

## Chapter.03

## 차트 종류를 결정하는 geoms

# | geom\_\*( ) 함수로 차트를 바꿔 가며 그려보자

M T W T F S S

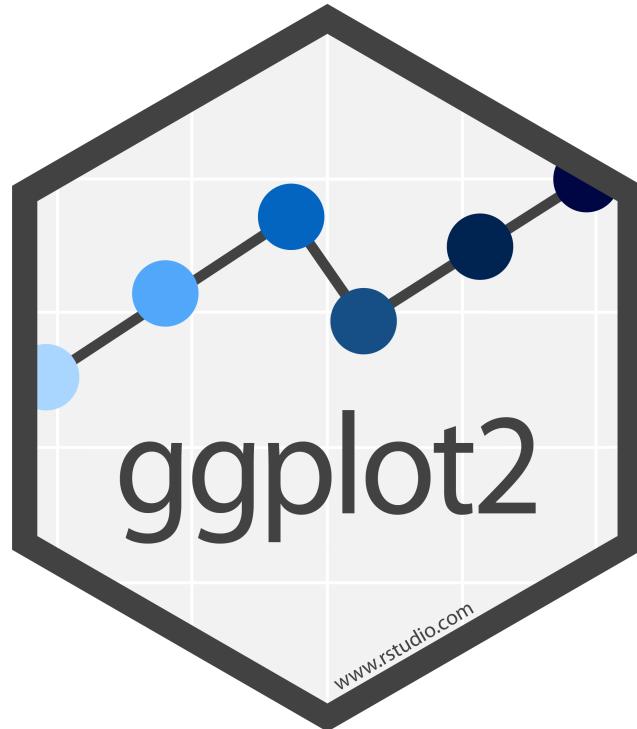
FASTCAMPUS  
ONLINE

금융공학/퀀트 |

강사. 박찬업

\* 표시는 다른 여러 글자들을 대표한다는 뜻입니다.

# 그림을 그리는 ggplot2



# I 차트 그리기 복습

## 템플릿

```
ggplot(data = { DATA }) +  
  { GEOM_FUNCTION }(mapping = aes({ MAPPINGS }))
```

## I 연결하기(mapping)와 설정하기(setting)

데이터 요소를 그림 요소에 연결(mapping)하거나 값을 직접 설정(setting)

```
ggplot(data = { DATA }) +  
  { GEOM_FUNCTION }(mapping = aes({ MAPPINGS }), { SETTING })
```

# I 연결한 데이터를 그리는 차트 종류 정하기

## 살펴볼 내용

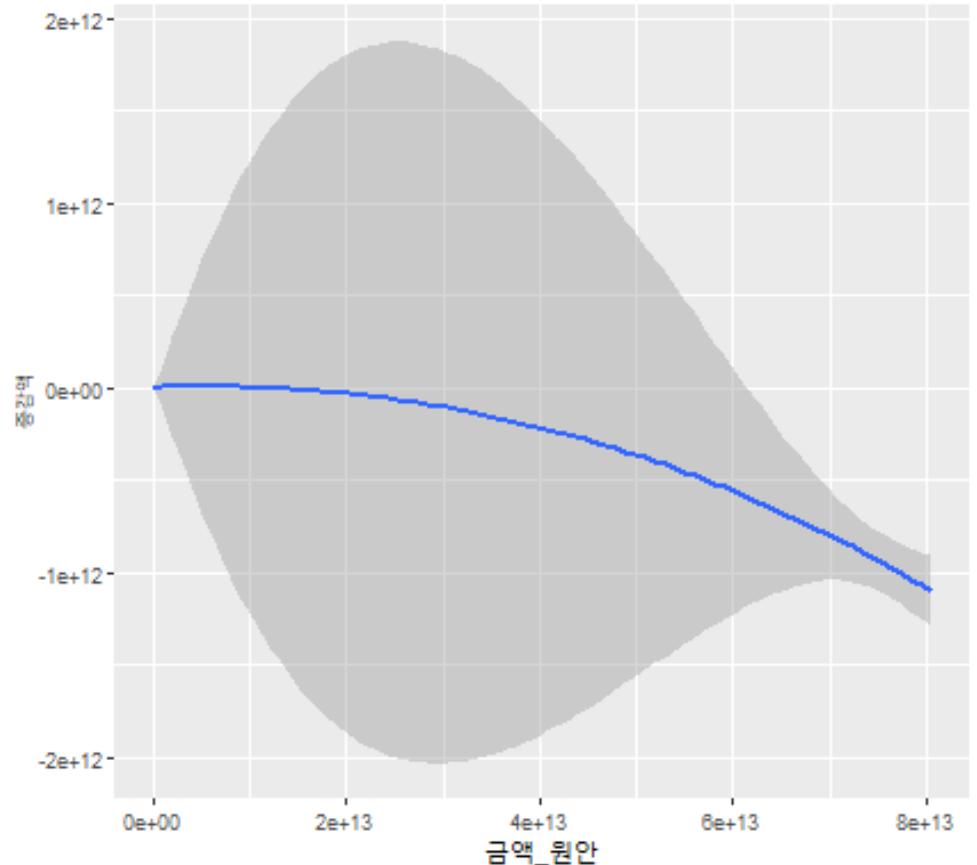
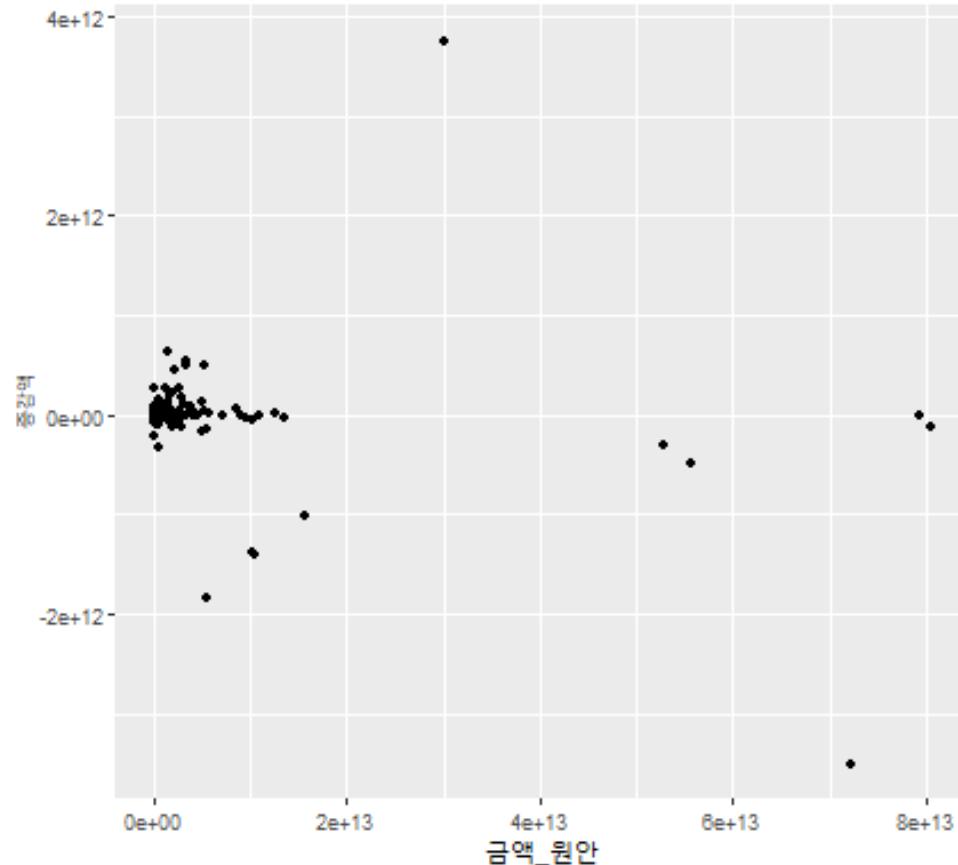
```
ggplot(data = { DATA }) +  
  { GEOM_FUNCTION }(mapping = aes({ MAPPINGS }))
```

Chapter.03

# 차트 모양을 결정하는 geom\_\*( ) 함수

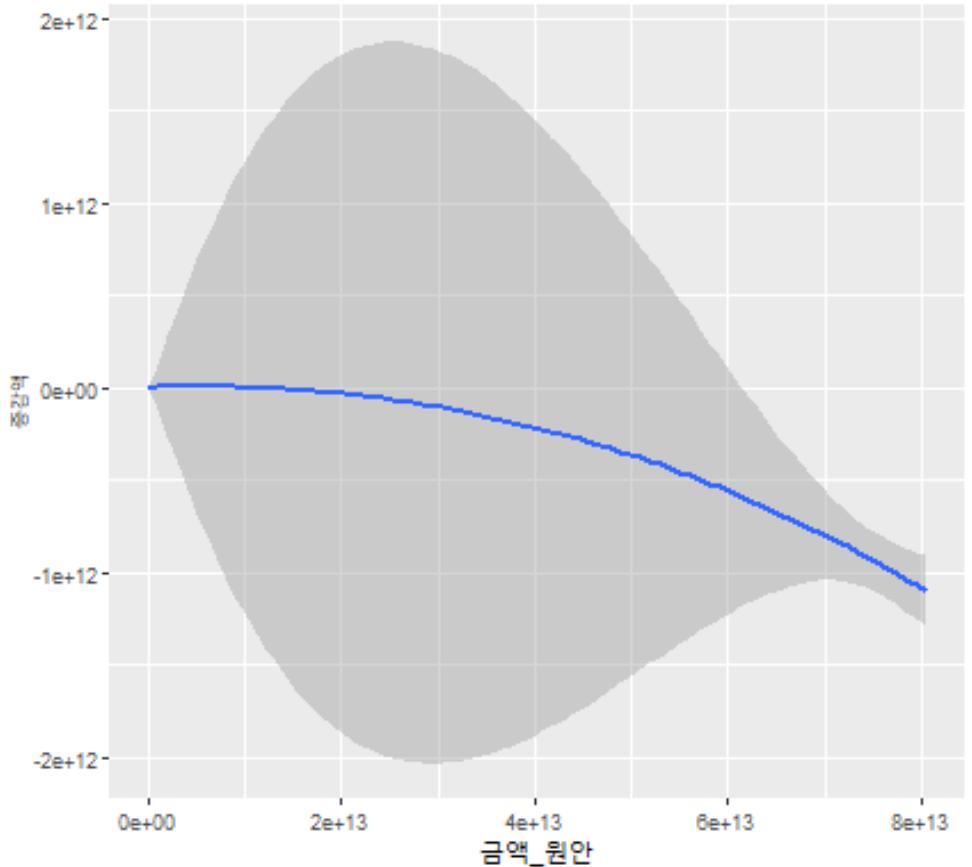
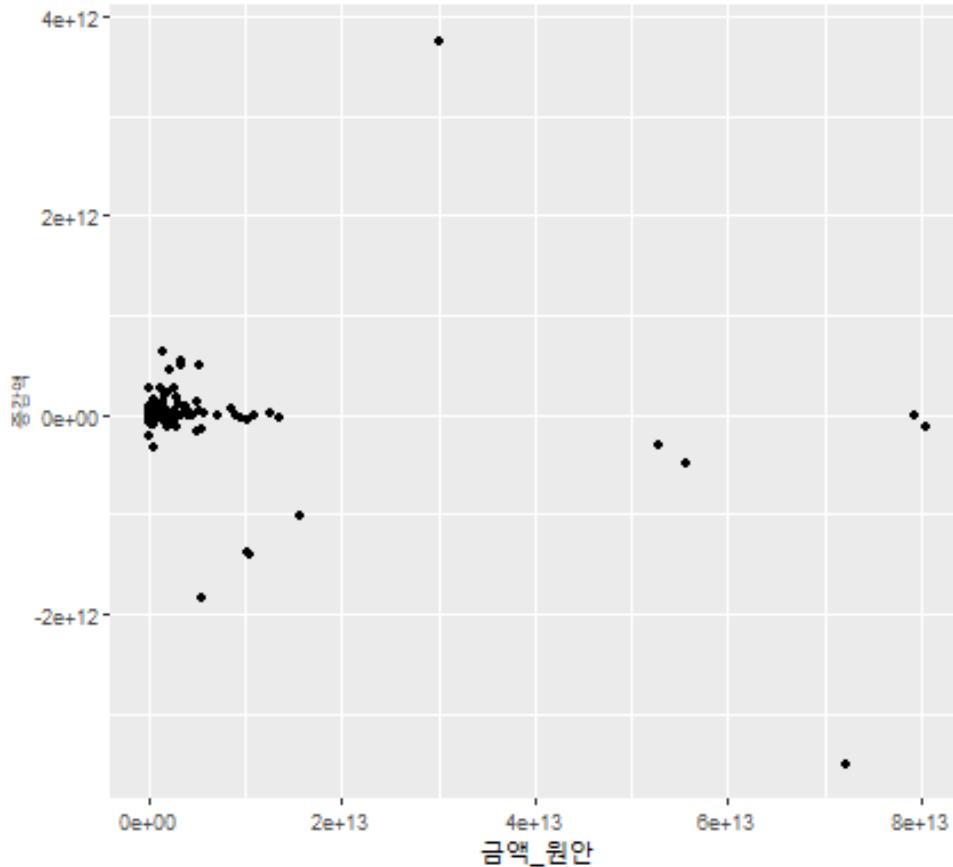
# | 모양을 정하는 geom\_\*(())

## 데이터, x, y가 같은 차트



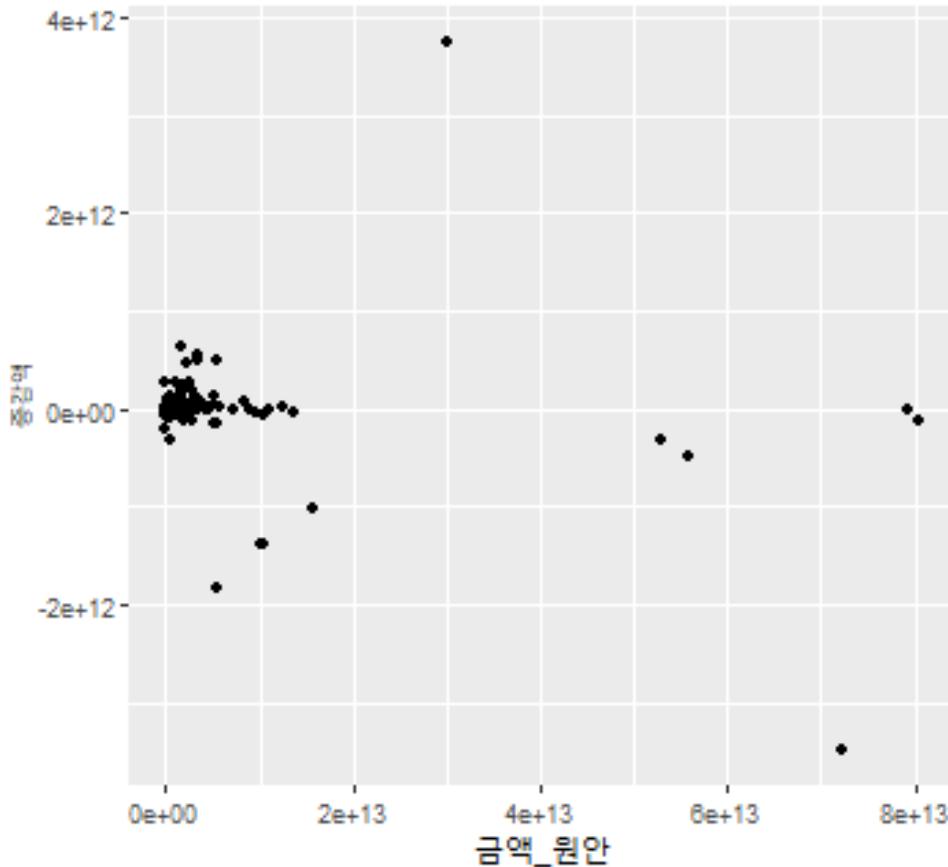
# | 모양을 정하는 geom\_\*(())

## geom\_\*(()) 함수를 다른 것을 사용함

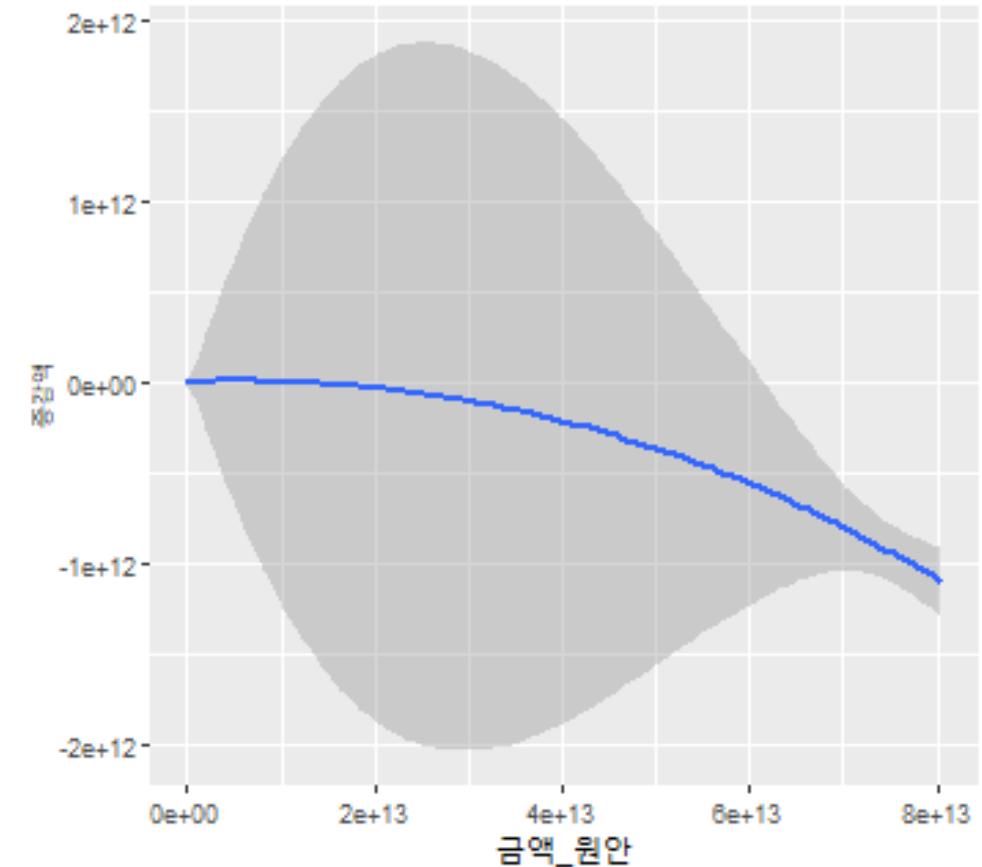


# I 모양을 정하는 geom\_\*(())

```
ggplot(krbg) +
  geom_point(aes(x = `금액_원안`, y = `증감액`))
```

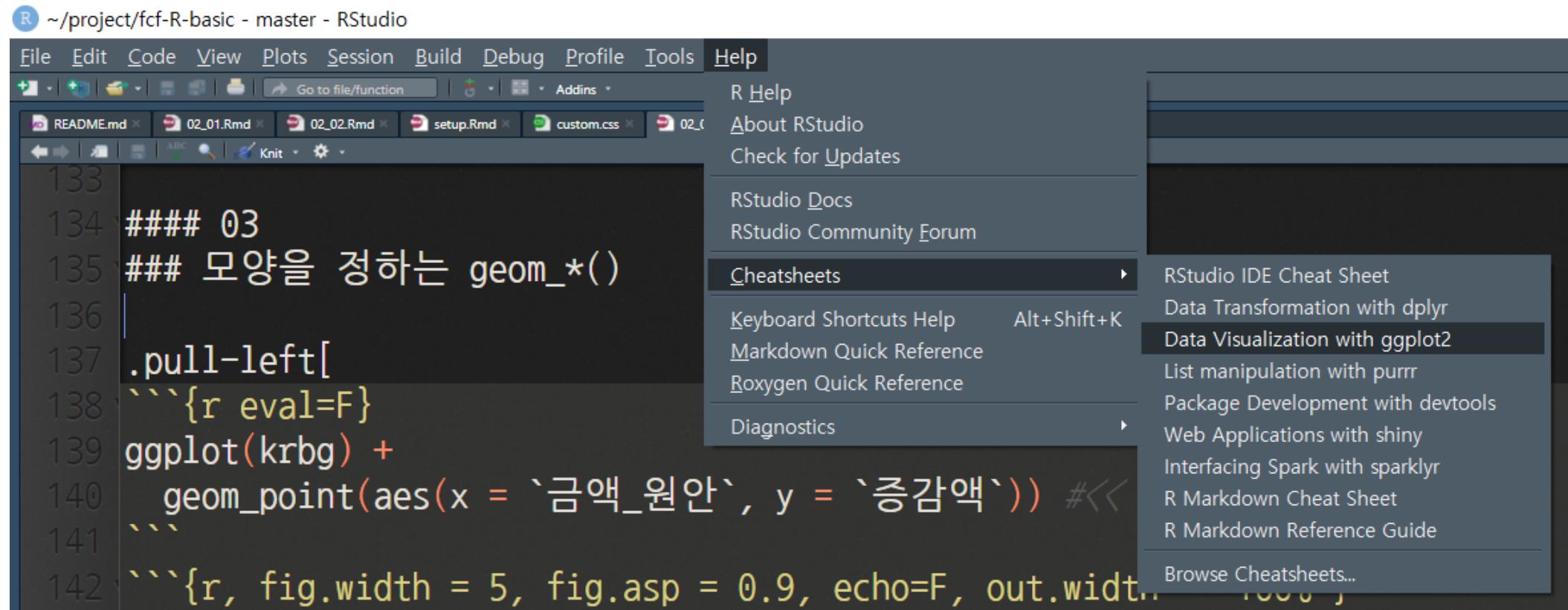


```
ggplot(krbg) +
  geom_smooth(aes(x = `금액_원안`, y = `증감액`))
```



# I 모양을 정하는 geom\_\*( )

## ggplot2의 설명서 보기



# I 모양을 정하는 geom\_\*(())

## ggplot2 설명서(cheat sheet)

### Data Visualization with ggplot2 :: CHEAT SHEET

#### 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>
  <FACTOR_FUNCTION>
  <SCALE_FUNCTION>
  <THEME_FUNCTION>
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) begins a plot that you finish by adding layers to. Add one geom function per layer.

geom<type> + <key> y = <key> data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last\_plot() Returns the last plot.

ggplot("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory.

Matches file type to file extension.

#### 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))  
c <- ggplot(diamonds, aes(carat, price))

+ a + b + c

continuous x , continuous y

e <- geom\_label(aes(label = cyl), nudge\_x = 1, nudge\_y = -1, check\_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

b + geom\_curve(aes(yend = lat + 1, send = long + 1, curvature = 0.5), x, yend, y, yend, alpha, angle, color, curvature, linetype, size, stroke)

a + geom\_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1), x, ymin, ymax, ymin, alpha, color, fill, linetype, size, stroke)

a + geom\_ribbon(aes(unemploy = 900, ymix = unemploy + 900), ymix, alpha, color, fill, group, linetype, size, stroke)

continuous x , continuous y

f <- geom\_abline(aes(intercept = 0, slope = 1))

b + geom\_hline(aes(intercept = lat))

b + geom\_vline(aes(intercept = long))

b + geom\_segment(aes(x0 = 1:115, x1 = 1:115, y0 = 1, y1 = 1))

one variable continuous

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

+ f + geom\_bar(aes(class, hwy))

discrete x , continuous y

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

+ f + geom\_boxplot(aes(class, hwy))

continuous x , discrete y

e <- geom\_dotplot(aes(class, hwy))

+ f + geom\_hex(aes(class, hwy))

discrete x , discrete y

g <- ggplot(diamonds, aes(cut, color))

+ f + geom\_count(aes(cut, color))

discrete

d <- ggplot(mpg, aes(class))

+ f + geom\_bar(aes(class))

#### TWO VARIABLES

continuous x , continuous y

e + geom\_label(aes(label = cyl), nudge\_x = 1, nudge\_y = -1, check\_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

b + geom\_jitter(aes(height = 2, width = 2))

x, y, alpha, color, fill, shape, size

e + geom\_point(aes(x, y, alpha, color, fill, shape, size, stroke))

e + geom\_quantile(aes(x, y, alpha, color, group, linetype, size, stroke))

b + geom\_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1), x, ymin, ymax, ymin, alpha, color, fill, linetype, size, stroke)

a + geom\_ribbon(aes(unemploy = 900, ymix = unemploy + 900), ymix, alpha, color, fill, group, linetype, size, stroke)

continuous function

i + geom\_area(aes(x, y, alpha, color, fill, linetype, size, stroke))

i + geom\_line(aes(x, y, alpha, color, group, linetype, size, stroke))

e + geom\_smooth(method = lm, x, y, alpha, color, fill, group, linetype, size, stroke)

e + geom\_text(aes(label = cyl), nudge\_x = 1, nudge\_y = 1, check\_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

visualizing error

d <- data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2)

j = ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

+ f + geom\_boxplot(aes(grp, hwy))

discrete x , discrete y

k = ggplot(diamonds, aes(cut, color))

+ f + geom\_hex(aes(cut, color))

map

data = tibble(murder = USArrests\$Murder,

map = map\_data("state")

k <- ggplot(data, aes(fit = murder))

+ f + geom\_hex(aes(x = state, y = map\_id, fill = map\_id))

+ f + geom\_point(aes(x = state, y = map\_id, fill = map\_id))

discrete

d <- ggplot(mpg, aes(class))

+ f + geom\_bar(aes(class))

continuous x , continuous y

l + geom\_raster(aes(x, y, alpha, color, group, linetype, size, weight))

continuous bivariate distribution

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

+ f + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous bivariate distribution

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

h + geom\_hex2d(aes(x, y, alpha, color, fill, linetype, size, weight))

continuous

<div data-bbox="28

# I 모양을 정하는 geom\_\*( )

## geoms 부분 참고

### 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, group, 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))

### TWO VARIABLES

```
c <- 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

### ggplot2

#### 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, ymin, ymax, alpha, color, fill, size)



# cheat sheet을 함께 살펴보겠습니다.