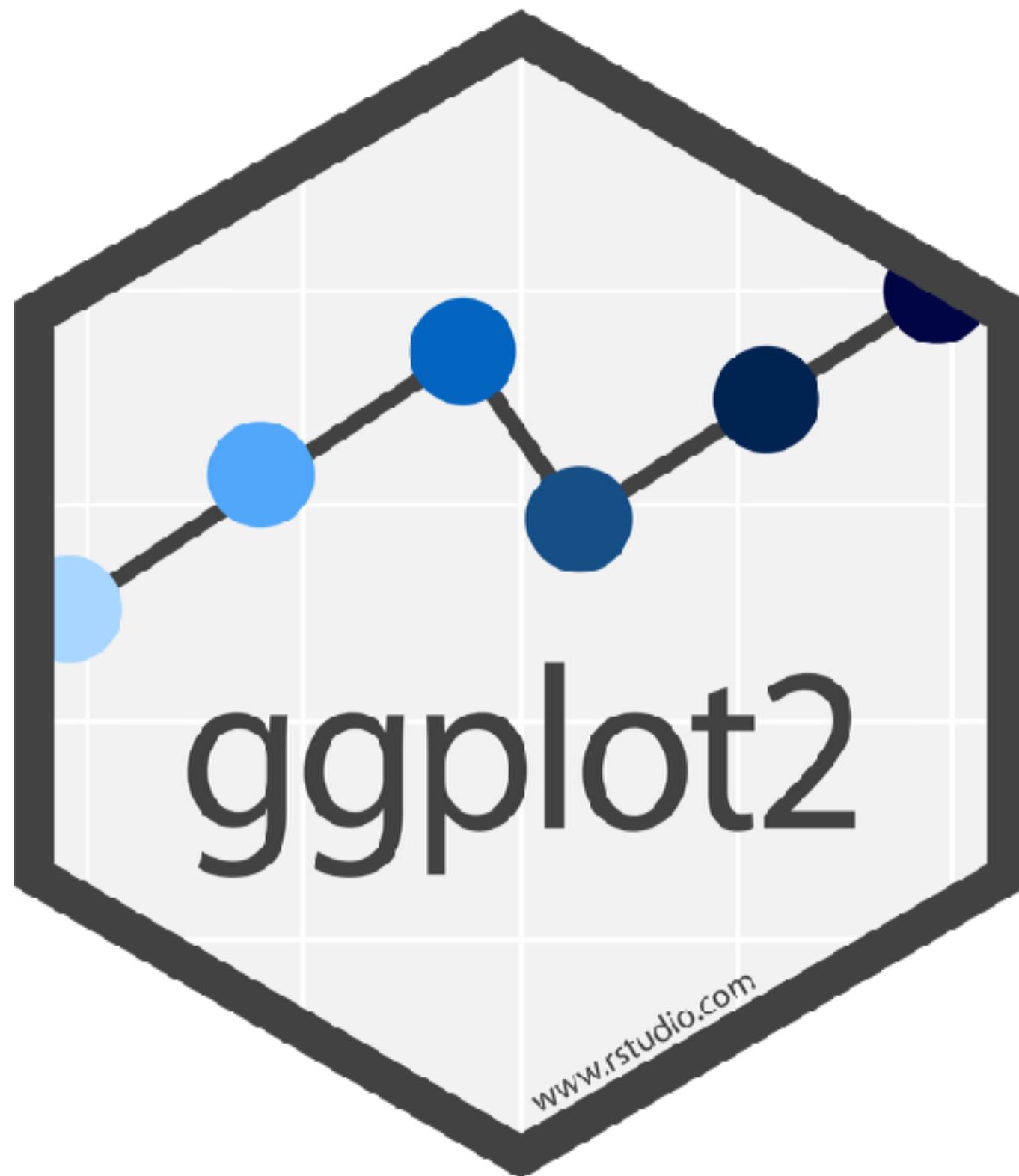


Visualize Data with



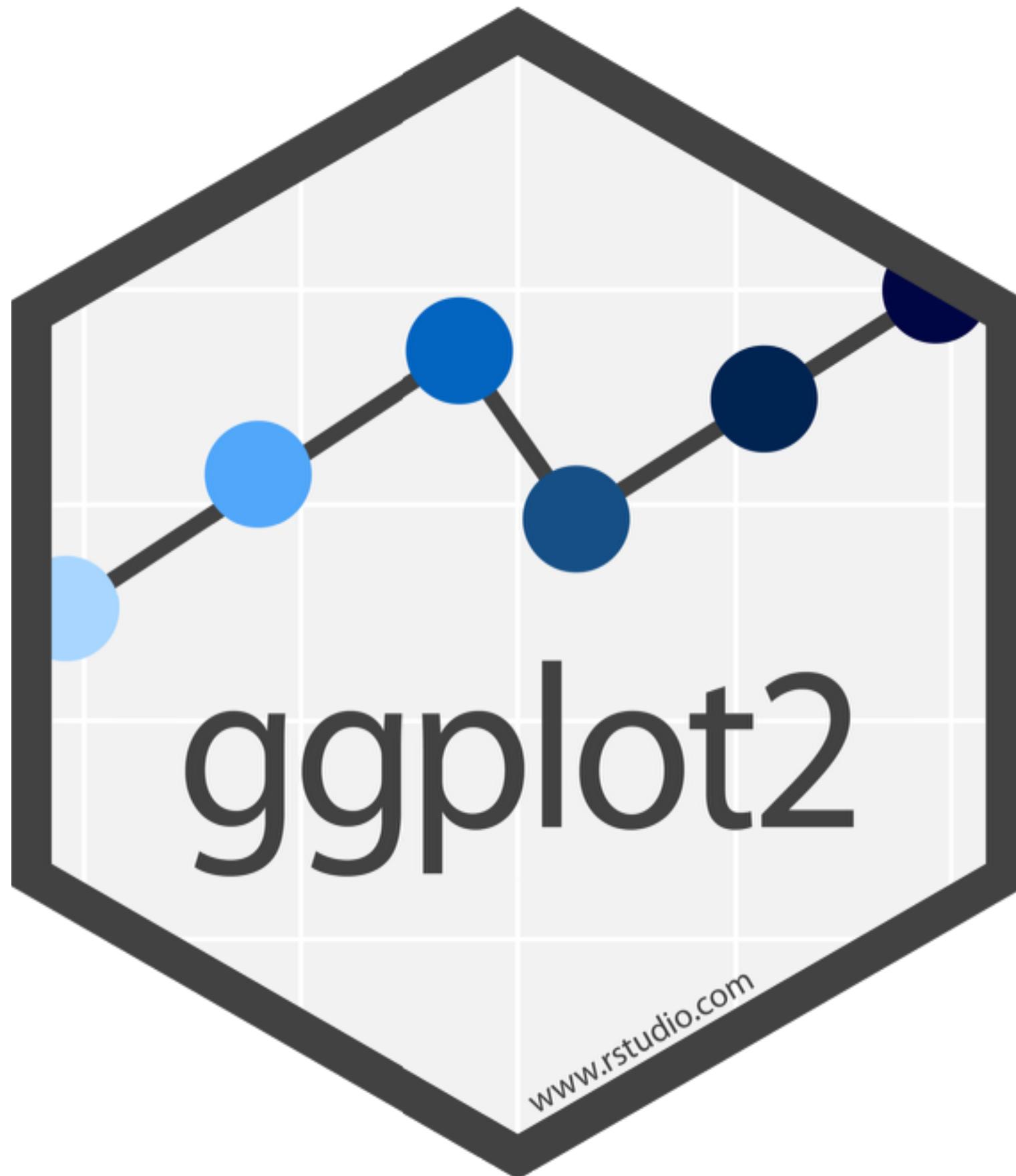
In R4DS

Visualizing Data

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

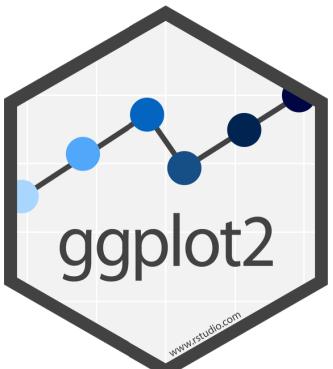
- John Tukey

ggplot2



One of the earliest members of the tidyverse.

Complicated plots, come from combining simple components.



Your Turn

Open 01-Visualize.Rmd



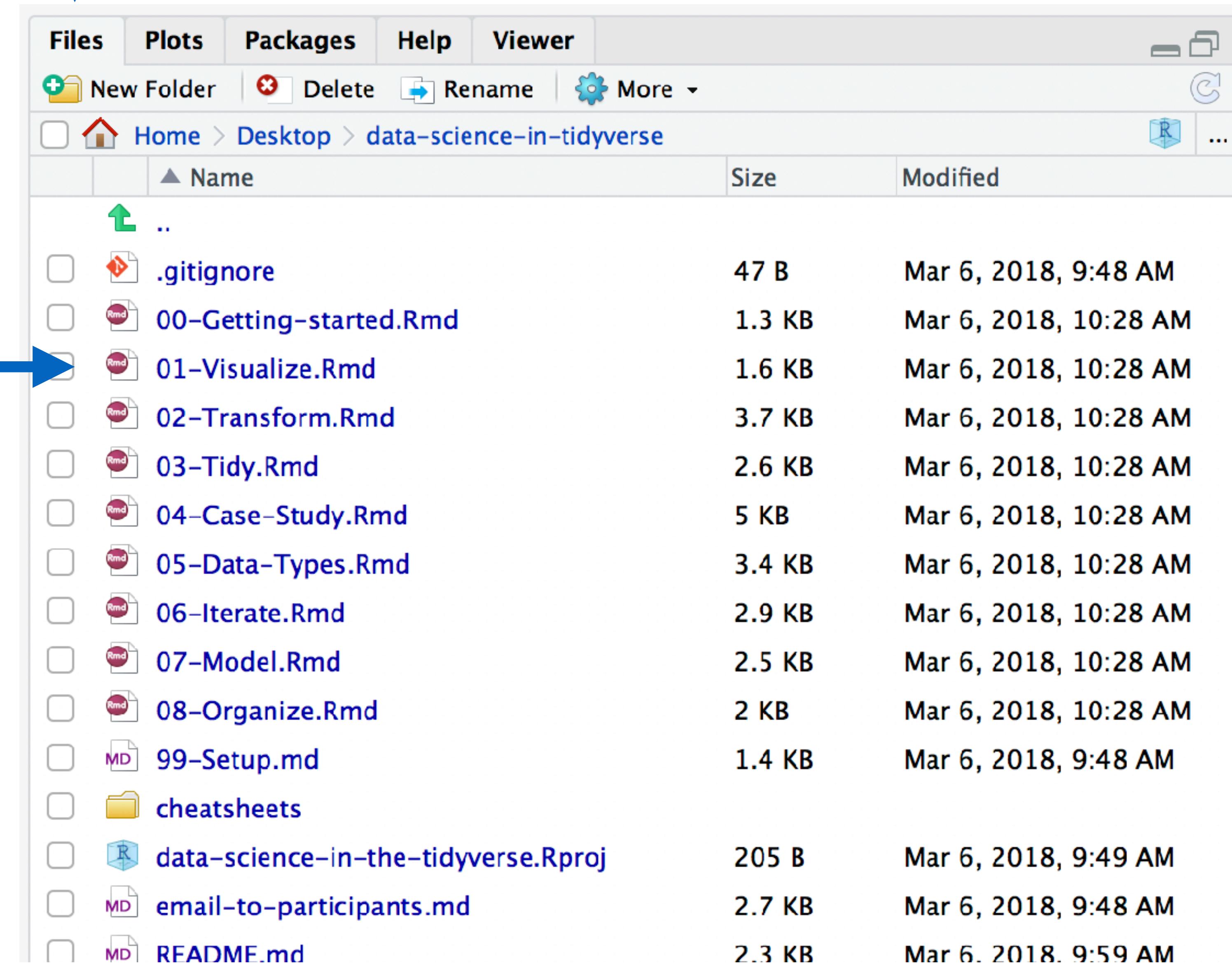
I'm working on it



I'm stuck!



I'm done!



The screenshot shows the RStudio interface with the 'Files' tab selected. The current working directory is 'Home > Desktop > data-science-in-tidyverse'. The file browser lists the following files and folders:

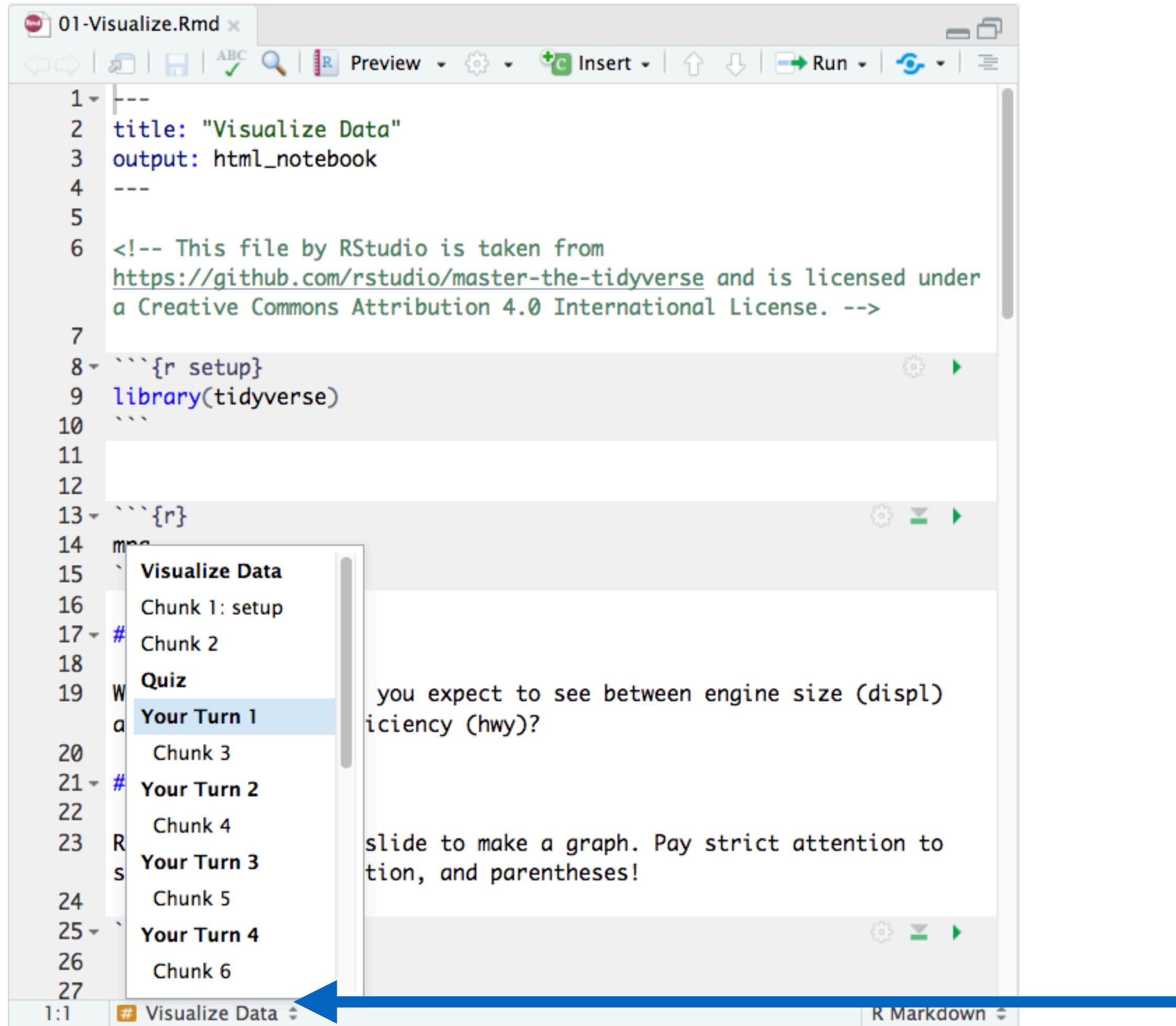
	Name	Size	Modified
	..		
	.gitignore	47 B	Mar 6, 2018, 9:48 AM
	00-Getting-started.Rmd	1.3 KB	Mar 6, 2018, 10:28 AM
	01-Visualize.Rmd	1.6 KB	Mar 6, 2018, 10:28 AM
	02-Transform.Rmd	3.7 KB	Mar 6, 2018, 10:28 AM
	03-Tidy.Rmd	2.6 KB	Mar 6, 2018, 10:28 AM
	04-Case-Study.Rmd	5 KB	Mar 6, 2018, 10:28 AM
	05-Data-Types.Rmd	3.4 KB	Mar 6, 2018, 10:28 AM
	06-Iterate.Rmd	2.9 KB	Mar 6, 2018, 10:28 AM
	07-Model.Rmd	2.5 KB	Mar 6, 2018, 10:28 AM
	08-Organize.Rmd	2 KB	Mar 6, 2018, 10:28 AM
	99-Setup.md	1.4 KB	Mar 6, 2018, 9:48 AM
	cheatsheets		
	data-science-in-the-tidyverse.Rproj	205 B	Mar 6, 2018, 9:49 AM
	email-to-participants.md	2.7 KB	Mar 6, 2018, 9:48 AM
	README.md	2.3 KB	Mar 6, 2018, 9:59 AM

If you get lost or need to restart

```
01-Visualize.Rmd x
1 title: "Visualize Data"
2 output: html_notebook
3 ---
4 <!-- This file by RStudio is taken from
5 https://github.com/rstudio/master-the-tidyverse and is licensed under
6 a Creative Commons Attribution 4.0 International License. -->
7
8 ``{r setup}
9 library(tidyverse)
10 ...
11
12
13 ``{r}
14 mpg
15 ...
16
17 ## Quiz
18
19 What relationship do you expect to see between engine size (displ)
20 and highway fuel efficiency (hwy)?
21
22 ## Your Turn 1
23
24 Run the code on the slide to make a graph. Pay strict attention to
25 spelling, capitalization, and parentheses!
26
27 ``{r}
R Markdown
```

Check you are in the right file

If you get lost or need to restart

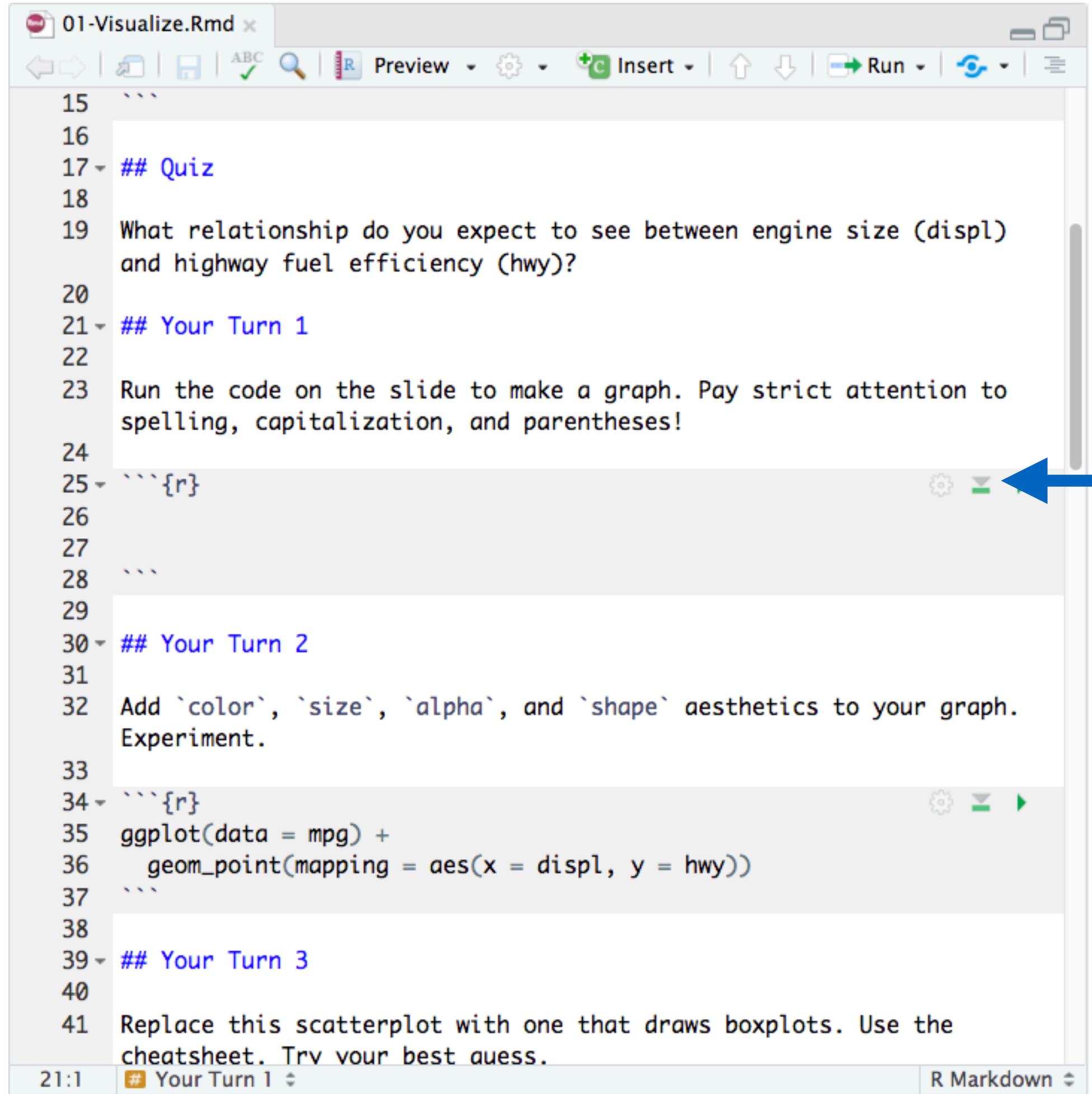


A screenshot of the RStudio interface. The top bar shows the file "01-Visualize.Rmd". The code editor on the left contains R Markdown code, including sections like "Visualize Data", "Quiz", and several "Your Turn" sections. A blue arrow points from the text "Use the section browser to quickly navigate to the right *Your Turn*" to the section browser panel on the right. The section browser lists sections such as "Visualize Data", "Quiz", and "Your Turn 1" through "Your Turn 4". The "Your Turn 1" section is currently selected. The main pane displays a question about engine size and fuel efficiency.

```
1 ---  
2 title: "Visualize Data"  
3 output: html_notebook  
4 ---  
5  
6 <!-- This file by RStudio is taken from  
7 https://github.com/rstudio/master-the-tidyverse and is licensed under  
8 a Creative Commons Attribution 4.0 International License. -->  
9  
10 ````{r setup}  
11 library(tidyverse)  
12 ````  
13 ````{r}  
14 mma  
15 `Visualize Data`  
16 Chunk 1: setup  
17 # Chunk 2  
18 Quiz  
19 W  
a Your Turn 1  
20 # Your Turn 2  
21 # Your Turn 3  
22 R  
s Your Turn 4  
23 C  
hunk 5  
24 Your Turn 5  
25 Your Turn 6  
26 Your Turn 7  
27 Chunk 6
```

Use the section browser to quickly navigate to the right *Your Turn*

If you get lost or need to restart



```
15 ``
16 
17 ## Quiz
18 
19 What relationship do you expect to see between engine size (displ) and highway fuel efficiency (hwy)?
20 
21 ## Your Turn 1
22 
23 Run the code on the slide to make a graph. Pay strict attention to spelling, capitalization, and parentheses!
24 
25 ``{r}
26 
27 
28 ``
29 
30 ## Your Turn 2
31 
32 Add `color`, `size`, `alpha`, and `shape` aesthetics to your graph. Experiment.
33 
34 ``{r}
35 ggplot(data = mpg) +
36   geom_point(mapping = aes(x = displ, y = hwy))
37 ```
38 
39 ## Your Turn 3
40 
41 Replace this scatterplot with one that draws boxplots. Use the cheatsheet. Try your best guess.
```

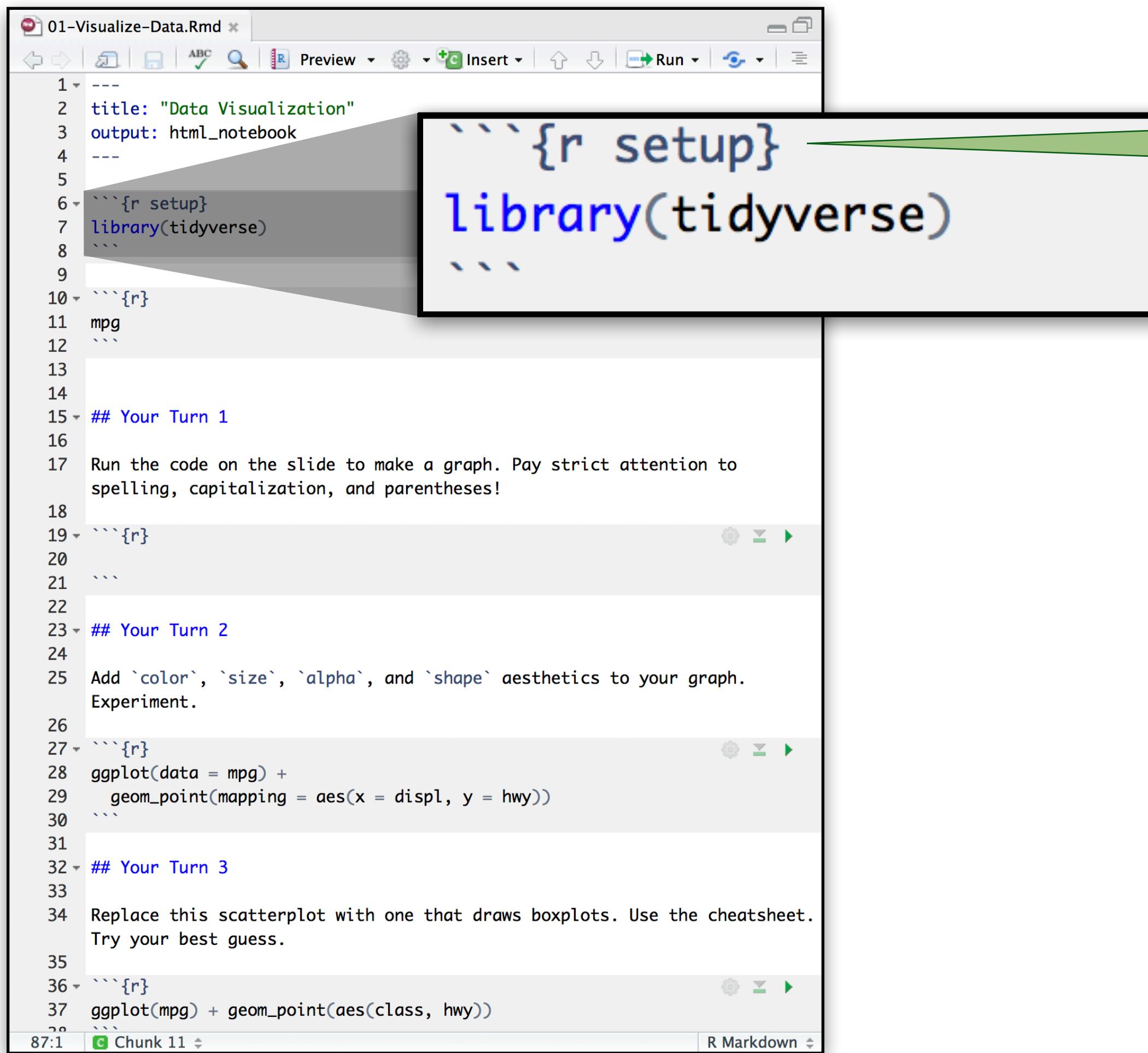
Click to run all
chunks before this
one.

You should be ready
to go.



Setup

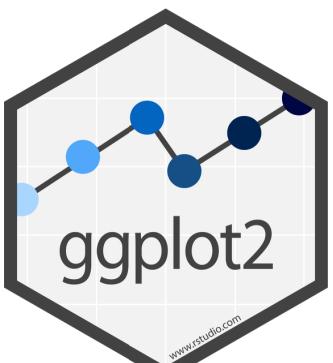
The setup chunk is always run once before anything else



A screenshot of an RStudio interface showing an R Markdown document 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  
39
```

A callout bubble points to the line `library(tidyverse)` with the text "(optional) label for chunk".

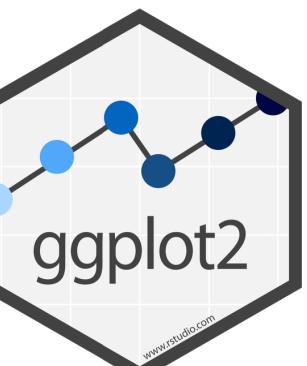


mpg

Fuel economy data for 38 models of car.

mpg

?mpg



Quiz

Confer with your neighbours.

What relationship do you expect to see between engine size (displ) and highway fuel efficiency (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!

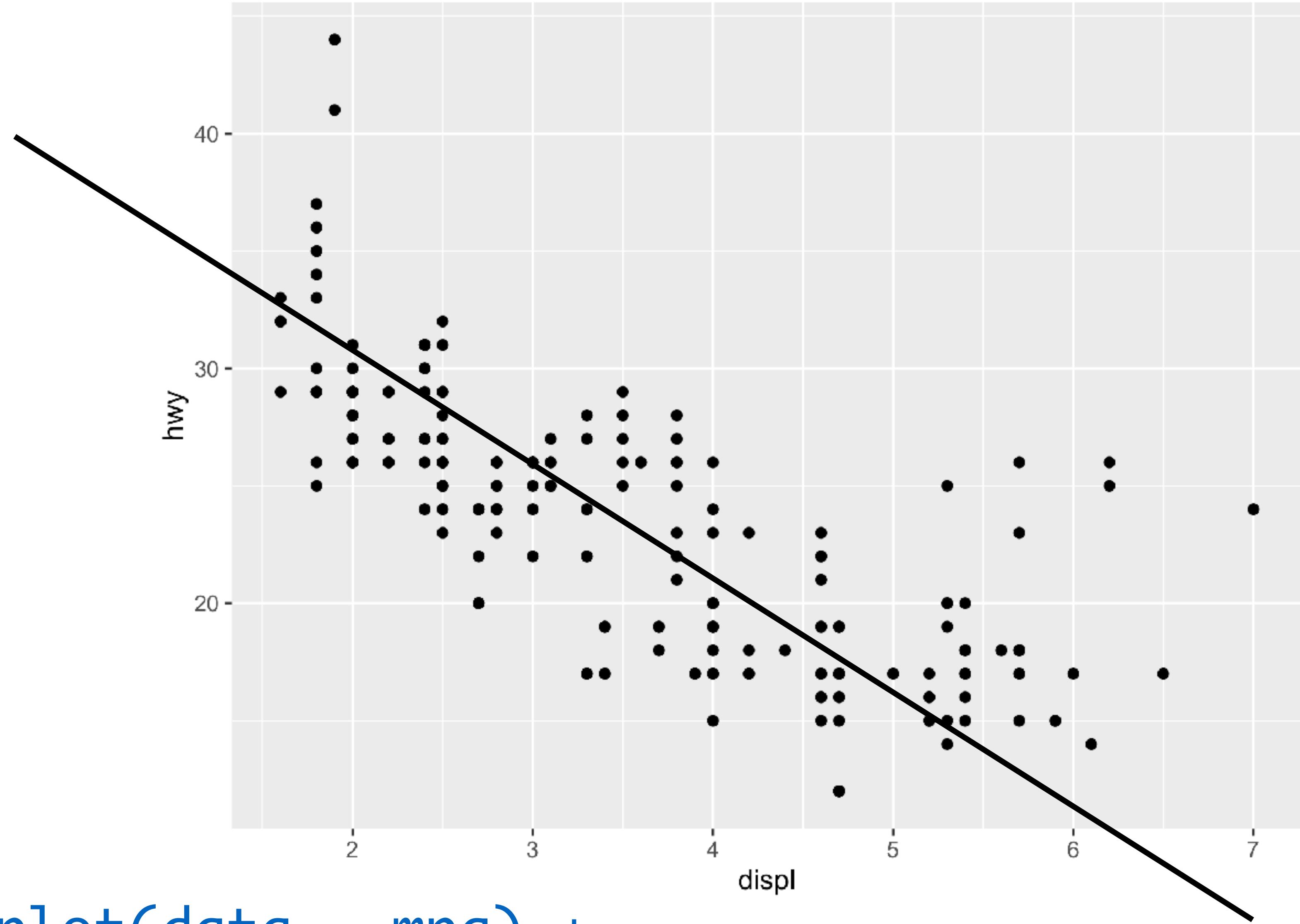
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

I'm working on it

I'm stuck!

I'm done!





```
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))
```

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

data

+ before new line

type of layer

aes()

x variable

y variable

A Template

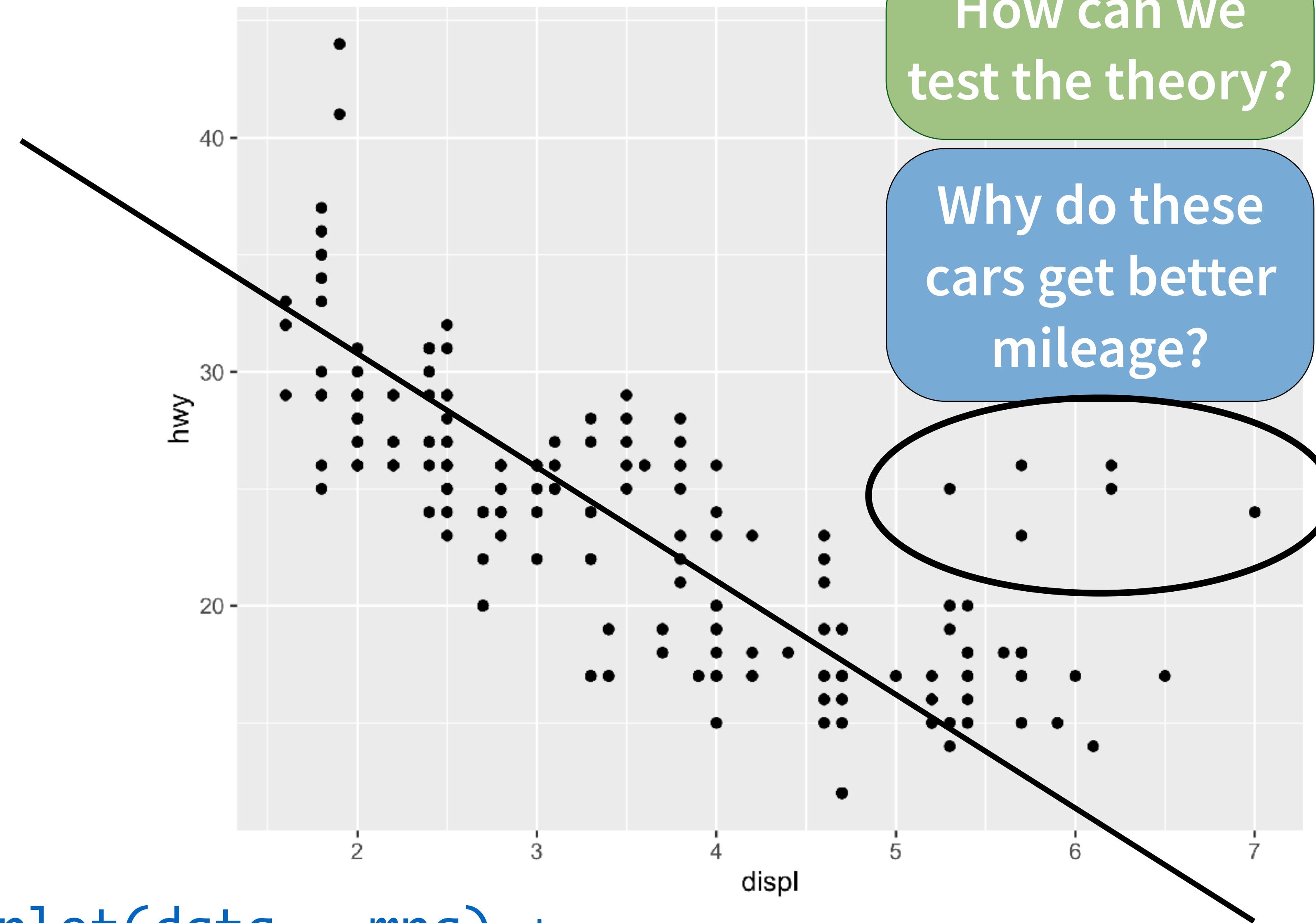
```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

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

A Template

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

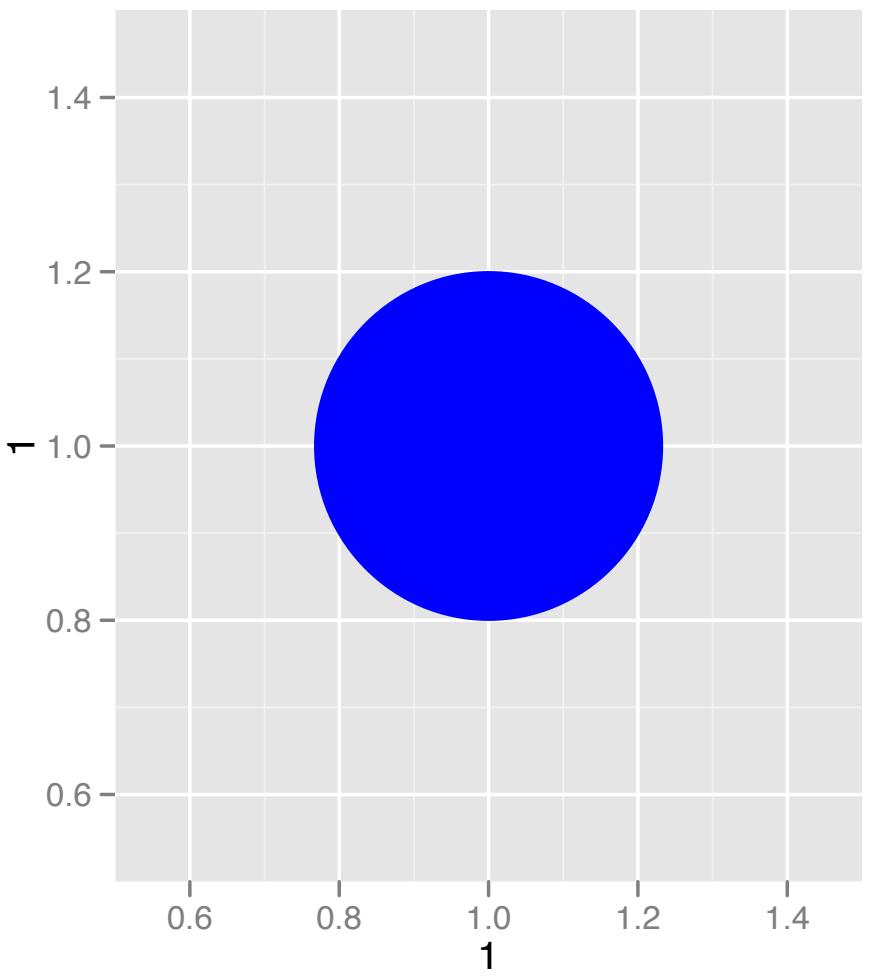
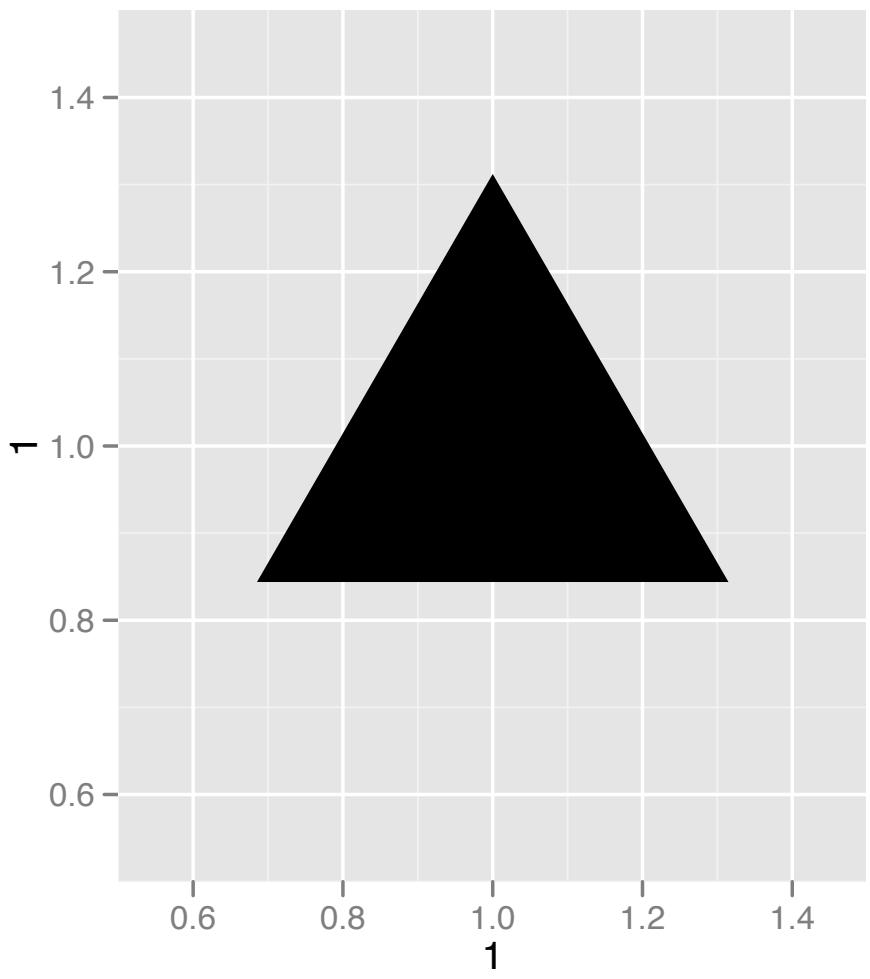
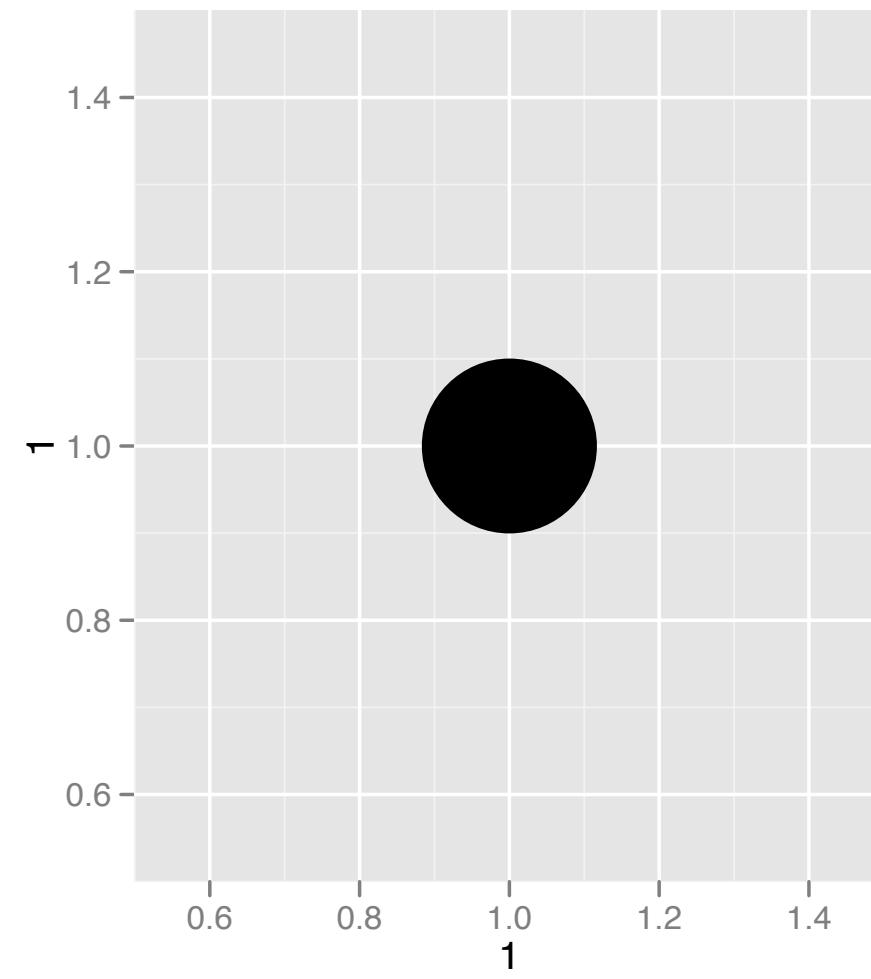
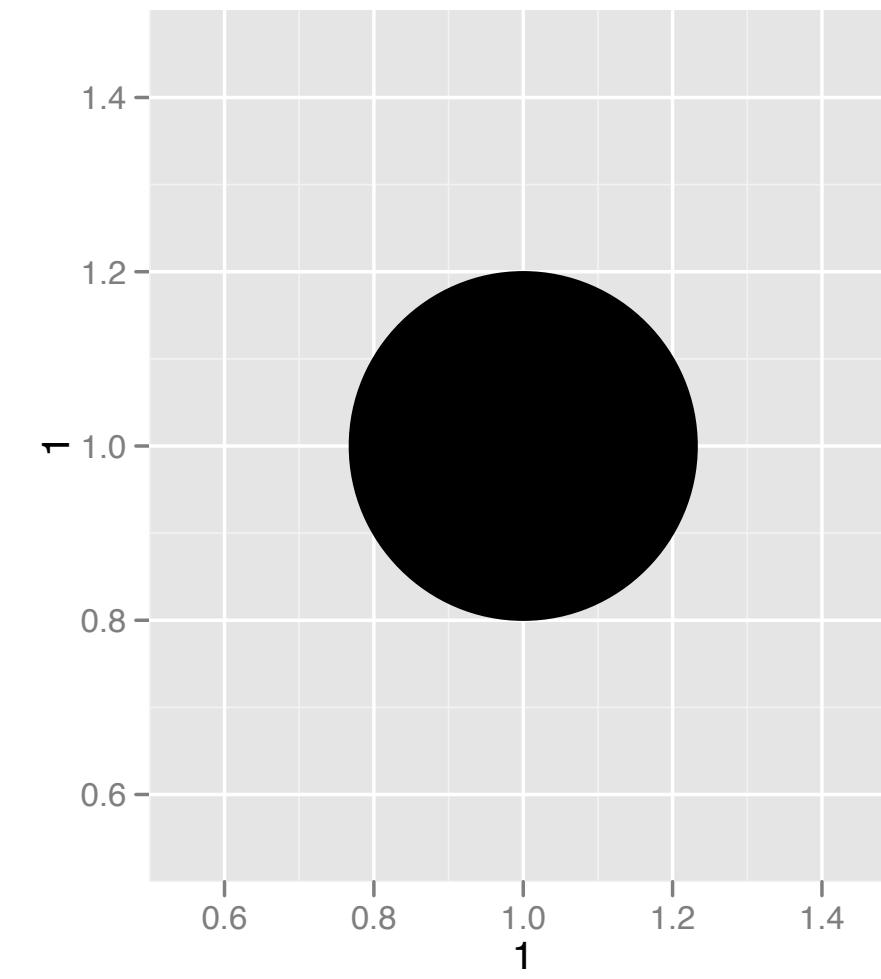
Mappings



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

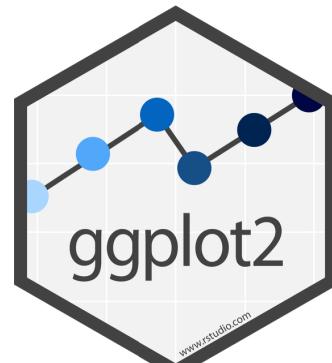
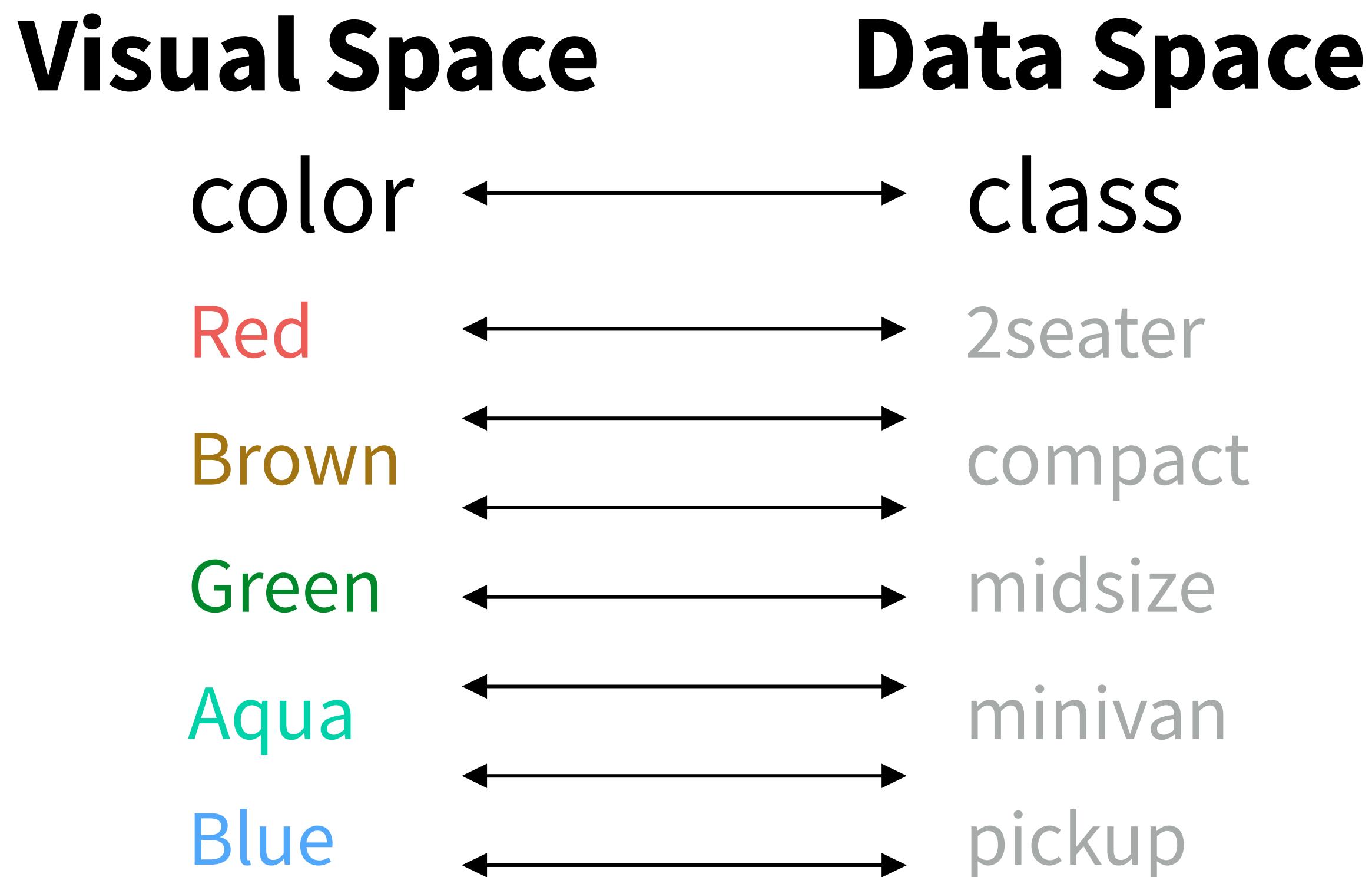
Aesthetics

Visual properties of a geometric object



How do the appearance of these points vary?

Mappings describe how aesthetics should relate to variables in the data.



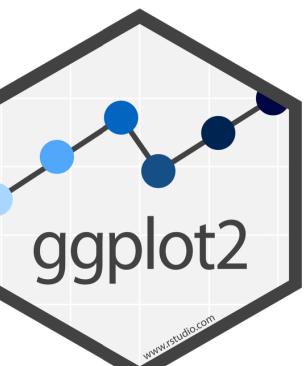
Aesthetics

aesthetic
property

Variable to
map it to

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

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



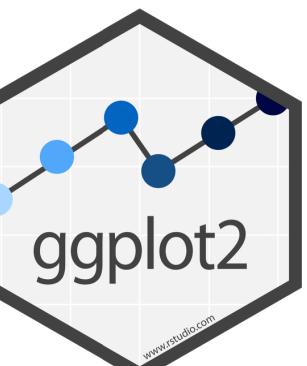
Aesthetics

aesthetic
property

Variable to
map it to

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

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



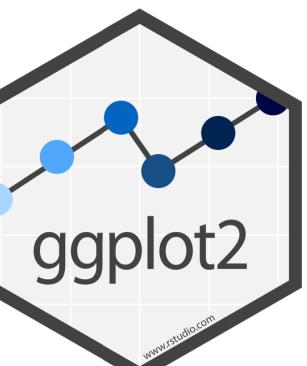
Aesthetics

aesthetic
property

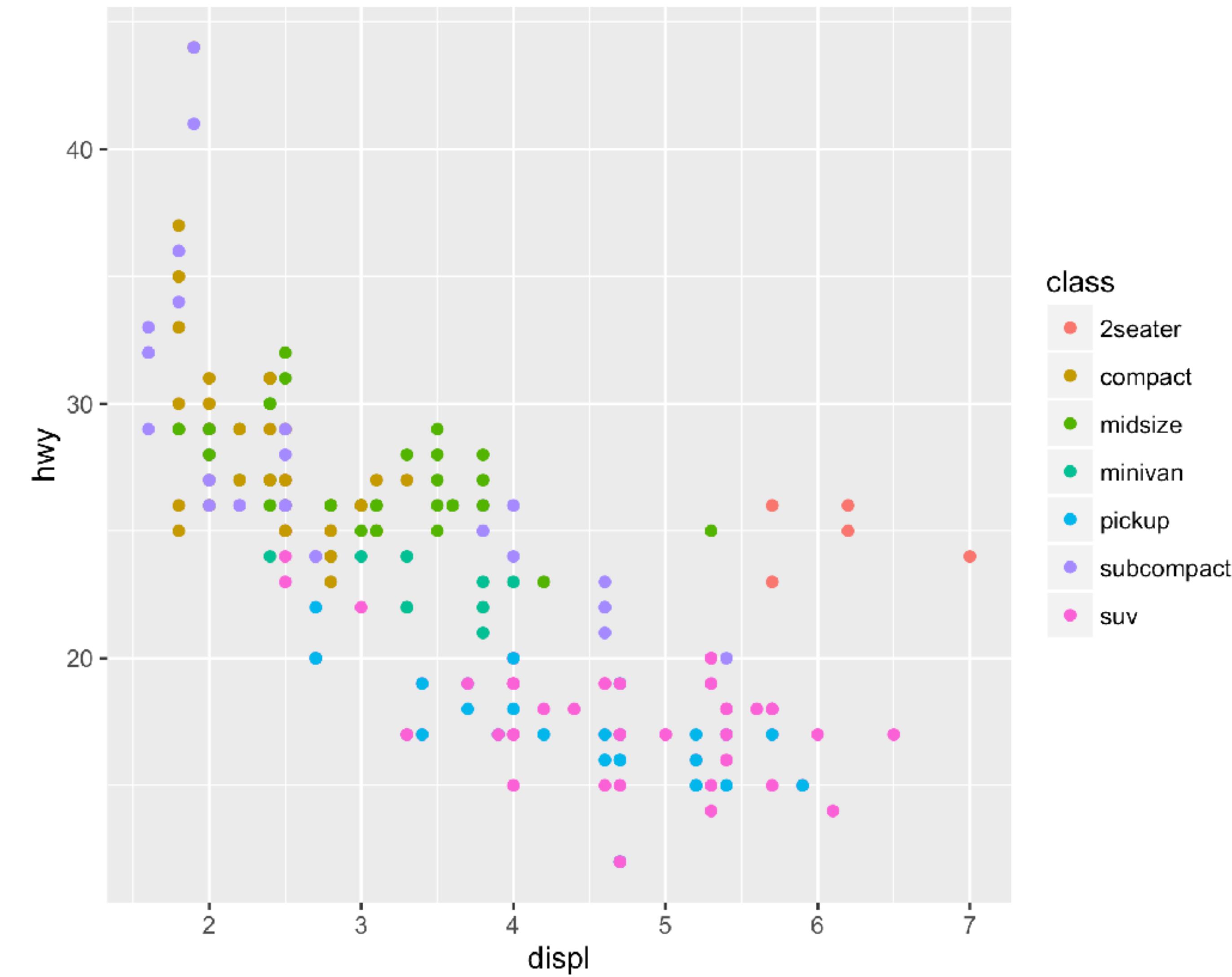
Variable to
map it to

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

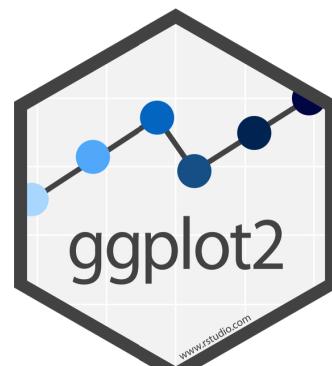
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
```



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



Legend
added
automatically



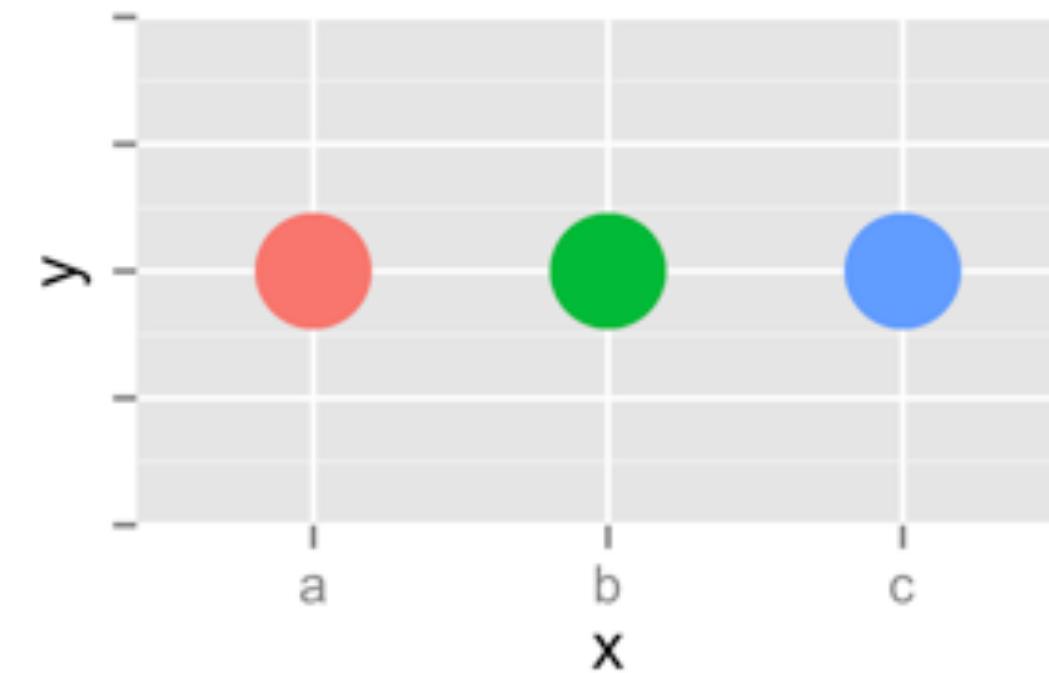
Your Turn 2

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

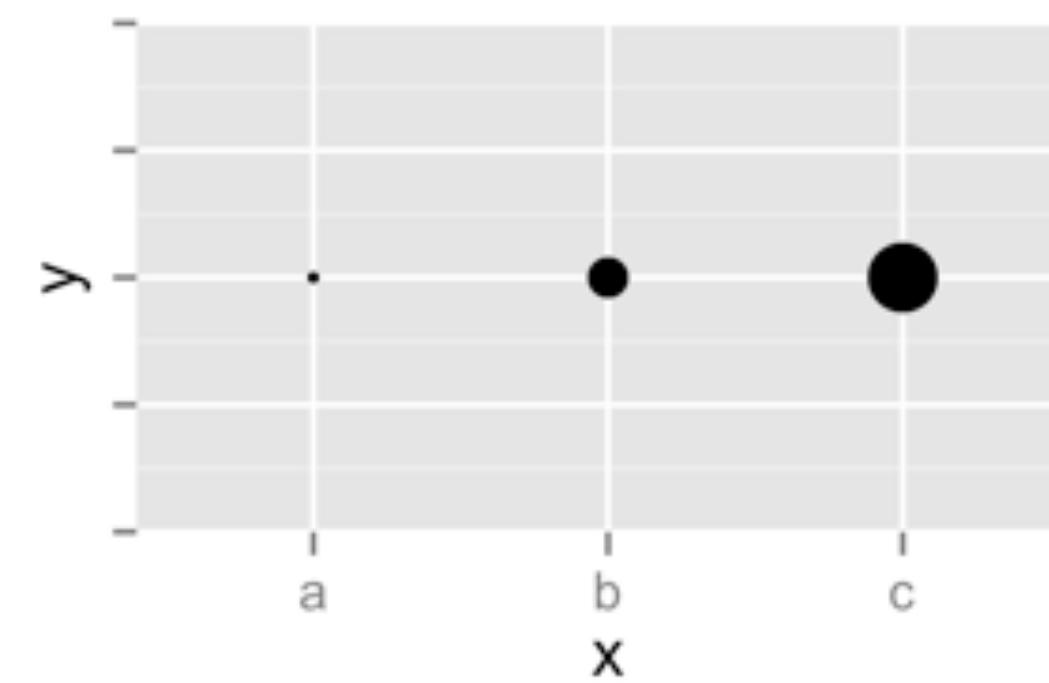
- Do different things happen when you map aesthetics to discrete and continuous variables?
- What happens when you use more than one aesthetic?



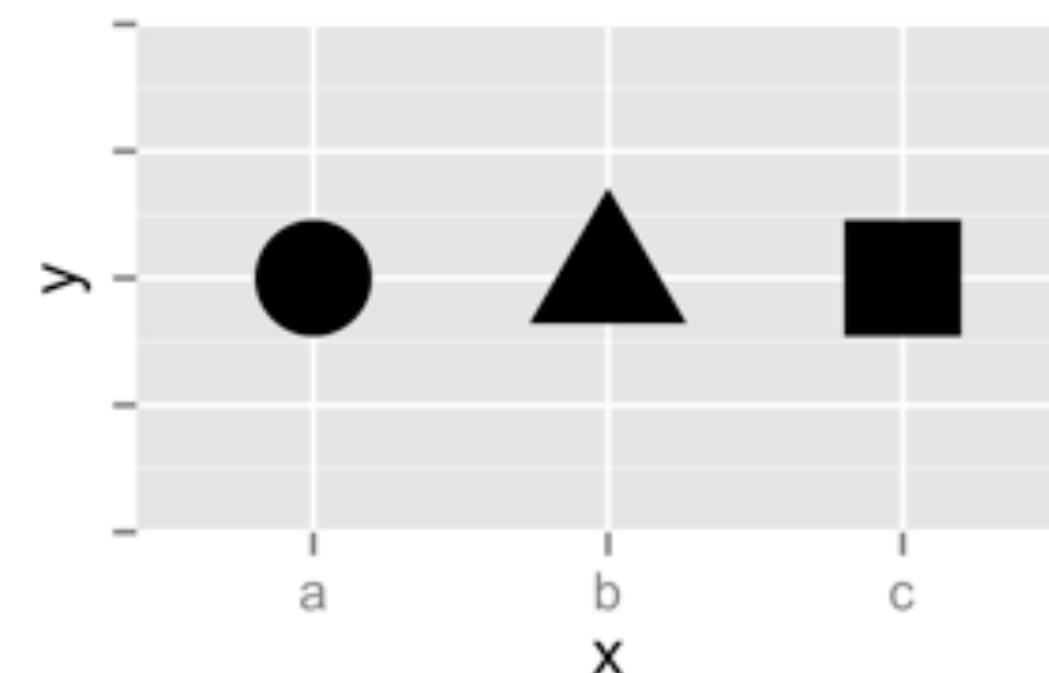
Color



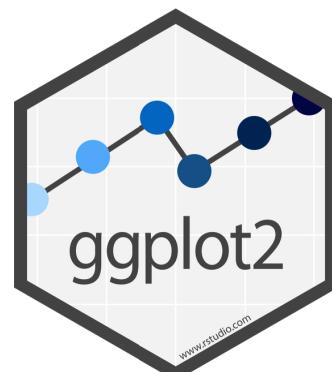
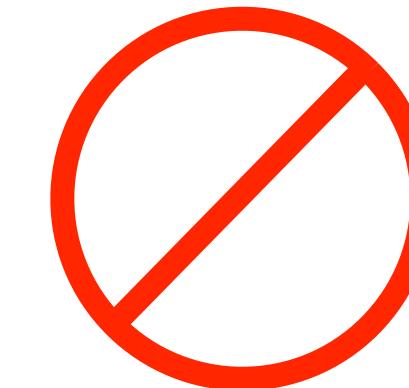
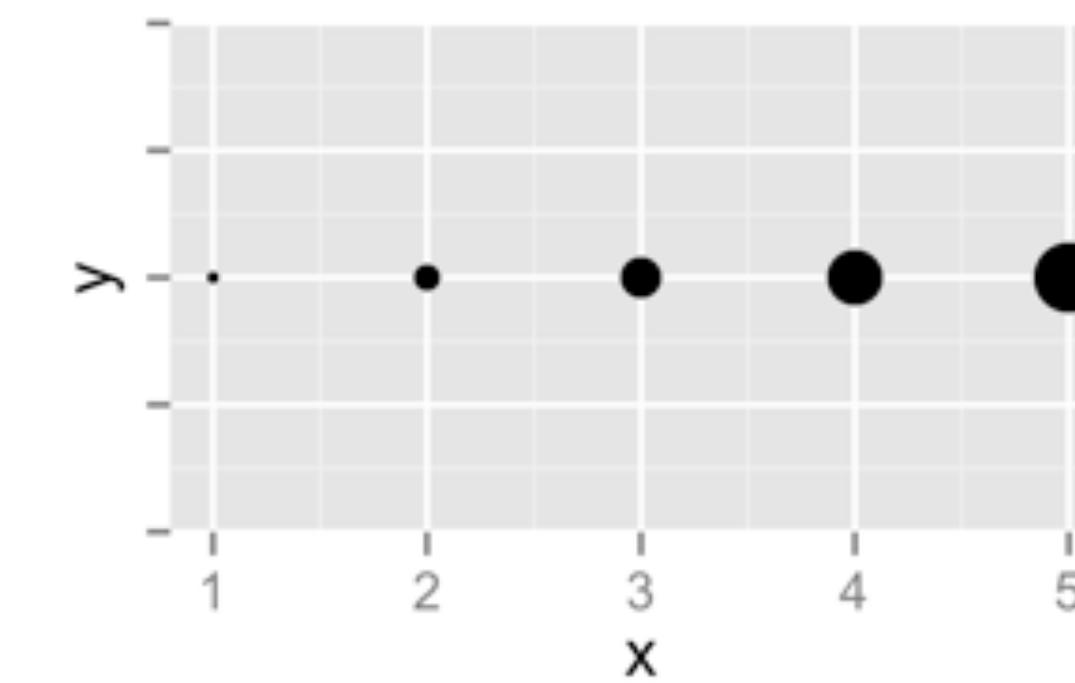
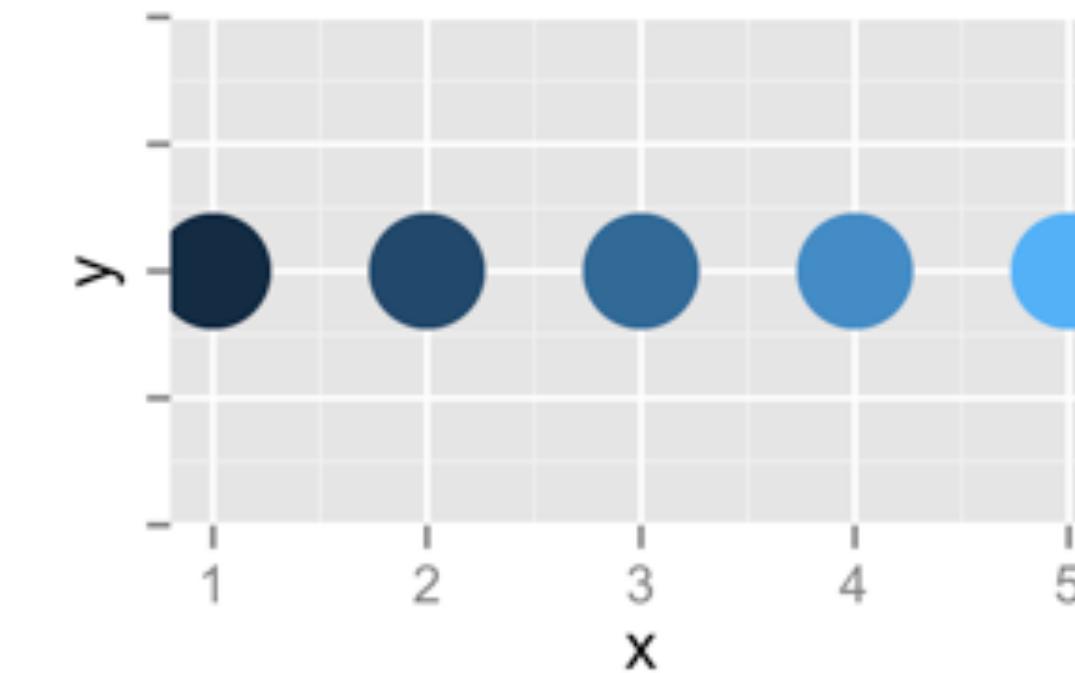
Size



Shape

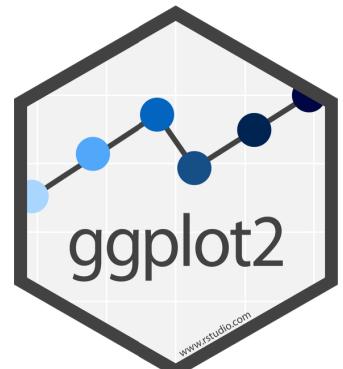
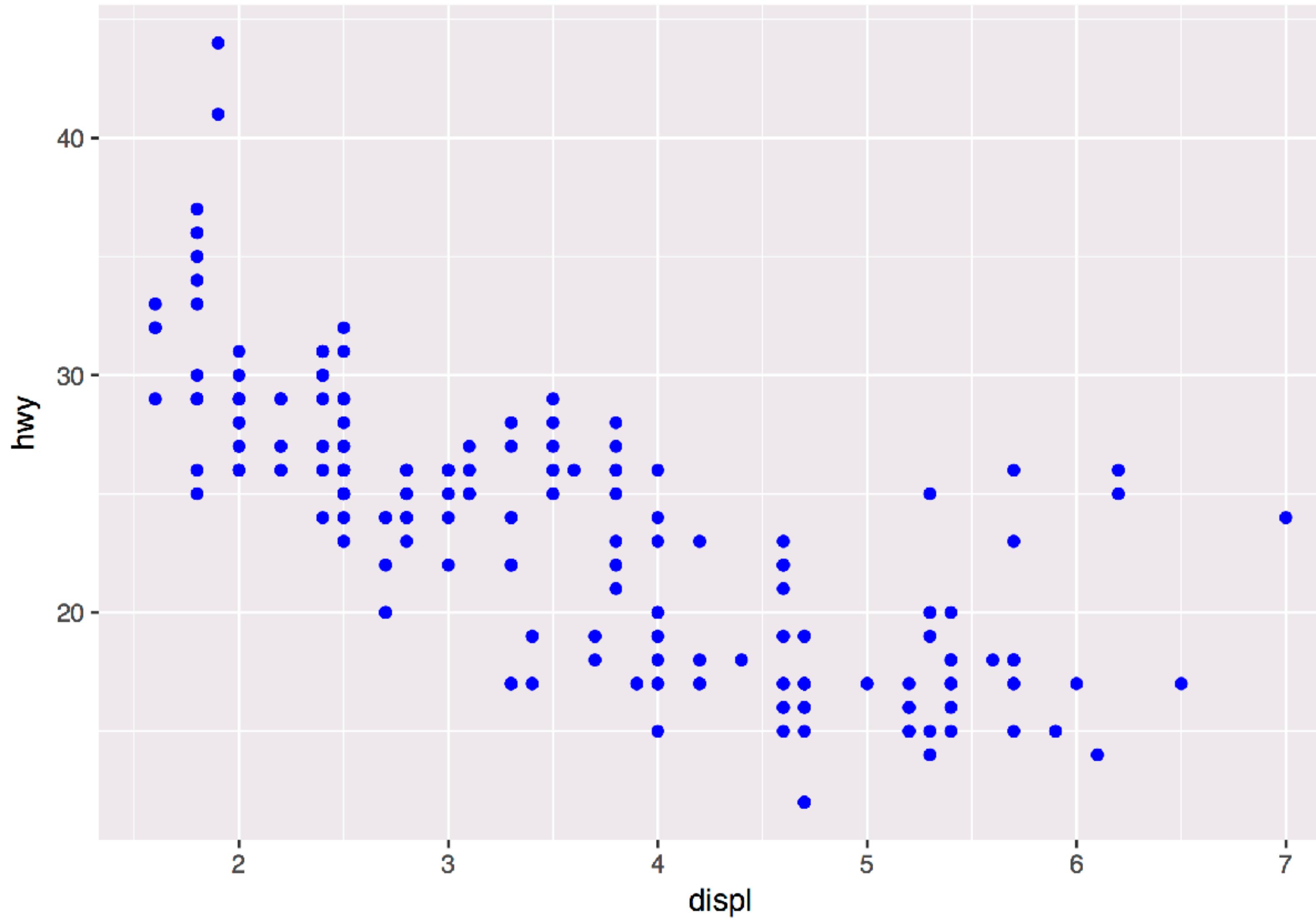


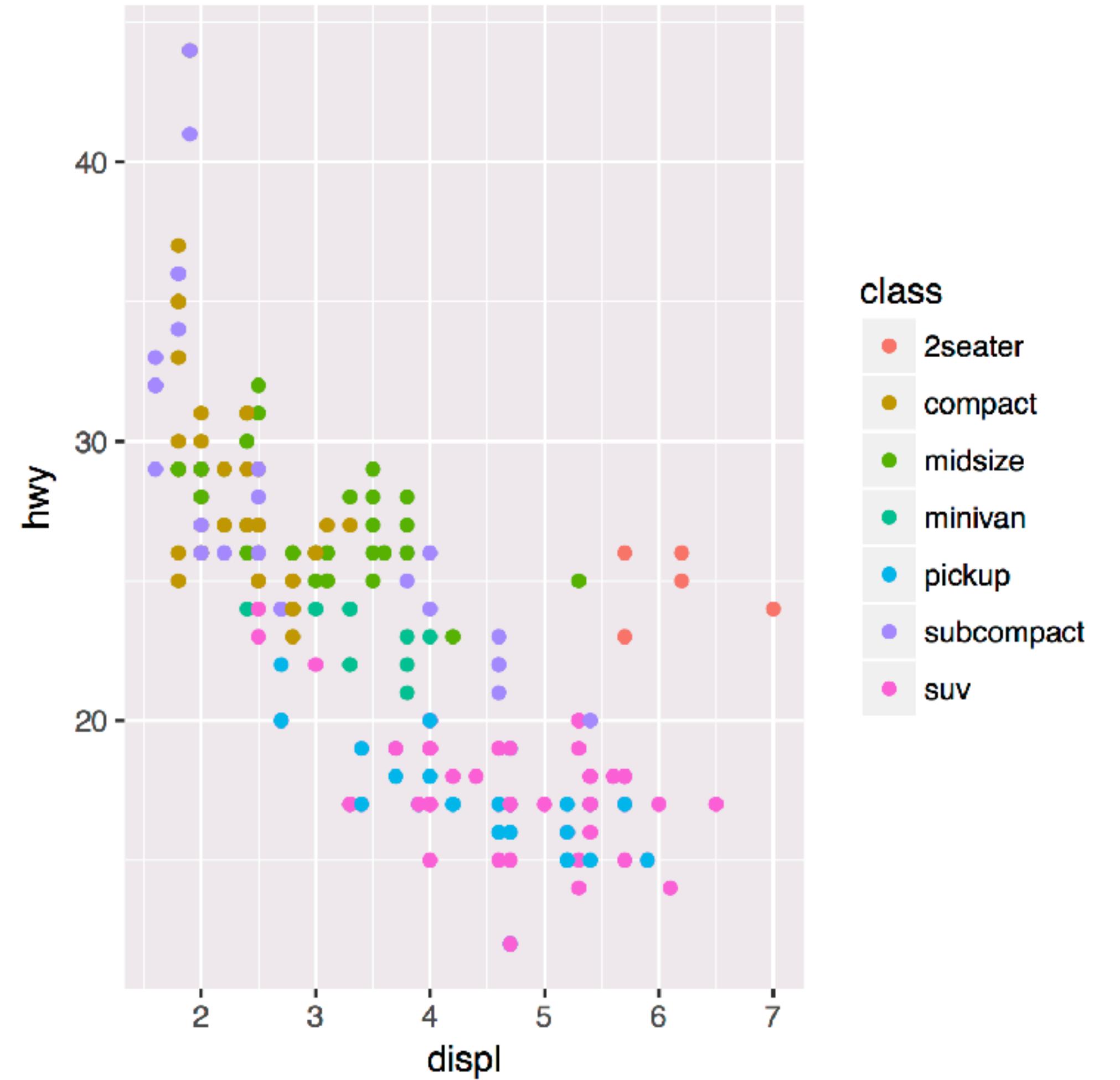
Continuous



Set vs. Map

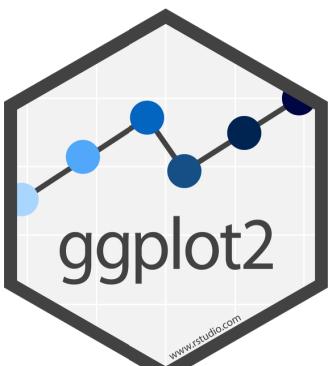
How would you make this plot?



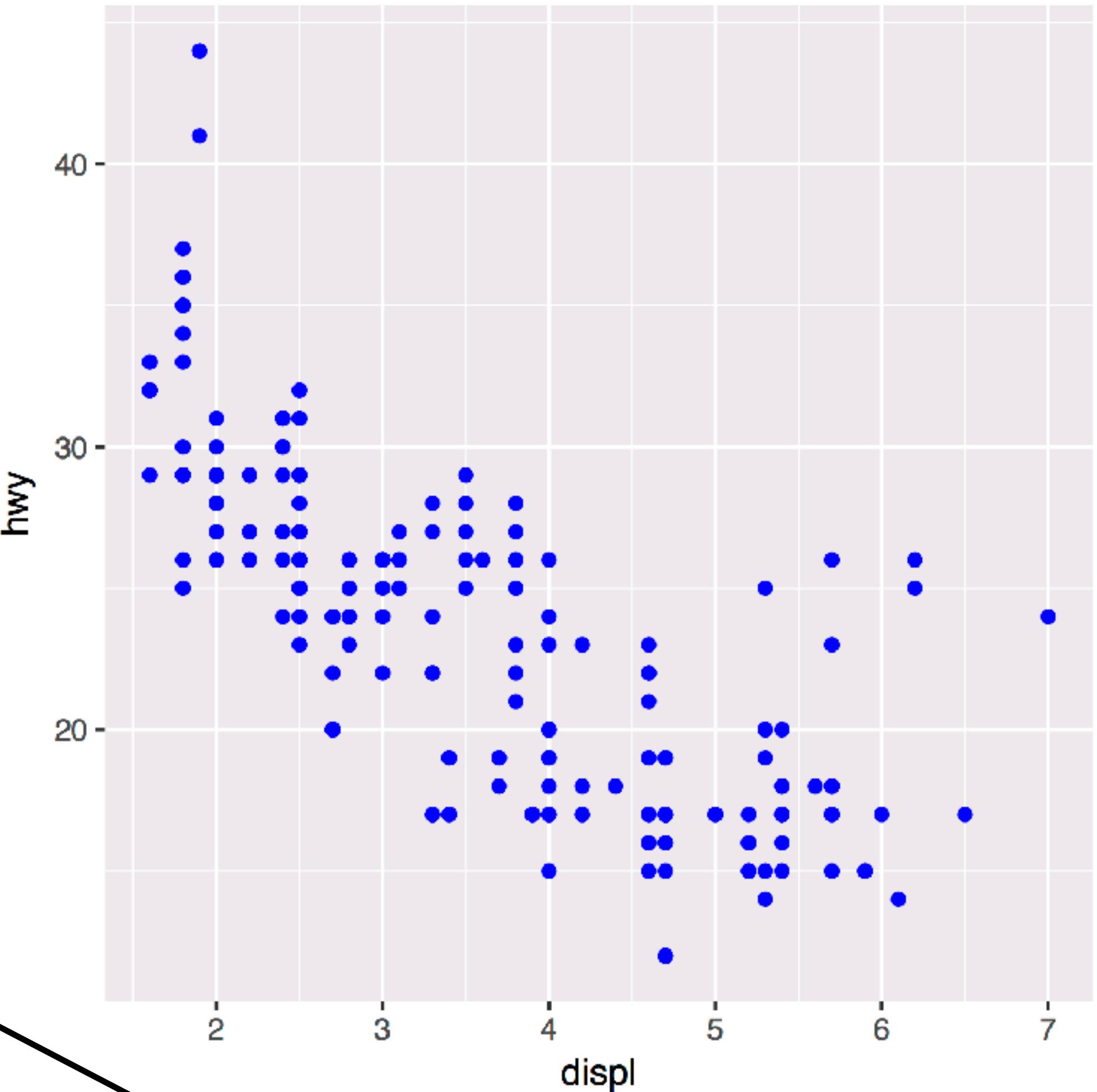


Inside of aes(): maps an aesthetic to a variable

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

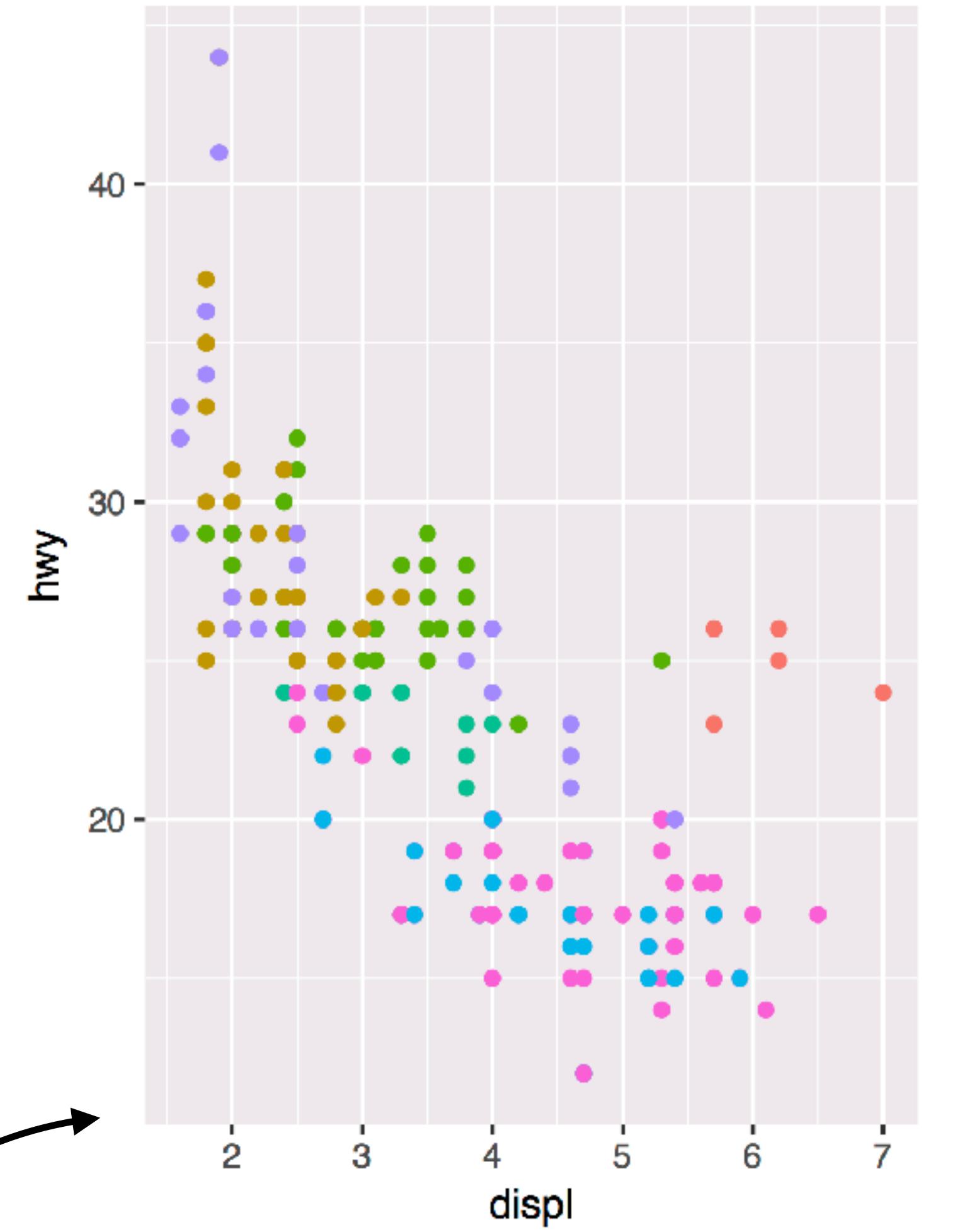


Outside of aes(): sets
an aesthetic to a value



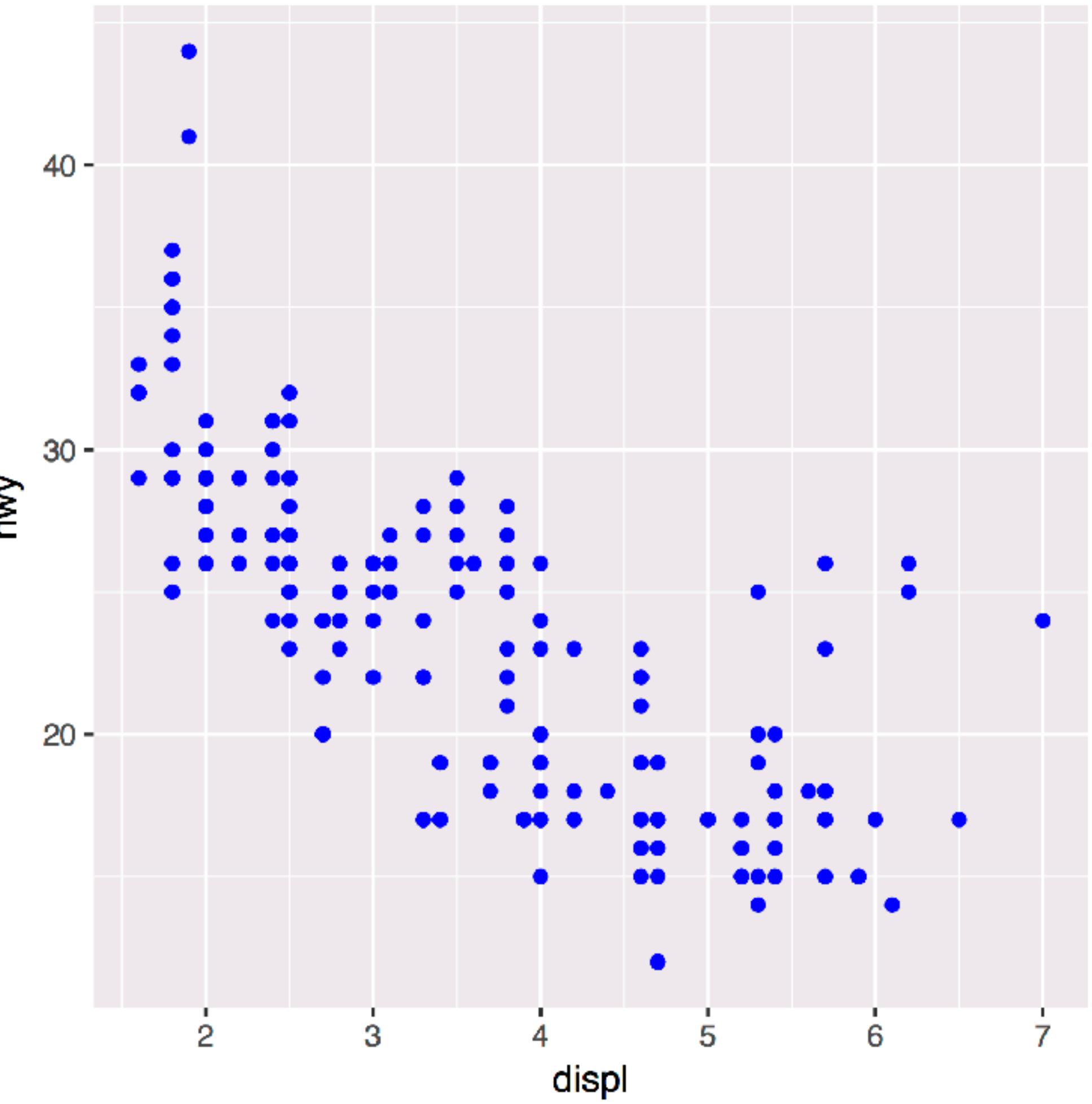
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

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



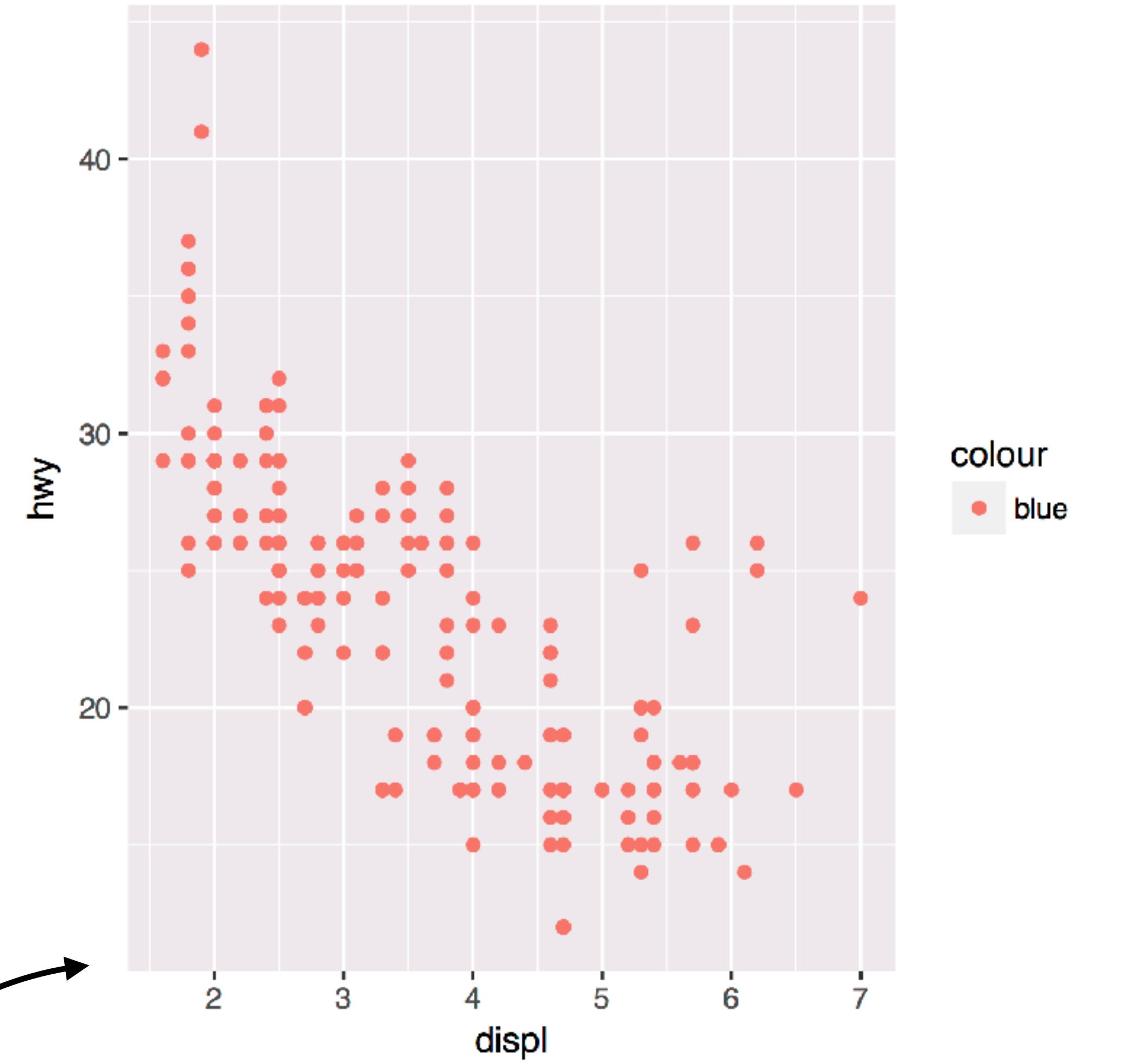
class

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

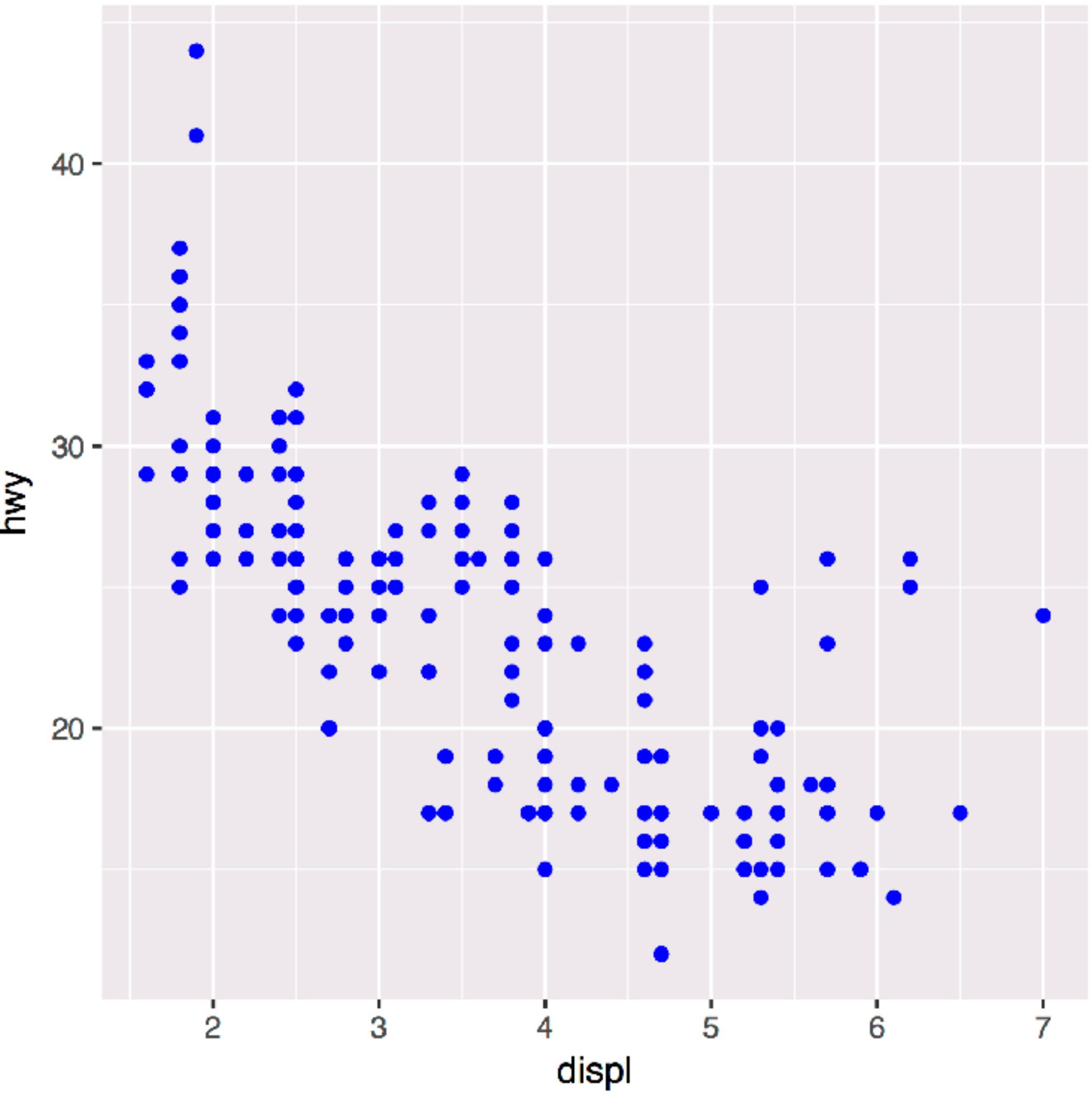


```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

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



colour
● blue



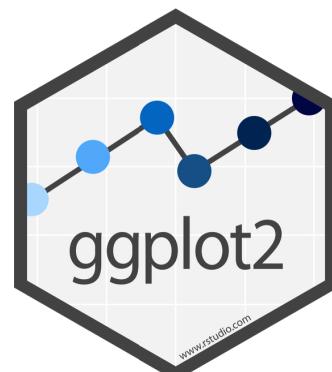
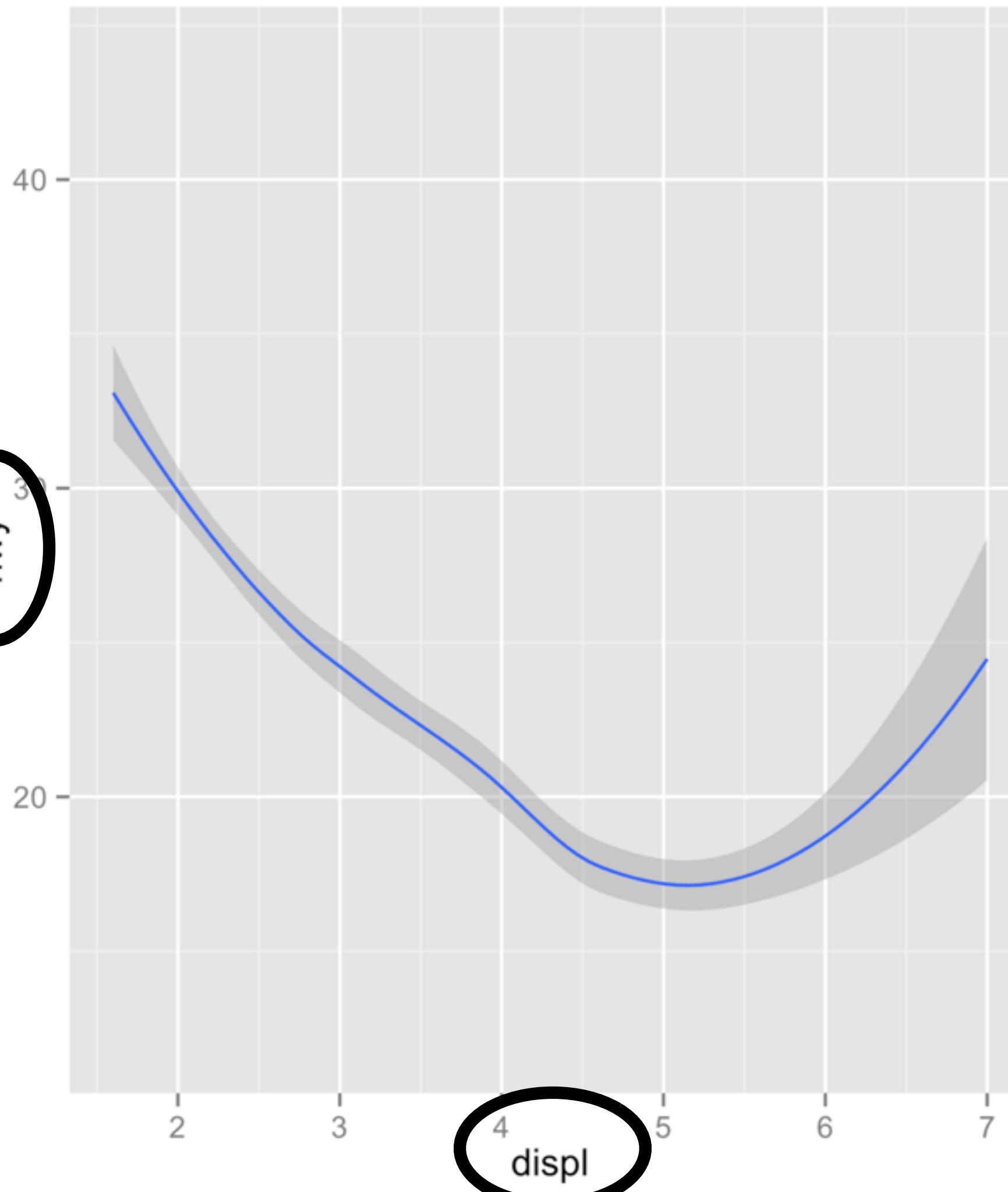
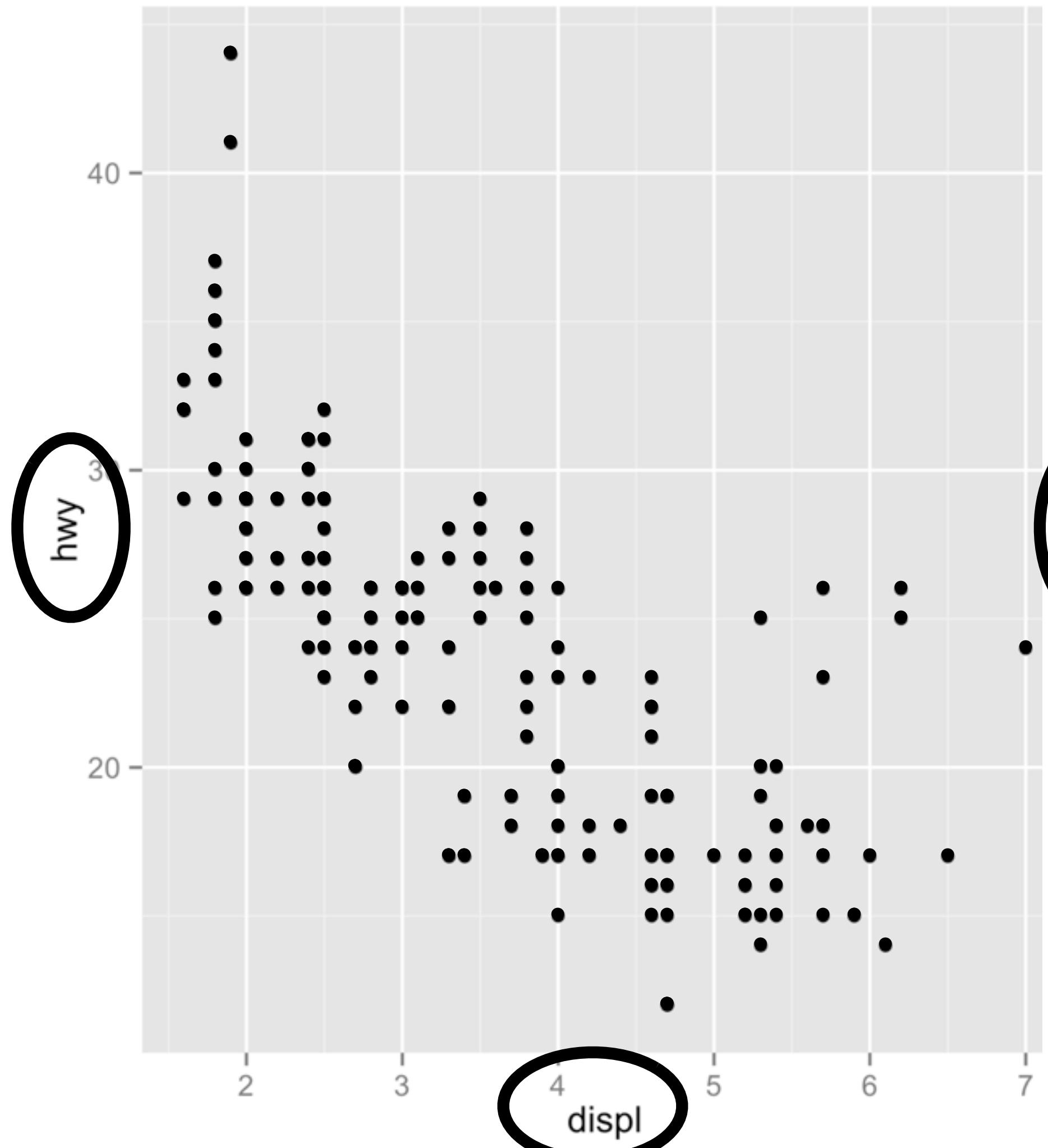
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = "blue"))
```

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

Geoms

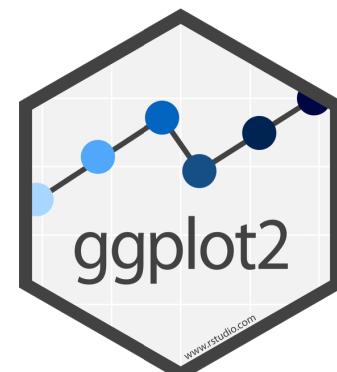
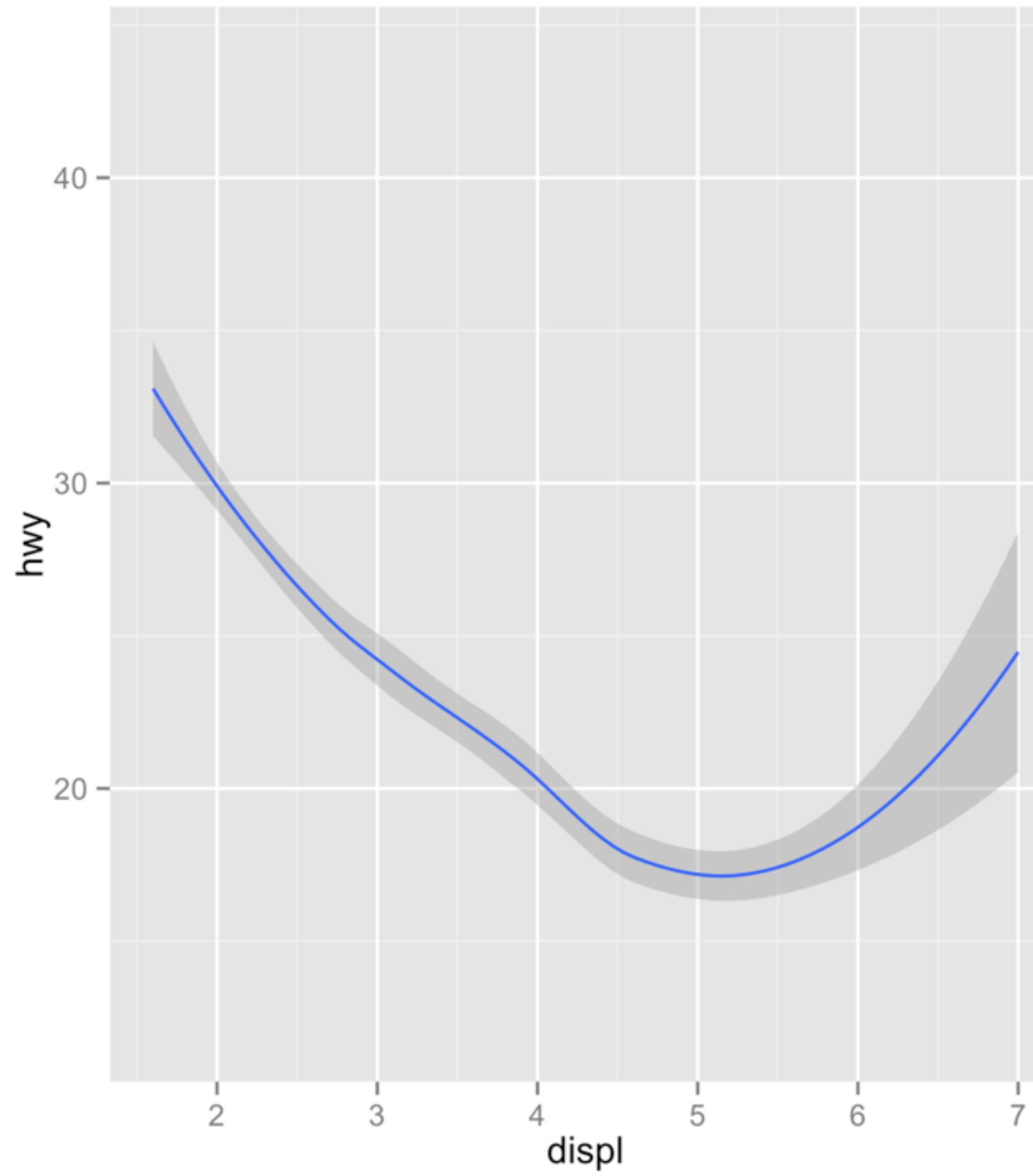
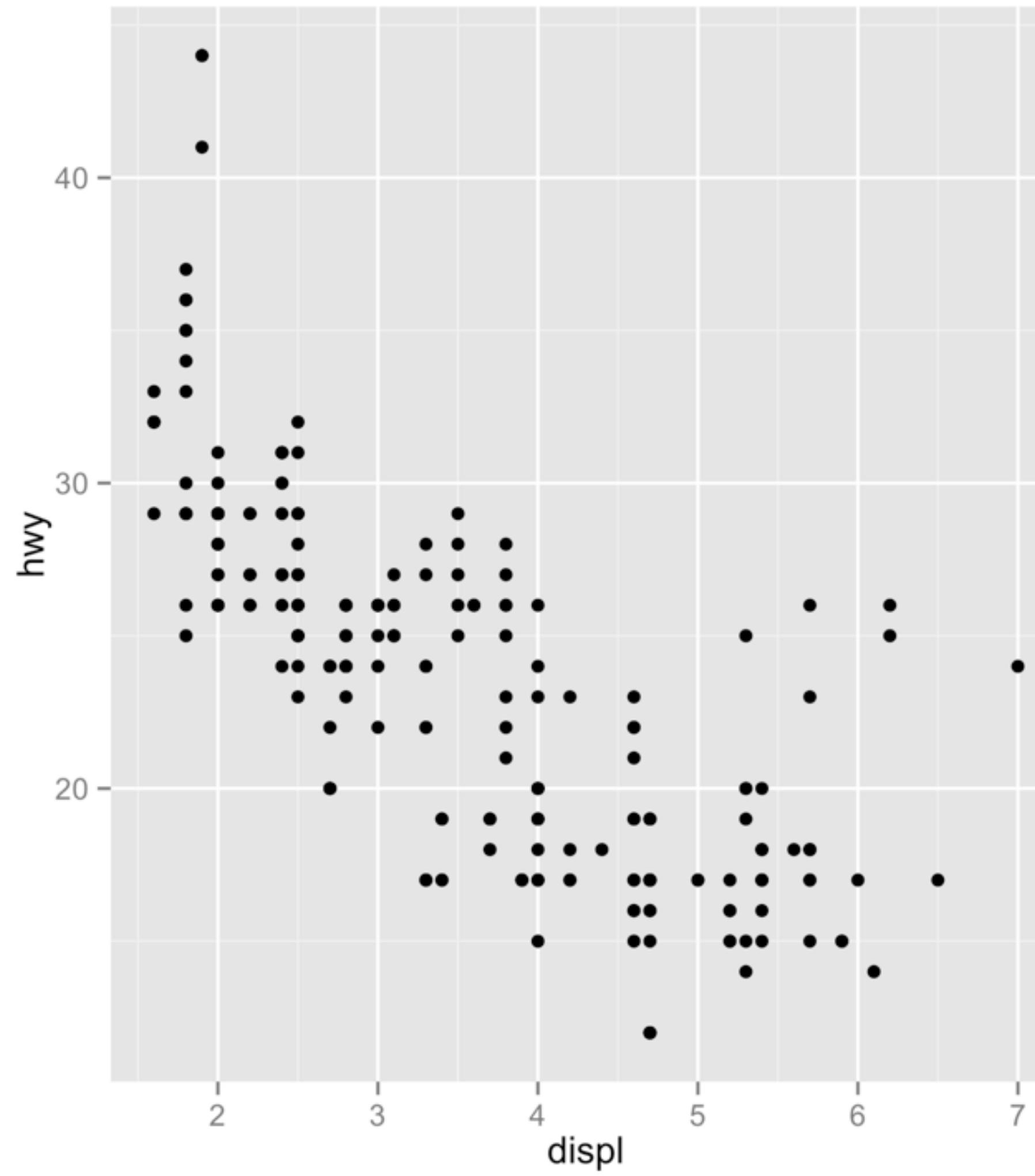
How are these plots similar?

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

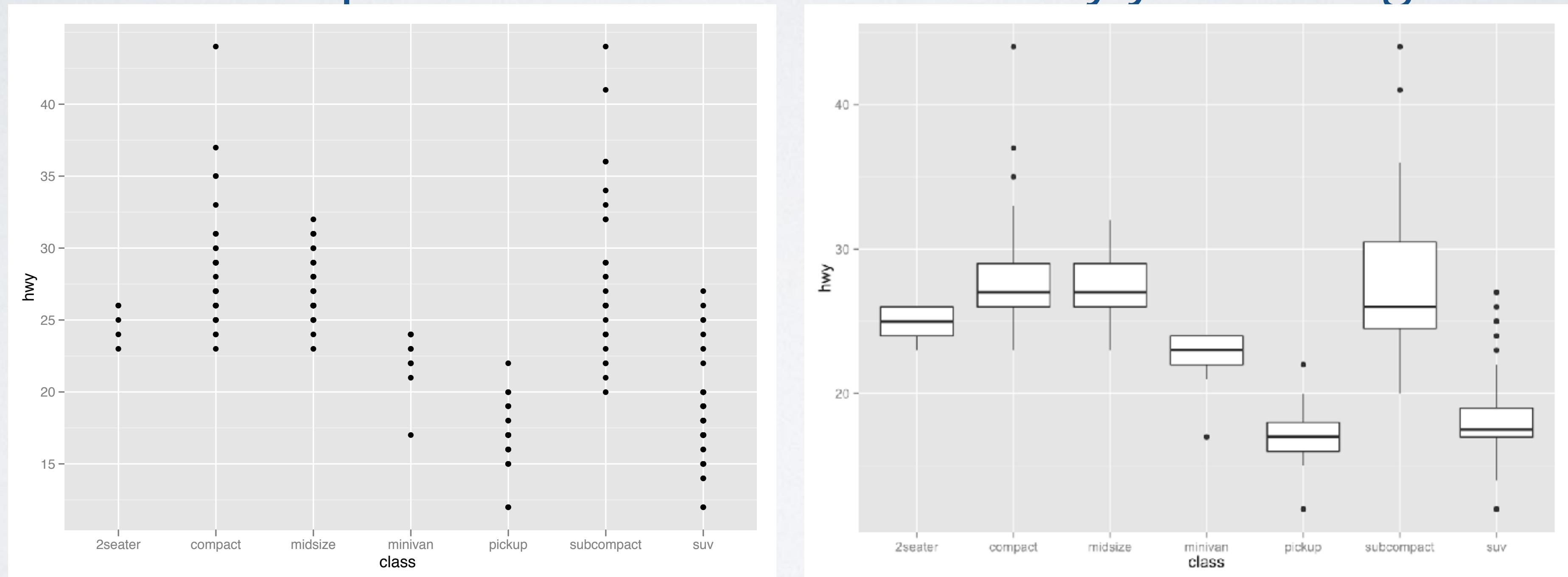


geoms

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

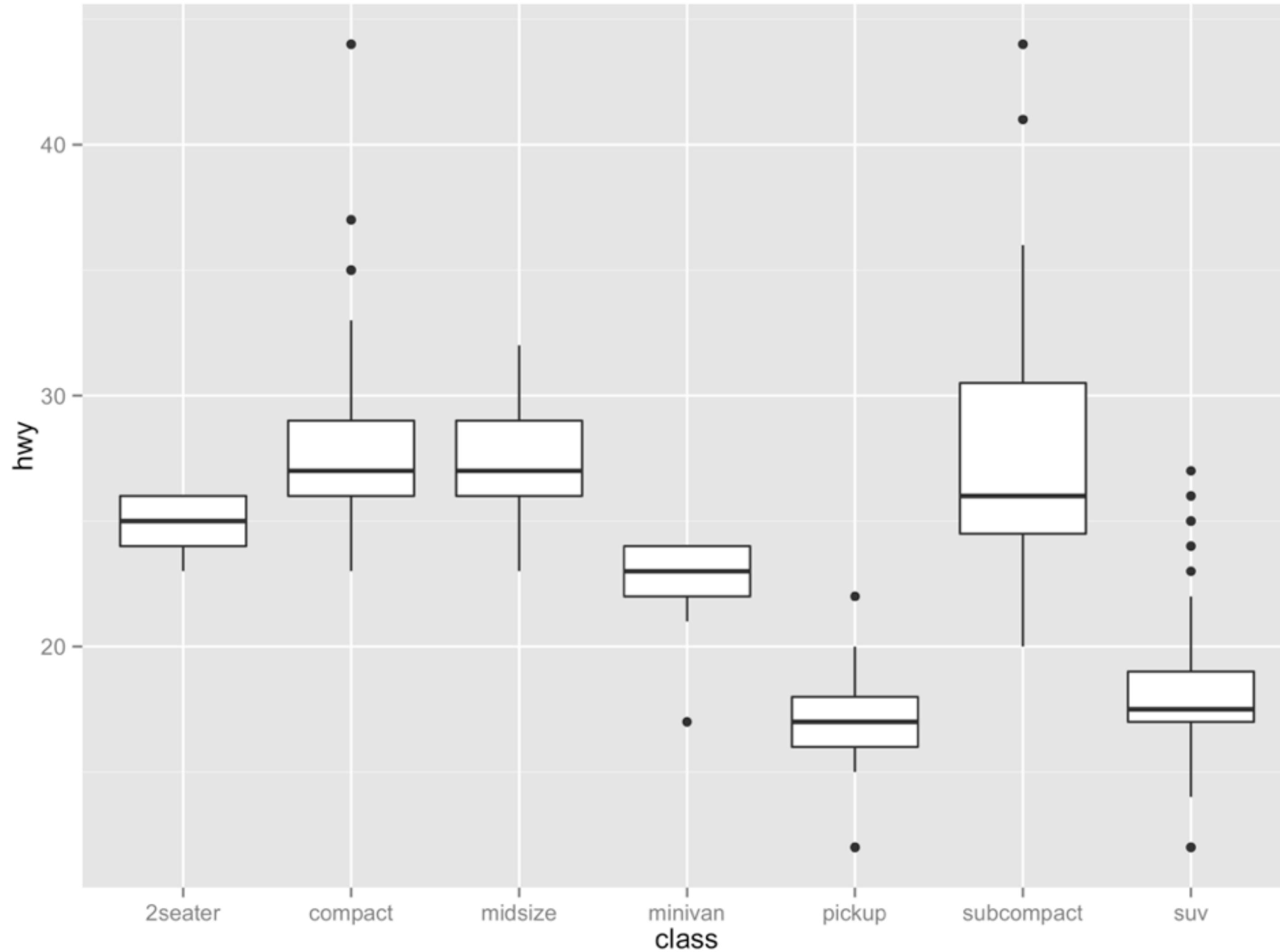

Your Turn 3

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))
```

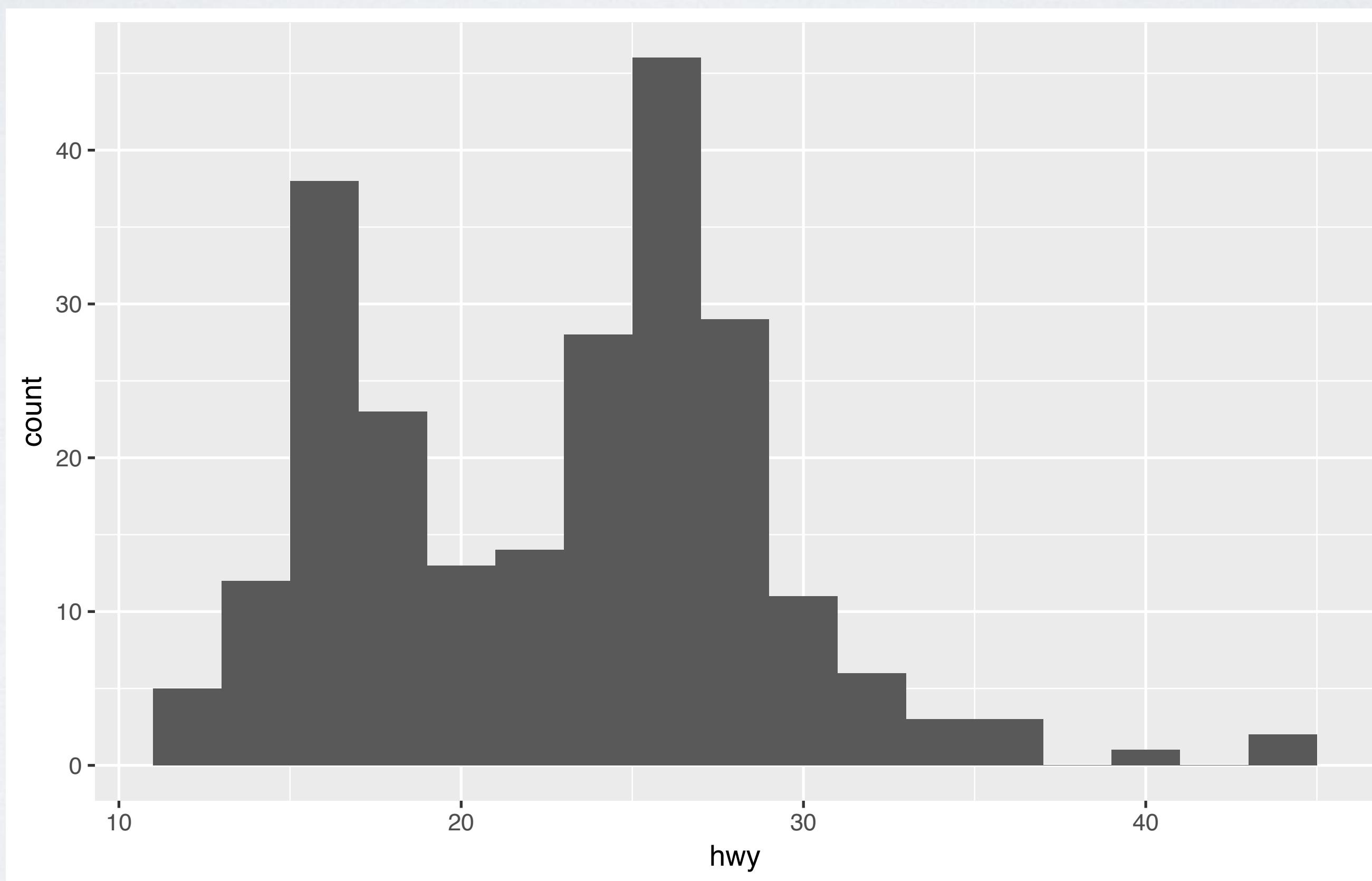
02 : 00

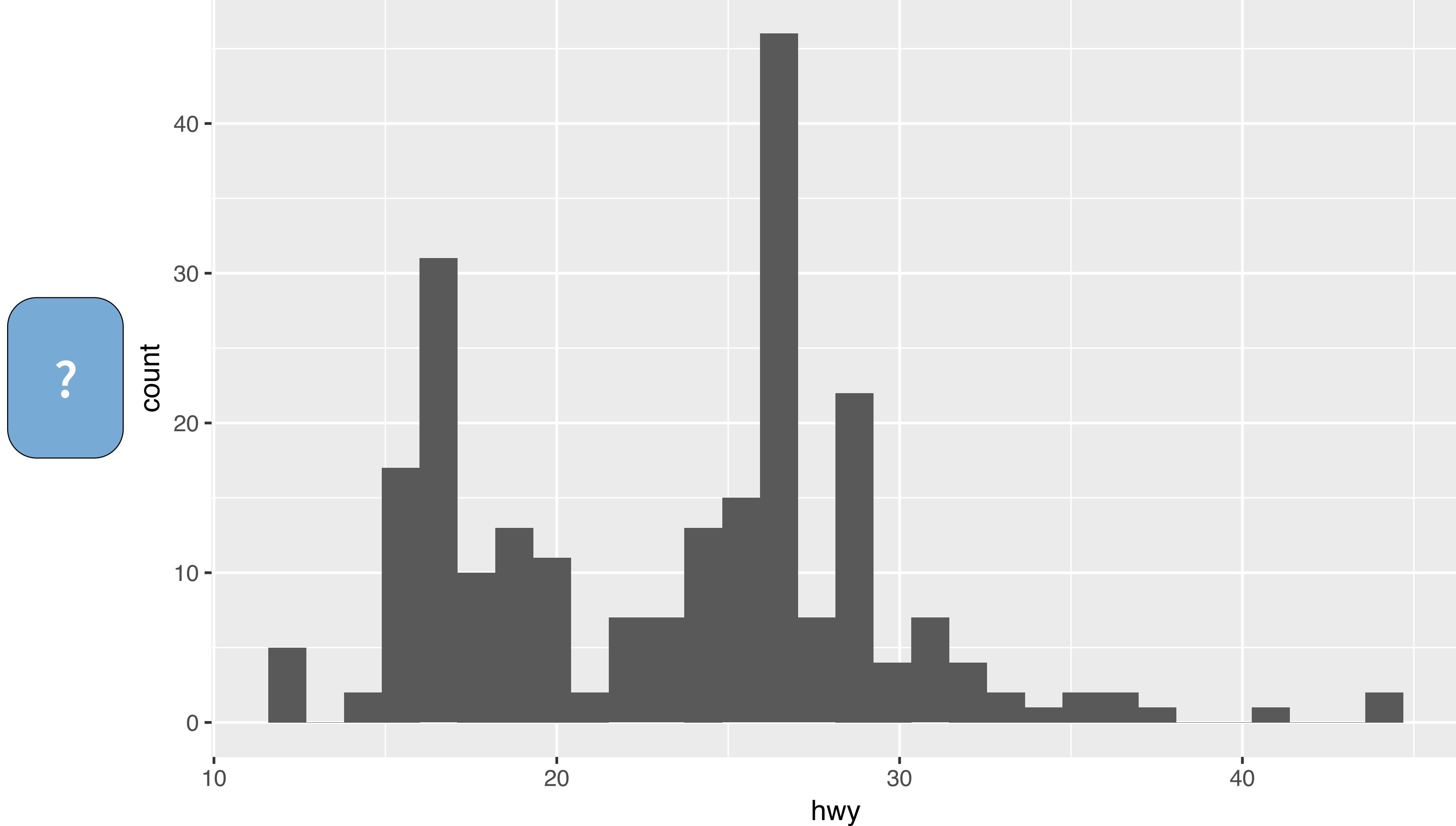


```
ggplot(data = mpg) +  
  geom_boxplot(mapping = aes(x = class, y = hwy))
```

Your Turn 4

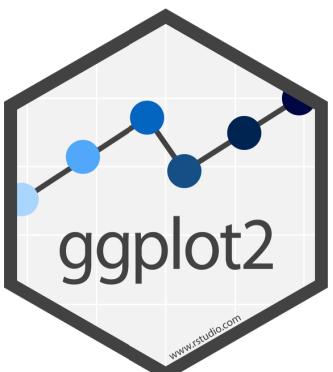
With your partner, make the **histogram** of hwy below. Use the cheatsheet. **Hint:** do not supply a y variable.

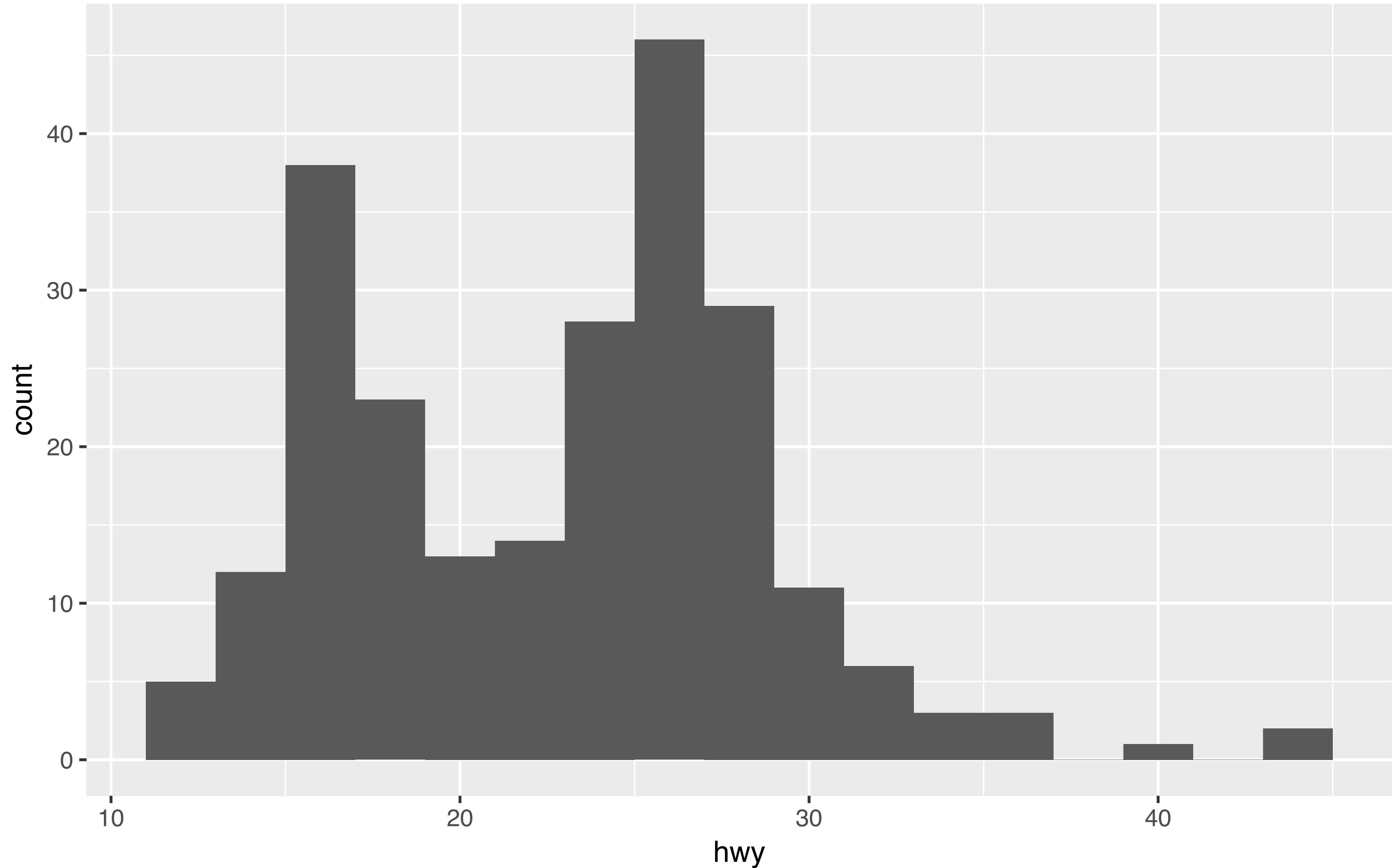




```
ggplot(data = mpg) +  
  geom_histogram(mapping = aes(x = hwy))
```

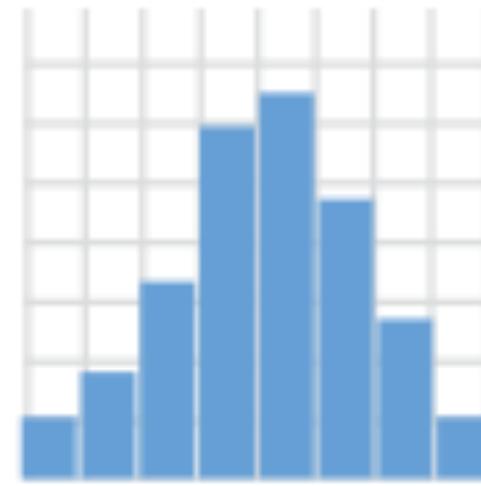
No y aesthetic





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

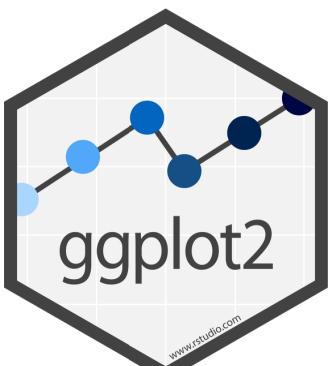
On the cheatsheat:



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

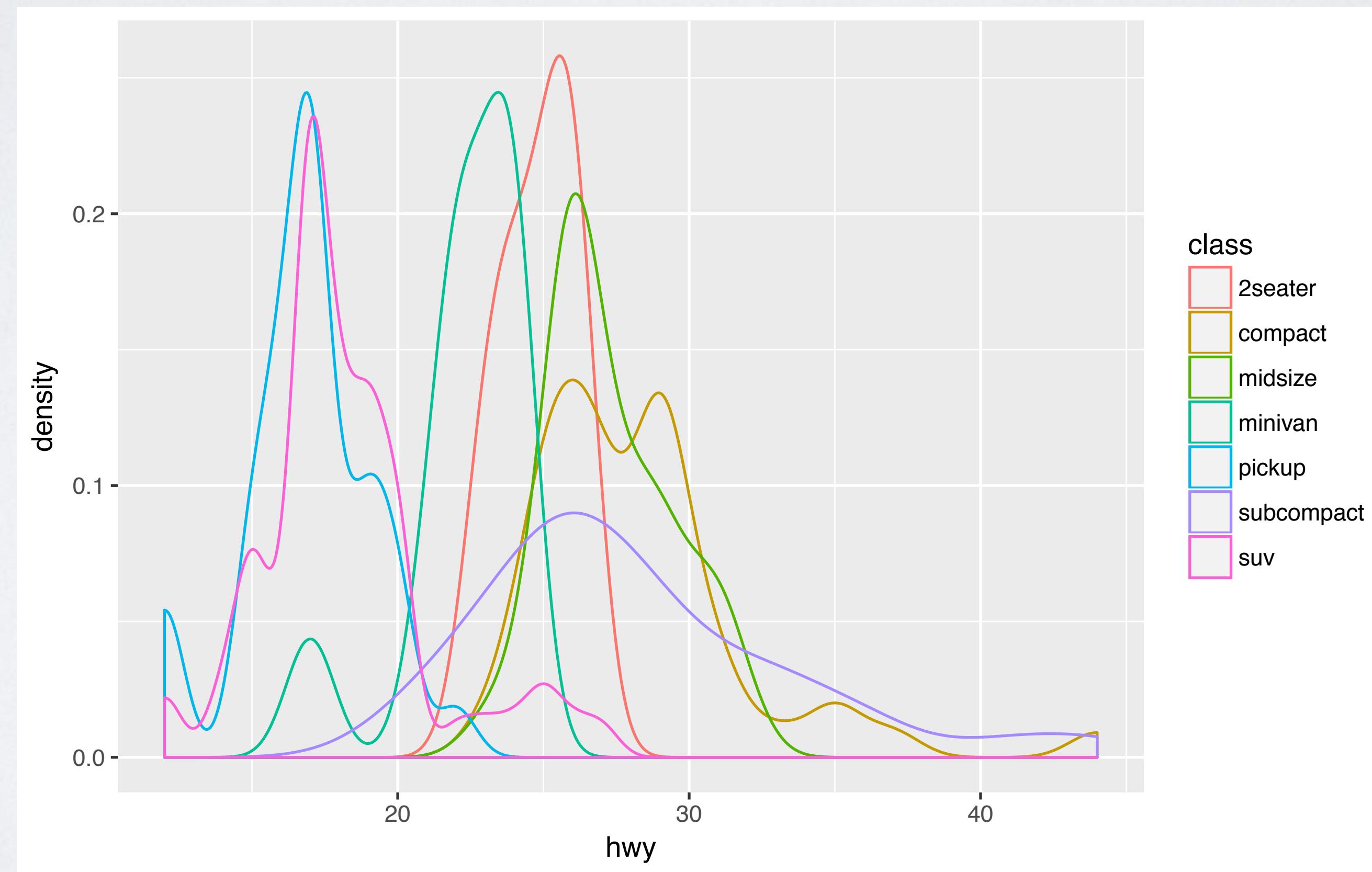
Arguments inside (), are
geom specific options

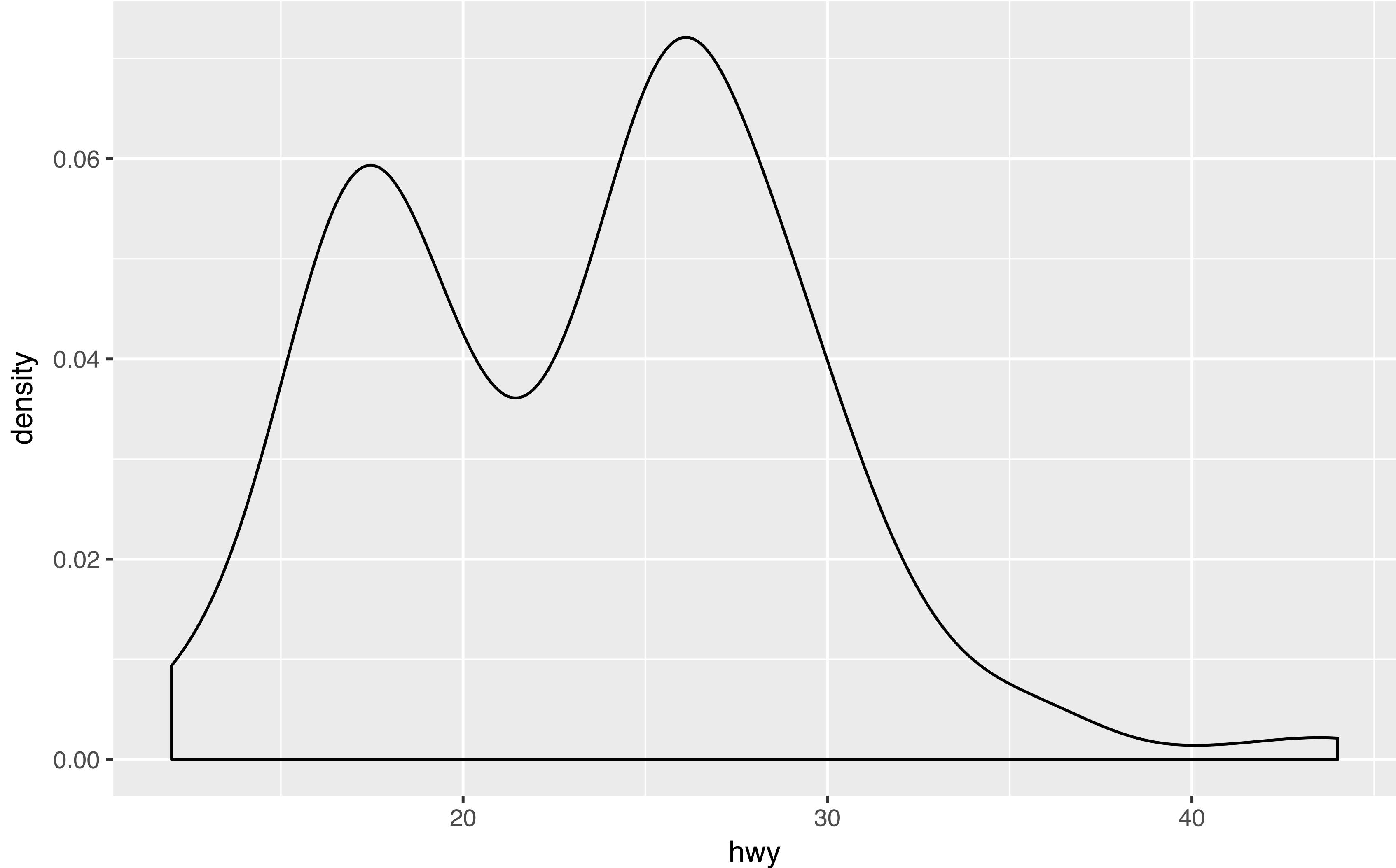
Outside the (), are
aesthetics that can be
mapped or set.



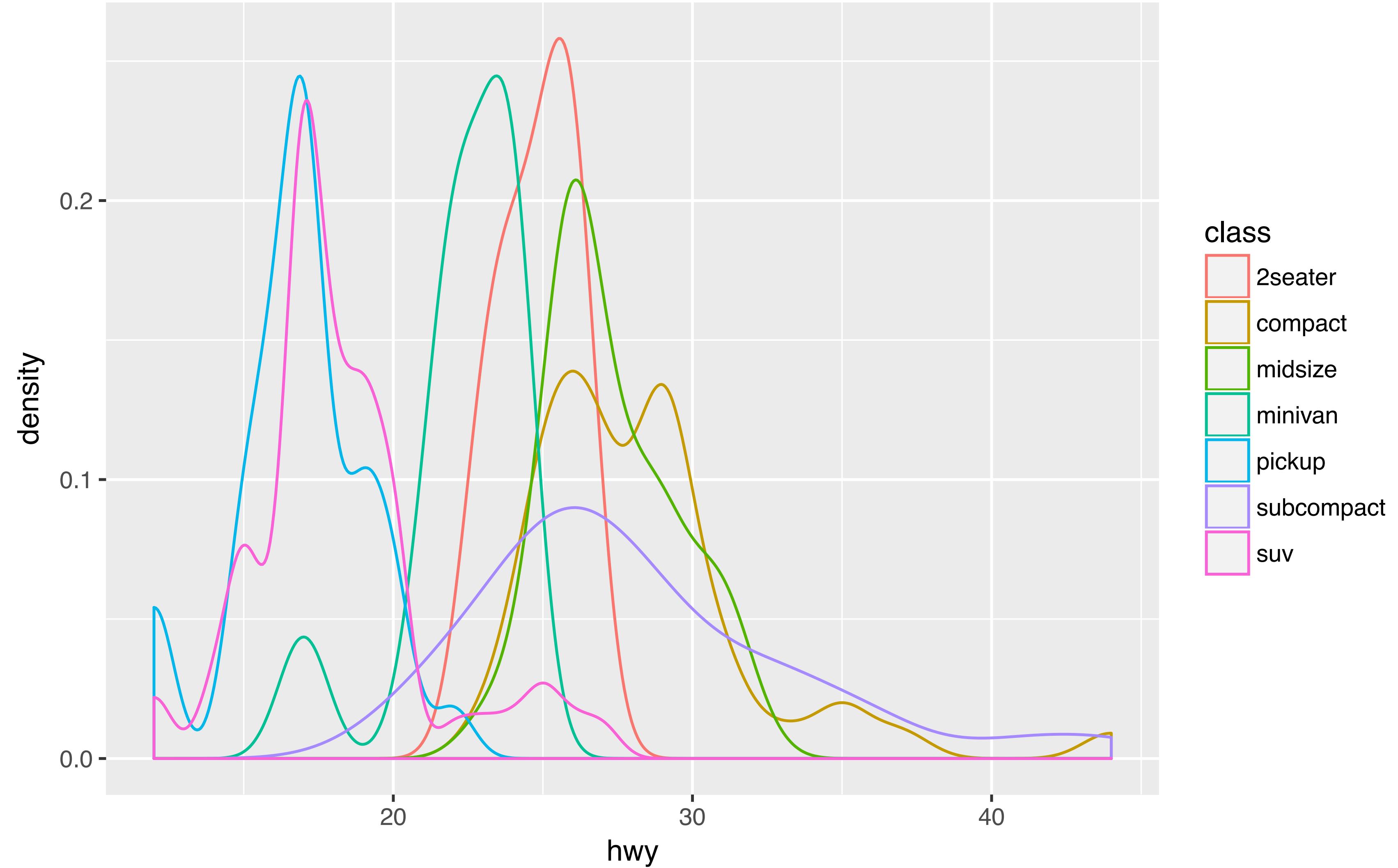
Your Turn 5

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

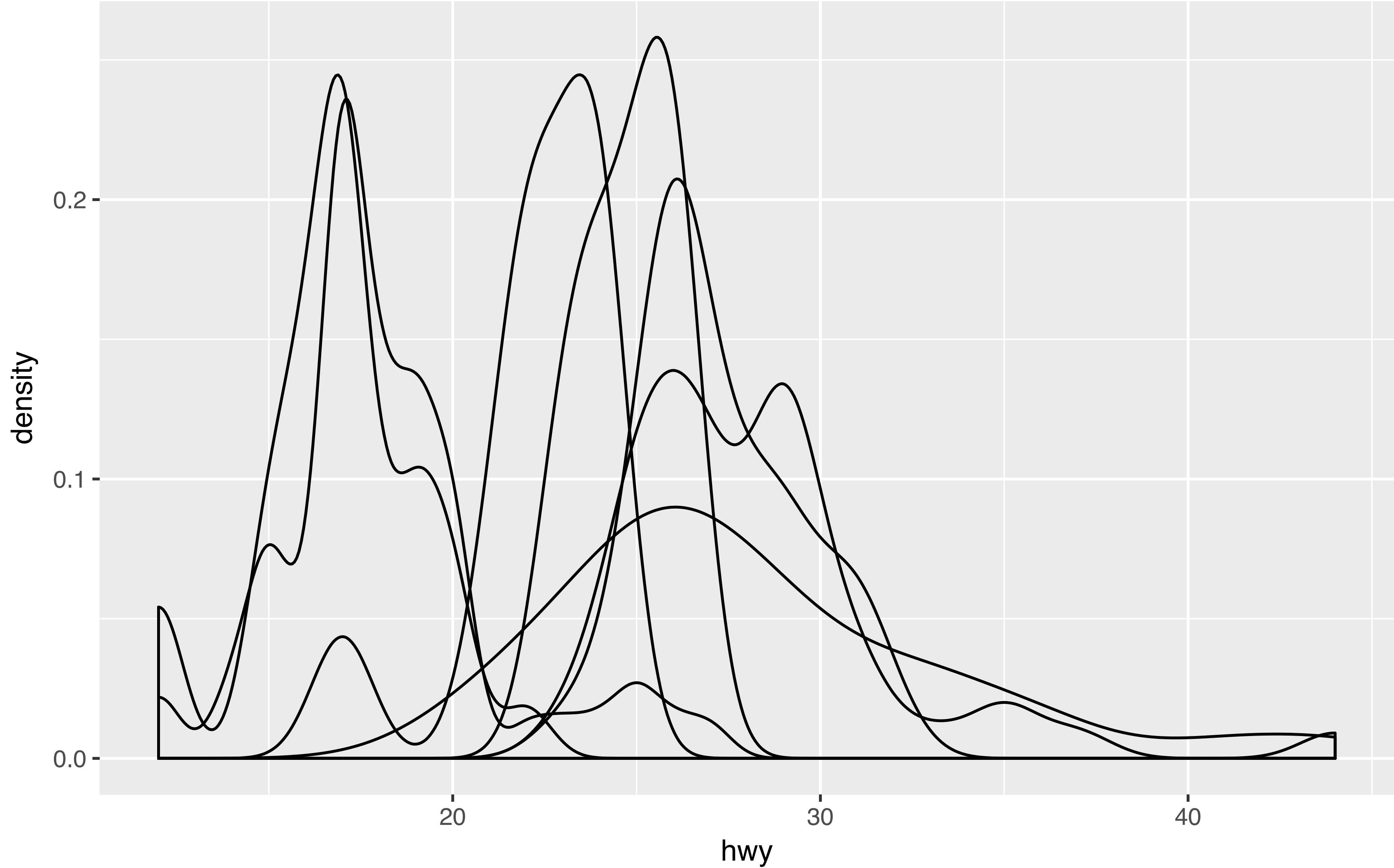




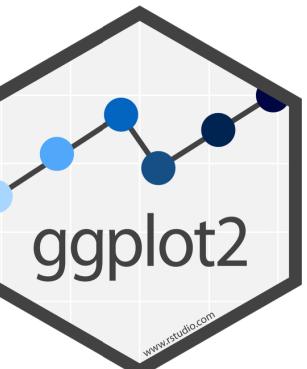
```
ggplot(data = mpg) +  
  geom_density(mapping = aes(x = hwy))
```



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

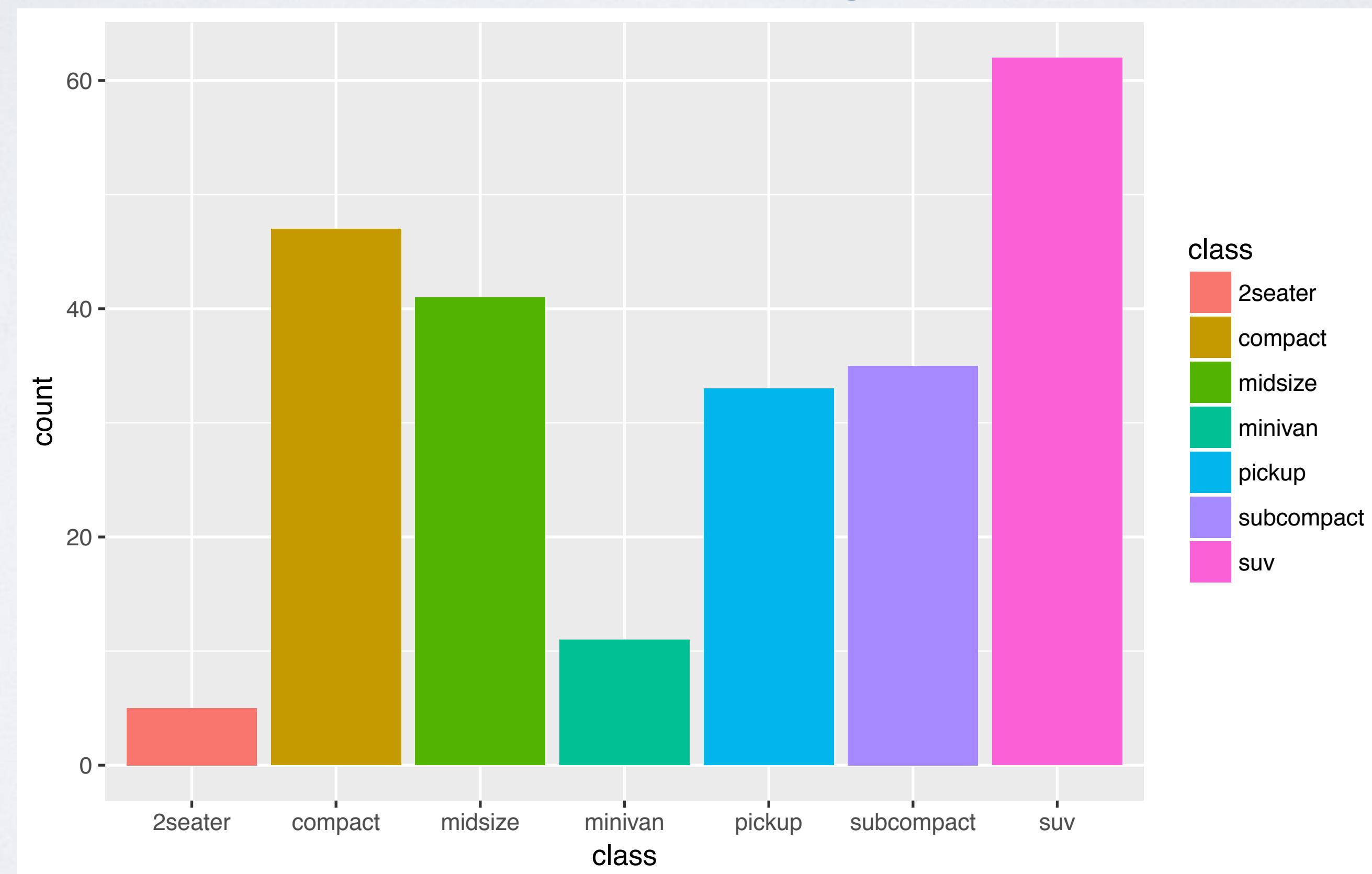


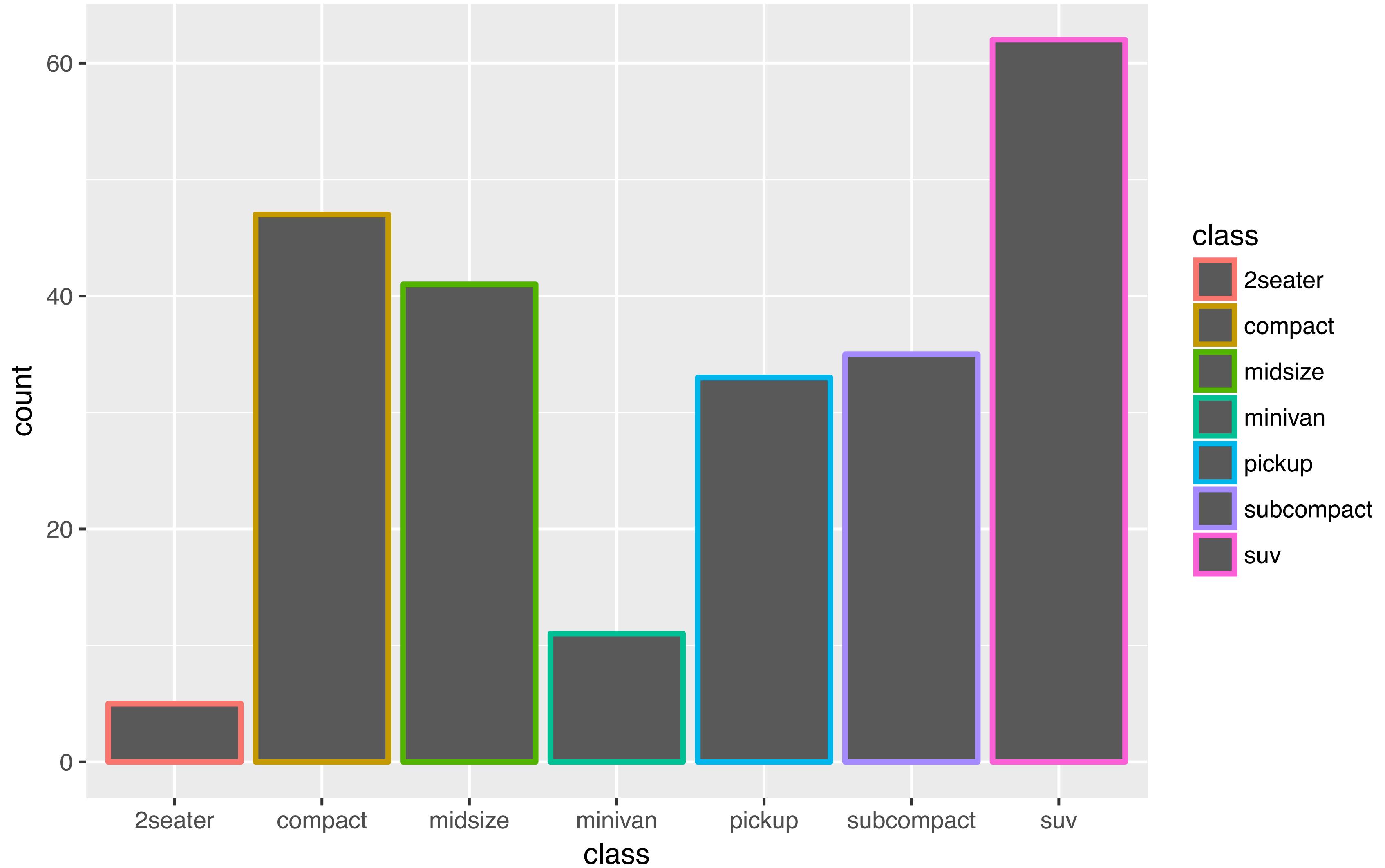
```
ggplot(data = mpg) +  
  geom_density(mapping = aes(x = hwy, group = class))
```



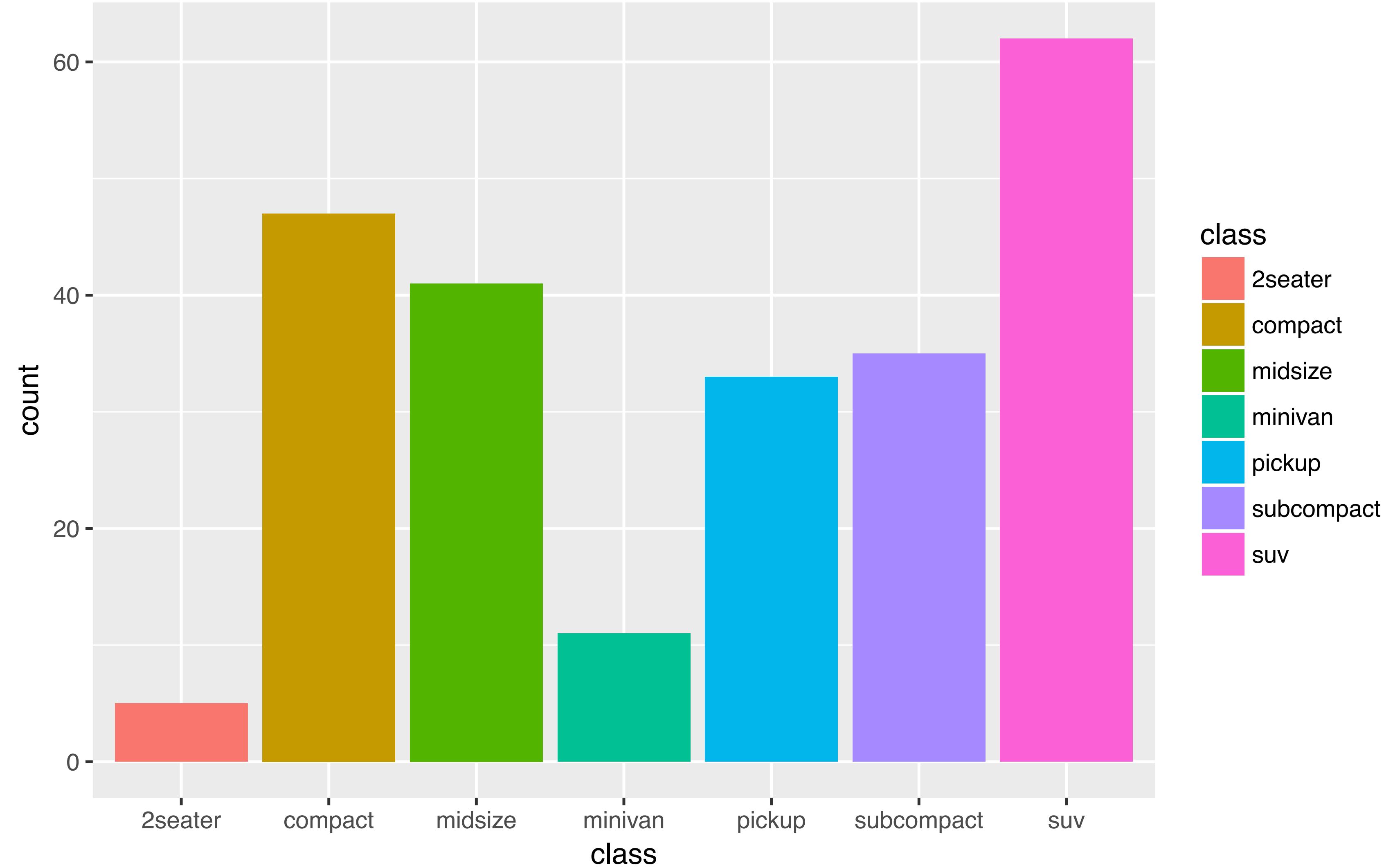
Your Turn 6

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))
```



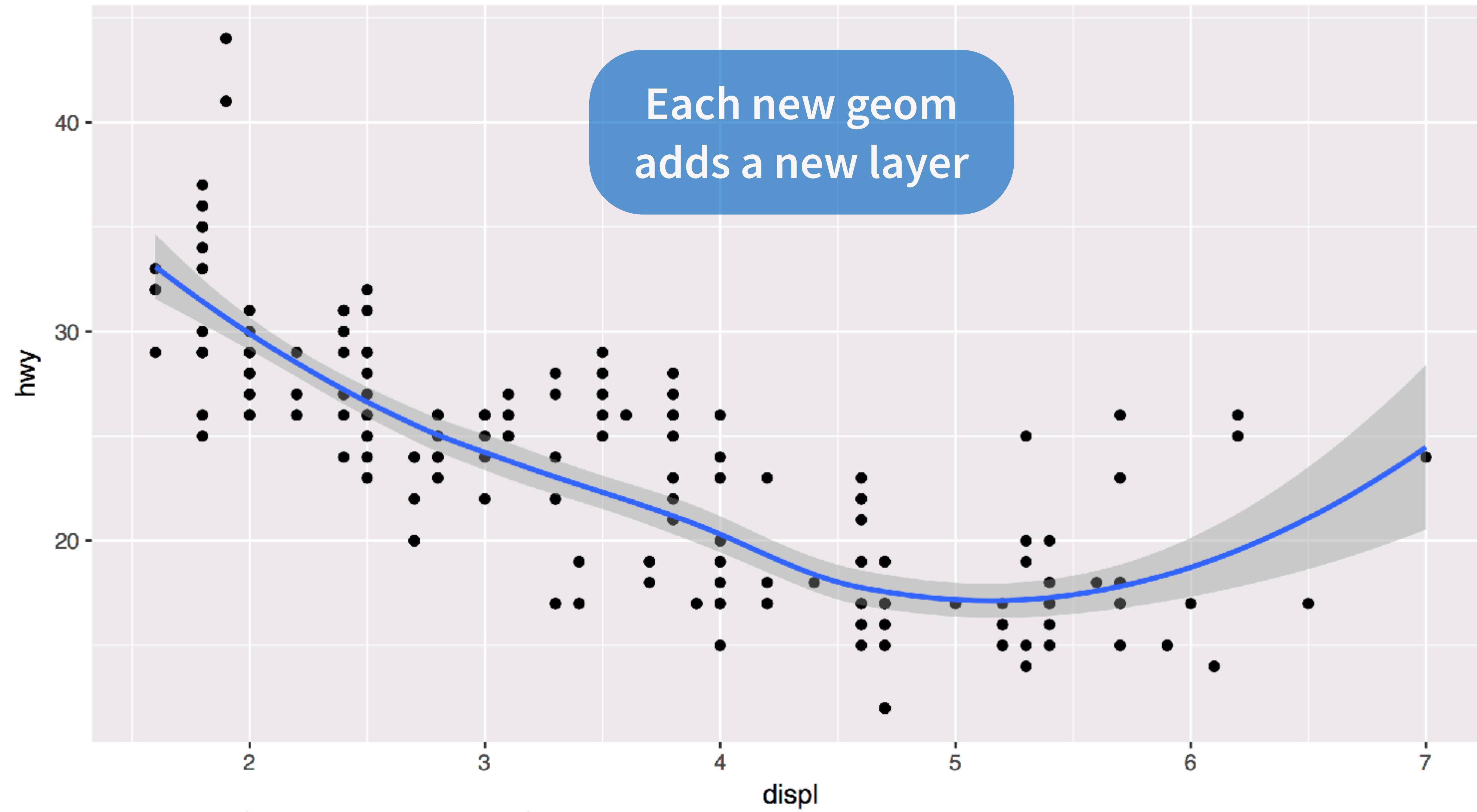
```
ggplot(data = mpg) +  
  geom_bar(mapping = aes(x = class, fill = class))
```

Your Turn 7

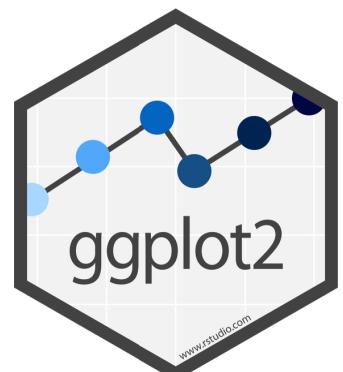
With your partner, predict what this code will do.
Then run it.

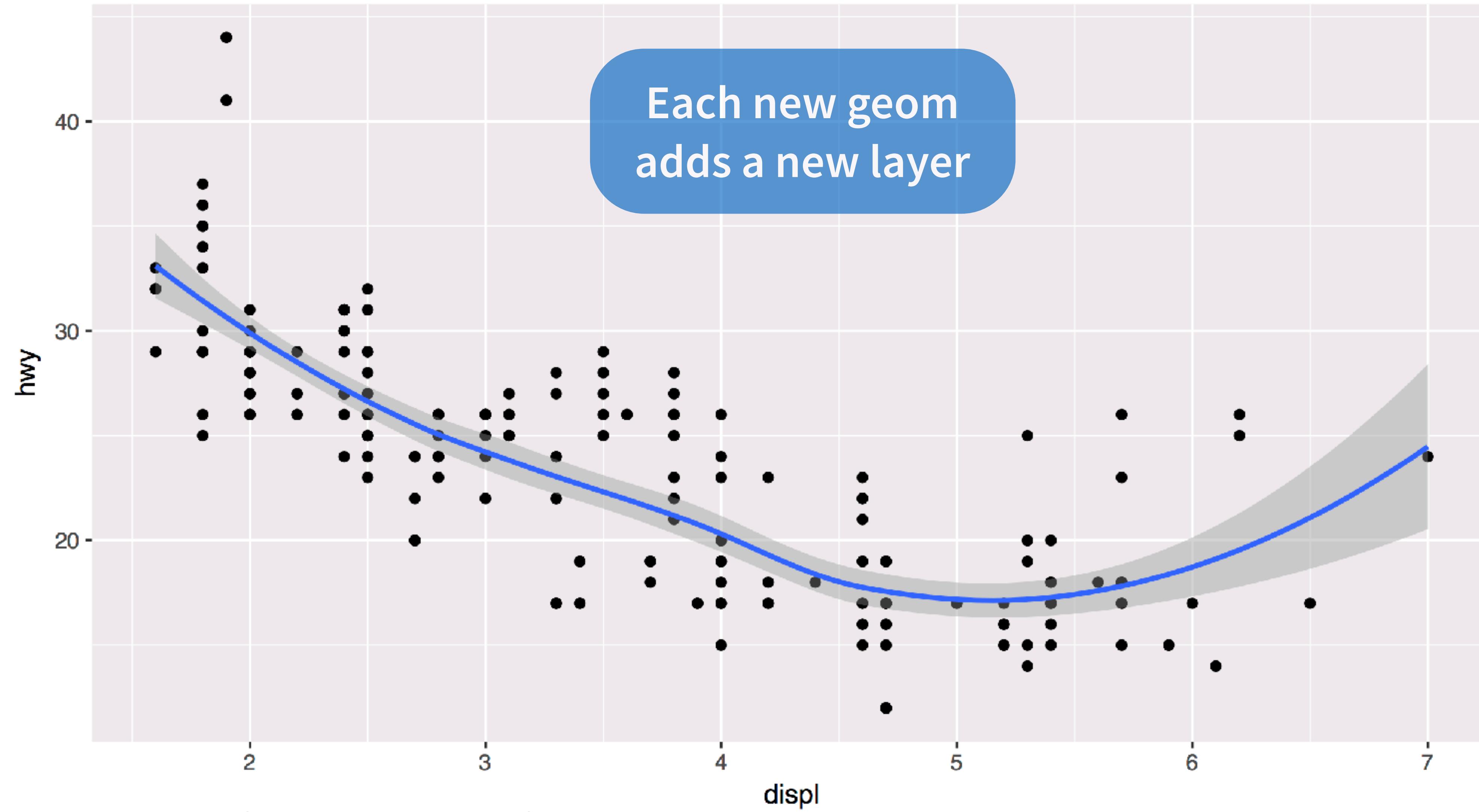
```
ggplot(mpg) +  
  geom_point(aes(displ, hwy)) +  
  geom_smooth(aes(displ, hwy))
```



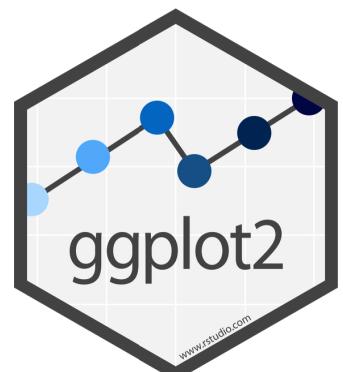


```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

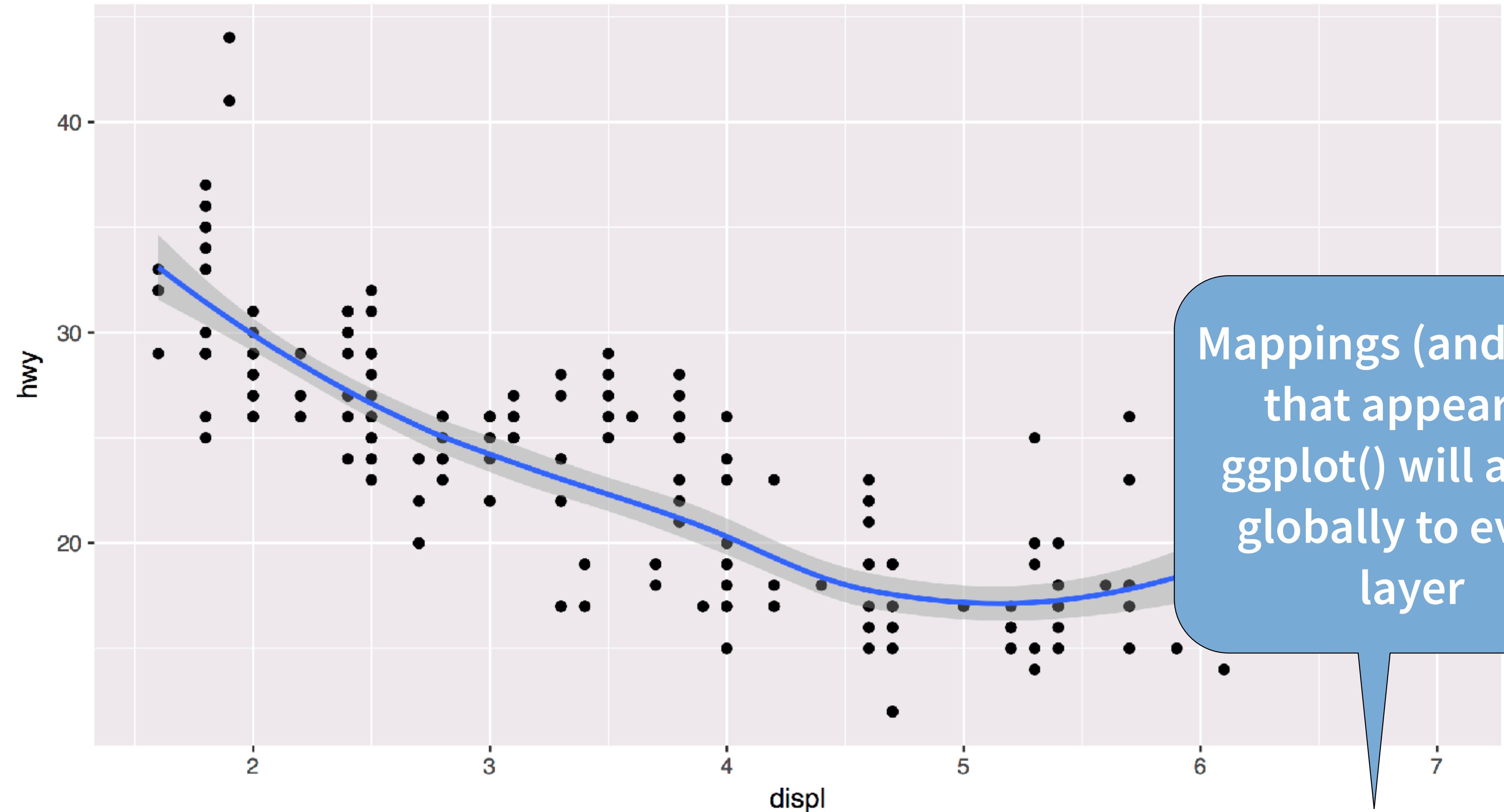




```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

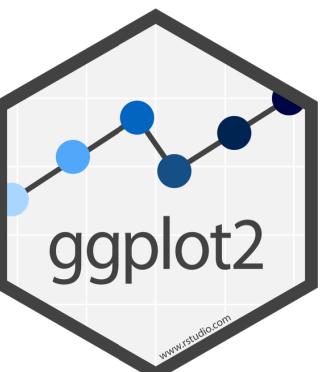


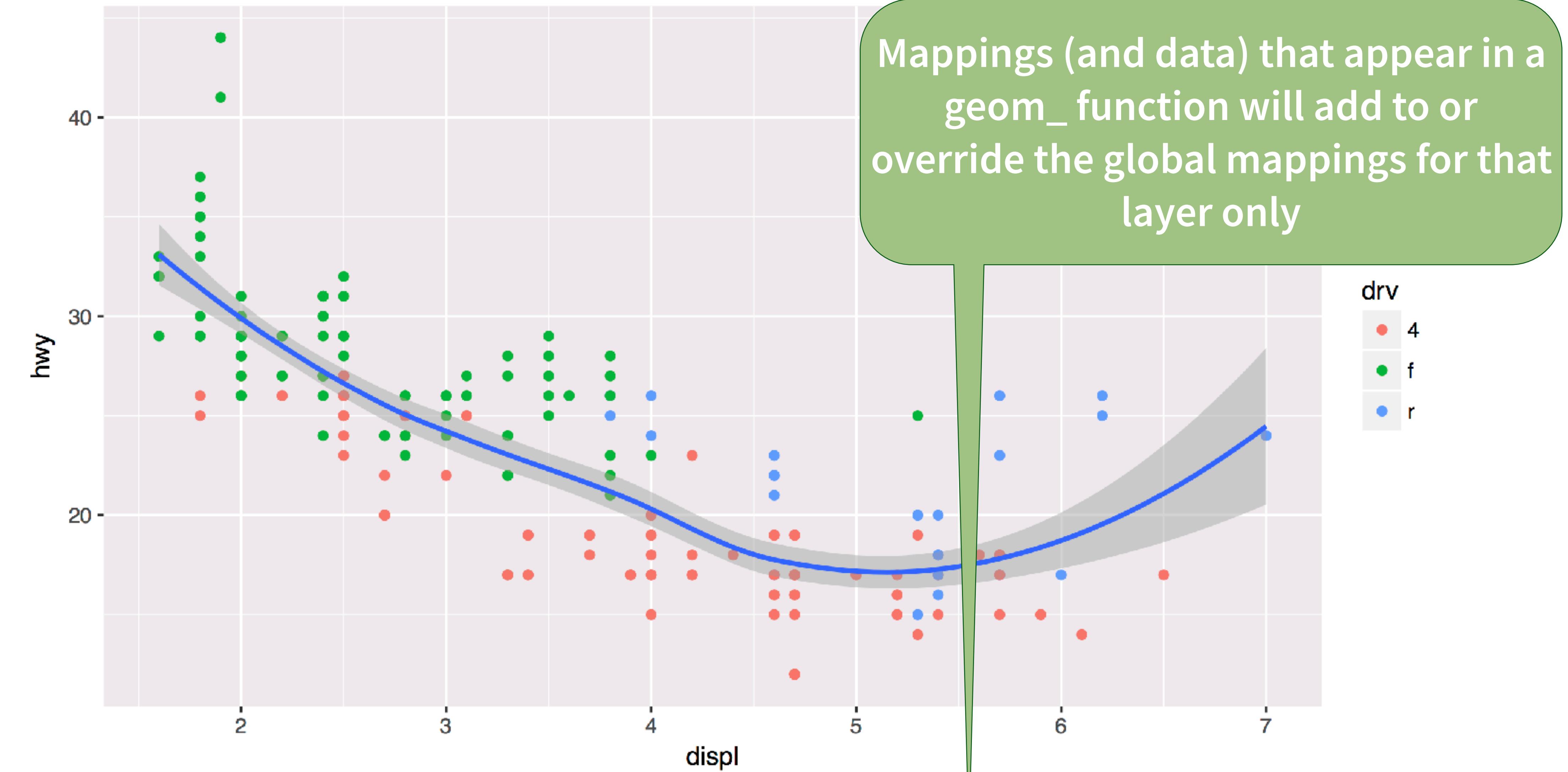
Global vs. Local



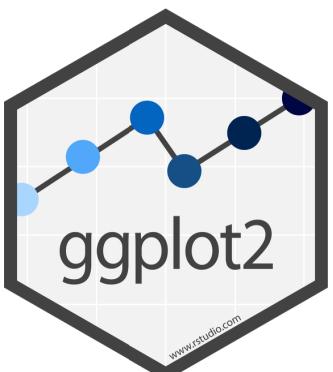
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
```

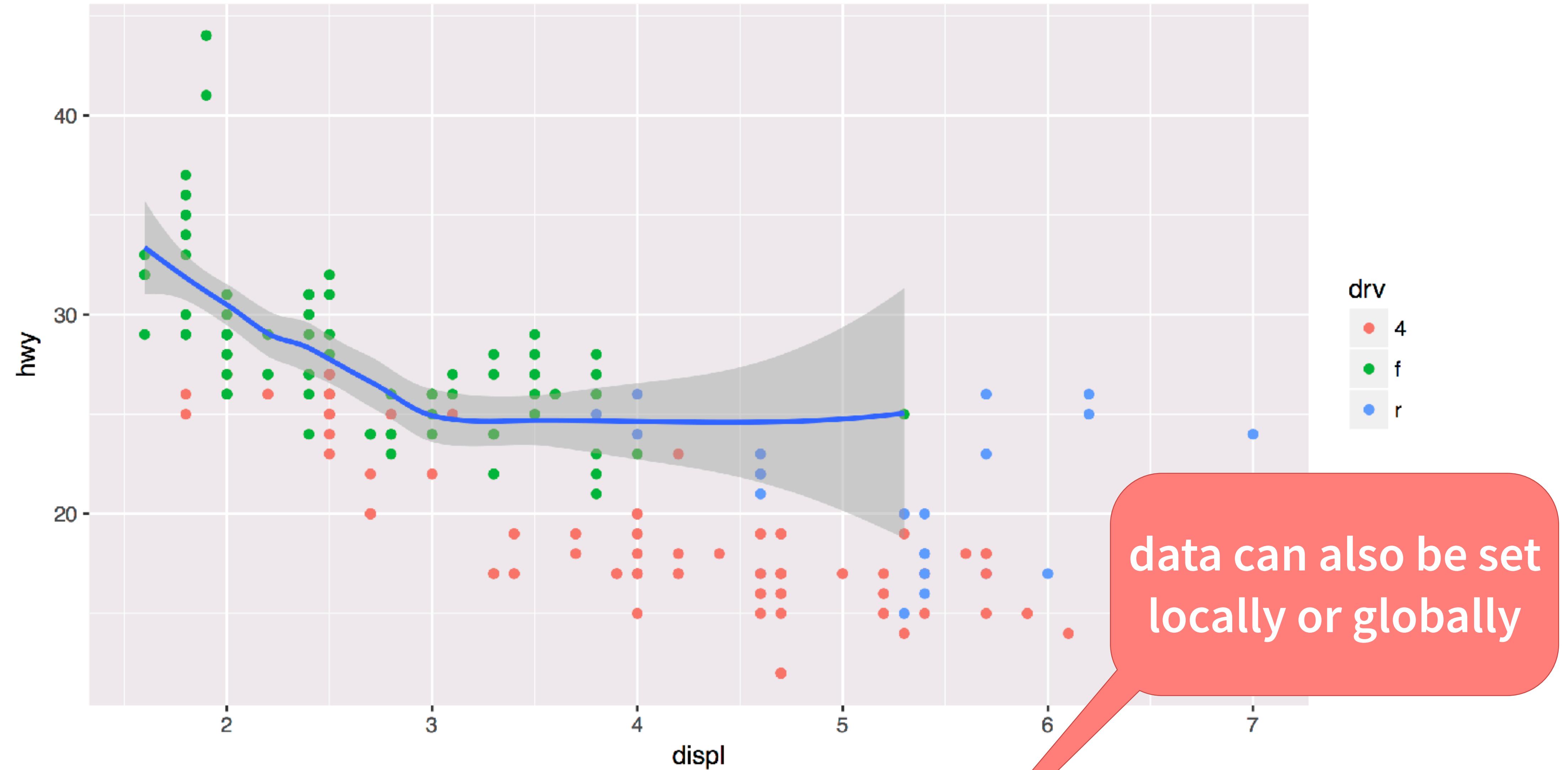
```
geom_point() +  
geom_smooth()
```





```
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

Your Turn 8

What does this command return?

`getwd()`



Working Directory

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

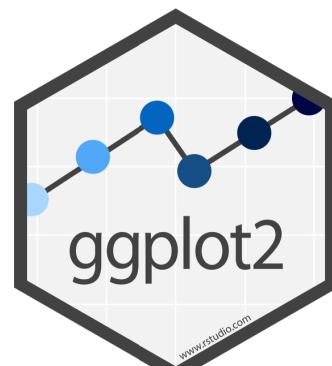
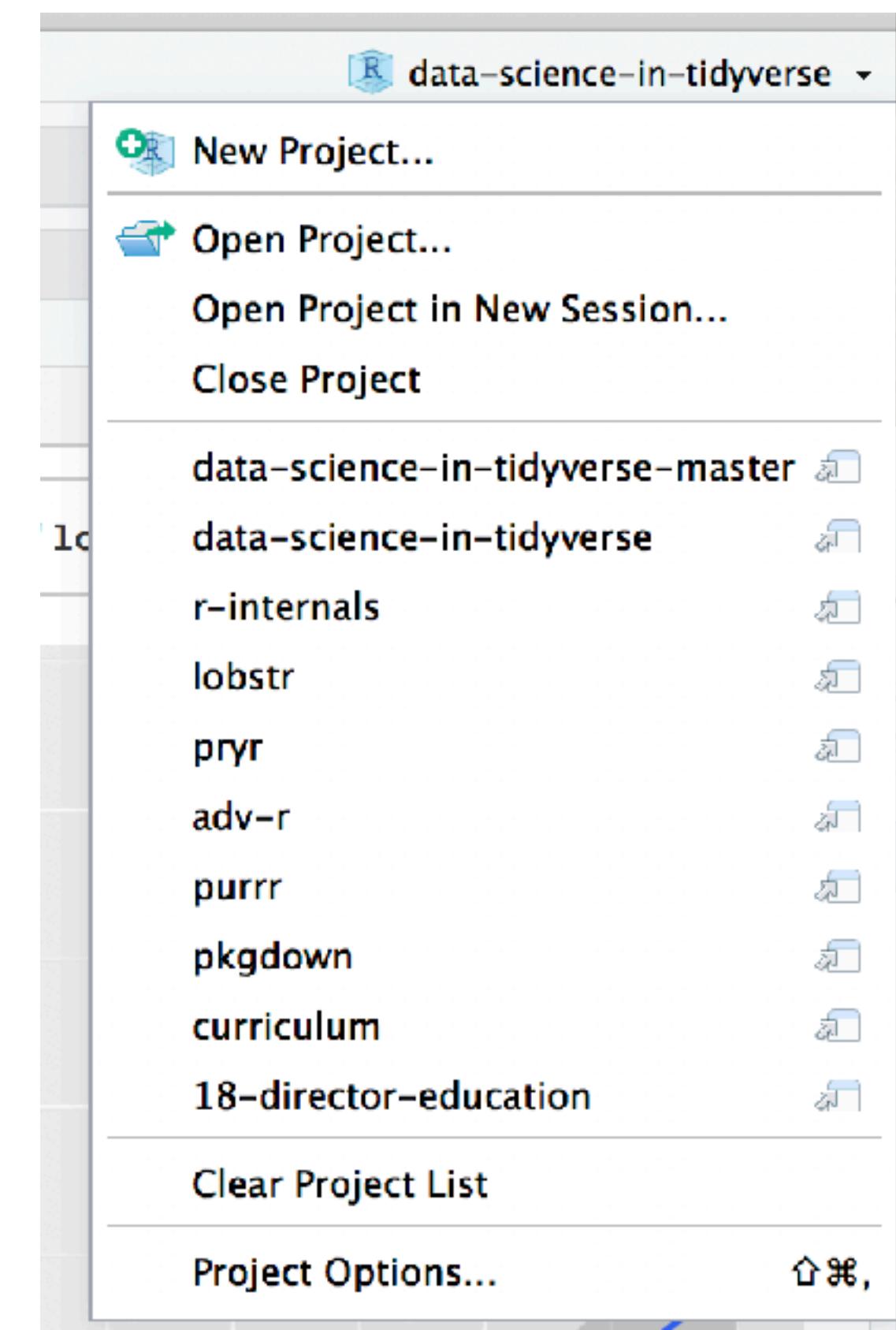
- This folder is known as your "working directory"
- When you save files, R will save them here
- When you load files, R will look for them here

Projects

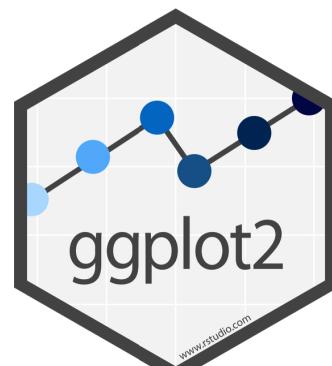
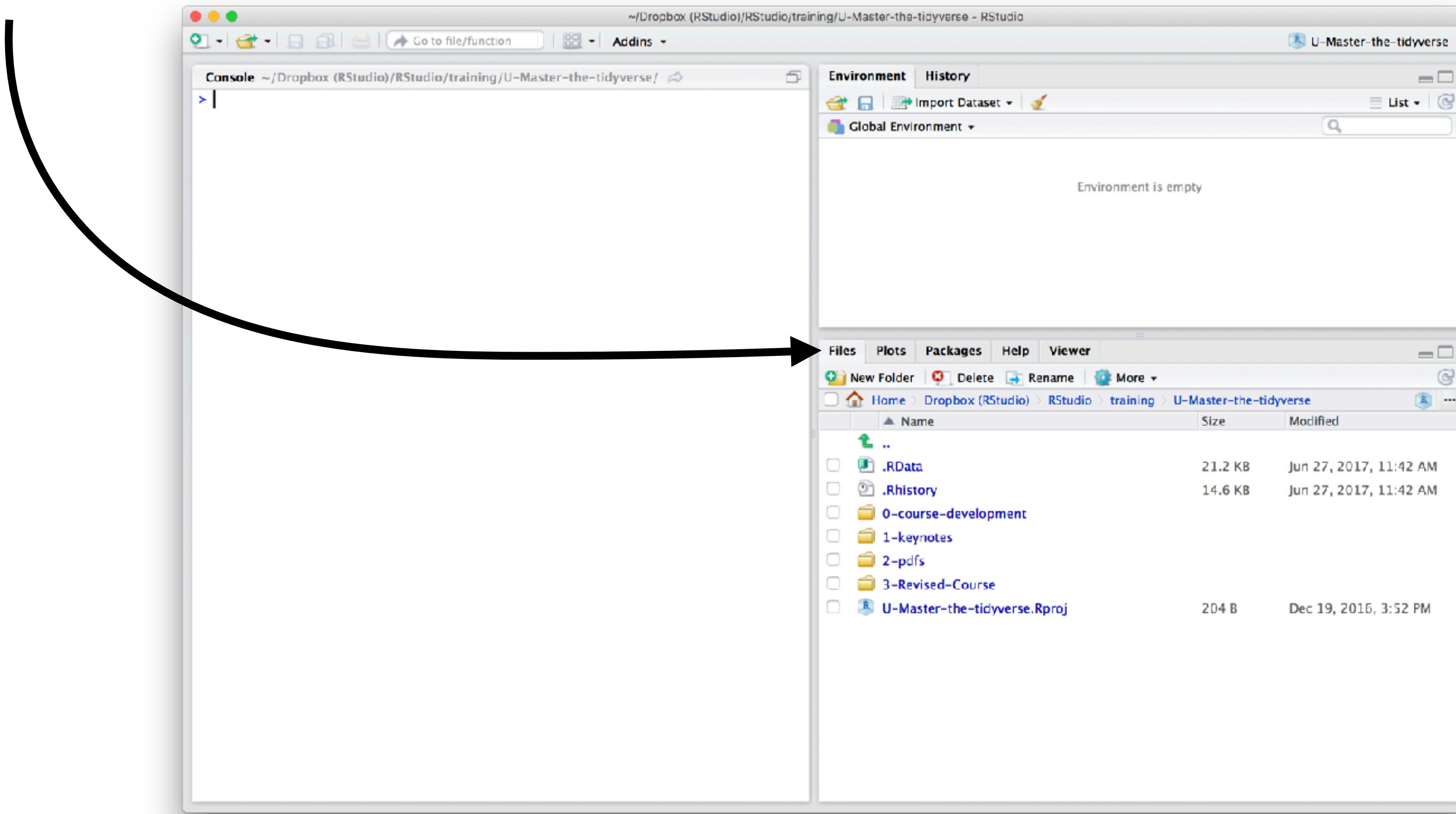
The best way of managing your working directory is with RStudio Projects.

One RStudio project = one real life project

One RStudio project = one directory



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



Saving plots

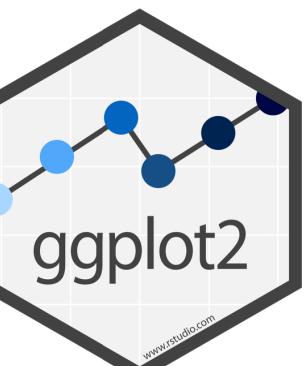
ggsave() saves the last plot.

Uses size on screen:

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

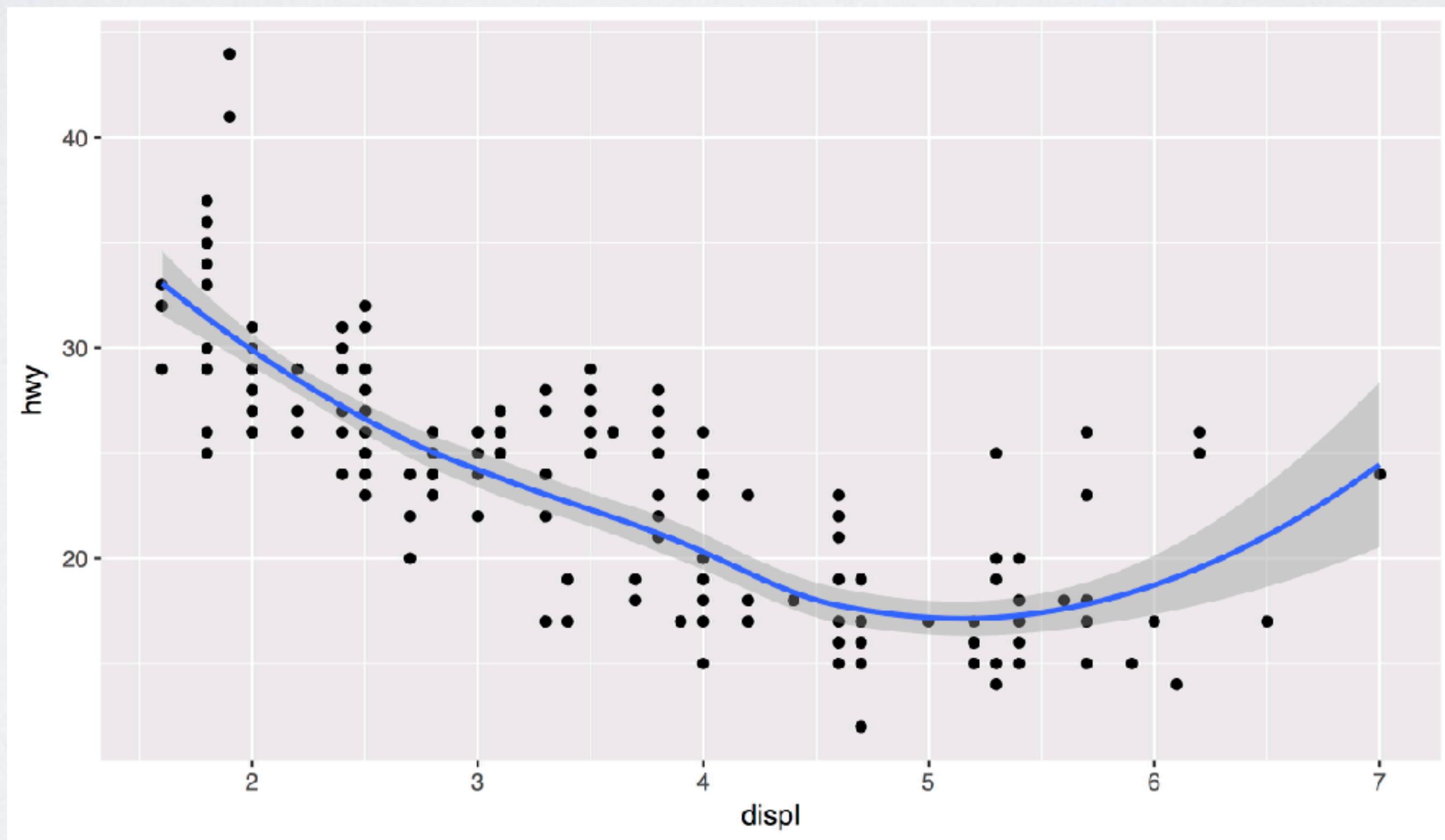
Specify size in inches

```
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>))
```

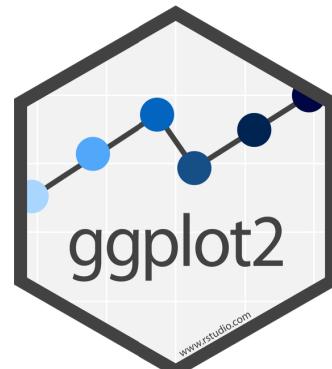
To make a graph

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	●

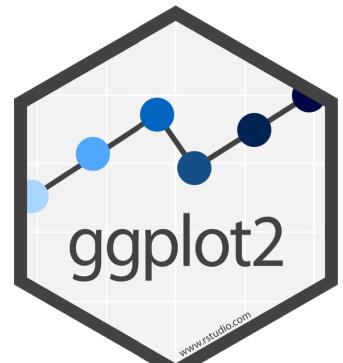
data

geom

1. Pick a **data** set

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

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



mappings

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

data

geom

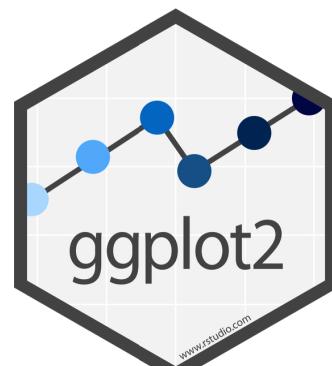
To make a graph

1. Pick a **data** set

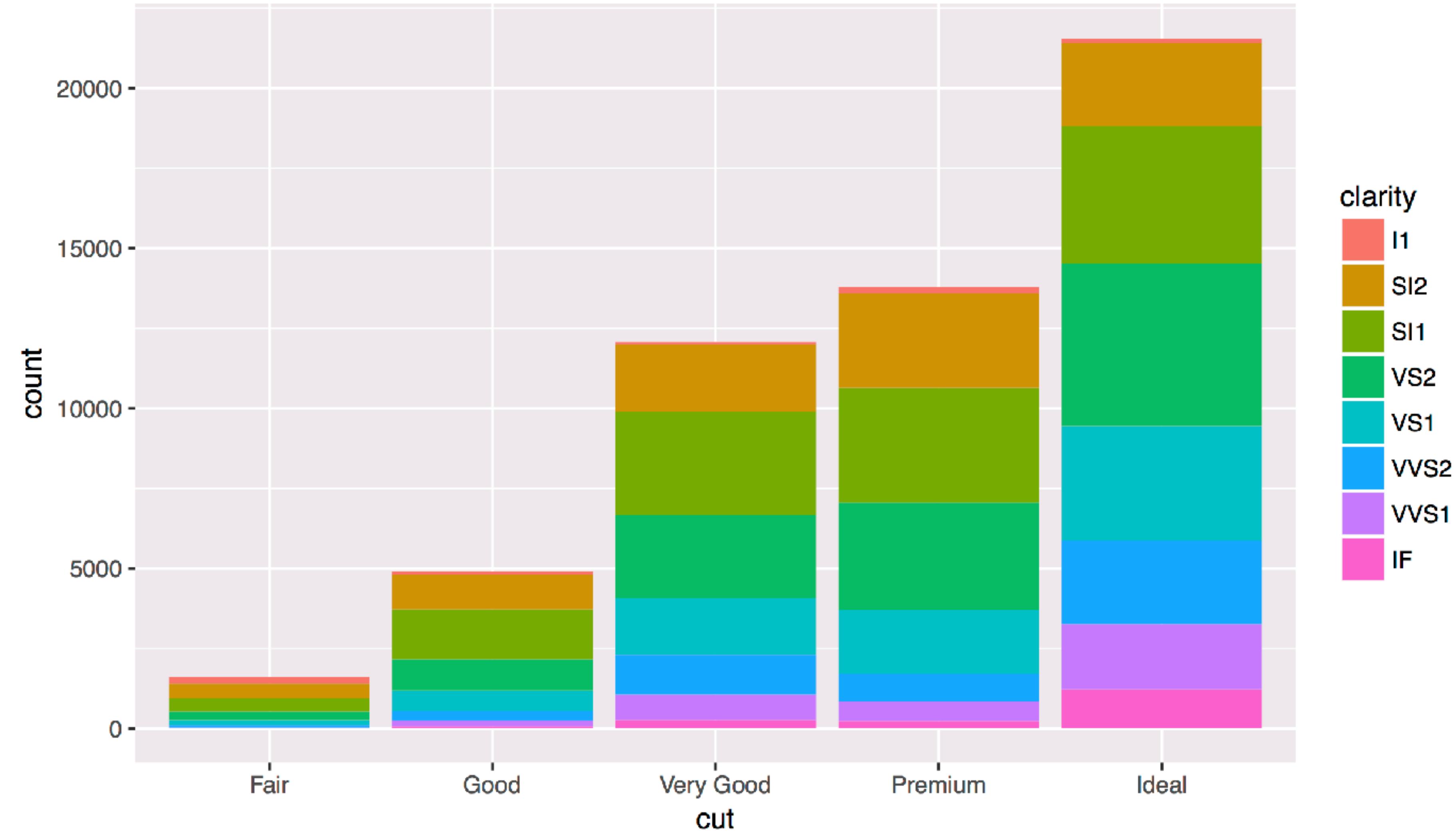
```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

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

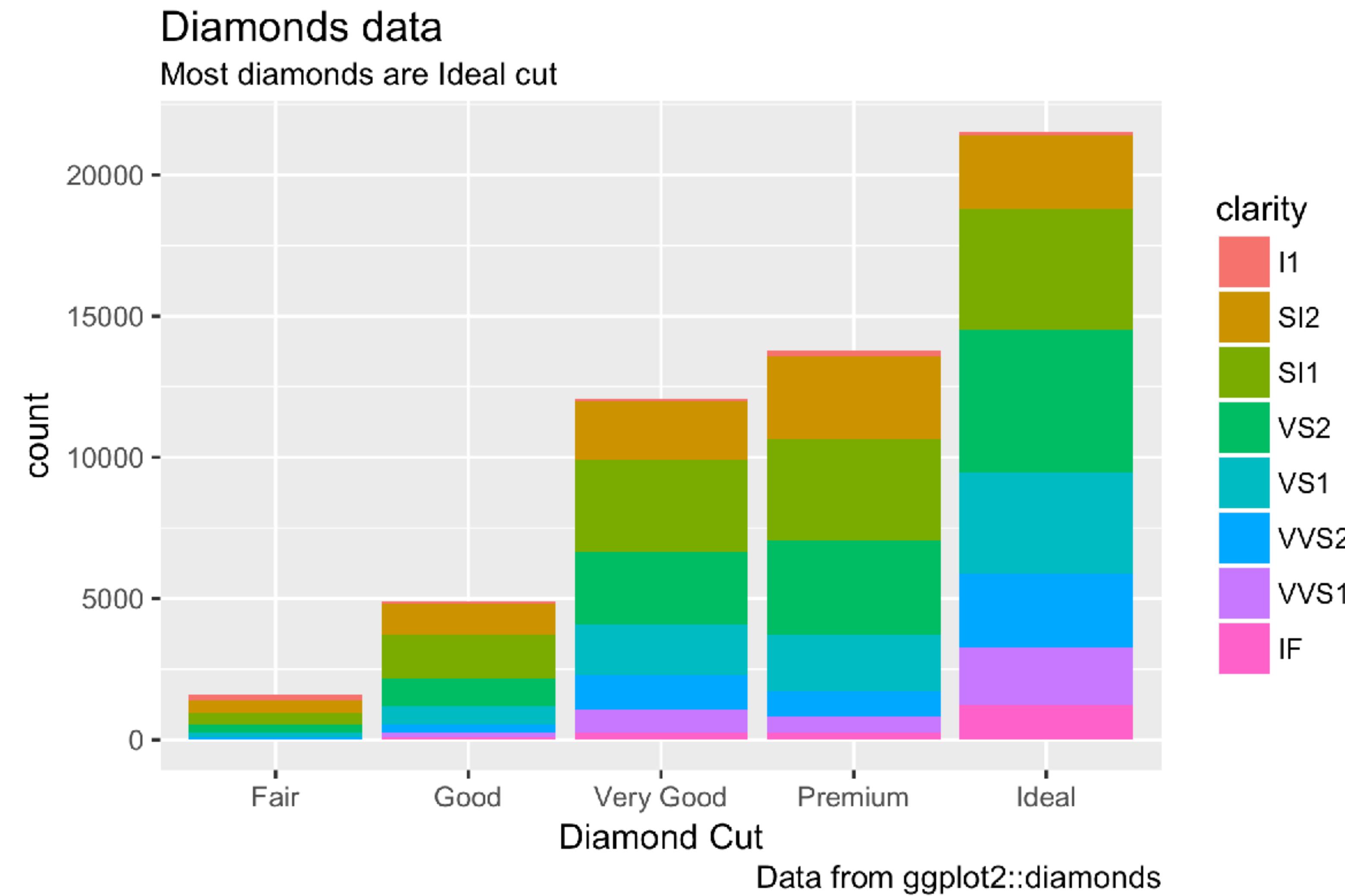
3. **Map** aesthetic
properties to
variables



What else?

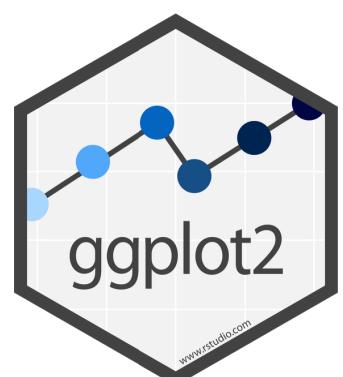
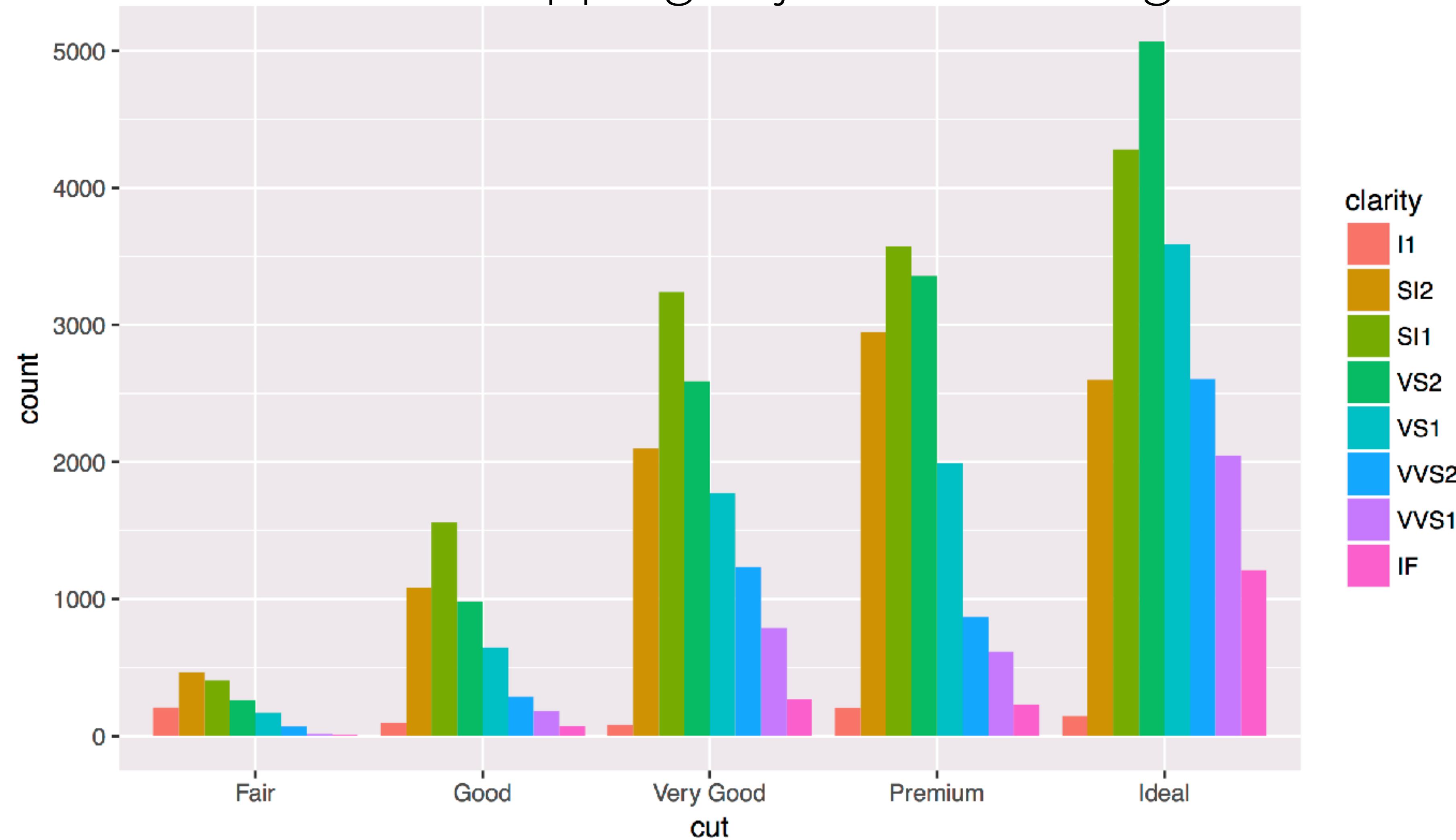


Titles and captions + labs()



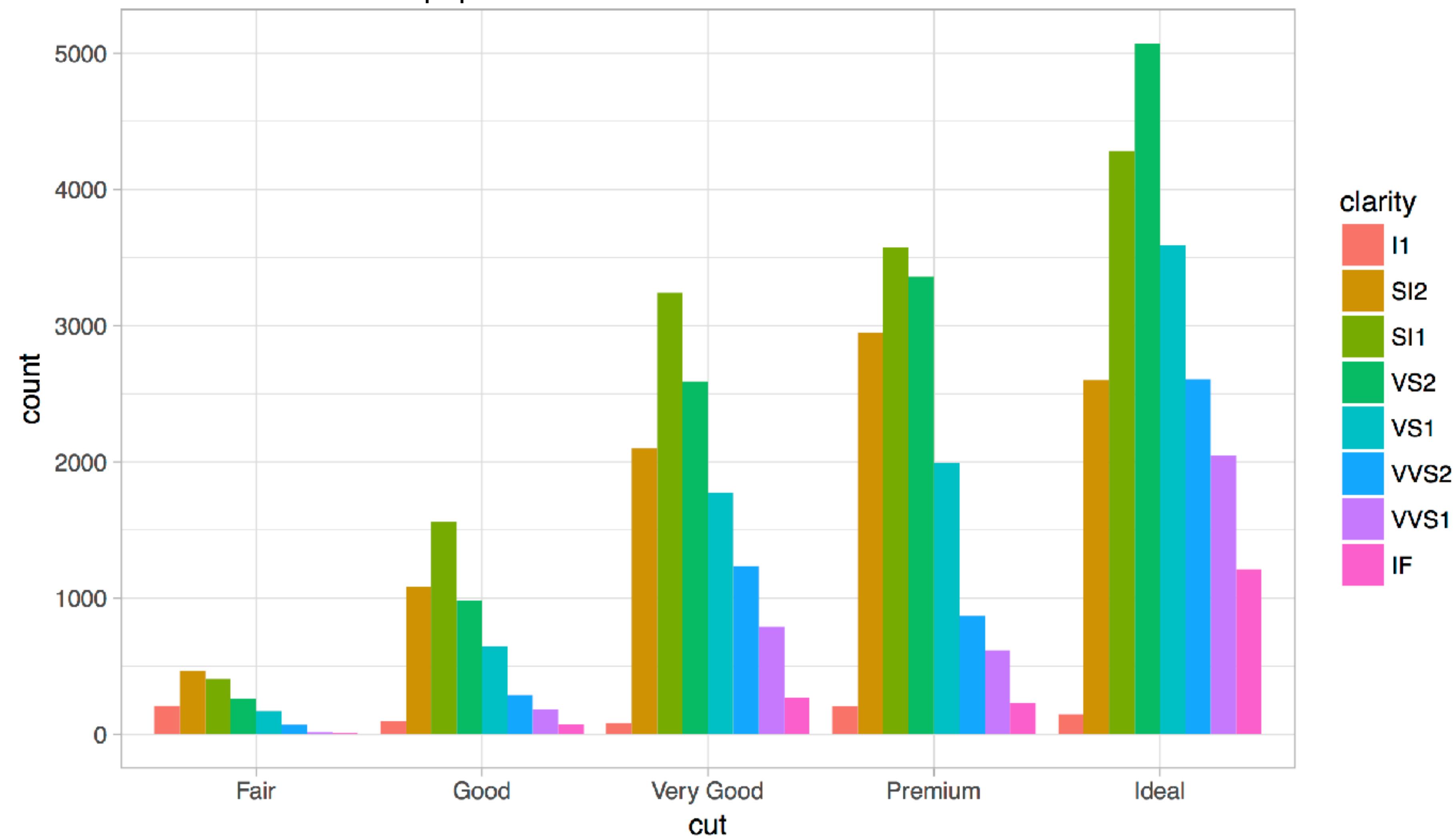
Position Adjustments

How overlapping objects are arranged



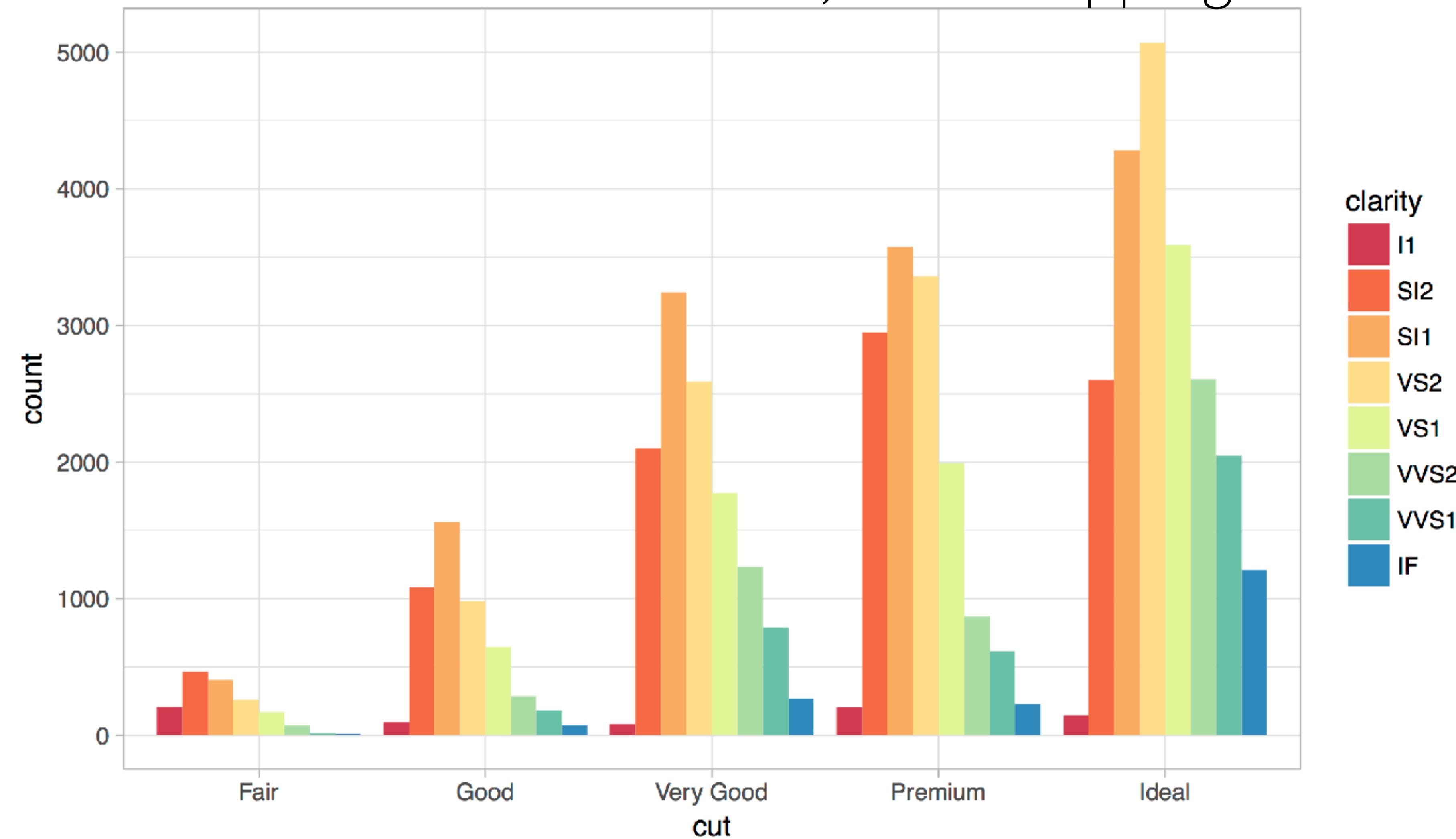
Themes

Visual appearance of non-data elements



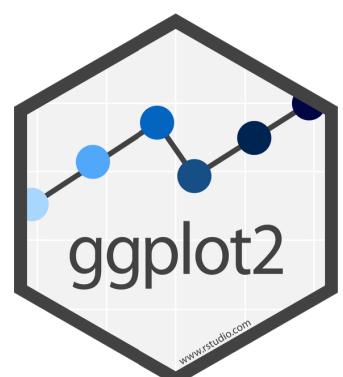
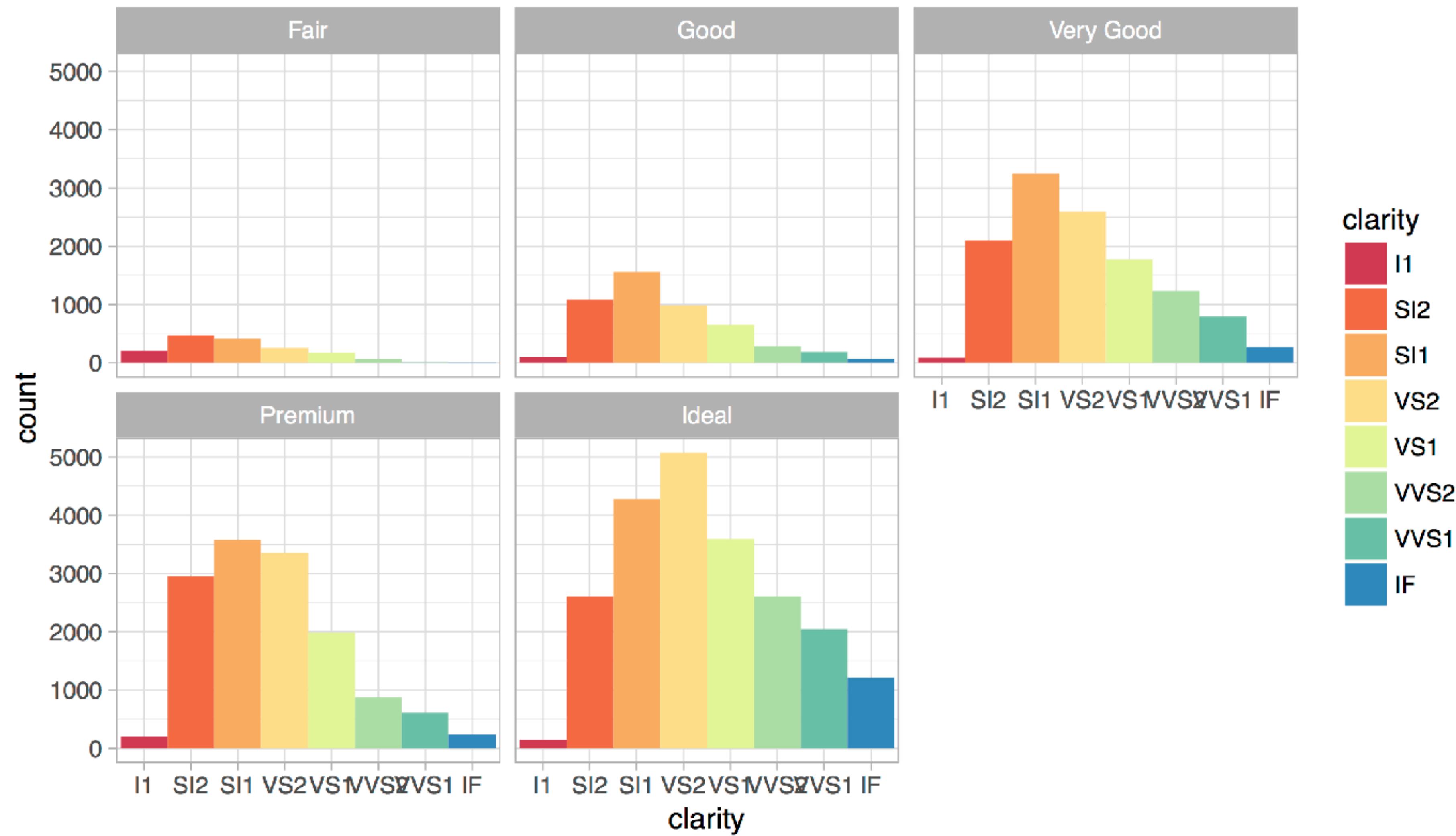
Scales

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

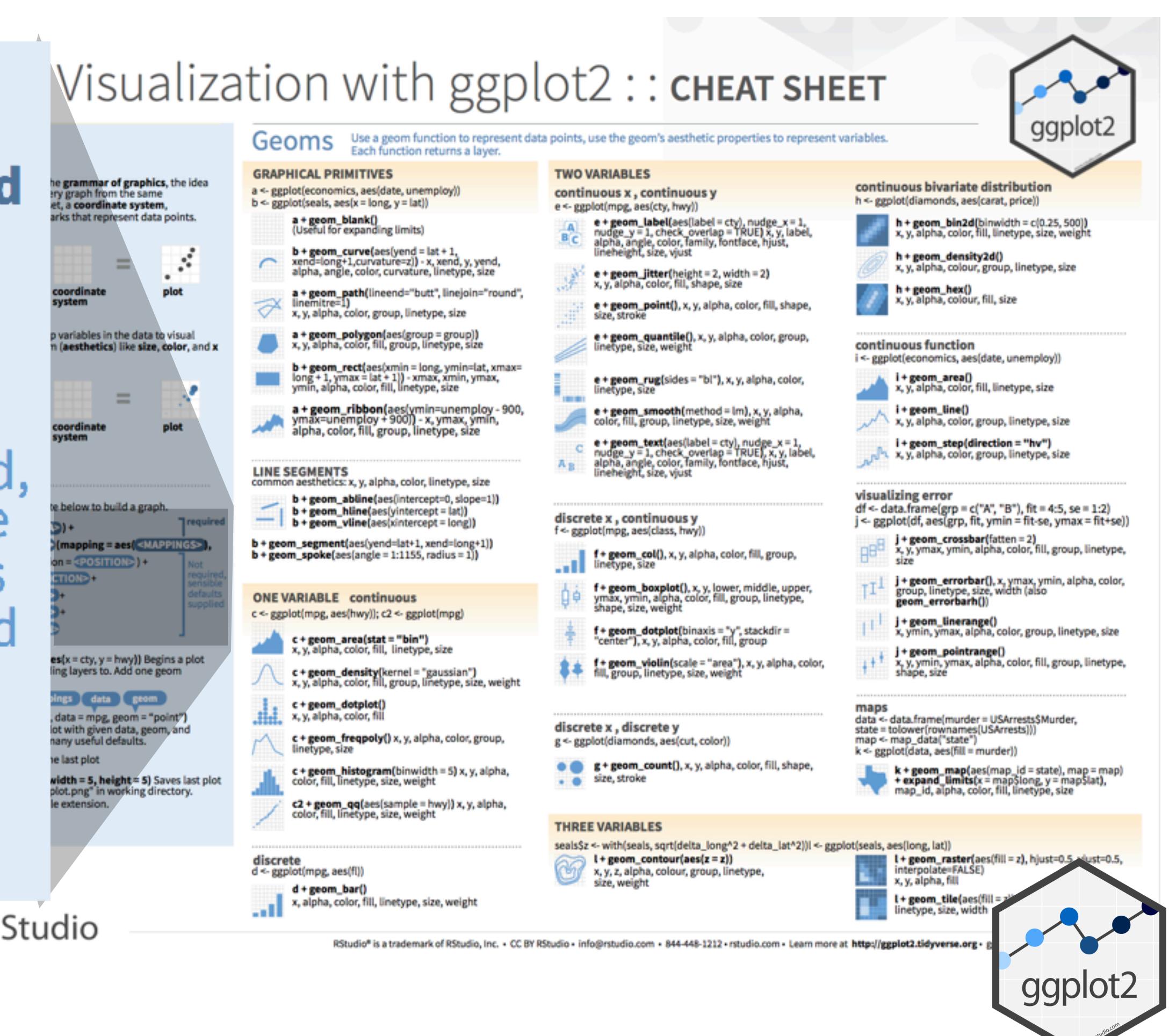
Complete the template below to build a graph.

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



The grammar of graphics, the idea of a graph from the same set, a coordinate system, marks that represent data points.

variables in the data to visualise in aesthetics like size, color, and x

coordinate system = plot

coordinate system = plot

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

LINE SEGMENTS

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

ONE VARIABLE continuous

maps

THREE VARIABLES

R Studio

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.tidyverse.org

The screenshot shows a web browser window displaying the ggplot2.tidyverse.org website. The title bar reads "Create Elegant Data Visualisati x" and "Garrett". The address bar shows the URL "ggplot2.tidyverse.org". The page content includes a header with the ggplot2 logo and "part of the tidyverse". A main section titled "Usage" contains text about the philosophy of ggplot2 and a code snippet. To the right, there are "Links" to CRAN, source code, bug reports, and more. At the bottom, there is a plot and developer information.

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()
```

A scatter plot showing fuel efficiency (mpg) on the y-axis versus engine displacement (displ) on the x-axis. The plot includes a legend indicating that purple dots represent the '2seater' class.

class

2seater

Links

- Download from CRAN at <https://cran.r-project.org/package=ggplot2>
- Browse source code at <https://github.com/tidyverse/ggplot2>
- Report a bug at <https://github.com/tidyverse/ggplot2/issues>
- Learn more at <http://r4ds.had.co.nz/data-visualisation.html>

License

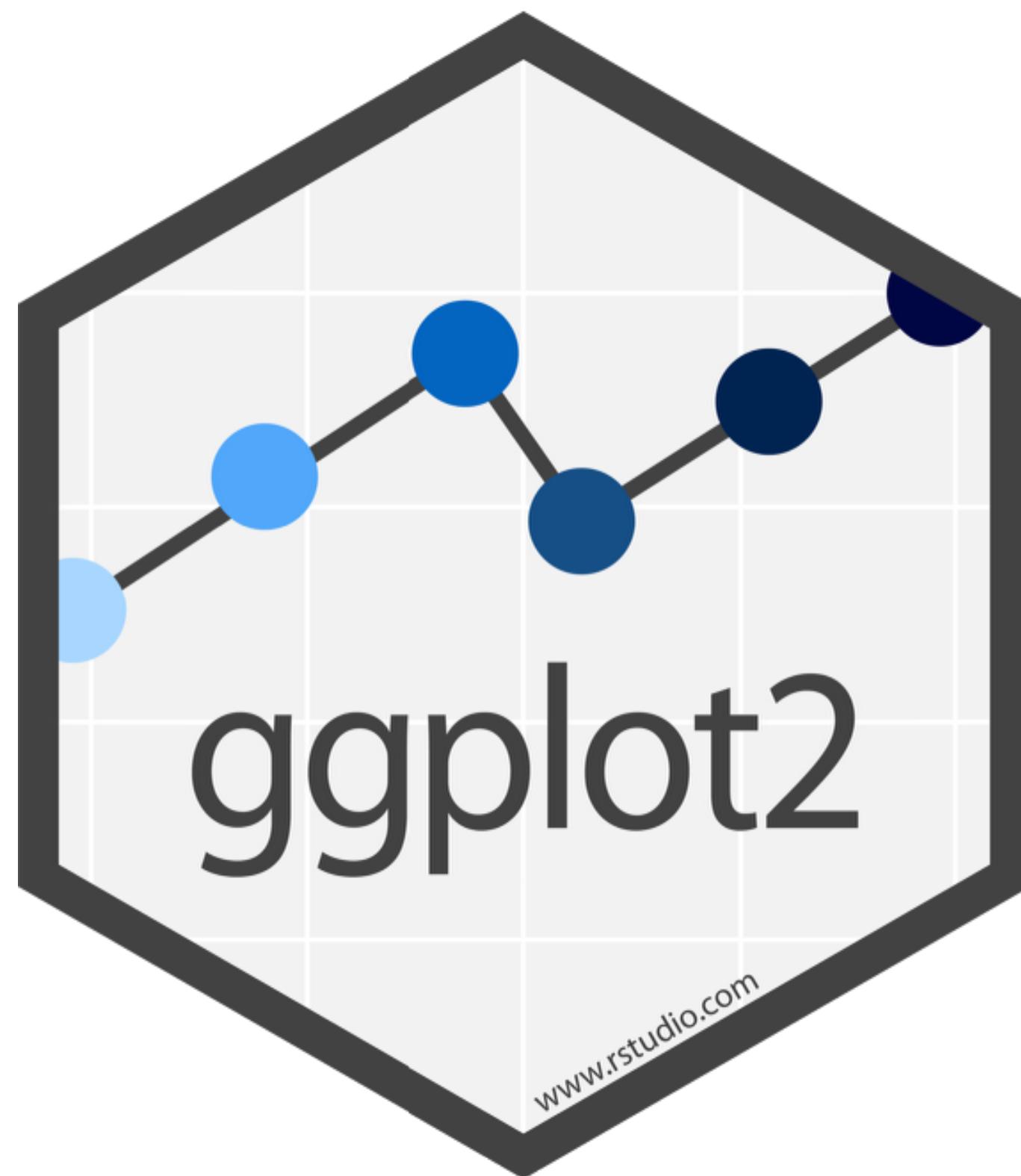
[GPL-2](#) | file [LICENSE](#)

Developers

Hadley Wickham
Author/maintainer

ggplot2

Visualize Data with



www.rstudio.com