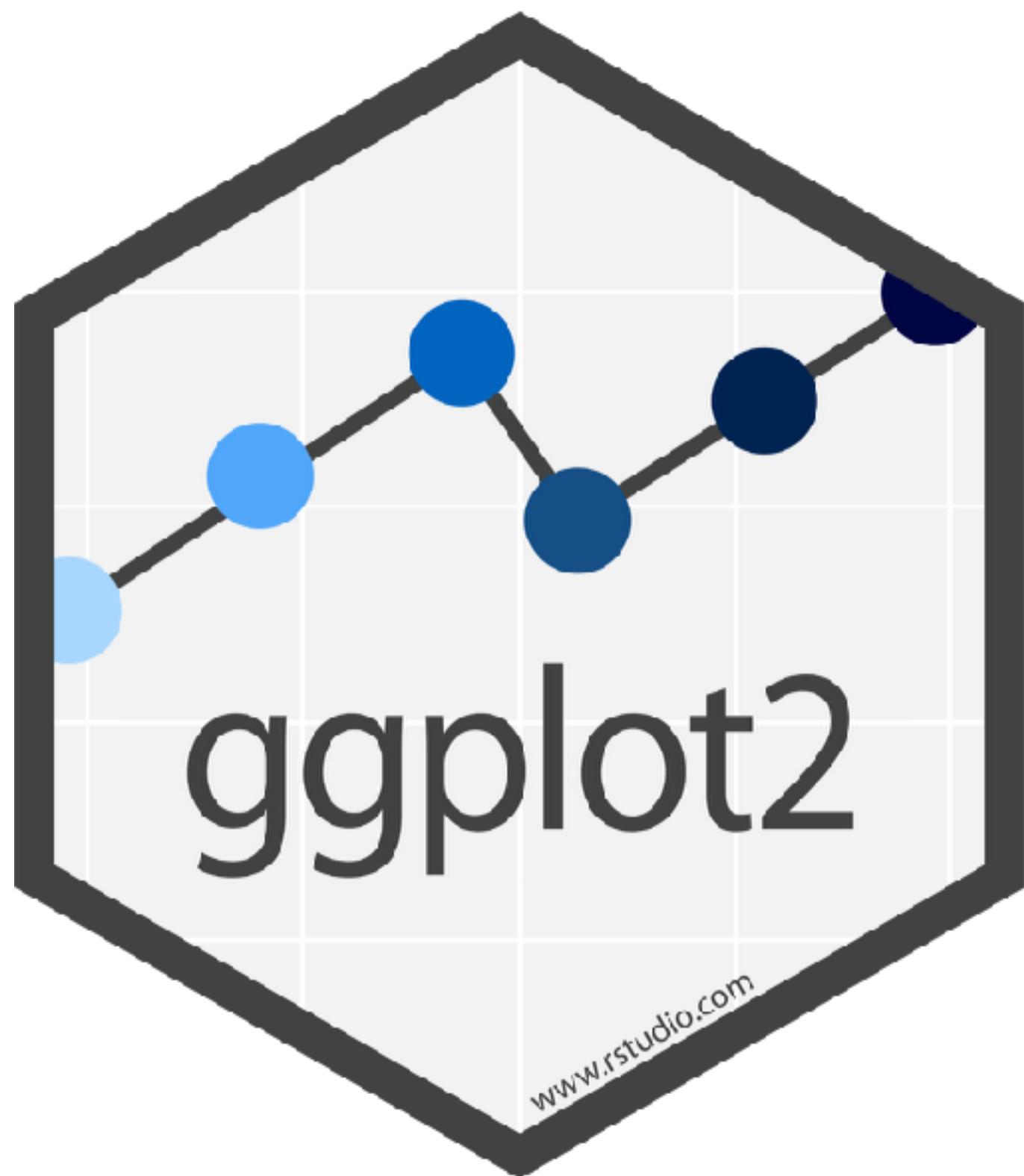
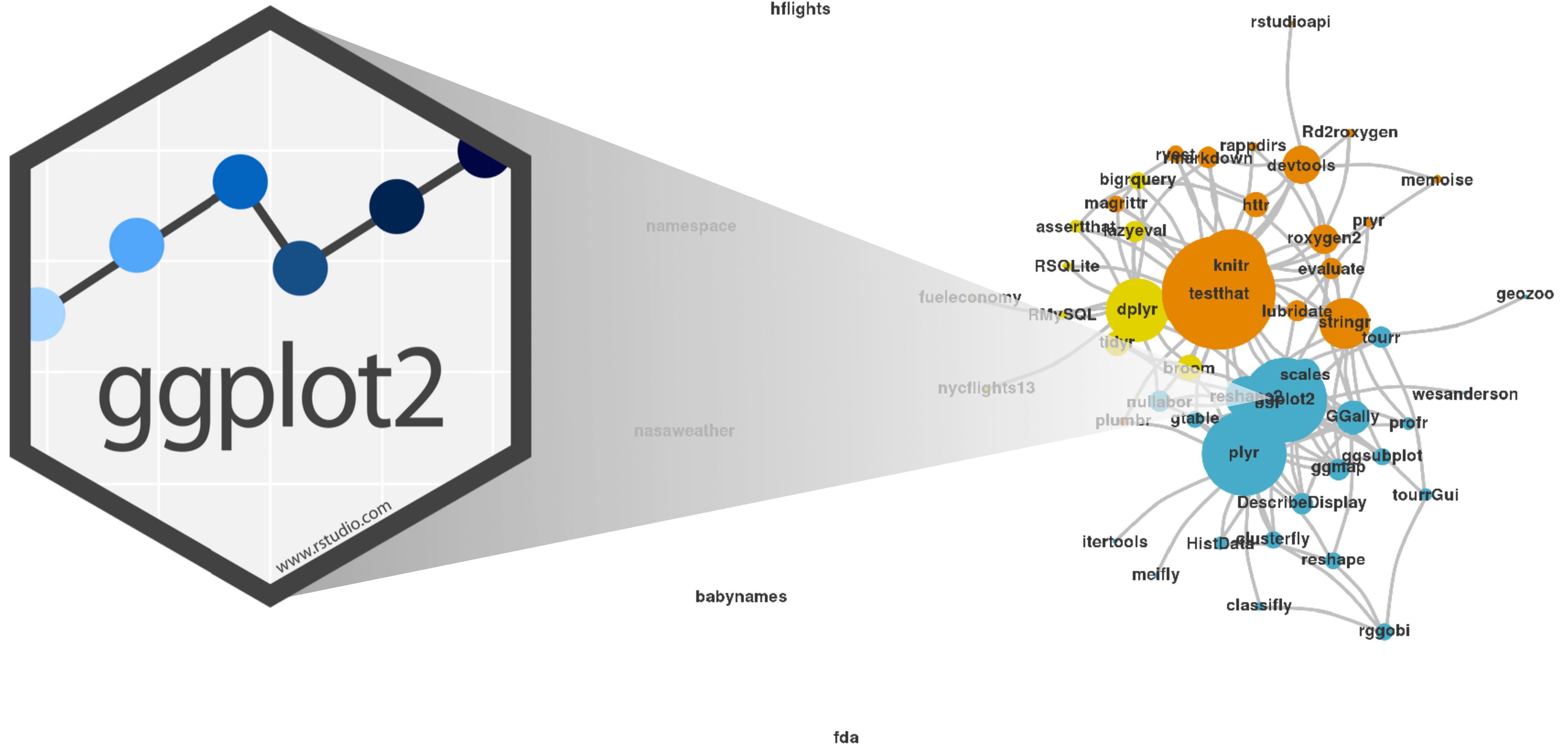


Visualize Data with



"The simple graph has brought more information to the data analyst's mind than any other device. "

- John Tukey



Setup

The setup chunk is always run once before anything else

The screenshot shows an RStudio interface with an R Markdown file titled "01-Visualize-Data.Rmd". The code editor displays the following content:

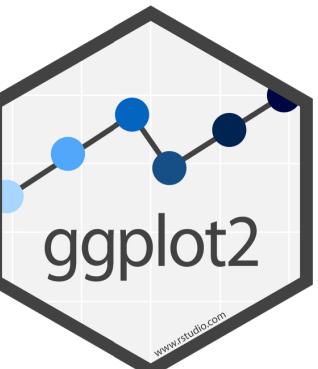
```
1 ---  
2 title: "Data Visualization"  
3 output: html_notebook  
4 ---  
5  
6 ```{r setup}  
7 library(tidyverse)  
8 ...  
9  
10```{r}  
11 mpg  
12 ...  
13  
14  
15## Your Turn 1  
16  
17 Run the code on the slide to make a graph. Pay strict attention to  
spelling, capitalization, and parentheses!  
18  
19```{r}  
20 ...  
21 ...  
22  
23## Your Turn 2  
24  
25 Add `color`, `size`, `alpha`, and `shape` aesthetics to your graph.  
Experiment.  
26  
27```{r}  
28 ggplot(data = mpg) +  
29   geom_point(mapping = aes(x = displ, y = hwy))  
30 ...  
31  
32## Your Turn 3  
33  
34 Replace this scatterplot with one that draws boxplots. Use the cheatsheet.  
Try your best guess.  
35  
36```{r}  
37 ggplot(mpg) + geom_point(aes(class, hwy))  
38 ...
```

A callout box highlights the code block starting at line 6, which includes an optional label "setup" followed by the command "library(tidyverse)". A large callout bubble on the right side contains the text "(optional) label for chunk".

mpg

Fuel economy data for 38 models of car.

mpg



Quiz

Confer with your group.

What relationship do you expect to see between engine size (displ) and mileage (hwy)?

No peeking ahead!



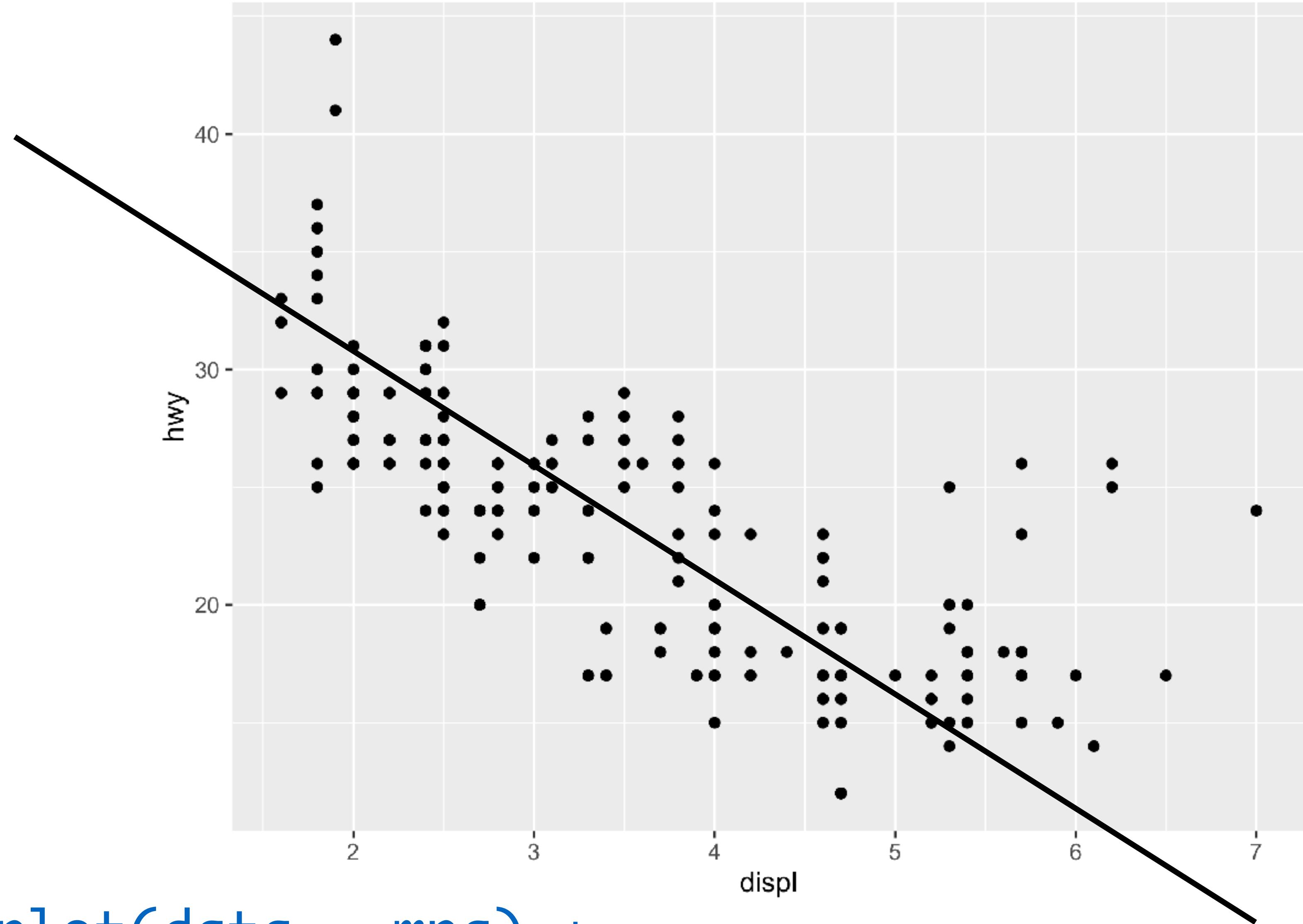
Your Turn 1

Run this code in your notebook to make a graph.

Pay strict attention to spelling, capitalization, and parentheses!

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```





```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

1. "Initialize" a plot with `ggplot()`
2. Add layers with `geom_` functions

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

Pro tip: Always put the + at the end
of a line, Never at the start

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

data

+ before new line

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

type of layer

aes()

x variable

y variable

A template

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

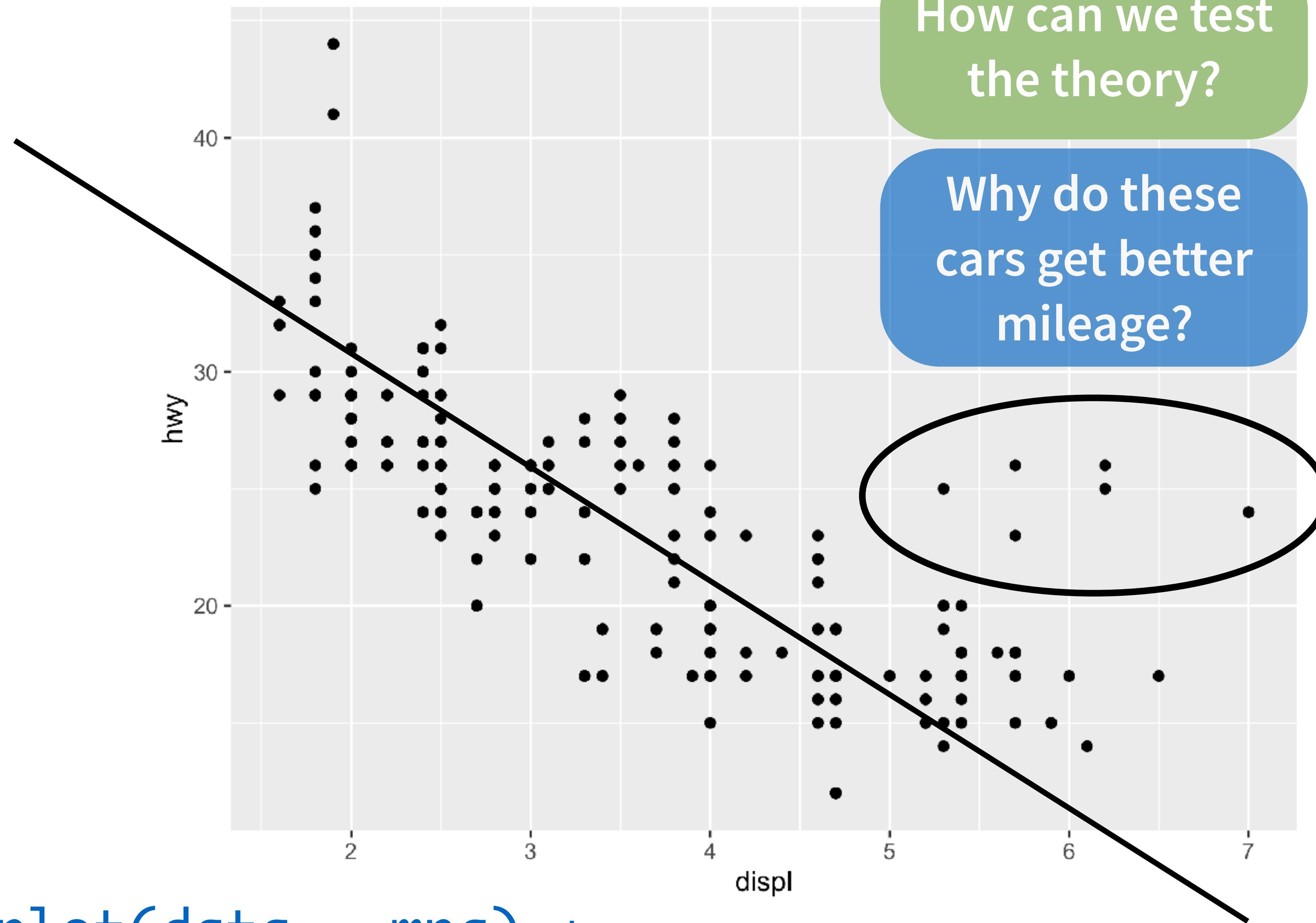
```
geom_point(mapping = aes(x = displ, y = hwy))
```

A template

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Mappings



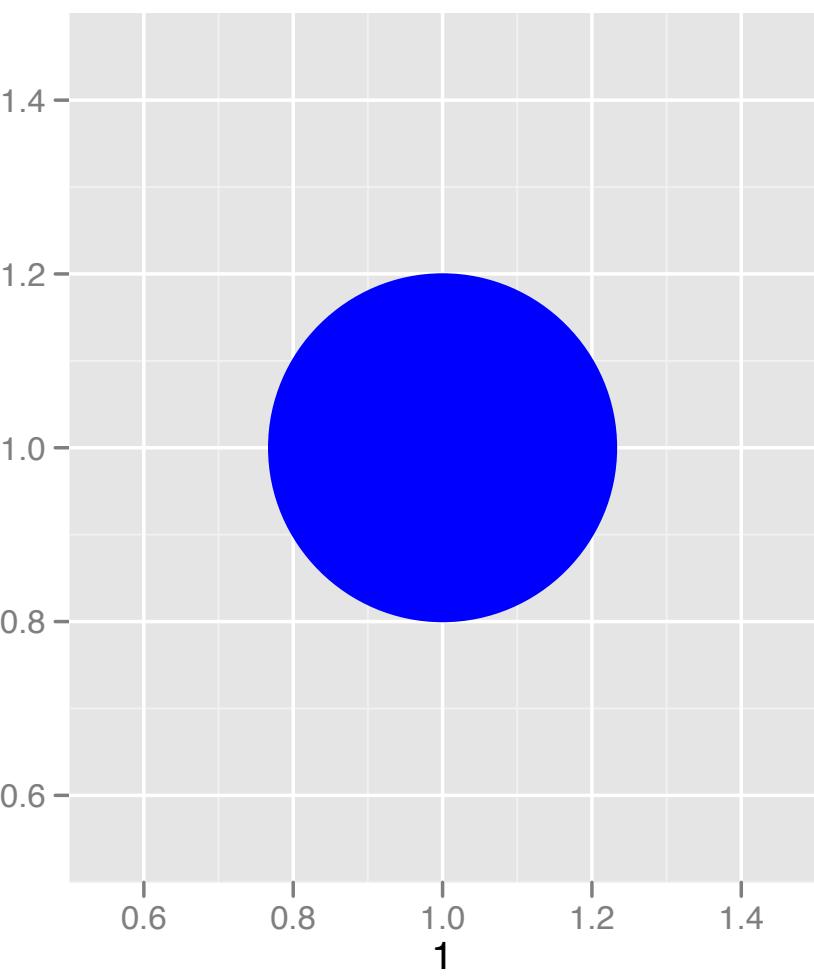
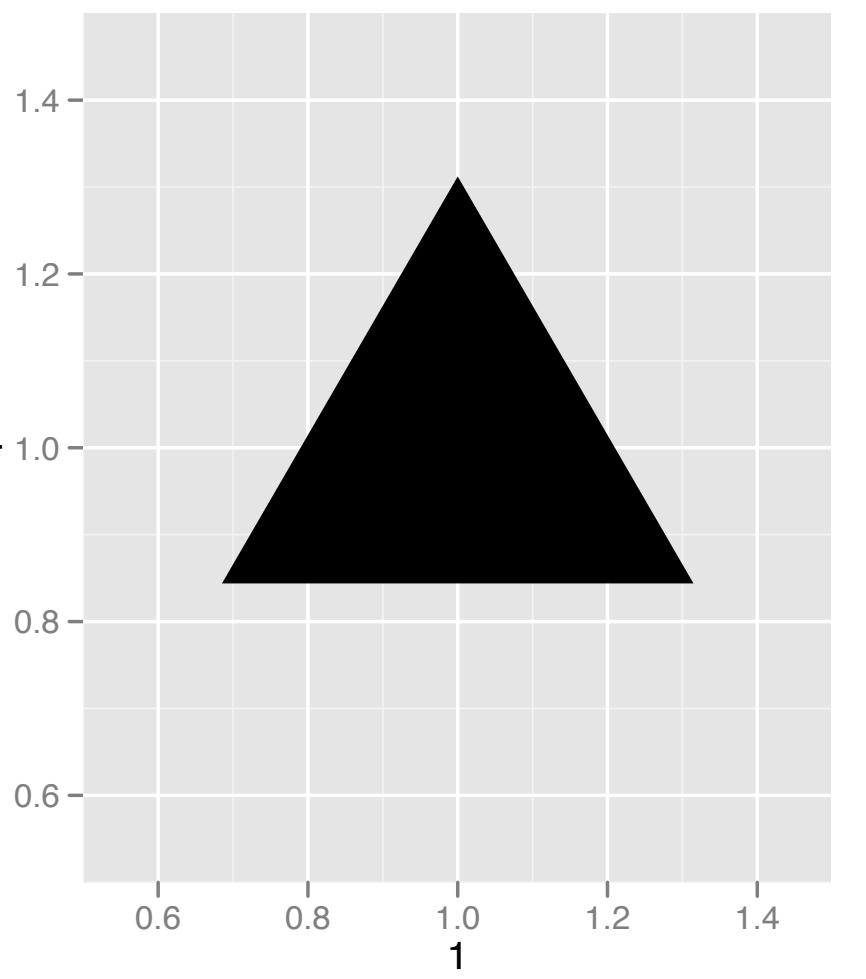
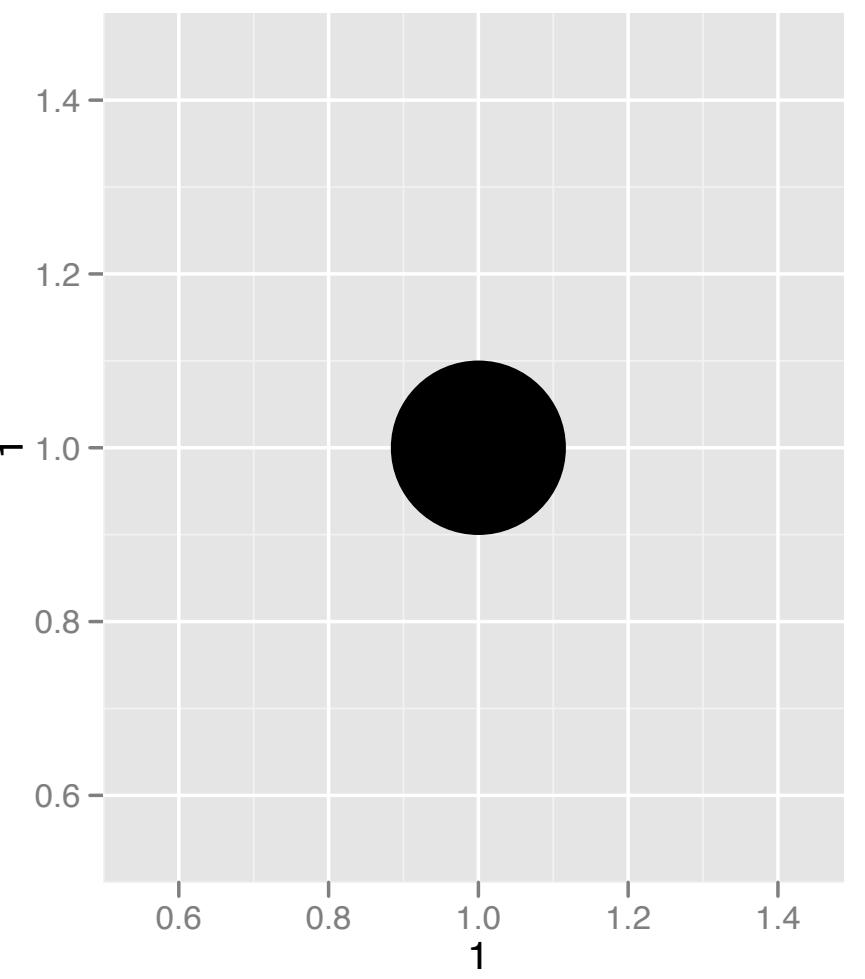
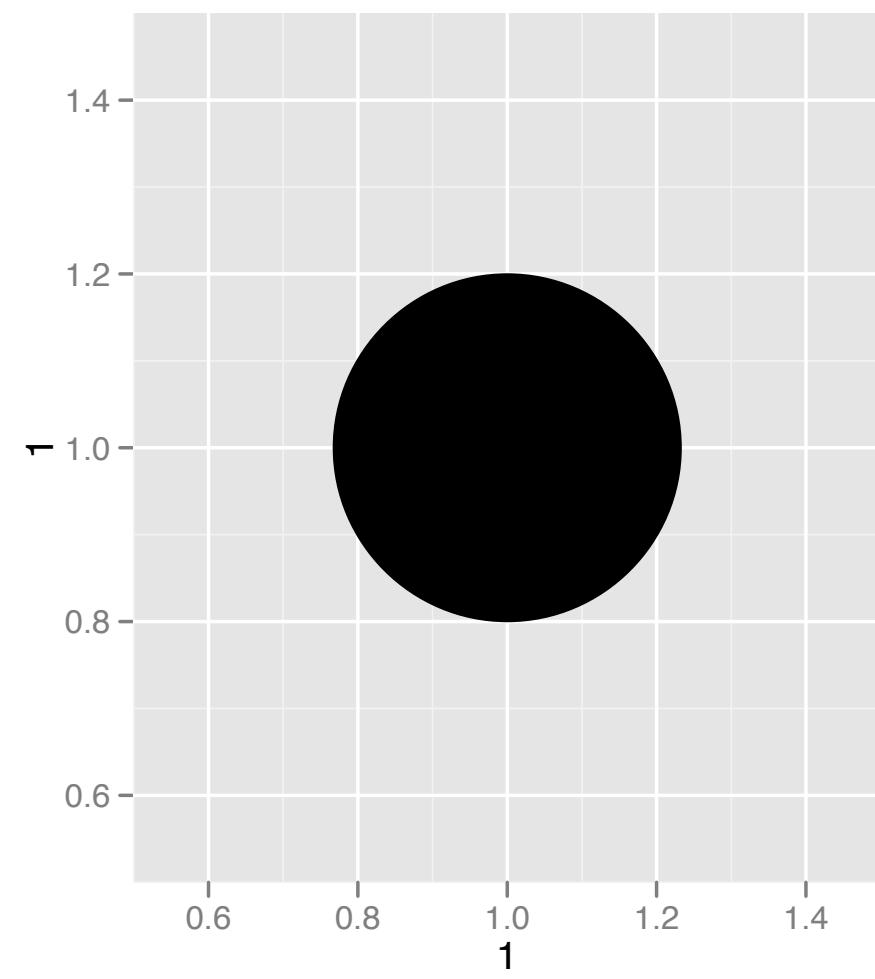


How can we test
the theory?

Why do these
cars get better
mileage?

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

Aesthetics



Visual Space

color

Red

Brown

Green

Aqua

Blue

Violet

Pink

Data Space

class

2seater

compact

midsize

minivan

pickup

subcompact

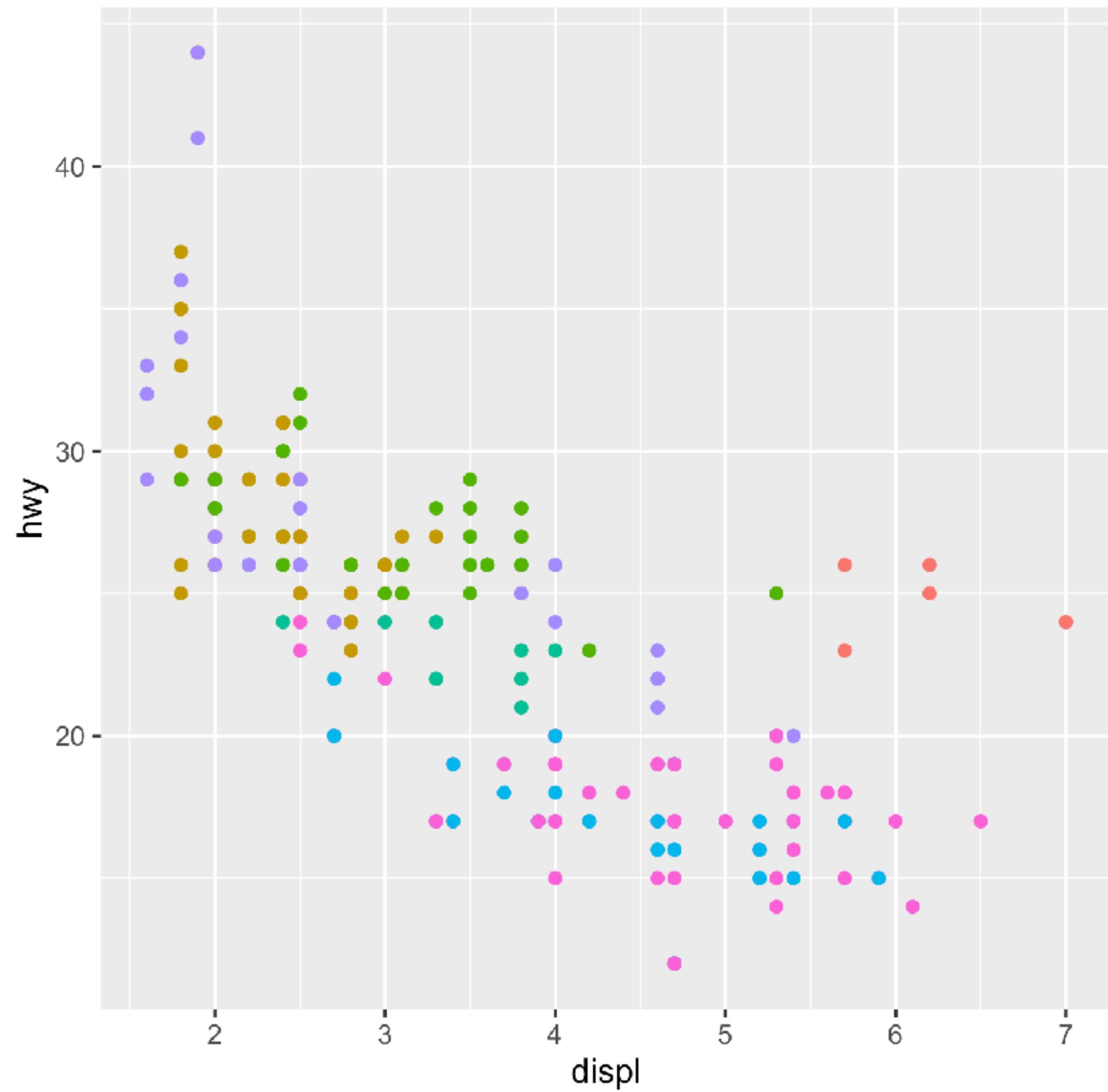
suv

Aesthetics

aesthetic
property

Variable to
map it to

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))  
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, size = class))  
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, shape = class))  
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, alpha = class))
```



Legend added
automatically

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

Your Turn 2

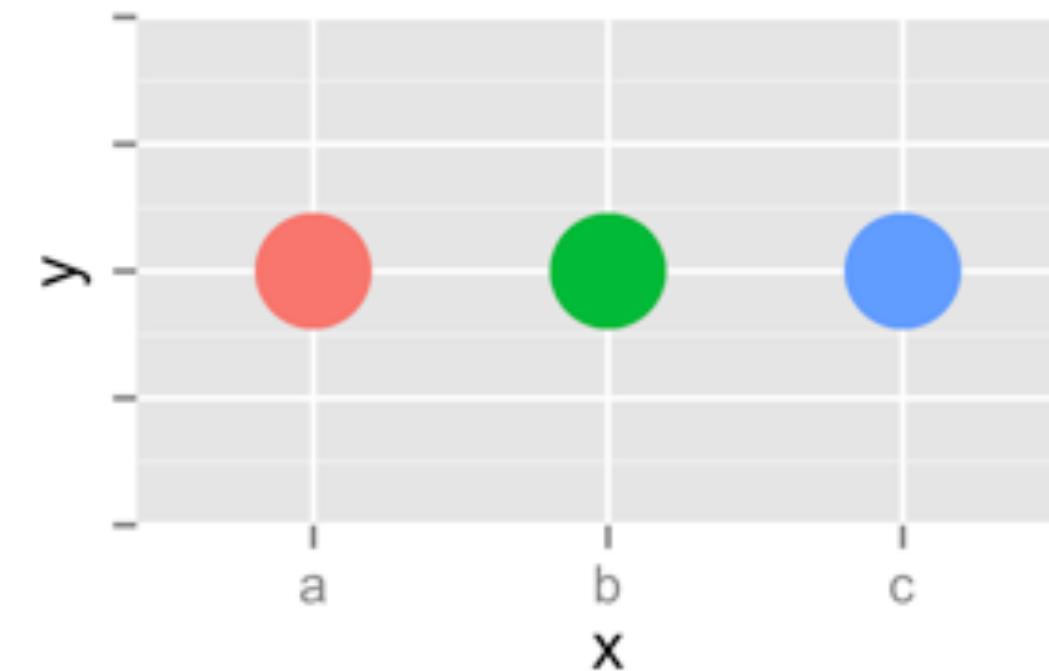
In the next chunk, add color, size, alpha, and shape aesthetics to your graph. Experiment.

Do different things happen when you map aesthetics to discrete and continuous variables?

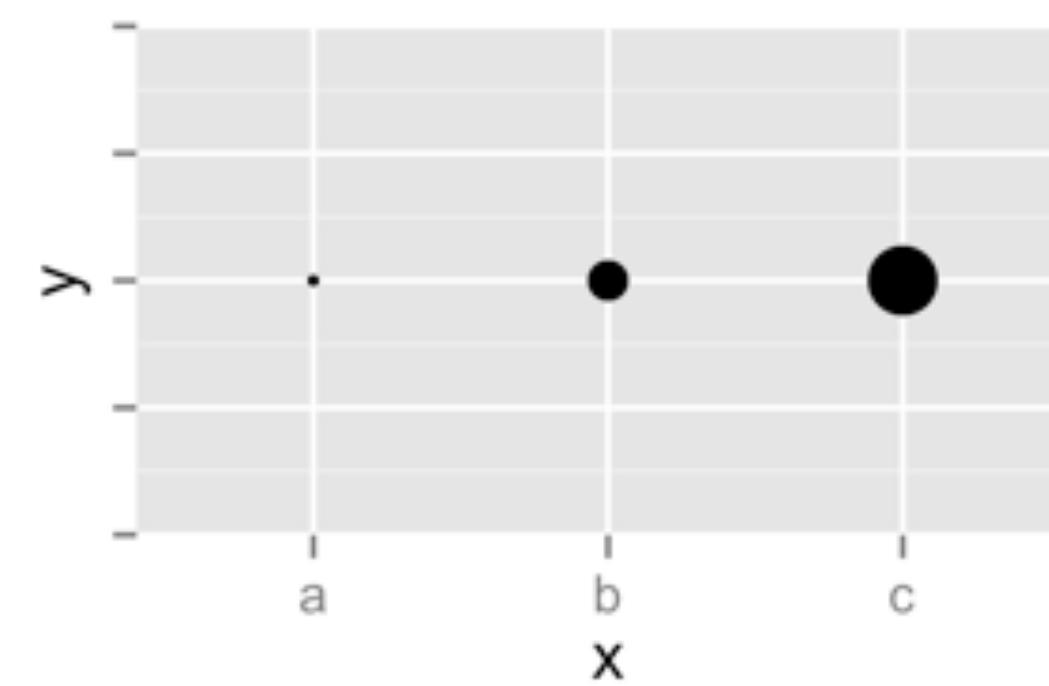
What happens when you use more than one aesthetic?



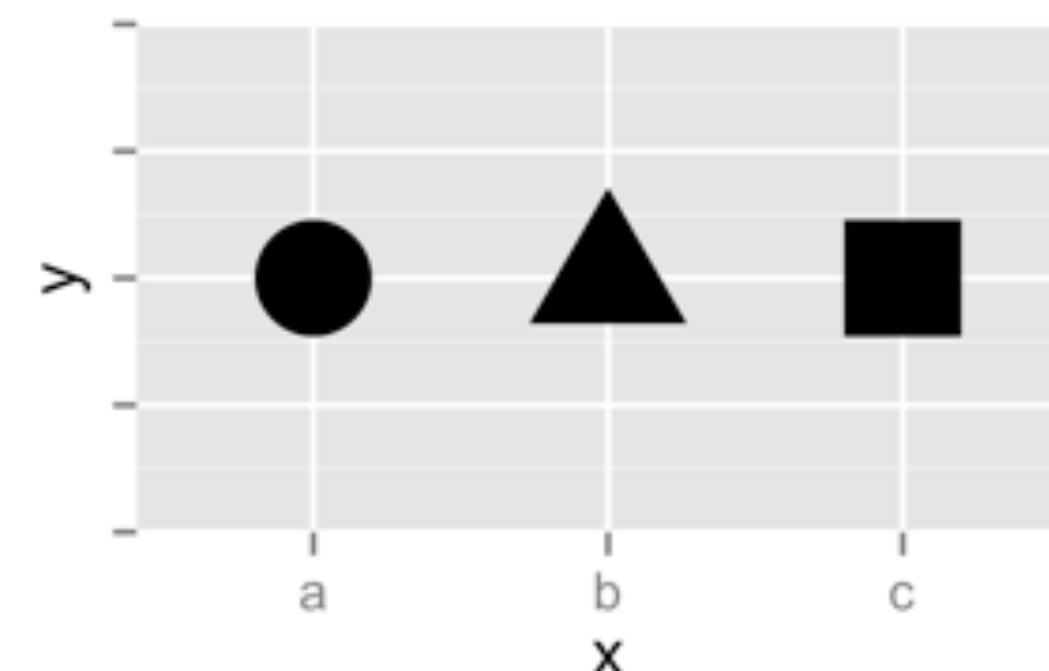
Color



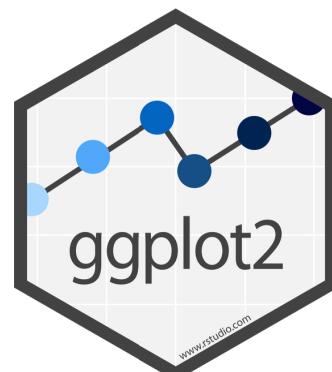
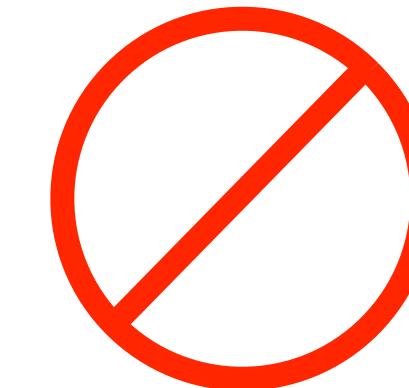
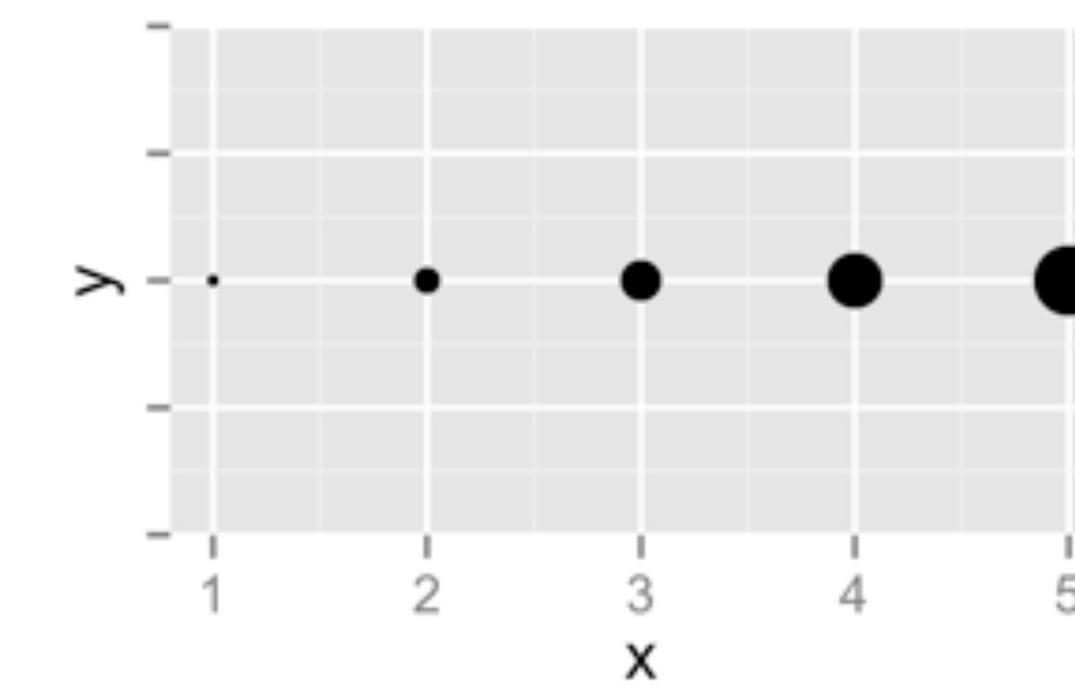
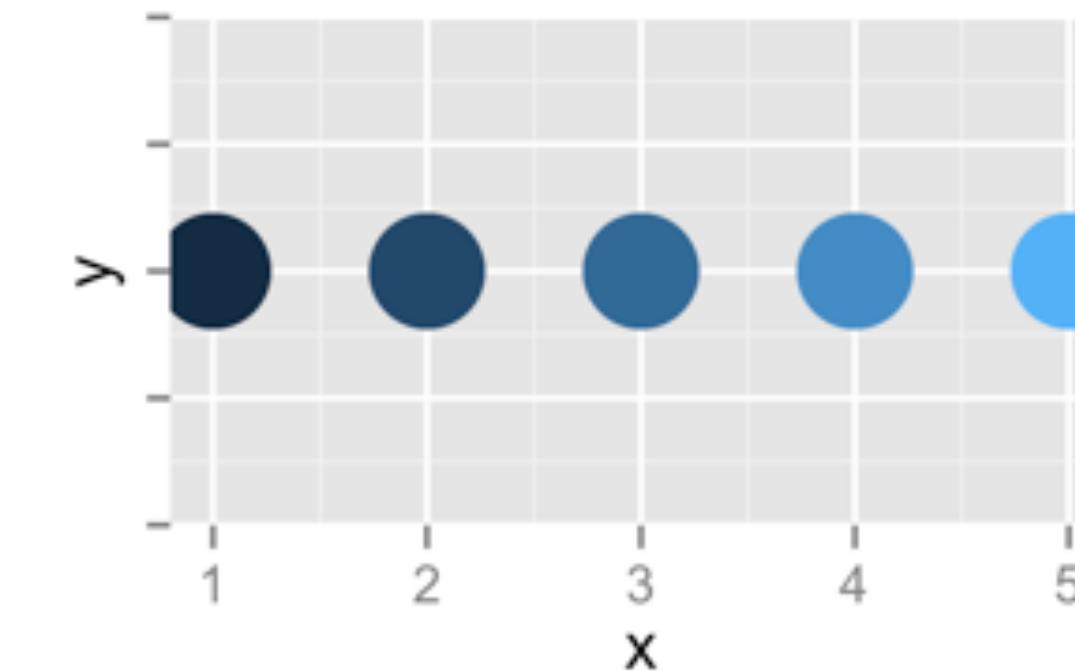
Size



Shape



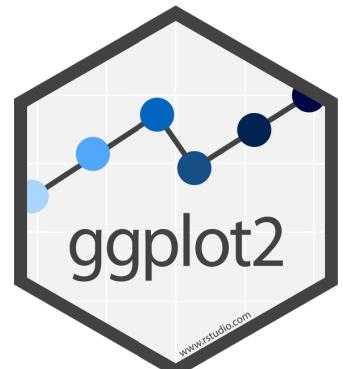
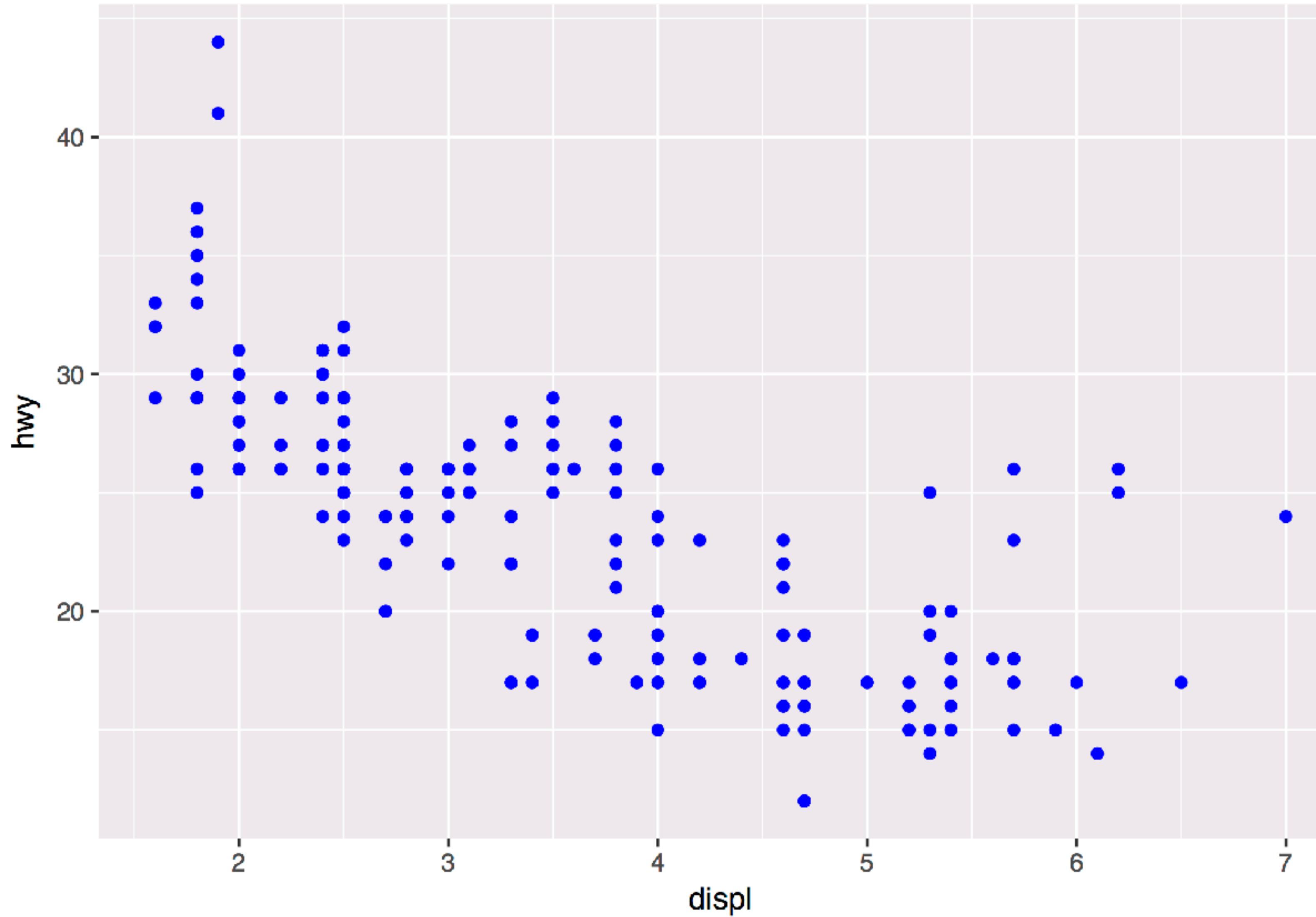
Continuous

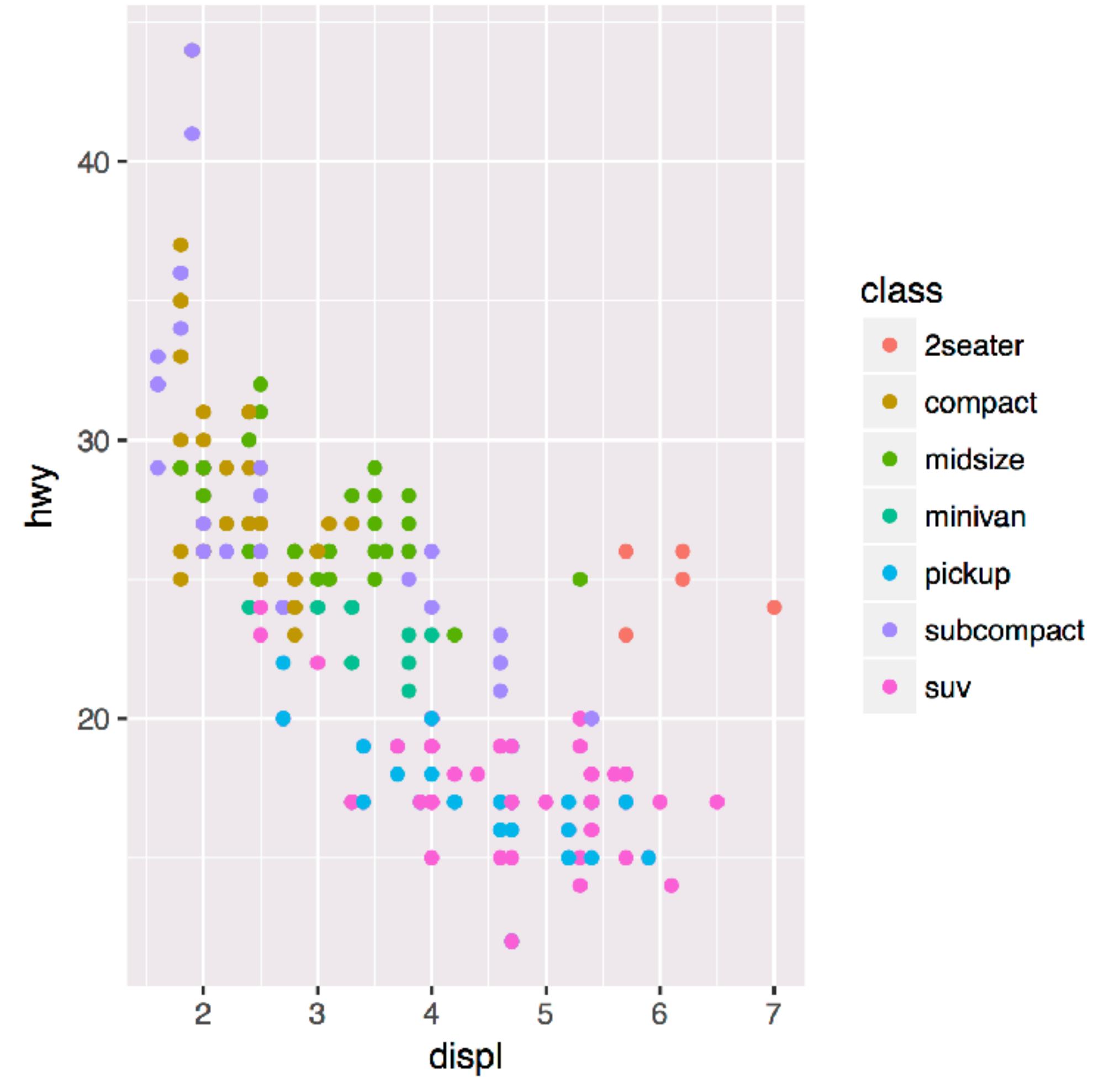


set vs. map

R

How would you make this plot?

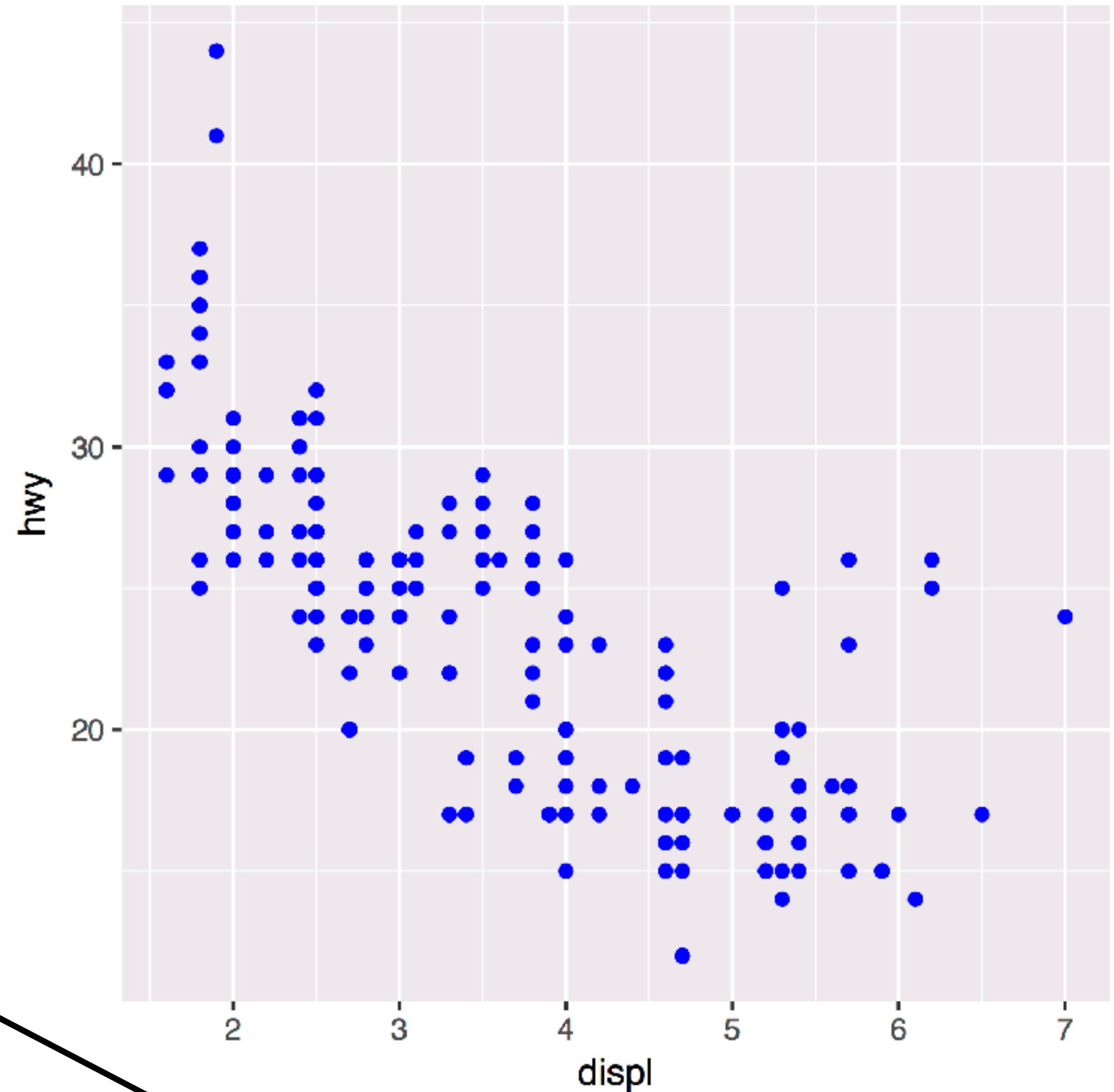




Inside of aes(): maps an aesthetic to a variable

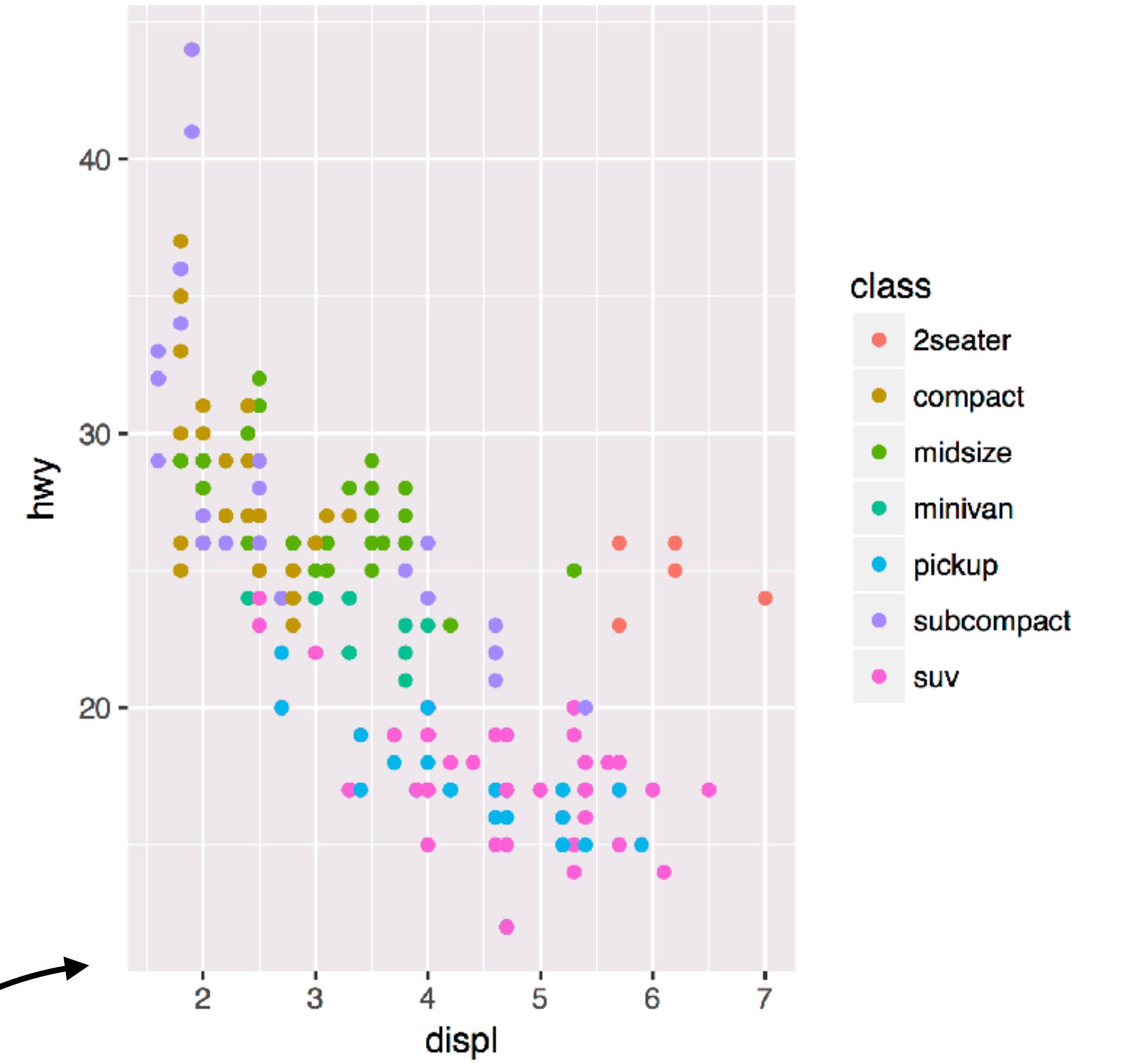
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

Outside of aes(): sets
an aesthetic to a value



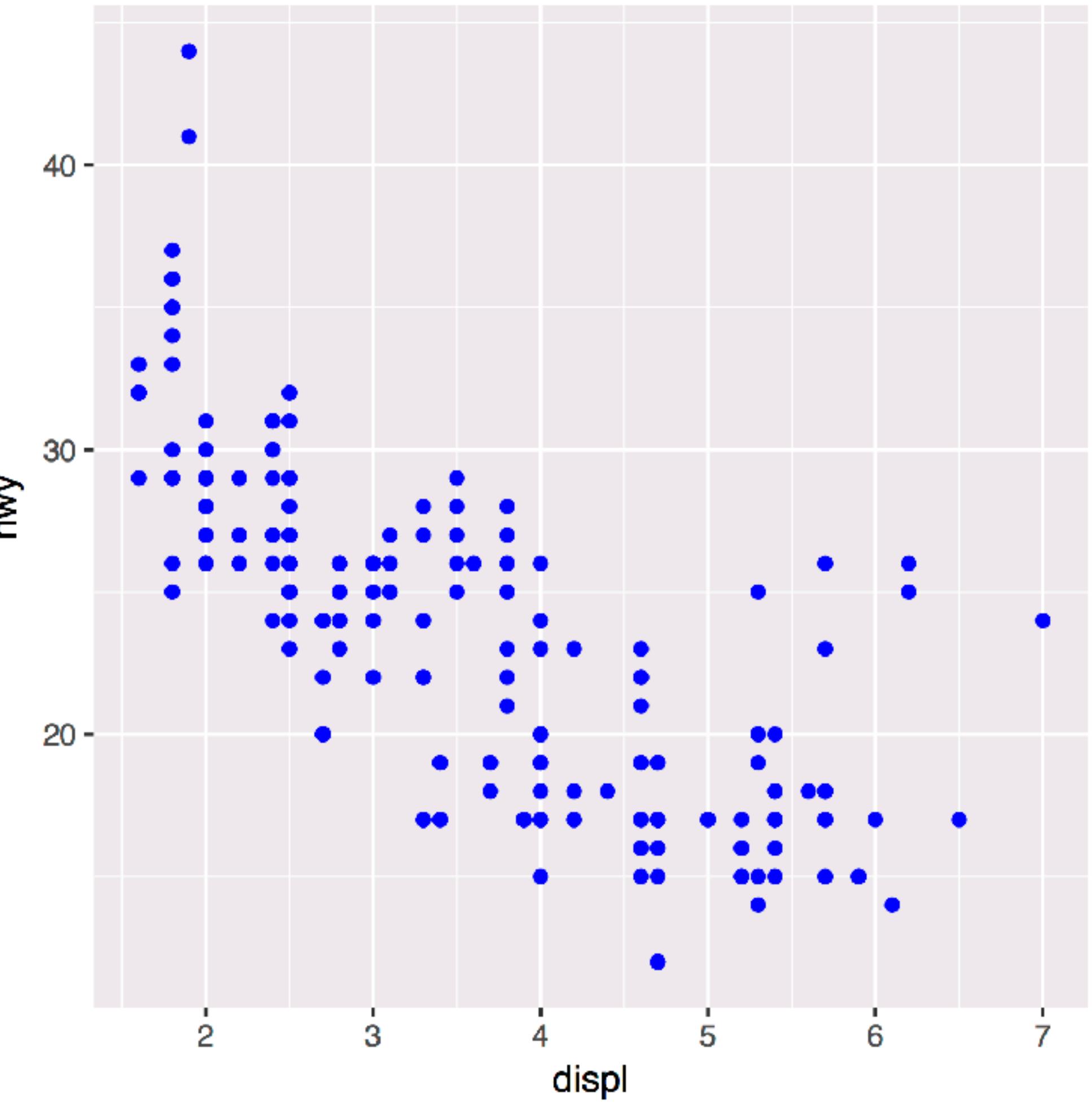
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```



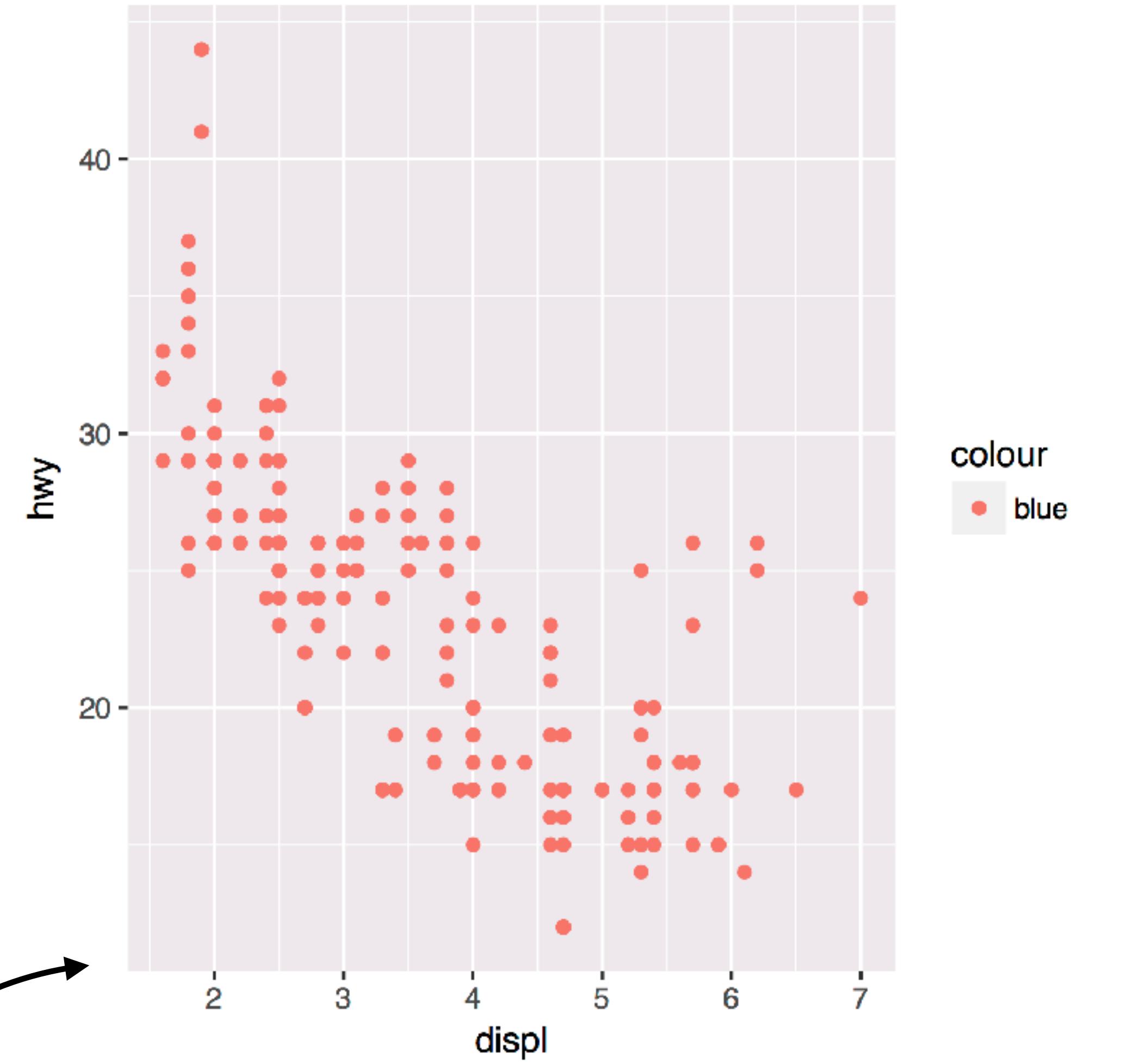
class

- 2seater
- compact
- midsize
- minivan
- pickup
- subcompact
- SUV

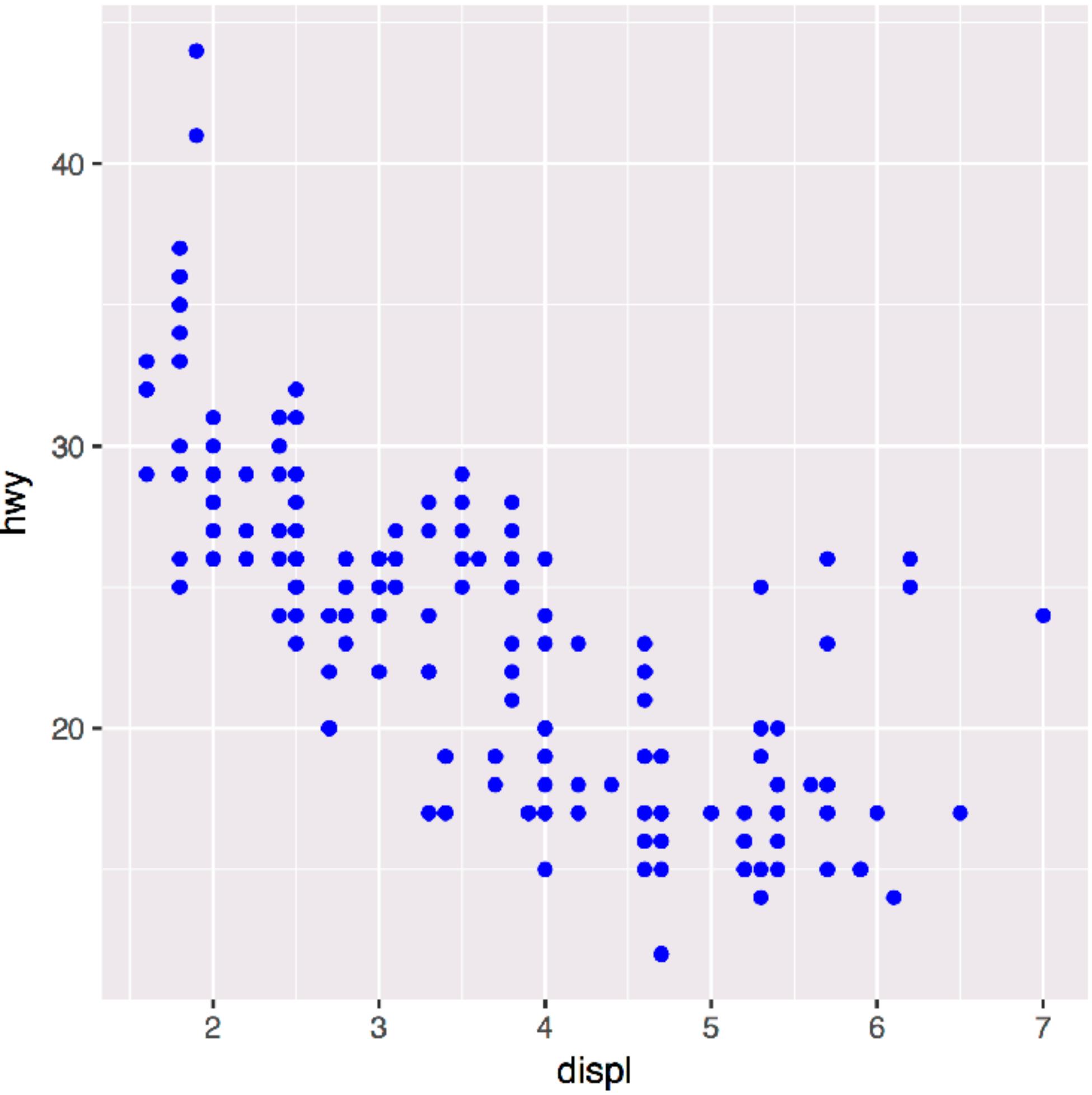


```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```



colour
● blue



```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = "blue"))
```

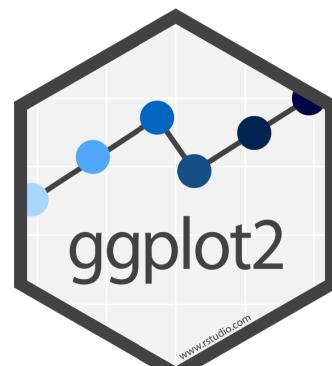
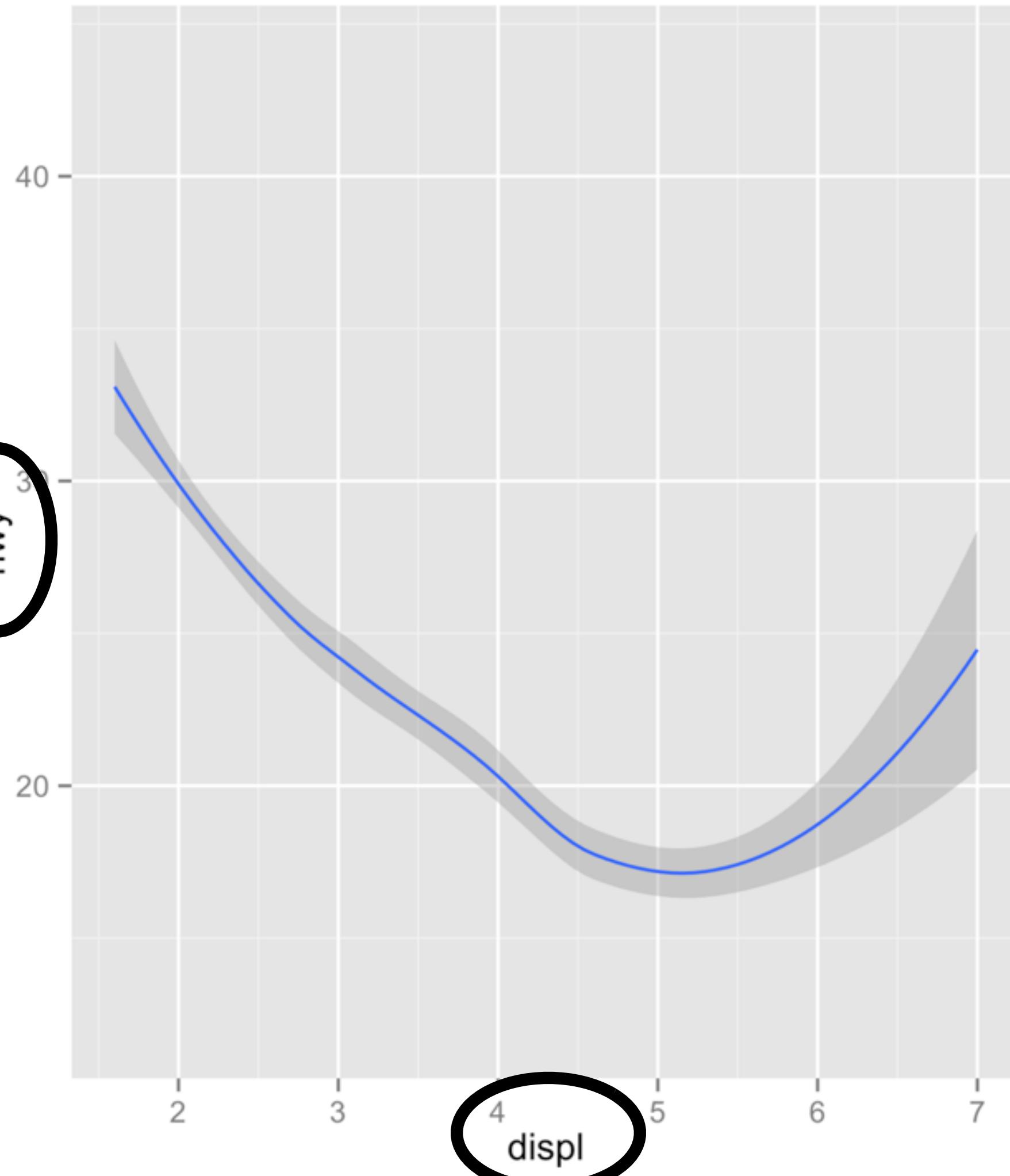
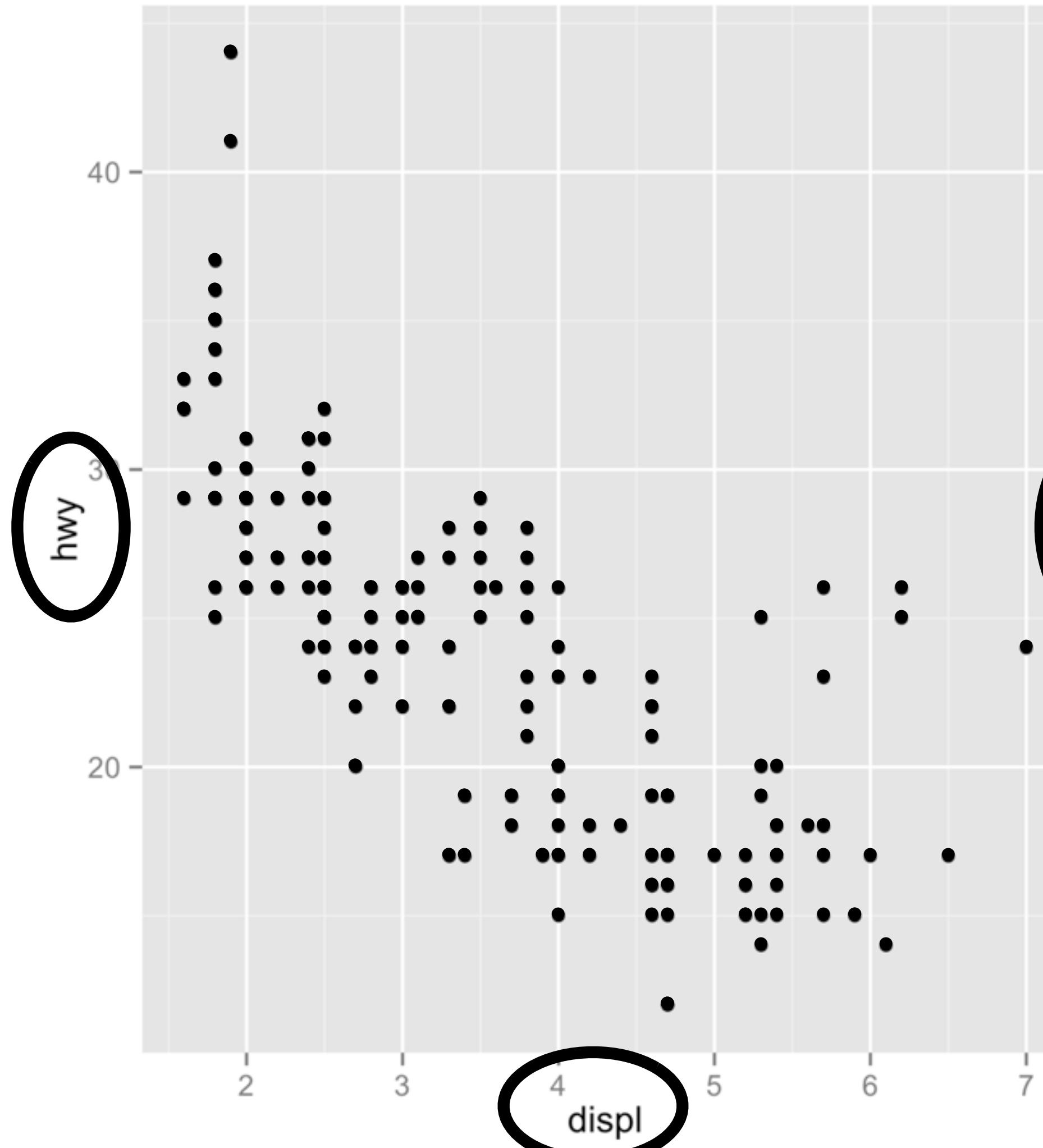
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```

Geoms

R

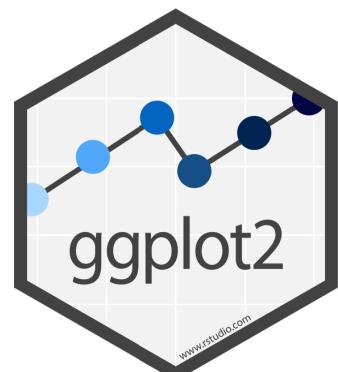
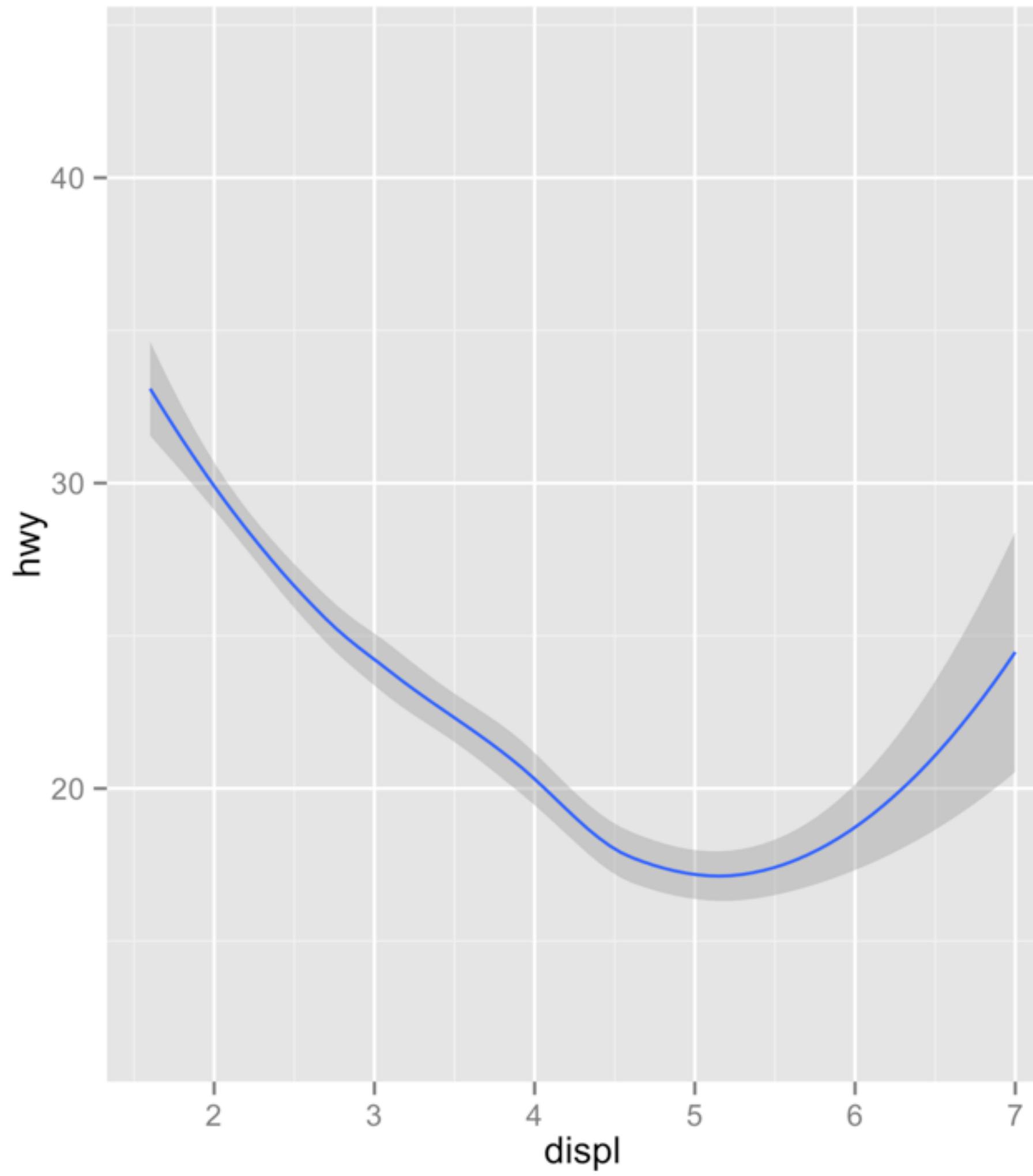
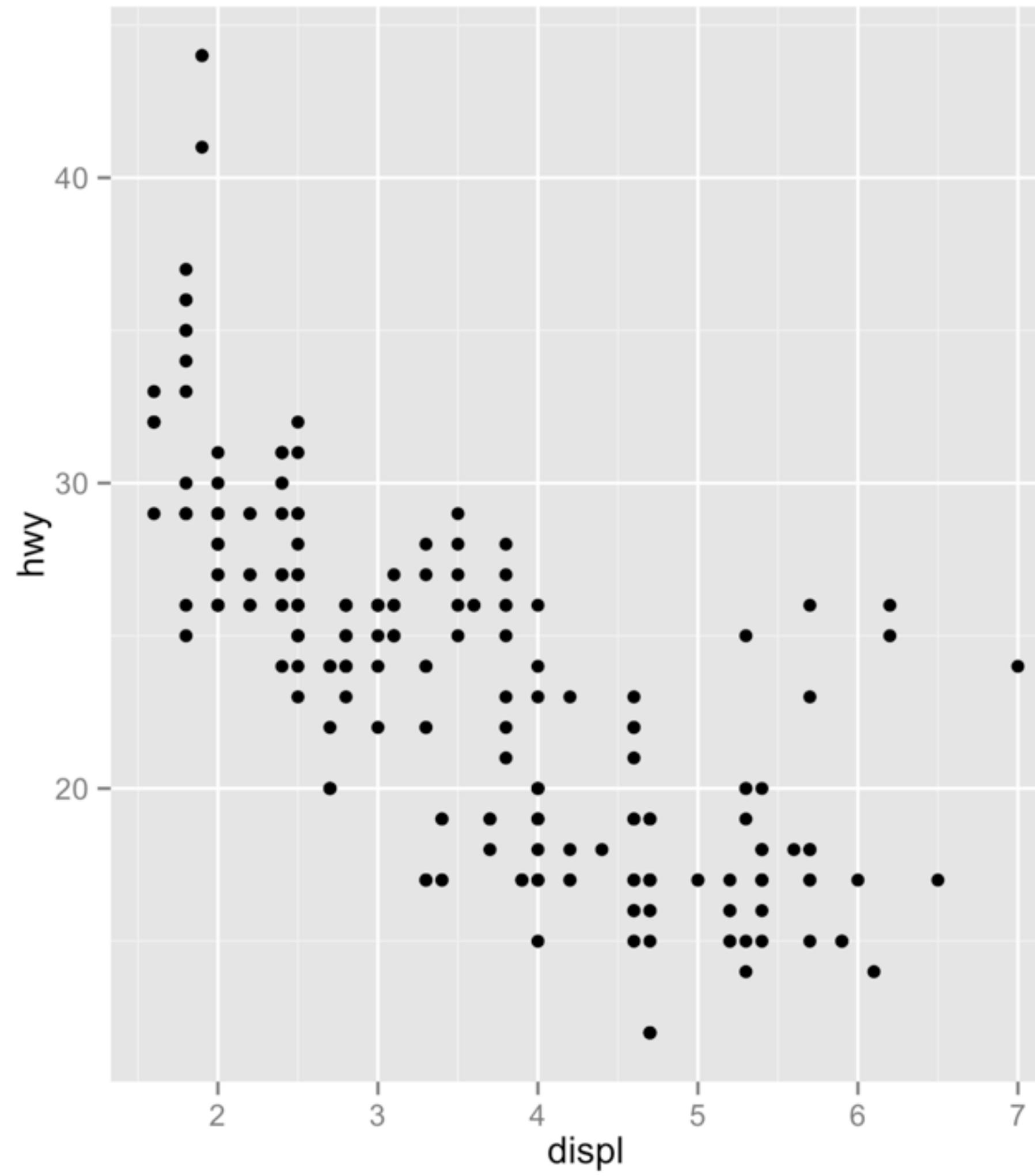
How are these plots similar?

Same: x var , y var , data



How are these plots different?

Different: geometric object (geom),
e.g. the visual object used to represent the data



geoms

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

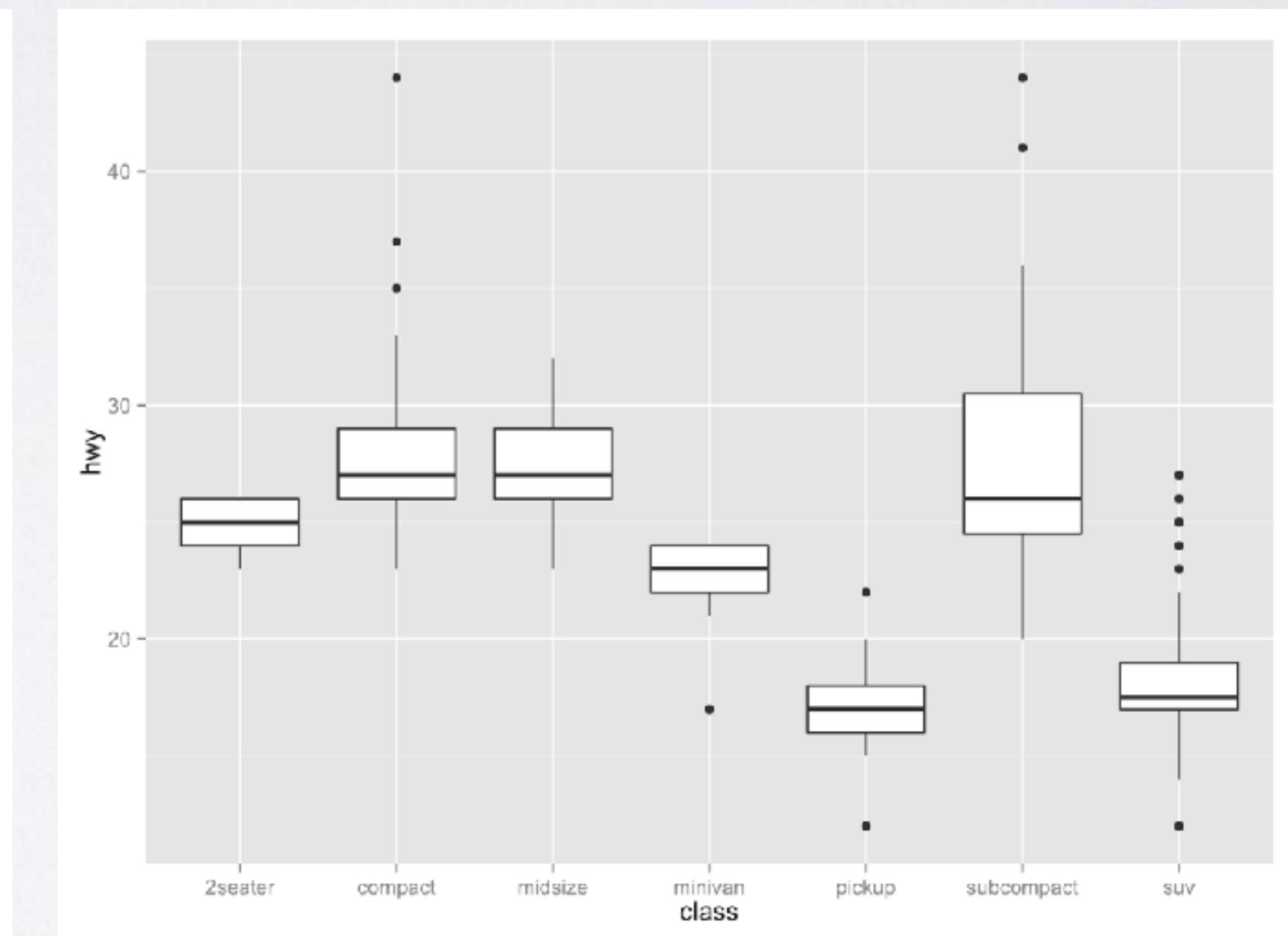
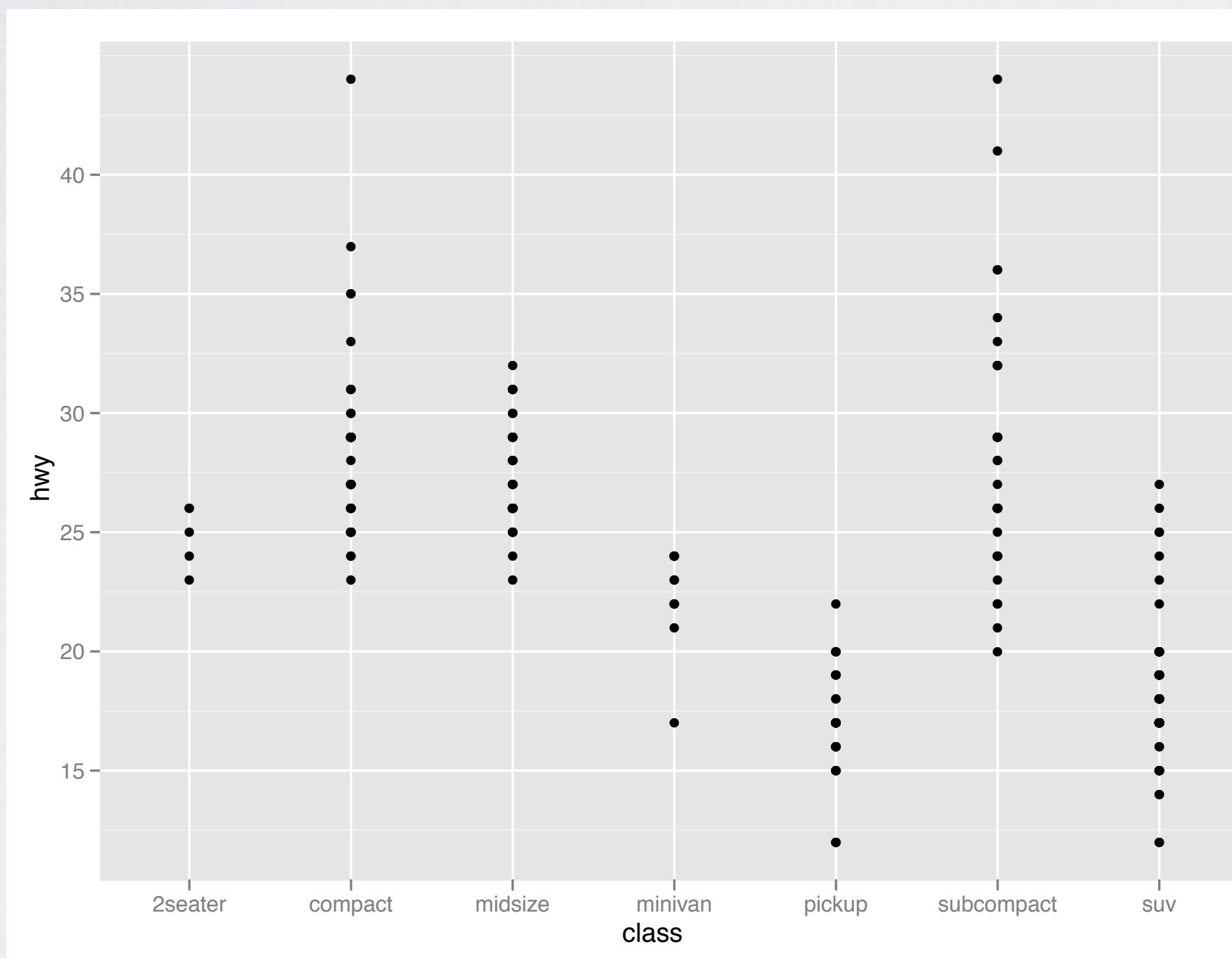

Your Turn

Pair up.



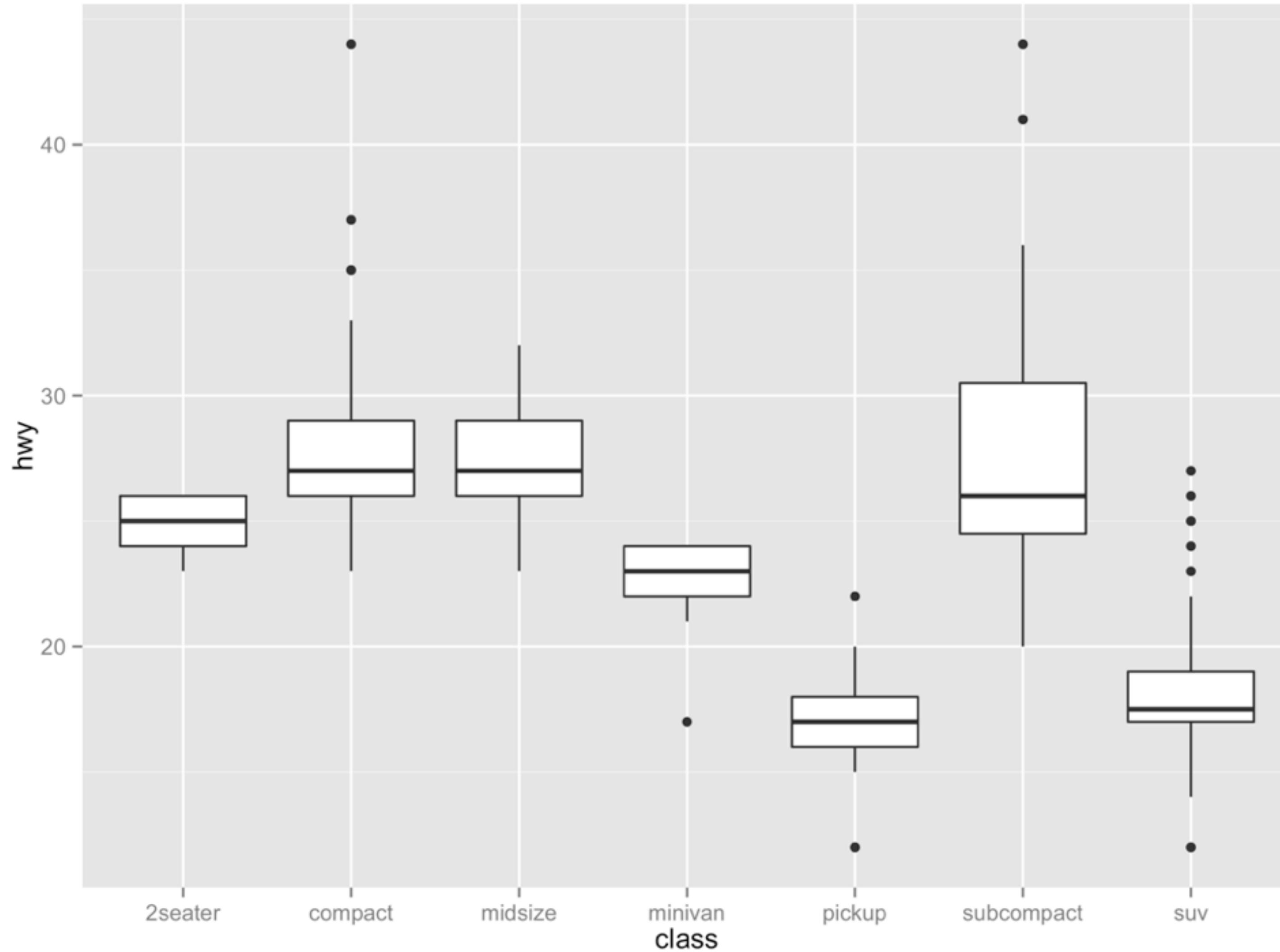
Your Turn 3

With your partner, decide how to replace this scatterplot with one that draws boxplots? Use the cheatsheet. Try your best guess.



```
ggplot(mpg) + geom_point(aes(class, hwy))
```

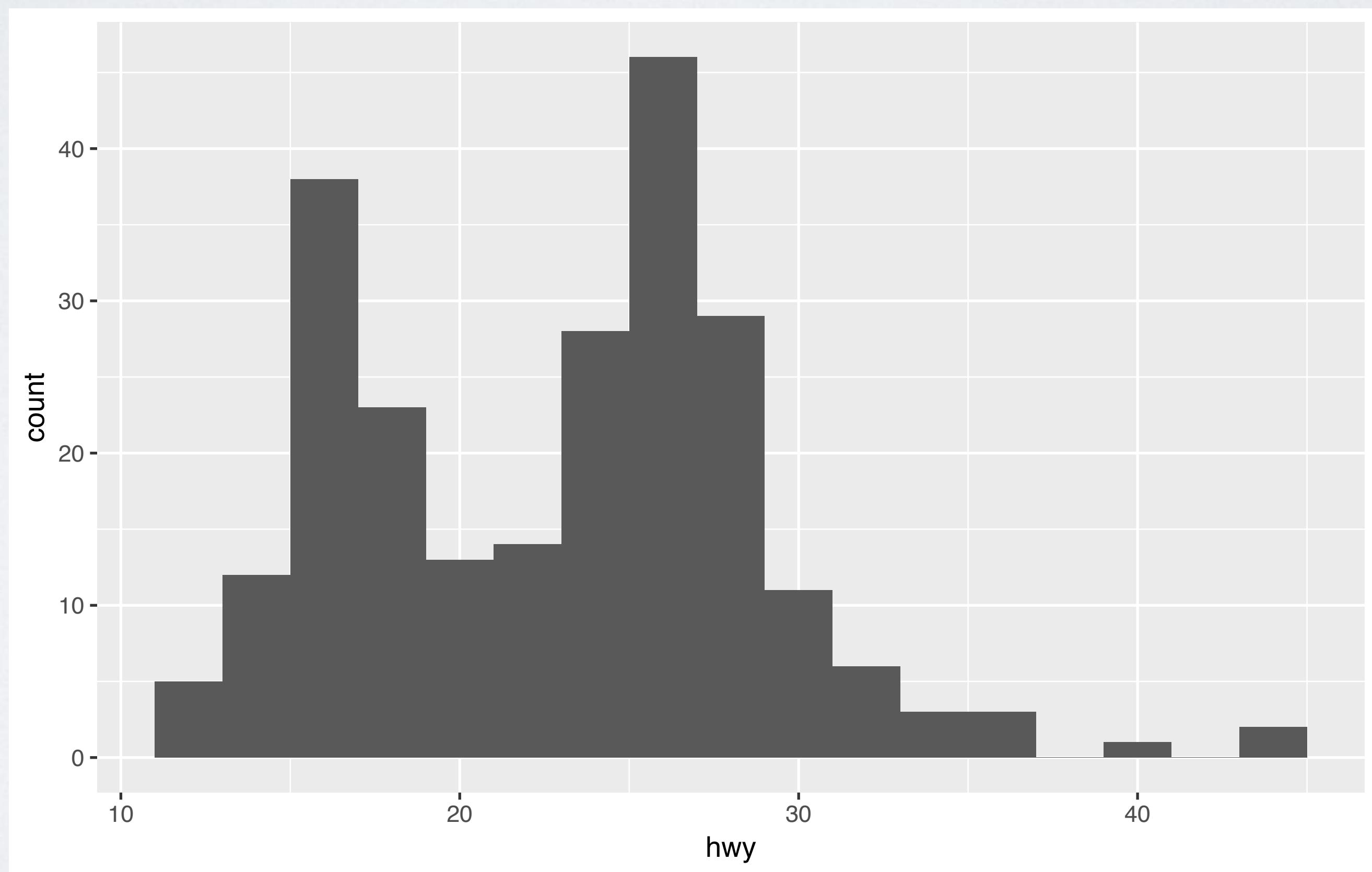
02 : 00

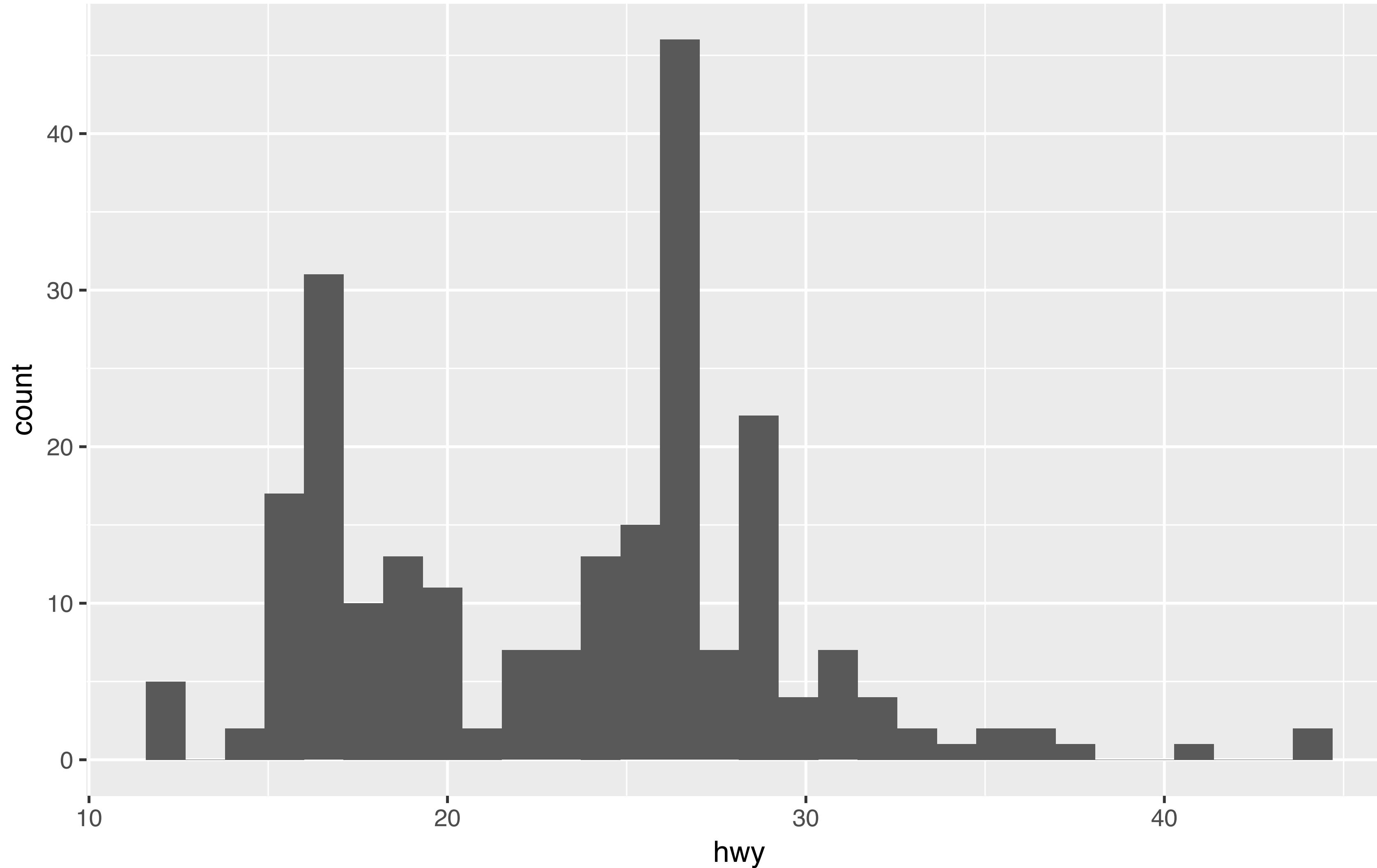


```
ggplot(data = mpg) +  
  geom_boxplot(mapping = aes(x = class, y = hwy))
```

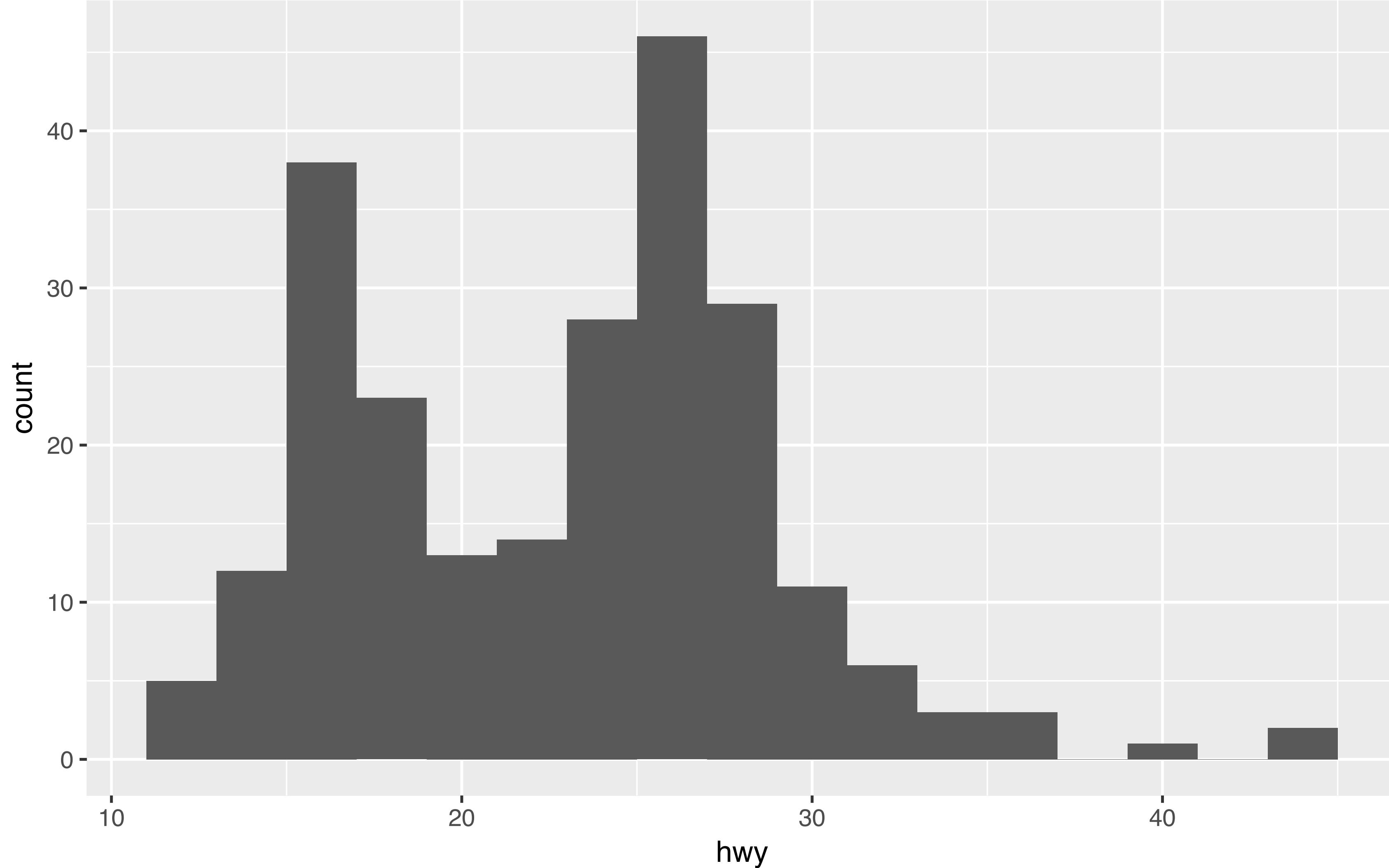
Your Turn 4

With your partner, make the histogram of `hwy` below. Use the cheatsheet. Hint: do not supply a `y` variable.

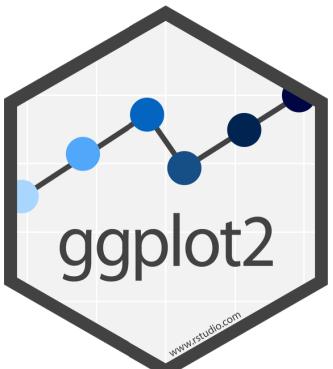




```
ggplot(data = mpg) +  
  geom_histogram(mapping = aes(x = hwy))
```

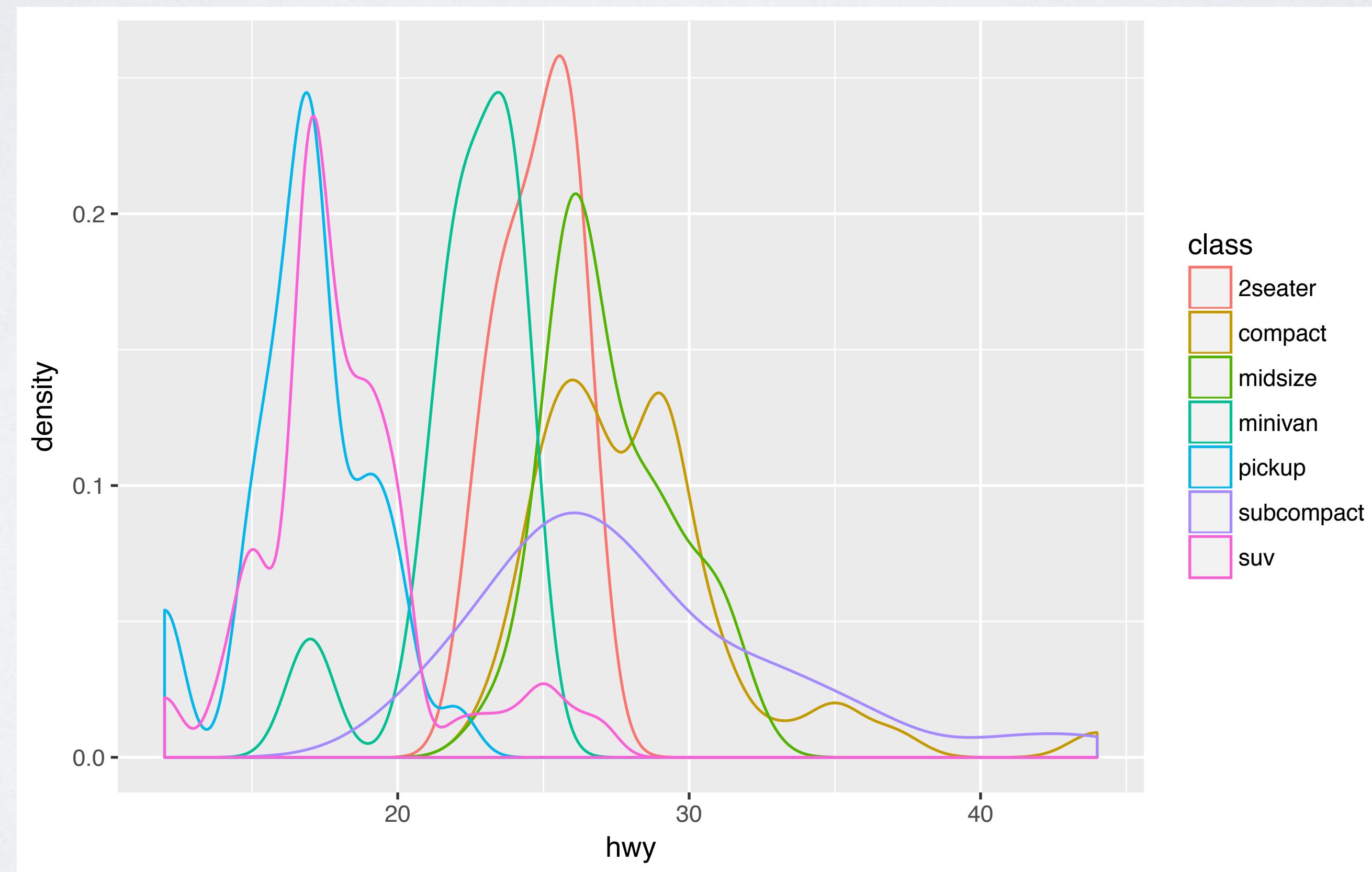


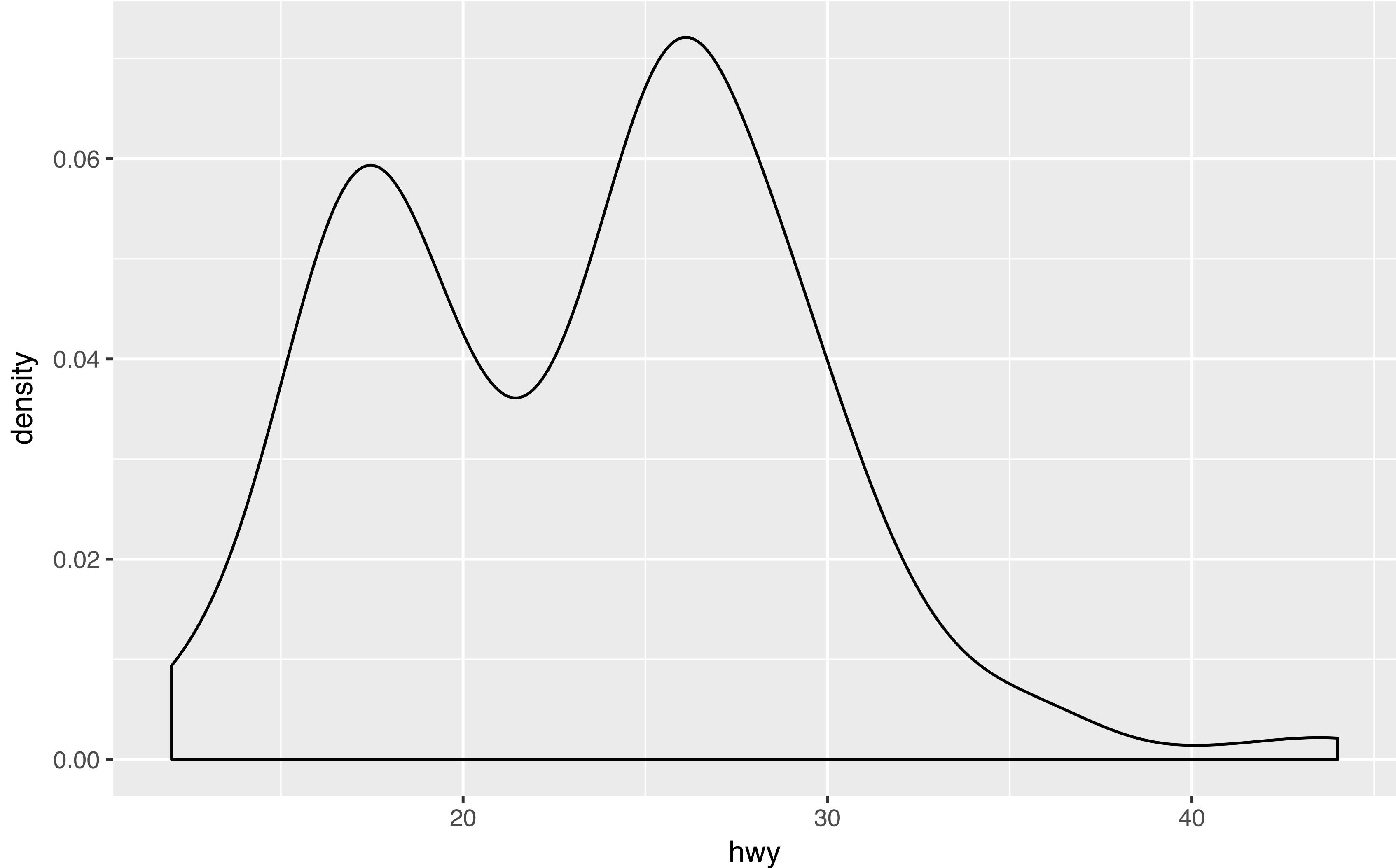
```
ggplot(data = mpg) +  
  geom_histogram(mapping = aes(x = hwy), binwidth = 2)
```



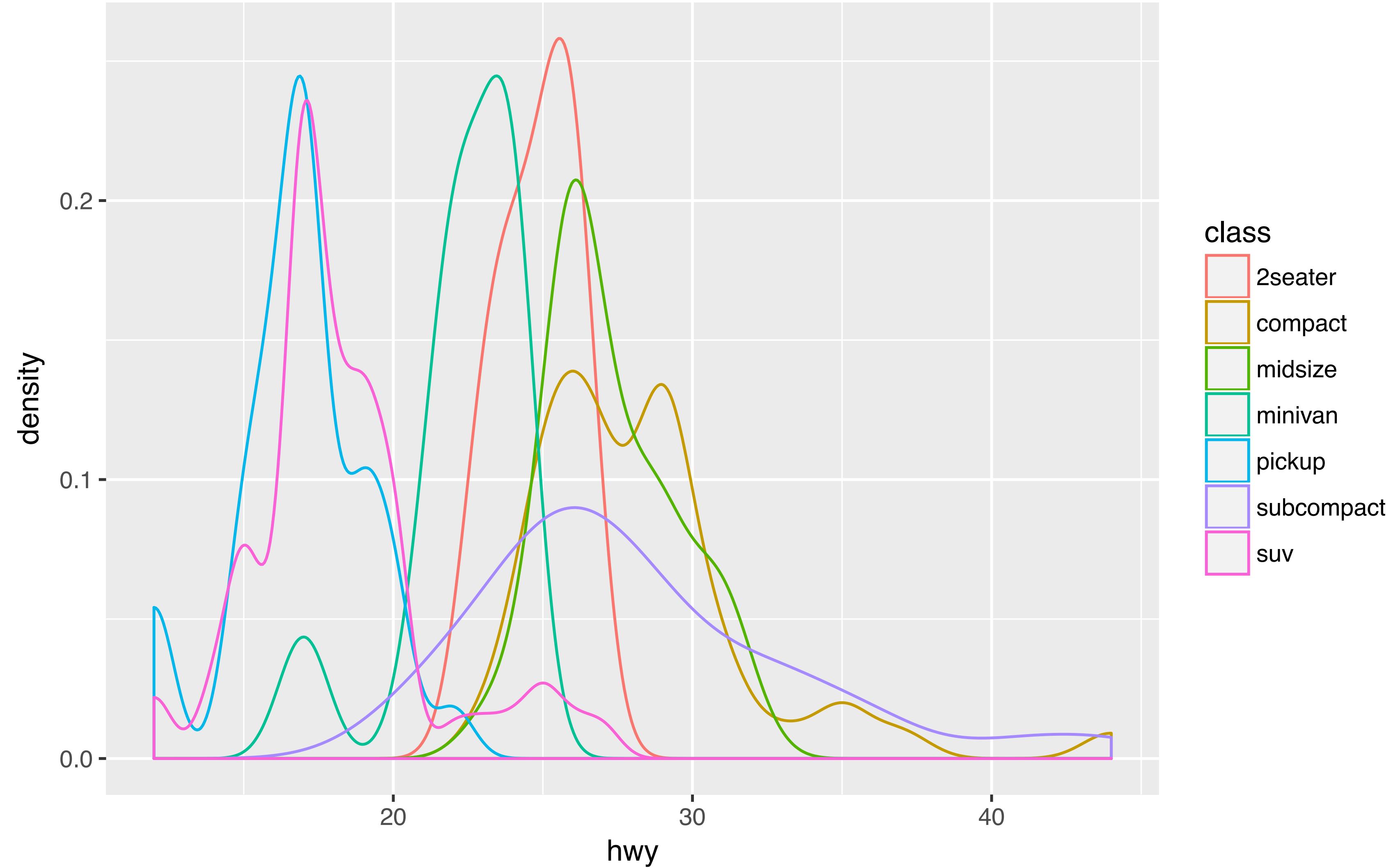
Your Turn 5

With your partner, make the density plot of **hwy** colored by **class** below. Use the cheatsheet. Try your best guess.

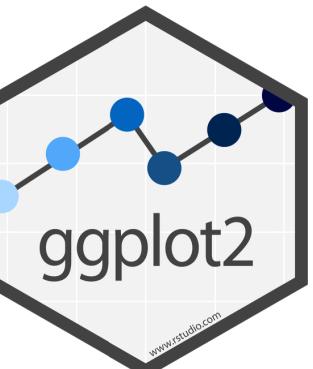


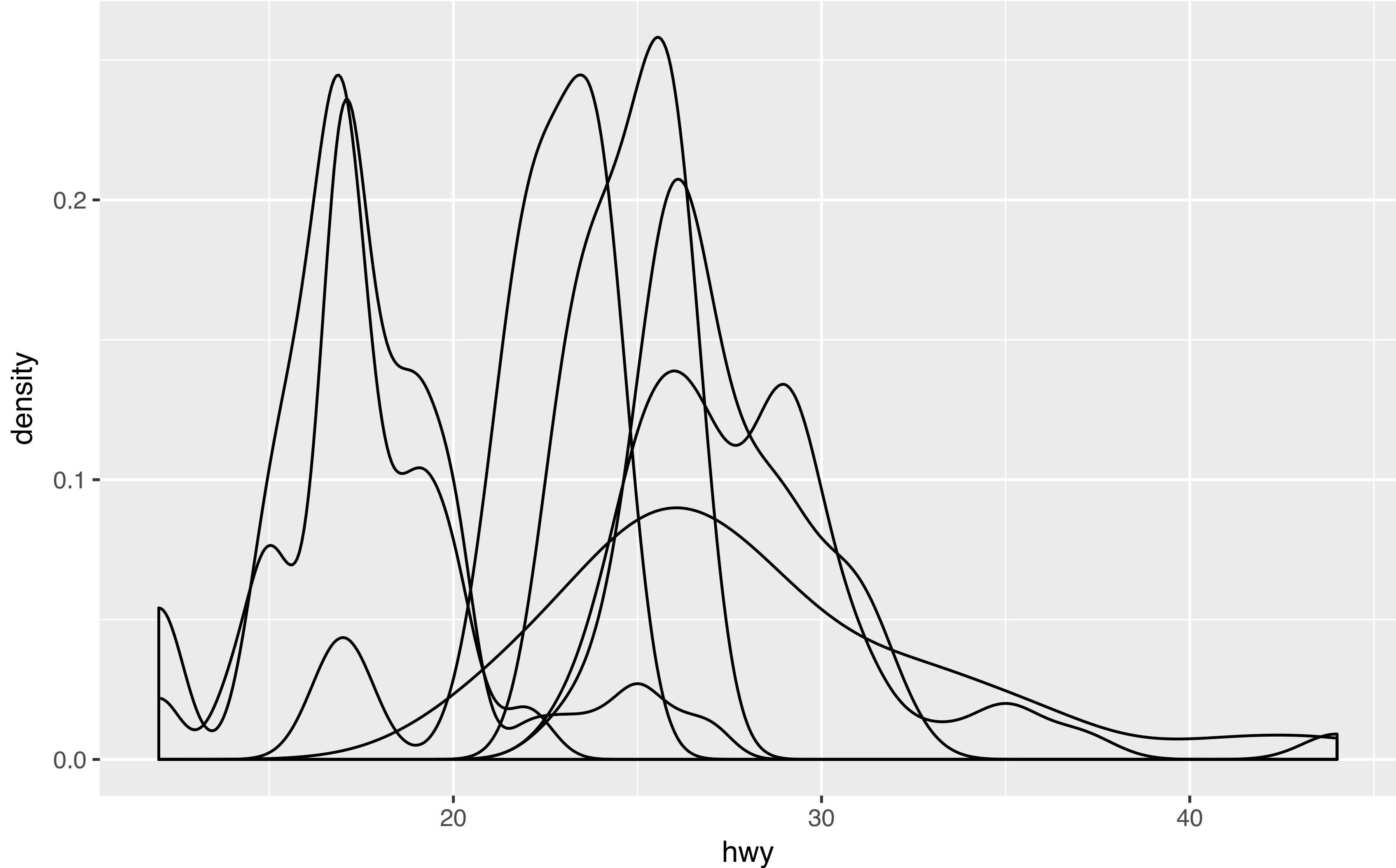


```
ggplot(data = mpg) +  
  geom_density(mapping = aes(x = hwy))
```

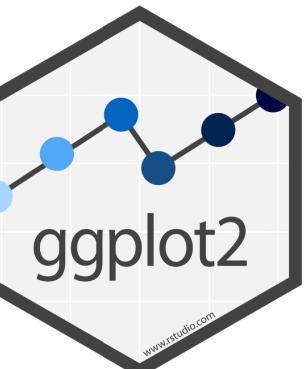


```
ggplot(data = mpg) +  
  geom_density(mapping = aes(x = hwy, color = class))
```



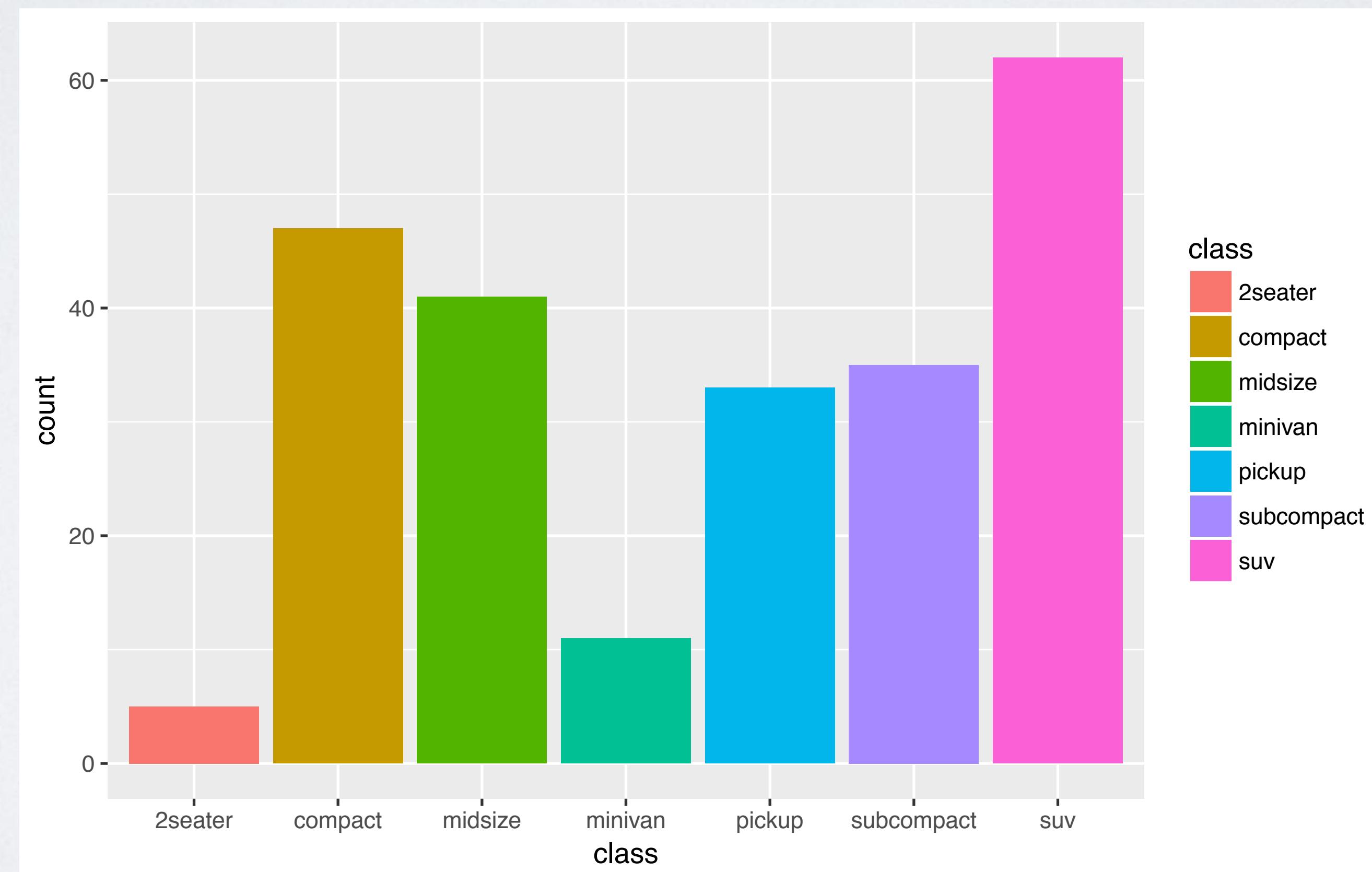


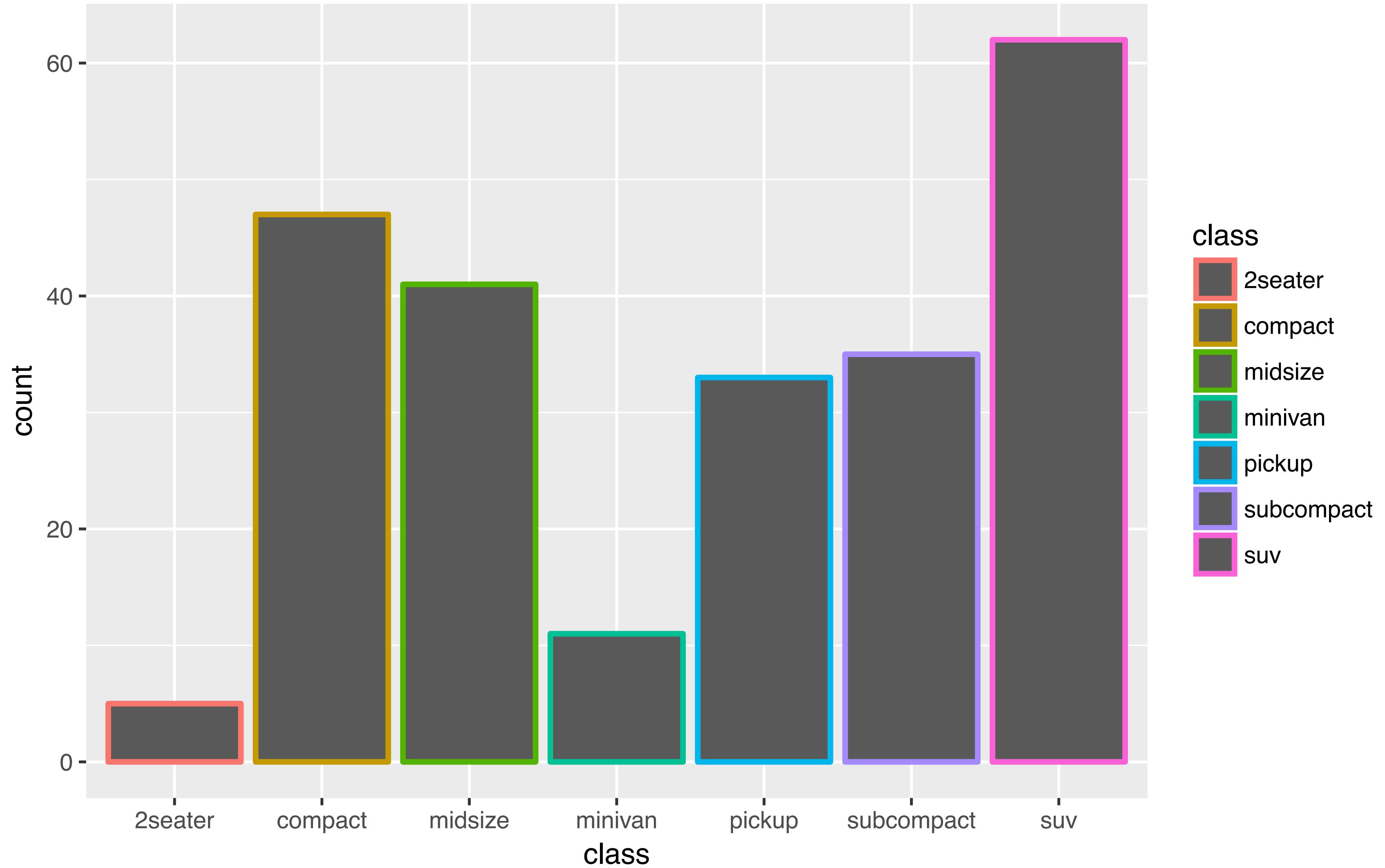
```
ggplot(data = mpg) +  
  geom_density(mapping = aes(x = hwy, group = class))
```



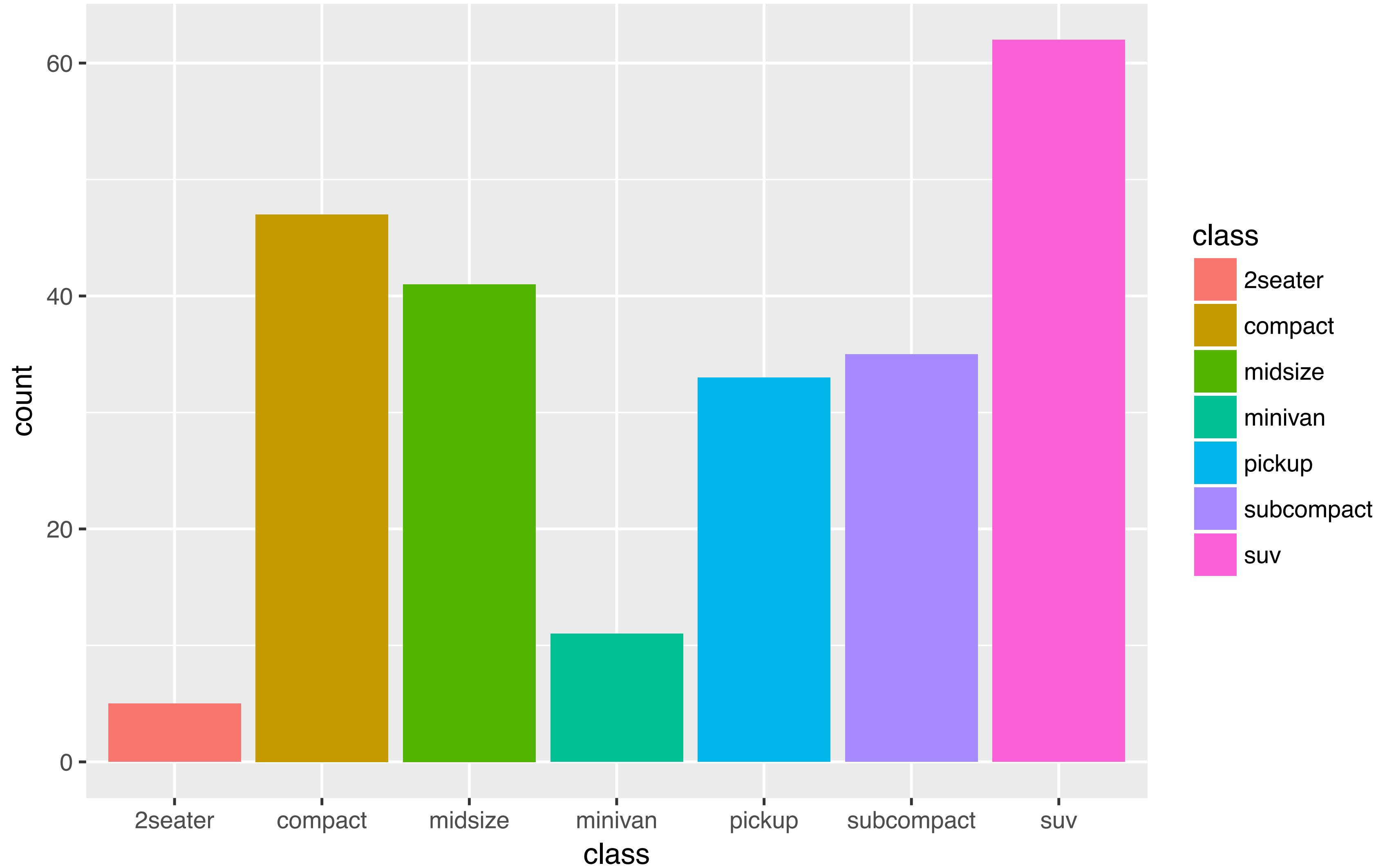
Your Turn 6

With your partner, make the bar chart of **class** colored by **class** below. Use the cheatsheet. Try your best guess.





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



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

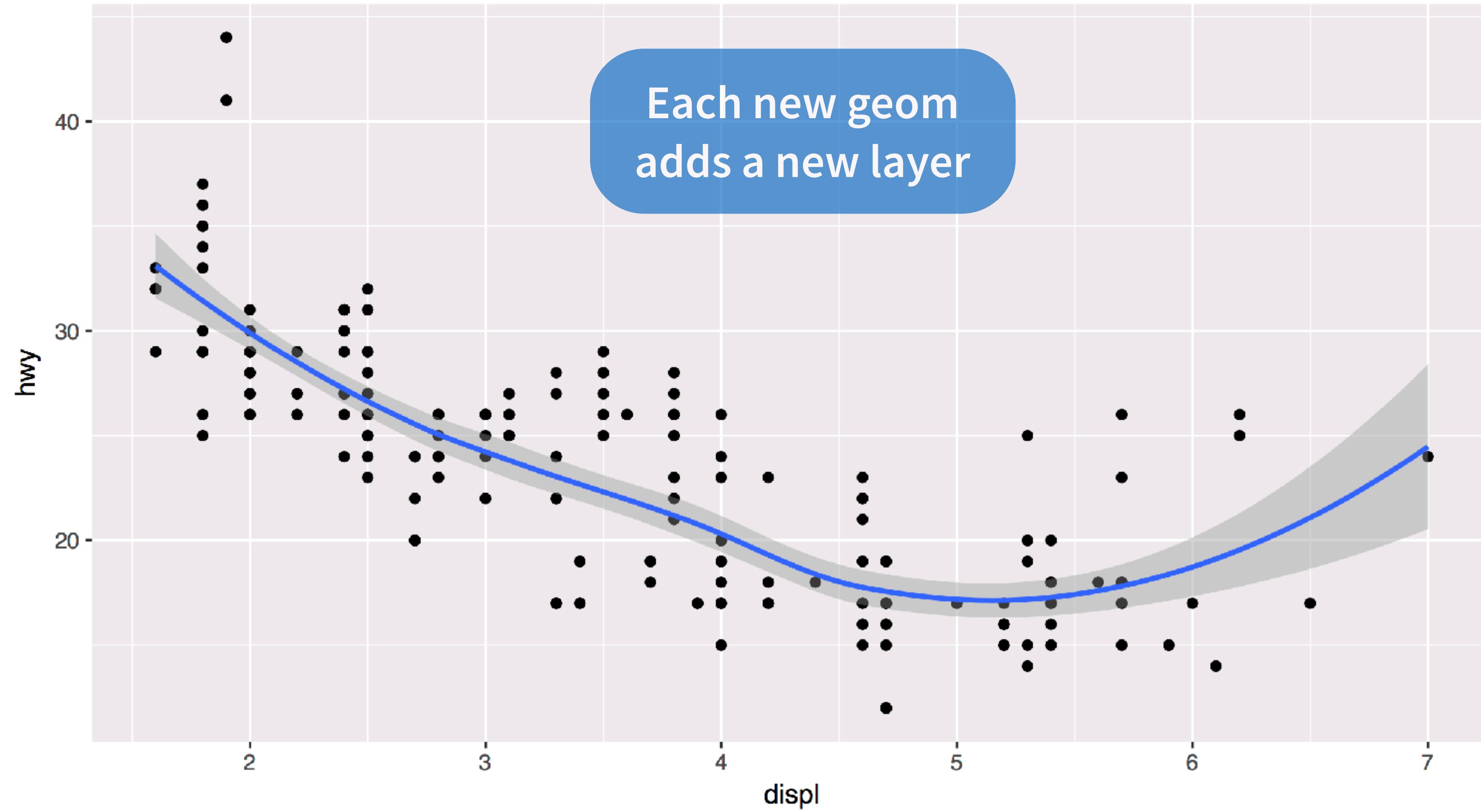
Your Turn 7

With your partner, predict what this code will do.

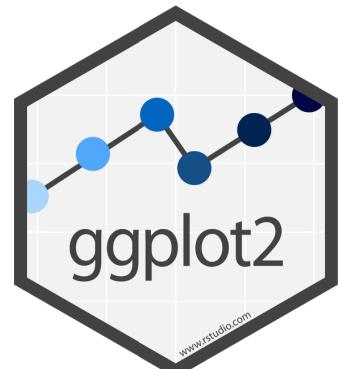
Then run it.

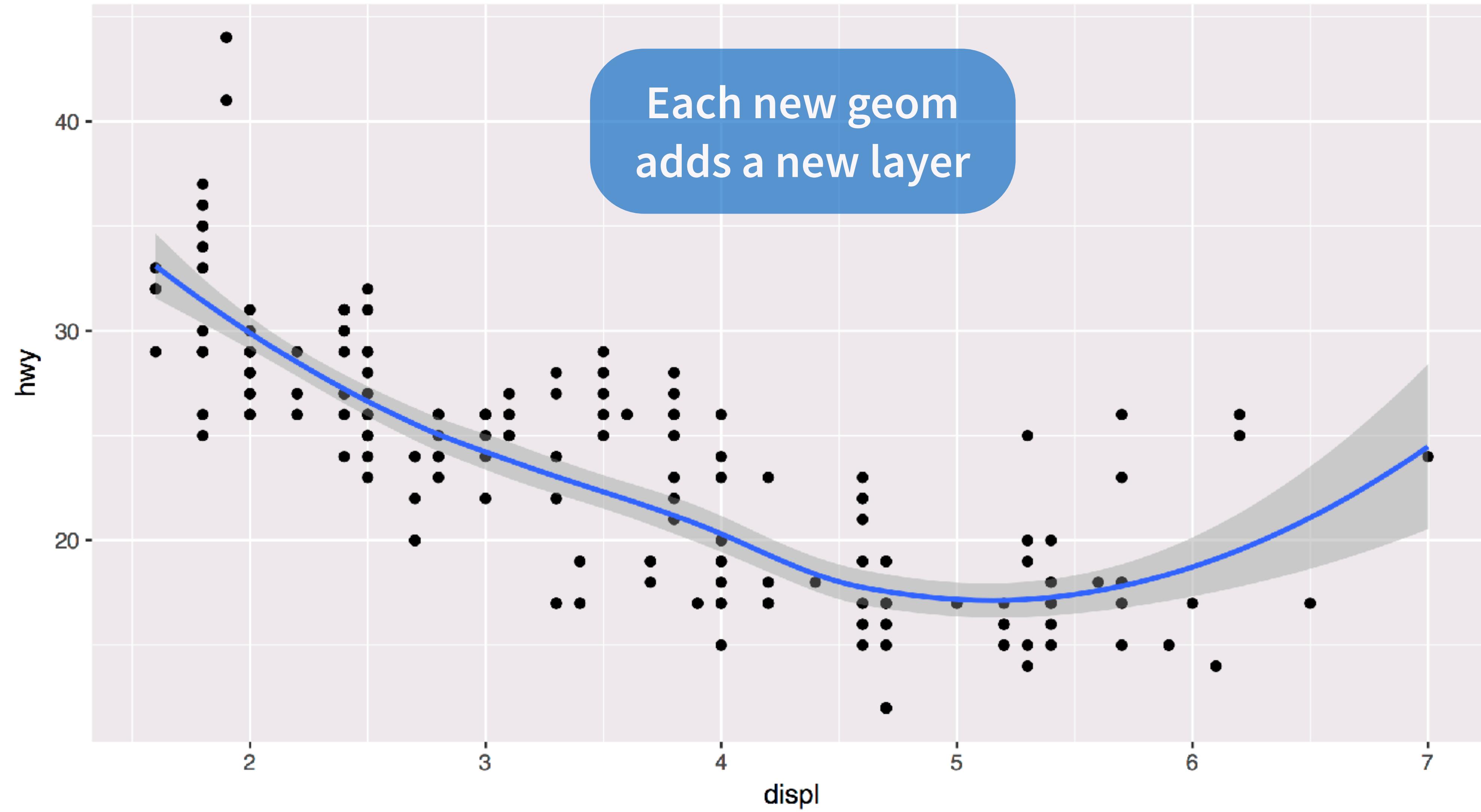
```
ggplot(mpg) +  
  geom_point(aes(displ, hwy)) +  
  geom_smooth(aes(displ, hwy))
```



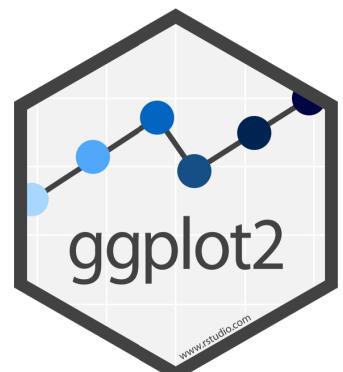


```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



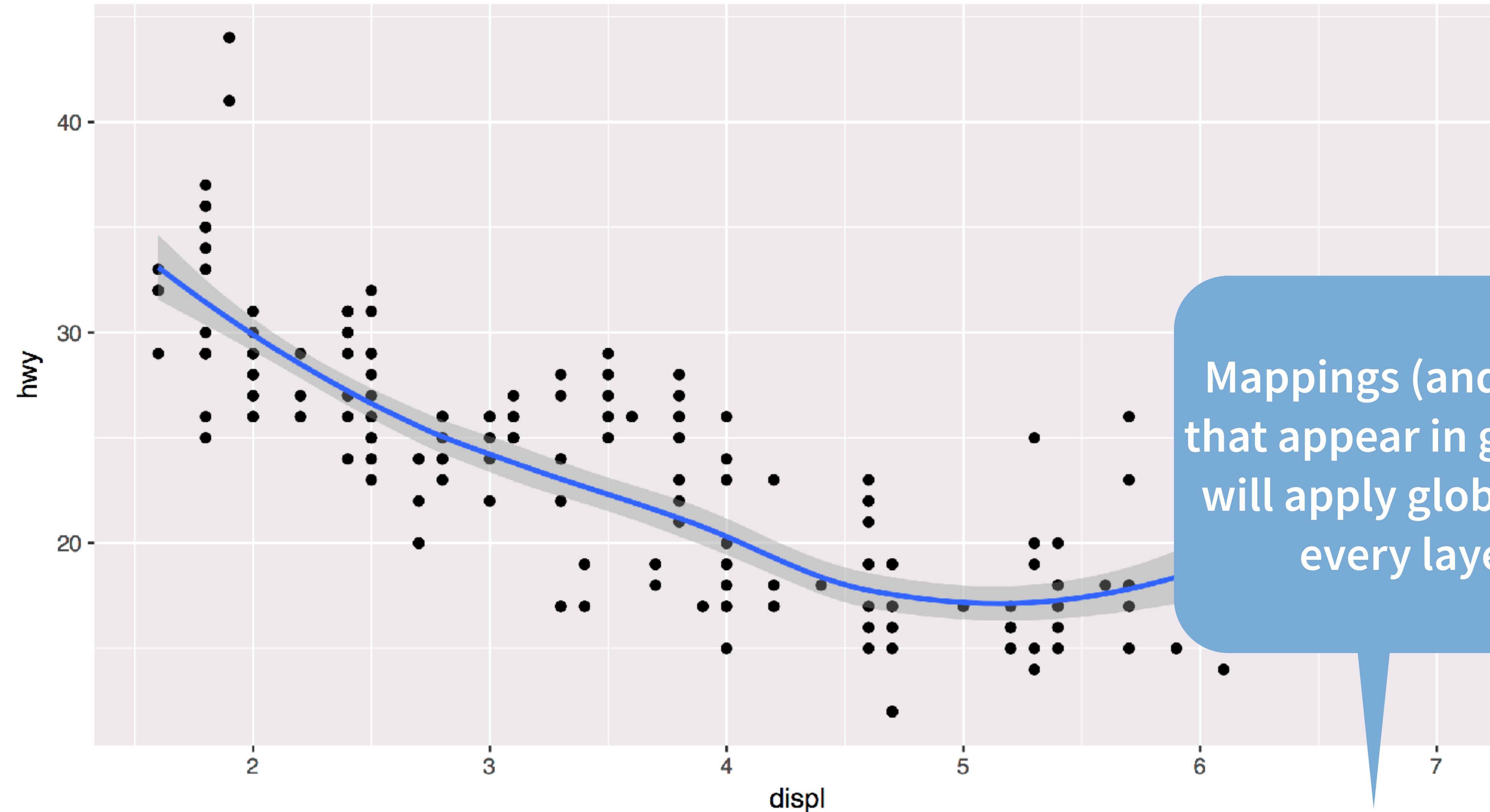


```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

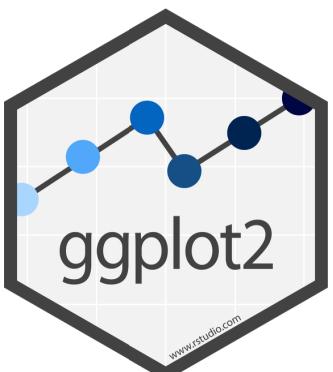


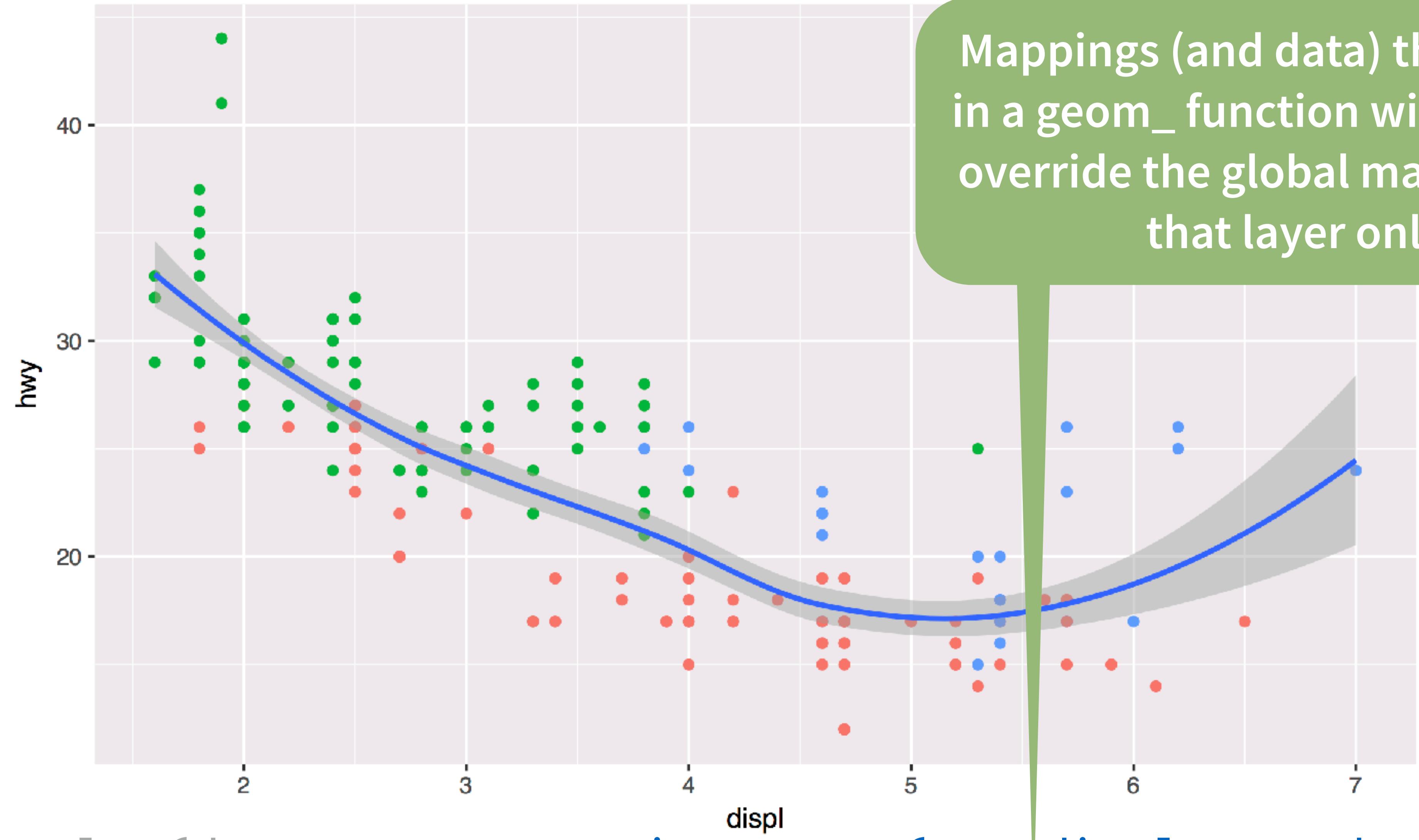
global vs. local

R

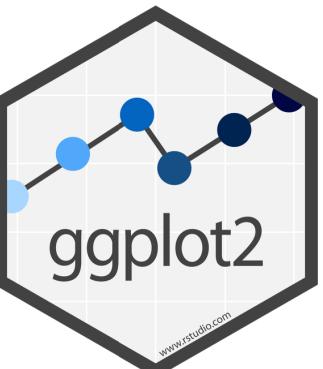


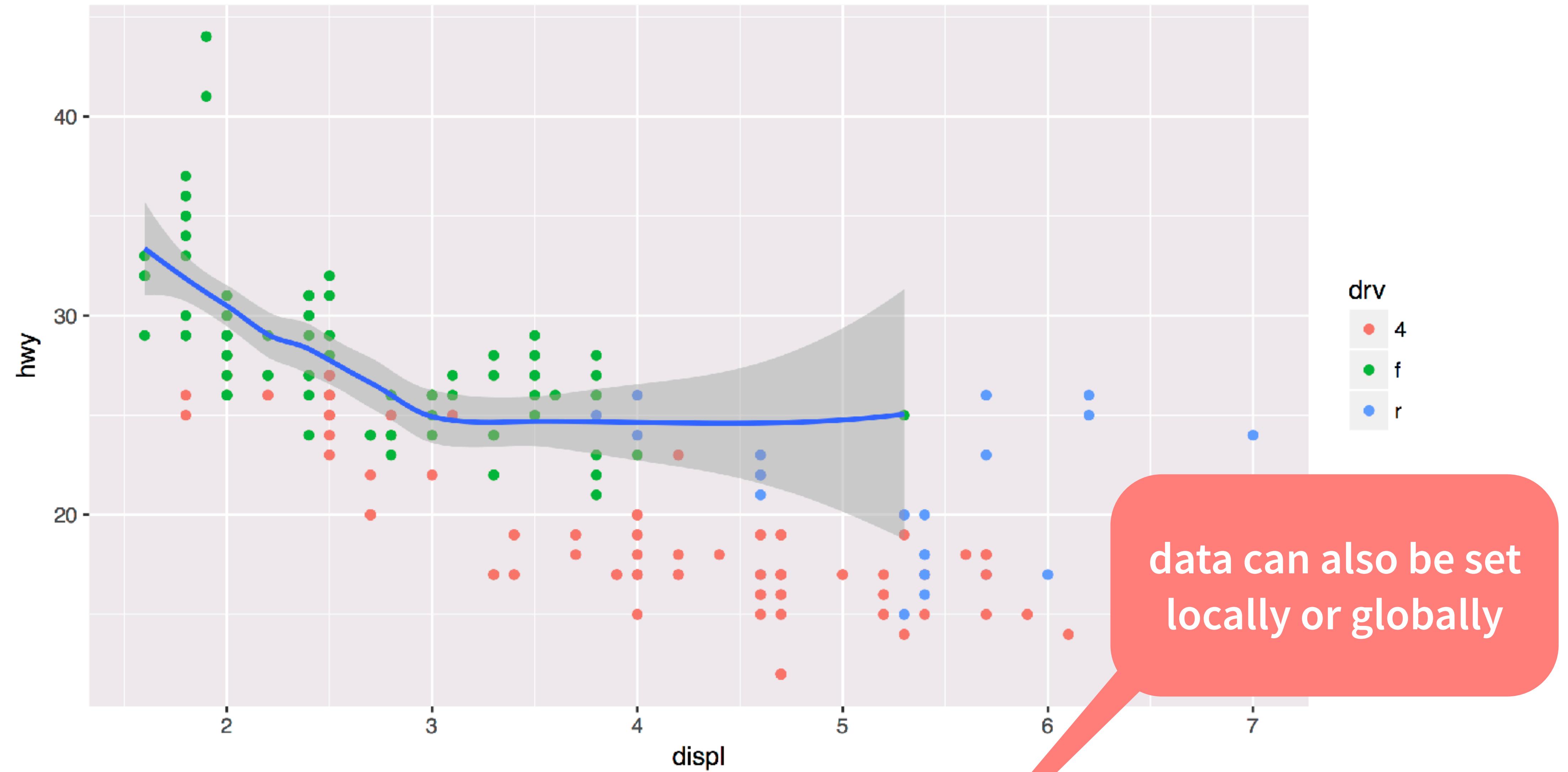
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth()
```



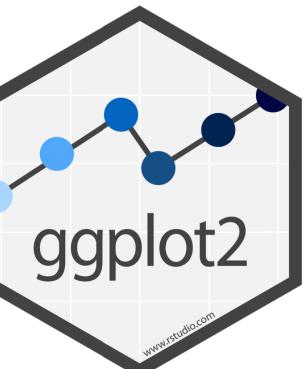


```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = drv)) +  
  geom_smooth()
```





```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = drv)) +  
  geom_smooth(data = filter(mpg, drv == "f"))
```



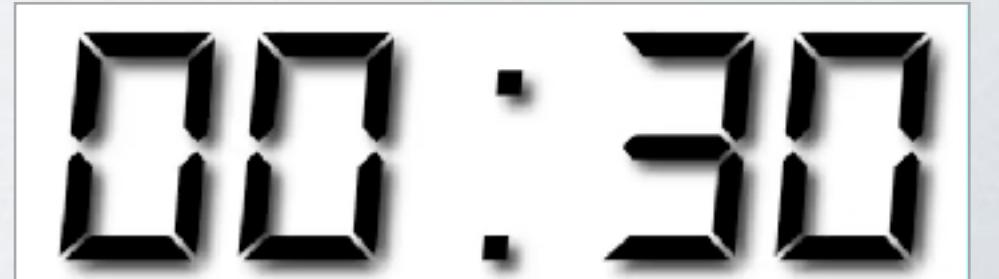
Saving graphs

R

Your Turn 8

What does this command return?

`getwd()`

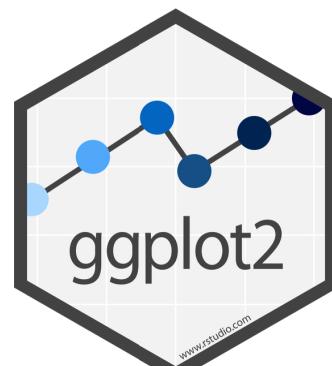
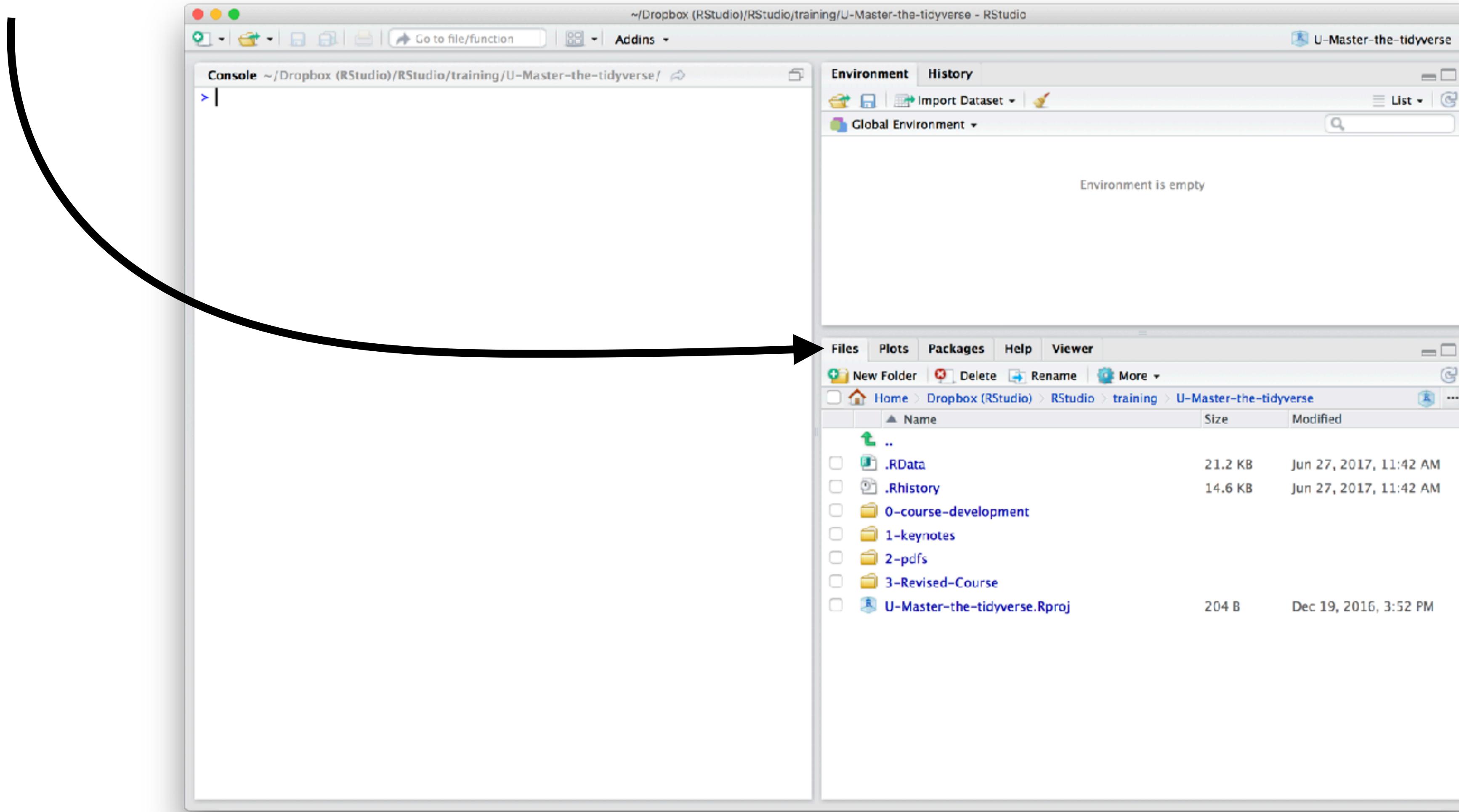


Working directory

R associates itself with a folder (i.e. directory) on your computer.

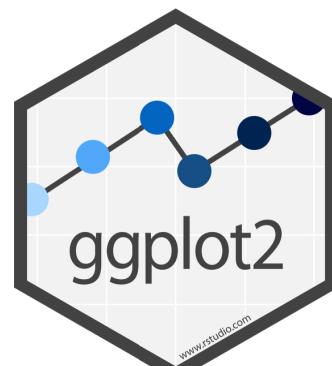
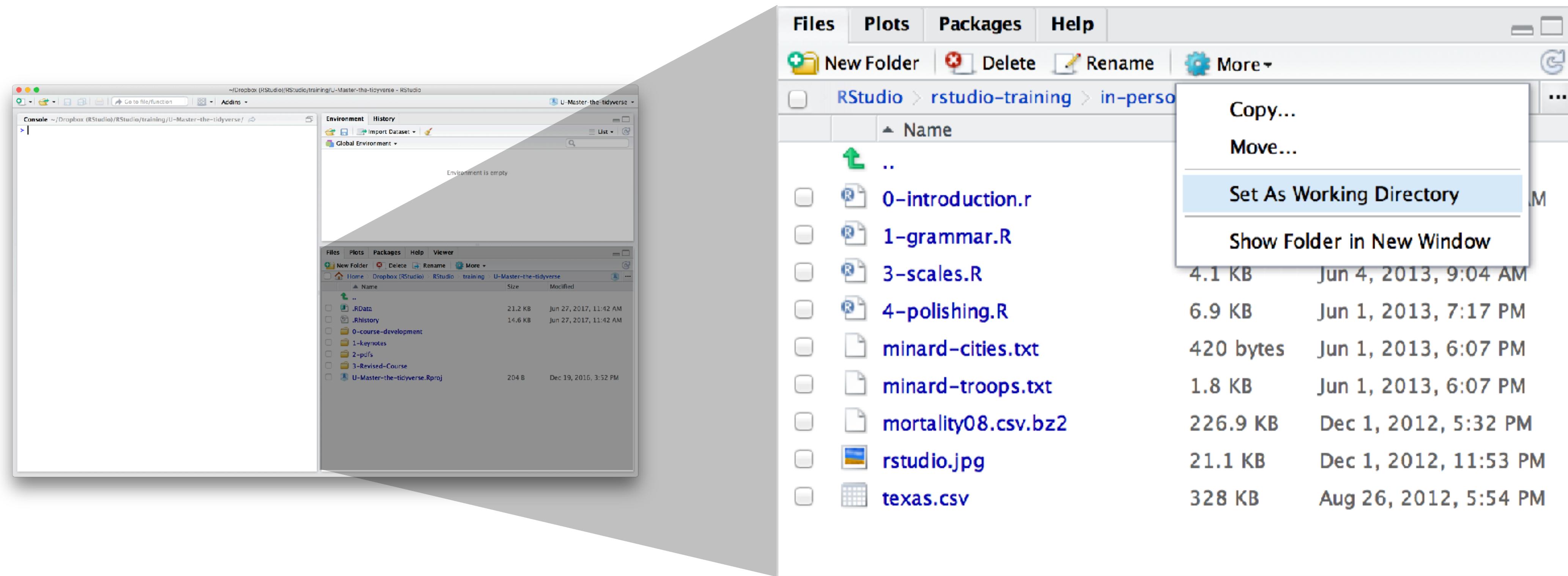
- This folder is known as your "**working directory**"
- When you save files, R will save them here
- When you load files, R will look for them here

The files pane of the IDE displays the contents of your working directory



Changing the Working directory

Navigate in the files pane to a new directory. Click
More > Set As Working Directory



Saving plots

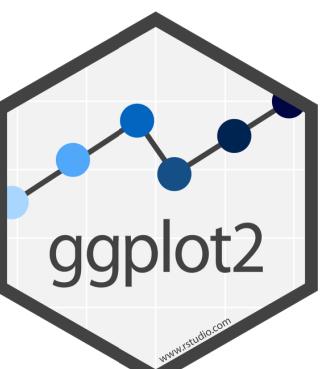
ggsave() saves the last plot.

Uses size on screen:

```
ggsave("my-plot.pdf")  
ggsave("my-plot.png")
```

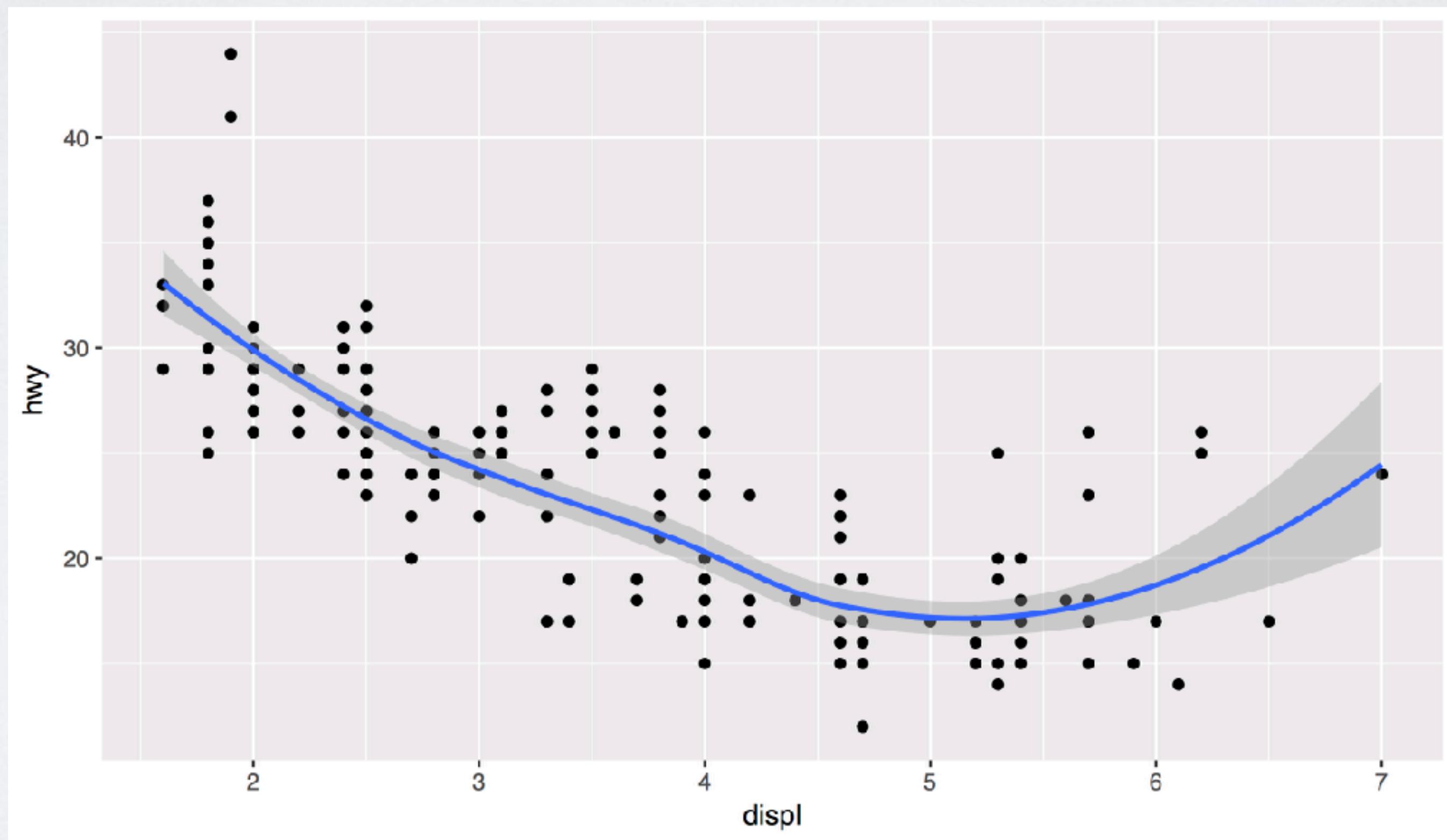
Specify size in inches

```
ggsave("my-plot.pdf", width = 6, height = 6)
```



Your Turn 9

Save your last plot and then locate it in your files pane.
(You may have to refresh the files list).



Grammar of Graphics



To make a graph

[template]

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

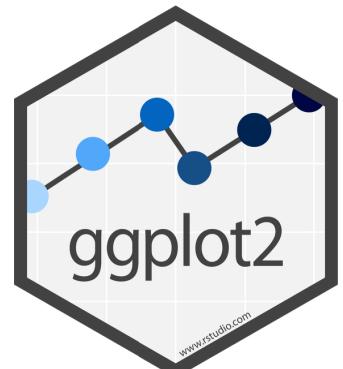
To make a graph

mpg	cyl	disp	hp
21.0	6	160.0	2
21.0	6	160.0	2
22.8	4	108.0	1
21.4	6	258.0	2
18.7	8	360.0	3
18.1	6	225.0	2
14.3	8	360.0	5
24.4	4	146.7	1
22.8	4	140.8	1
19.2	6	167.6	2
17.8	6	167.6	2
16.4	8	275.8	3
17.3	8	275.8	3
15.2	8	275.8	3
10.4	8	472.0	4
10.4	8	460.0	4
14.7	8	440.0	4
32.4	4	78.7	1
30.4	4	75.7	1
33.9	4	71.1	1

data

1. Pick a **data** set

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```



To make a graph

mpg	cyl	disp	hp	
21.0	6	160.0	2	●
21.0	6	160.0	2	●
22.8	4	108.0	1	●
21.4	6	258.0	2	●
18.7	8	360.0	3	●
18.1	6	225.0	2	●
14.3	8	360.0	5	●
24.4	4	146.7	1	●
22.8	4	140.8	1	●
19.2	6	167.6	2	●
17.8	6	167.6	2	●
16.4	8	275.8	3	●
17.3	8	275.8	3	●
15.2	8	275.8	3	●
10.4	8	472.0	4	●
10.4	8	460.0	4	●
14.7	8	440.0	4	●
32.4	4	78.7	1	●
30.4	4	75.7	1	●
33.9	4	71.1	1	●

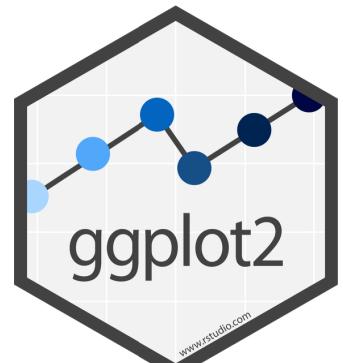
data

geom

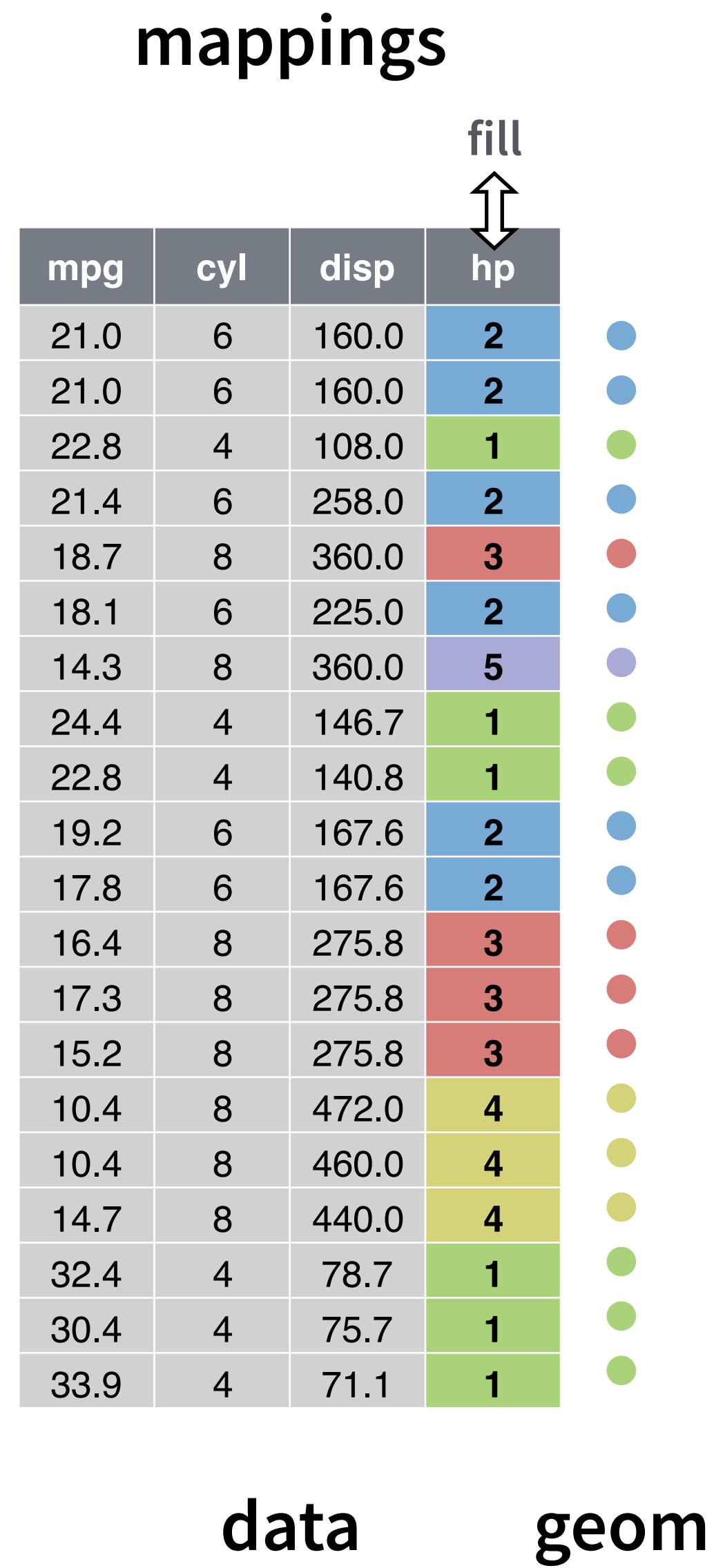
1. Pick a **data** set

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

2. Choose a **geom**
to display cases



To make a graph

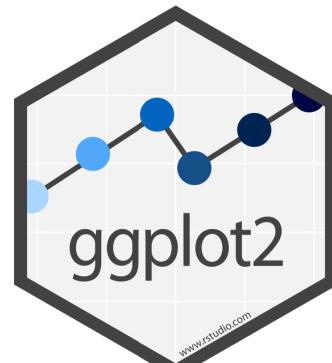


1. Pick a **data** set

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

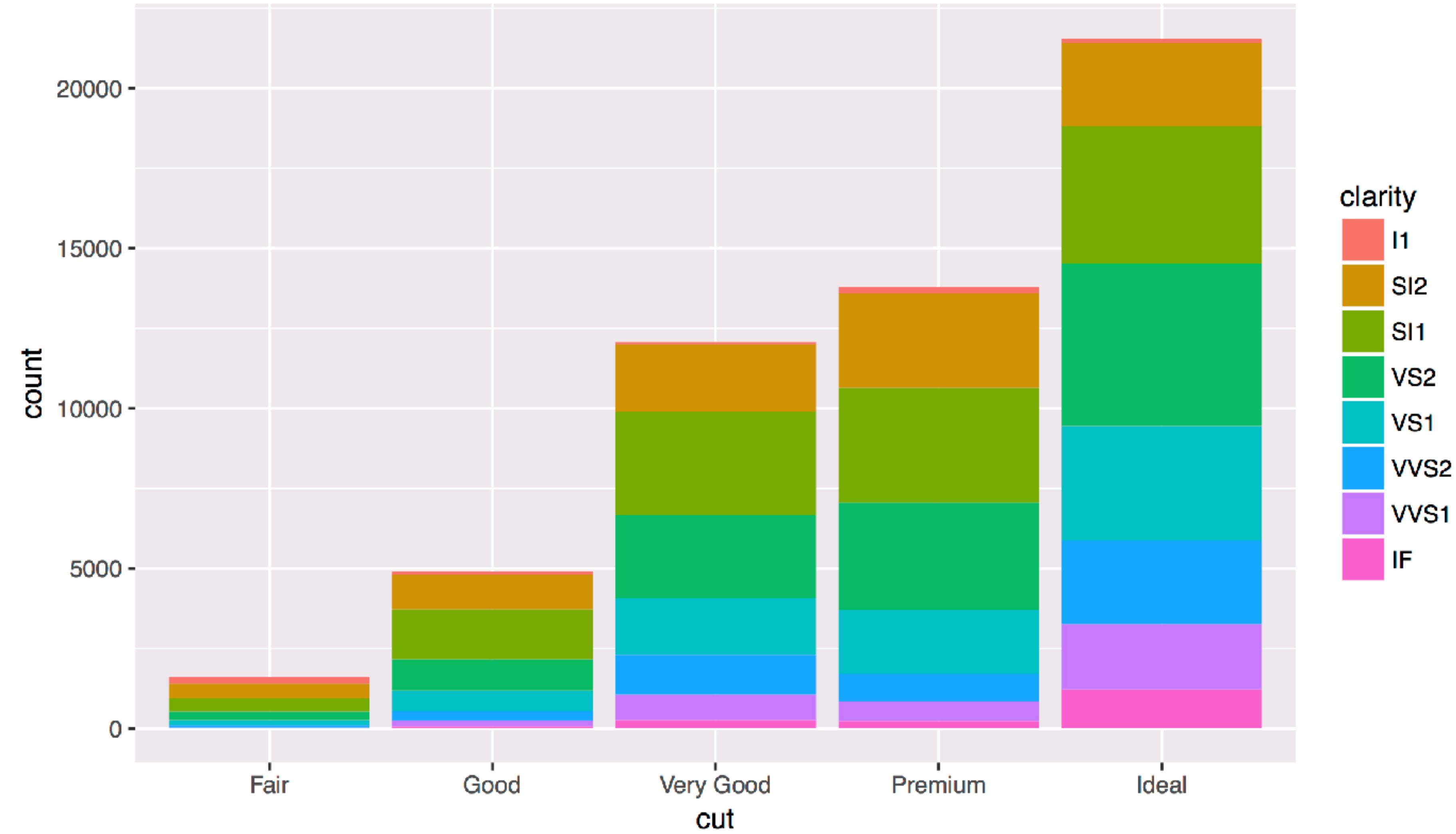
2. Choose a **geom**
to display cases

3. **Map** aesthetic
properties to
variables

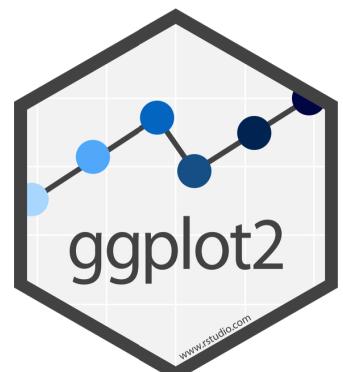
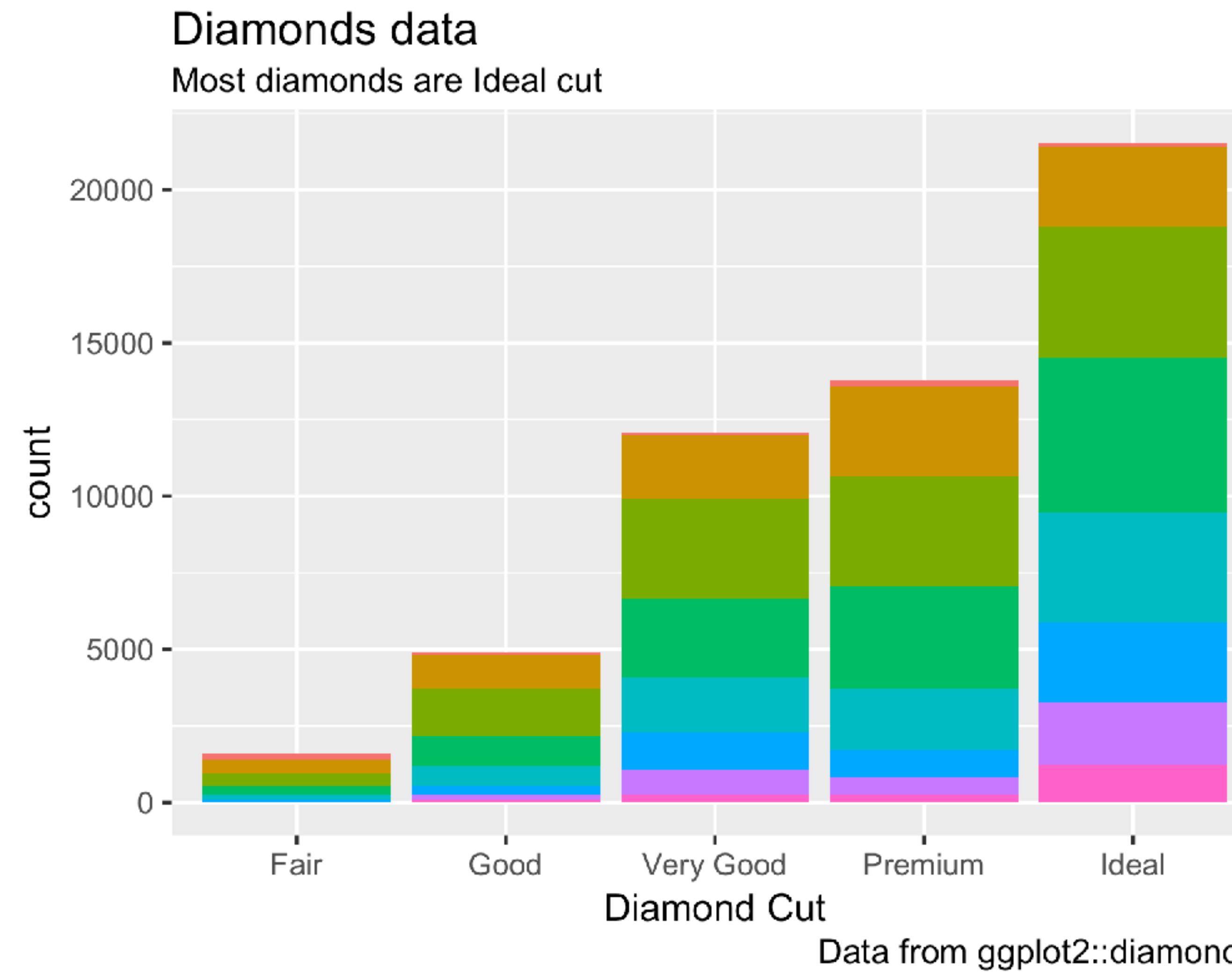


what else?

R

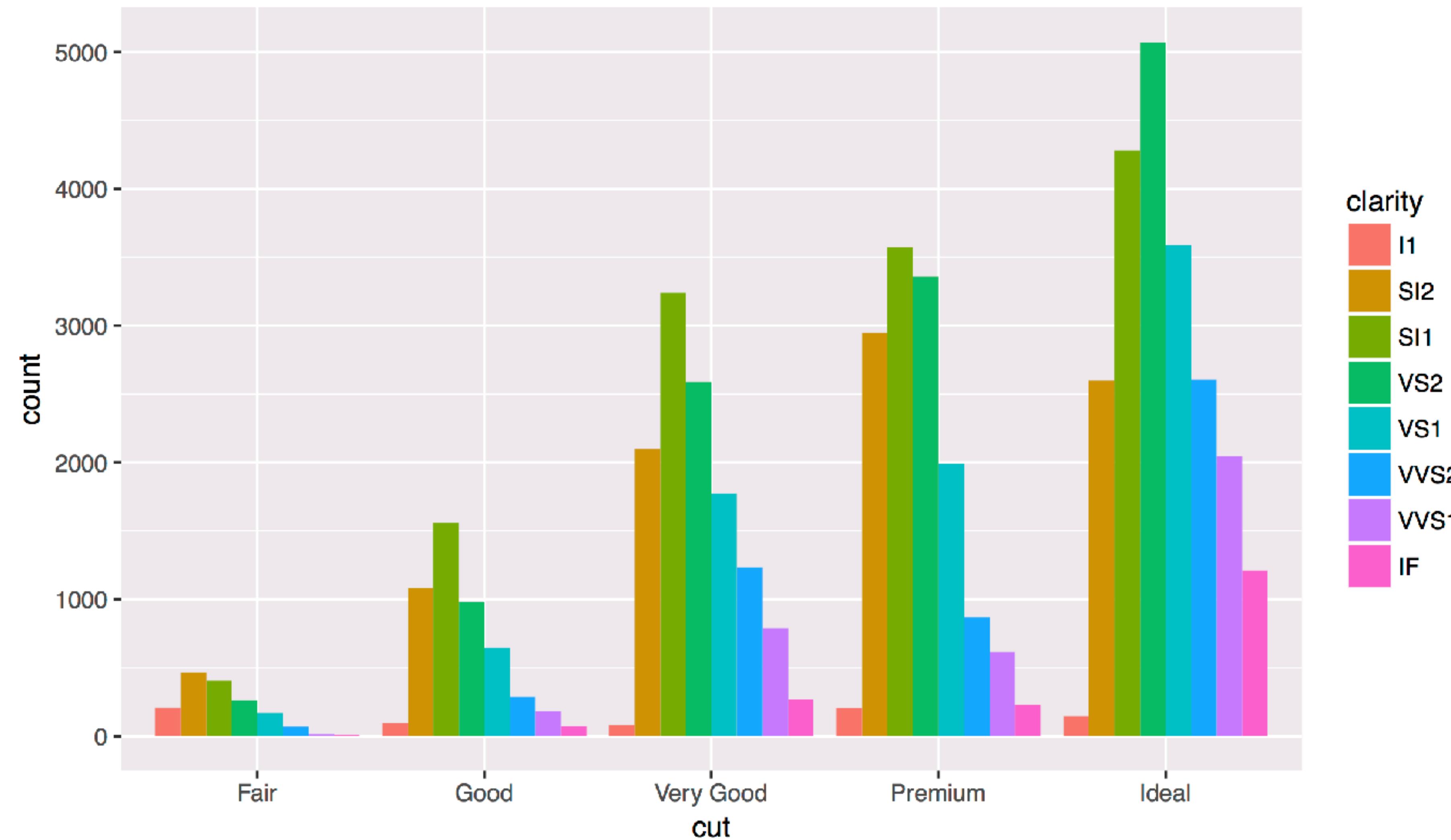


Titles and captions



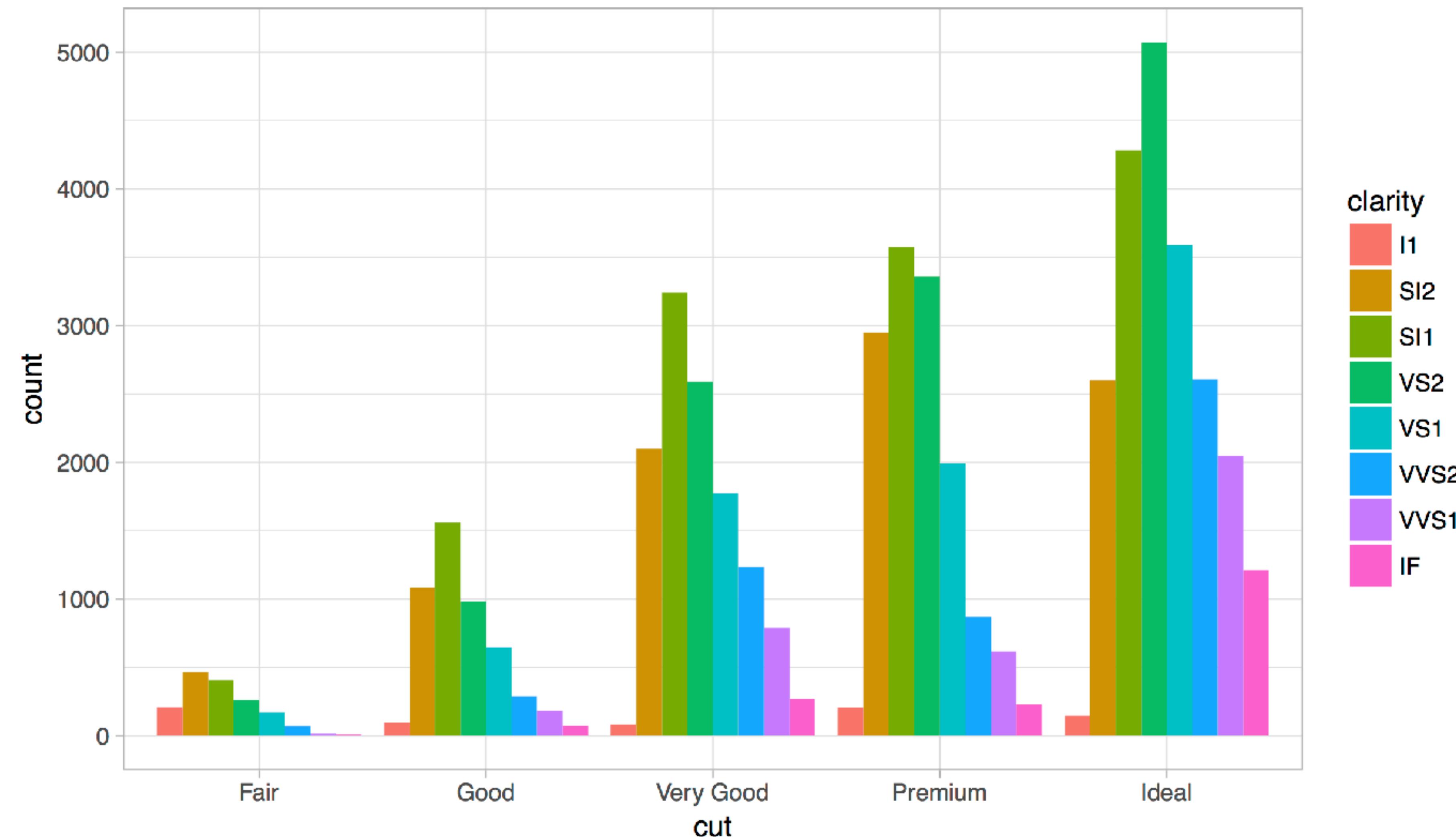
Position Adjustments

How overlapping objects are arranged



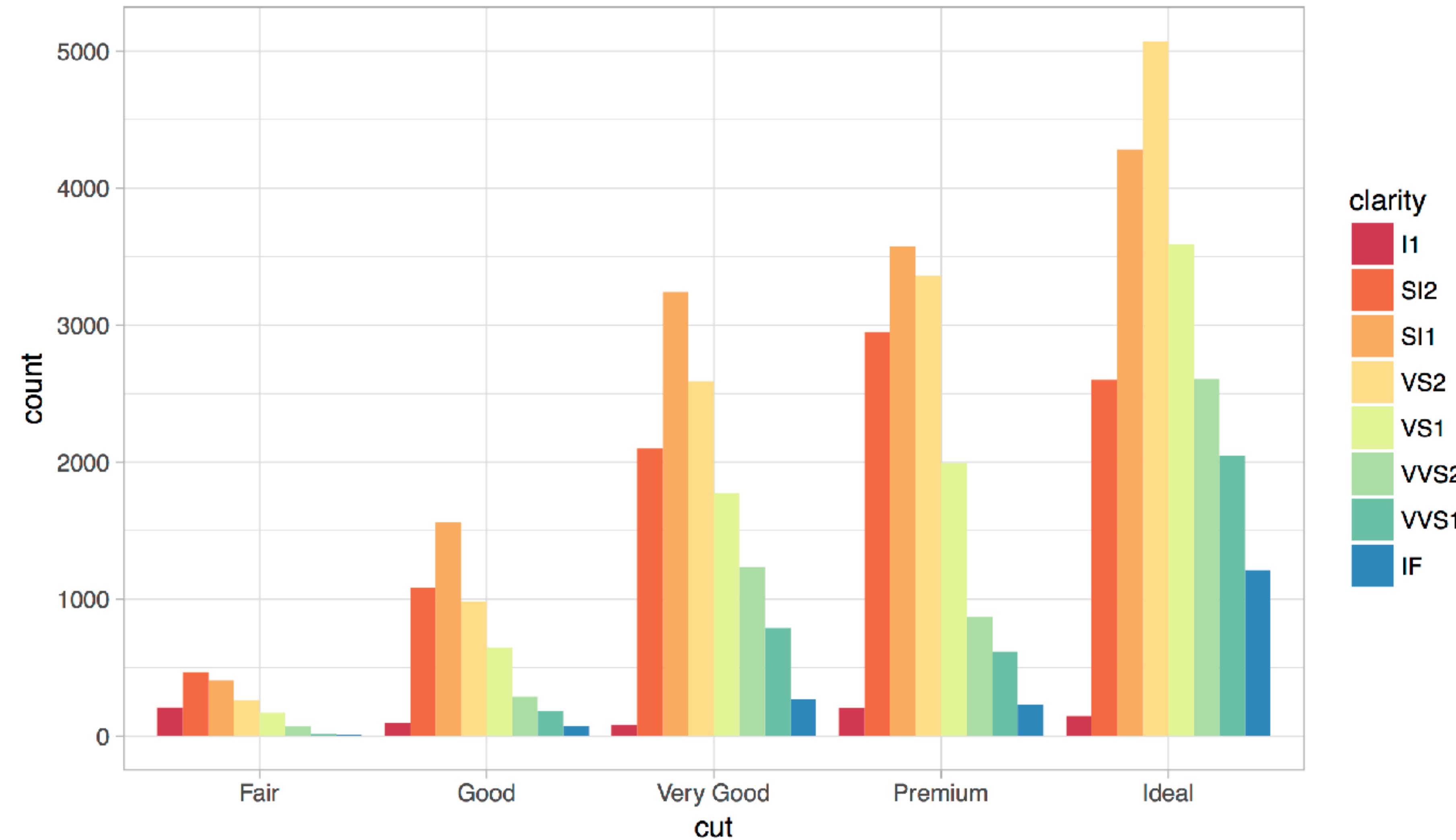
Themes

Visual appearance of non-data elements



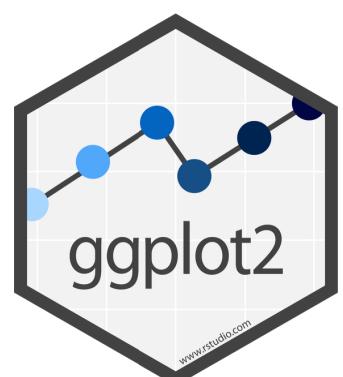
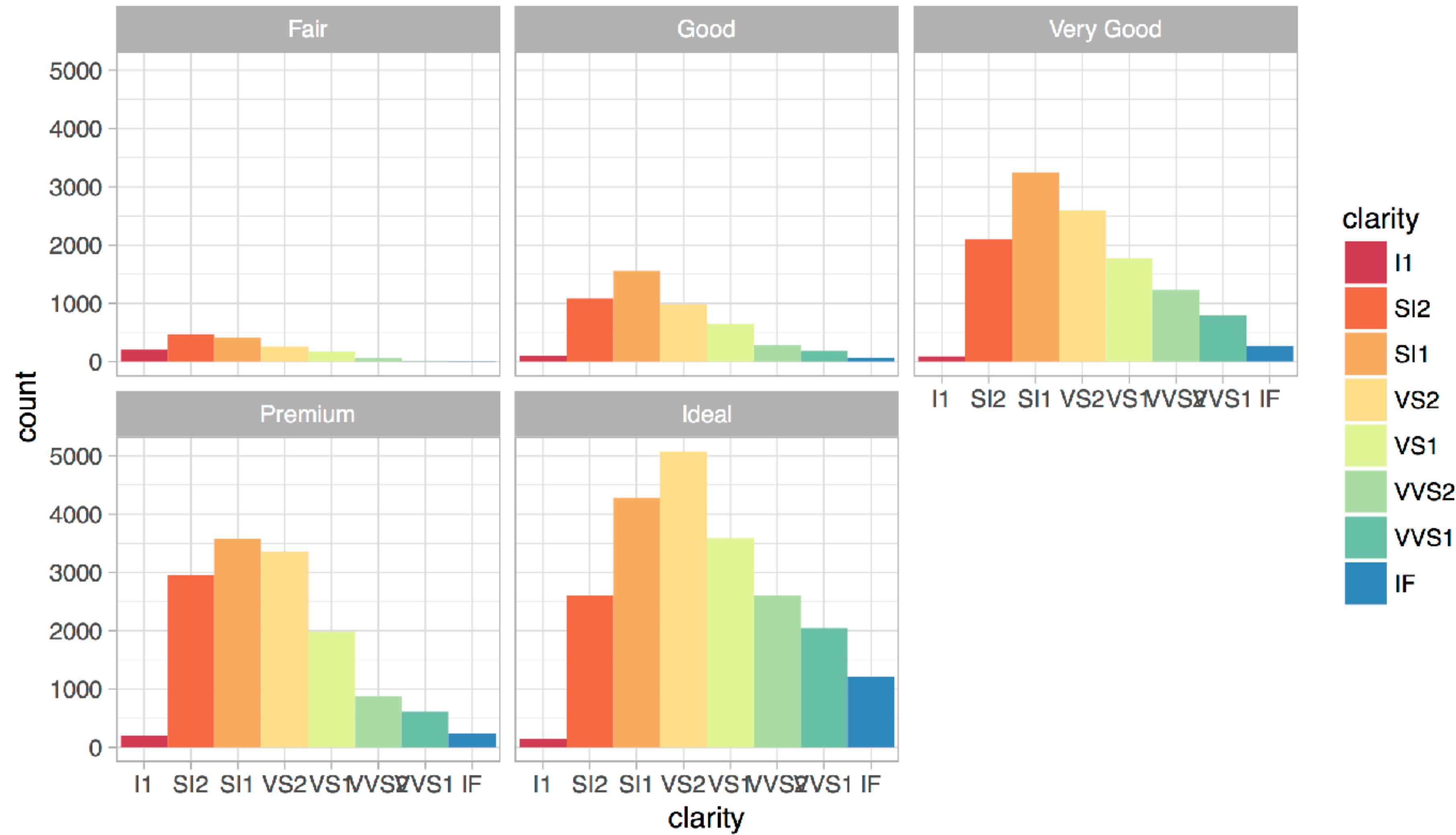
Scales

Customize color scales, other mappings



Facets

Subplots that display subsets of the data.



Coordinate systems



A ggplot2 template

Make any plot by filling in the parameters of this template

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

Visualization with ggplot2 :: CHEAT SHEET

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))
```

TWO VARIABLES

continuous x , continuous y

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

continuous bivariate distribution

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

continuous function

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

visualizing error

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

maps

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

THREE VARIABLES

```
sealsSz <- with(seals, sqrt(delta_long^2 + delta_lat^2))
```



RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1232 • rstudio.com • Learn more at <http://ggplot2.tidyverse.org> • ggplot2 2.1.0 • Updated: 2016-11

ggplot2.tidyverse.org

The screenshot shows a web browser displaying the ggplot2.tidyverse.org website. The page title is "Create Elegant Data Visualisation" and the user is logged in as "Garrett". The URL in the address bar is "ggplot2.tidyverse.org". The main content area features the ggplot2 logo and the text "part of the tidyverse". Below this, a section titled "Usage" contains a brief explanation of how ggplot2 works and a code snippet demonstrating its usage with the mpg dataset.

Usage

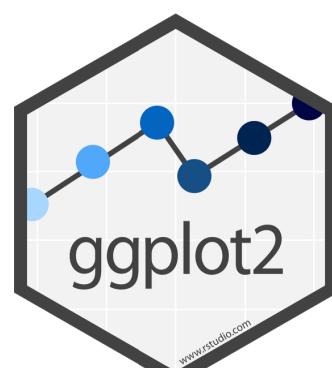
It's hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation. However, in most cases you start with `ggplot()`, supply a dataset and aesthetic mapping (with `aes()`). You then add on layers (like `geom_point()` or `geom_histogram()`), scales (like `scale_colour_brewer()`), faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```

The page also includes a "Links" sidebar with links to CRAN, GitHub source code, reporting bugs, and learning more. A "License" section links to the GPL-2 license file. A "Developers" section lists Hadley Wickham as the author/maintainer. At the bottom right is the ggplot2 logo.

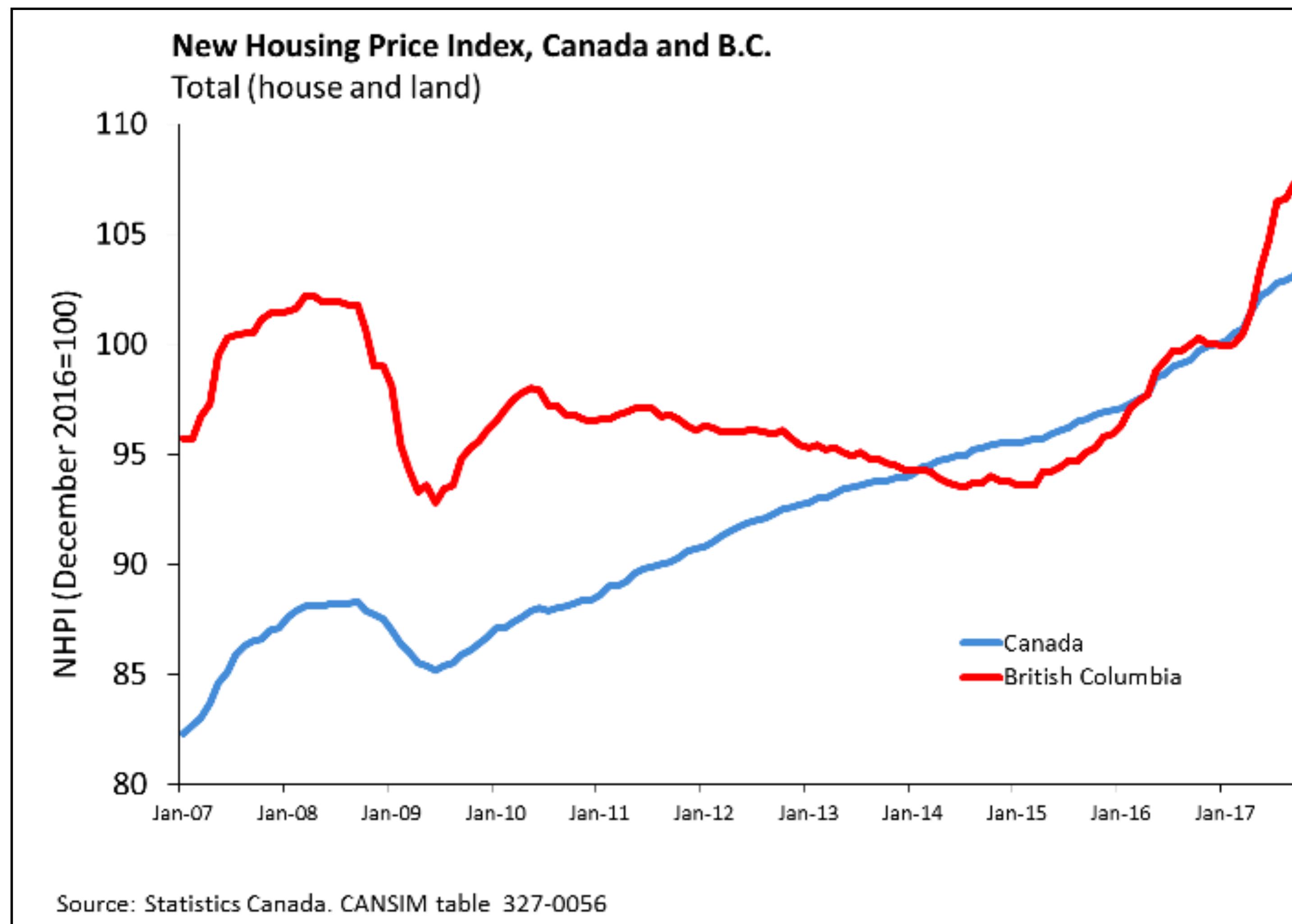
CC by RStudio



Project

New Housing Price Index

<https://www2.gov.bc.ca/gov/content/data/statistics/infoline/infoline-2017/17-146-price-new-housing>



Your Turn

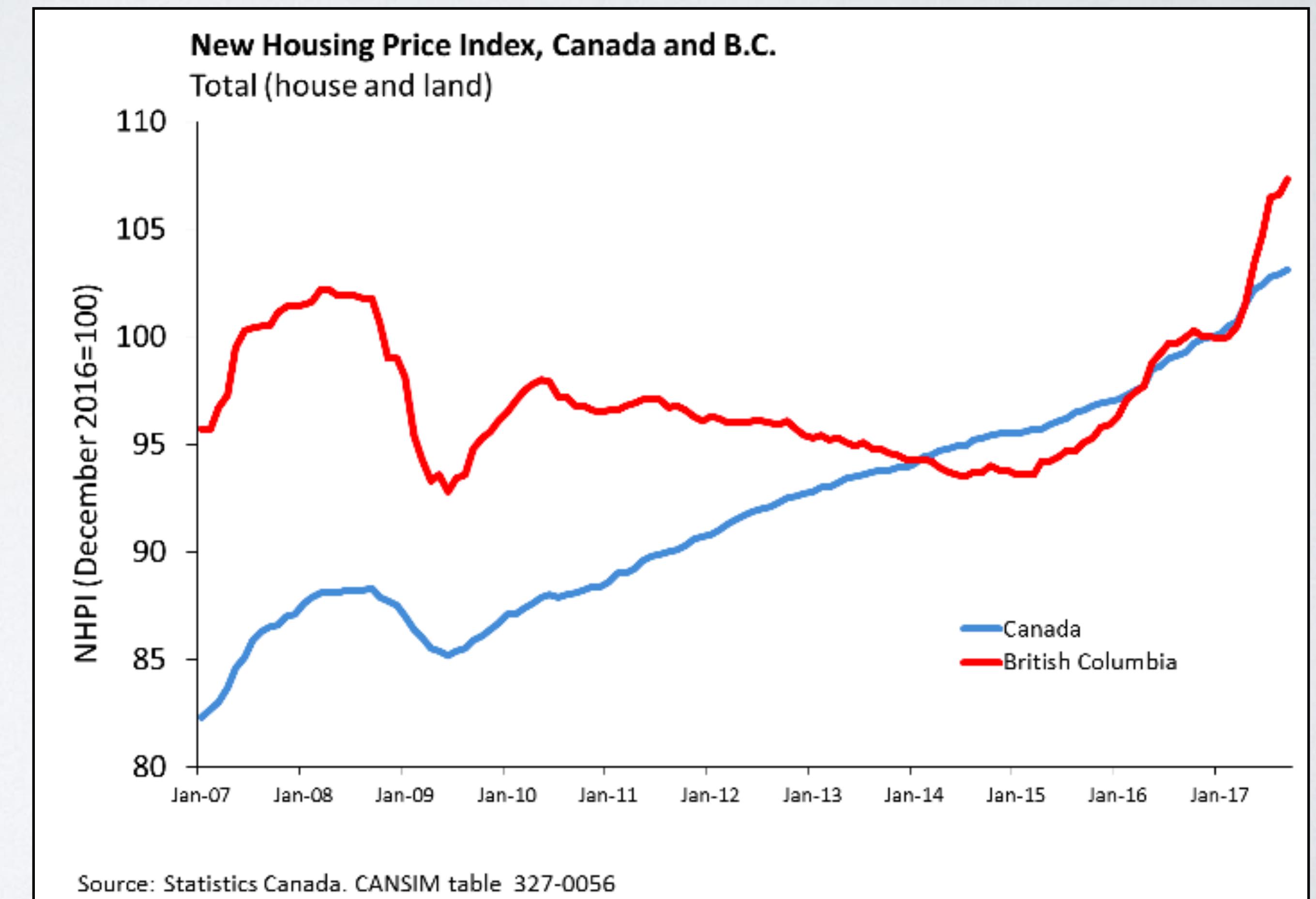
Save your open files.

Navigate into the project_housing directory in the Files pane.

Click on project-housing.Rproj

Your Turn

1. Open 01-visualize.Rmd
2. Take a look at housing
3. Create a visualization that imitates this one:



Your Turn

Switch back to the
r_intro_bc_stats project.

Open **04-Transform-Data.Rmd**.



Visualize Data with

