CPE 213 Data Models
(a.k.a. Data Modeling and Visualization)

Lecture 3: Basic types of data visualization

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

BIG DATA EXPERIENCE

Lecture 3 Overview

- Data visualization with ggplot2
- What types of visuals to use?

KMUTT g-able

What is ggplot2?

- ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics.
- You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

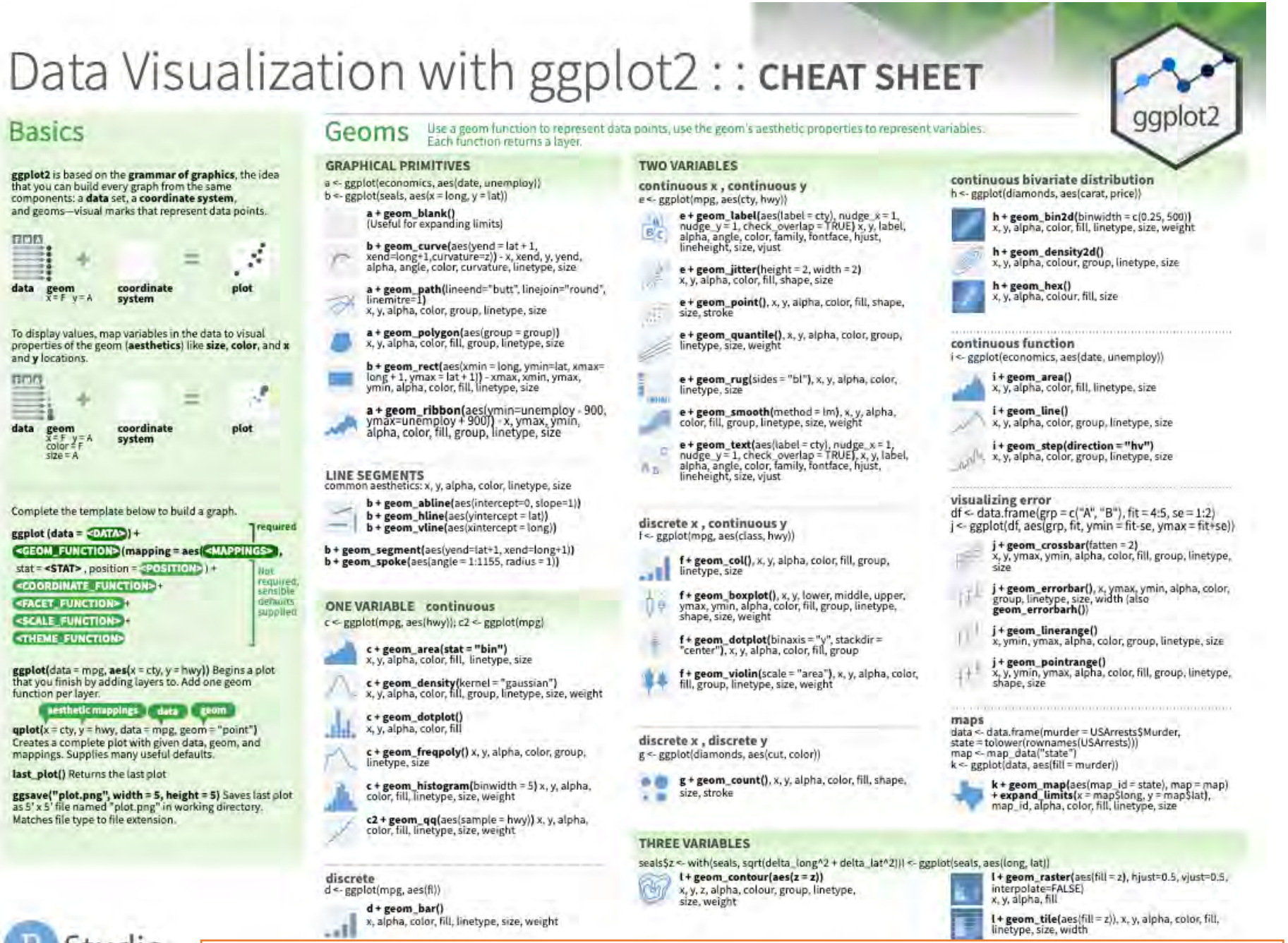
```
library(tidyverse) #ggplot2 is a part of tidyverse
p <- ggplot(data = mpg)
p + geom_col(mapping = aes(x = class, y = hwy))
```

Data

What to plot? + Mapping

BIG DATA EXPERIENCE

What can you do with ggplot2?



Data Visualization with ggplot2 : : CHEAT SHEET

ggplot2

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.

data + geom (x = F, y = A) = coordinate system + plot

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

data + geom (x = F, y = A, color = F, size = A) = coordinate system + plot

Complete the template below to build a graph.

required

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

Not required, sensible defaults supplied

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

aesthetic mappings | data | geom

qplot(x = cty, y = hwy, data = mpg, geom = "point"**)** Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("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))
```

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

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)** x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete

```
d <- ggplot(mpg, aes(fl))
```

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

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

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

g + geom_count(), x, y, alpha, color, fill, shape, size, stroke

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")
k <- ggplot(data, aes(fill = murder))
```

k + geom_map(aes(map_id = state), map = map**)**
+ expand_limits(x = map$long, y = map$lat), map_id, alpha, color, fill, linetype, size

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)))l <- ggplot(seals, aes(long, lat))
```

l + geom_contour(aes(z = z))
x, y, z, alpha, colour, group, linetype, size, weight

l + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE)
x, y, alpha, fill

l + geom_tile(aes(fill = z)), x, y, alpha, color, fill, linetype, size, width

RStudio

https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

KMUTT g-able

Why ggplot2?

Advantages of ggplot2

- consistent underlying grammar of graphics (Wilkinson, 2005)
- plot specification at a high level of abstraction
- very flexible
- theme system for polishing plot appearance
- mature and complete graphics system
- many users, active community

When to not using ggplot2:

- 3-dimensional graphics (see the rgl package)
- Graph-theory type graphs (see the igraph package)
- Interactive graphics (see the ggvis package)

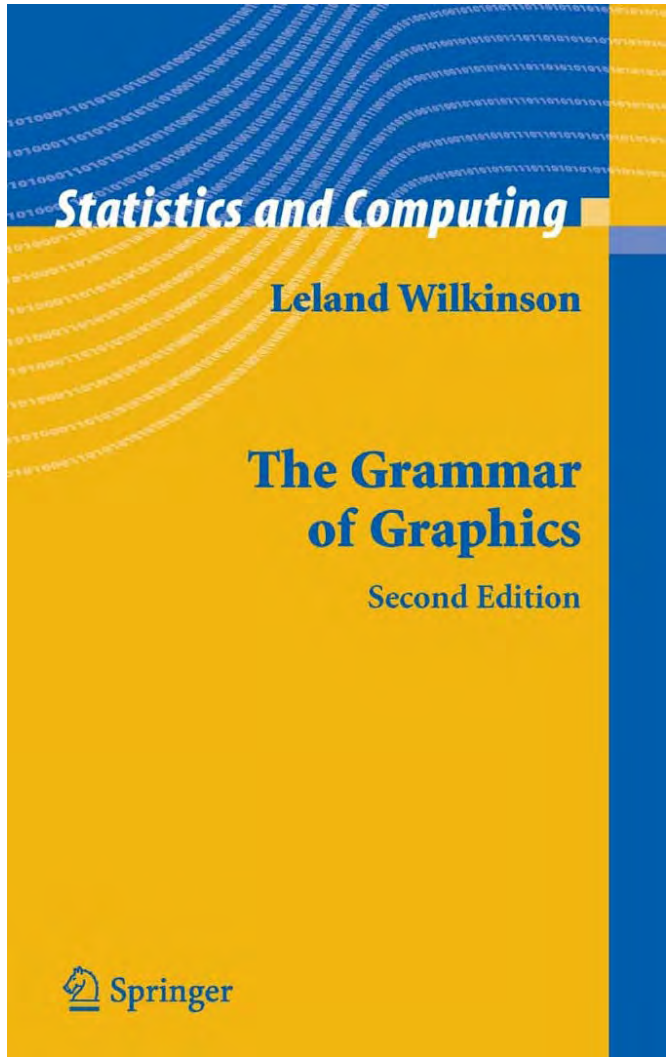
What Is The Grammar Of Graphics?

- The basic idea: independently specify plot building blocks and combine them to create just about any kind of graphical display you want.
- Building blocks of a graph include:
  - data
  - aesthetic mapping
  - geometric object
  - statistical transformations
  - scales
  - coordinate system
  - position adjustments
  - faceting

BIG DATA EXPERIENCE

Origin of ggplot2

- Foundation for producing almost every quantitative graphic found
- Designed for a distributed computing environment
- Make graphics easier



BIG DATA EXPERIENCE

Data: mpg (in tidyverse/ggplot2)

Fuel economy data from 1999 and 2008 for 38 popular models of car

```
> mpg
# A tibble: 234 x 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
 6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
 7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
 8 audi         a4 quattro   1.8  1999     4 manu… 4        18    26 p     comp…
 9 audi         a4 quattro   1.8  1999     4 auto… 4        16    25 p     comp…
10 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
# ... with 224 more rows
```

KMUTT g-able

BIG DATA EXPERIENCE

mpg: meaning

Field	Meaning
manufacturer model	model name
displ	engine displacement, in litres
year	year of manufacturer
cyl	number of cylinders
trans	type of transmission
drv	f = front-wheel drive, r = rear, 4 = 4wd
cty	city miles per gallon
hwy	highway miles per gallon
fl	fuel type
class	"type" of car

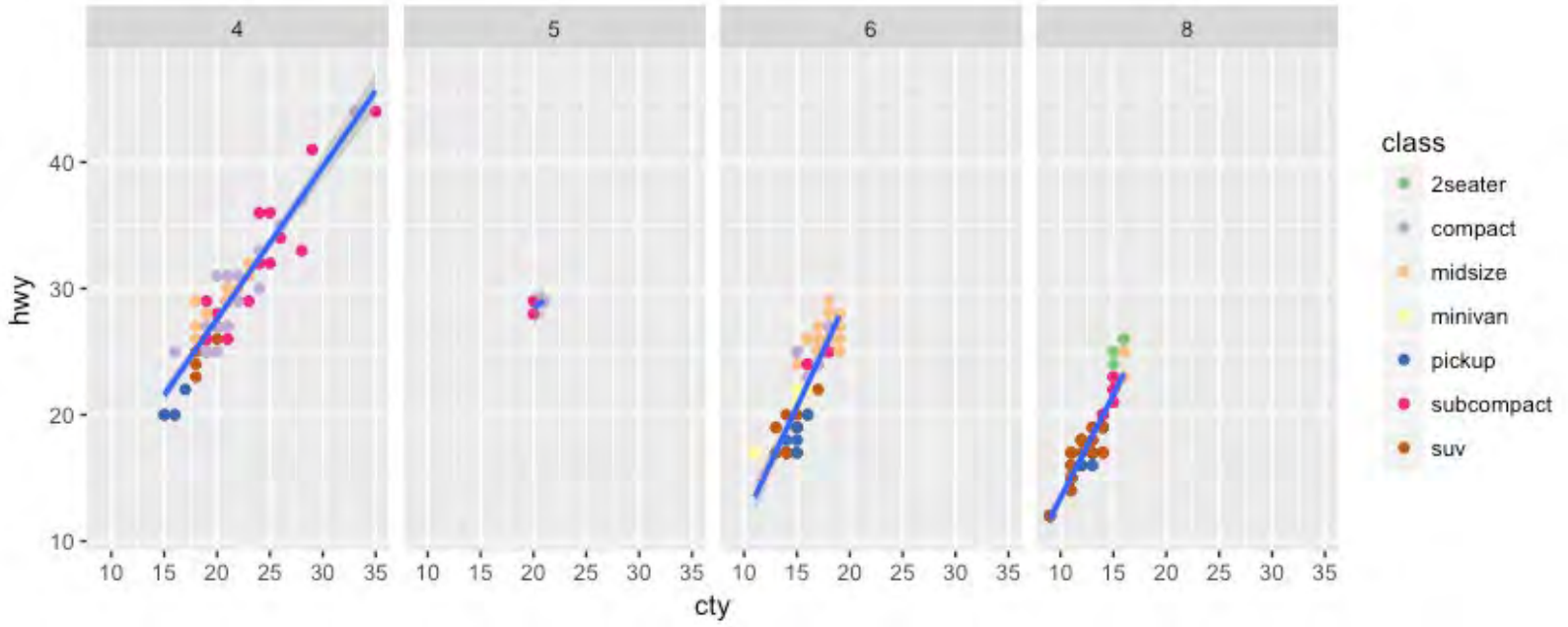
KMUTT g-able

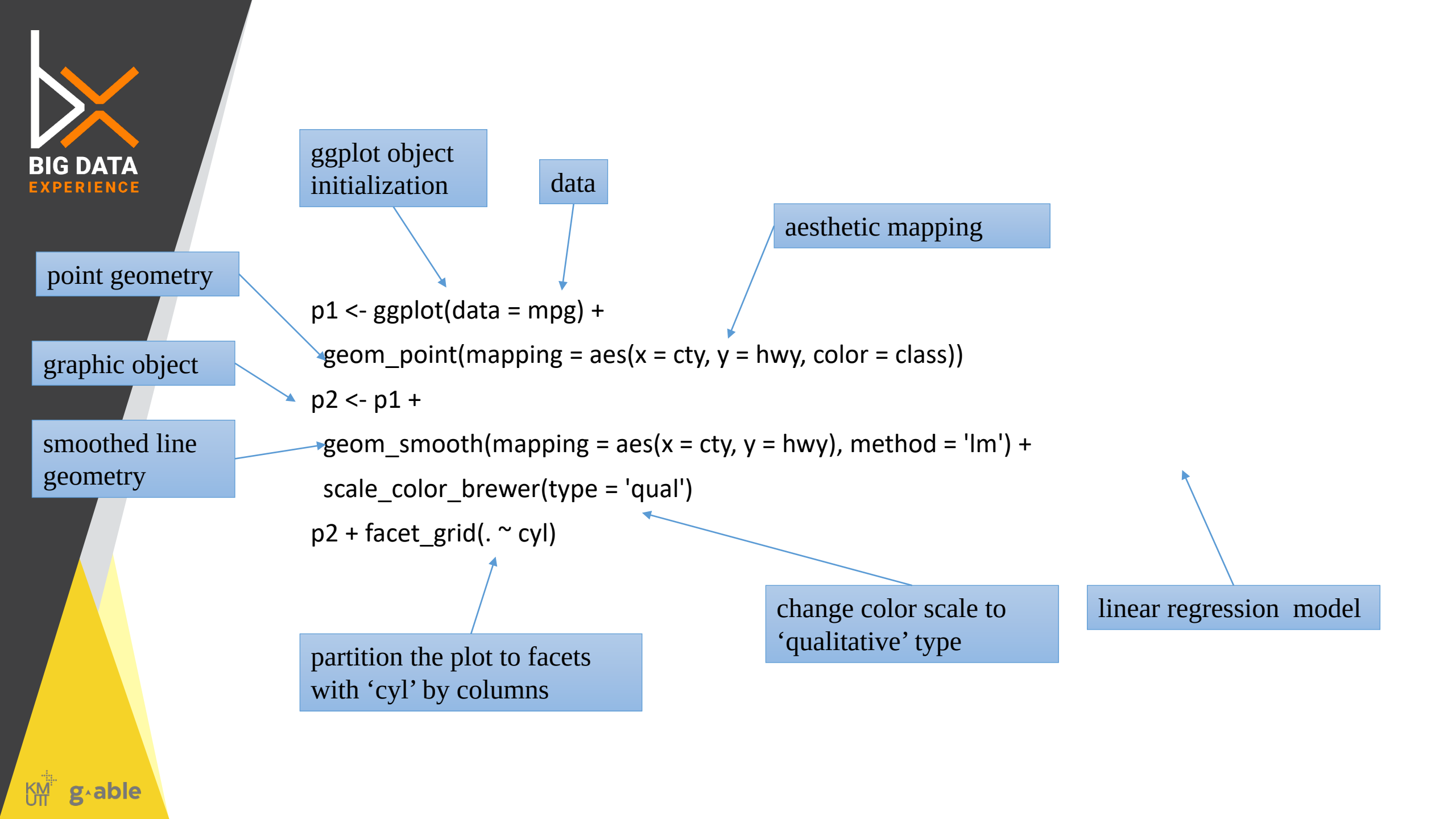
mpg: data summary

```
> mpg %>% mutate_if(is.character, as.factor) -> mpg1
> summary(mpg1)
     manufacturer                    model         displ            year            cyl       
 dodge      :37   caravan 2wd           : 11   Min.   :1.600   Min.   :1999   Min.   :4.000  
 toyota     :34   ram 1500 pickup 4wd   : 10   1st Qu.:2.400   1st Qu.:1999   1st Qu.:4.000  
 volkswagen :27   civic                 :  9   Median :3.300   Median :2004   Median :6.000  
 ford       :25   dakota pickup 4wd     :  9   Mean   :3.472   Mean   :2004   Mean   :5.889  
 chevrolet  :19   jetta                 :  9   3rd Qu.:4.600   3rd Qu.:2008   3rd Qu.:8.000  
 audi       :18   mustang               :  9   Max.   :7.000   Max.   :2008   Max.   :8.000  
 (Other)    :74   (Other)               :177                                                 
       trans       drv             cty              hwy             fl               class    
 auto(l4)  :83   4:103   Min.   : 9.00   Min.   :12.00   c:  1   2seater   : 5  
 manual(m5):58   f:106   1st Qu.:14.00   1st Qu.:18.00   d:  5   compact   :47  
 auto(l5)  :39   r: 25   Median :17.00   Median :24.00   e:  8   midsize   :41  
 manual(m6):19           Mean   :16.86   Mean   :23.44   p: 52   minivan   :11  
 auto(s6)  :16           3rd Qu.:19.00   3rd Qu.:27.00   r:168   pickup    :33  
 auto(l6)  : 6           Max.   :35.00   Max.   :44.00           subcompact:35  
 (Other)   :13                                                   suv       :62  
```

ggplot2: full form example

```
p1 <- ggplot(data = mpg) +
  geom_point(mapping = aes(x = cty, y = hwy, color = class))
p2 <- p1 +
  geom_smooth(mapping = aes(x = cty, y = hwy), method = 'lm') +
  scale_color_brewer(type = 'qual')
p2 + facet_grid(. ~ cyl)
```




ggplot object initialization
data
aesthetic mapping
point geometry
graphic object
smoothed line geometry
p1 <- ggplot(data = mpg) +
geom_point(mapping = aes(x = cty, y = hwy, color = class))
p2 <- p1 +
geom_smooth(mapping = aes(x = cty, y = hwy), method = 'lm') +
scale_color_brewer(type = 'qual')
p2 + facet_grid(. ~ cyl)
partition the plot to facets with ‘cyl’ by columns
change color scale to ‘qualitative’ type
linear regression model

BIG DATA EXPERIENCE

Geometric Objects And Aesthetics Aesthetic Mapping

- In ggplot land, aesthetic means "something you can see". Examples include:
  - position (i.e., on the x and y axes)
  - color ("outside" color)
  - fill ("inside" color)
  - shape (of points)
  - linetype
  - Size

- Each type of geom accepts only a subset of all aesthetics–refer to the geom help pages to see what mappings each geom accepts. Aesthetic mappings are set with the aes() function.

BIG DATA EXPERIENCE

Geometric Objects (geom)

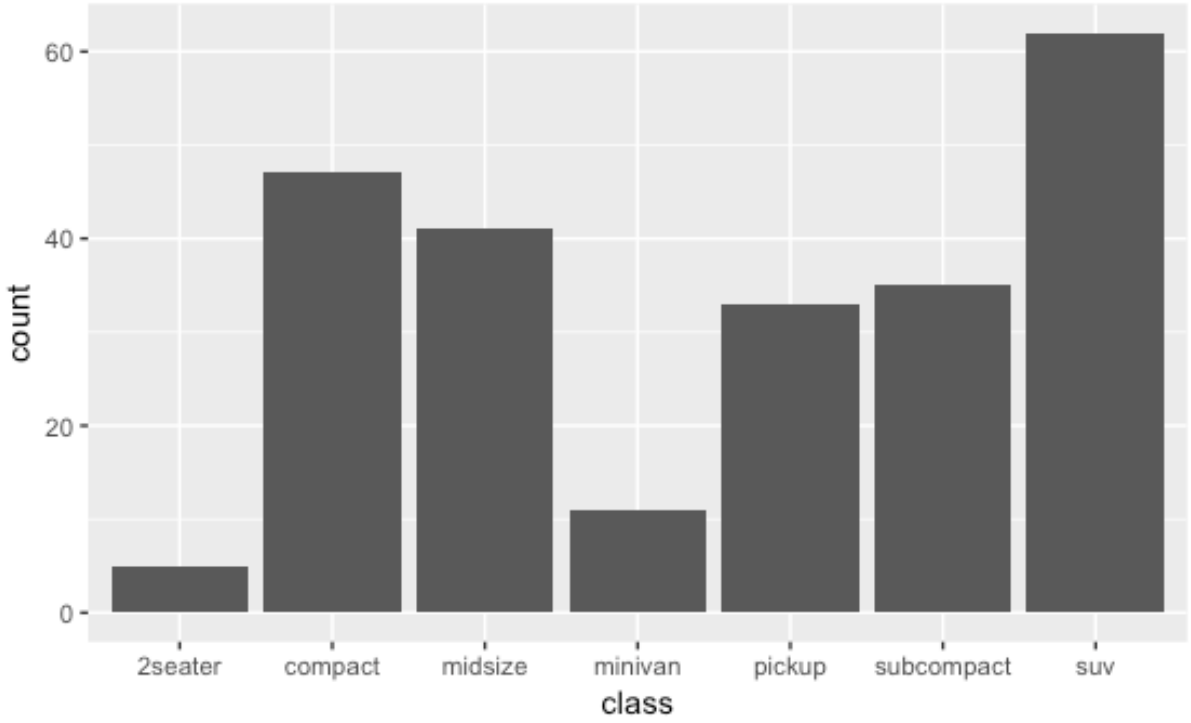
- Geometric objects are the actual marks we put on a plot. Examples include:
  - points (geom_point, for scatter plots, dot plots, etc)
  - lines (geom_line, for time series, trend lines, etc)
  - boxplot (geom_boxplot, for, well, boxplots!)

- A plot must have at least one geom; there is no upper limit.
- You can add a geom to a plot using the + operator
- You can get a list of available geometric objects using the code below:

```
help.search("geom_", package = "ggplot2")
```

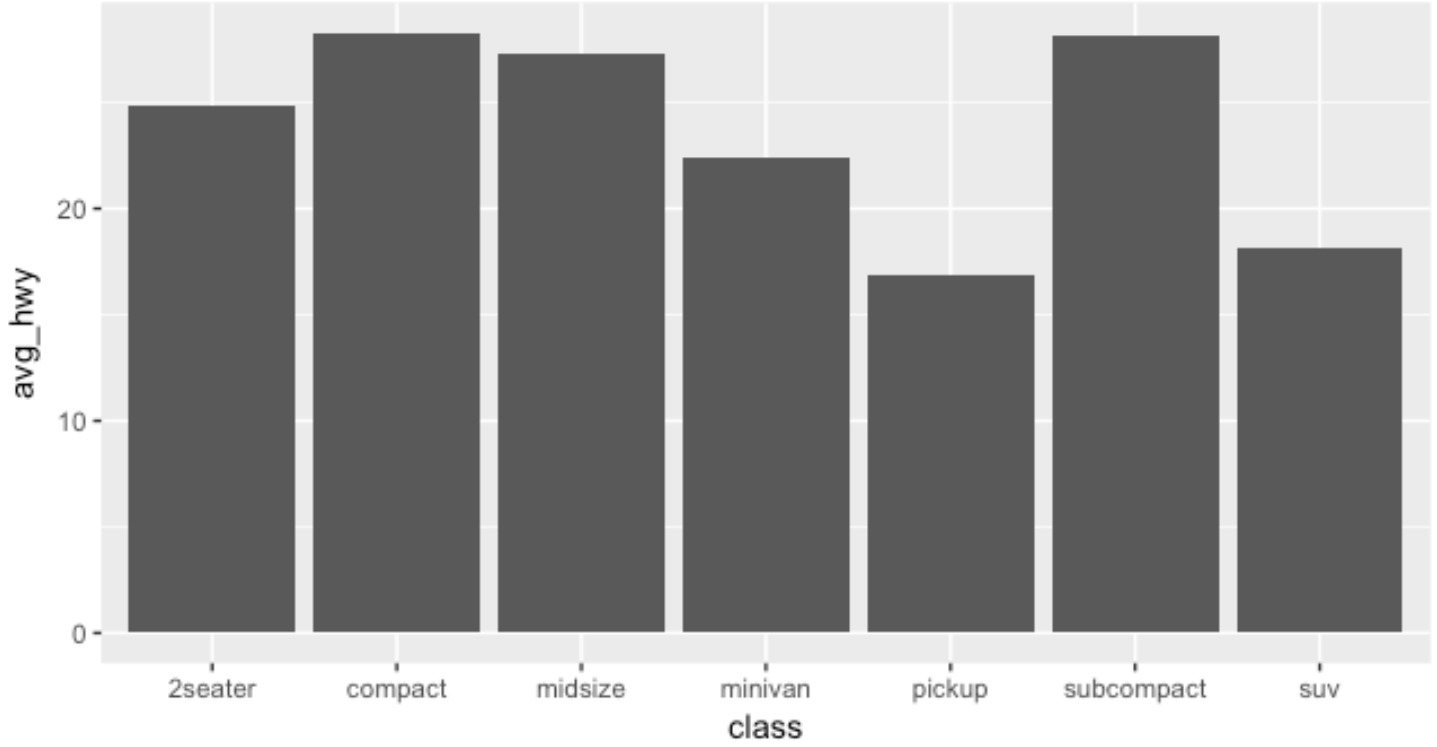
Bar chart (count)
geom_bar

```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = class))
```



Bar chart (value)
geom_col

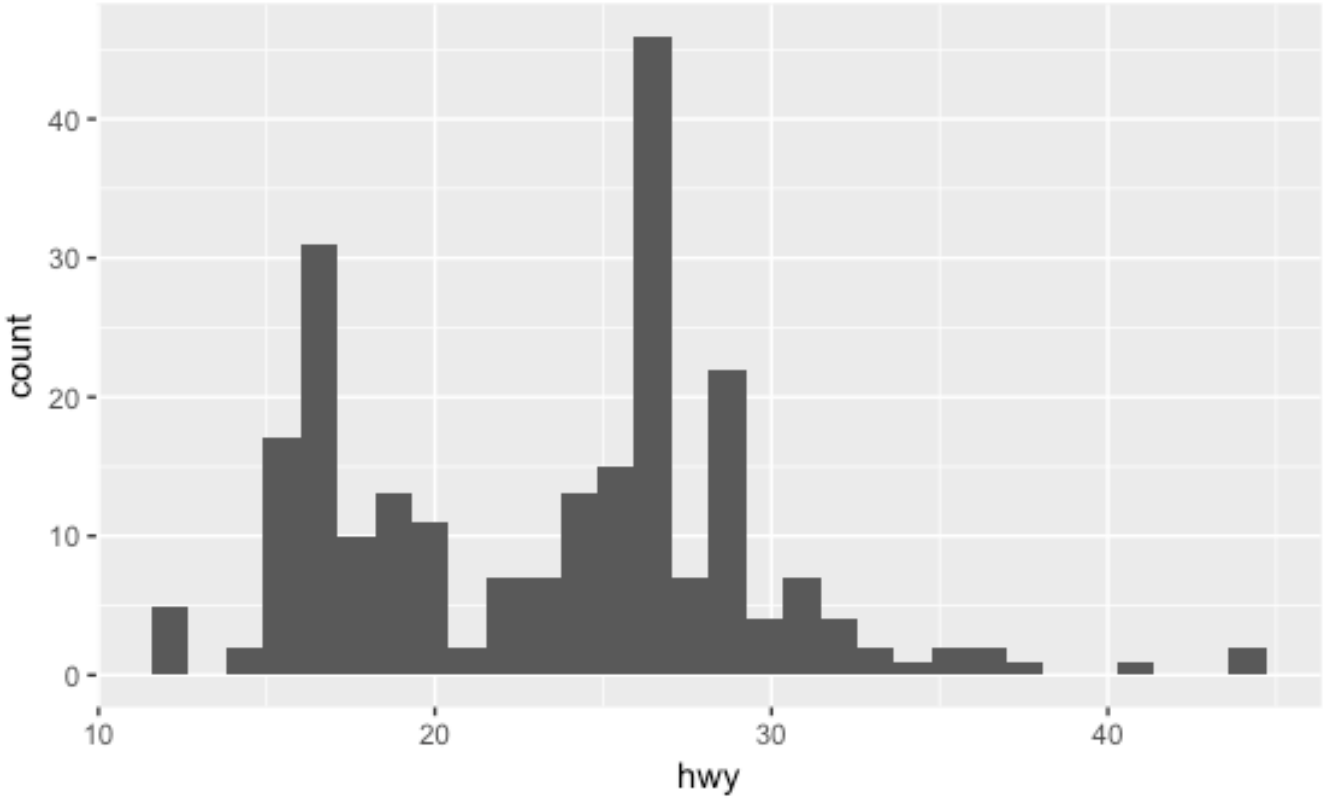
```
mpg %>%
  group_by(class) %>%
  summarise(avg_hwy = mean(hwy)) %>%
  ggplot() +
  geom_col(mapping = aes(x = class, y = avg_hwy))
```



BIG DATA EXPERIENCE

Histogram
geom_histogram

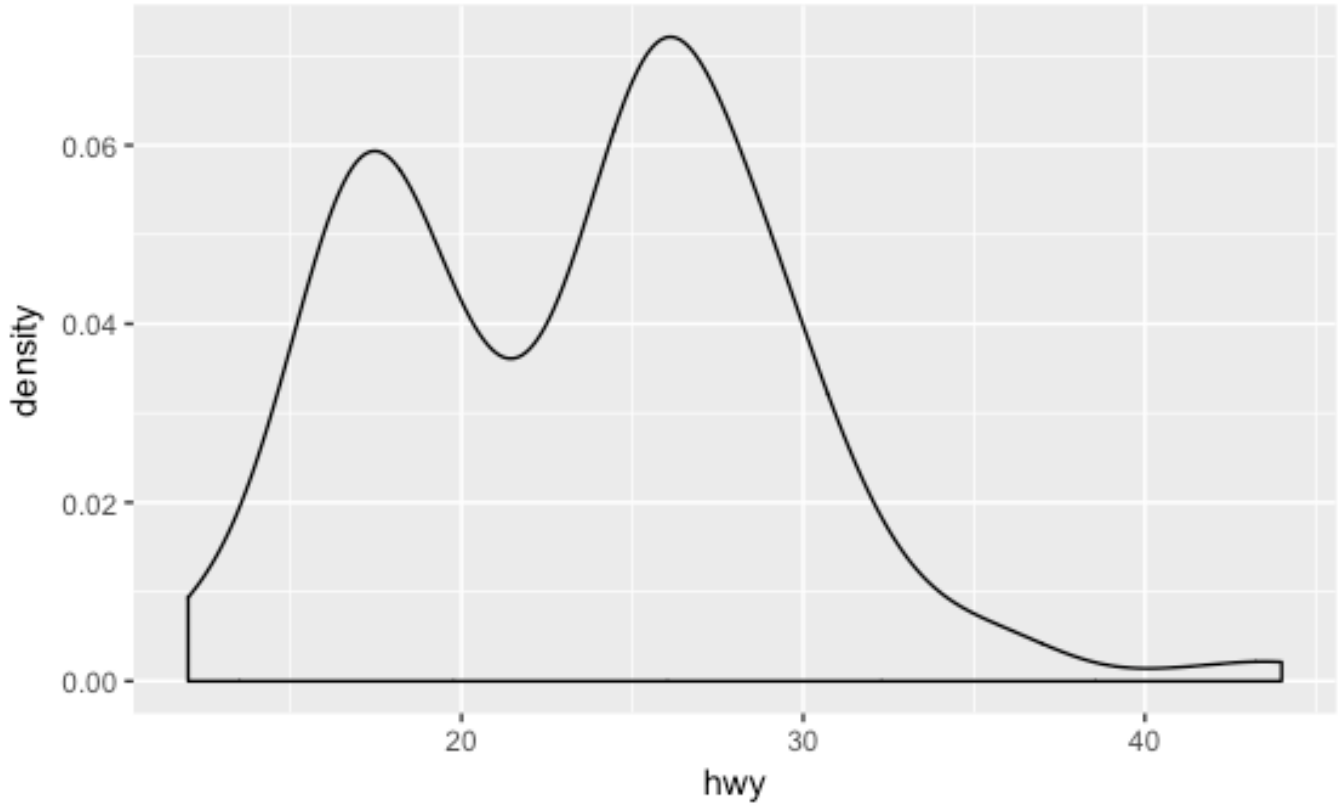
```
ggplot(mpg) +
  geom_histogram(aes(x = hwy))
```



KMUTT g-able

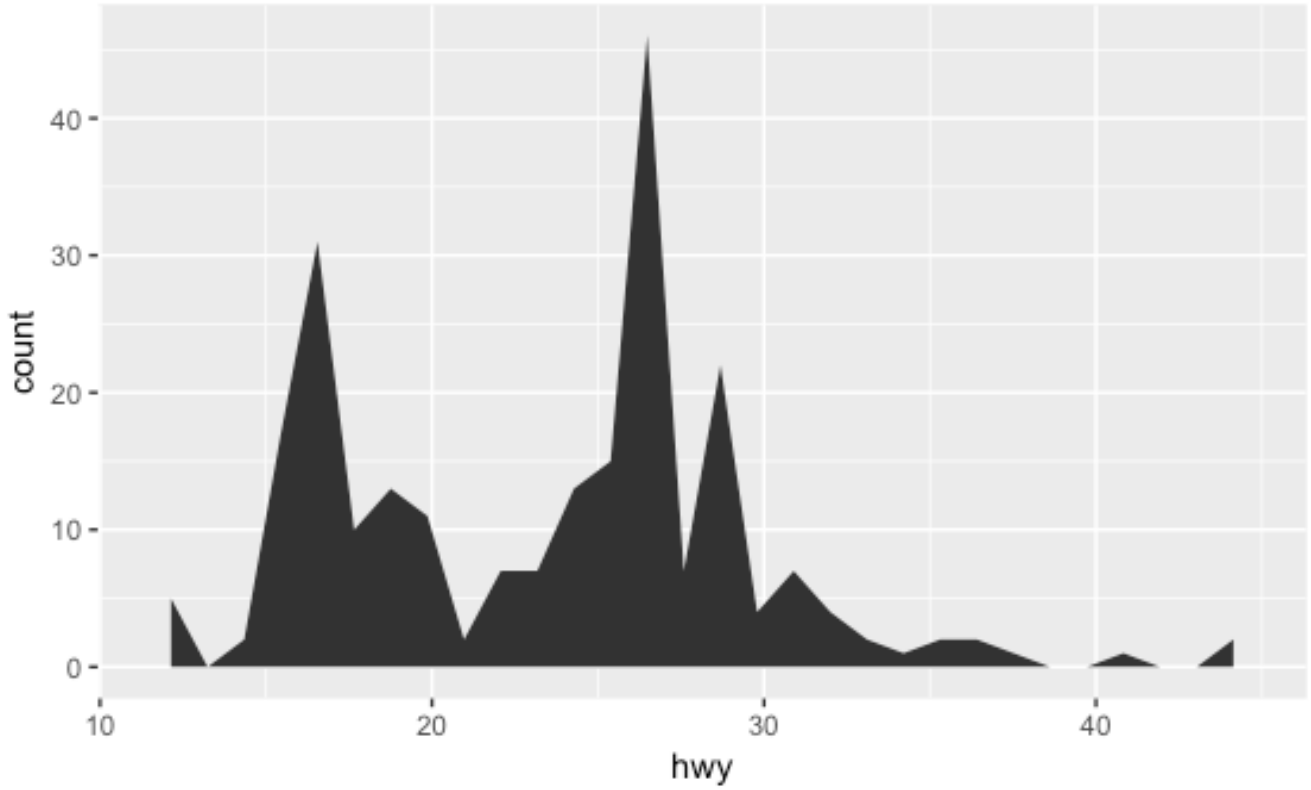
Density
geom_density

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



Density with geom_area
geom_area

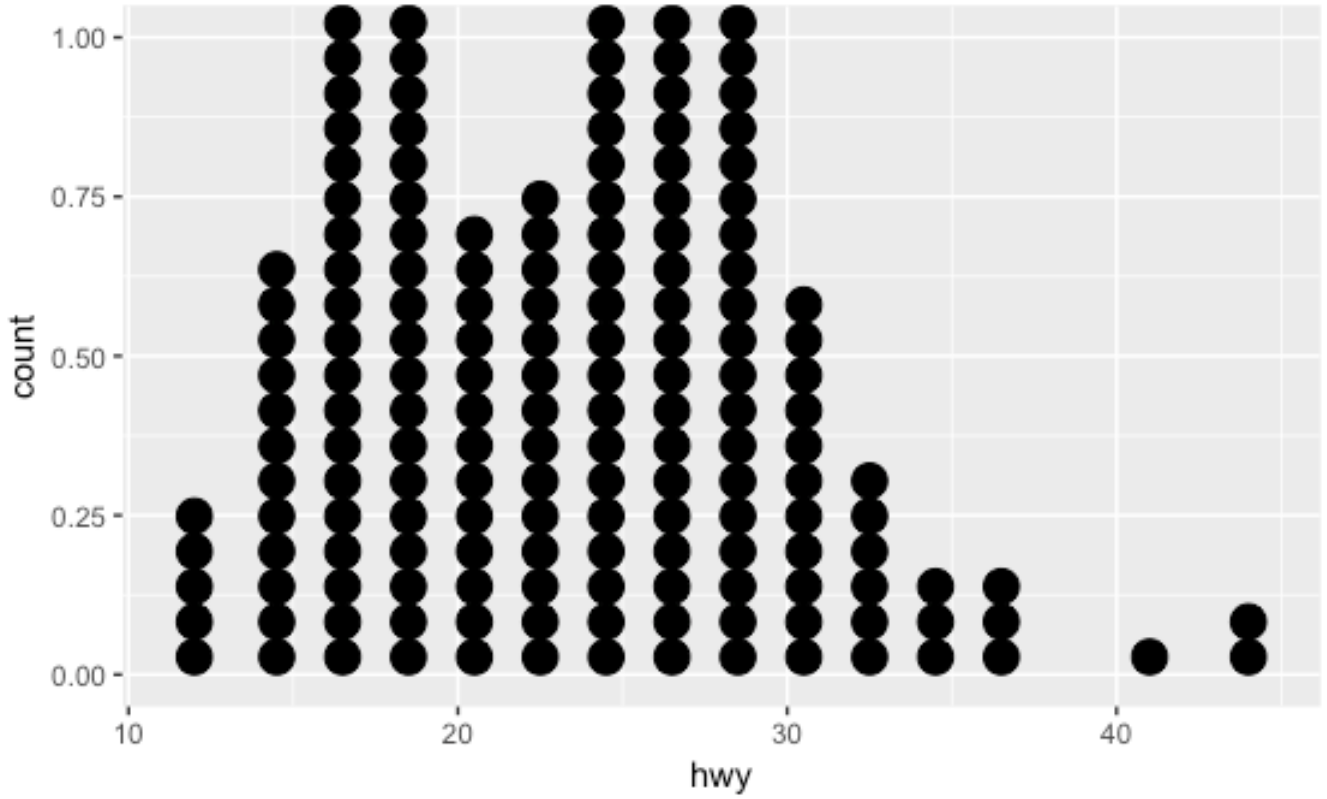
```
ggplot(mpg) +
  geom_area(aes(x = hwy), stat = 'bin')
```



BIG DATA EXPERIENCE

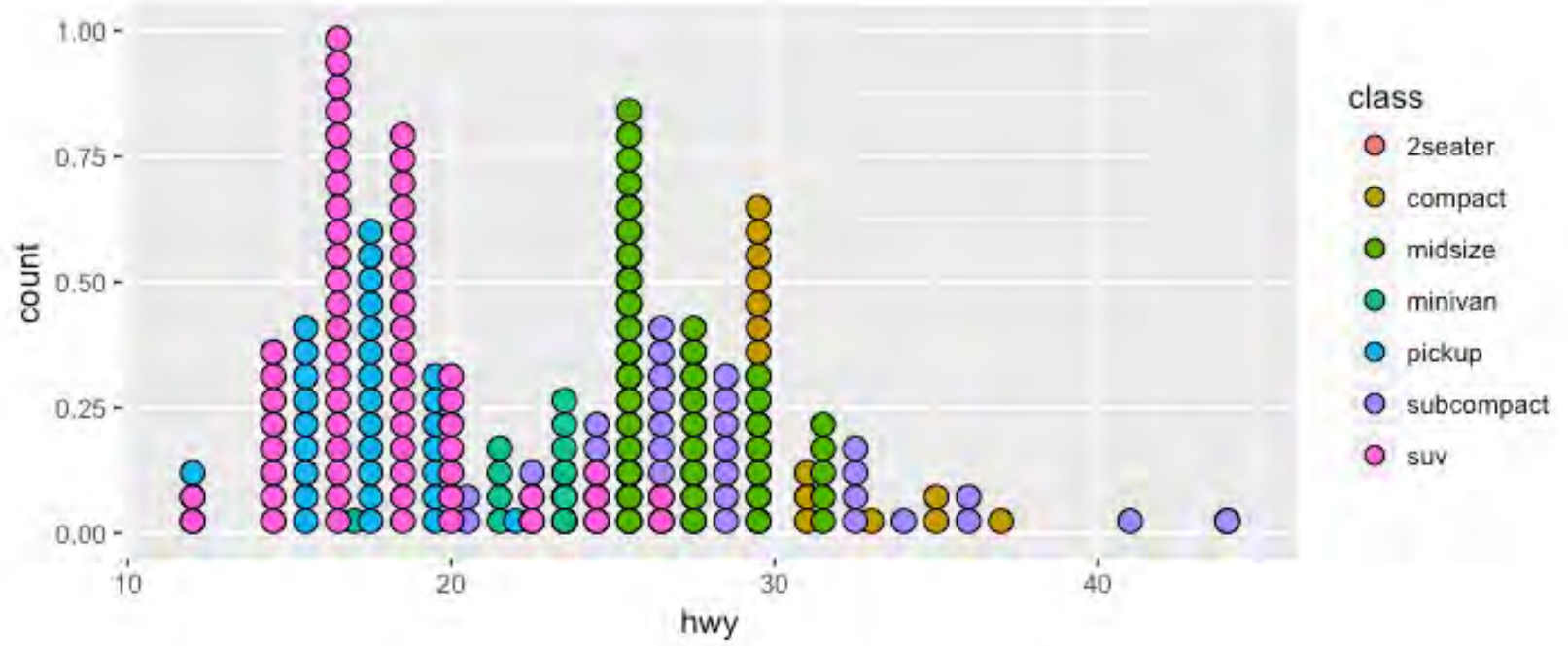
Dotplot
geom_dotplot

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



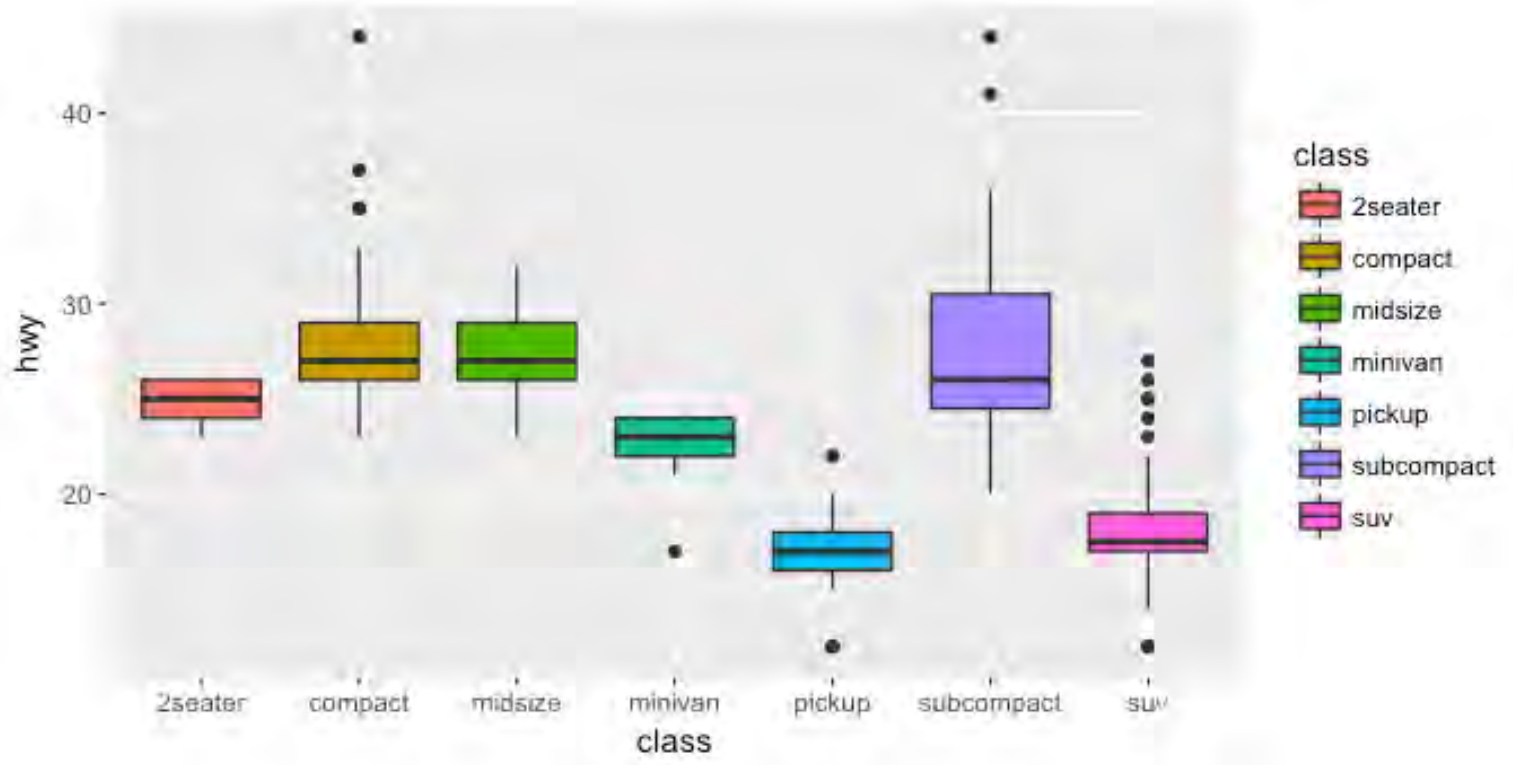
Dotplot with colour geom_dotplot

```
ggplot(mpg) +
  geom_dotplot(aes(x = hwy, fill = class),
               dotsize = 0.7)
```



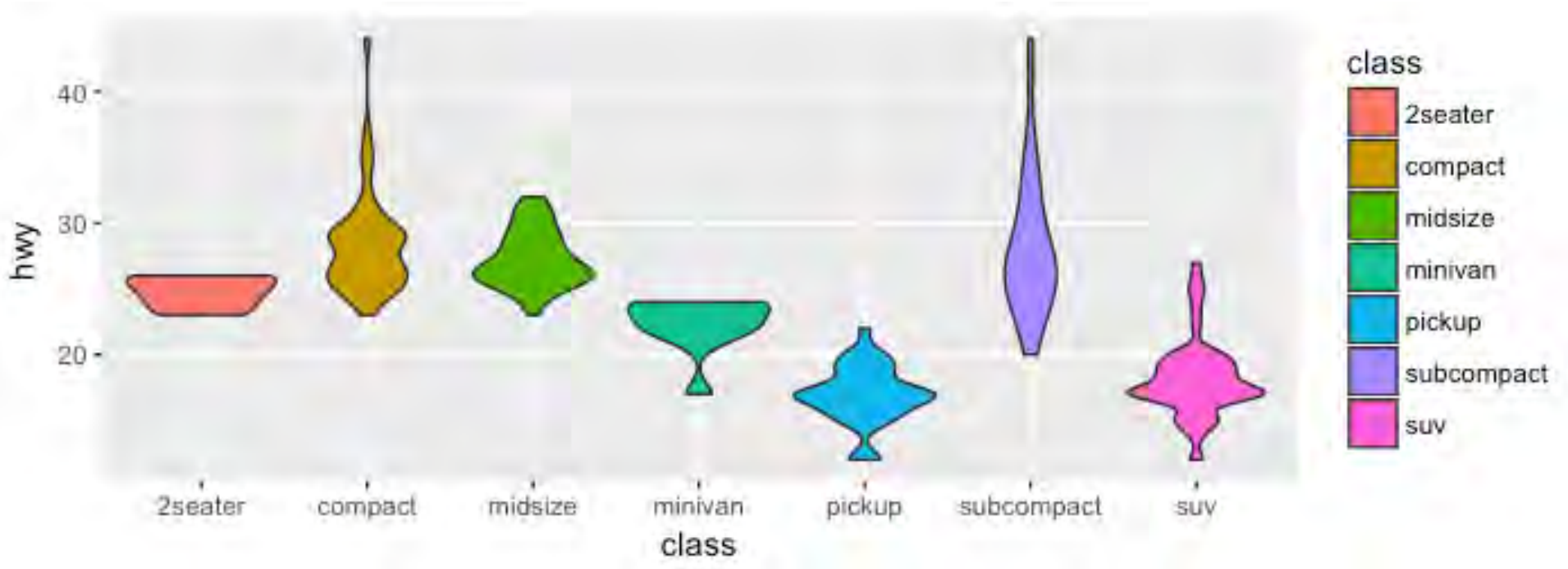
Boxplot
geom_boxplot

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



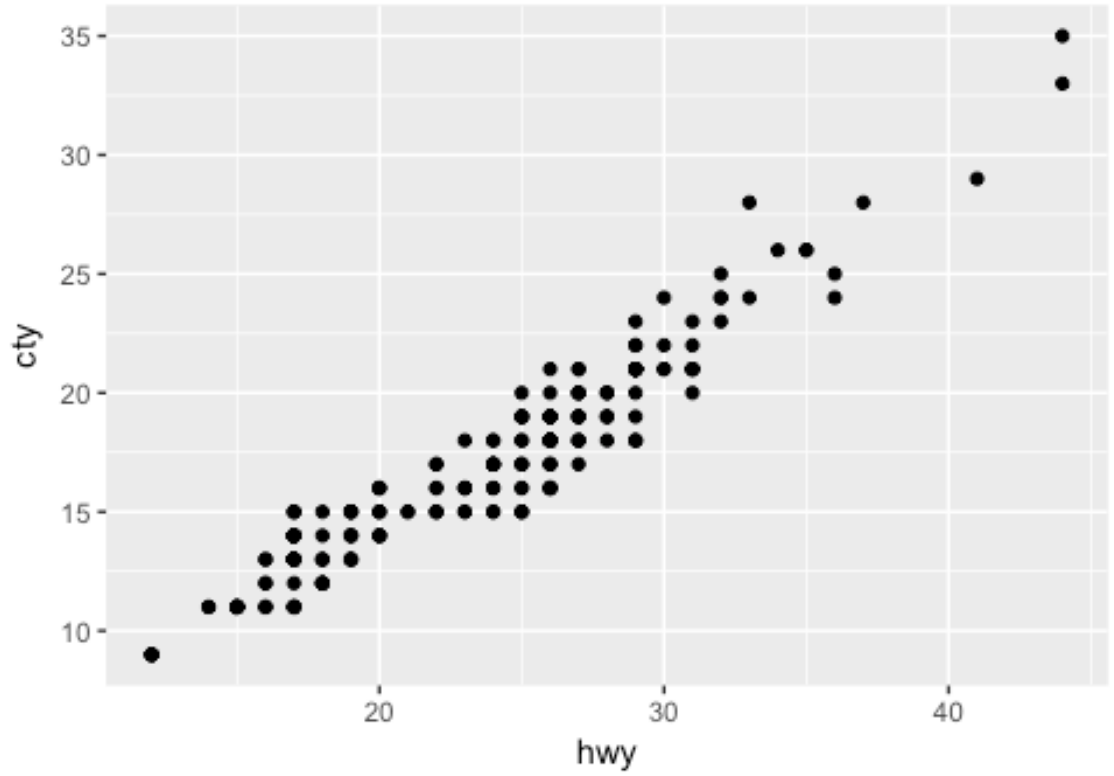
Violin plot (box + density) geom_violin

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



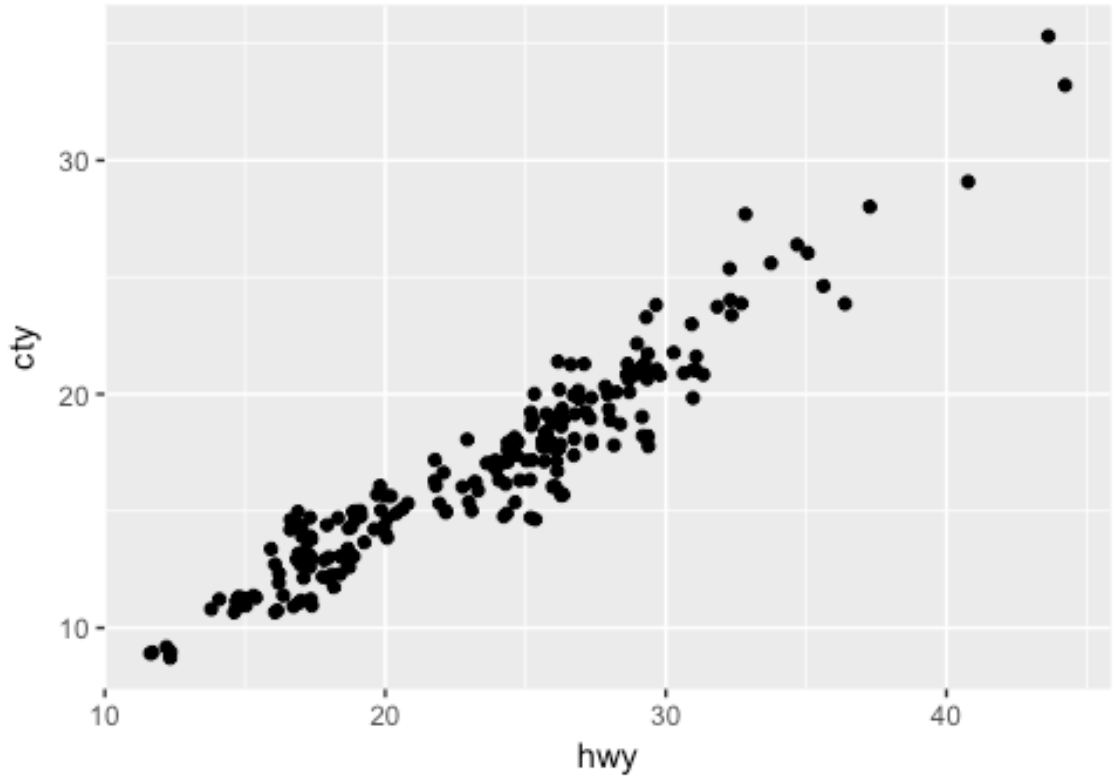
Scatter plot
geom_point

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



Jitter plot
geom_jitter

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



Jitter plot adds small variation to the points that are on top of other points

Scales: Controlling Aesthetic Mapping

- Aesthetic mapping (i.e., with aes()) only says that a variable should be mapped to an aesthetic.
- It doesn't say how that should happen.
- For example, when mapping a variable to shape with aes(shape = x) you don't say what shapes should be used.
- Similarly, aes(color = z) doesn't say what colors should be used.
- Describing what colors/shapes/sizes etc. to use is done by modifying the corresponding scale.

BIG DATA EXPERIENCE

Scales in ggplot2

- In ggplot2 scales include
  - position
  - color and fill
  - size
  - shape
  - line type
- Scales are modified with a series of functions using a scale_<aesthetic>_<type> naming scheme. Try typing scale_<tab> to see a list of scale modification functions.

Common Scale Arguments

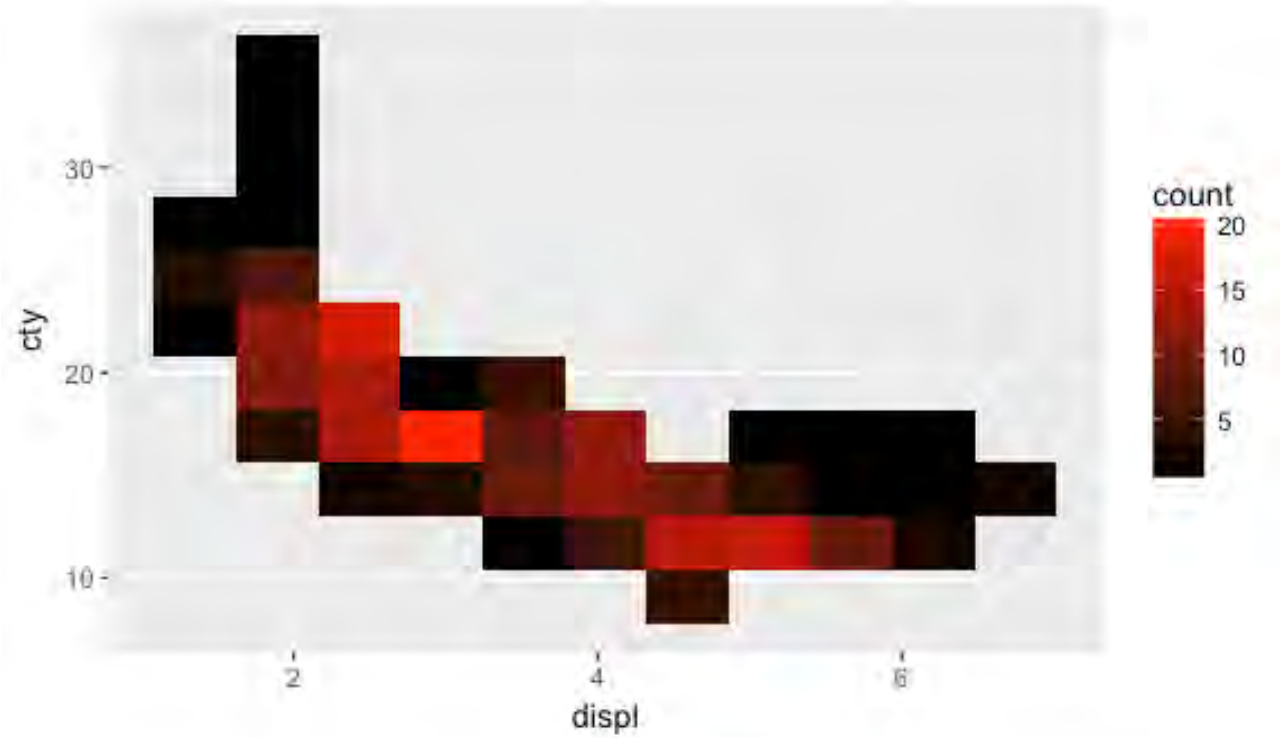
The following arguments are common to most scales in ggplot2:

- **name**
  - the first argument gives the axis or legend title
- **limits**
  - the minimum and maximum of the scale
- **breaks**
  - the points along the scale where labels should appear
- **labels**
  - the labels that appear at each break

Specific scale functions may have additional arguments; for example, the scale_color_continuous function has arguments low and high for setting the colors at the low and high end of the scale.

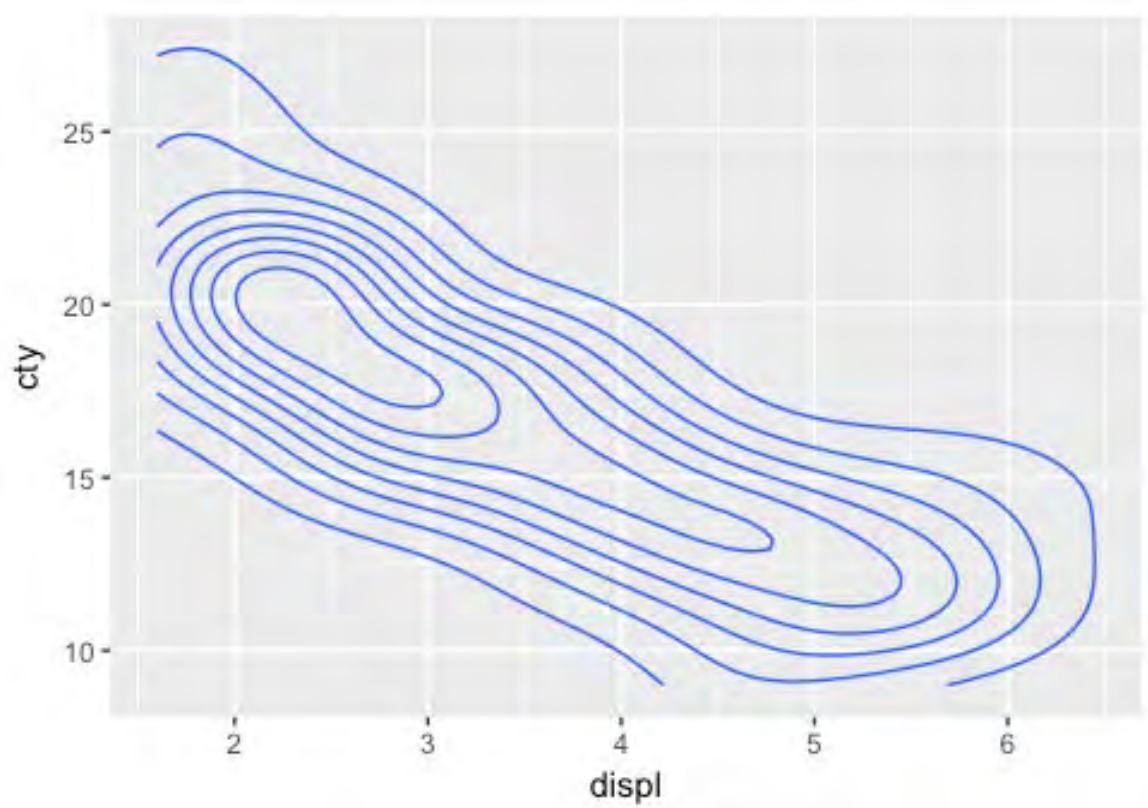
2D distribution bin plot geom_bin2d

```
ggplot(mpg) +
  geom_bin2d(aes(x = displ, y = cty), bins = 10) +
  scale_fill_gradient(low = "#000000", high = "#FF0000")
```



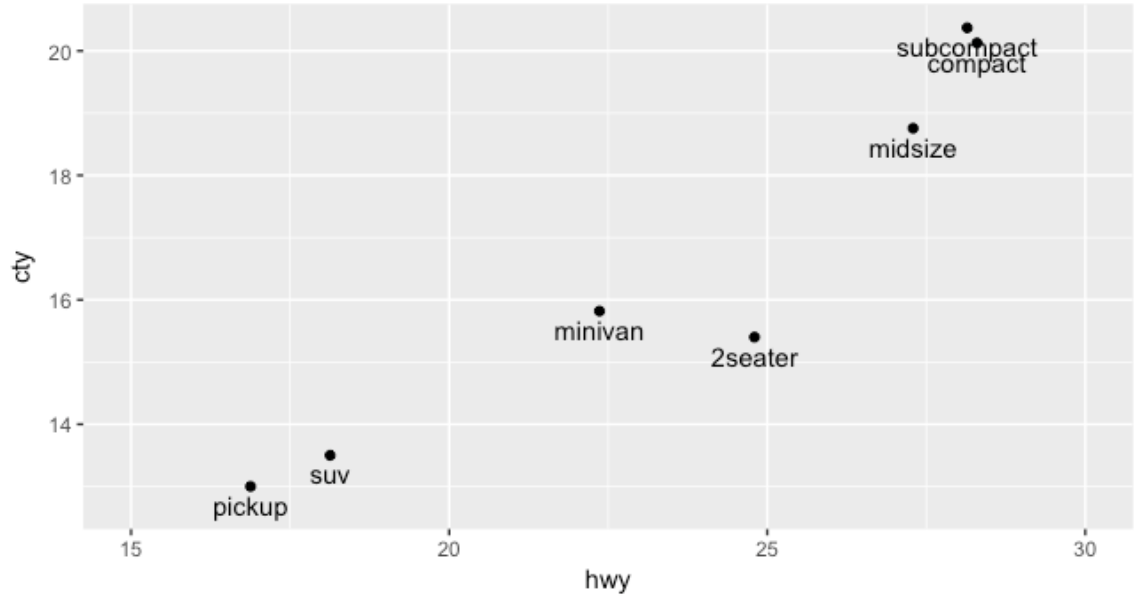
2D density plot
geom_density2d

```
ggplot(mpg) +
  geom_density2d(aes(x = displ, y = cty))
```



Aggregated scatter plot with label geom_point + geom_text

```
mpg %>%
  group_by(class) %>%
  summarise(cty = mean(cty), hwy = mean(hwy)) %>%
  ggplot(aes(x = hwy, y = cty)) + geom_point() +
  geom_text(aes(label = class),nudge_y = -0.3) +
  xlim(15,30)
```

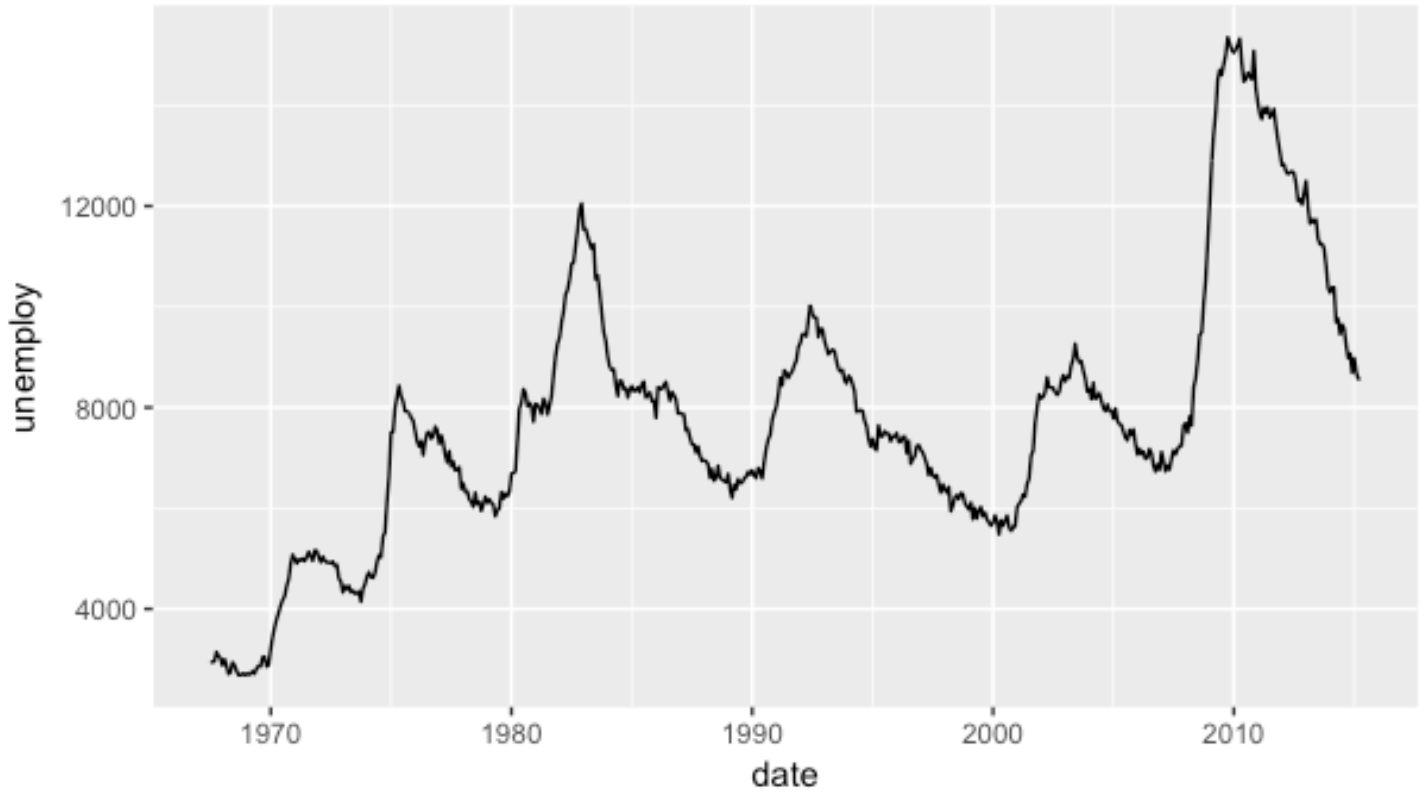


Economic data economics

```
> economics
# A tibble: 574 x 6
   date         pce    pop psavert uempmed unemploy
   <date>     <dbl>  <int>   <dbl>   <dbl>    <int>
 1 1967-07-01  507. 198712    12.5     4.5     2944
 2 1967-08-01  510. 198911    12.5     4.7     2945
 3 1967-09-01  516. 199113    11.7     4.6     2958
 4 1967-10-01  513. 199311    12.5     4.9     3143
 5 1967-11-01  518. 199498    12.5     4.7     3066
 6 1967-12-01  526. 199657    12.1     4.8     3018
 7 1968-01-01  532. 199808    11.7     5.1     2878
 8 1968-02-01  534. 199920    12.2     4.5     3001
 9 1968-03-01  545. 200056    11.6     4.1     2877
10 1968-04-01  545. 200208    12.2     4.6     2709
# ... with 564 more rows
```

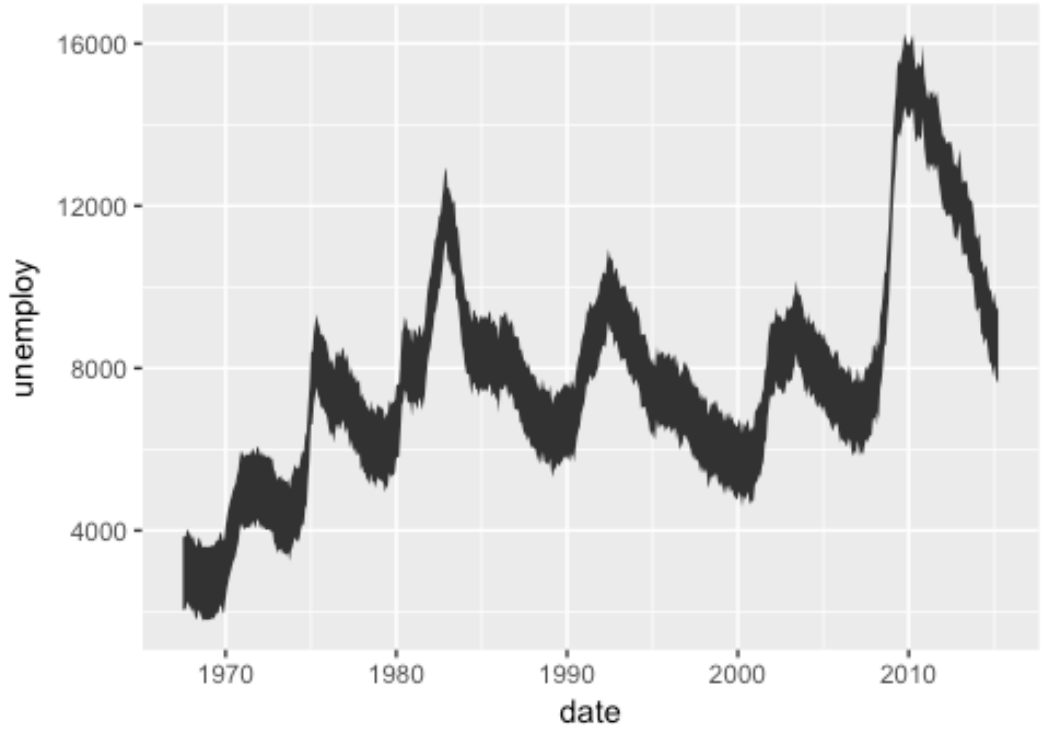
Line plot
geom_line

```
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_line()
```



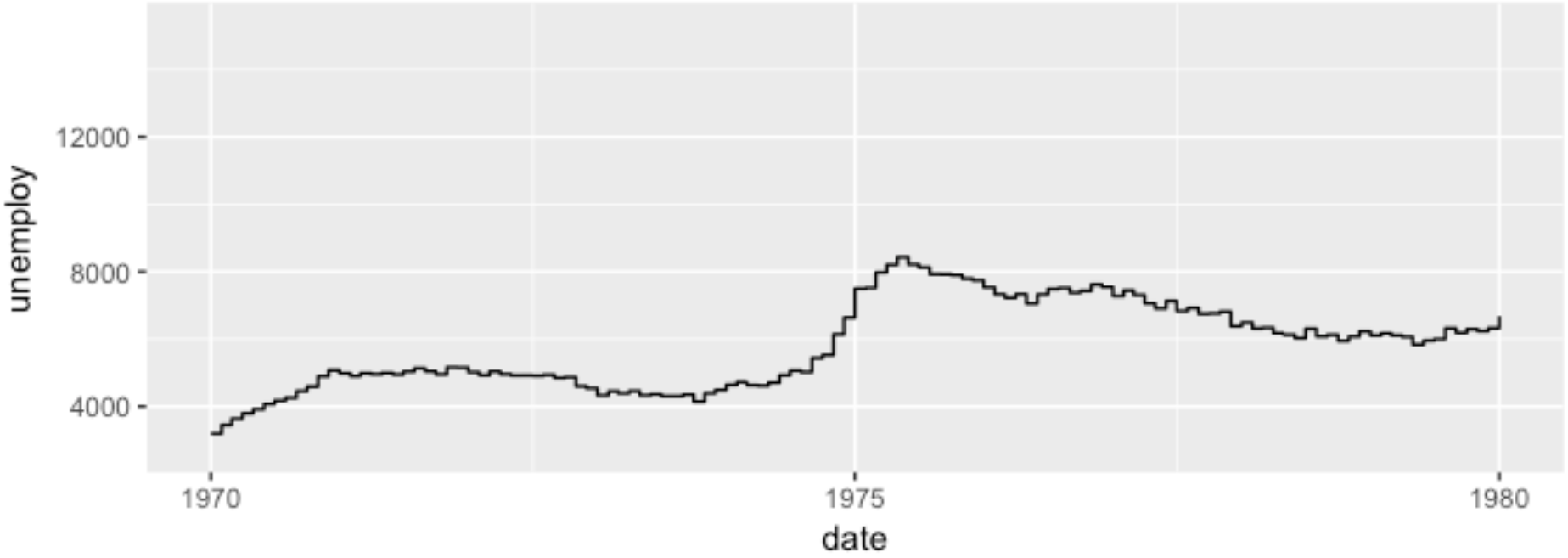
Ribbon plot
geom_ribbon

```
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_ribbon(aes(ymin = unemploy - 900,
                  ymax = unemploy + 900))
```



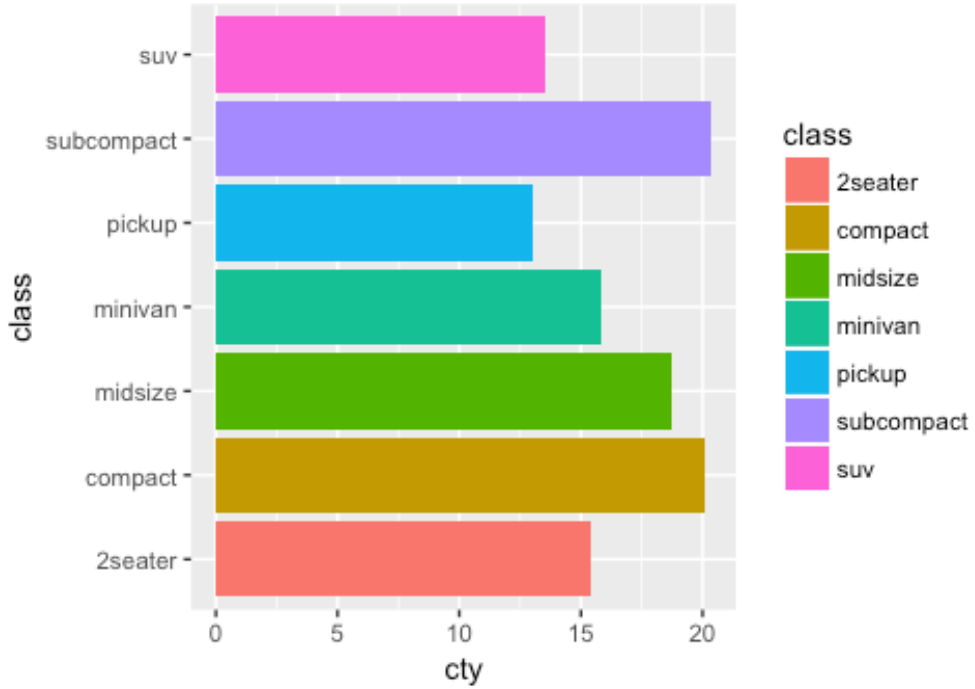
Step plot
geom_step

```
ggplot(economics, aes(x = date, y = unemploy)) +
  geom_step(direction = 'hv') +
  xlim(as.Date(c('1/1/1970', '1/1/1980'),
               format="%d/%m/%Y") )
```



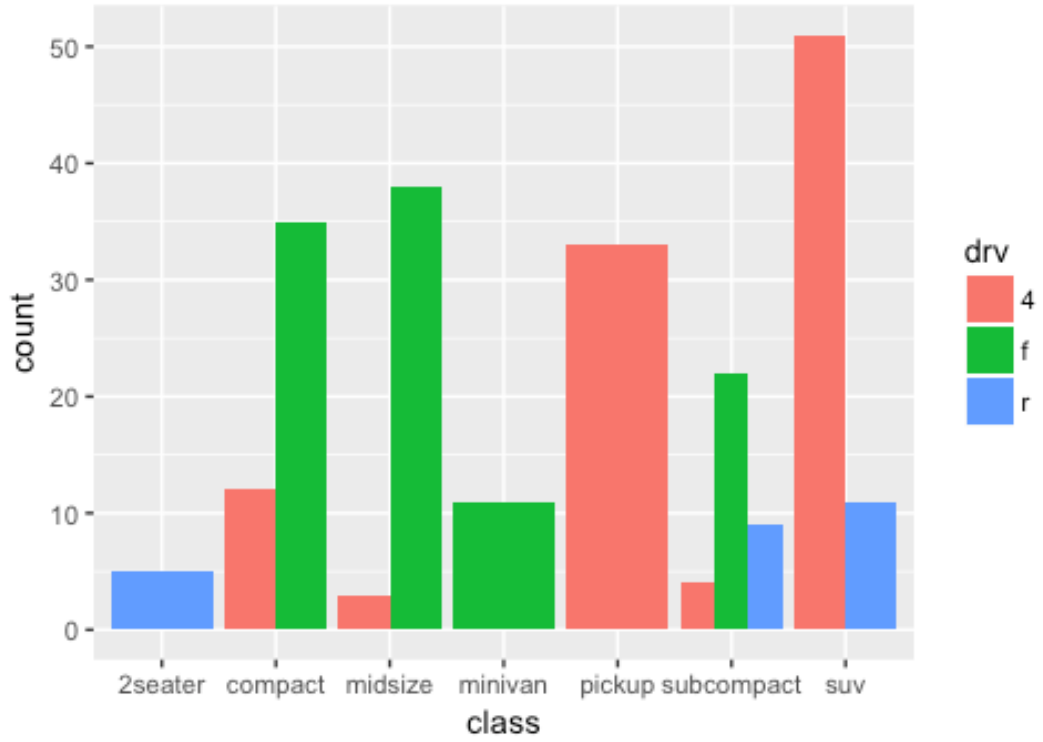
Coordinate
coord_flip

```
ggplot(mpg) +
  geom_bar(aes(x = class, y = cty, fill = class),
           stat = 'summary') +
  coord_flip()
```



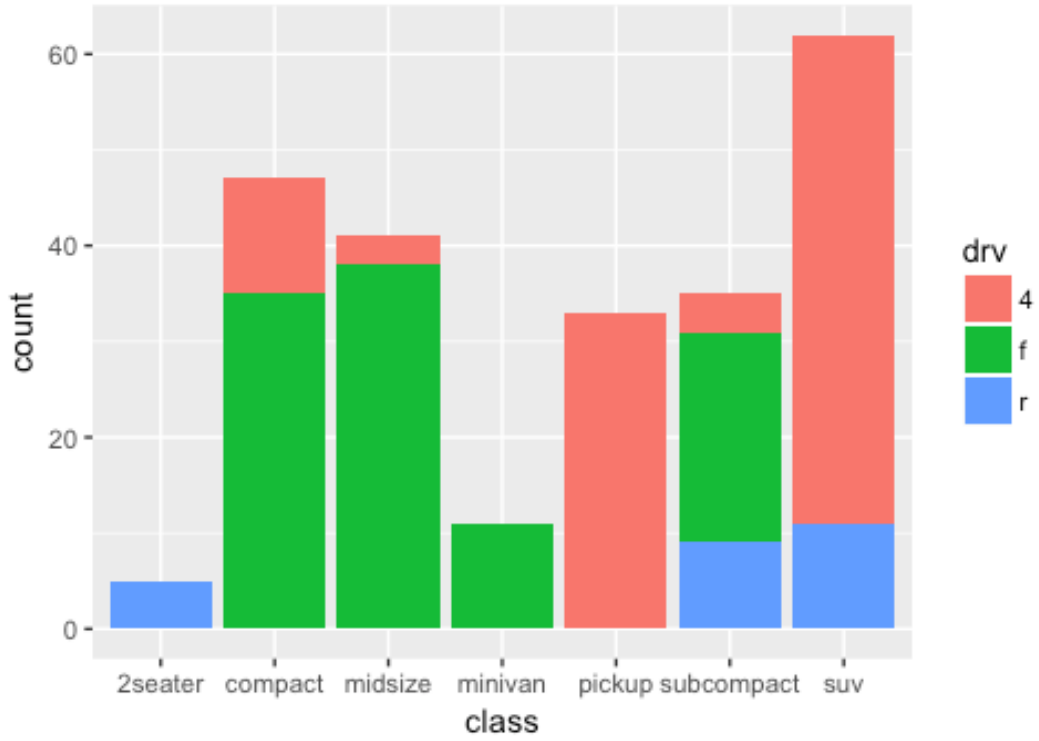
Position: dodge

```
ggplot(mpg) +
  geom_bar(aes(x = class, fill = drv),
           position = 'dodge')
```



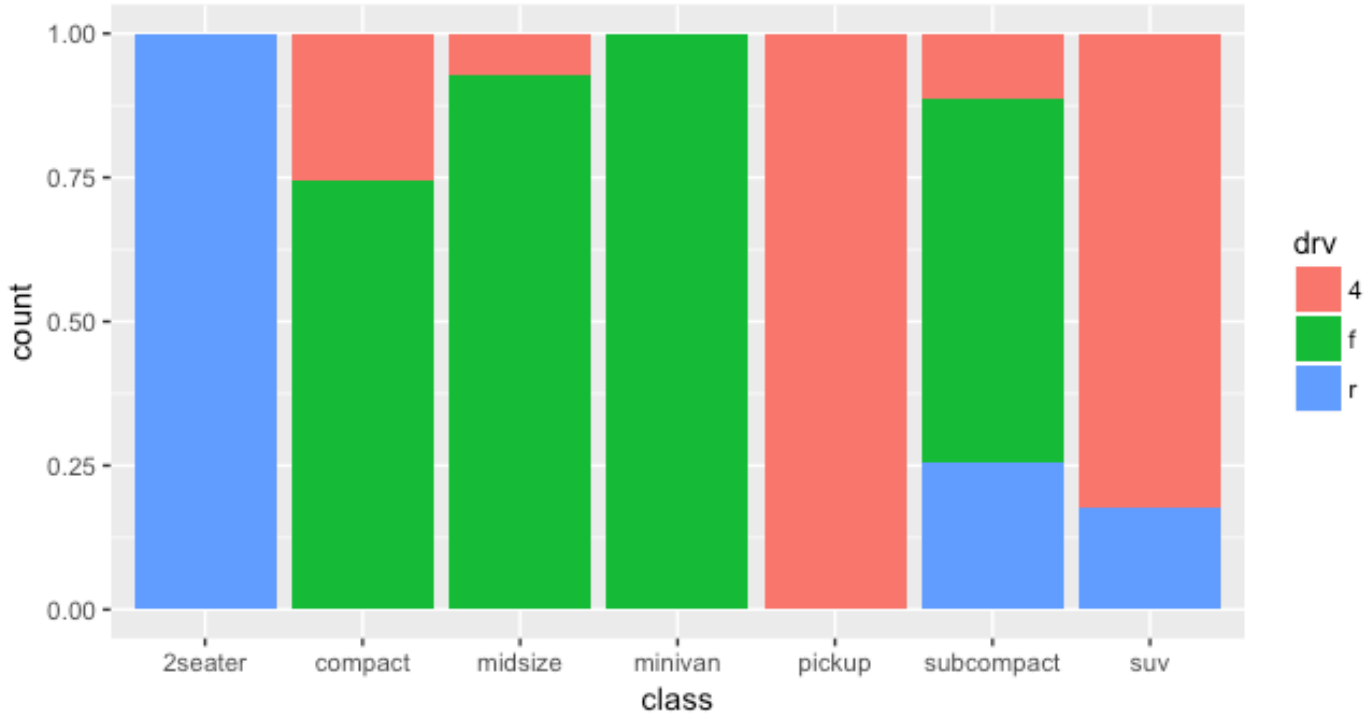
Position : stack

```
ggplot(mpg) +
  geom_bar(aes(x = class, fill = drv),
           position = 'stack')
```



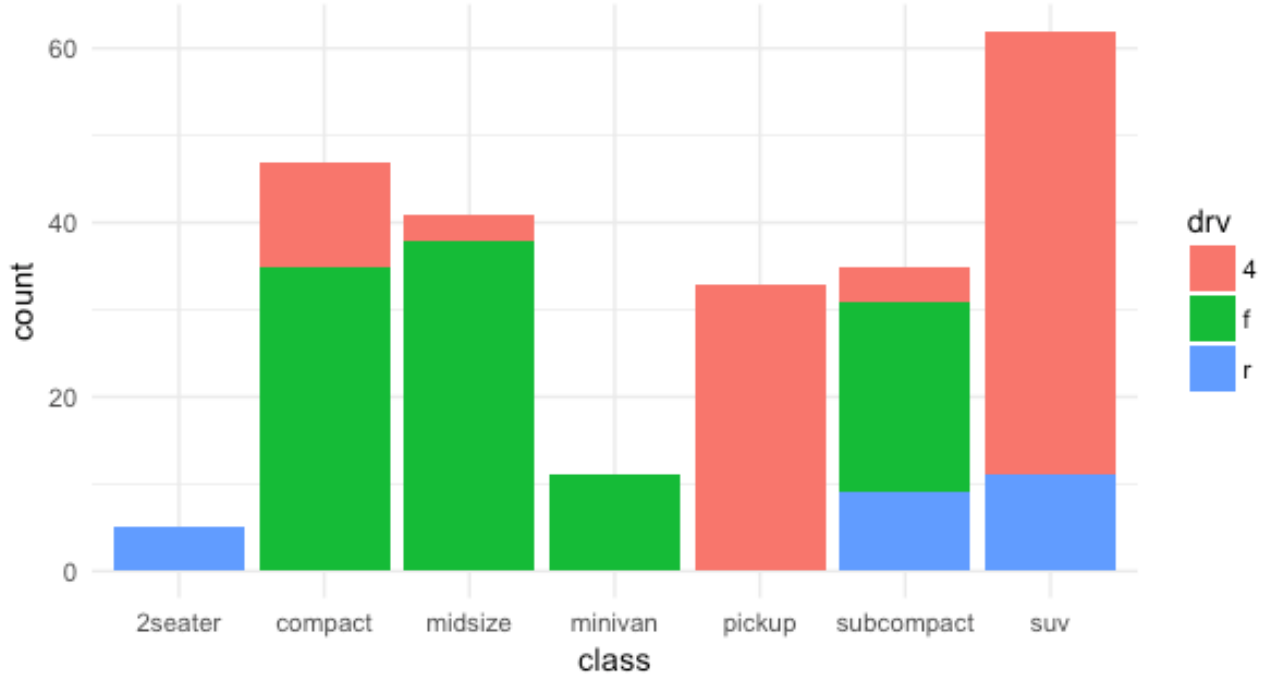
Position : fill

```
ggplot(mpg) +
  geom_bar(aes(x = class, fill = drv),
           position = 'fill')
```



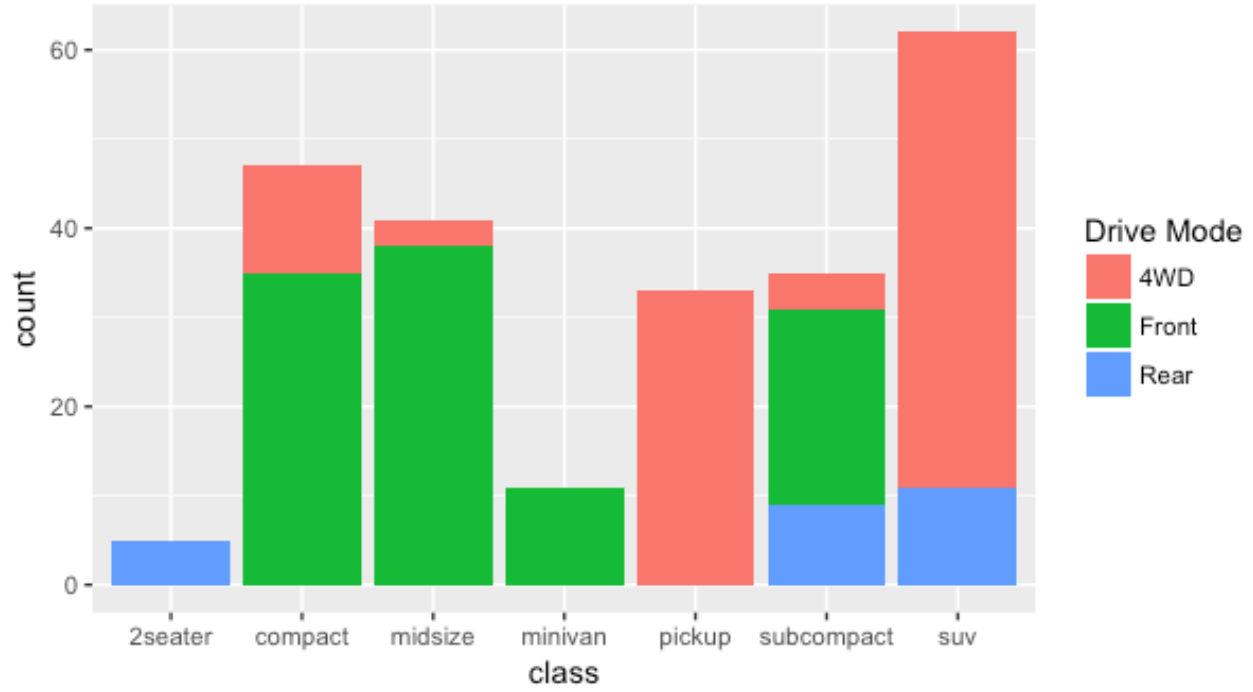
Theme : minimal (example)

```
ggplot(mpg) +
  geom_bar(aes(x = class, fill = drv)) +
  theme_minimal()
```



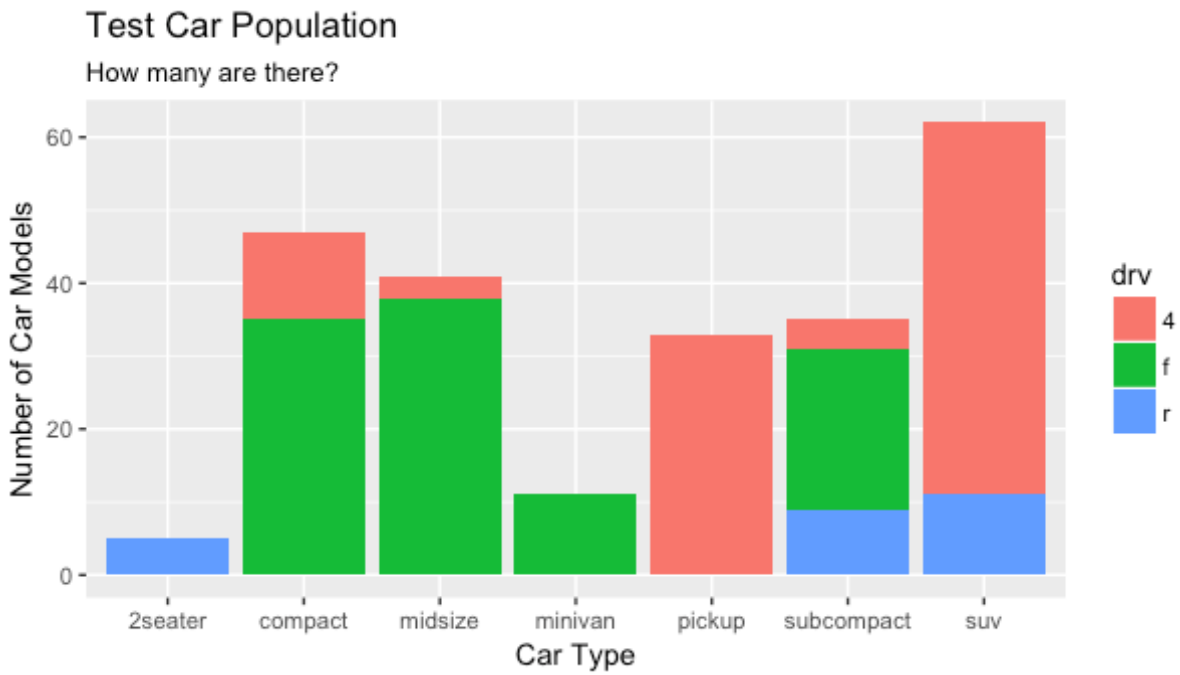
Legend

```
ggplot(mpg) +
  geom_bar(aes(x = class, fill = drv)) +
  scale_fill_discrete(name = "Drive Mode",
                      labels = c('4WD','Front','Rear'))
```



Labels

```
ggplot(mpg) +
  geom_bar(aes(x = class, fill = drv)) +
  ggtitle("Test Car Population",subtitle = "How many are there?") +
  xlab("Car Type") + ylab("Number of Car Models")
```



Lab

- Given the data ‘superstore.csv’ http://fastdata.in.th/data-model-2021/superstore.csv
- Plot 3 graphs from the data
- Explain (1) the meaning of each graph and (2) what question required this graph as an answer

BIG DATA
EXPERIENCE

Thank you

Question?

KMUTT g-able