L09. *ggplot* (3)

Various plots

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

SEOULTECH 서울과학기술대학교 데이터사이언스학과

1. I. One variable
2. II. Two variables (basic)
3. III. Two variables (advanced)

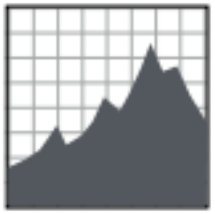
Section 1

I. One variable

One Variable

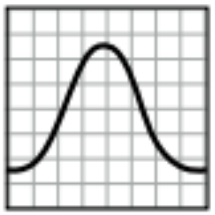
Continuous

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



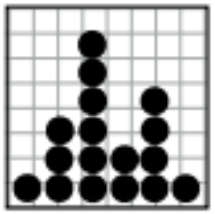
c + geom_area(stat = "bin")

x, y, alpha, color, fill, linetype, size



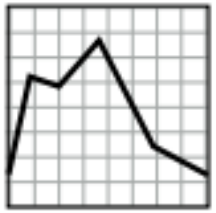
c + geom_density(kernel = "gaussian"**)**

x, y, alpha, color, fill, group, linetype, size, weight



c + geom_dotplot()

x, y, alpha, color, fill



c + geom_freqpoly()

x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)**

x, y, alpha, color, fill, linetype, size, weight

1. Continuous X

Dataset

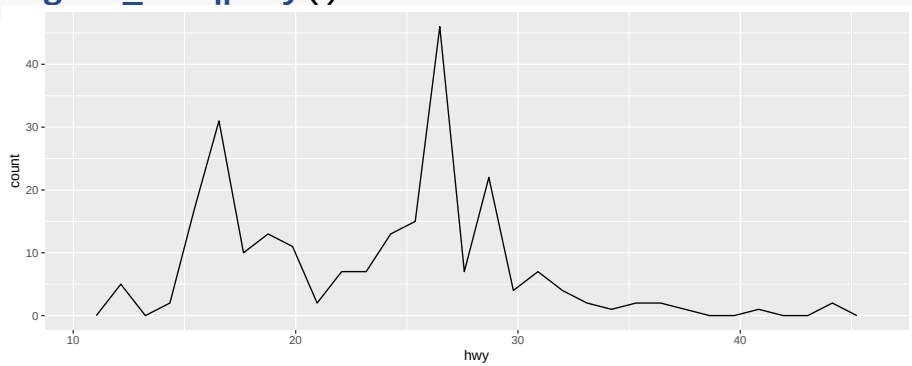
```
mpg$hwy %>% sample(10) # randomly choose 10 elements.
```

```
##  [1] 19 30 17 33 19 17 27 15 29 27
```

geom_freqpoly()

- x, y, alpha, color, group, linetype, size

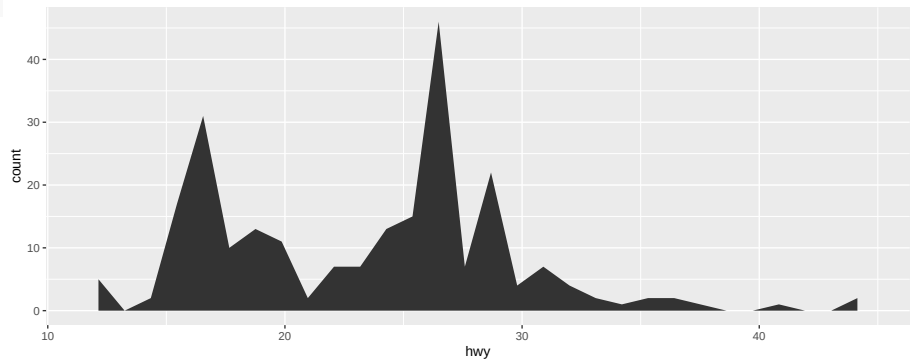
```
ggplot(mpg, aes(x=hwy)) +
  geom_freqpoly()
```



geom_area()

- x, y, alpha, color, fill, linetype, size

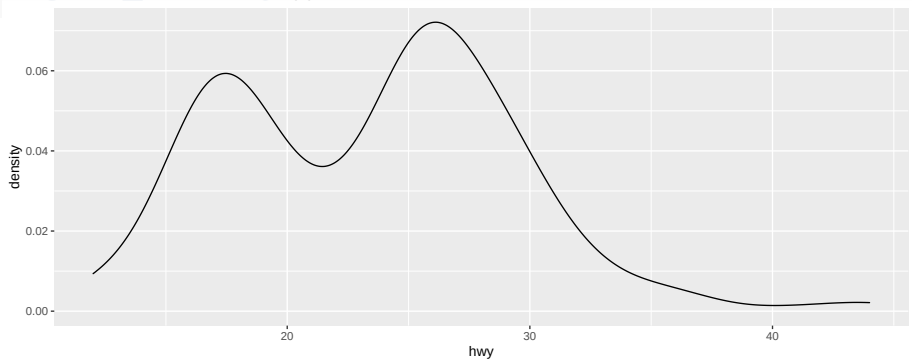
```
ggplot(mpg, aes(x=hwy)) +
  geom_area(stat="bin")
```



geom_density()

- x, y, alpha, color, fill, group, linetype, size, weight

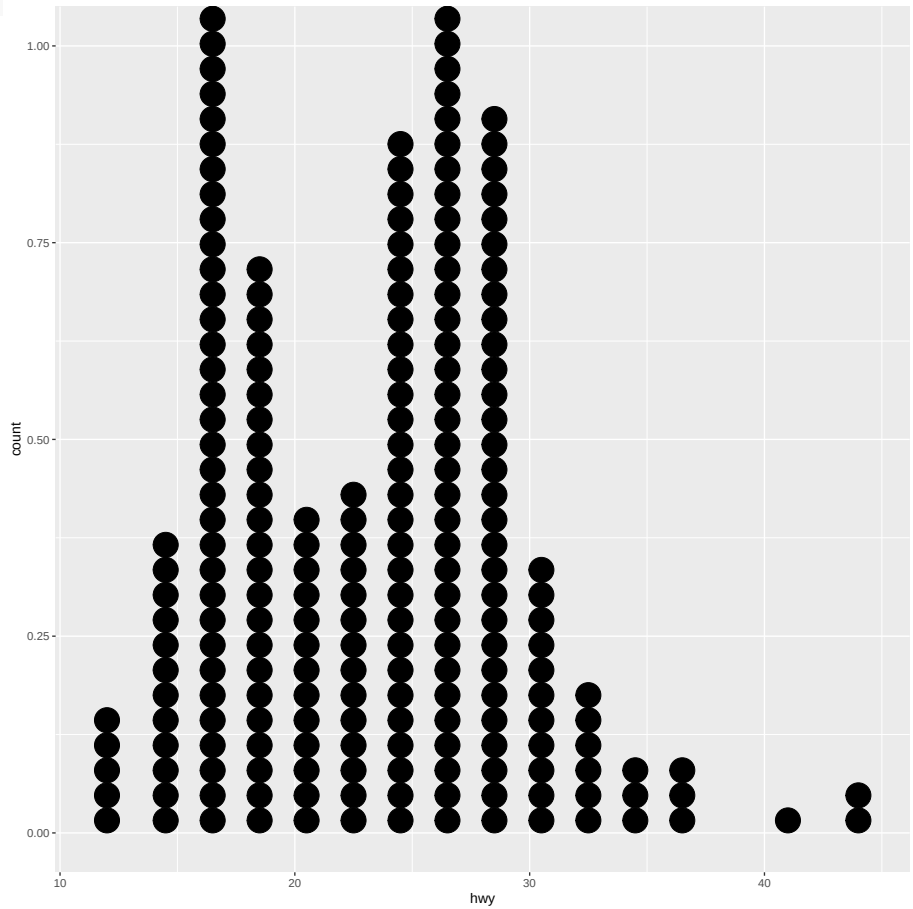
```
ggplot(mpg, aes(x=hwy)) +
  geom_density()
```



geom_dotplot()

- x, y, alpha, color, fill

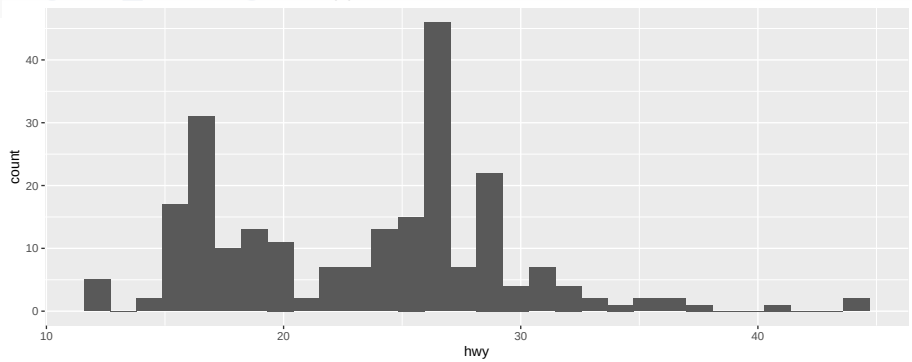
```
ggplot(mpg, aes(x=hwy)) +
  geom_dotplot()
```



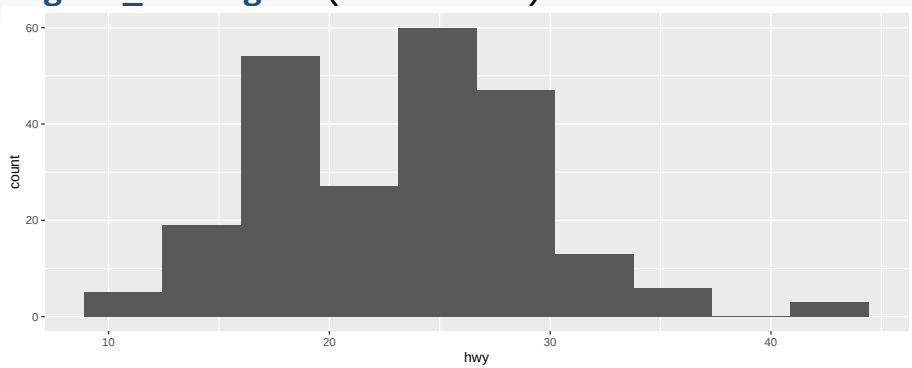
geom_histogram()

- x, y, alpha, color, fill, linetype, size, weight

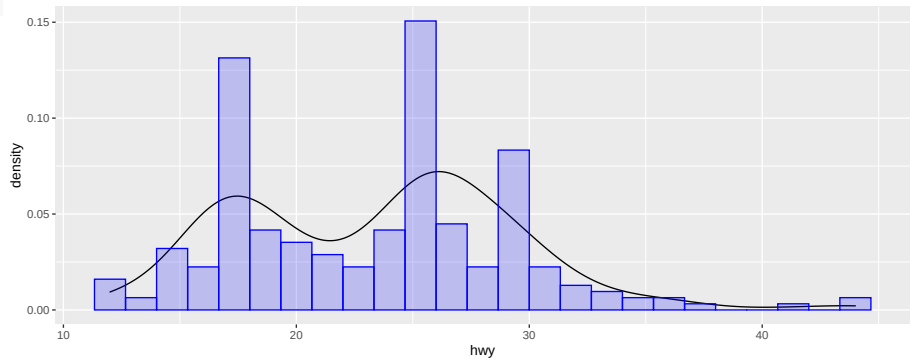
```
ggplot(mpg, aes(x=hwy)) +
  geom_histogram()
```




```
ggplot(mpg, aes(x=hwy)) +
  geom_histogram(bins = 10)
```




```
ggplot(mpg, aes(x=hwy)) +
  geom_density() +
  geom_histogram(aes(y = ..density..),
    bins = 25, color = "blue",
    fill = "blue", alpha = 0.2)
```



geom_histogram()은

- default로 bin에 속한 x값을 count하여
- y aesthetic으로 사용하여 표현함.
- count가 아니라 density로 하려면,
- y=..density..를 위와 같이 mapping

2. Discrete X

Dataset

```
mpg$fl %>% unique() # fuel type
```

```
## [1] "p" "r" "e" "d" "c"
```

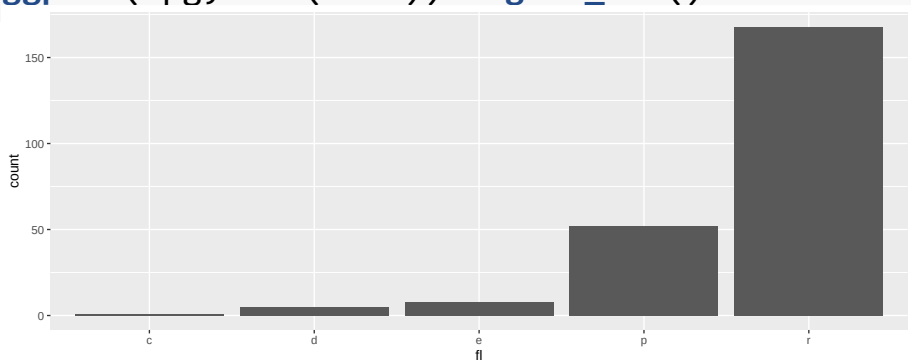
```
table(mpg$fl)
```

```
##
##   c   d   e   p   r
##   1   5   8  52 168
```

geom_bar()

- x, alpha, color, fill, linetype, size, weight

```
ggplot(mpg, aes(x=fl)) + geom_bar()
```

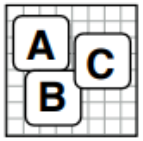


Section 2

II. Two variables (basic)

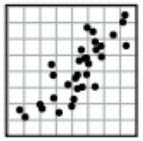
Continuous X, Continuous Y

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



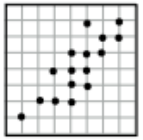
e + **geom_label(**aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)**

x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



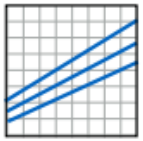
e + **geom_jitter(**height = 2, width = 2**)**

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



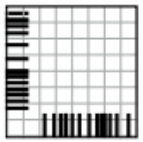
e + **geom_point()**

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



e + **geom_quantile()**

x, y, alpha, color, group, linetype, size, weight

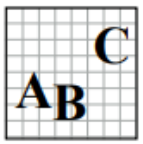


e + **geom_rug(**sides = "bl"**)**

x, y, alpha, color, linetype, size

e + **geom_smooth(**method = lm**)**

x, y, alpha, color, fill, group, linetype, size, weight

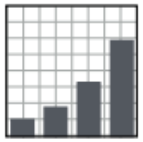


e + **geom_text(**aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)**

x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

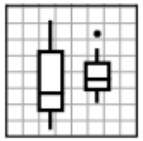
Discrete X, Continuous Y

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



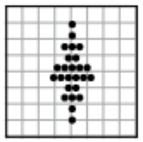
f + **geom_col()**

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"**)**

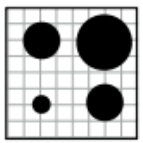
x, y, alpha, color, fill, group

f + **geom_violin(**scale = "area"**)**

x, y, alpha, color, fill, group, linetype, size, weight

Discrete X, Discrete Y

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



g + **geom_count()**

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

Continuous X, Continuous Y

Dataset

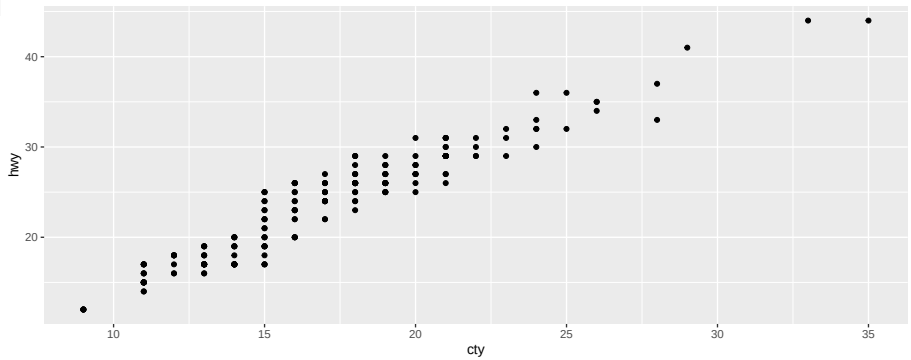
```
mpg %>% select(cty, hwy, manufacturer) %>% head()
```

```
## # A tibble: 6 x 3
##     cty   hwy manufacturer
##   <int> <int> <chr>
## 1    18    29 audi
## 2    21    29 audi
## 3    20    31 audi
## 4    21    30 audi
## 5    16    26 audi
## 6    18    26 audi
```

geom_point()

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

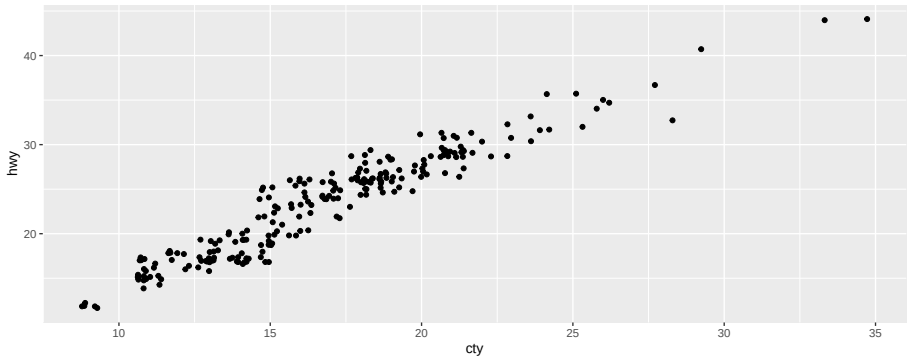
```
ggplot(mpg, aes(x=cty, y=hwy)) + geom_point()
```



geom_jitter()

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

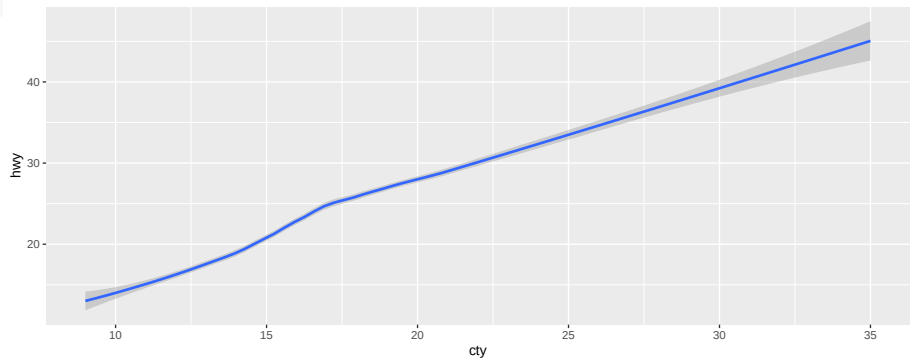
```
ggplot(mpg, aes(x=cty, y=hwy)) + geom_jitter()
```



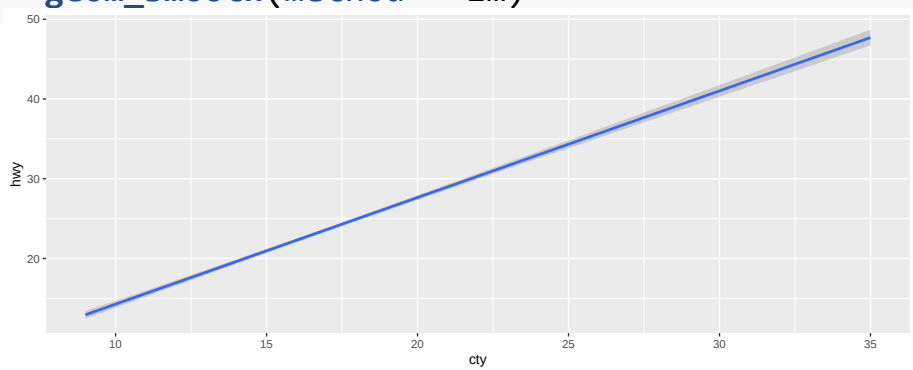
geom_smooth()

- x, y, alpha, color, fill, group, linetype, size, weight

```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_smooth(method = loess)
```



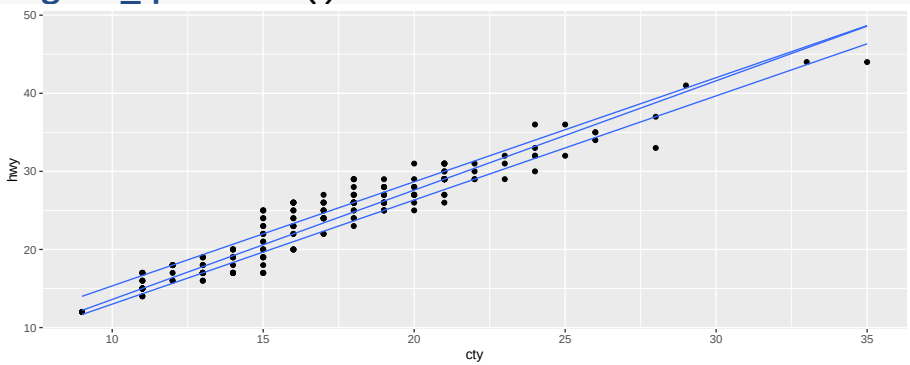

```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_smooth(method = lm)
```



geom_quantile()

- x, y, alpha, color, group, linetype, size, weight

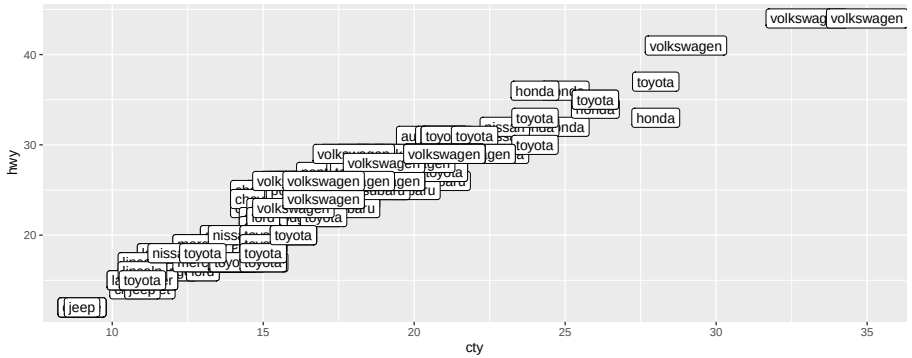
```
# install.packages("quantreg")
# (cf. Quantile regression)
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_point() +
  geom_quantile()
```



geom_label()

- x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

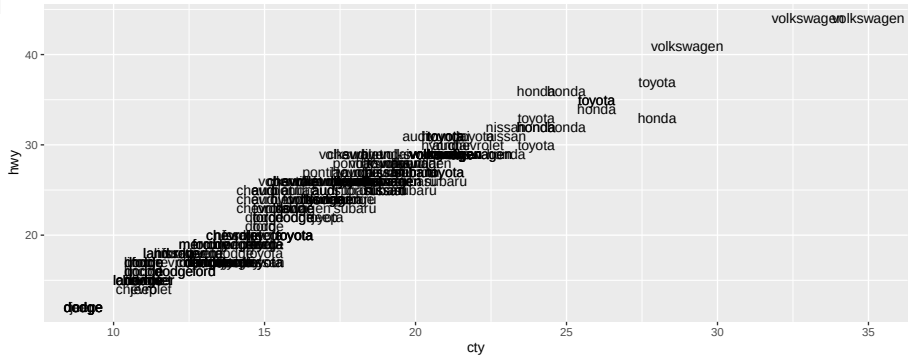
```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_label(aes(label=manufacturer))
```



geom_text()

- x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

```
ggplot(mpg, aes(x=cty, y=hwy)) +
  geom_text(aes(label=manufacturer))
```



Discrete X, Continuous Y

Dataset

```
dim(mpg)
```

```
## [1] 234  11
```

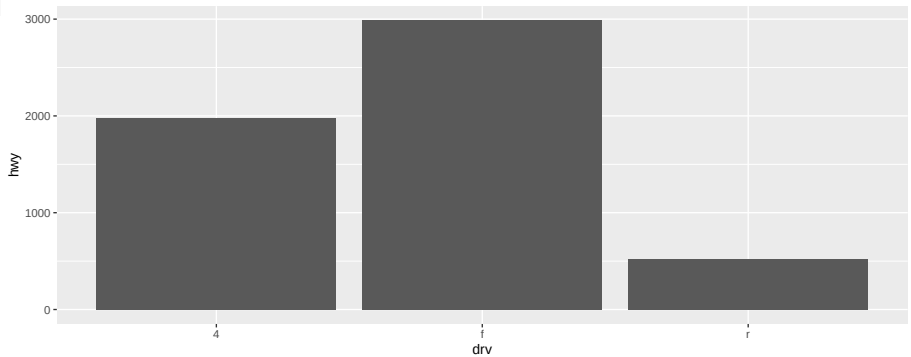
```
# `sample_n()` is from dplyr
mpg %>% select(drv, hwy) %>% sample_n(10)
```

```
## # A tibble: 10 x 2
##    drv     hwy
##    <chr> <int>
##  1 f        23
##  2 4        20
##  3 f        27
##  4 f        24
##  5 f        28
##  6 r        21
##  7 4        25
##  8 f        24
##  9 r        23
## 10 4        19
```

geom_col()

- x, y, alpha, color, fill, group, linetype, size

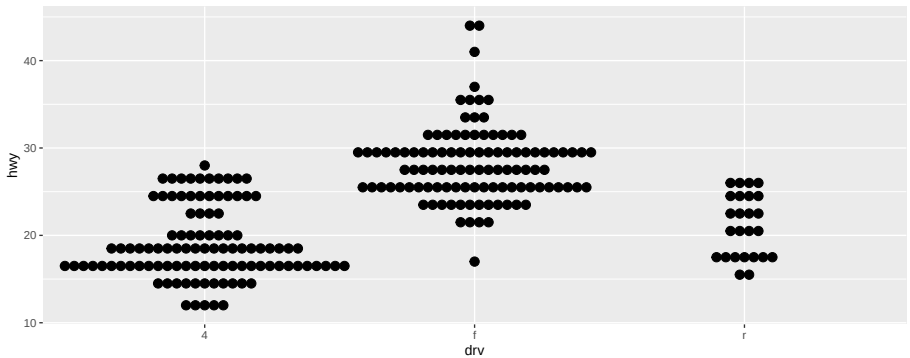
```
ggplot(mpg, aes(x=drv, y=hwy)) + geom_col()
```



geom_dotplot()

- x, y, alpha, color, fill, group

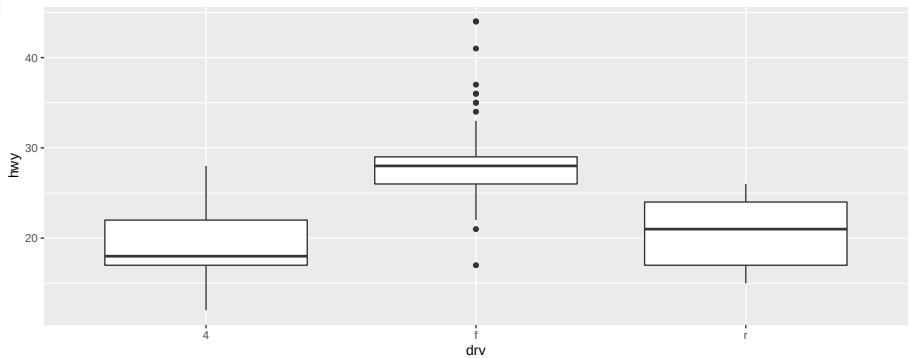
```
ggplot(mpg, aes(x=drv, y=hwy)) +
  geom_dotplot(binaxis = "y", stackdir = "center")
```



geom_boxplot()

- x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

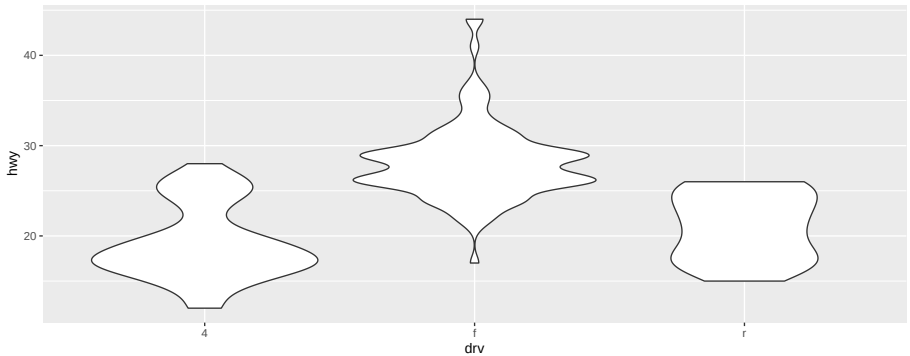
```
ggplot(mpg, aes(x=drv, y=hwy)) +
  geom_boxplot()
```



geom_violin()

- x, y, alpha, color, fill, group, linetype, size, weight

```
ggplot(mpg, aes(x=drv, y=hwy)) +
  geom_violin()
```



Discrete X, Discrete Y

Dataset

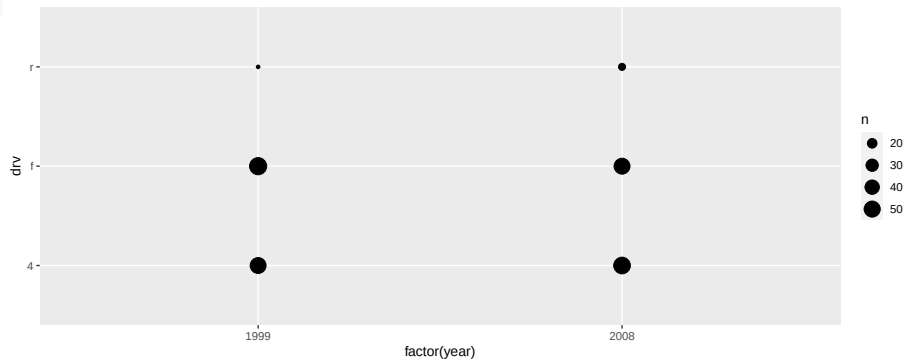
```
mpg %>% select(year, drv) %>% sample_n(10)
```

```
## # A tibble: 10 x 2
##     year drv
##    <int> <chr>
##  1  2008 f
##  2  1999 f
##  3  2008 f
##  4  2008 r
##  5  2008 f
##  6  2008 f
##  7  2008 f
##  8  2008 4
##  9  1999 f
## 10  2008 4
```

geom_count()

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

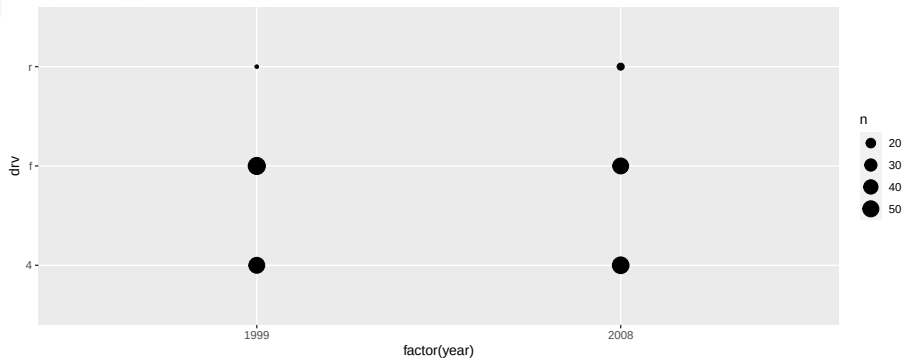
```
ggplot(mpg, aes(x=factor(year), y=drv)) +
  geom_count()
```



Some extra

1. 마지막 슬라이드는 아래와 같은데, geom_text와 연관시킬 수는 없을까?

```
ggplot(mpg, aes(x=factor(year), y=drv)) +
  geom_count()
```



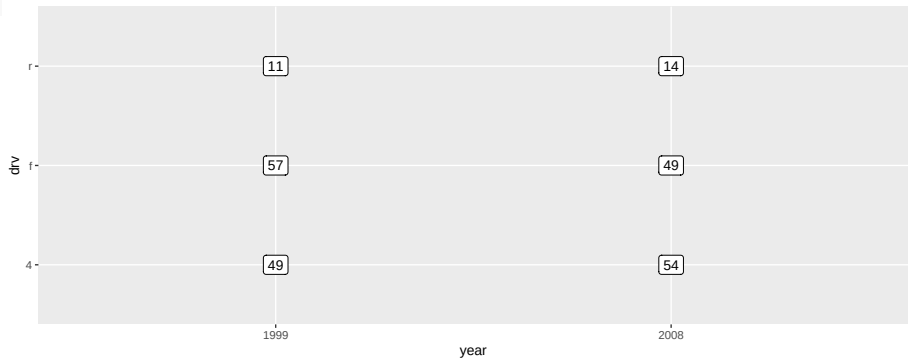
2. 결국 위의 그림도 아래를 이용한 것일 테니까…

```
mpg2 <- mpg %>%
  group_by(year=factor(year), drv=factor(drv)) %
  summarise(count=n())
mpg2
```

```
## # A tibble: 6 x 3
## # Groups:   year [2]
##   year  drv   count
##   <fct> <fct> <int>
## 1 1999  4        49
## 2 1999  f        57
## 3 1999  r        11
## 4 2008  4        54
## 5 2008  f        49
## 6 2008  r        14
```

3. `geom_text()`와 `geom_label()`는 앞에서 conti X와 conti Y라고 하였지만 반드시 conti일 필요는 없지 않을까?

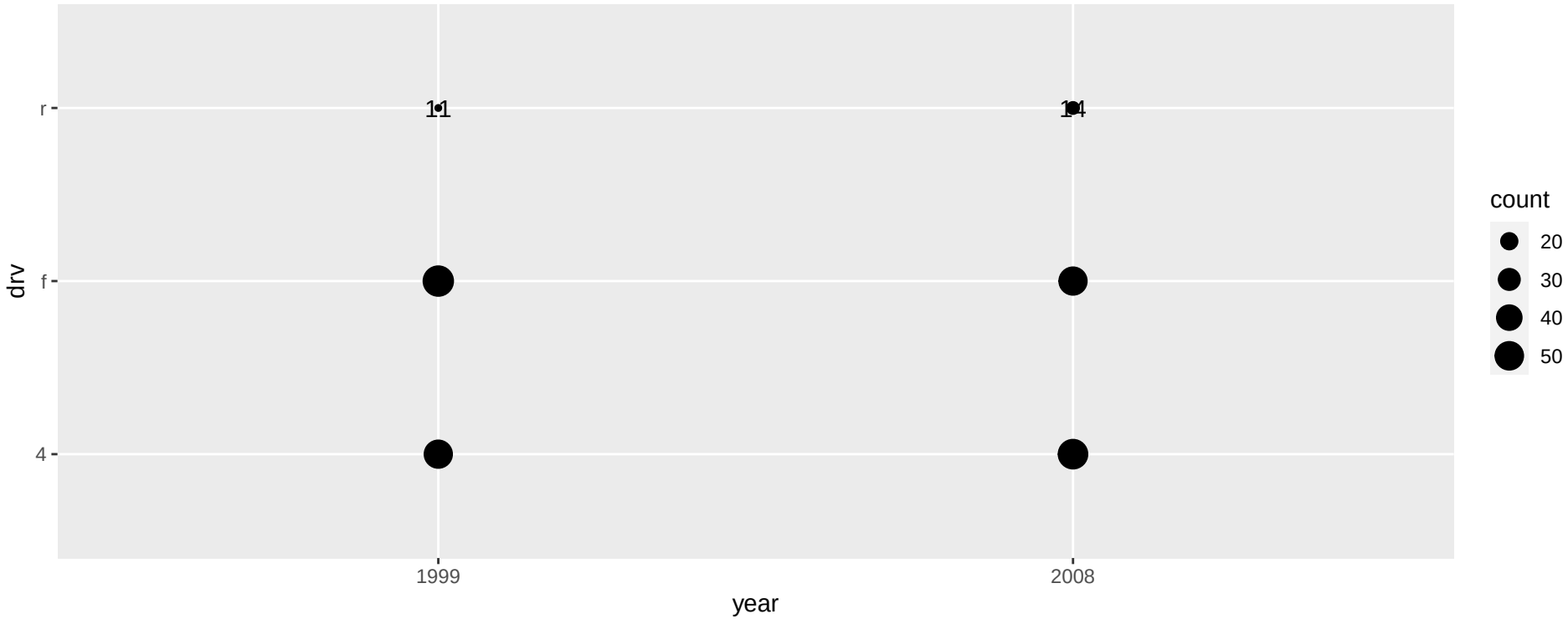
```
mpg2 %>% ggplot(aes(x=year, y=drv)) +
  geom_label(aes(label=count))
```




```
mpg2 %>% ggplot(aes(x=year, y=drv)) +
  geom_text(aes(label=count))
```

4. geom_point()와 geom_text()를 동시에 사용하면 더 informative 하지 않을까?

```
fig <- mpg2 %>% ggplot(aes(x=year, y=drv)) +
  geom_text(aes(label=count)) +
  geom_point(aes(size=count))
fig
```



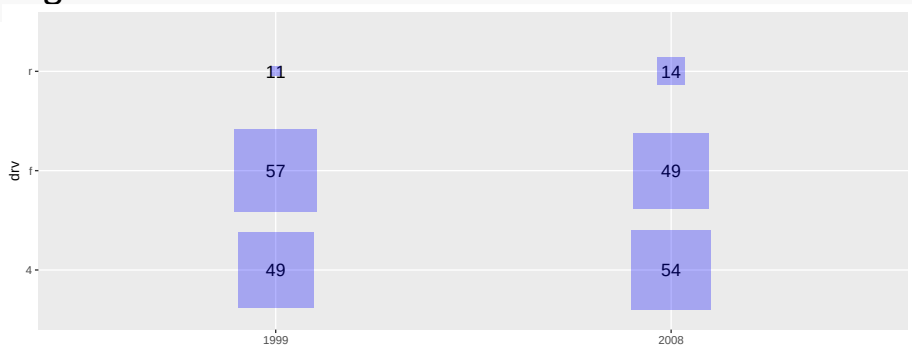
5. 개선사항
   1. text size가 작다
   2. text가 있다면 굳이 size legend는 없어도 된다.
   3. 점의 크기가 너무 작고 대비가 안된다.

6 개선사항 반영

```
fig <- mpg2 %>% ggplot(aes(x=year, y=drv)) +
  geom_text(aes(label=count), size=5) +
  geom_point(aes(size=count), alpha=0.3, color="blue", shape = "square") +
  theme(legend.position = "none") +
  scale_size_continuous(range = c(3, 30))
```

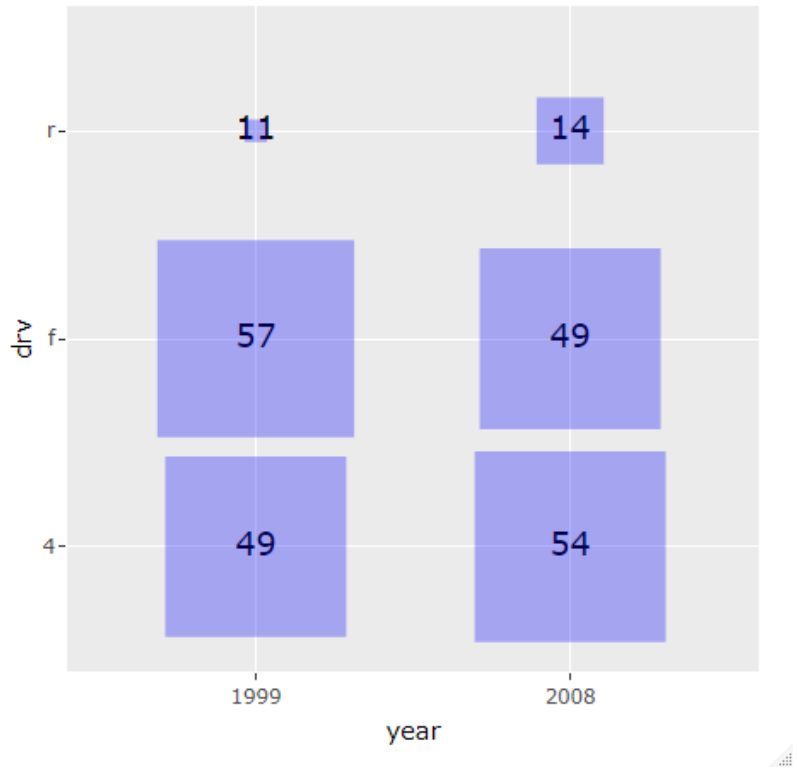
- `size=5`: text의 사이즈를 일괄적으로 크게 했음
- `alpha=0.3`: point를 투명하게 하여 text의 가독성을 높임
- `color="blue"`: point의 색을 바꾸어 text와의 대비를 높임
- `shape="square"`: point를 원이 아닌 사각으로 하여 text를 더 잘 수용
- `theme(legend.position = "none")`: legend를 없애버림
- `scale_size_continuous(range = c(3, 30))`: `size=count`의 효과를 더 크게 보이게 하기 위해서 스케일을 조정하였음.

```
fig
```



7. Finally…

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

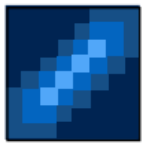


Section 3

III. Two variables (advanced)

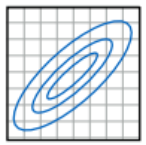
Continuous Bivariate Distribution

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()

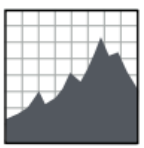
x, y, alpha, colour, group, linetype, size

h + geom_hex()

x, y, alpha, colour, fill, size

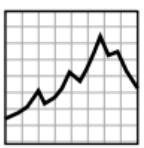
Continuous Function

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



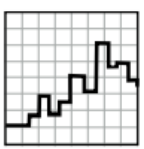
i + geom_area()

x, y, alpha, color, fill, linetype, size



i + geom_line()

x, y, alpha, color, group, linetype, size



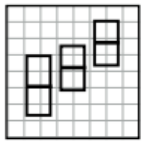
i + geom_step(direction = "hv"**)**

x, y, alpha, color, group, linetype, size

Visualizing error

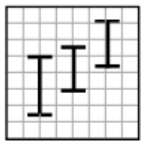
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)

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



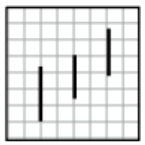
j + geom_crossbar(fatten = 2**)**

x, y, ymax, ymin, alpha, color, fill, group, linetype, size



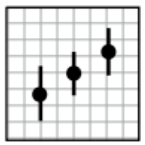
j + geom_errorbar()

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

Continuous Bivariate Distribution

A bivariate probability density function $f(x, y)$를 그리는 기능

`diamonds` dataset

- A dataset containing the prices and other attributes of almost 54,000 diamonds.
- The variables are as follows: `carat, cut, color, clarity, depth, table, price, x, y, z`

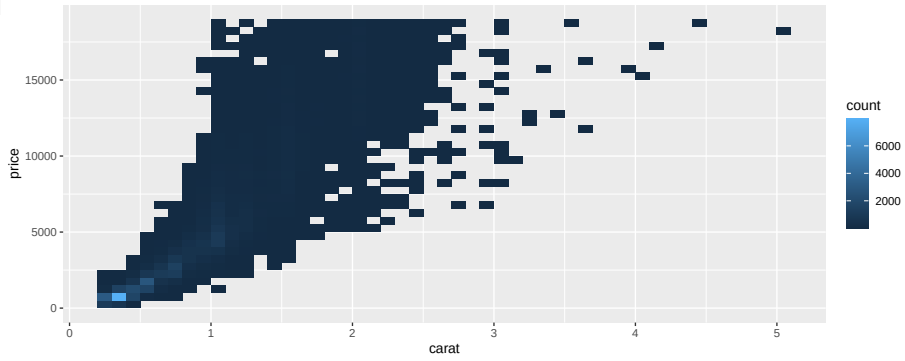
```
diamonds %>%
  select(carat, price) %>%
  sample_n(6)
```

```
## # A tibble: 6 x 2
##   carat price
##   <dbl> <int>
## 1  0.3    421
## 2  0.7   2196
## 3  0.31   802
```

`geom_bin2d()`

- `x, y, alpha, color, fill, linetype, size, weight`
- 바로 위 예제의 continuous 버전!

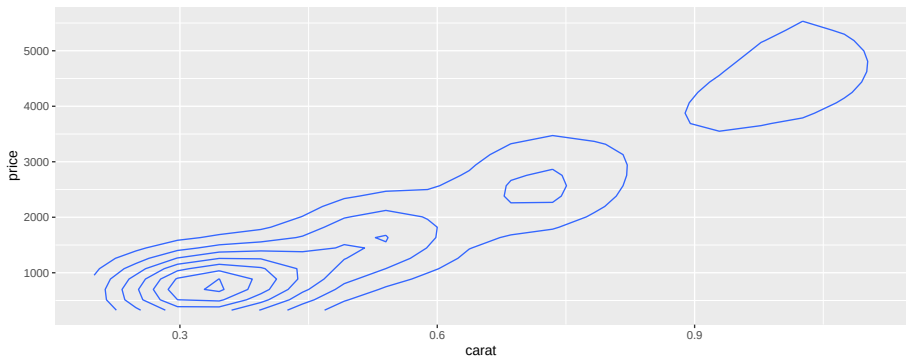
```
ggplot(diamonds, aes(x=carat, y=price)) +
  geom_bin2d(binwidth = c(0.1, 500))
```



geom_density2d()

- x, y, alpha, colour, group, linetype, size

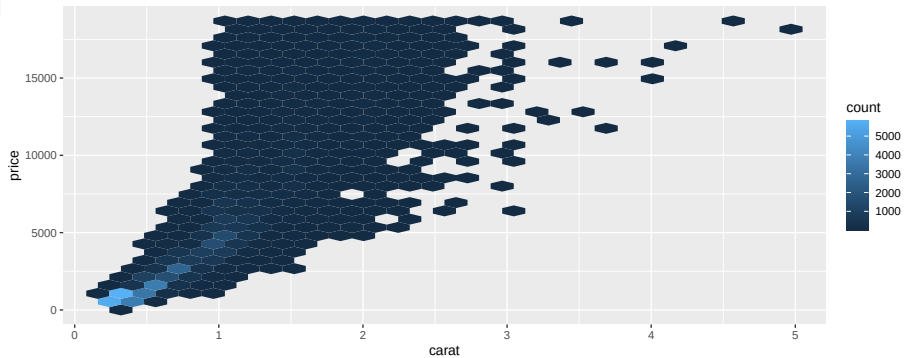
```
ggplot(diamonds, aes(x=carat, y=price)) +
  geom_density2d()
```



geom_hex()

- x, y, alpha, colour, fill, size

```
ggplot(diamonds, aes(x=carat, y=price)) +
  geom_hex()
```



Continuous Function (time-series)

`economics` dataset

- This dataset was produced from US economic time series data available from http://research.stlouisfed.org/fred2.
- `economics` is in 'wide' format, `economics_long` is in 'long' format.

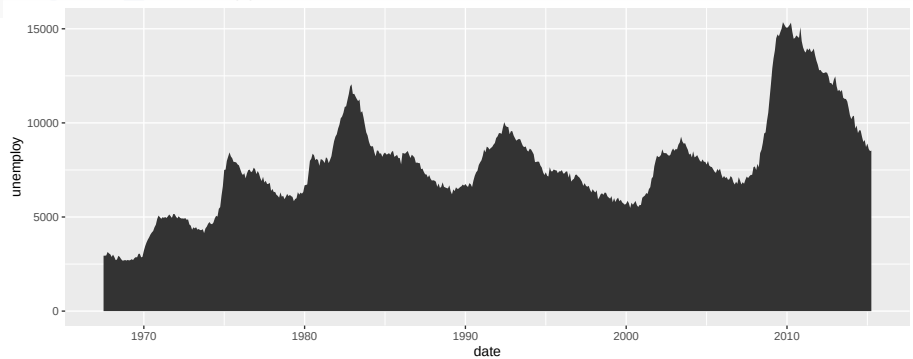
```
economics %>%
  select(date, unemploy) %>%
  head(3)
```

```
## # A tibble: 3 x 2
##   date       unemploy
##   <date>        <dbl>
## 1 1967-07-01     2944
## 2 1967-08-01     2945
## 3 1967-09-01     2958
```

`geom_area()`

- `x, y, alpha, color, fill, linetype, size`
- One variable에서의 `geom_area()`와 비교!

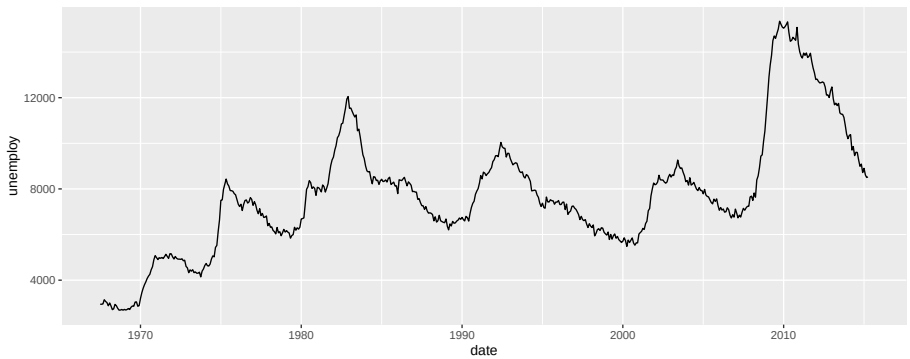
```
ggplot(economics, aes(date, unemploy)) +
  geom_area()
```



geom_line()

- x, y, alpha, color, group, linetype, size

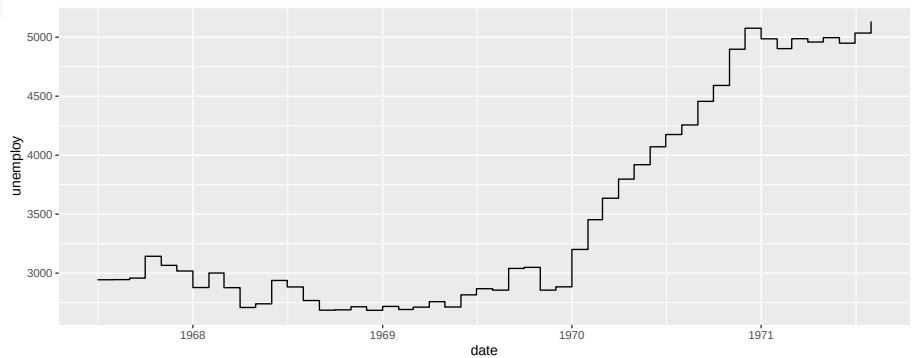
```
ggplot(economics, aes(date, unemploy)) +
  geom_line()
```



geom_step()

- x, y, alpha, color, group, linetype, size

```
economics %>% head(50) %>%
ggplot(aes(date, unemploy)) +
  geom_step(direction = "hv")
```



Visualizing error

Dataset prep.

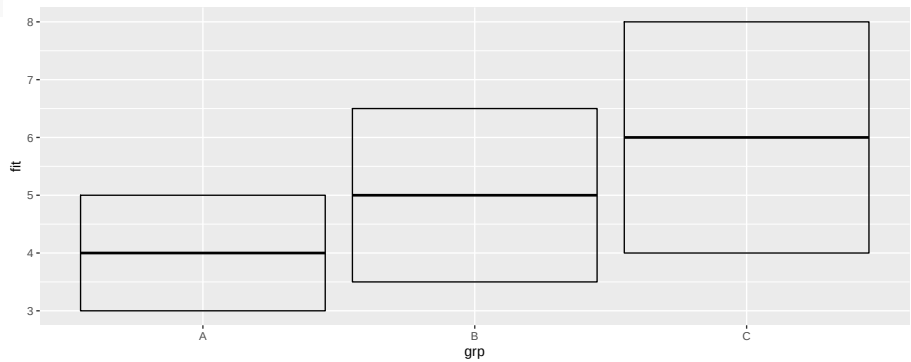
```
df <- data.frame(grp = c("A", "B", "C"),
                 fit = 4:6,
                 se = c(1,1.5,2))
df
```

```
##   grp fit  se
## 1   A   4 1.0
## 2   B   5 1.5
## 3   C   6 2.0
```

geom_crossbar(fatten = 2)

- x, y, ymax, ymin, alpha, color, fill, group, linetype, size

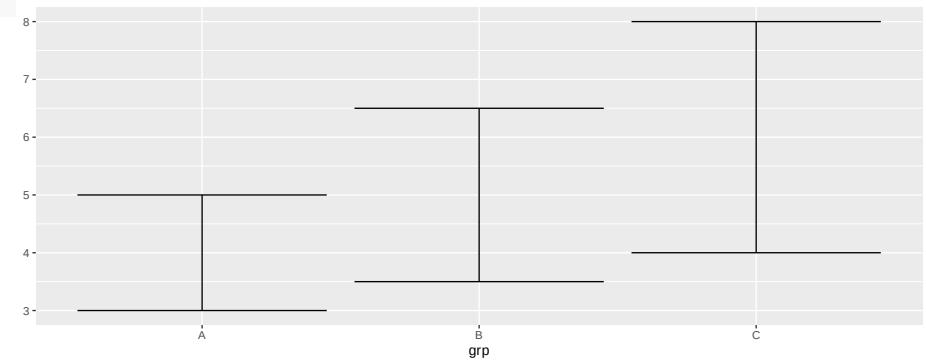
```
ggplot(df, aes(grp, fit,
               ymin = fit-se, ymax = fit+se)) +
  geom_crossbar(fatten = 2)
```



geom_errorbar() (also geom_errorbarh())

- x, ymax, ymin, alpha, color, group, linetype, size, width

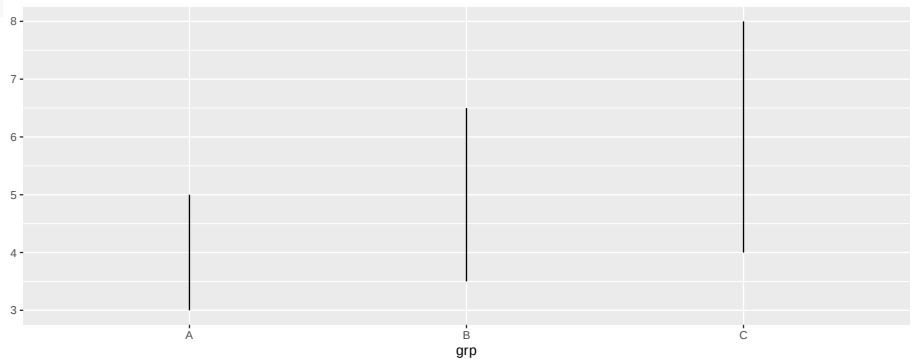
```
ggplot(df, aes(grp,
               ymin = fit-se, ymax = fit+se)) +
  geom_errorbar()
```



geom_linerange()

- x, ymin, ymax, alpha, color, group, linetype, size

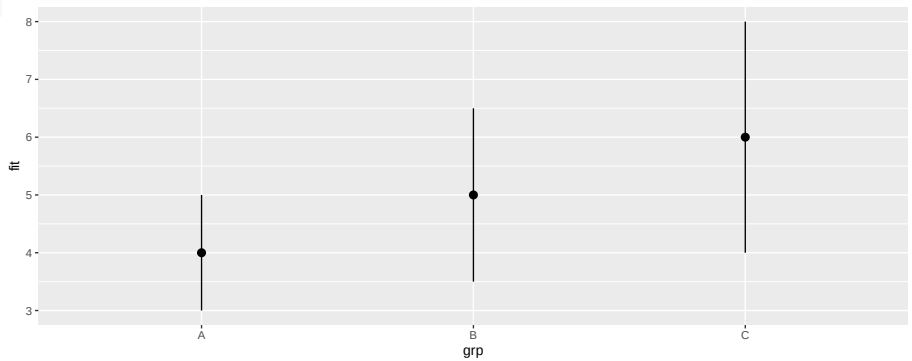
```
ggplot(df, aes(grp,
               ymin = fit-se, ymax = fit+se)) +
  geom_linerange()
```



geom_pointrange()

- x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

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

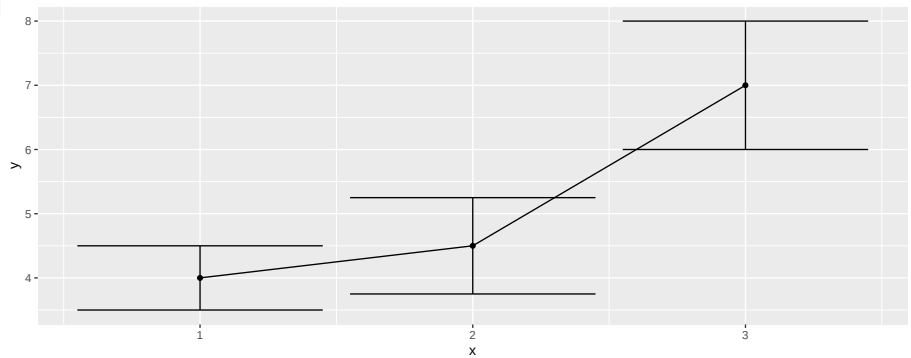


A realistic example

```
df <- data.frame(
  x = 1:3, y = c(4,4.5,7), se=seq(0.5,1,0.25))
df
```

```
##   x   y   se
## 1 1 4.0 0.50
## 2 2 4.5 0.75
## 3 3 7.0 1.00
```

```
ggplot(df) +
  geom_errorbar(aes(x=x, ymax=y+se, ymin=y-se)) +
  geom_line(aes(x=x, y=y)) +
  geom_point(aes(x=x, y=y))
```



Maps

Data prep.

```
data <- data.frame(murder = USArrests$Murder,
                   state = tolower(rownames(USArrests)))
data %>% head(3)
```

```
##   murder   state
## 1   13.2 alabama
## 2   10.0  alaska
## 3    8.1 arizona
```

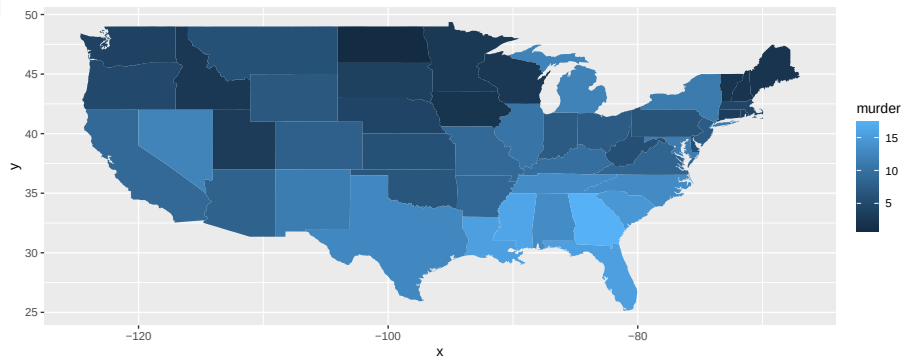
```
map <- map_data("state")
map %>% head(3)
```

```
##      long     lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama    <NA>
## 2 -87.48493 30.37249     1     2 alabama    <NA>
## 3 -87.52503 30.37249     1     3 alabama    <NA>
```

geom_map()

- map_id, alpha, color, fill, linetype, size

```
ggplot(data, aes(fill = murder)) +
  geom_map(aes(map_id = state), map = map) +
  expand_limits(x = map$long, y = map$lat)
```



A quick summary

차트의 종류	geometric object	variables
Barchart	`geom_bar()`	`x`: disc.
Histogram	`geom_histogram()`	`x`: conti.
Density	`geom_density()`	`x`: conti.
Scatterplot	`geom_point()`	`x`: conti., `y`:conti.
Bubblechart	`geom_point()` with `size`	`x`: conti., `y`:conti., `size`: numeric
Linechart	`geom_line()`	`x`: conti., `y`:conti.
(fitted line)	`geom_smooth(method="lm")`	`x`: conti., `y`:conti.
Boxplot	`geom_boxplot()`	`x`: disc., `y`: conti.
Violin	`geom_violin()`	`x`: disc., `y`: conti.

```
"Tantum videmus quantum scimus."
```

```
## [1] "Tantum videmus quantum scimus."
```