5. ggplot2 패키지를 알아보자

5. ggplot2 패키지를 알아보자

- Grammer of Graphics
- ggplot 이전과 이후로 나뉜다

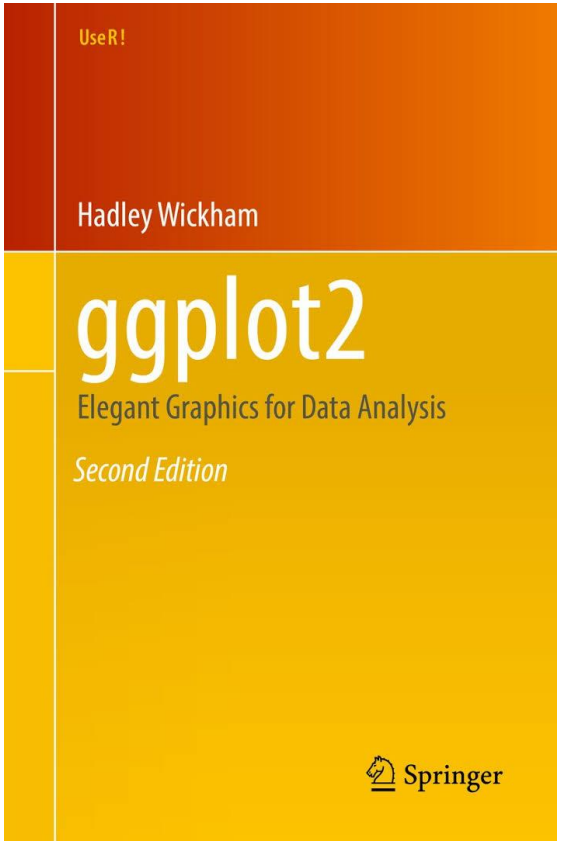
Source : https://blog.naver.com/definitice/221127682474

5. ggplot2 패키지를 알아보자

ggplot2 패키지

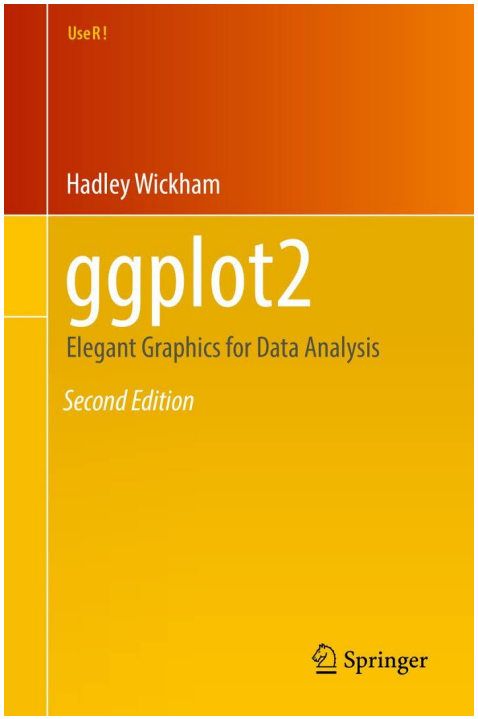
chief scientist at R Studio
University of Auckland,
Stanford University and
Rice University.

dplyr
reshape2
ggplot2
ggvis
rvest
.....

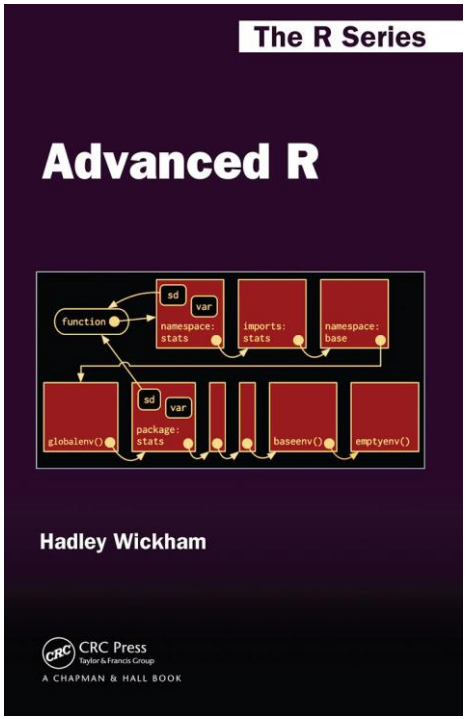


5. ggplot2 패키지를 알아보자

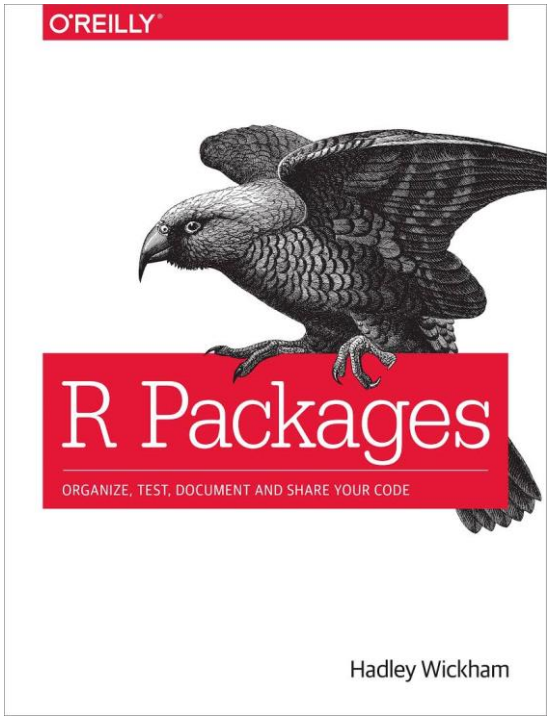
위컴의 책



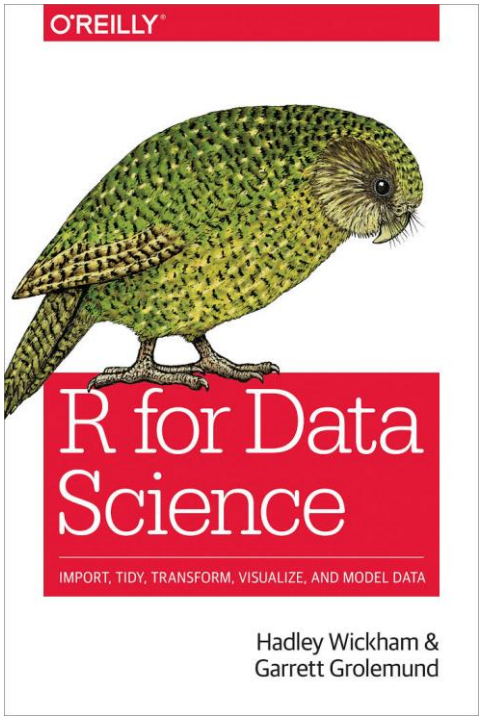
2009



2014



2015



2016

5. ggplot2 패키지를 알아보자

RStudio

readxl

dplyr

ggplot2

tibble

rmarkdown

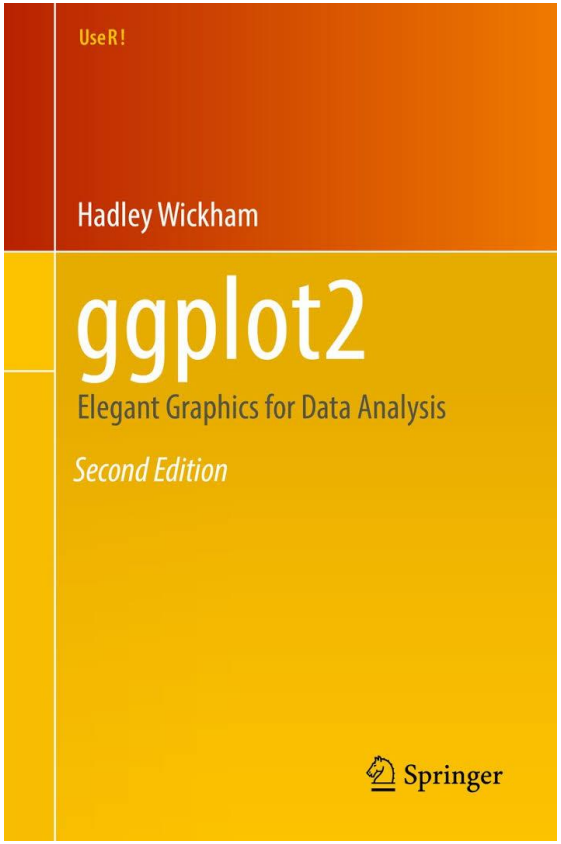
Shiny

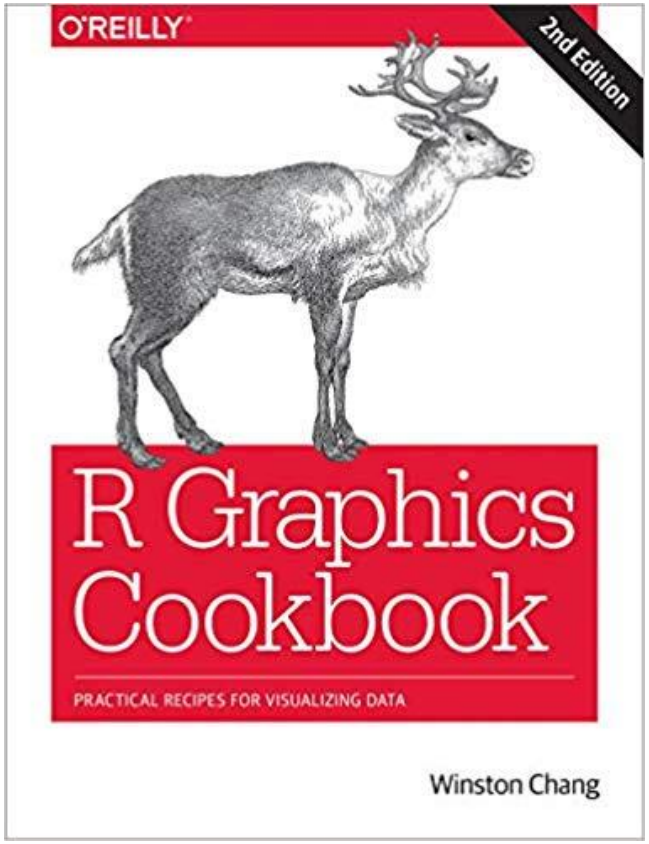
sparklyr

tidyr

5. ggplot2 패키지를 알아보자

ggplot2 유명한 책





5. ggplot2 패키지를 알아보자

cheatsheat (치트키?) 구글에서 ggplot2 cheatsheat를 검색해 보자

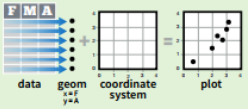
Data Visualization with ggplot2

Cheat Sheet

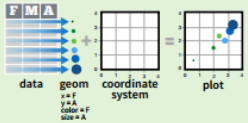
RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

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

Required

Not required, sensible defaults supplied

ggplot(data = mpg, **aes**(x = cty, y = hwy))

Geoms - Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

Graphical Primitives

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + **geom_blank()**
(Useful for expanding limits)

b + **geom_curve**(aes(yend = lat + 1, xend=long+1,curvature=z)) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + **geom_path**(lineend="butt", linejoin="round', linemitre=1)
x, y, alpha, color, group, linetype, size

a + **geom_polygon**(aes(group = group))
x, y, alpha, color, fill, group, linetype, size

b + **geom_rect**(aes(xmin = long, ymin=lat, xmax= long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + **geom_ribbon**(aes(ymin=unemploy - 900, ymax=unemploy + 900)) - x, ymax, ymin
alpha, color, fill, group, linetype, size

Line Segments

common aesthetics: x, y, alpha, color, linetype, size

b + **geom_abline**(aes(intercept=0, slope=1))
b + **geom_hline**(aes(yintercept = lat))
b + **geom_vline**(aes(xintercept = long))
b + **geom_segment**(aes(yend=lat+1, xend=long+1))
b + **geom_spoke**(aes(angle = 1:1155, radius = 1))

One Variable

Continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + **geom_area(stat = "bin")**
x, y, alpha, color, fill, linetype, size

c + **geom_density**(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + **geom_dotplot()**
x, y, alpha, color, fill

Two Variables

Continuous X, Continuous Y

e <- ggplot(mpg, aes(cty, hwy))

e + **geom_label**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + **geom_jitter**(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

e + **geom_point()**
x, y, alpha, color, fill, shape, size, stroke

e + **geom_quantile()**
x, y, alpha, color, group, linetype, size, weight

e + **geom_rug**(sides = "bl")
x, y, alpha, color, linetype, size

e + **geom_smooth**(method = lm)
x, y, alpha, color, fill, group, linetype, size, weight

e + **geom_text**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

f <- ggplot(mpg, aes(class, hwy))

f + **geom_col()**
x, y, alpha, color, fill, group, linetype, size

f + **geom_boxplot()**
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + **geom_dotplot**(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group

f + **geom_violin**(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight

Discrete X, Discrete Y

Continuous Bivariate Distribution

h <- ggplot(diamonds, aes(carat, price))

h + **geom_bin2d**(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + **geom_density2d()**
x, y, alpha, colour, group, linetype, size

h + **geom_hex()**
x, y, alpha, colour, fill, size

Continuous Function

i <- ggplot(economics, aes(date, unemploy))

i + **geom_area()**
x, y, alpha, color, fill, linetype, size

i + **geom_line()**
x, y, alpha, color, group, linetype, size

i + **geom_step**(direction = "hv")
x, y, alpha, color, group, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

j + **geom_crossbar**(fatten = 2)
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + **geom_errorbar()**
x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

j + **geom_linerange()**
x, ymin, ymax, alpha, color, group, linetype, size

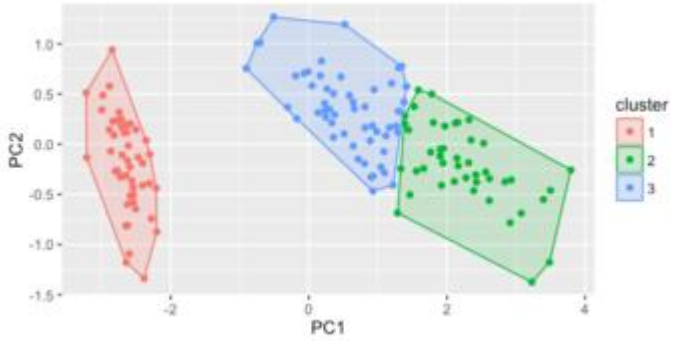
j + **geom_pointrange()**
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

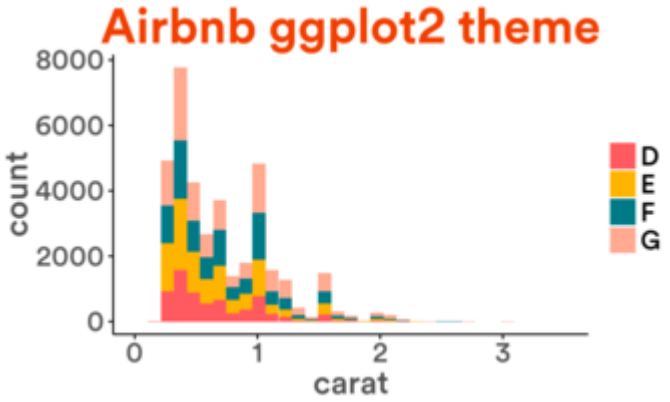
Maps

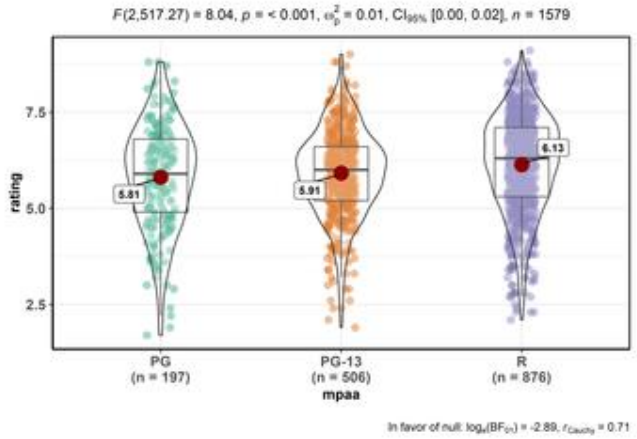
data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")

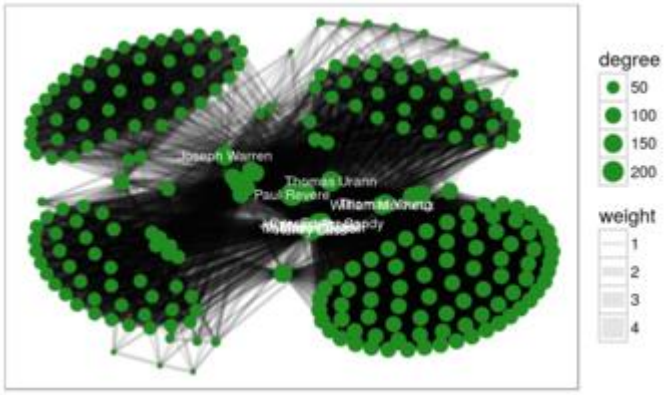
5. ggplot2 패키지를 알아보자

더 다양한 시각화 http://www.ggplot2-exts.org/gallery/









5. ggplot2 패키지를 알아보자

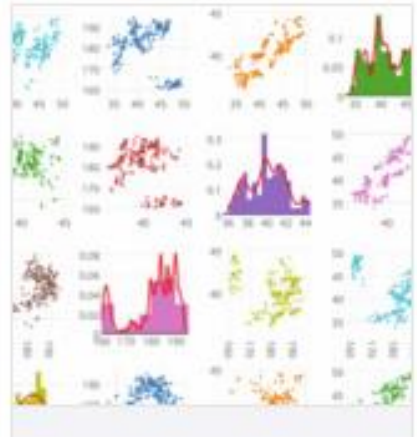
더 다양한 시각화 https://plot.ly/r/

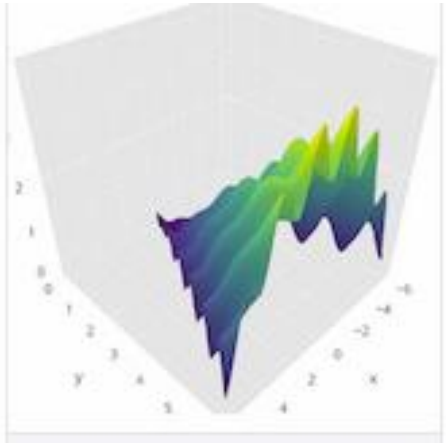
plotly는 Interactive 그래프를 그려주는 라이브러리입니다

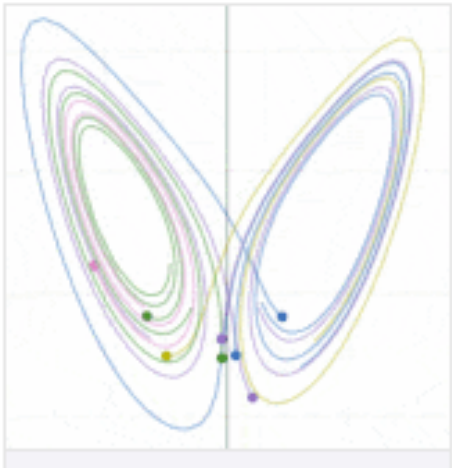
Scala, R, Python, Javascript, MATLAB 등에서 사용할 수 있습니다

시각화를 위해 D3.js를 사용하고 있습니다

사용해보면 사용이 쉽고, 세련된 느낌을 받습니다

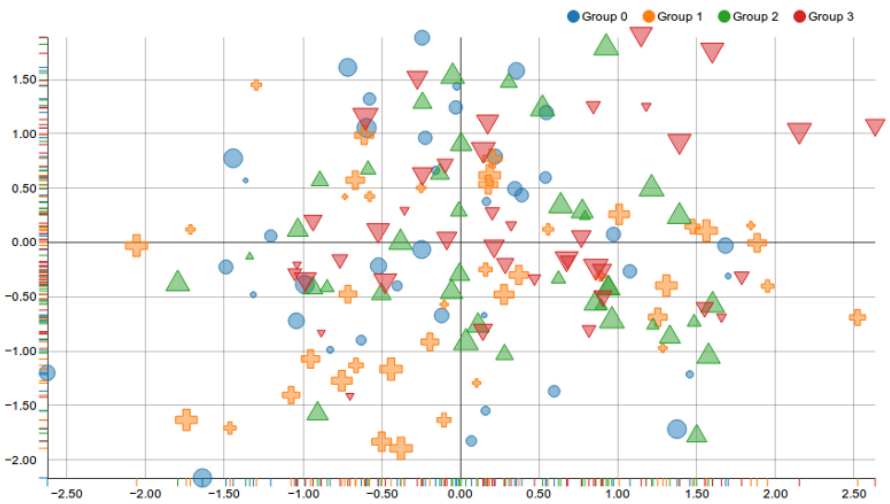


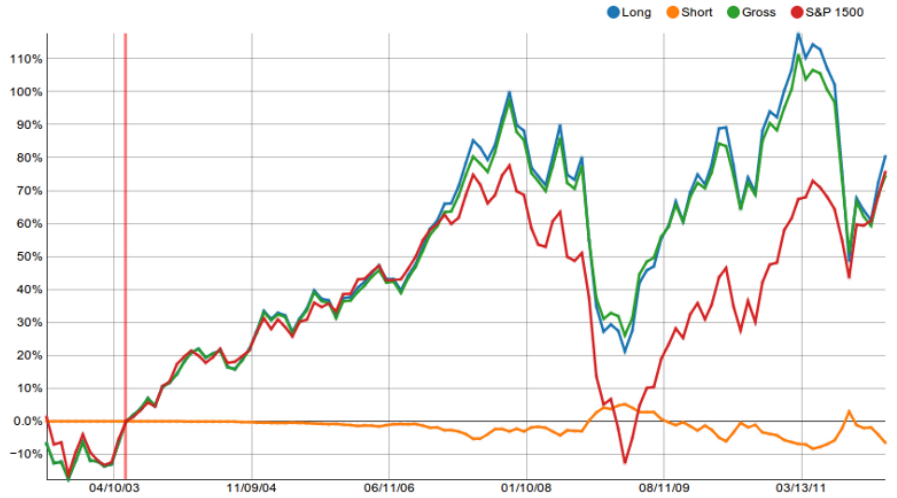




5. ggplot2 패키지를 알아보자

더 다양한 시각화 D3.js (Data-Driven Documents)





https://d3js.org/
http://nvd3.org/examples/index.html

5. ggplot2 패키지를 알아보자

기본 그래픽과 비교

```
mtcars
str(mtcars)

# cyl 가 num 으로 되어 있어서 factor
mtcars$cyl <- as.factor(mtcars$cyl)

# mpg는 miles per galloon, hp는 horse
plot(mpg ~ hp , data=mtcars,
     col=cyl, pch=c(4,6,8)[mtcars$cyl
legend("topright", legend=levels(mtca
       pch = c(4,6,8),
       col=levels(mtcars$cyl))
```

```
> str(mtcars)
'data.frame':   32 obs. of  11 vari
 $ mpg : num  21 21 22.8 21.4 18.7 1
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 .
 $ disp: num  160 160 108 258 360 .
 $ hp  : num  110 110 93 110 175 10
 $ drat: num  3.9 3.9 3.85 3.08 3.1
 $ wt  : num  2.62 2.88 2.32 3.21 3
 $ qsec: num  16.5 17 18.6 19.4 17
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 .
 $ am  : num  1 1 1 0 0 0 0 0 0 0 .
 $ gear: num  4 4 4 3 3 3 3 4 4 4 .
 $ carb: num  4 4 1 1 2 1 4 2 2 4 .
```

5. ggplot2 패키지를 알아보자

기본 그래픽과 비교

```
mtcars
str(mtcars)

# cyl 가 num 으로 되어 있어서 factor 로 바꾸어 준다
mtcars$cyl <- as.factor(mtcars$cyl)

# mpg는 miles per galloon, hp는 horse power
plot(mpg ~ hp , data=mtcars,
     col=cyl, pch=c(4,6,8)[mtcars$cyl], cex=1.2)
legend("topright", legend=levels(mtcars$cyl),
       pch = c(4,6,8),
       col=levels(mtcars$cyl))
```

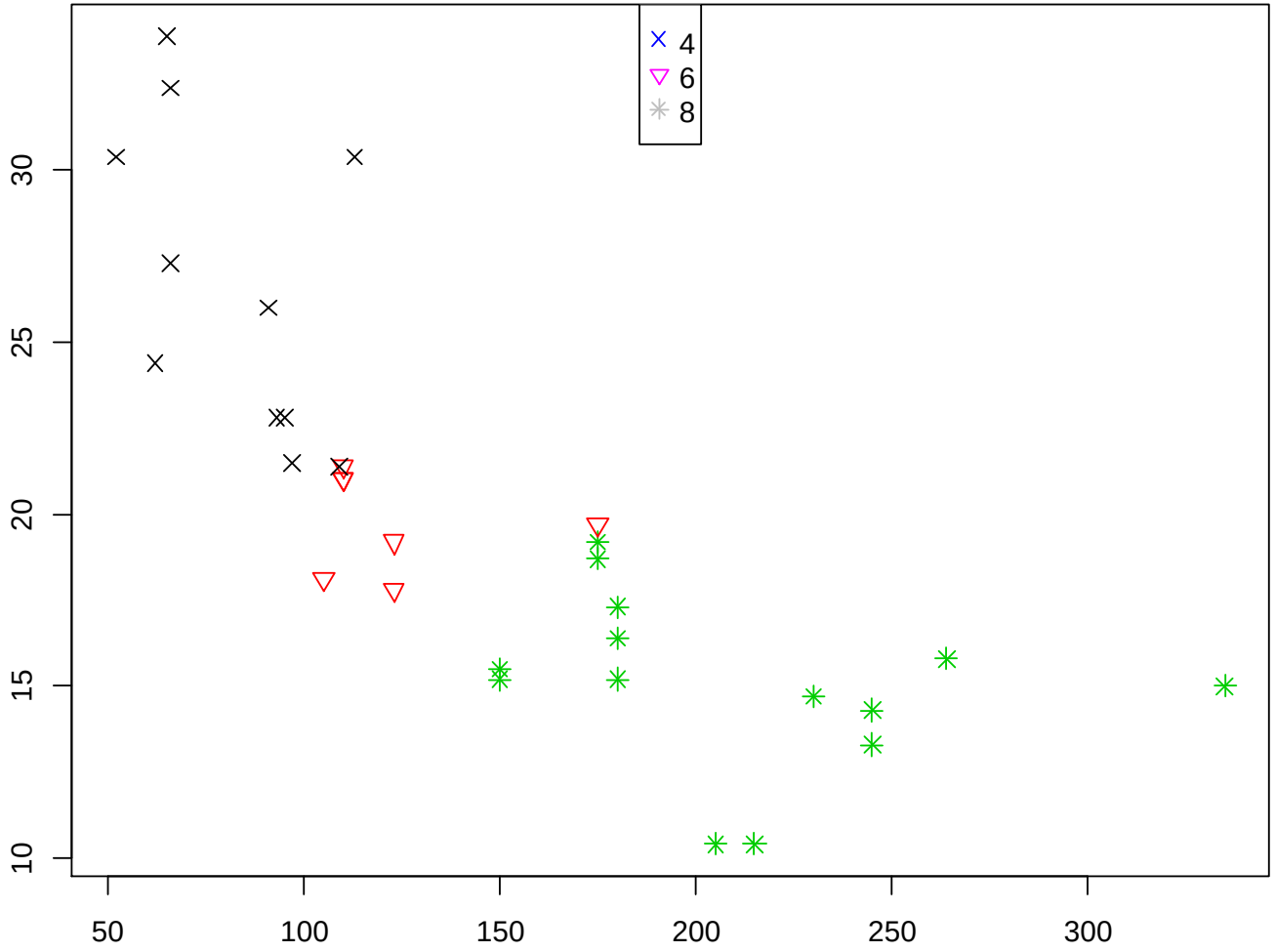
5. ggplot2 패키지를 알아보자

기본 그래픽과 비교

```
mtcars
str(mtcars)

# cyl 가 num 으로 되어 있어서 factor 로 바꾸어 준다
mtcars$cyl <- as.factor(mtcars$cyl)

# mpg는 miles per galloon, hp는 horse power
plot(mpg ~ hp , data=mtcars,
     col=cyl, pch=c(4,6,8)[mtcars$cyl], cex=1.2)
legend("topright", legend=levels(mtcars$cyl),
       pch = c(4,6,8),
       col=levels(mtcars$cyl))
```

5. ggplot2 패키지를 알아보자

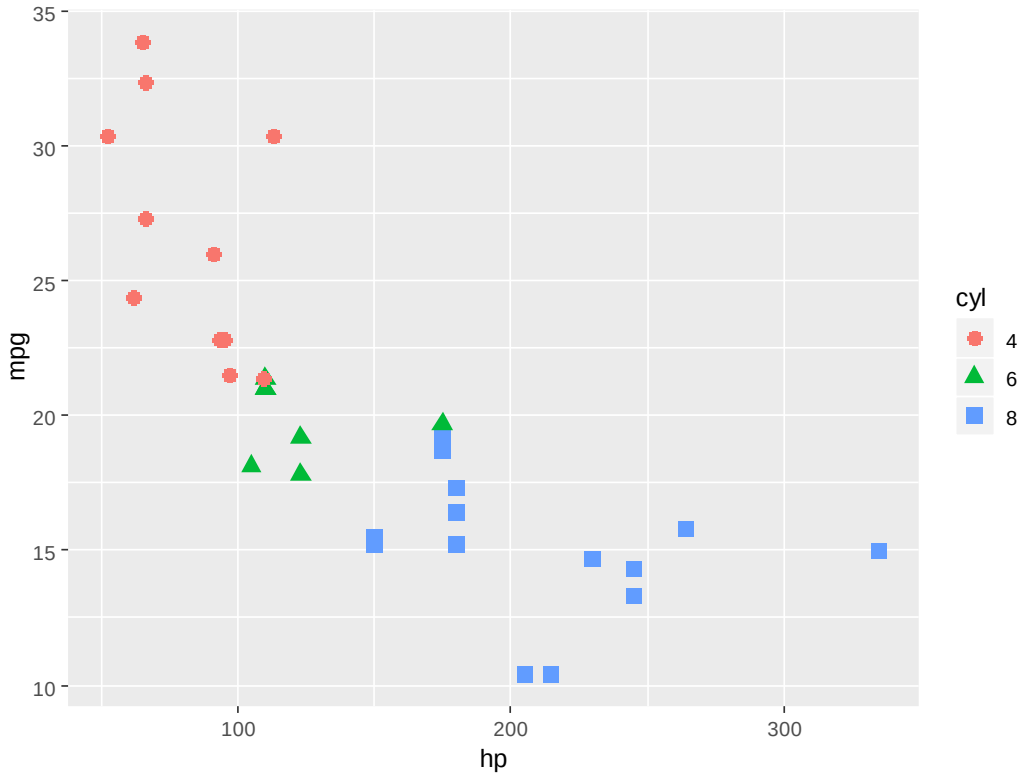
기본 그래픽



5. ggplot2 패키지를 알아보자

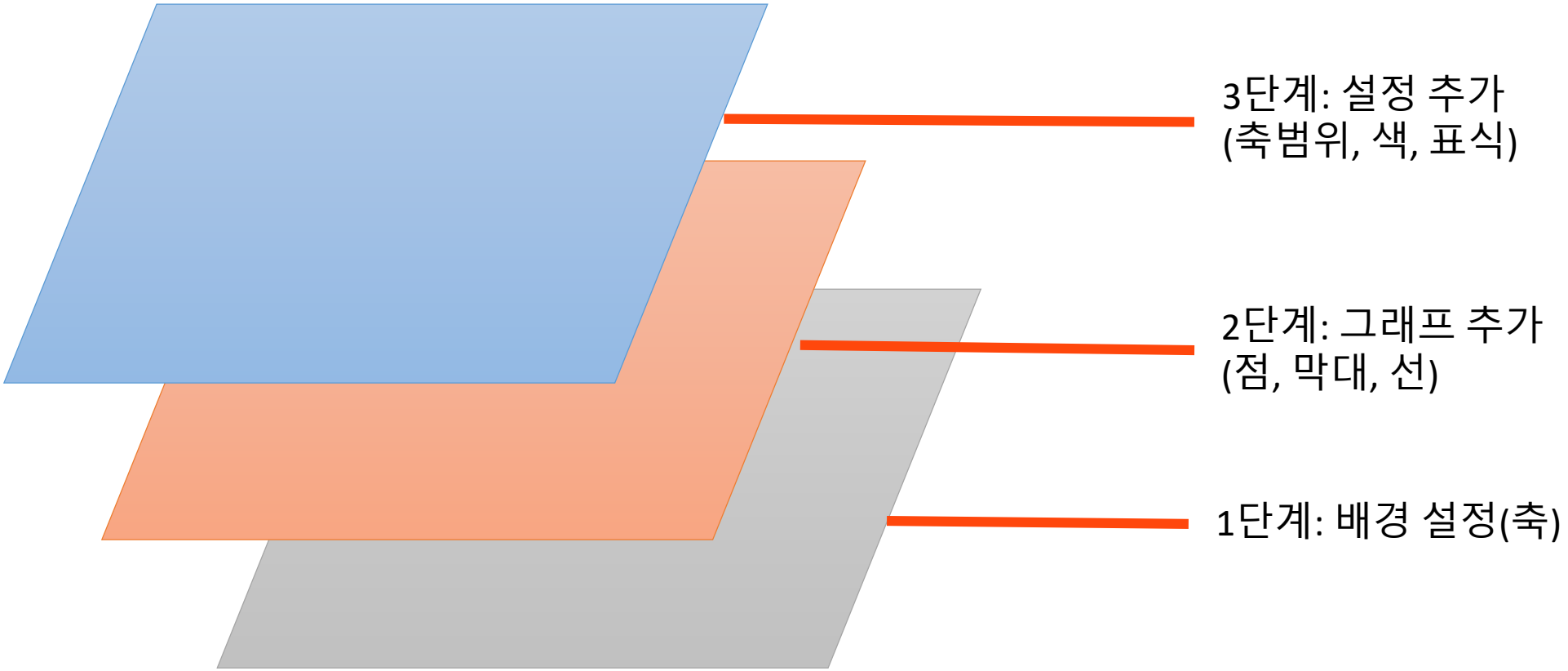
ggplot2

```
# ggplot 으로는 이렇게...  참 쉽죠!
library(ggplot2)
ggplot(mtcars, aes(x=hp, y=mpg, color=cyl, shape=cyl)) +
  geom_point(size=3)
```



5. ggplot2 패키지를 알아보자

ggplot2 레이어 구조 이해하기



ggplot을 그리는 2+3 단계

1. 평면 세팅	ggplot(diamonds, aes(x = , y =))
2. 도형선택	+ geom_point( )
3. 라벨	+ labs(title=" ", x=" ", y=" ")
4. 테마	+ theme( )
5. 패싯	+ facet_wrap(~ cut, ncol = 3)

Source : https://blog.naver.com/definitice/221127682474

5. ggplot2 패키지를 알아보자

1. 평면 세팅 ggplot(data= , aes(x = , y =))

주요 함수

ggplot(data = 데이터 셋명) : 데이터를 불러오는 역할

mapping = aes(x = , y =) : x축, y축의 꾸미기로 사용한다

geom_function() : 어떤 그래프를 그릴지 정하는 함수

mapping = aes(항목1=값1, 항목2=값2)

: geom_function() 의 옵션으로 꾸미기로 사용한다.

position(x, y), color(색상), fill(채우기), shape(모양), linetype(선 형태),

size(크기) 등

5. ggplot2 패키지를 알아보자

산점도 – 변수 간 관계 표현하기

```
# ggplot2 패키지 설치하기
install.packages("ggplot2")
library(ggplot2)
```

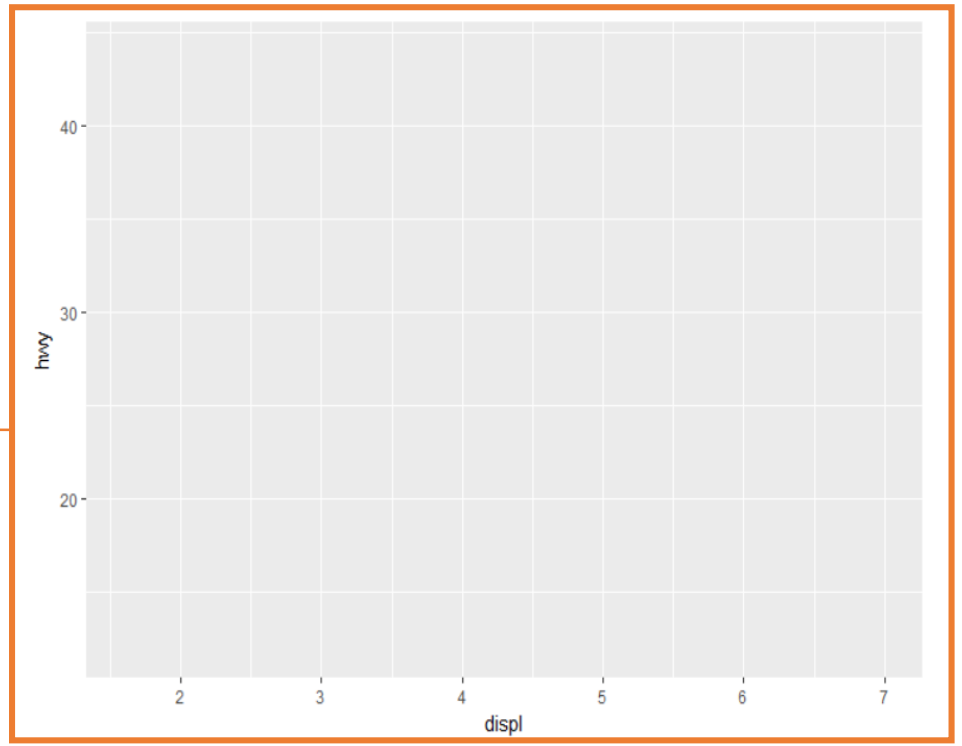
1. R 시각화 패키지인 ggplot2 패키지 설치
2. library(ggplot2)로 패키지 로드하기

```
# 1단계 배경설정(축)
ggplot(data=mpg, aes(x = displ, y = hwy))

# 배경에 산점도 추가
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point()

# x축 범위 3~6으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6)

# x축 범위 3~6, y축 범위 10~30으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6) + ylim(10,30)
```

1. 평면 세팅

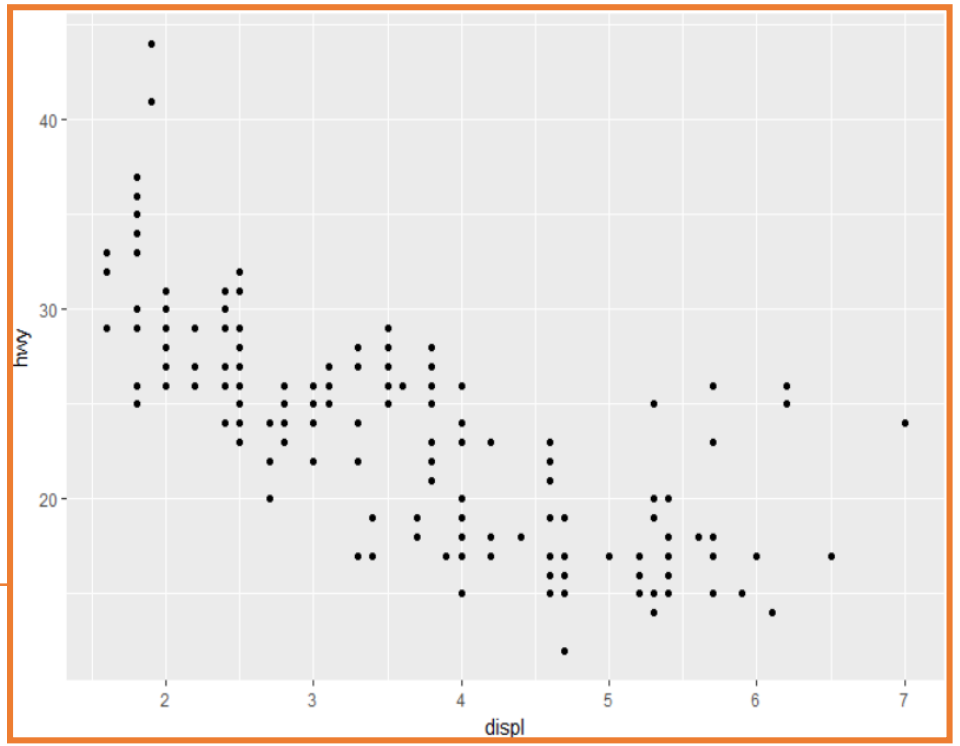
ggplot(diamonds, aes(x = , y =))

```r
# ggplot2 패키지 설치하기
install.packages("ggplot2")
library(ggplot2)

# 1단계 배경설정(축)
ggplot(data=mpg, aes(x = displ, y = hwy))

# 배경에 산점도 추가
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point()

# x축 범위 3~6으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6)

# x축 범위 3~6, y축 범위 10~30으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6) + ylim(10,30)
```



+ geom_point( )

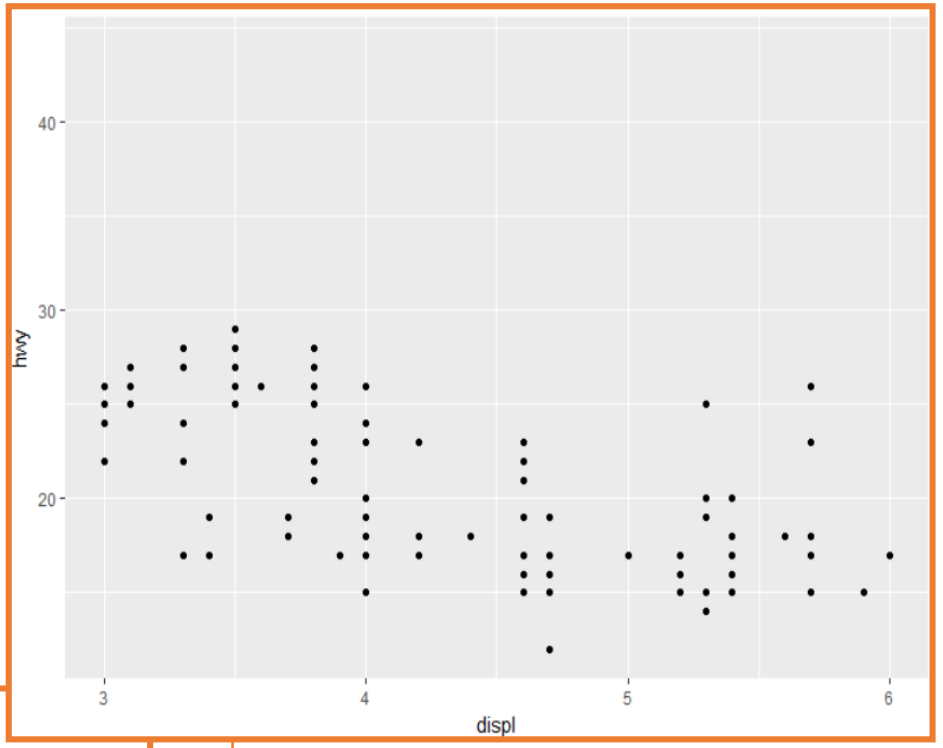
```
# ggplot2 패키지 설치하기
install.packages("ggplot2")
library(ggplot2)

# 1단계 배경설정(축)
ggplot(data=mpg, aes(x = displ, y = hwy))

# 배경에 산점도 추가
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point()

# x축 범위 3~6으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6)

# x축 범위 3~6, y축 범위 10~30으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6) + ylim(10,30)
```



5. ggplot2 패키지를 알아보자

산점도 – 변수 간 관계 표현하기

```
# ggplot2 패키지 설치하기
install.packages("ggplot2")
library(ggplot2)

# 1단계 배경설정(축)
ggplot(data=mpg, aes(x = displ, y = hwy))

# 배경에 산점도 추가
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point()

# x축 범위 3~6으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6)

# x축 범위 3~6, y축 범위 10~30으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6) + ylim(10,30)
```

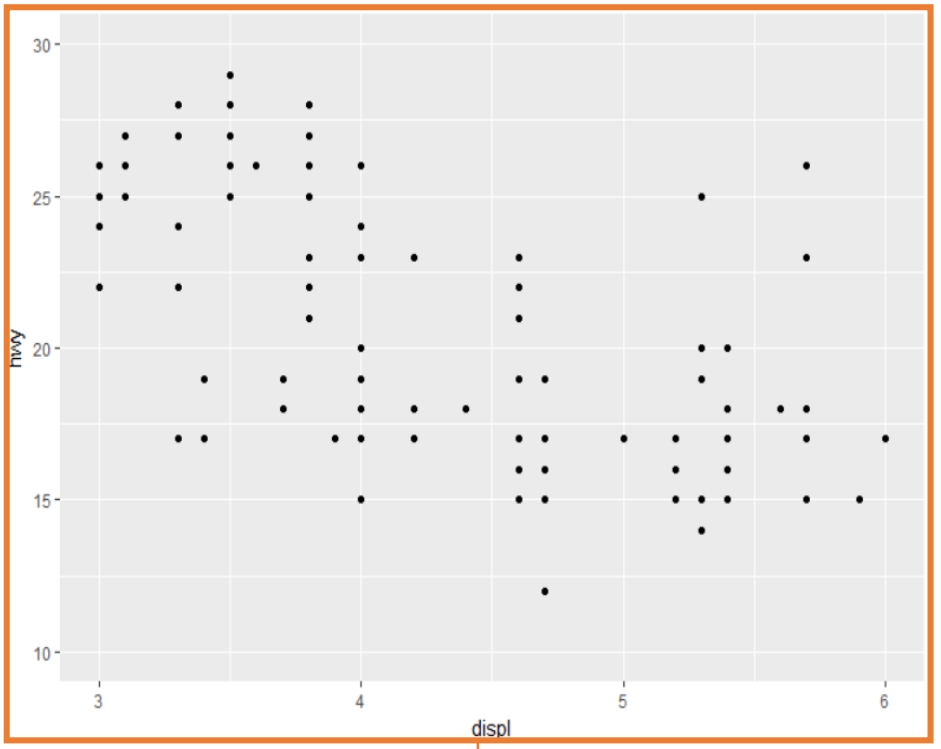



```
ggplot(data =   ,  aes( x =   , y = ) )  +  geom_points + xlim( , )  + ylim(  , )
```

5. ggplot2 패키지를 알아보자

산점도 – 변수 간 관계 표현하기

```
# ggplot2 패키지 설치하기
install.packages("ggplot2")
library(ggplot2)

# 1단계 배경설정(축)
ggplot(data=mpg, aes(x = displ, y = hwy))

# 배경에 산점도 추가
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point()

# x축 범위 3~6으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6)

# x축 범위 3~6, y축 범위 10~30으로 지정
ggplot(data=mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3,6) + ylim(10,30)
```



ggplot(mpg, aes(displ, hwy))

5. ggplot2 패키지를 알아보자

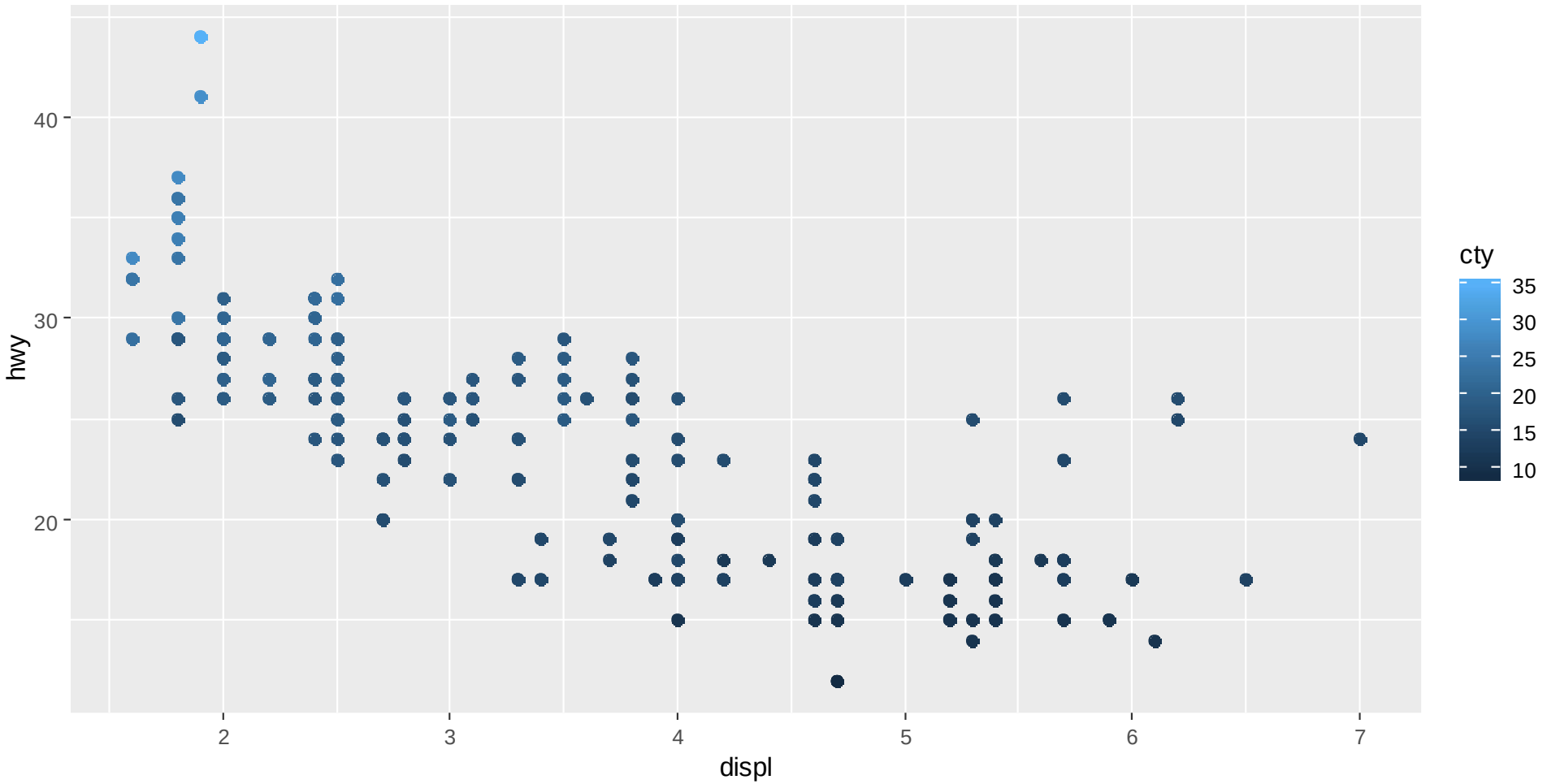
```
str(mpg)
dplyr::glimpse(mpg)
head(mpg)
# 연속형 변수일 때 컬러
ggplot(data = mpg, aes( x = displ, y = hwy,  color = cty )  ) +
  geom_point(size = 2)

# 범주형 변수일 때 컬러
ggplot(data = mpg, aes( x = displ, y = hwy,  color = drv )  ) +
  geom_point(size = 2)
```

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

5. ggplot2 패키지를 알아보자

```
str(mpg)
dplyr::glimpse(mpg)
head(mpg)
# 연속형 변수일 때 컬러
ggplot(data = mpg, aes( x = displ, y = hwy,  color = cty )  ) +
  geom_point(size = 2)

# 범주형 변수일 때 컬러
ggplot(data = mpg, aes( x = displ, y = hwy,  color = drv )  ) +
  geom_point(size = 2)
```

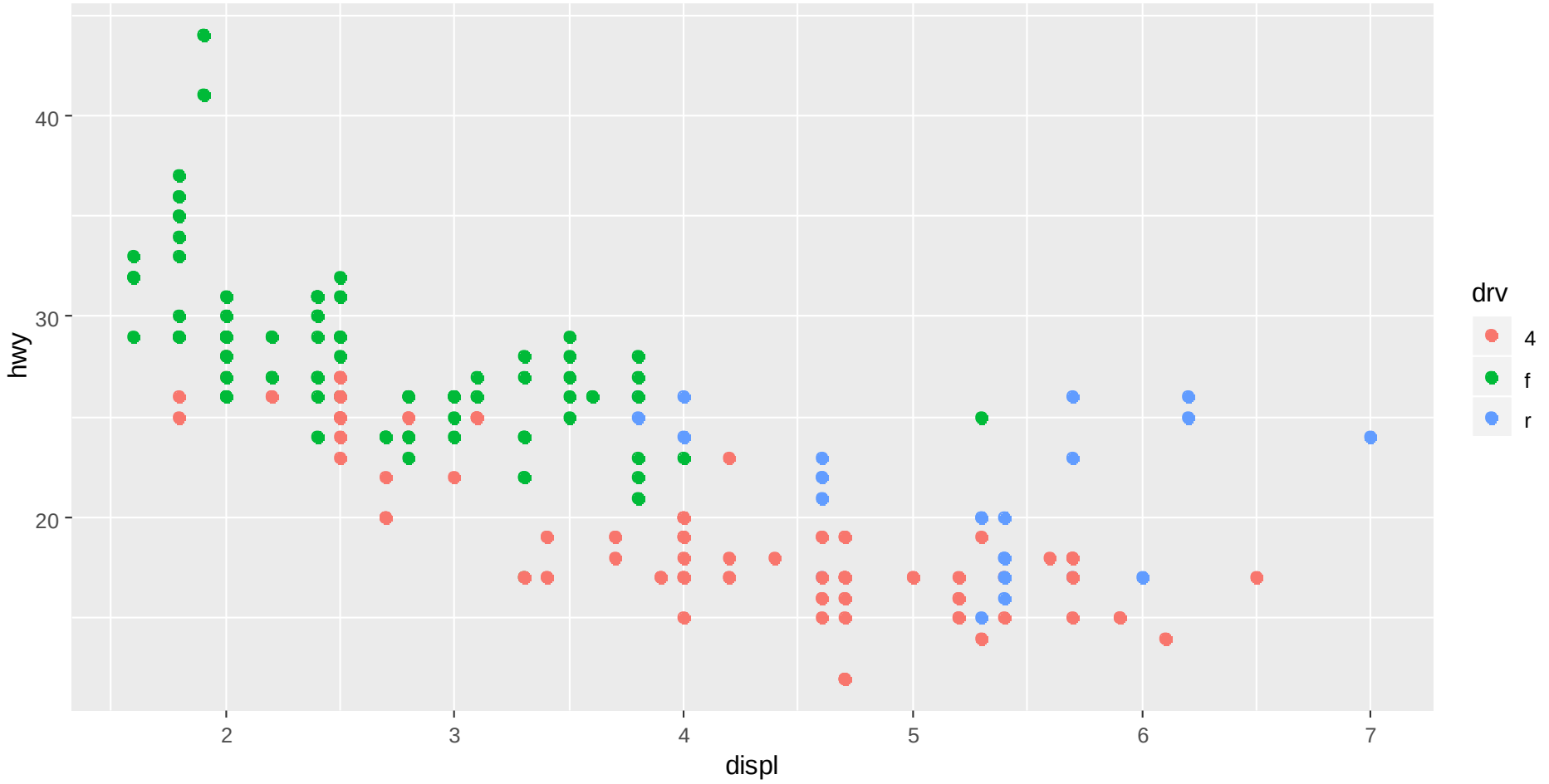
5. ggplot2 패키지를 알아보자



5. ggplot2 패키지를 알아보자

```
str(mpg)
dplyr::glimpse(mpg)
head(mpg)
# 연속형 변수일 때 컬러
ggplot(data = mpg, aes( x = displ, y = hwy,  color = cty )  ) +
  geom_point(size = 2)

# 범주형 변수일 때 컬러
ggplot(data = mpg, aes( x = displ, y = hwy,  color = drv )  ) +
  geom_point(size = 2)
```

5. ggplot2 패키지를 알아보자

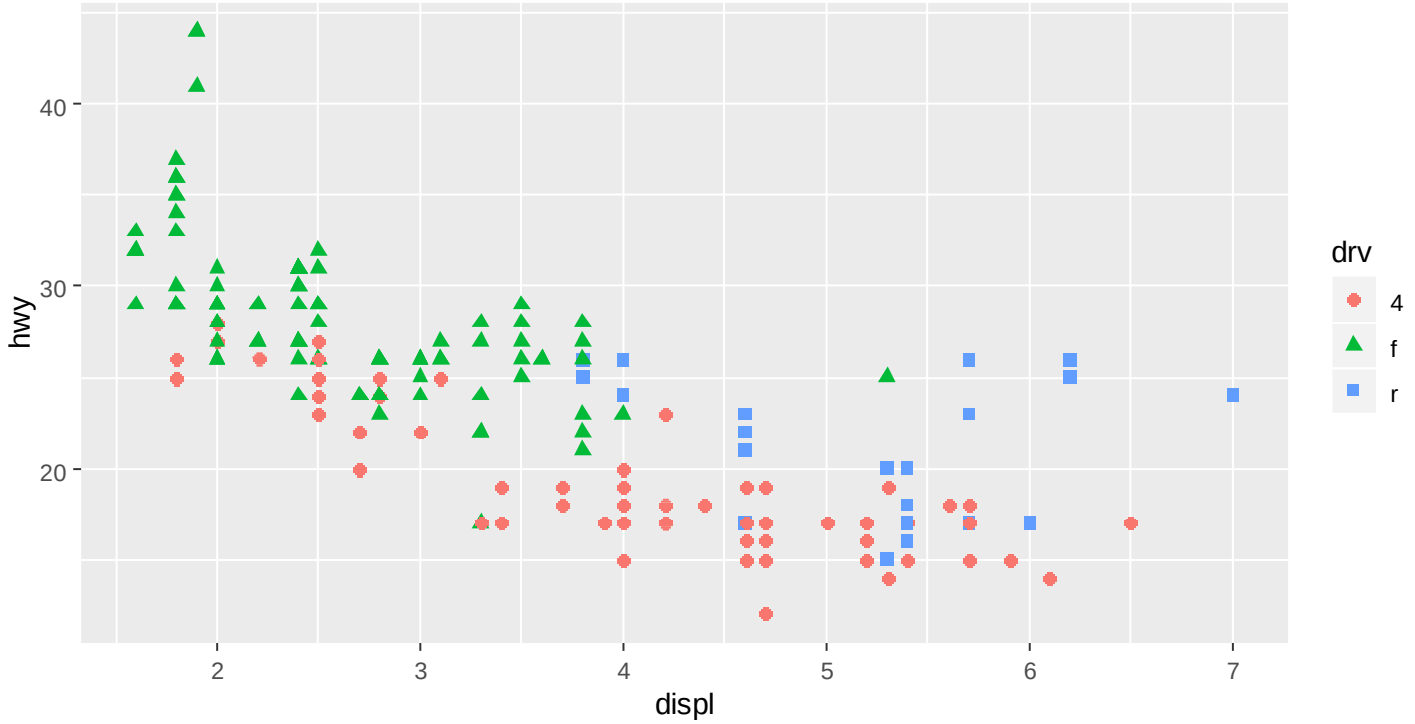


Geometry

name	description
geom_point	Scatterplot
geom_bar	Bar plot
geom_histogram	Histogram
geom_density	Prabablity distribution plot
geom_boxplot	Box and whiskers plot
geom_text	Textual annotations in a plot
geom_errorbar	Error bars

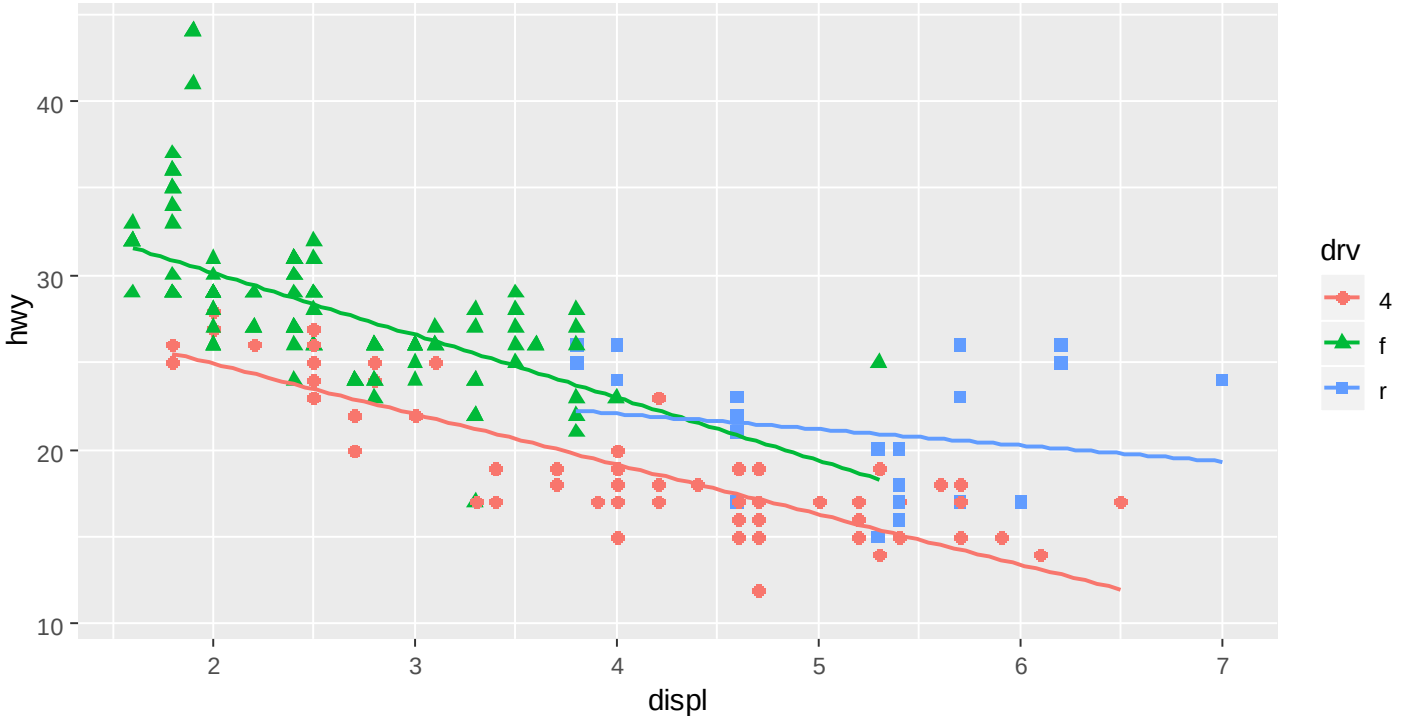
5. ggplot2 패키지를 알아보자

```r
# shape
ggplot(data = mpg,
       aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)


# geom_smooth() to fit linear regressions for each level
ggplot(data = mpg, aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)  +
  geom_smooth(method="lm")
```



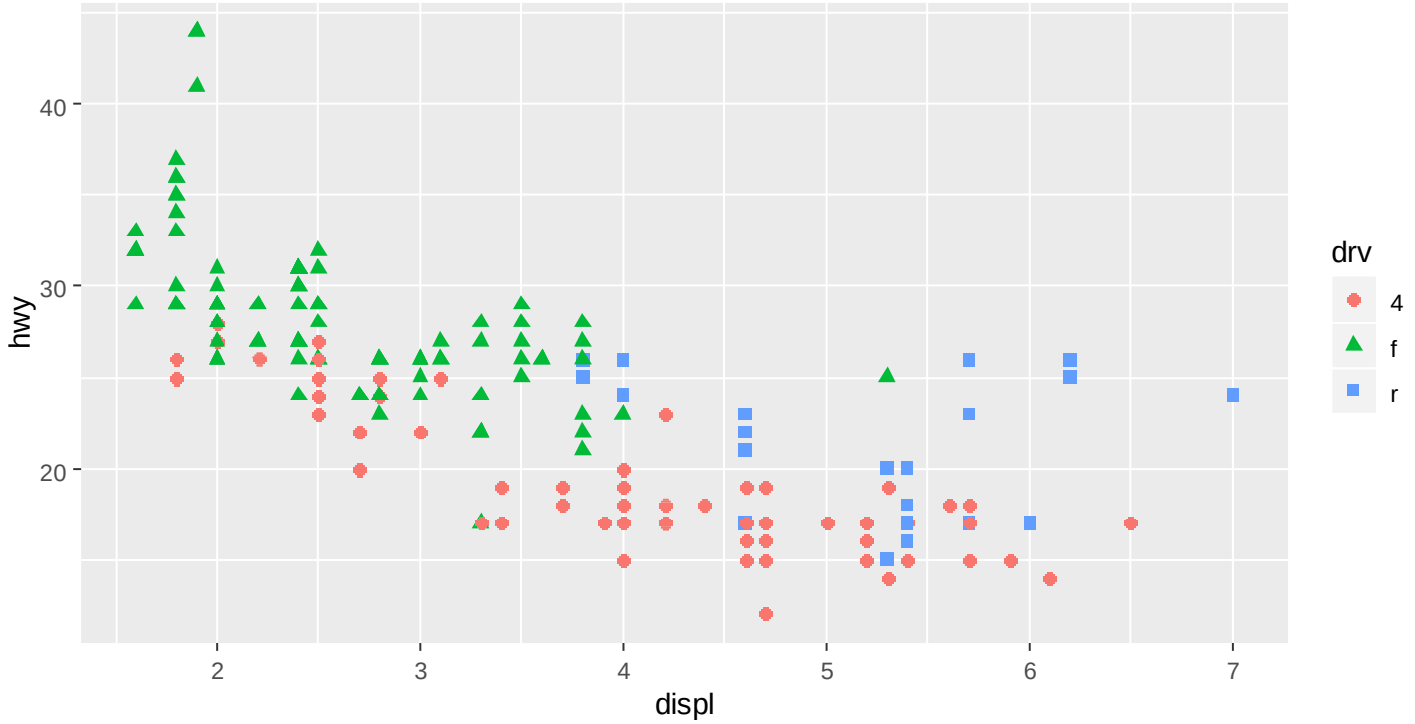
5. ggplot2 패키지를 알아보자

```r
# shape
ggplot(data = mpg,
       aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)


# geom_smooth() to fit linear regressions for each level
ggplot(data = mpg, aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)  +
  geom_smooth(method="lm")
```



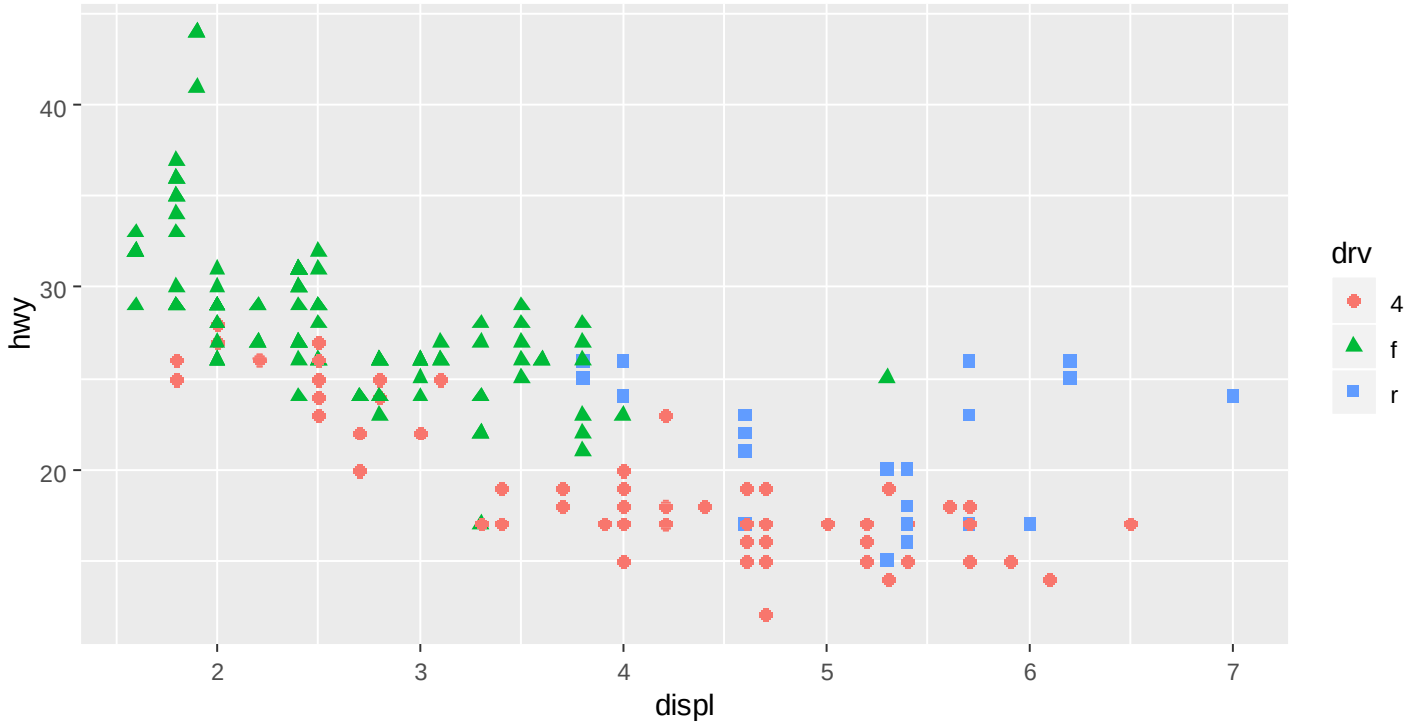
5. ggplot2 패키지를 알아보자

```r
# shape
ggplot(data = mpg,
       aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)


# geom_smooth() to fit linear regressions for each level
ggplot(data = mpg, aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)  +
  geom_smooth(method="lm")
```



5. ggplot2 패키지를 알아보자

```
p2 <- ggplot(data = mpg,
       aes(x = displ, y = hwy,  color = drv, shape = drv )) +
    geom_point(size = 2)
p2

p2 + geom_smooth(method="lm")

p2 + geom_smooth(method="lm") +
  theme_dark()
```

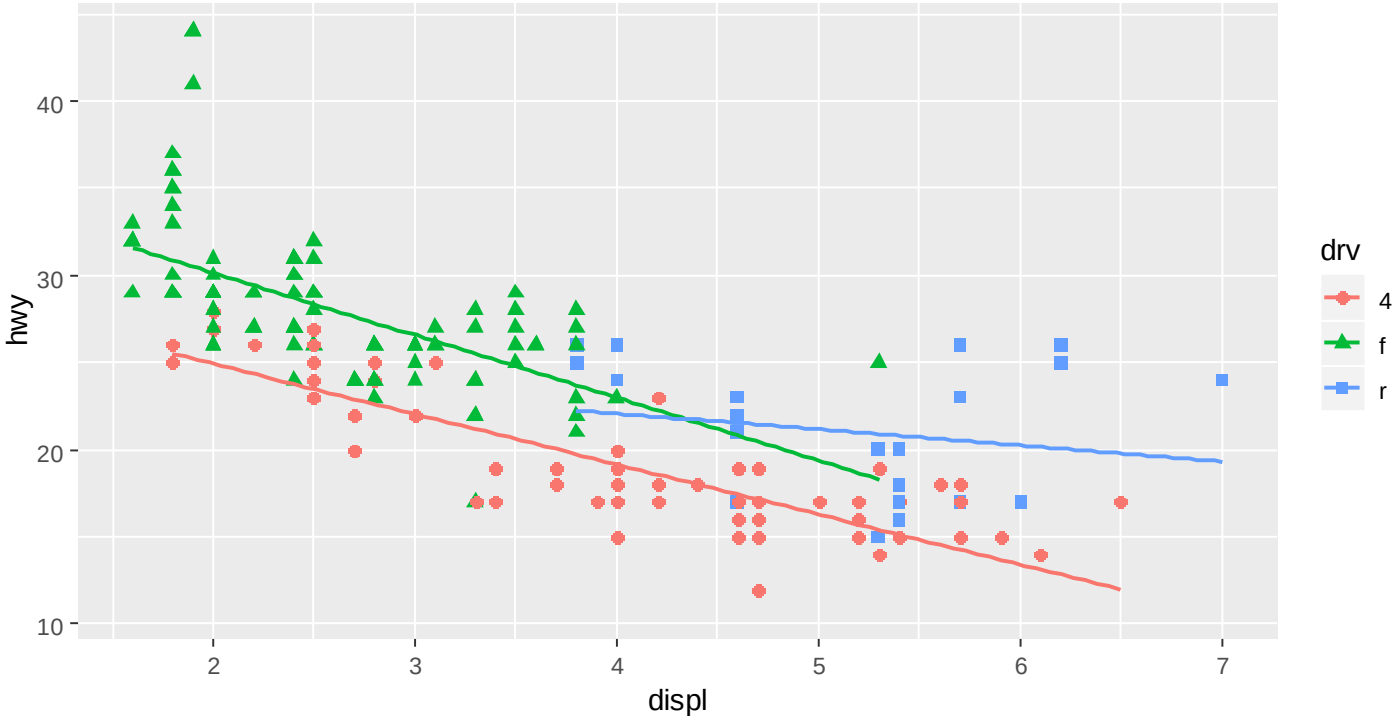


5. ggplot2 패키지를 알아보자

```
p2 <- ggplot(data = mpg,
       aes(x = displ, y = hwy,  color = drv, shape = drv )) +
    geom_point(size = 2)
p2
```

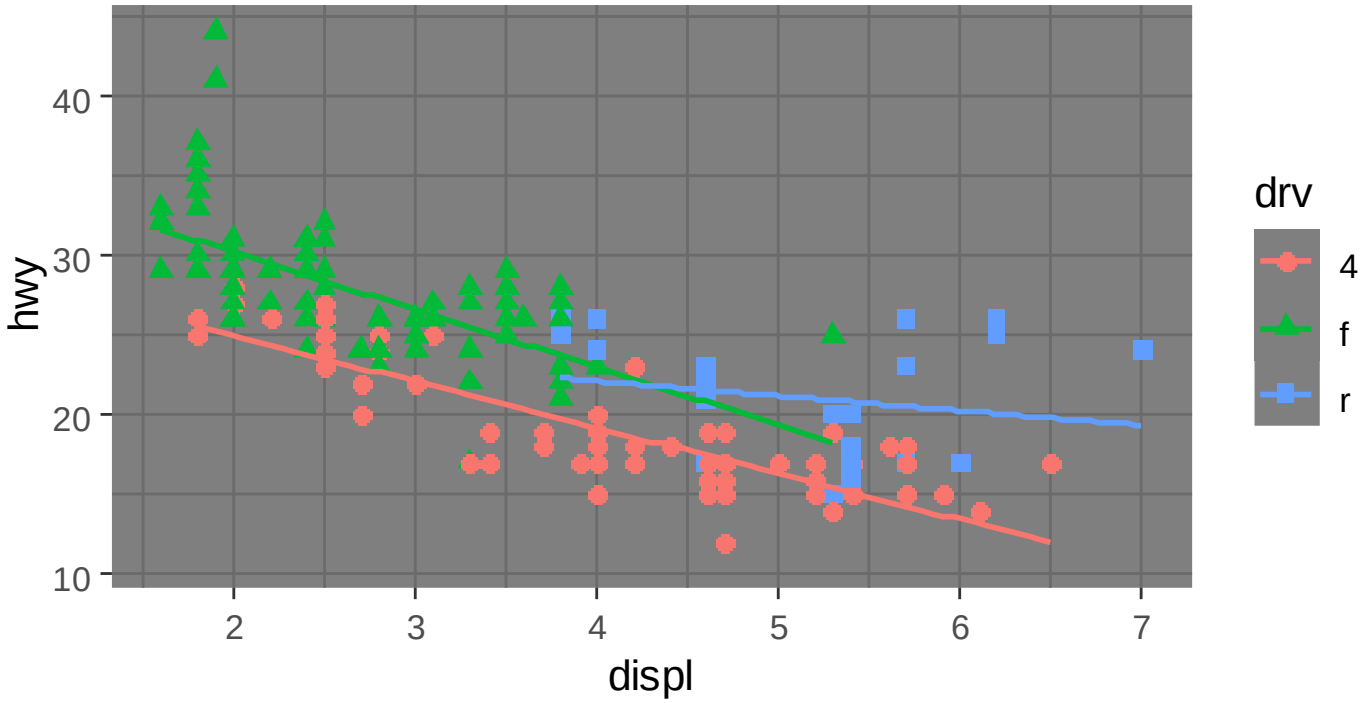
```
p2 + geom_smooth(method="lm")
```

```
p2 + geom_smooth(method="lm") +
  theme_dark()
```



5. ggplot2 패키지를 알아보자

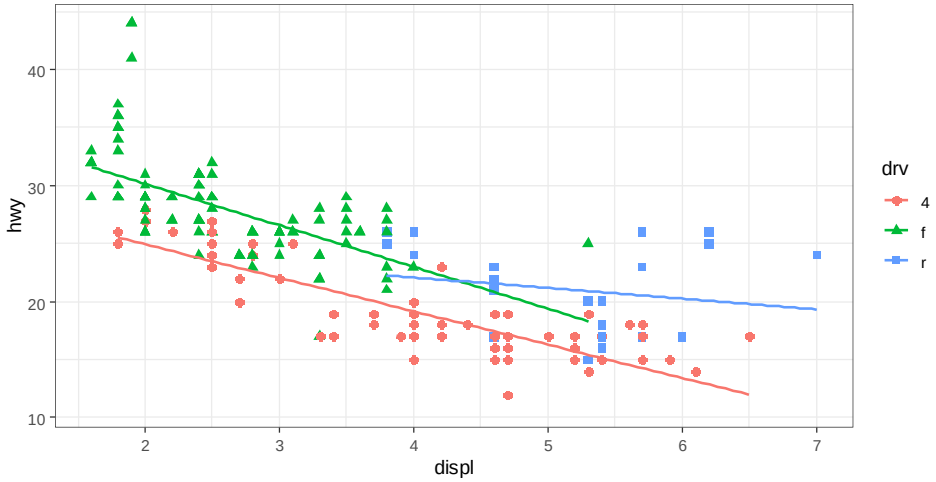
```
p2 <- ggplot(data = mpg,
        aes(x = displ, y = hwy,  color = drv, shape = drv )) +
    geom_point(size = 2)
p2

p2 + geom_smooth(method="lm")

p2 + geom_smooth(method="lm") +
  theme_dark()
```

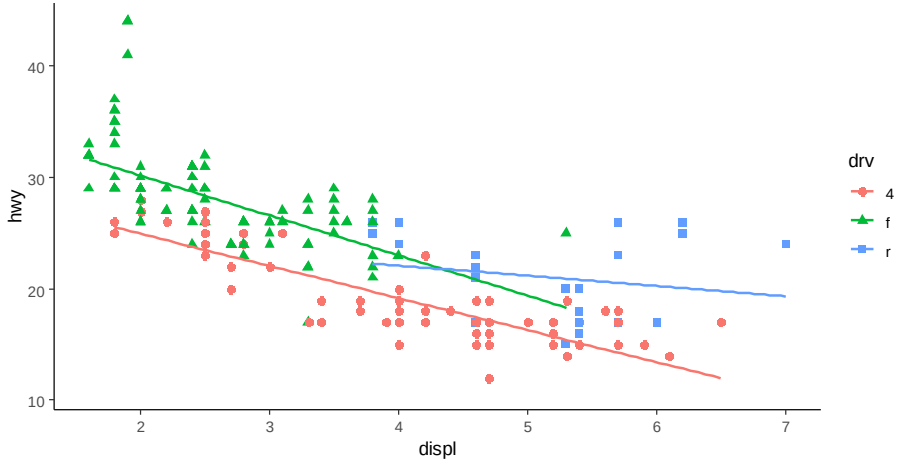


5. ggplot2 패키지를 알아보자

```
p3 <-
  ggplot(data = mpg,
         aes(x = displ, y = hwy,  color = drv, shape = drv )) +
  geom_point(size = 2)  +
  geom_smooth(method="lm")
```

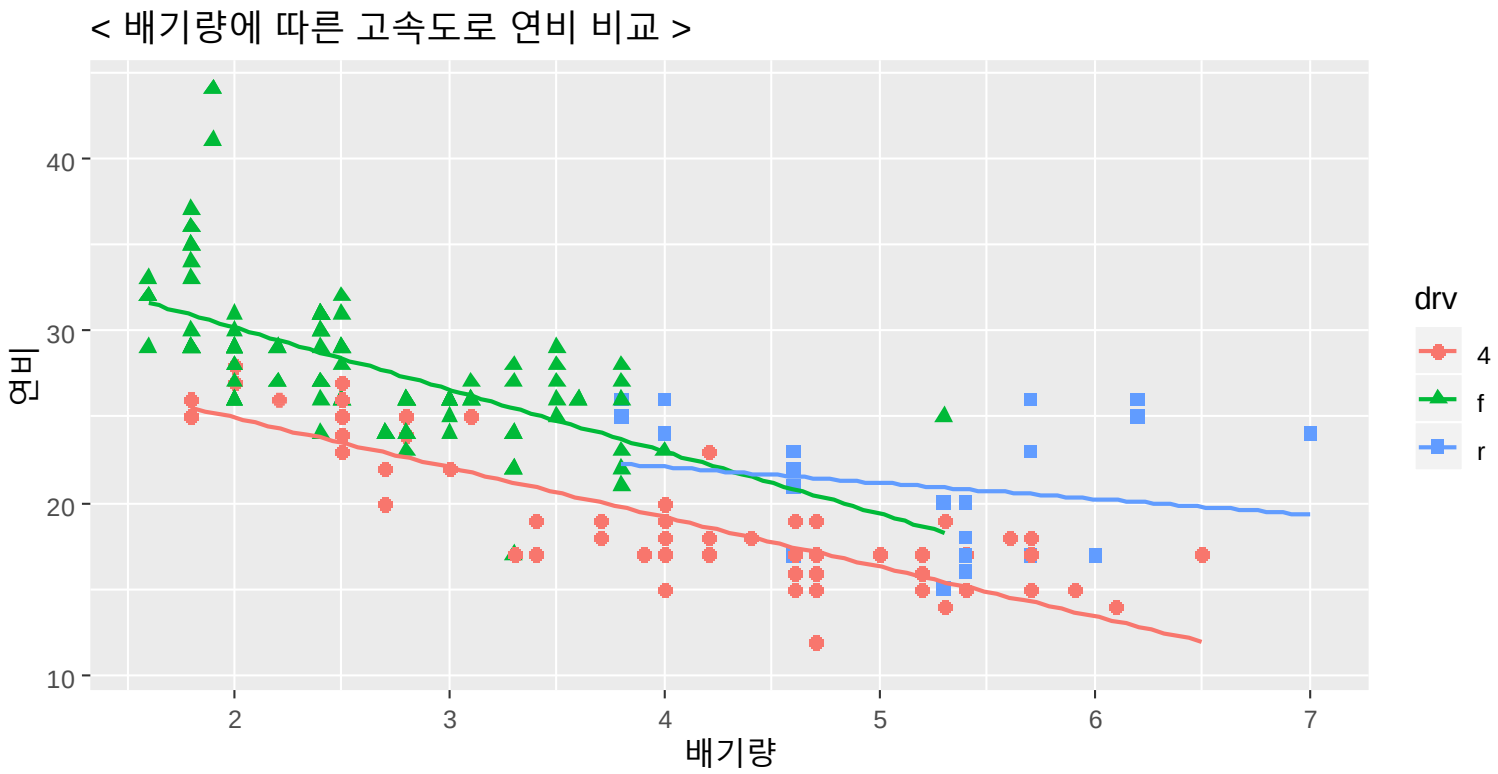
```
p3 + theme_dark()
p3 + theme_bw()
p3 + theme_classic()
```





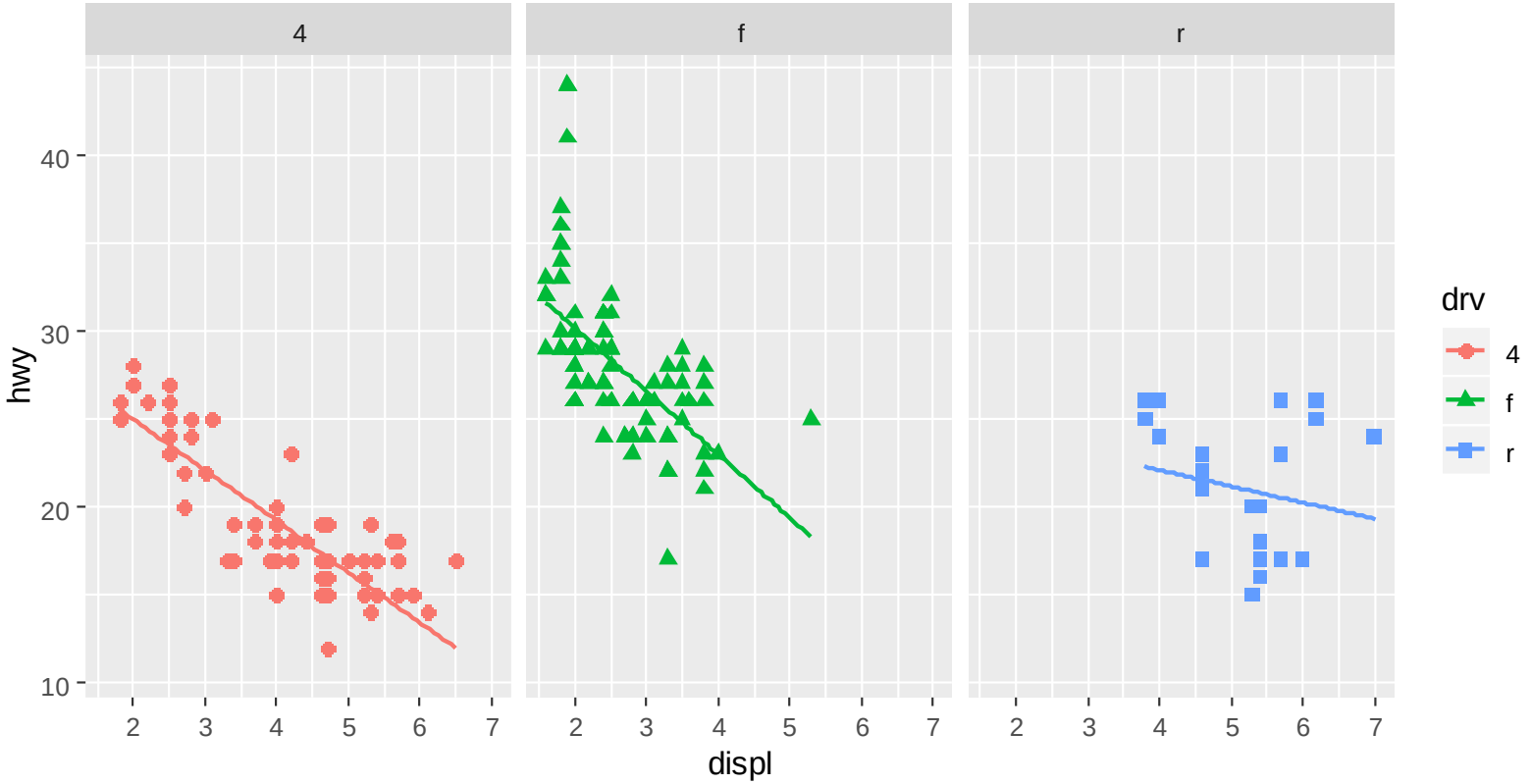
5. ggplot2 패키지를 알아보자

```
p3 + labs(title="< 배기량에 따른 고속도로 연비 비교 >", x = "배기량", y = "연비")

p3 + facet_wrap( ~ drv)
p2 + facet_wrap( ~ class)
```



5. ggplot2 패키지를 알아보자

```
p3 + labs(title="< 배기량에 따른 고속도로 연비 비교 >", x = "배기량", y = "연비")

p3 + facet_wrap( ~ drv)
p2 + facet_wrap( ~ class)
```



5. ggplot2 패키지를 알아보자

```
p3 + labs(title="< 배기량에 따른 고속도로 연비 비교 >", x = "배기량", y = "연비")

p3 + facet_wrap( ~ drv)
p2 + facet_wrap( ~ class)
```