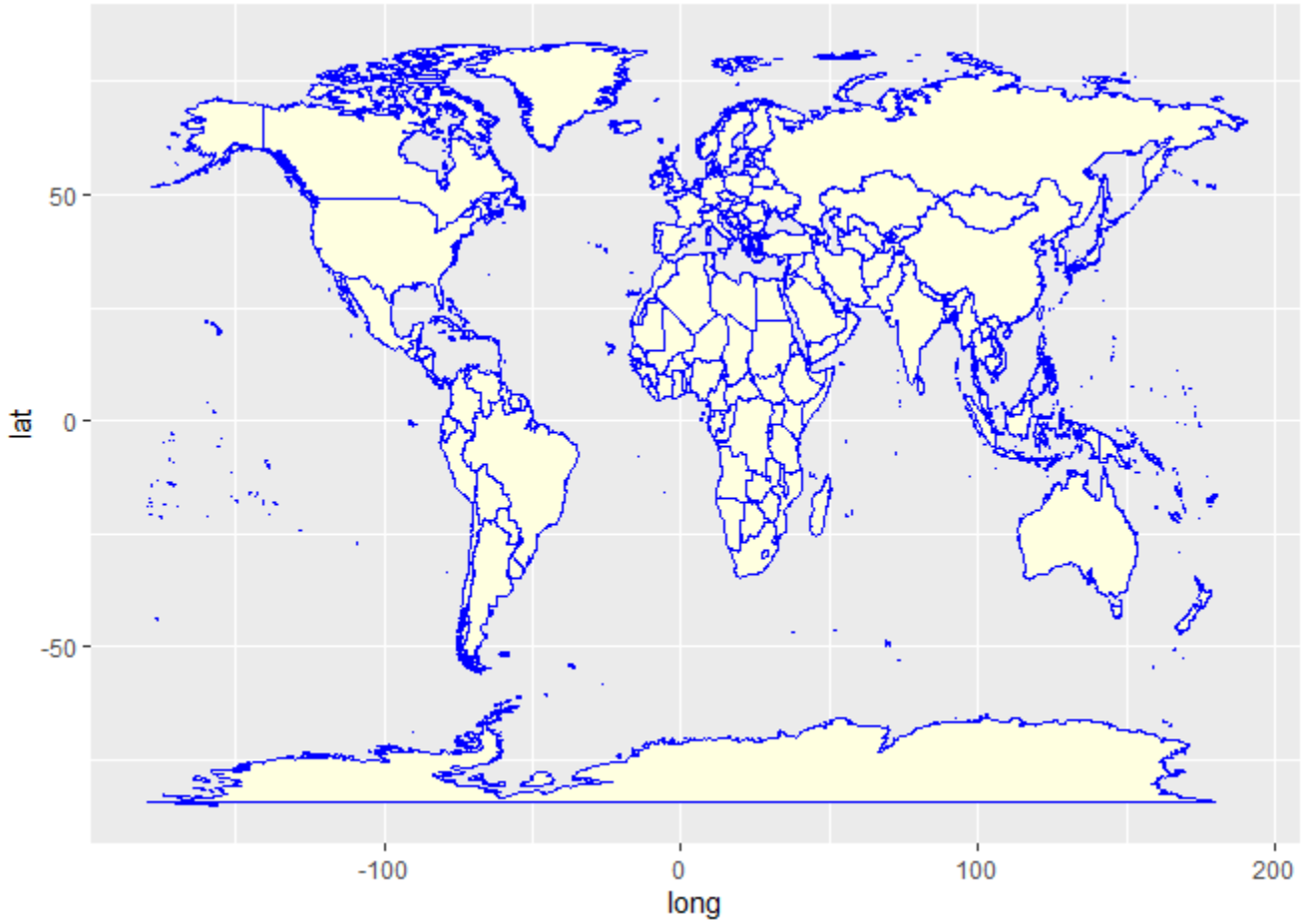
02. 데이터 시각화 실습

```
p + geom_bar(show.legend = F, width = 1) +
    labs(x = NULL, y = NULL) +
    theme(aspect.ratio = 1) +
    coord_polar()
```



02. 데이터 시각화 실습

```
world <- map_data("world")
ggplot(world, aes(long, lat, group = group)) +
    geom_polygon(fill = "lightyellow", color = "blue")
```



02. 데이터 시각화 실습

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION>
```

02. 데이터 시각화 실습

1. Begin with the **`diamonds`** data set

2. Compute counts for each cut value with **`stat_count()`**.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

02. 데이터 시각화 실습

3. Represent each observation with a bar.
4. Map the **`fill`** of each bar to the **`..count..`** variable.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

02. 데이터 시각화 실습

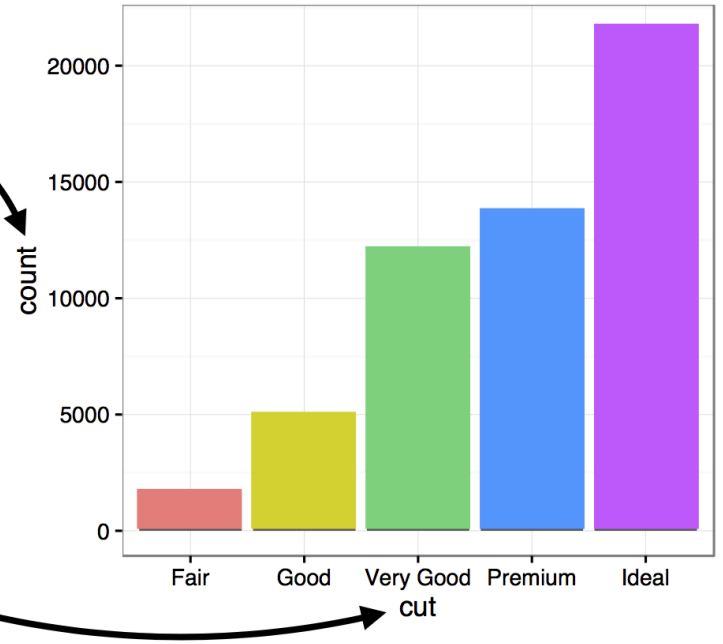
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
…	…	…	…	…	…	…	…	…	…

stat_count()

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

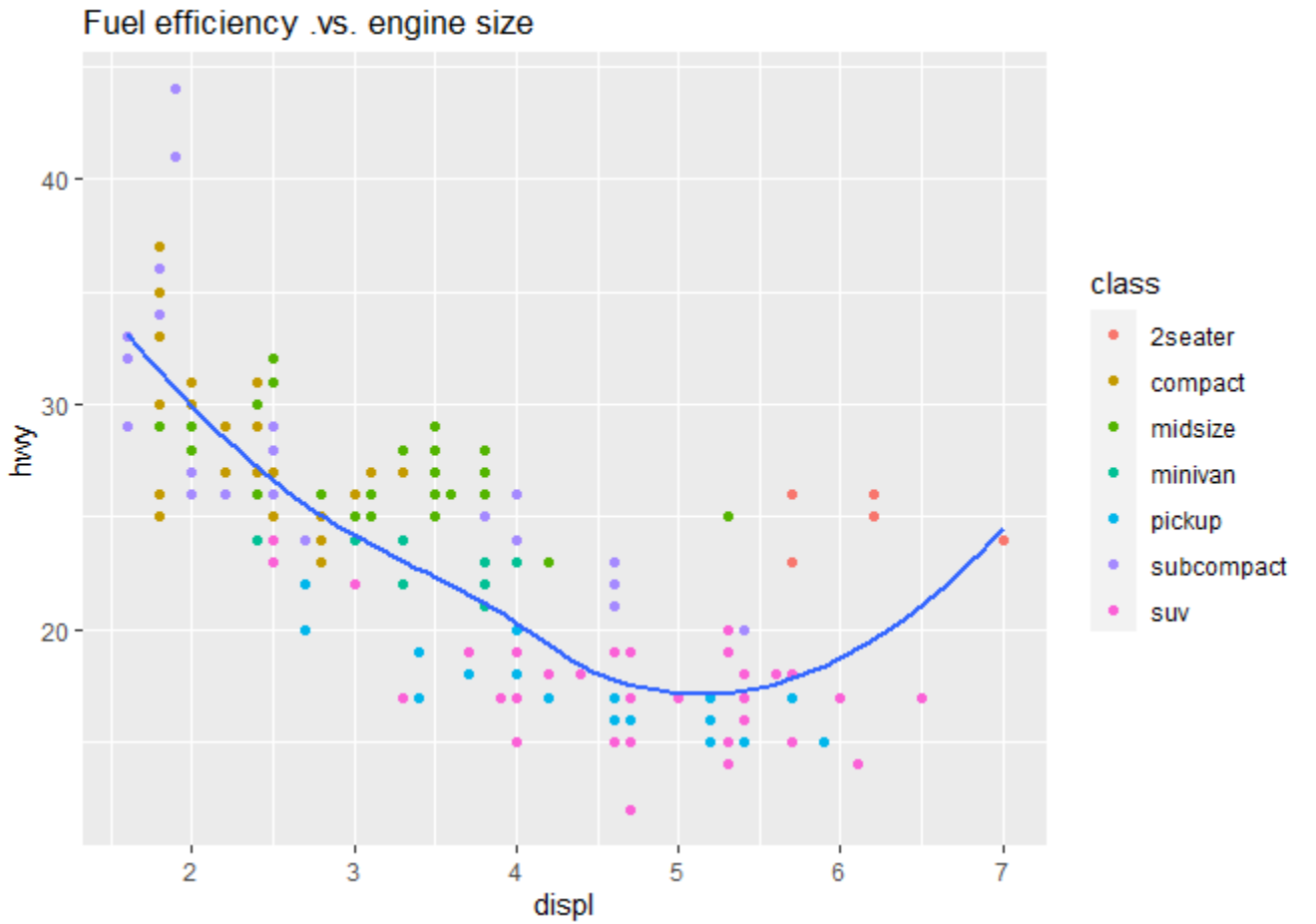
5. Place geoms in a cartesian coordinate system.

6. Map the y values to **..count..** and the x values to **cut**.



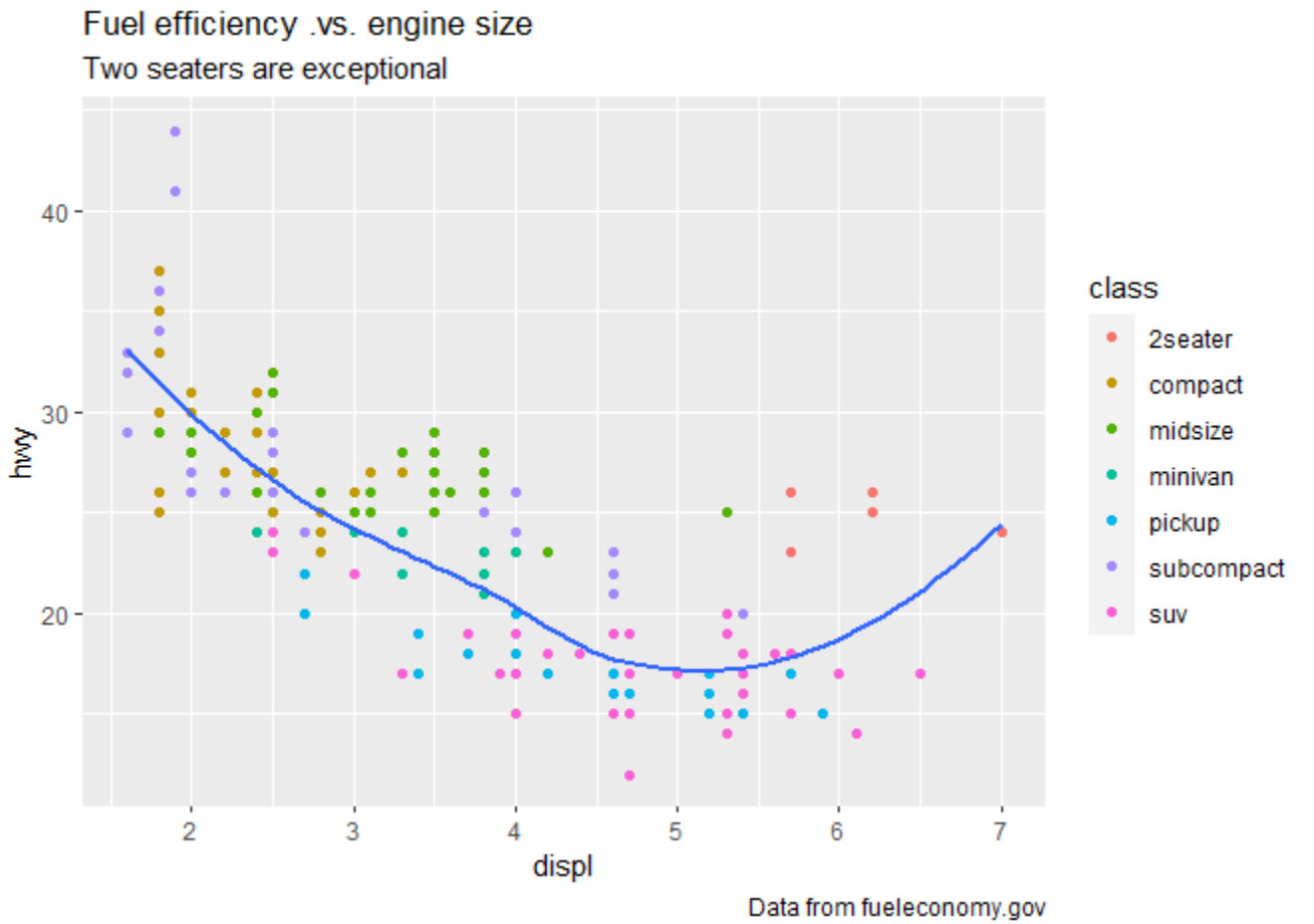
02. 데이터 시각화 실습

```
p <- ggplot(mpg, aes(displ, hwy)) +
    geom_point(aes(color = class)) +
    geom_smooth(se = FALSE)
p + labs(title = "Fuel efficiency .vs. engine size")
```



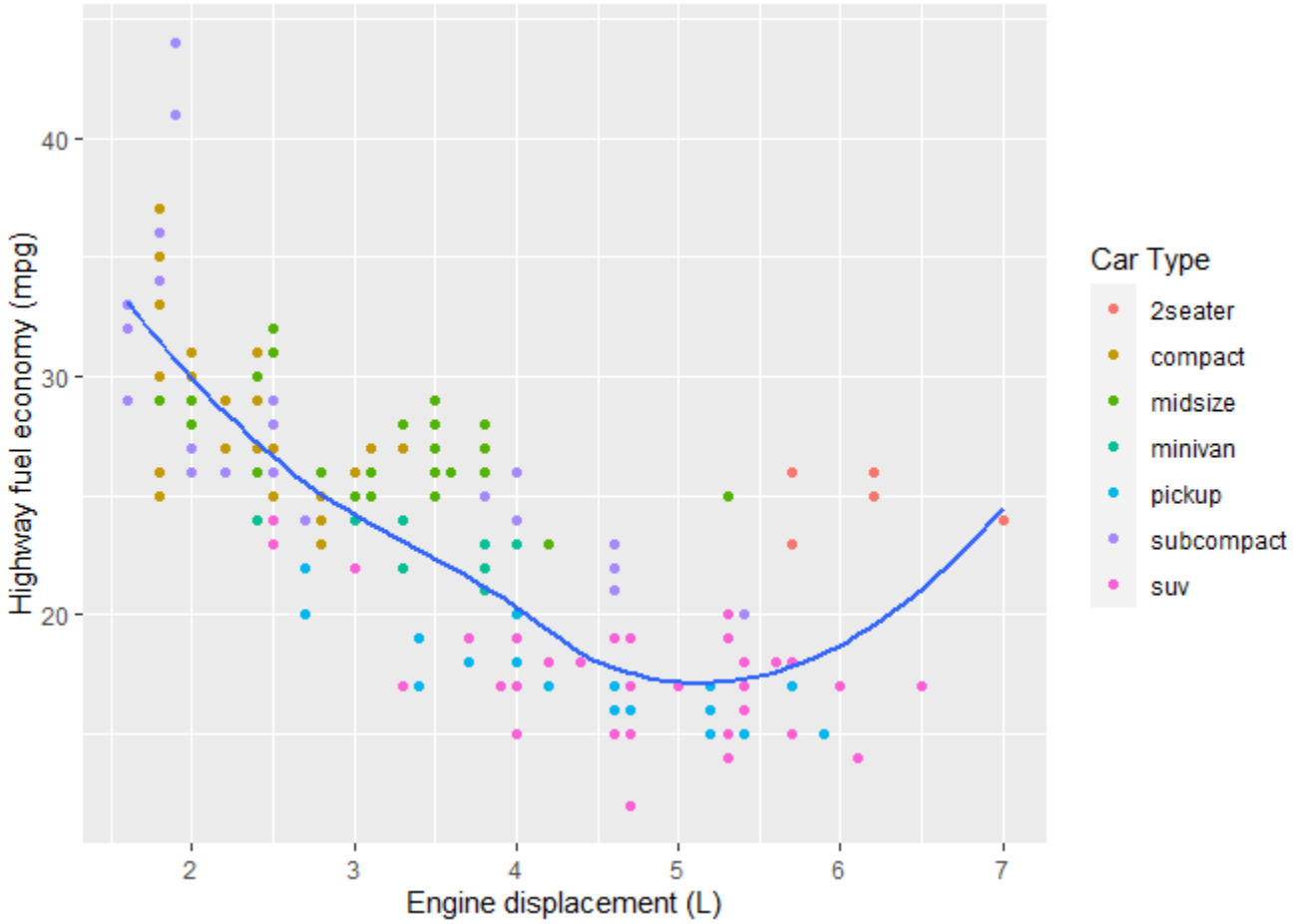
02. 데이터 시각화 실습

```
p + labs(title = "Fuel efficiency .vs. engine size",
         subtitle = "Two seaters are exceptional",
         caption = "Data from fueleconomy.gov")
```



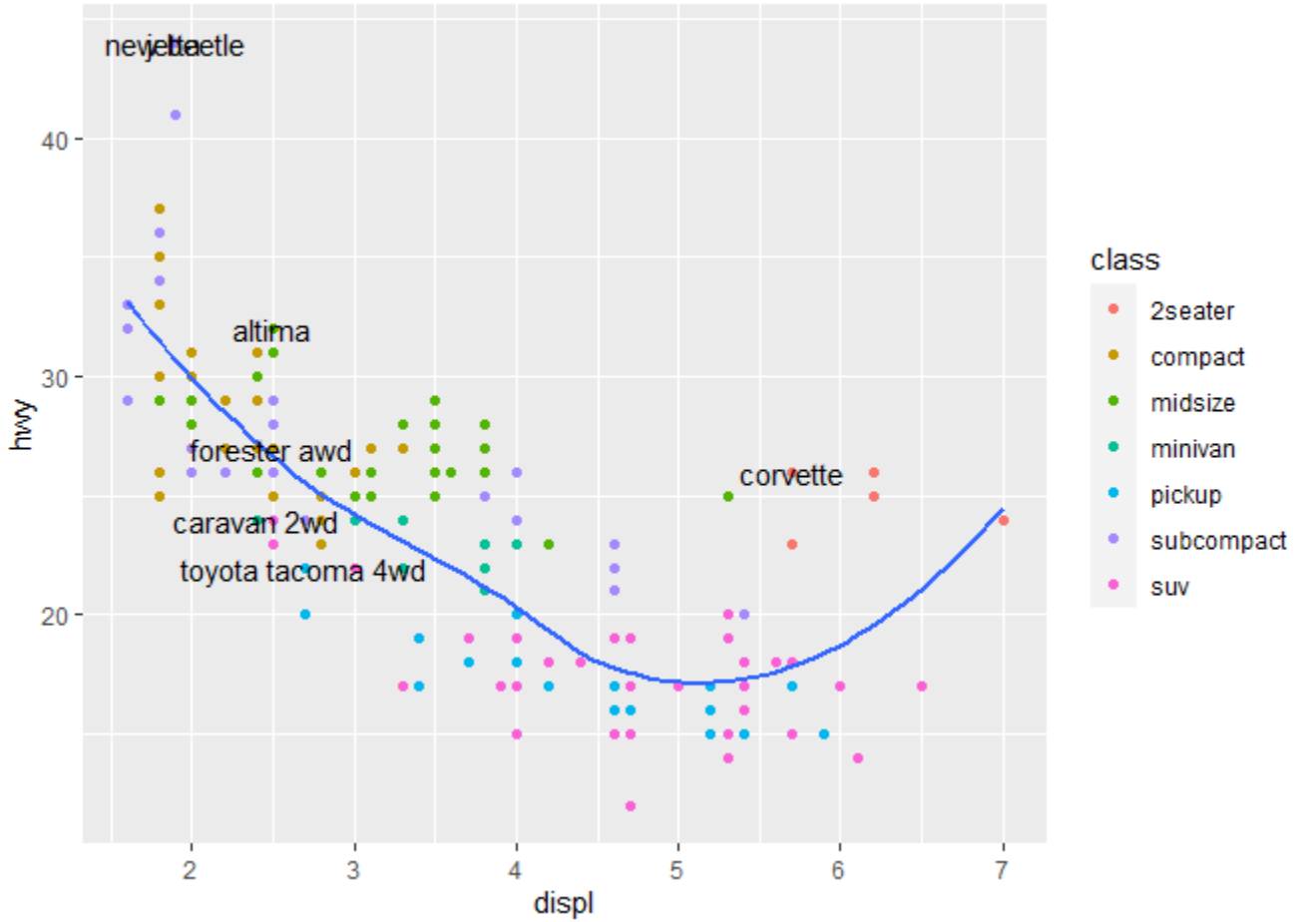
02. 데이터 시각화 실습

```
p + labs(x = "Engine displacement (L)",
         y = "Highway fuel economy (mpg)",
         color = "Car Type")
```



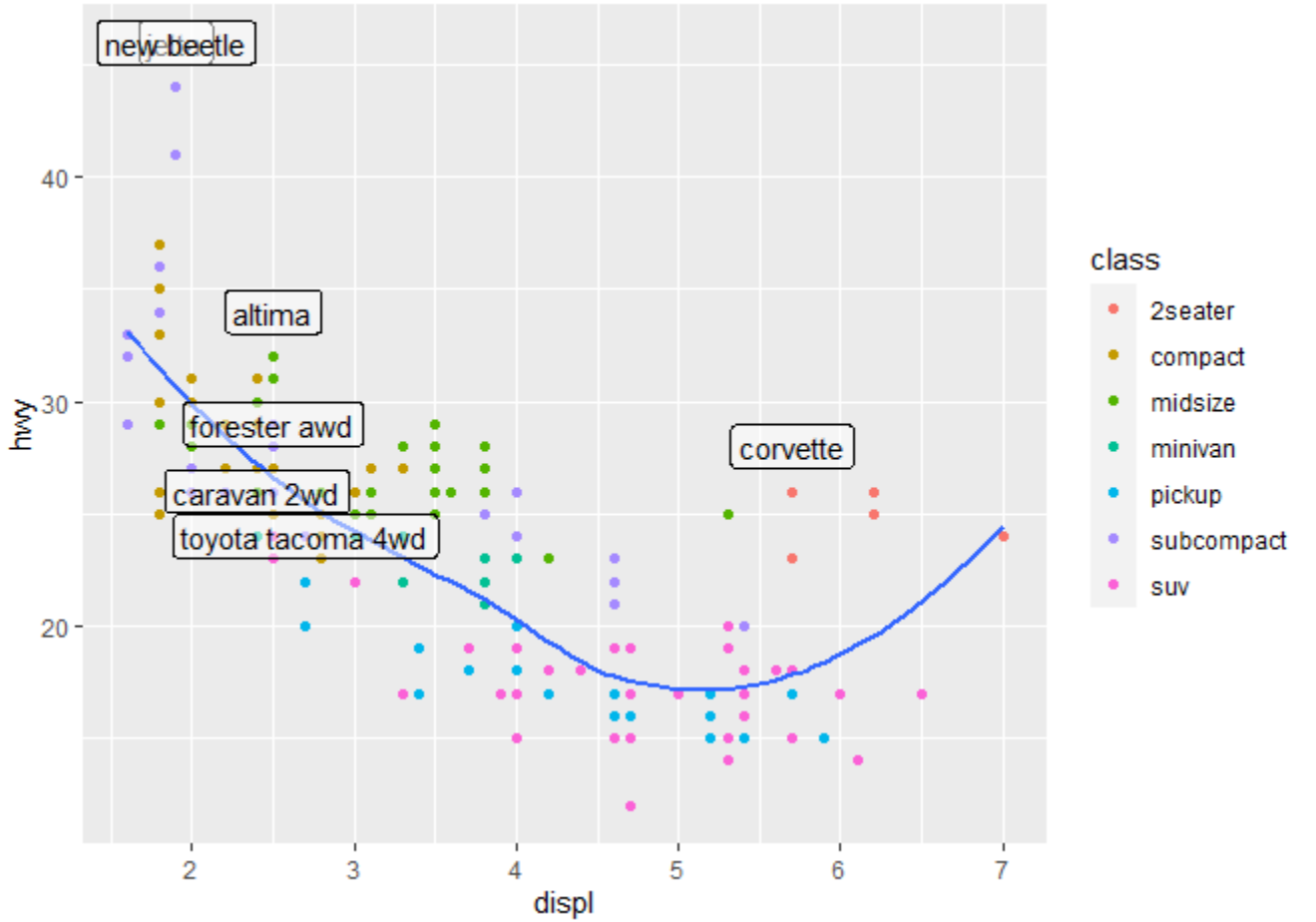
02. 데이터 시각화 실습

```
best_in_class <- mpg %>%
    group_by(class) %>%
    filter(row_number(desc(hwy)) == 1)

p + geom_text(aes(label = model),
              data = best_in_class)
```



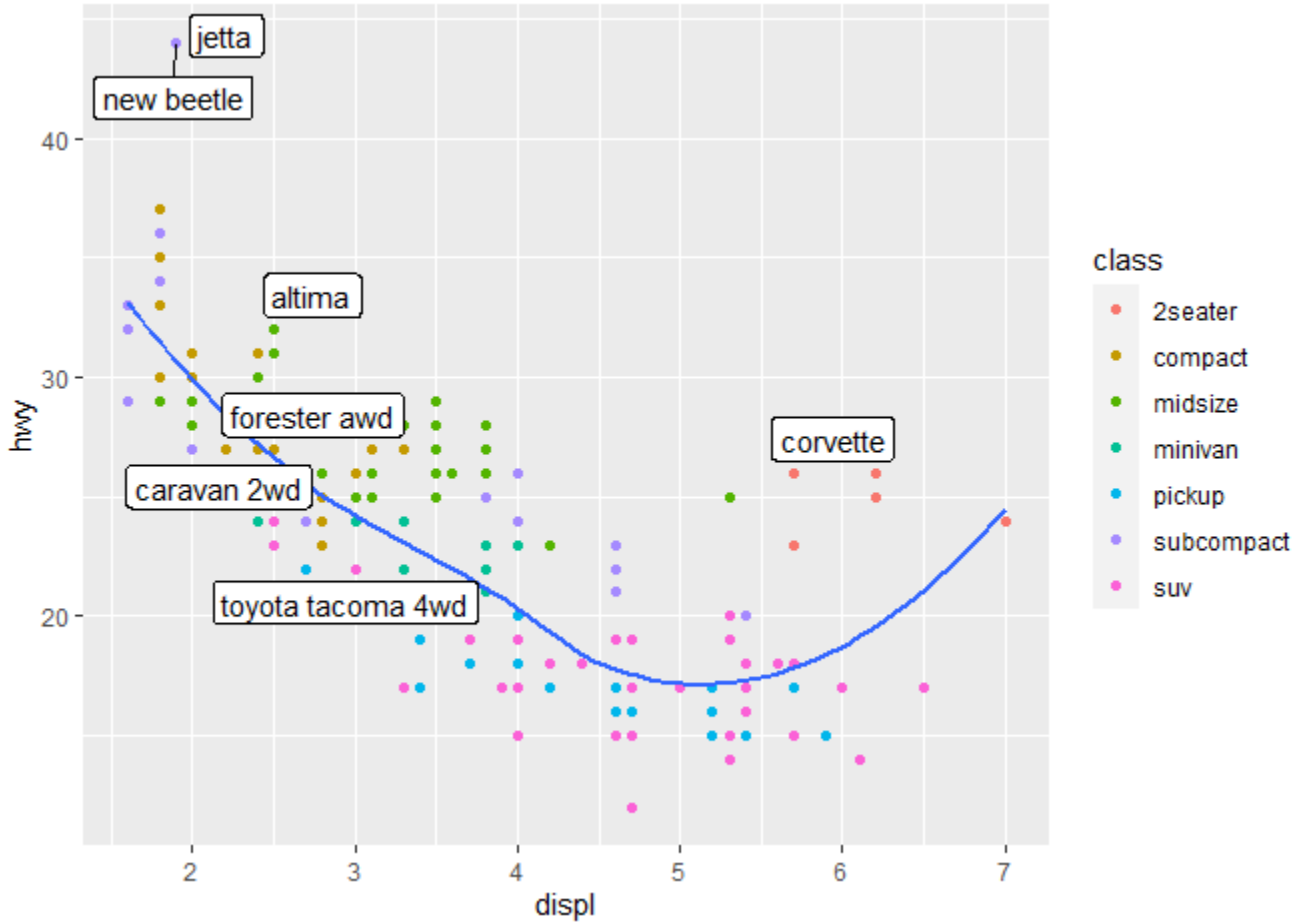
02. 데이터 시각화 실습

```
p + geom_label(aes(label = model),
               data = best_in_class,
               nudge_y = 2, alpha = 0.5)
```



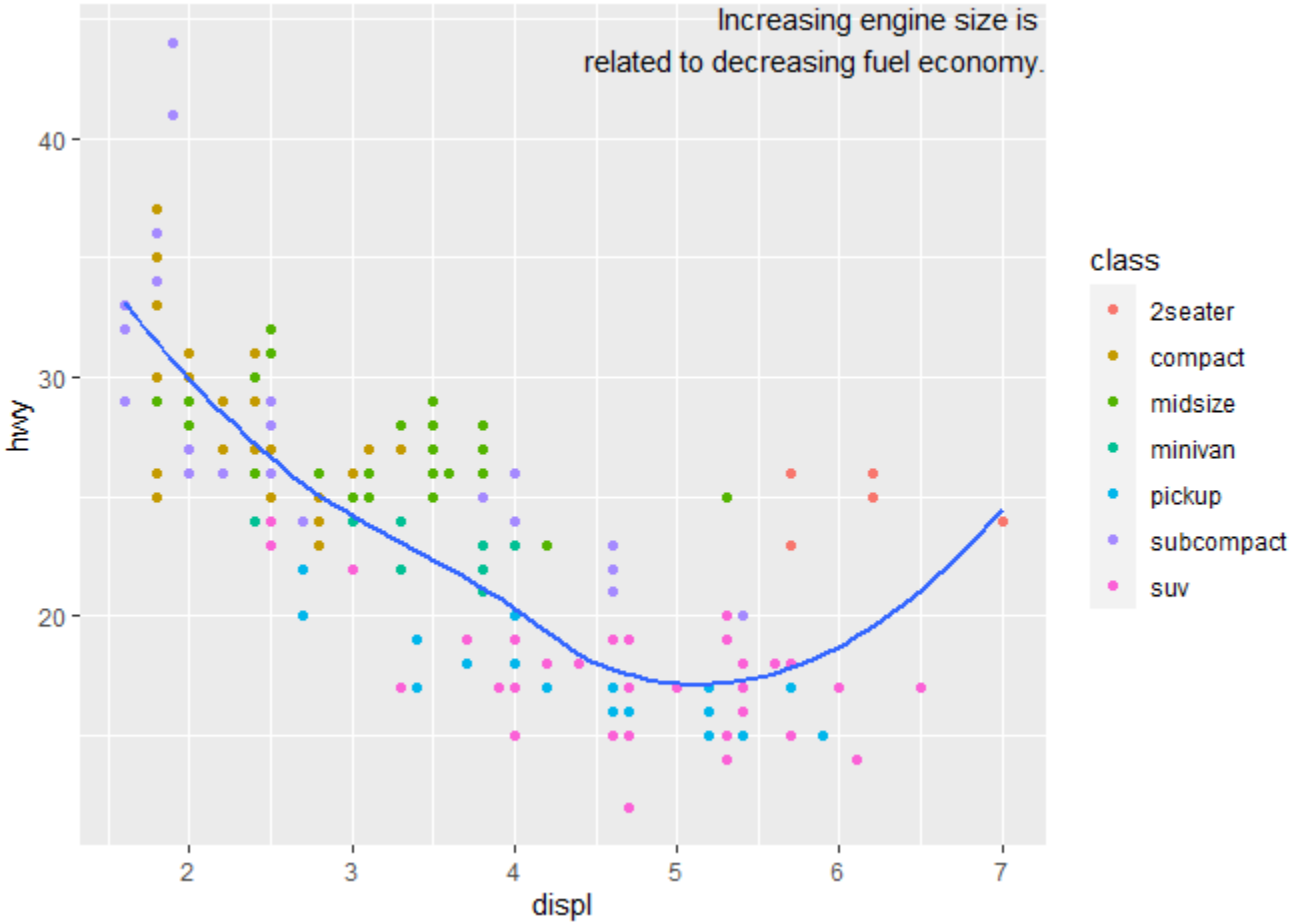
02. 데이터 시각화 실습

```
p + ggrepel::geom_label_repel(aes(label = model),
                              data = best_in_class)
```



02. 데이터 시각화 실습

```
label <- tibble(
    displ = Inf, hwy = Inf,
    label = "Increasing engine size is \n related to decreasing fuel economy."
    )

p + geom_text(aes(label = label),
              data = label,
              vjust = "top",
              hjust = "right")
```

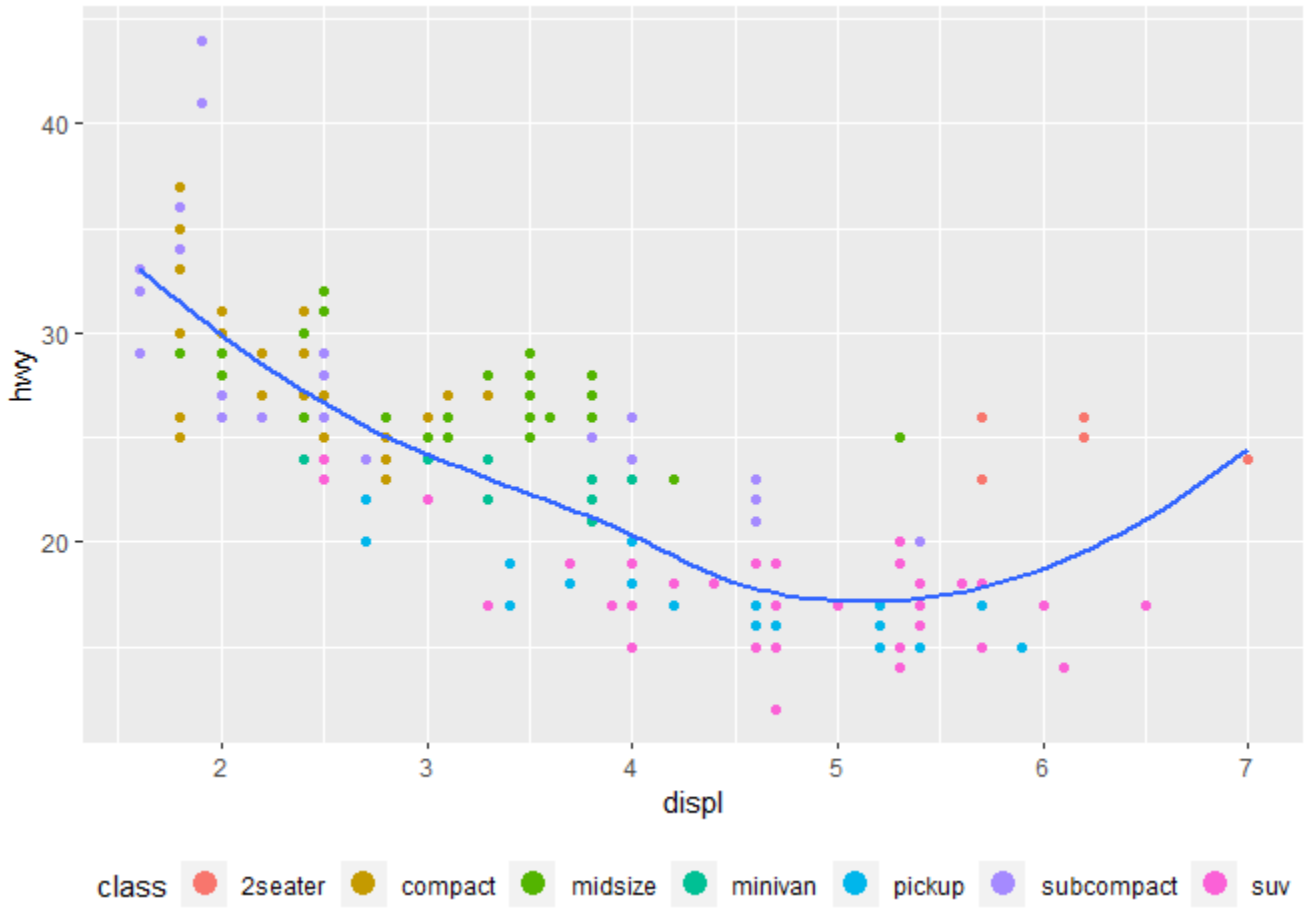


02. 데이터 시각화 실습

```
p + theme(legend.position = "top")
p + theme(legend.position = "bottom")
p + theme(legend.position = "left")
p + theme(legend.position = "right")
```

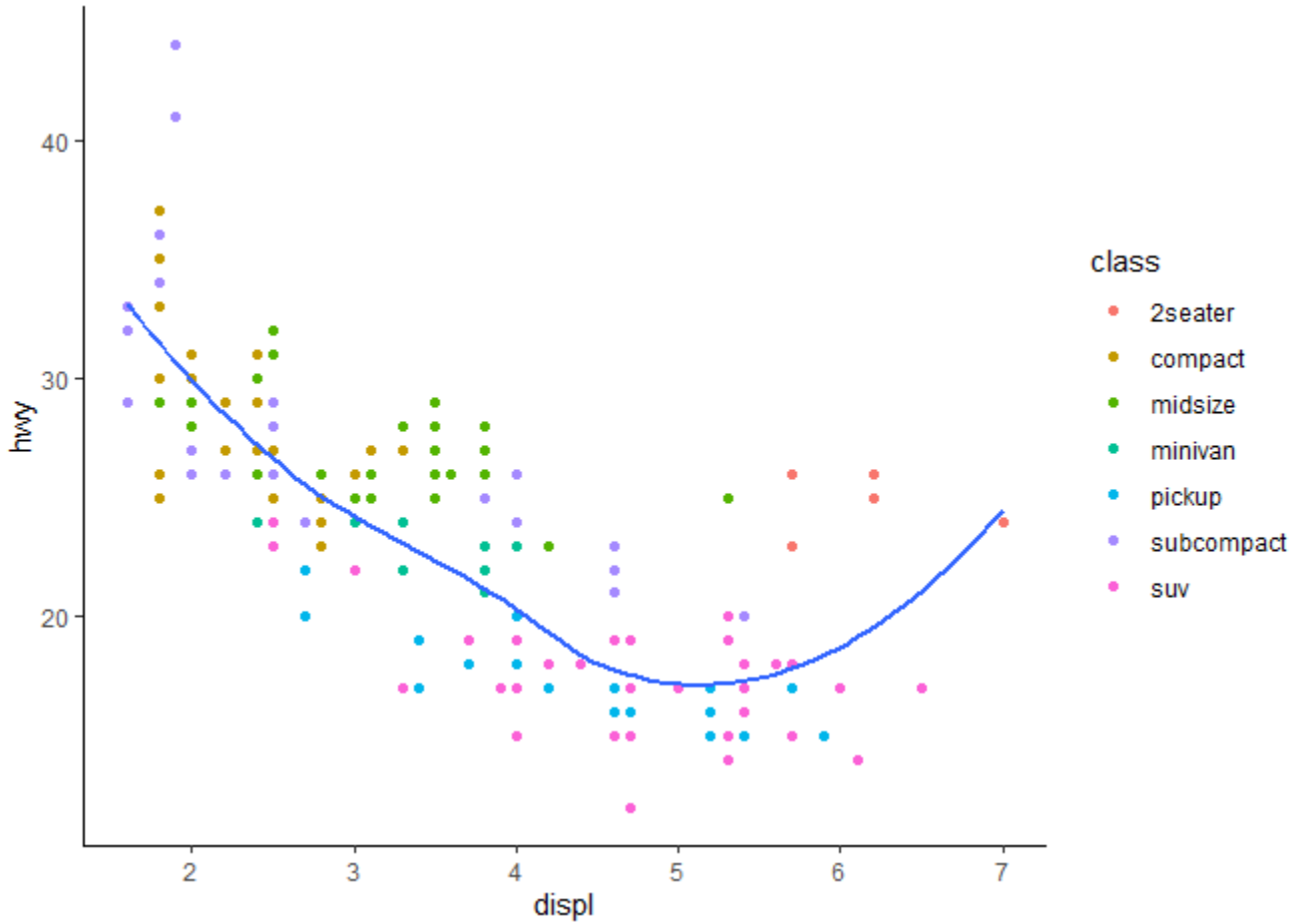
02. 데이터 시각화 실습

```
p + theme(legend.position = "bottom") +
    guides(color = guide_legend(nrow = 1,
                                override.aes = list(size = 4)))
```

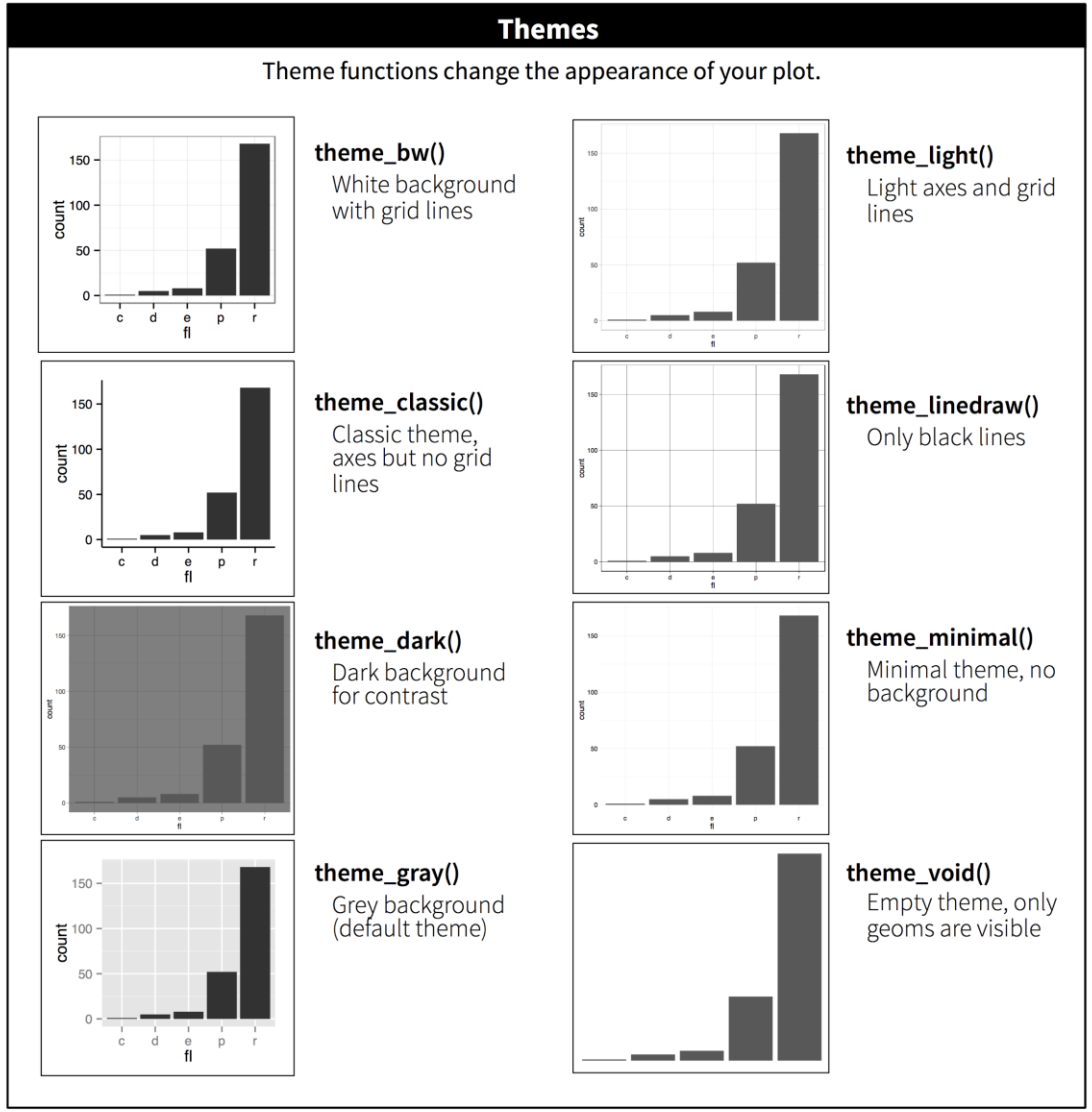


02. 데이터 시각화 실습

```
p + theme_classic()
p + theme_bw()
p + theme_light()
p + theme_linedraw()
p + theme_dark()
p + theme_minimal()
p + theme_gray()
p + theme_void()
```



02. 데이터 시각화 실습

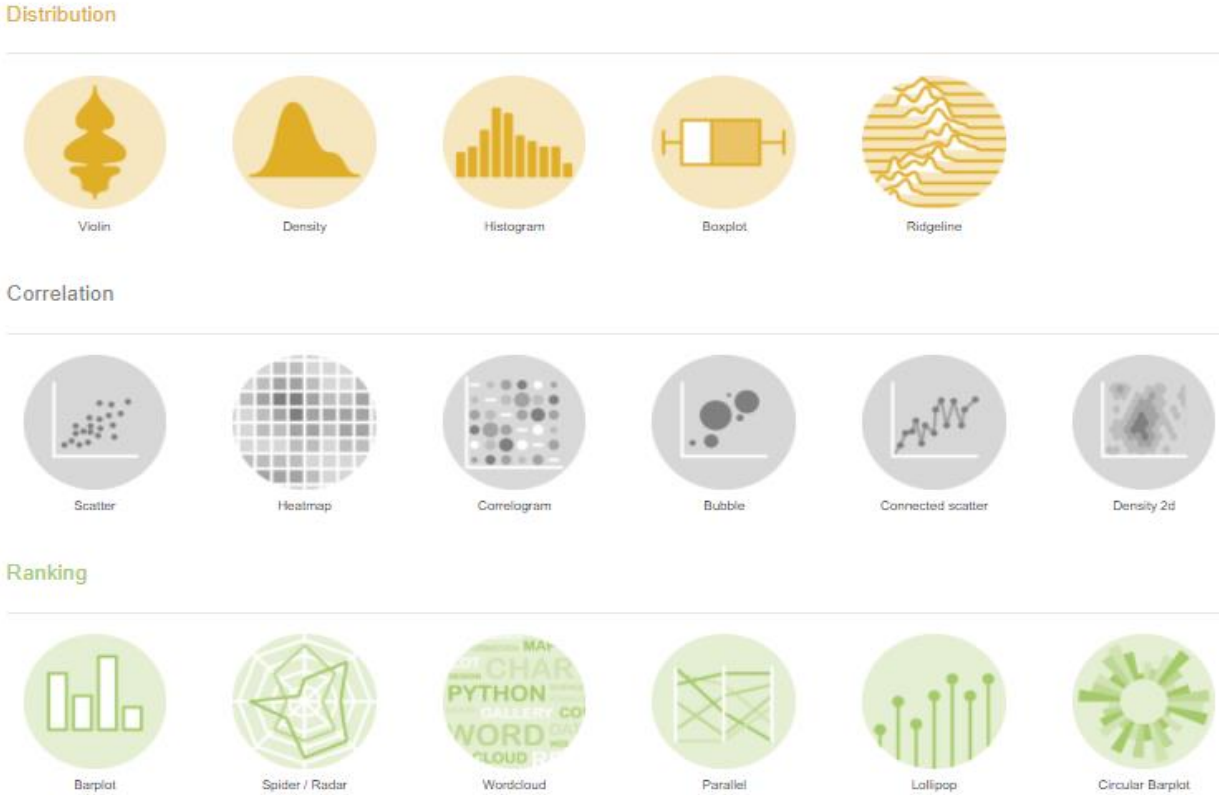

Themes
Theme functions change the appearance of your plot.
theme_bw()
White background with grid lines
theme_light()
Light axes and grid lines
theme_classic()
Classic theme, axes but no grid lines
theme_linedraw()
Only black lines
theme_dark()
Dark background for contrast
theme_minimal()
Minimal theme, no background
theme_gray()
Grey background (default theme)
theme_void()
Empty theme, only geoms are visible
count
fl

02. 데이터 시각화 실습

```
> ggsave(file="myplot.pdf")
Saving 6.92 x 4.92 in image

> ggsave(file="myplot.png",
         width = 1920, height = 1080, units = "px")
```

02. 데이터 시각화 실습

The R Graph Gallery https://www.r-graph-gallery.com/



R Gallery Book

Kyle W. Brown

2021-01-29

https://bookdown.org/content/b298e479-b1ab-49fa-b83d-a57c2b034d49/

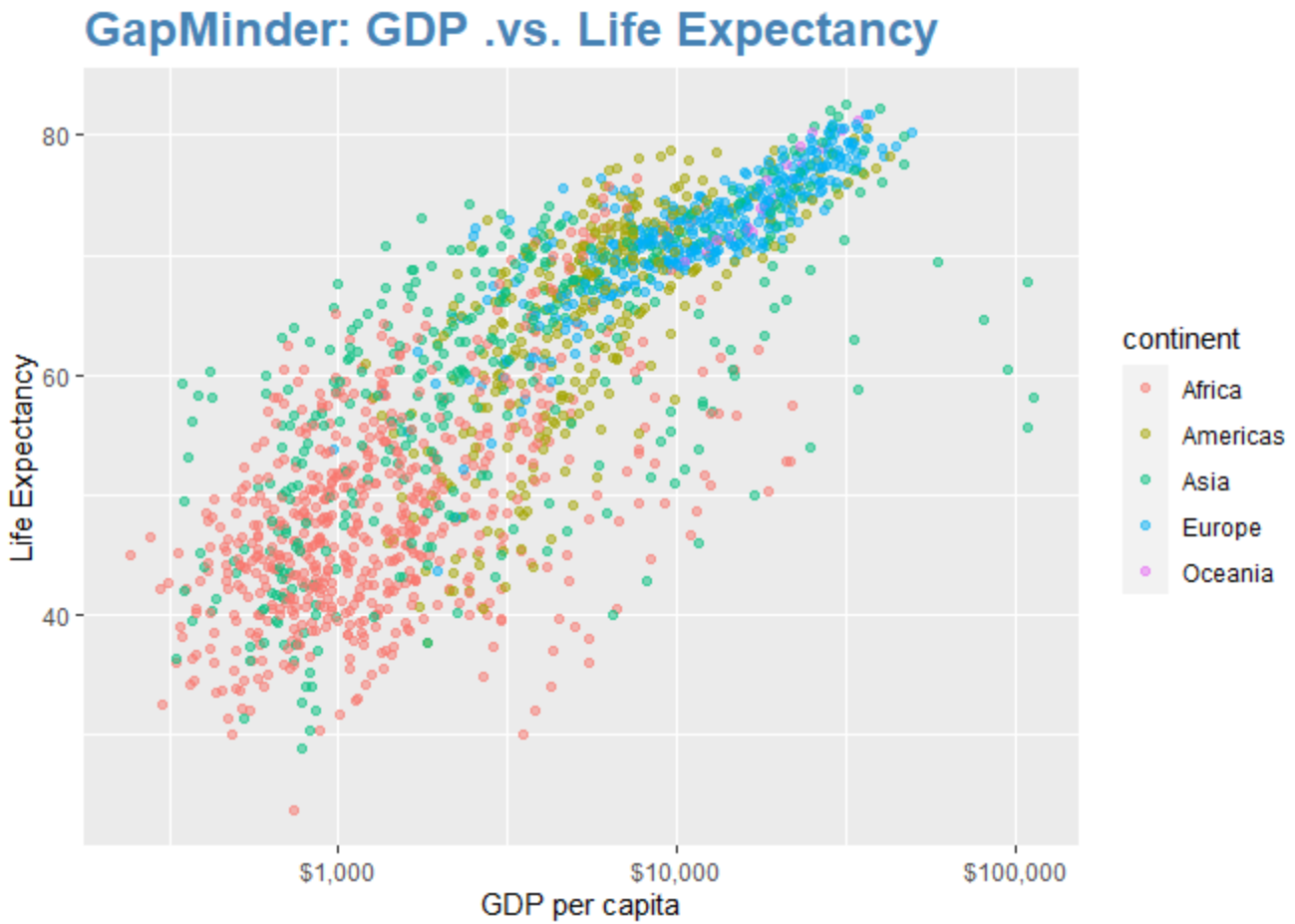
02. 데이터 시각화 실습

```
> install.packages("gapminder")
> library(gapminder)
> str(gapminder)
tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
 $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
 $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
 $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 ...
 $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

02. 데이터 시각화 실습

```
ggplot(gapminder,
       aes(gdpPercap, lifeExp, color = continent)) +
    geom_point(alpha = 0.5) +
    scale_x_log10(labels = scales::dollar) +
    labs(title = "GapMinder: GDP .vs. Life Expectancy",
         x = "GDP per capita", y = "Life Expectancy") +
    theme(plot.title = element_text(size=18,
                                    face="bold",
                                    color="steelblue"))
```

02. 데이터 시각화 실습


GapMinder: GDP .vs. Life Expectancy
Life Expectancy
80
60
40
$1,000
$10,000
$100,000
GDP per capita
continent
Africa
Americas
Asia
Europe
Oceania

02. 데이터 시각화 실습

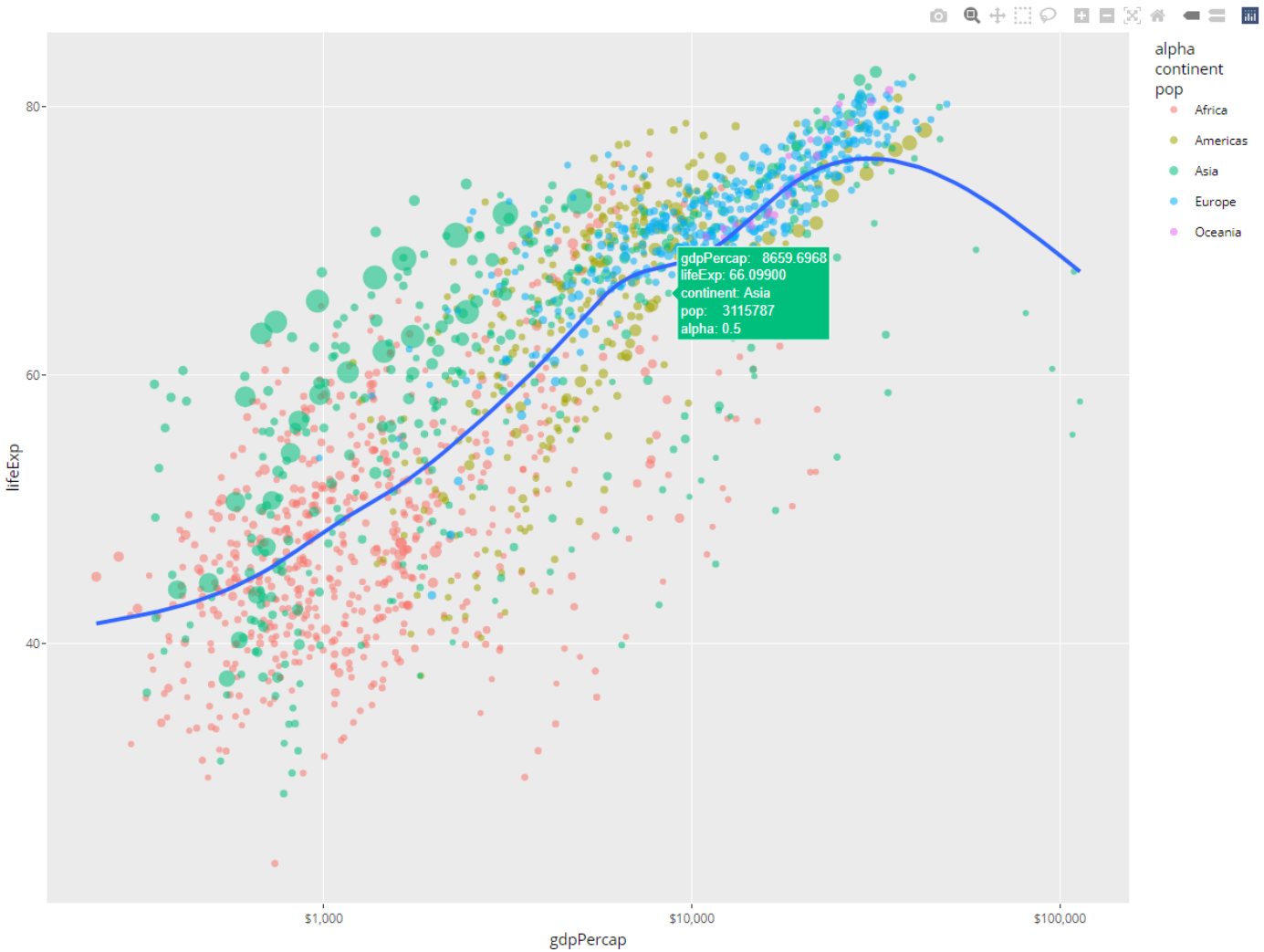
- Plotly
  - **인터렉티브 웹 그래픽을 만들 수 있는 오픈소스 라이브러리** (plotly.js)
  - R/ggplot2 **등과 연동되는 인터렉티브 그래픽을 쉽게 만들 수 있음**
  - Plotly R: https://plotly.com/r/
  - Plotly ggplot2: https://plotly.com/ggplot2/

```
> install.packages("plotly")
> library(plotly)
```

02. 데이터 시각화 실습

```
library(ggplot2)
library(plotly)
library(gapminder)

p <- ggplot(gapminder, aes(gdpPercap, lifeExp)) +
    geom_point(aes(color = continent,
                   size = pop,
                   alpha = 0.5)) +
    geom_smooth(se = F) +
    scale_x_log10(label = scales::dollar)

ggplotly(p)
```

02. 데이터 시각화 실습


alpha
continent
pop
Africa
Americas
Asia
Europe
Oceania
gdpPercap: 8659.6968
lifeExp: 66.09900
continent: Asia
pop: 3115787
alpha: 0.5
80
60
40
lifeExp
$1,000
$10,000
$100,000
gdpPercap

02. 데이터 시각화 실습

- **지도 시각화 실습:**
  - **공공데이터를 이용한 지도 시각화 – 코로나19 선별진료소**
    - https://m.post.naver.com/viewer/postView.naver?volumeNo=33299712&memberNo=25379965

    - **주제 선정: 대구시에 코로나19 선별진료소가 어디에 있을까?**
    - **데이터 수집: 공공데이터포털에서 데이터 수집**
    - **데이터 가공: 시각화에 필요한 행과 열 추출**
    - **데이터 분석: 위치 데이터 추출 및 결측치 처리**
    - **데이터 시각화: 구글맵을 이용한 선별진료소 위치 시각화**

02. 데이터 시각화 실습

```
> library(xlsx)
> mydf <- read.xlsx("선별진료소_20220216072734.xls", 1)
> str(mydf)
'data.frame':    636 obs. of  12 variables:
 $ 기준일                : chr  "2022년 02월 16일 07시" "2022년 02월 16일 07시" "2022년 02월 ...
 $ 시도                  : chr  "서울" "서울" "서울" "서울" ...
 $ 시군구                : chr  "강남구" "강남구" "강남구" "강남구" ...
 $ 의료기관명            : chr  "강남구보건소" "삼성서울병원" "강남세브란스병원" "강남베드로병원" ...
 $ 주소                  : chr  "서울 강남구 삼성동(삼성2동) 8 강남구보건소" "서울 강남구 일원로81 ...
 $ 평일.운영시간         : chr  "09:00~18:00" "09:00~16:30" "09:00~12:00" "09:00~21:00" ...
 $ 토요일.운영시간       : chr  "09:00~13:00" "09:00~16:30" "미운영" "09:00~21:00" ...
...(이하생략)
```

02. 데이터 시각화 실습

```
> mydf <- mydf[, c(2, 3, 4, 5)]
> head(mydf)
  시도 시군구       의료기관명                                                주소
1 서울 강남구     강남구보건소 서울 강남구 삼성동(삼성2동) 8 강남구보건소
2 서울 강남구     삼성서울병원           서울 강남구 일원로81 삼성서울병원
3 서울 강남구 강남세브란스병원     서울 강남구 언주로211 강남세브란스병원
4 서울 강남구   강남베드로병원                                  남부순환로2637
5 서울 강동구     강동구보건소                        서울 강동구 성내로45
6 서울 강동구     중앙보훈병원              서울 강동구 진황도로 61길 53
```

02. 데이터 시각화 실습

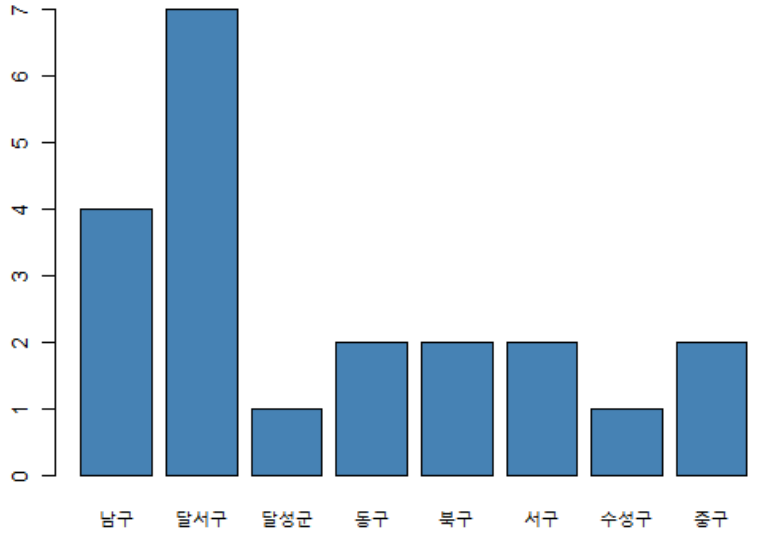
```
> names(mydf) <- c("city", "sector", "name", "addr")
> str(mydf)
'data.frame':    636 obs. of  4 variables:
 $ city  : chr  "서울" "서울" "서울" "서울" ...
 $ sector: chr  "강남구" "강남구" "강남구" "강남구" ...
 $ name  : chr  "강남구보건소" "삼성서울병원" "강남세브란스병원" "강남베드로병원" ...
 $ addr  : chr  "서울 강남구 삼성동(삼성2동) 8 강남구보건소" "서울 강남구 일원로81 삼성서울병원"...
```

02. 데이터 시각화 실습

```
> daegu <- mydf[mydf$city=="대구", ]
> head(daegu)
    city sector                    name                                    addr
119 대구 달서구          달서구보건소 대구 달서구 월성동(월성2동) 281 달서구청
120 대구 달서구 계명대학교 동산병원                 대구 달서구 달구벌대로 1035
121 대구 달서구               삼일병원                      대구 달서구 월배로 436
122 대구 달서구                 구병원                   대구 달서구 감삼북길 141
123 대구 달서구               세강병원                      대구 달서구 구마로 220
124 대구 달서구                  W병원                 대구 달서구 달구벌대로 1632
```

02. 데이터 시각화 실습

```
> nrow(daegu)
[1] 21
> table(daegu$sector)
   남구 달서구 달성군   동구   북구   서구  수성구   중구
     4      7      1      2      2      2      1      2
> barplot(table(daegu$sector), col = "steelblue")
```



02. 데이터 시각화 실습

```
library(ggmap)

ggmap_key <- "your API Key here"
register_google(ggmap_key)

## geocode 가져오기
daegu.loc <- mutate_geocode(data=daegu,
                            location=addr,
                            source="google")
```

02. 데이터 시각화 실습

```
> daegu.loc[, c("lon", "lat", "name")]

> daegu.loc <- na.omit(daegu.loc)
> daegu.loc[, c("lon", "lat", "name")]
        lon      lat                                  name
1  128.5326 35.82973                             달서구보건소
2  128.5203 35.85320                  계명대학교 동산병원
3  128.5536 35.83249                             삼일병원
4  128.5414 35.85065                               구병원
5  128.5517 35.83667                             세강병원
6  128.5203 35.85320                               W병원
7  128.5518 35.80290                          대구보훈병원
8  128.4447 35.69213                         달성군 보건소
9  128.5918 35.85382                 대구광역시 남구보건소
10 128.5828 35.84696                       영남대학교 병원
...(이하 생략)
```

02. 데이터 시각화 실습

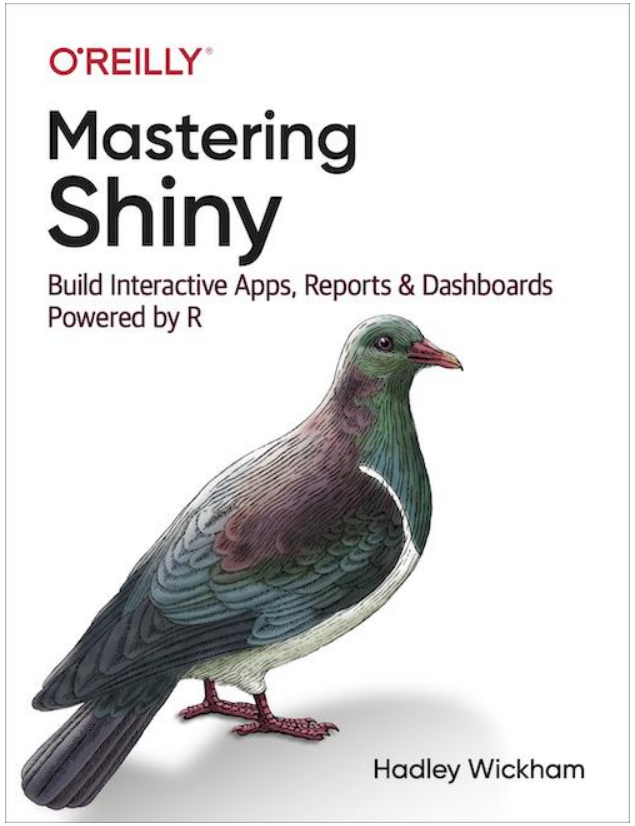
```
# 지도정보 표시하기
daegu.map <- get_googlemap(c("대구"), maptype = "roadmap", zoom = 11)
ggmap(daegu.map)

ggmap(daegu.map) +
    geom_point(data = daegu.loc,
               aes(x = lon, y = lat, color = factor(name)),
               size = 3)
```

02. 데이터 시각화 실습

```
# 지도 위에 마커 표시하기
marker <- data.frame(daegu.loc$lon, daegu.loc$lat)
daegu.map <- get_googlemap(c("대구"),
                           maptype = "roadmap",
                           zoom = 11,
                           markers=marker)
ggmap(daegu.map) +
    geom_text(data = daegu.loc,
              aes(x = lon, y = lat),
              size = 3,
              label = daegu.loc$name)

daegu.loc$name
```

02. 데이터 시각화 실습

- Mastering Shiny
  - https://mastering-shiny.org/
  - https://shiny.rstudio.com/tutorial/written-tutorial/lesson1/



02. 데이터 시각화 실습

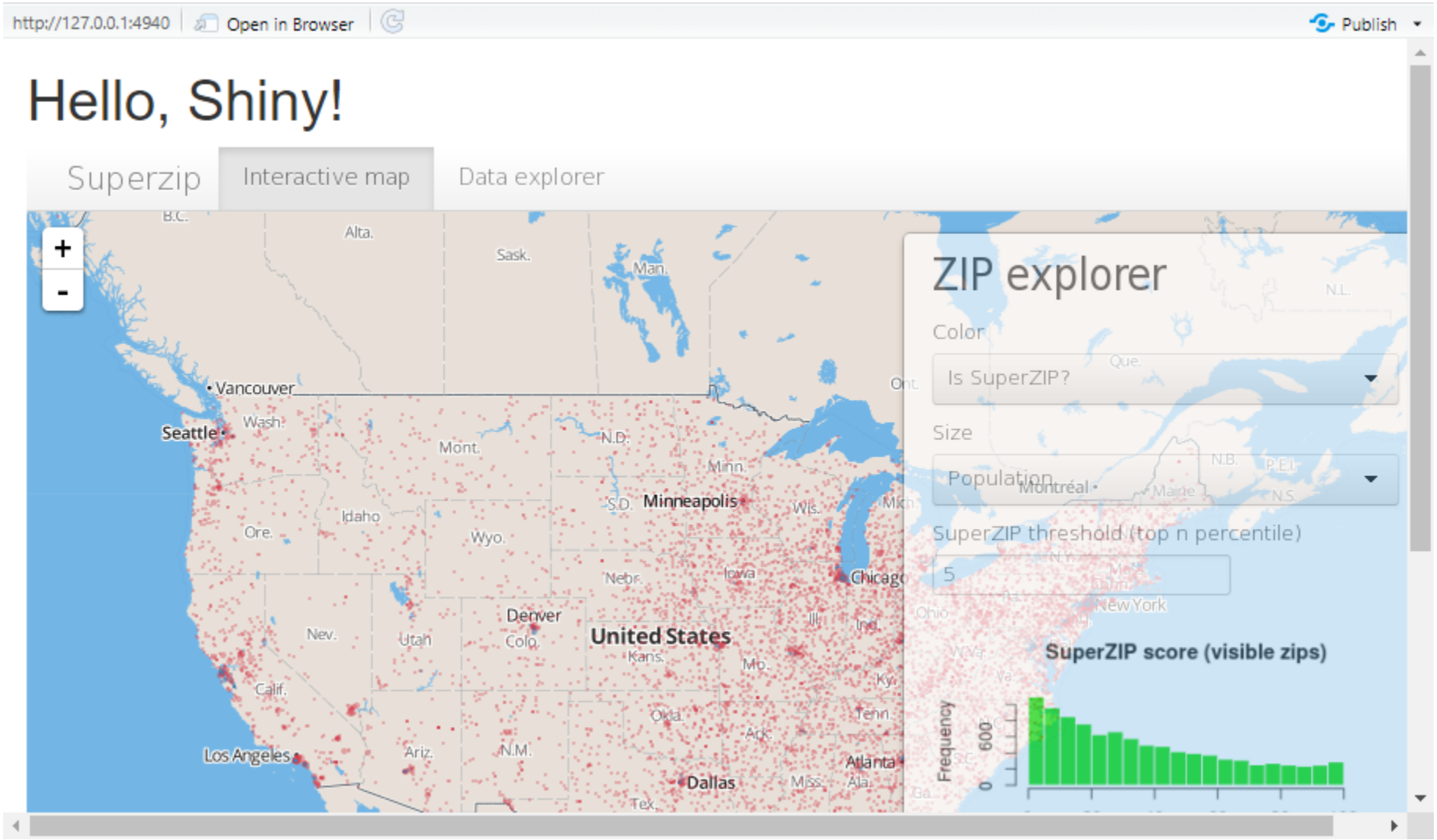
- Shiny
  - R 코드로 웹 애플리케이션을 개발할 수 있는 프레임워크
  - 웹 애플리케이션 개발 지식이 없어도 웹앱을 개발할 수 있다는 장점이 있음

```
> install.packages("shiny")
> library(shiny)
```

02. 데이터 시각화 실습

```
# shiny.1.R

ui <- fluidPage(
    tags$h1("Hello, Shiny!"),
    tags$img(src="https://shiny.rstudio.com/gallery/images/screenshots/superzip-example.png")
)

server <- function (input, output, session) {
    # Do something here!
}

shinyApp(ui = ui, server = server)
```

02. 데이터 시각화 실습


http://127.0.0.1:4940
Open in Browser
Publish
Hello, Shiny!
Superzip
Interactive map
Data explorer
ZIP explorer
Color
Is SuperZIP?
Size
Population
SuperZIP threshold (top n percentile)
5
SuperZIP score (visible zips)
Frequency
0
600
Vancouver
Seattle
Minneapolis
Chicago
Denver
United States
Los Angeles
Dallas
Atlanta
New York
Montréal

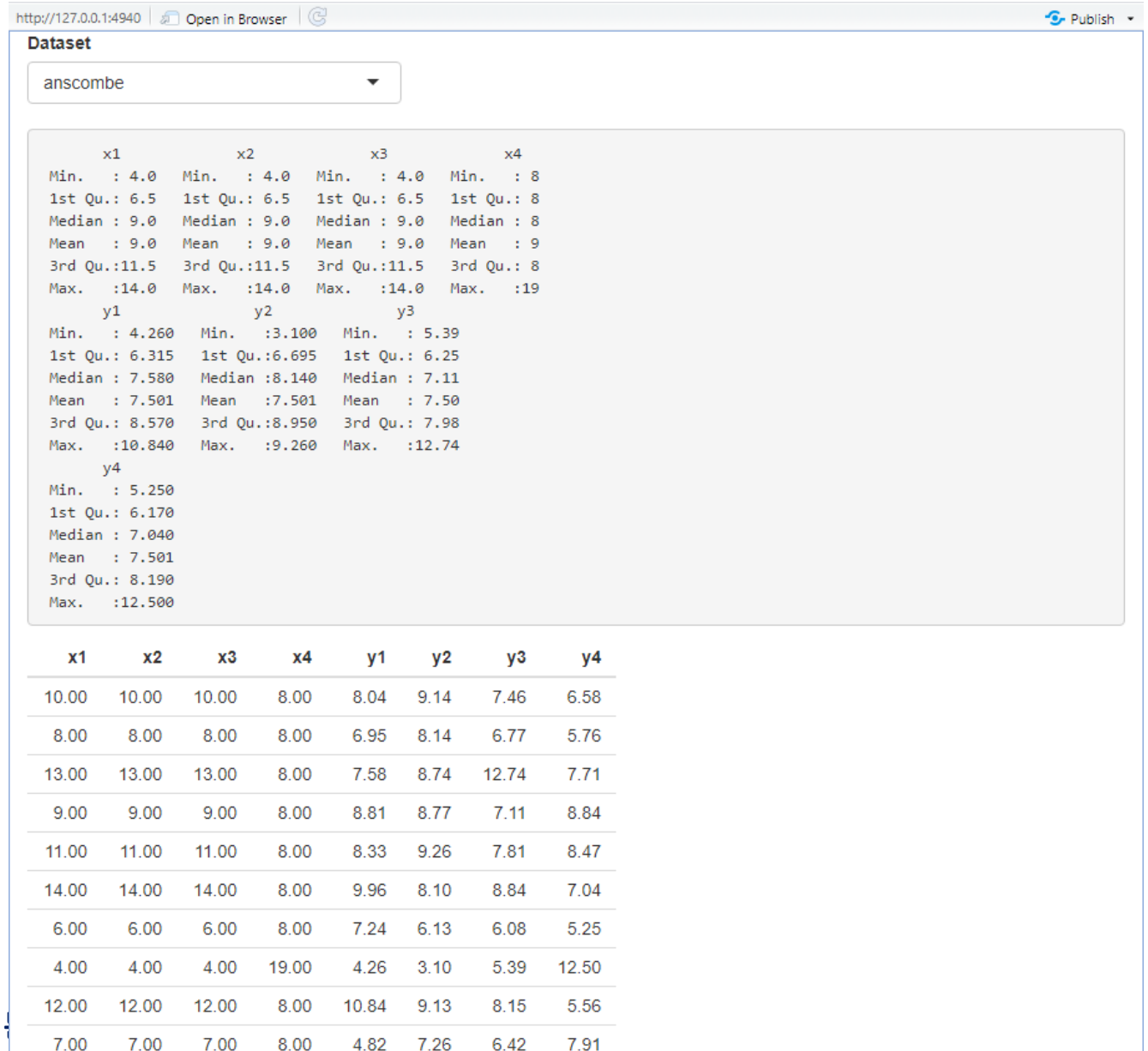
02. 데이터 시각화 실습

```
# ./shiny.1/ui.R

ui <- fluidPage(
    selectInput("dataset",
                label = "Dataset",
                choices = ls("package:datasets")),
    verbatimTextOutput("summary"),
    tableOutput("table")
)
```

02. 데이터 시각화 실습

```
# ./shiny.1/server.R

server <- function(input, output) {

    output$summary <- renderPrint({
        dataset <- get(input$dataset, "package:datasets")
        summary(dataset)
    })

    output$table <- renderTable({
        dataset <- get(input$dataset, "package:datasets")
        dataset
    })
}
```

02. 데이터 시각화 실습


http://127.0.0.1:4940
Open in Browser
Publish
Dataset
anscombe
x1 x2 x3 x4
Min. : 4.0 Min. : 4.0 Min. : 4.0 Min. : 8
1st Qu.: 6.5 1st Qu.: 6.5 1st Qu.: 6.5 1st Qu.: 8
Median : 9.0 Median : 9.0 Median : 9.0 Median : 8
Mean : 9.0 Mean : 9.0 Mean : 9.0 Mean : 9
3rd Qu.:11.5 3rd Qu.:11.5 3rd Qu.:11.5 3rd Qu.: 8
Max. :14.0 Max. :14.0 Max. :14.0 Max. :19
y1 y2 y3
Min. : 4.260 Min. :3.100 Min. : 5.39
1st Qu.: 6.315 1st Qu.:6.695 1st Qu.: 6.25
Median : 7.580 Median :8.140 Median : 7.11
Mean : 7.501 Mean :7.501 Mean : 7.50
3rd Qu.: 8.570 3rd Qu.:8.950 3rd Qu.: 7.98
Max. :10.840 Max. :9.260 Max. :12.74
y4
Min. : 5.250
1st Qu.: 6.170
Median : 7.040
Mean : 7.501
3rd Qu.: 8.190
Max. :12.500
x1 x2 x3 x4 y1 y2 y3 y4
10.00 10.00 10.00 8.00 8.04 9.14 7.46 6.58
8.00 8.00 8.00 8.00 6.95 8.14 6.77 5.76
13.00 13.00 13.00 8.00 7.58 8.74 12.74 7.71
9.00 9.00 9.00 8.00 8.81 8.77 7.11 8.84
11.00 11.00 11.00 8.00 8.33 9.26 7.81 8.47
14.00 14.00 14.00 8.00 9.96 8.10 8.84 7.04
6.00 6.00 6.00 8.00 7.24 6.13 6.08 5.25
4.00 4.00 4.00 19.00 4.26 3.10 5.39 12.50
12.00 12.00 12.00 8.00 10.84 9.13 8.15 5.56
7.00 7.00 7.00 8.00 4.82 7.26 6.42 7.91

02. 데이터 시각화 실습

```
# shiny.2/server.R

server <- function(input, output) {

    # Create a reactive expression
    dataset <- reactive({
        get(input$dataset, "package:datasets")
    })

    output$summary <- renderPrint({
        # Use a reactive expression by calling it like a function
        summary(dataset())
    })

    output$table <- renderTable({
        dataset()
    })
}
```

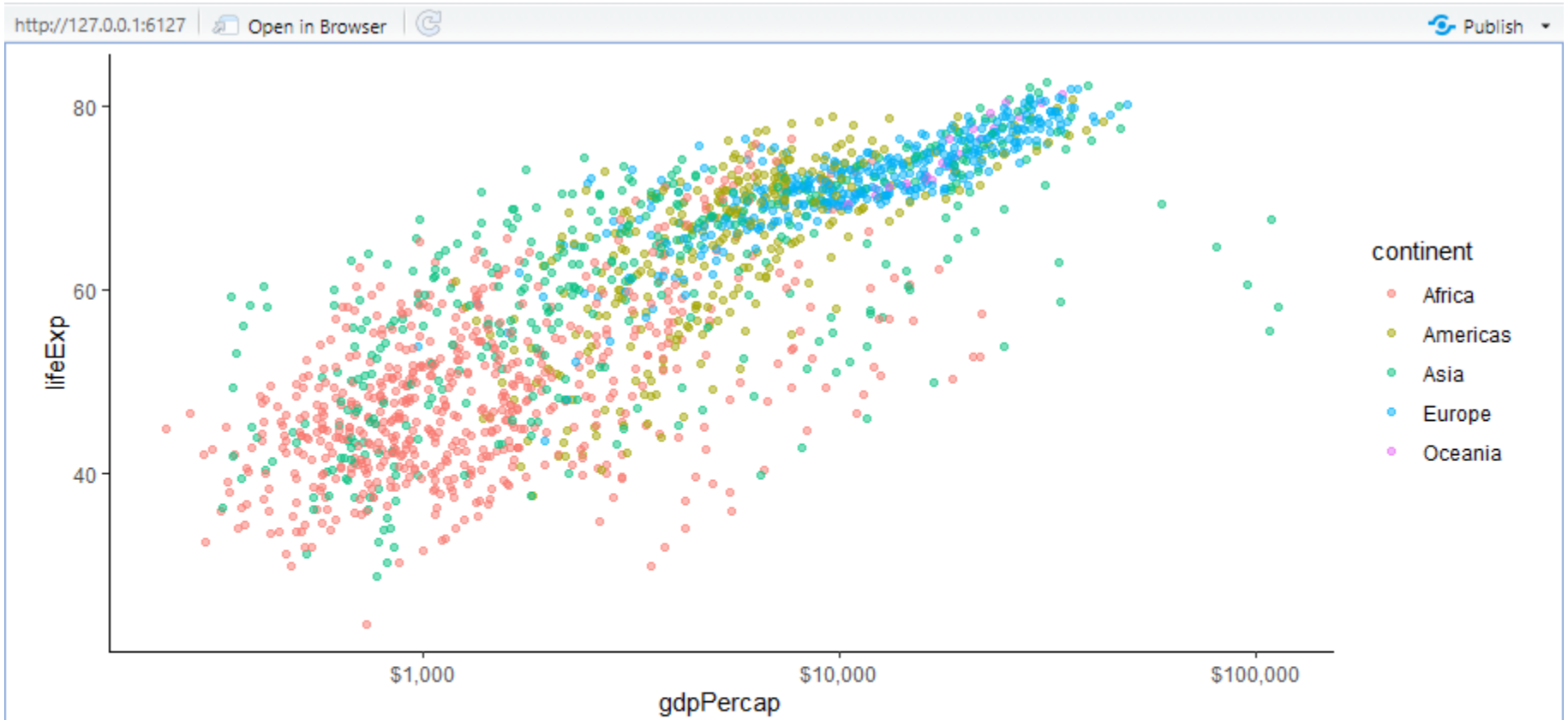
02. 데이터 시각화 실습

```
library(shiny)
library(ggplot2)
library(gapminder)

ui <- fluidPage(
    plotOutput("plot", click = "plot_click"),
    tableOutput("data")
)
```

02. 데이터 시각화 실습

```
server <- function(input, output, session) {
    output$plot <- renderPlot({
        ggplot(gapminder, aes(gdpPercap, lifeExp,
                              color = continent)) +
            geom_point(alpha = 0.5) +
            scale_x_log10(labels = scales::dollar) +
            theme_classic()
    }, res = 96)

    output$data <- renderTable({
        req(input$plot_click)
        nearPoints(gapminder, input$plot_click)
    })
}
```

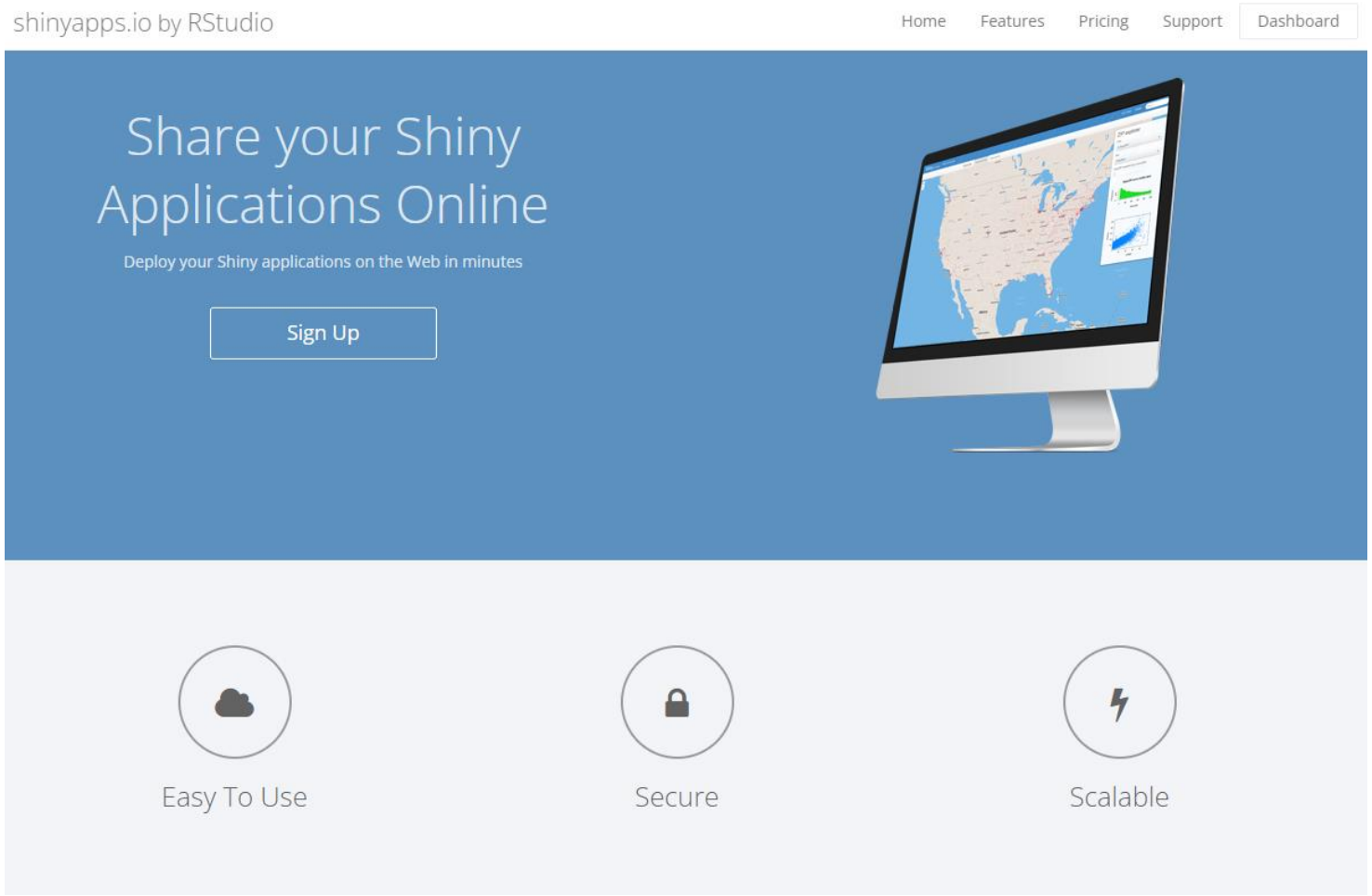
02. 데이터 시각화 실습



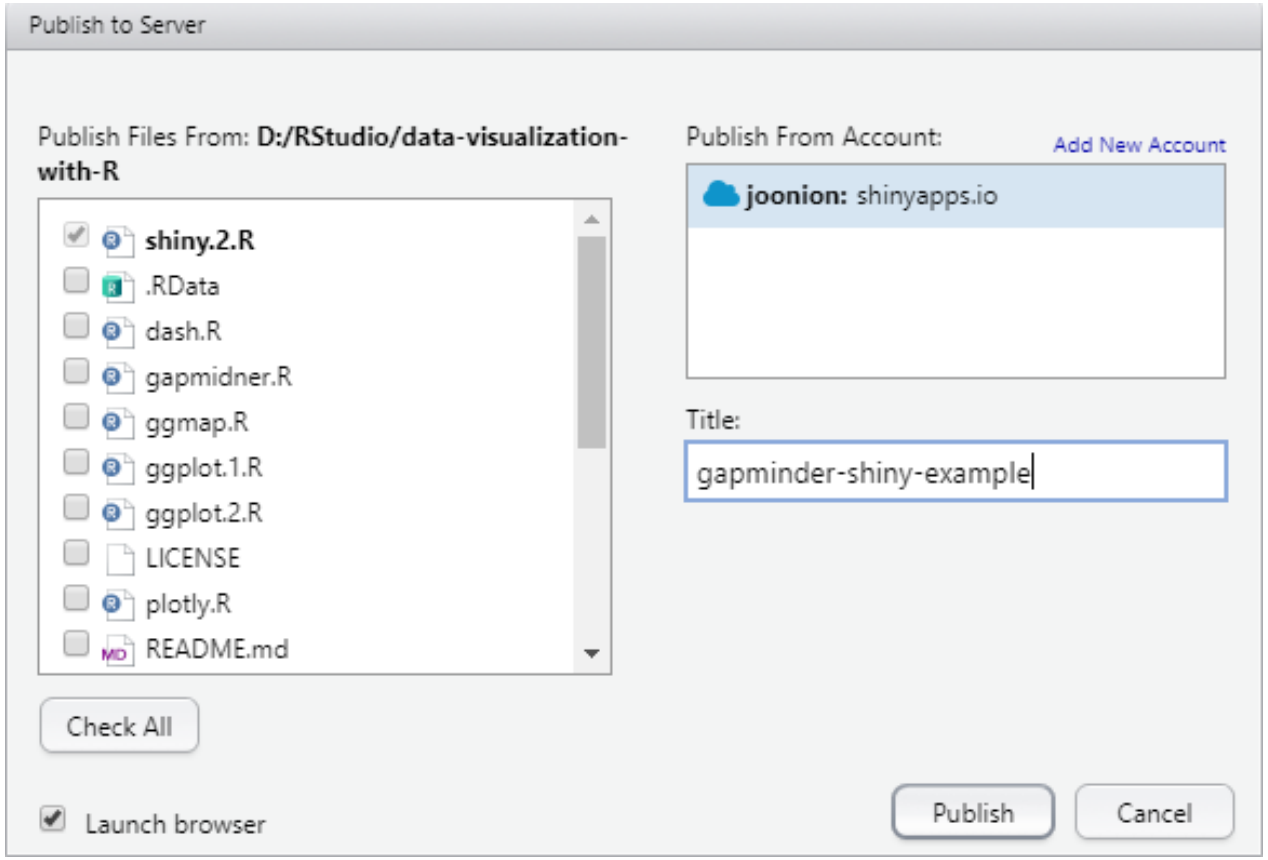
country	continent	year	lifeExp	pop	gdpPercap
Argentina	Americas	1972	67.06	24779799	9443.04
Mexico	Americas	1982	67.41	71640904	9611.15
Austria	Europe	1957	67.48	6965860	8842.60

02. 데이터 시각화 실습

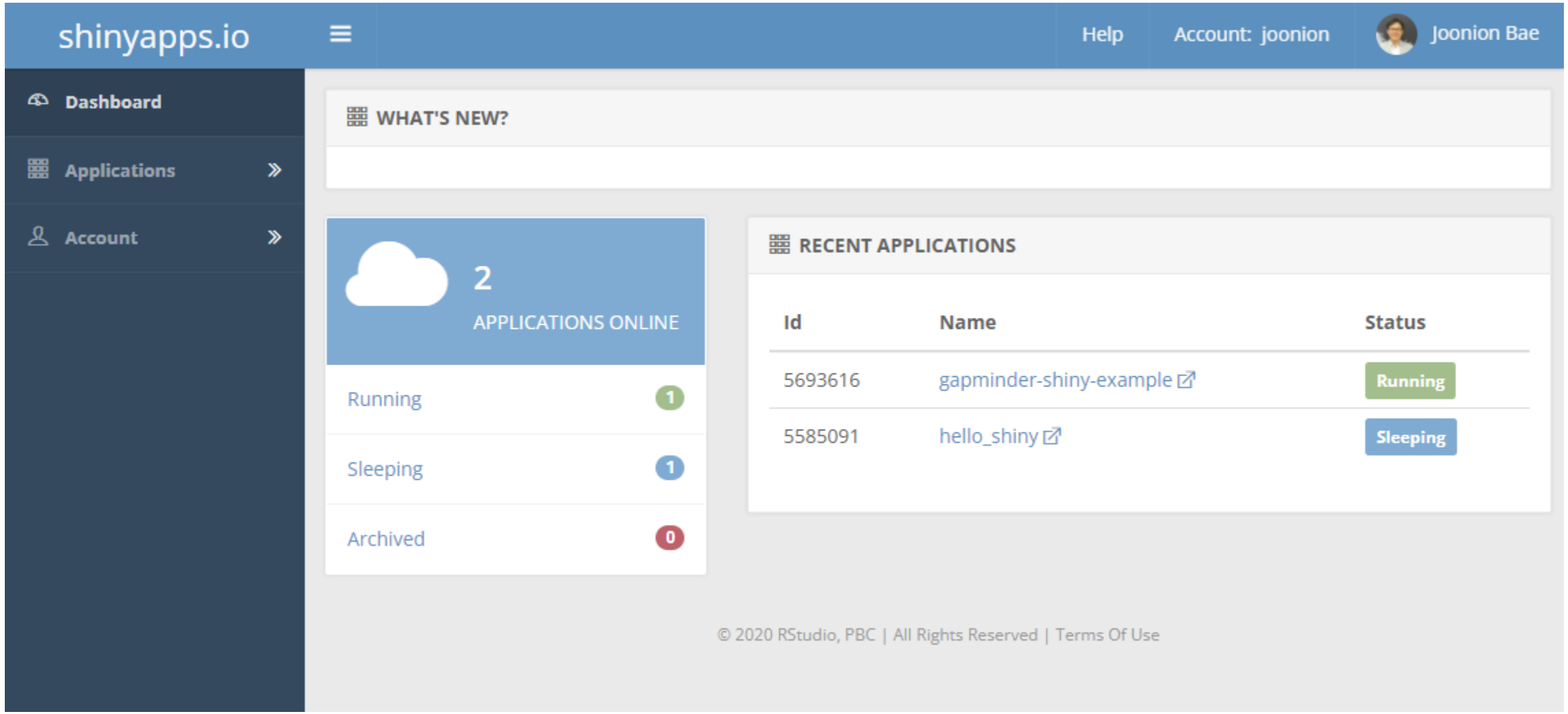
- **제작한 웹앱을 Publish하기:**
  - shinyapps.io: **회원가입**



02. 데이터 시각화 실습

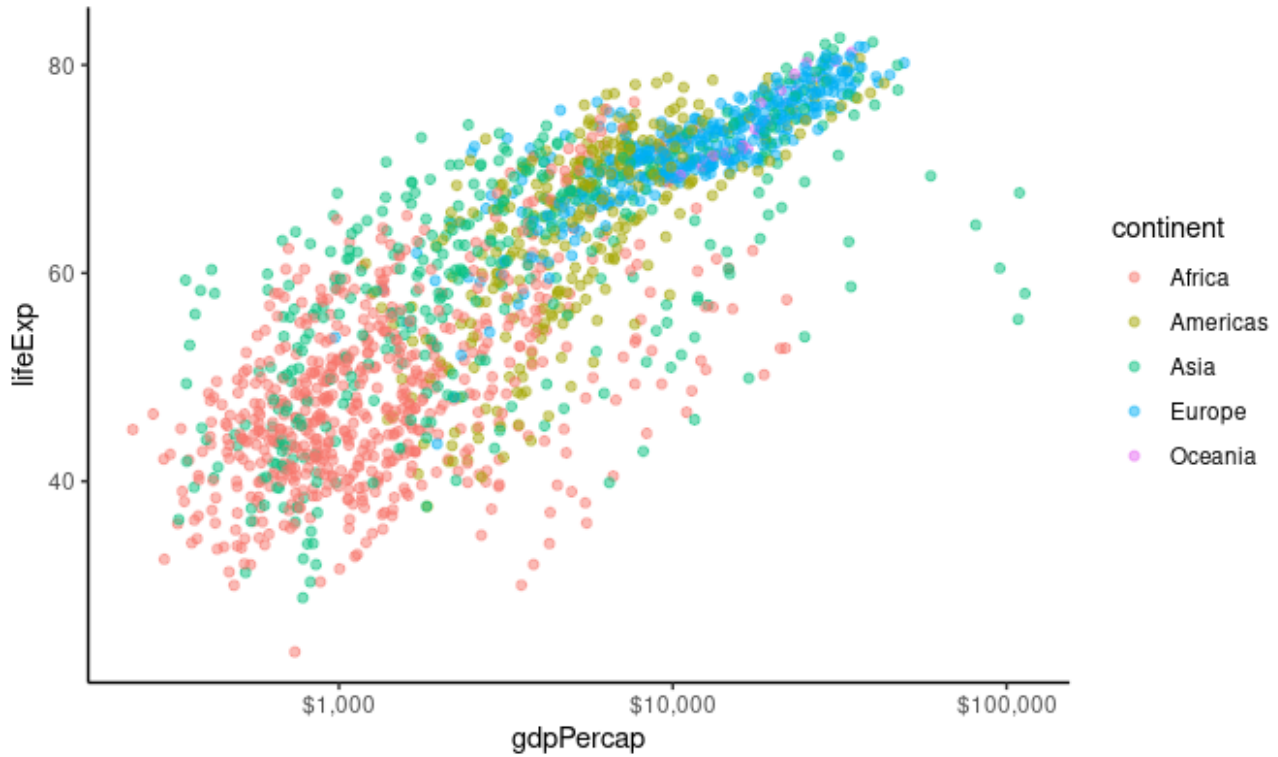

Publish to Server
Publish Files From: D:/RStudio/data-visualization-with-R
shiny.2.R
.RData
dash.R
gapmidner.R
ggmap.R
ggplot.1.R
ggplot.2.R
LICENSE
plotly.R
README.md
Check All
Launch browser
Publish From Account:
Add New Account
joonion: shinyapps.io
Title:
gapminder-shiny-example
Publish
Cancel

02. 데이터 시각화 실습


shinyapps.io
Help
Account: joonion
Joonion Bae
Dashboard
Applications
Account
WHAT'S NEW?
2
APPLICATIONS ONLINE
Running 1
Sleeping 1
Archived 0
RECENT APPLICATIONS
Id
Name
Status
5693616
gapminder-shiny-example
Running
5585091
hello_shiny
Sleeping
© 2020 RStudio, PBC | All Rights Reserved | Terms Of Use

02. 데이터 시각화 실습

https://joonion.shinyapps.io/gapminder-shiny-example/

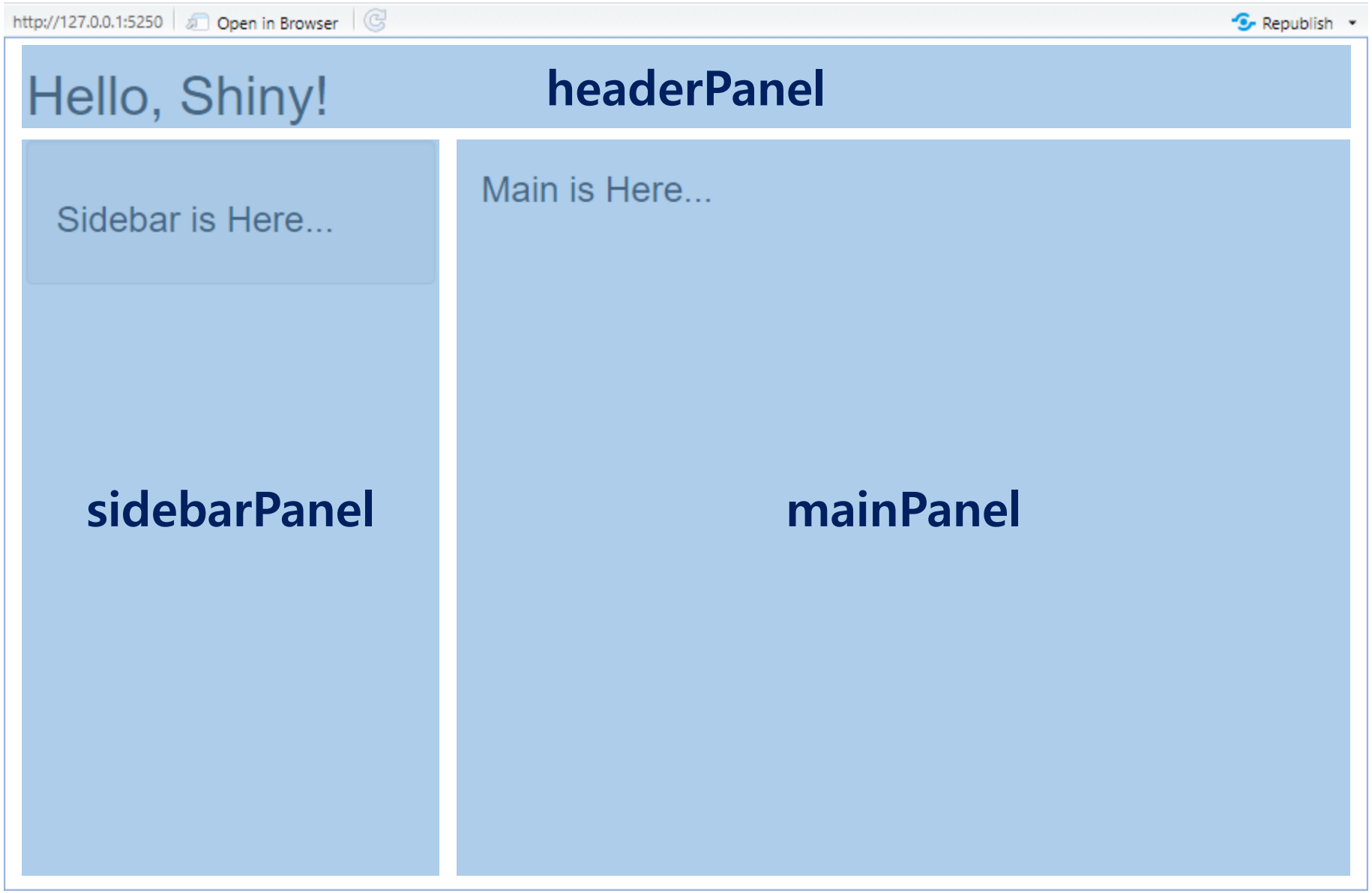


country	continent	year	lifeExp	pop	gdpPercap
Swaziland	Africa	2007	39.61	1133066	4513.48

02. 데이터 시각화 실습

```
ui <- pageWithSidebar(

    # 1. 헤더 패널
    headerPanel(
        h1("Hello, Shiny!")),

    # 2. 사이드바 패널
    sidebarPanel(
        h3("Sidebar is Here...")),

    # 3. 메인 패널
    mainPanel(
        h3("Main is Here..."))
)
```

02. 데이터 시각화 실습


http://127.0.0.1:5250
Open in Browser
Republish
Hello, Shiny!
headerPanel
Sidebar is Here...
Main is Here...
sidebarPanel
mainPanel

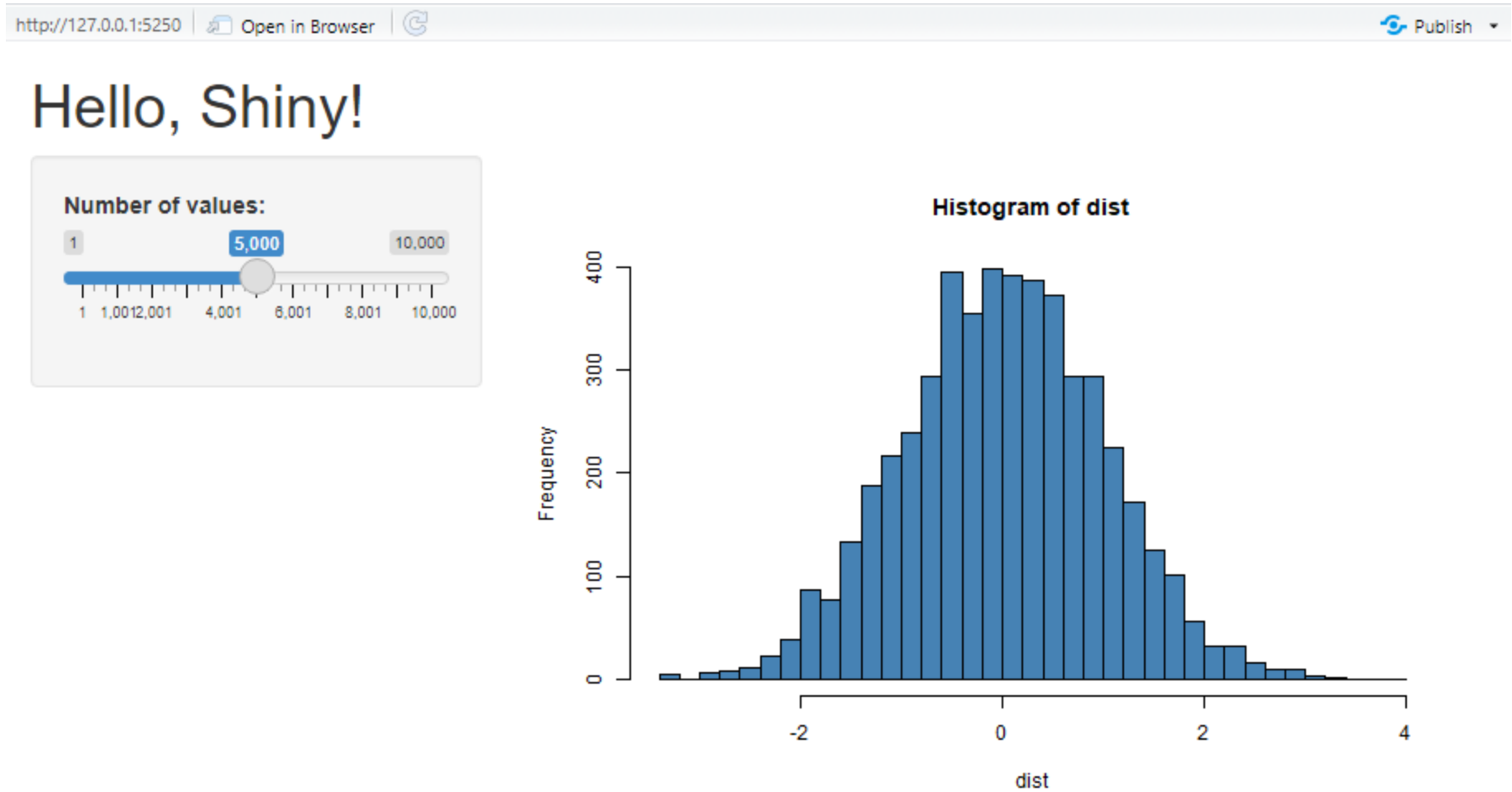
02. 데이터 시각화 실습

```
ui <- pageWithSidebar(
    # 1. 헤더 패널
    headerPanel(h1("Hello, Shiny!")),
    # 2. 사이드바 패널
    sidebarPanel(
        sliderInput("count",
                    "Number of values: ",
                    min = 1,
                    max = 10000,
                    value = 5000)),
    # 3. 메인 패널
    mainPanel(
        plotOutput("distPlot"))
)
```

02. 데이터 시각화 실습

```
server <- function(input, output) {
    output$distPlot <- renderPlot({
        dist <- rnorm(input$count)
        hist(dist, col = "steelblue", breaks = 50)
    })
}
```

02. 데이터 시각화 실습


http://127.0.0.1:5250
Open in Browser
Publish
Hello, Shiny!
Number of values:
1
5,000
10,000
1
1,001
2,001
4,001
6,001
8,001
10,000
Histogram of dist
Frequency
0
100
200
300
400
-2
0
2
4
dist

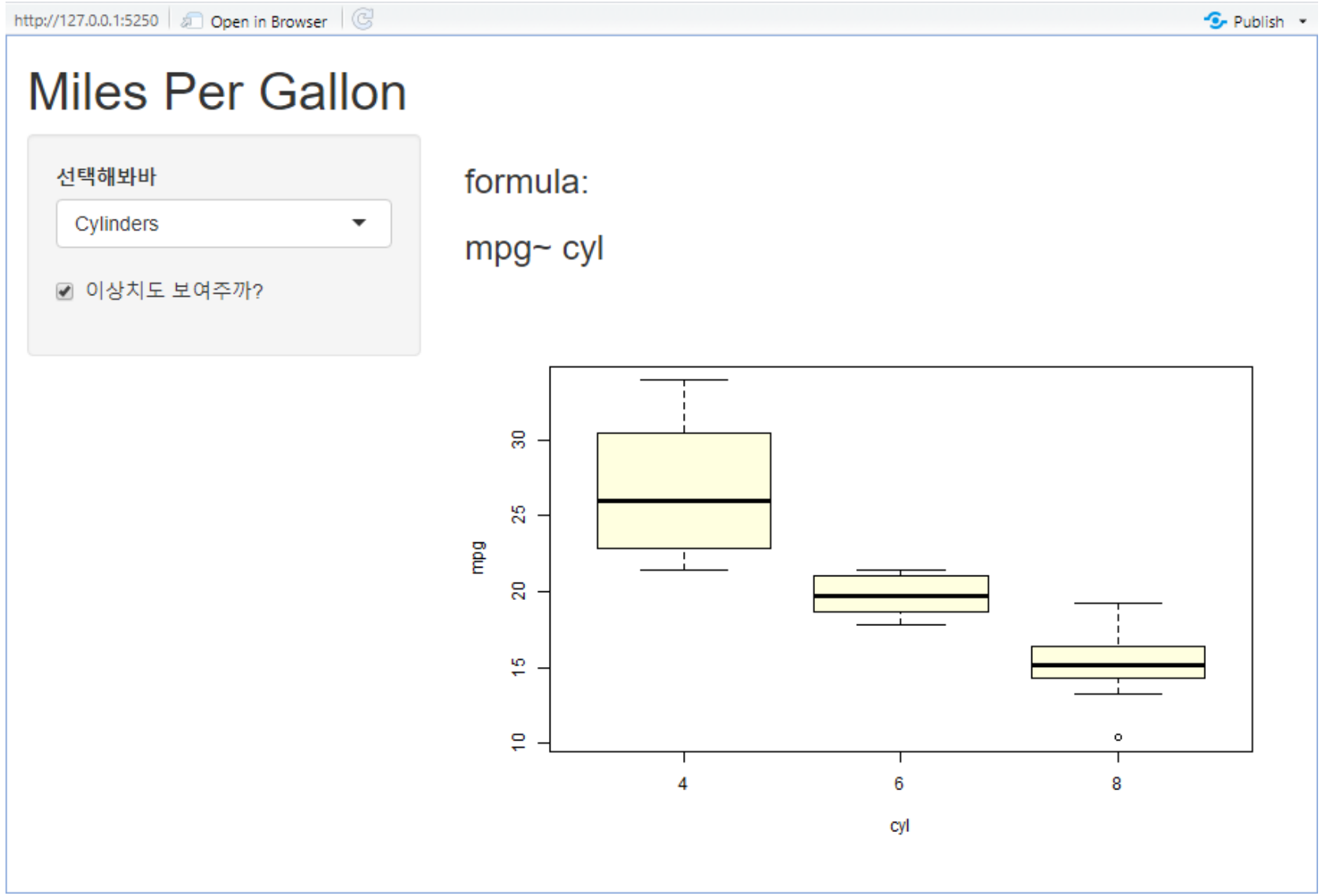
02. 데이터 시각화 실습

```
ui <- pageWithSidebar(
    headerPanel(h1("Miles Per Gallon")),

    sidebarPanel(
        selectInput("variable",
                    "선택해봐바",
                    list("Cylinders" = "cyl",
                         "Transmission" = "am",
                         "Gears" = "gear")),
        checkboxInput("outliers",
                      "이상치도 보여주까?",
                      FALSE)),


    mainPanel(
        h3("formula: "),
        h3(textOutput("caption")),
        plotOutput("mpgPlot"))
)
```

02. 데이터 시각화 실습

```
mpgData <- mtcars
mpgData$am <- factor(mpgData$am, labels = c("Automatic", "Manual"))

server <- function(input, output) {

    formulaText <- reactive({
        paste("mpg~", input$variable)
    })

    output$caption <- renderText({
        formulaText()
    })

    output$mpgPlot <- renderPlot({
        boxplot(as.formula(formulaText()),
                data <- mpgData,
                outline = input$outliers,
                col = "lightyellow")
    })
}
```

02. 데이터 시각화 실습


http://127.0.0.1:5250
Open in Browser
Publish
Miles Per Gallon
선택해봐바
Cylinders
이상치도 보여주까?
formula:
mpg~ cyl
mpg
10
15
20
25
30
4
6
8
cyl

02. 데이터 시각화 실습

```
server <- function (input, output) {
    output$mytable1 <- renderDataTable({
        diamonds[, input$showvars, drop = FALSE]
    })
    output$mytable2 <- renderDataTable({
        mtcars
    }, options = list(bSortClasses = TRUE))
    output$mytable3 <- renderDataTable({
        iris
    }, options = list(aLengthMenu = c(5, 30, 50),
                      iDisplayLength = 5))
}
```

02. 데이터 시각화 실습

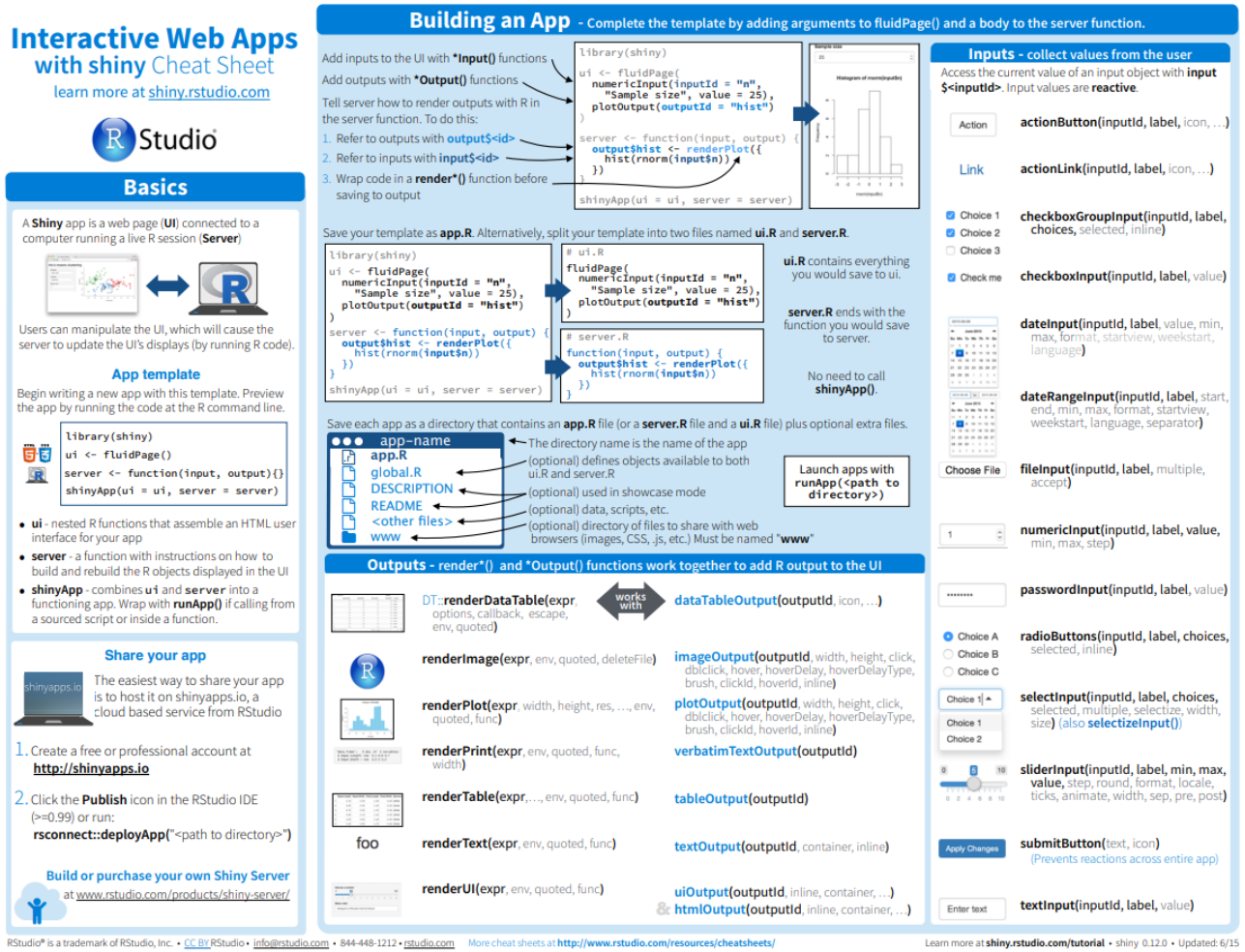
```
ui <- pageWithSidebar(
    headerPanel(h1("데이터테이블 예제")),
    sidebarPanel(
        checkboxGroupInput("showvars",
                           "컬럼을 선택해보세요:",
                           names(diamonds),
                           selected = names(diamonds)),
        helpText("오른쪽에서 탭을 선택하면 다른 데이터도 볼 수 있음.")
    ),
    mainPanel(
        tabsetPanel(
            tabPanel("diamonds",
                     dataTableOutput("mytable1")),
            tabPanel("mtcars",
                     dataTableOutput("mytable2")),
            tabPanel("iris",
                     dataTableOutput("mytable3")),
        )
    )
)
```

02. 데이터 시각화 실습


http://127.0.0.1:5250
Open in Browser
Publish
데이터테이블 예제
컬럼을 선택해보세요:
carat
cut
color
clarity
depth
table
price
x
y
z
오른쪽에서 탭을 선택하면 다른 데이터도 볼 수 있음.
diamonds
mtcars
iris
Show 25 entries
Search:
mpg cyl disp hp drat wt qsec vs am gear carb
21 6 160 110 3.9 2.62 16.46 0 1 4 4
21 6 160 110 3.9 2.875 17.02 0 1 4 4
22.8 4 108 93 3.85 2.32 18.61 1 1 4 1
21.4 6 258 110 3.08 3.215 19.44 1 0 3 1
18.7 8 360 175 3.15 3.44 17.02 0 0 3 2
18.1 6 225 105 2.76 3.46 20.22 1 0 3 1
14.3 8 360 245 3.21 3.57 15.84 0 0 3 4
24.4 4 146.7 62 3.69 3.19 20 1 0 4 2
22.8 4 140.8 95 3.92 3.15 22.9 1 0 4 2
19.2 6 167.6 123 3.92 3.44 18.3 1 0 4 4
17.8 6 167.6 123 3.92 3.44 18.9 1 0 4 4

02. 데이터 시각화 실습

https://shiny.rstudio.com/articles/cheatsheet.html



Interactive Web Apps with shiny Cheat Sheet

learn more at shiny.rstudio.com

RStudio

Basics

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)

Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

App template

Begin writing a new app with this template. Preview the app by running the code at the R command line.

```
library(shiny)
ui <- fluidPage()
server <- function(input, output){}
shinyApp(ui = ui, server = server)
```

- **ui** - nested R functions that assemble an HTML user interface for your app
- **server** - a function with instructions on how to build and rebuild the R objects displayed in the UI
- **shinyApp** - combines **ui** and **server** into a functioning app. Wrap with **runApp()** if calling from a sourced script or inside a function.

Share your app

shinyapps.io

The easiest way to share your app is to host it on shinyapps.io, a cloud based service from RStudio

1. Create a free or professional account at **http://shinyapps.io**
2. Click the **Publish** icon in the RStudio IDE (>=0.99) or run: **rsconnect::deployApp("<path to directory>")**

Build or purchase your own Shiny Server

at www.rstudio.com/products/shiny-server/

Building an App - Complete the template by adding arguments to fluidPage() and a body to the server function.

Add inputs to the UI with ***Input()** functions

Add outputs with ***Output()** functions

Tell server how to render outputs with R in the server function. To do this:

1. Refer to outputs with **output$<id>**
2. Refer to inputs with **input$<id>**
3. Wrap code in a **render*()** function before saving to output

```
library(shiny)
ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)
server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
shinyApp(ui = ui, server = server)
```

Save your template as **app.R**. Alternatively, split your template into two files named **ui.R** and **server.R**.

```
library(shiny)
ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)
server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
shinyApp(ui = ui, server = server)
```

```
# ui.R
fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)
```

ui.R contains everything you would save to ui.

```
# server.R
function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
```

server.R ends with the function you would save to server.

No need to call **shinyApp()**.

Save each app as a directory that contains an **app.R** file (or a **server.R** file and a **ui.R** file) plus optional extra files.

app-name
- app.R
- global.R — The directory name is the name of the app
- DESCRIPTION — (optional) defines objects available to both ui.R and server.R
- README — (optional) used in showcase mode
- <other files> — (optional) data, scripts, etc.
- www — (optional) directory of files to share with web browsers (images, CSS, .js, etc.) Must be named "**www**"

Launch apps with `runApp(<path to directory>)`

Outputs - render*() and *Output() functions work together to add R output to the UI

	works with	
DT::**renderDataTable**(expr, options, callback, escape, env, quoted)		**dataTableOutput**(outputId, icon, ...)
renderImage(expr, env, quoted, deleteFile)		**imageOutput**(outputId, width, height, click, dbclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
renderPlot(expr, width, height, res, ..., env, quoted, func)		**plotOutput**(outputId, width, height, click, dbclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
renderPrint(expr, env, quoted, func, width)		**verbatimTextOutput**(outputId)
renderTable(expr,..., env, quoted, func)		**tableOutput**(outputId)
foo **renderText**(expr, env, quoted, func)		**textOutput**(outputId, container, inline)
renderUI(expr, env, quoted, func)		**uiOutput**(outputId, inline, container, ...) & **htmlOutput**(outputId, inline, container, ...)

Inputs - collect values from the user

Access the current value of an input object with **input $<inputId>**. Input values are **reactive**.

Action	**actionButton**(inputId, label, icon, ...)
Link	**actionLink**(inputId, label, icon, ...)
Choice 1, Choice 2, Choice 3	**checkboxGroupInput**(inputId, label, **choices**, selected, inline)
Check me	**checkboxInput**(inputId, label, value)
	dateInput(inputId, label, value, min, max, format, startview, weekstart, language)
	dateRangeInput(inputId, label, start, end, min, max, format, startview, weekstart, language, separator)
Choose File	**fileInput**(inputId, label, multiple, accept)
1	**numericInput**(inputId, label, value, min, max, step)
•••••••	**passwordInput**(inputId, label, value)
Choice A, Choice B, Choice C	**radioButtons**(inputId, label, choices, selected, inline)
Choice 1, Choice 1, Choice 2	**selectInput**(inputId, label, choices, selected, multiple, selectize, width, size) (also **selectizeInput**())
0, 5, 10	**sliderInput**(inputId, label, min, max, **value**, step, round, format, locale, ticks, animate, width, sep, pre, post)
Apply Changes	**submitButton**(text, icon) (Prevents reactions across entire app)
Enter text	**textInput**(inputId, label, value)

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com

More cheat sheets at http://www.rstudio.com/resources/cheatsheets/

Learn more at shiny.rstudio.com/tutorial • shiny 0.12.0 • Updated: 6/15

Any Questions?