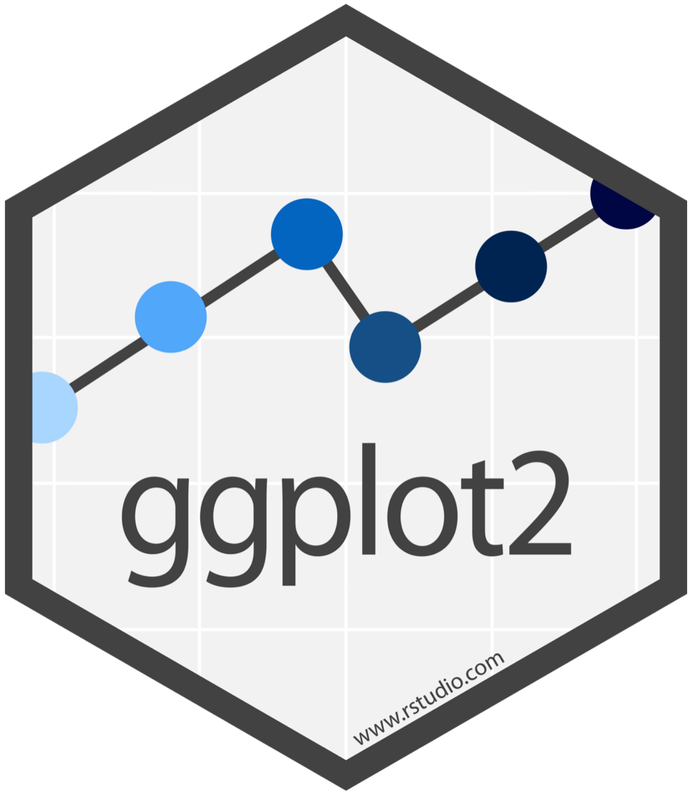
Visualize Data with



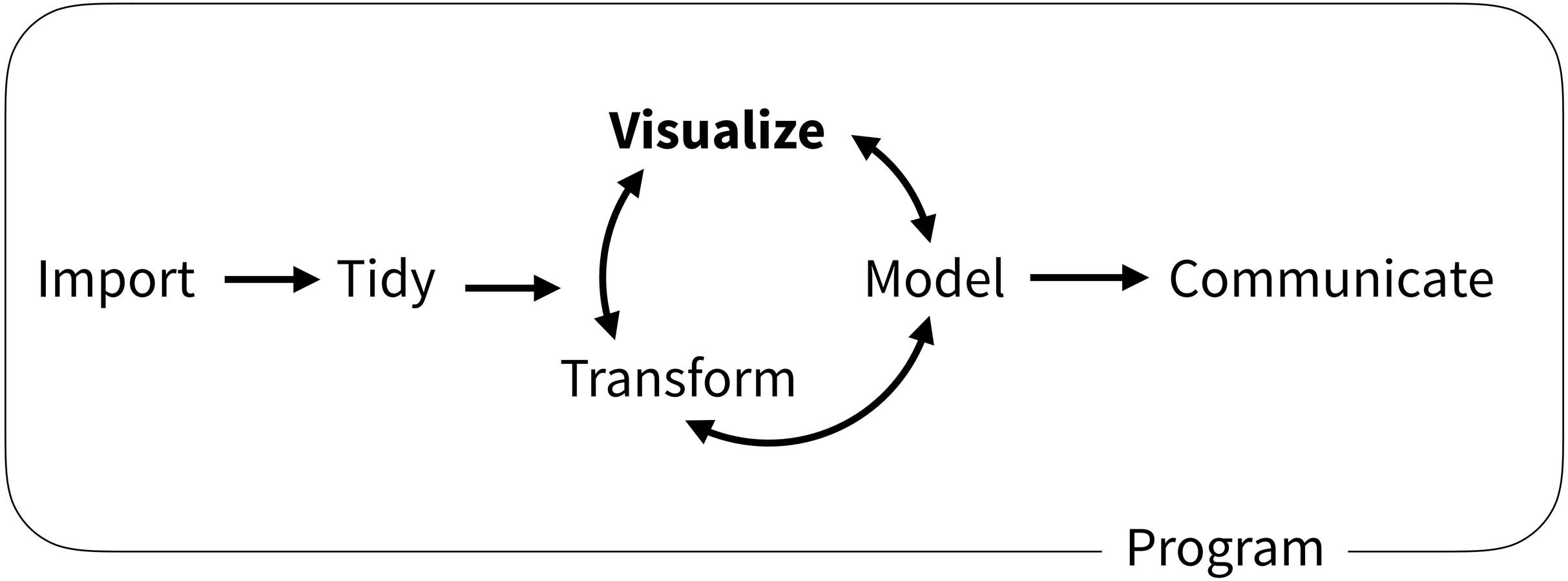
In R4DS

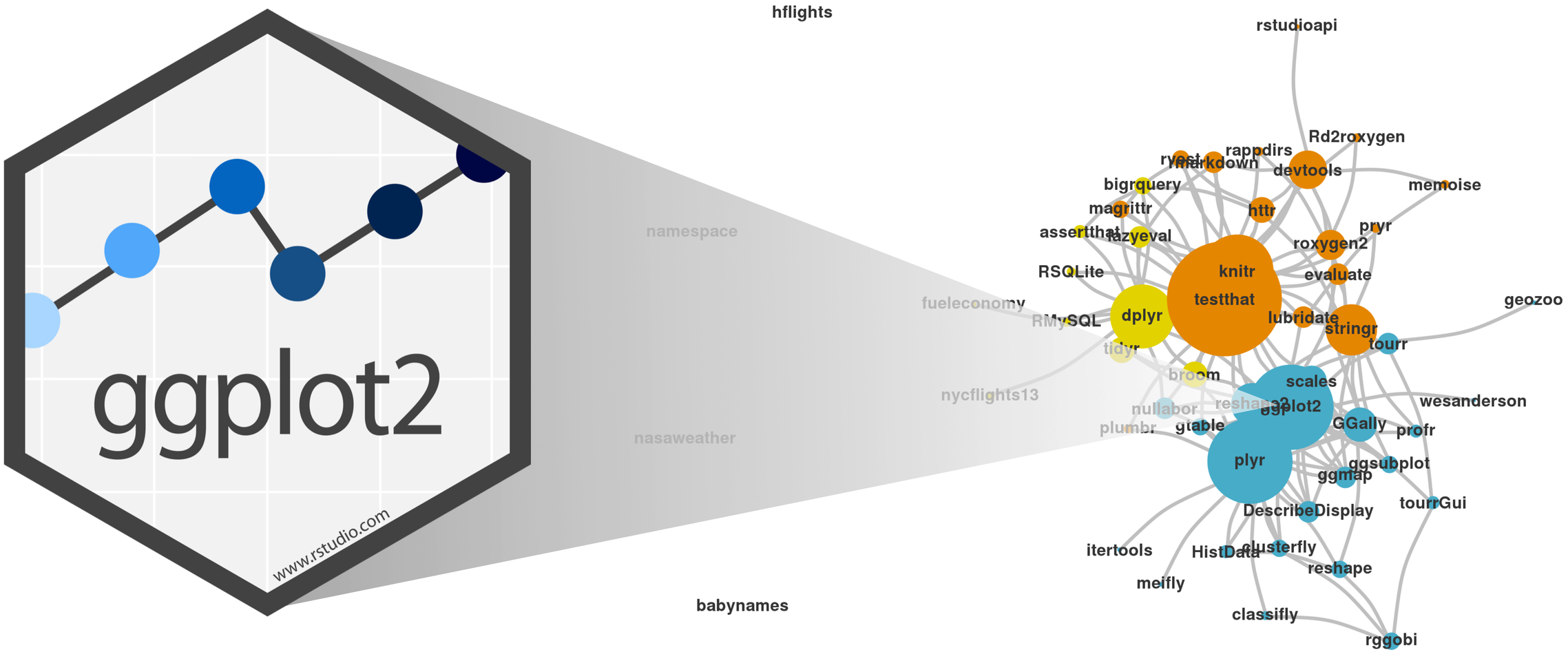
Visualizing Data

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

- John Tukey

(Applied) **Data Science**





Display by Adolfo Álvarez

RStudio

02-Visualize-Data.Rmd

```
---
title: "Visualize Data"
output:
  html_document:
    df_print: paged
---

## Setup

The first chunk in an R Notebook is usually titled "setup," and by convention includes the R packages you want to load. Remember, in order to use an R package you have to run some `library()` code every session. Execute these lines of code to load the packages.

```{r setup}
library(ggplot2)
library(fivethirtyeight)
```

Bechdel test data
```

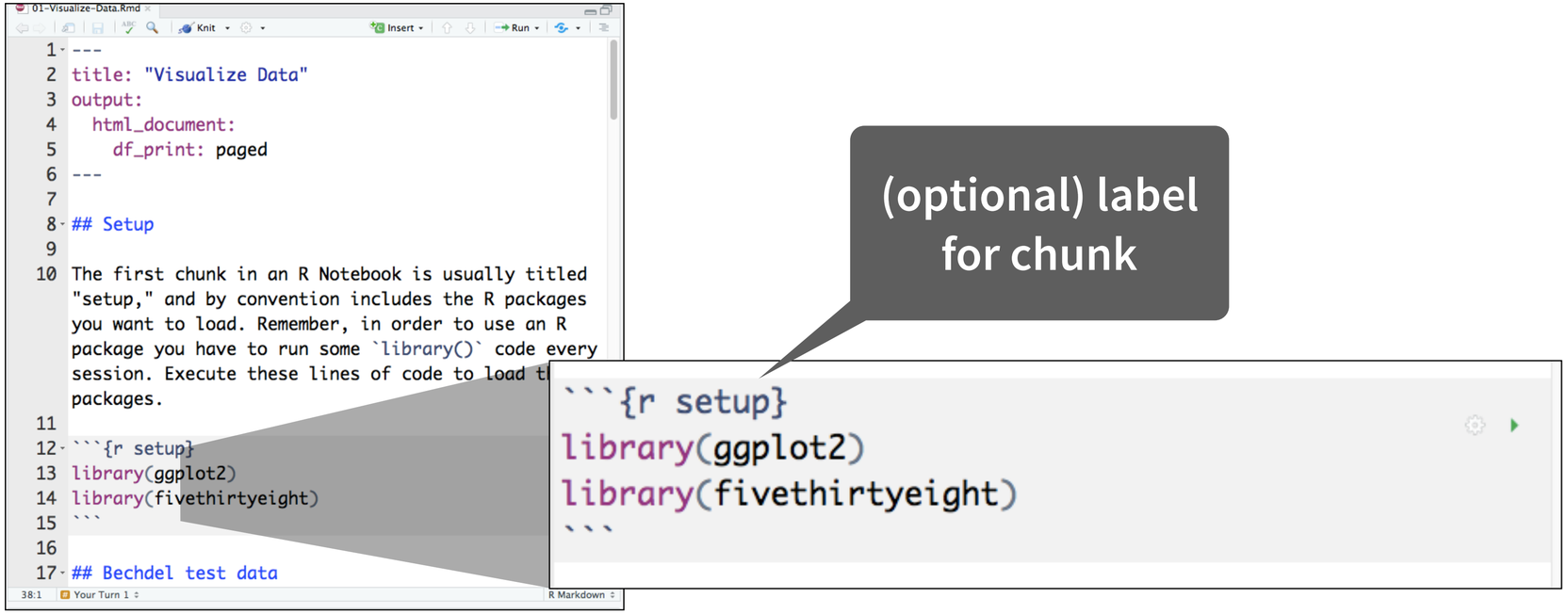
Open the R Notebook 01-Visualize.Rmd

Home > Dropbox > Intro_to_R_and_RStudio > Day1

- code
- keynotes
- slides

Setup

The setup chunk is always run once before anything else



bechdel

Data on movies and the Bechtel test

```
bechdel
?bechdel
```

Consider

Confer with the people around you.

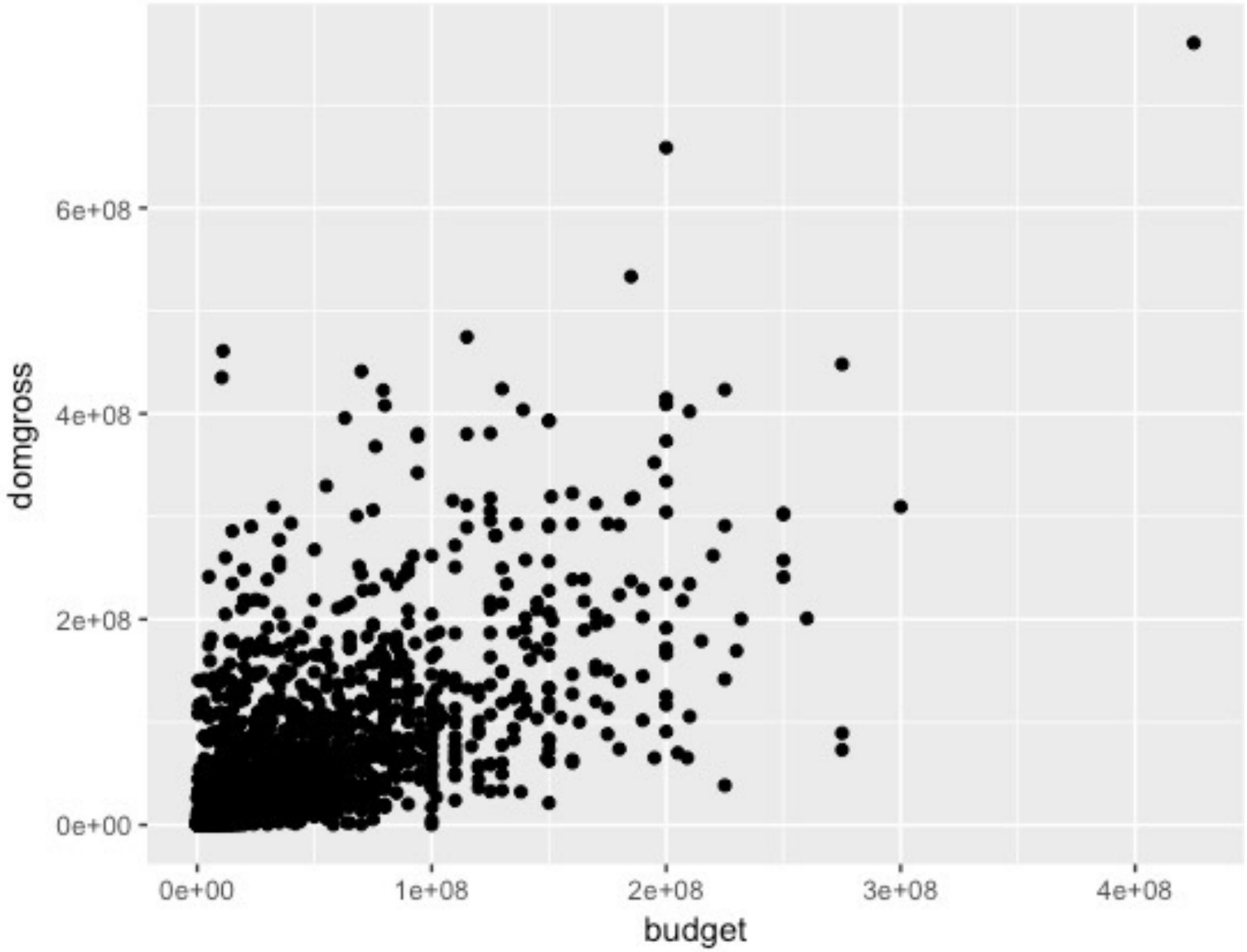
What relationship do you expect to see between movie budget (`budget`) and domestic gross(`domgross`)?

Your Turn 1

Run this code in your notebook to make a graph.

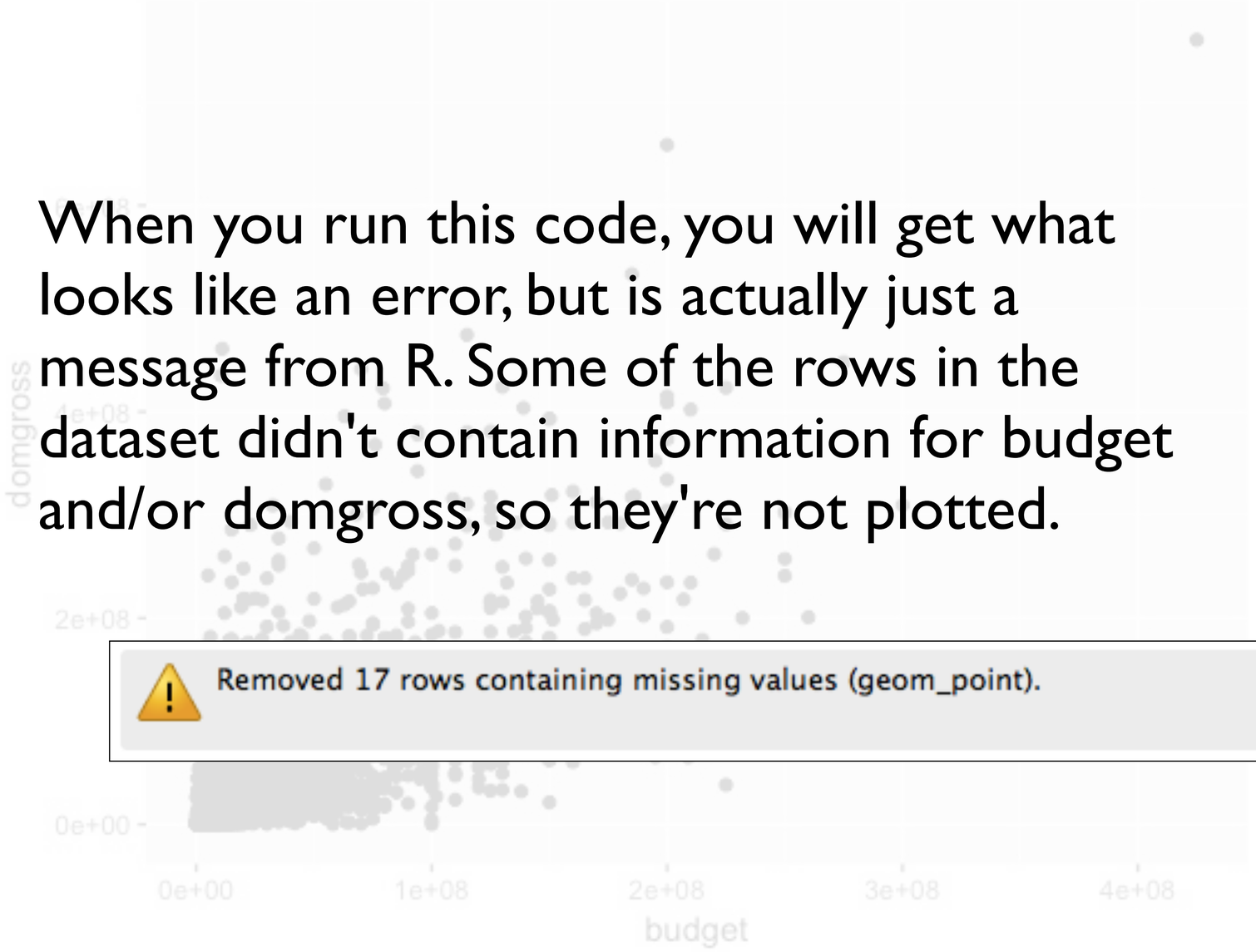
Pay strict attention to spelling, capitalization, and parentheses!

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross))
```




```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross))
```

When you run this code, you will get what looks like an error, but is actually just a message from R. Some of the rows in the dataset didn't contain information for budget and/or domgross, so they're not plotted.



Removed 17 rows containing missing values (geom_point).

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross))
```

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

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross))
```

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

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross))
```

data

+ before new line

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross))
```

type of layer

aes()

x variable

y variable

A template

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

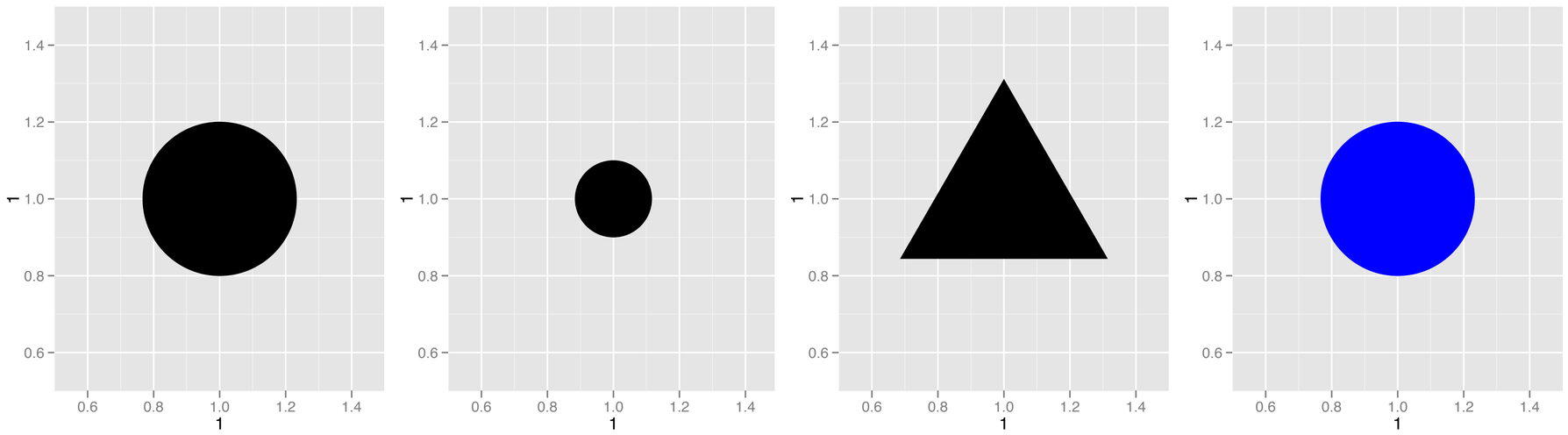
```
    geom_point(mapping = aes(x = budget, y = domgross))
```

A template

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

Aesthetics

Aesthetics



Visual Space		Data Space
color	⟷	clean_test
Purple	⟷	nowomen
Blue	⟷	notalk
Teal	⟷	men
Lime	⟷	dubious
Yellow	⟷	ok

Aesthetics

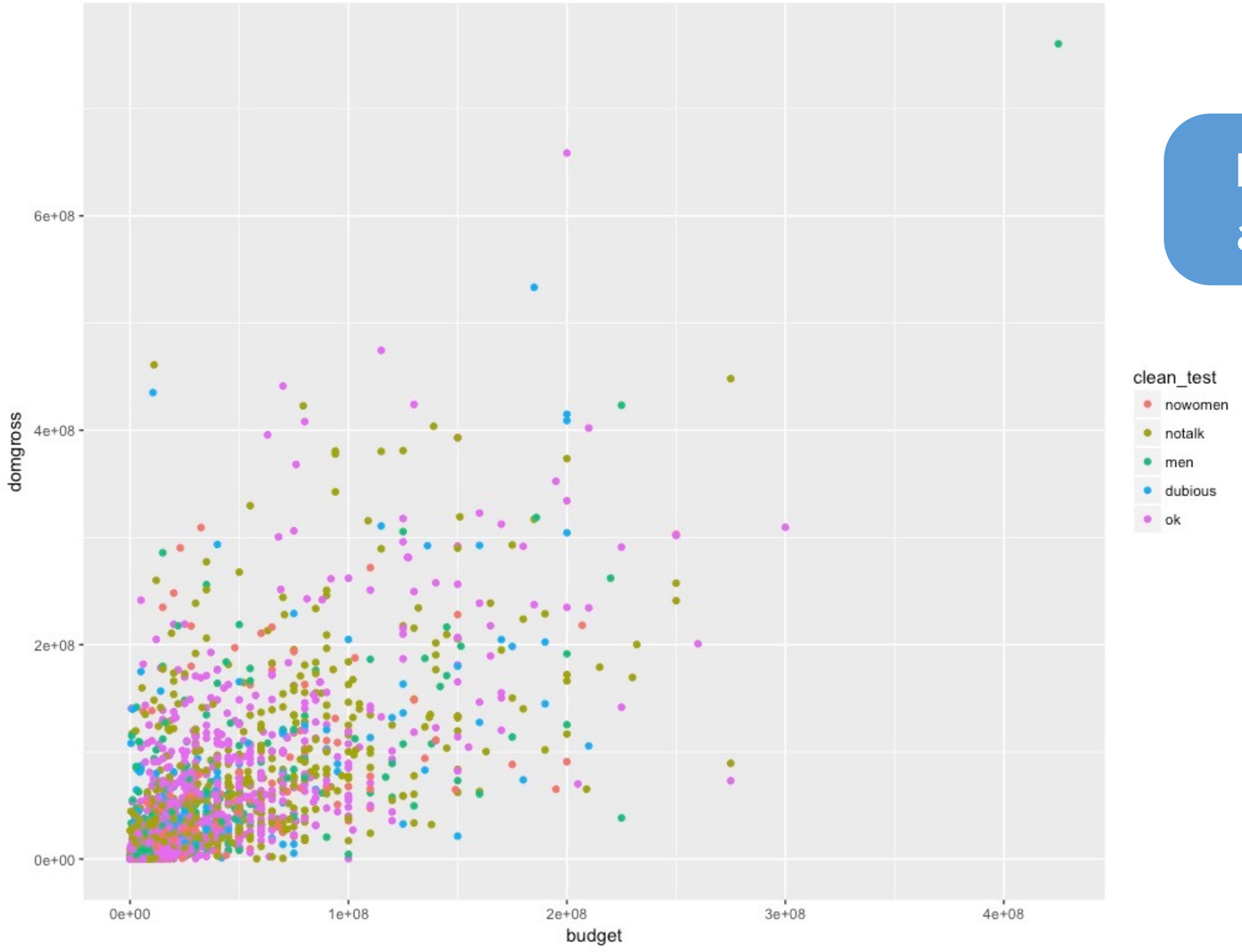
aesthetic property

Variable to map it to

```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, color = clean_test))
```

```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, size = clean_test))
```

```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, color=clean_test))
```



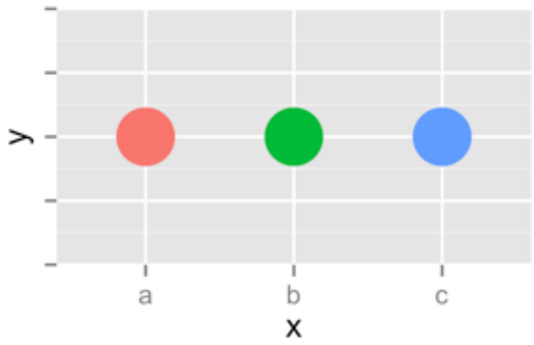
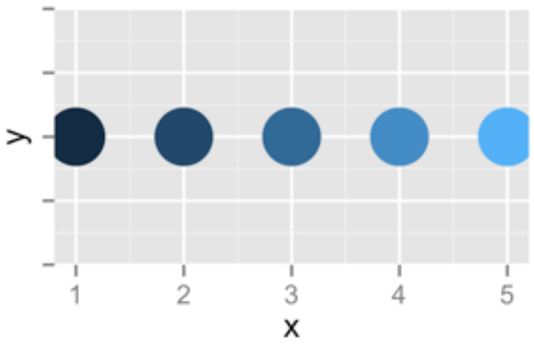
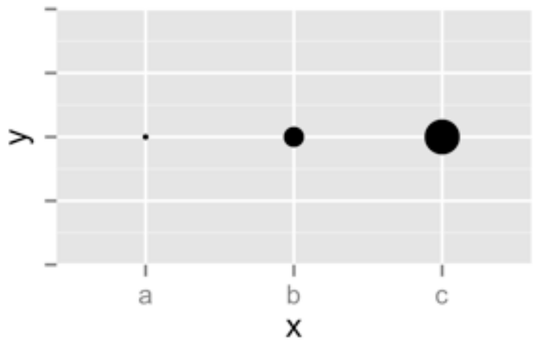
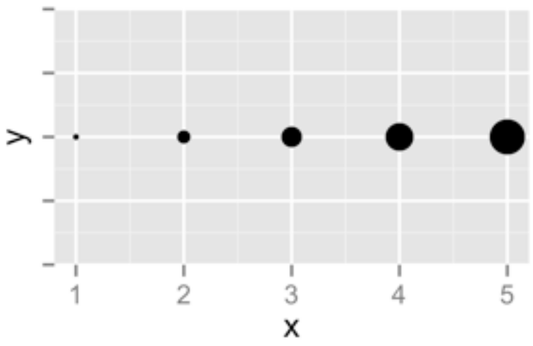
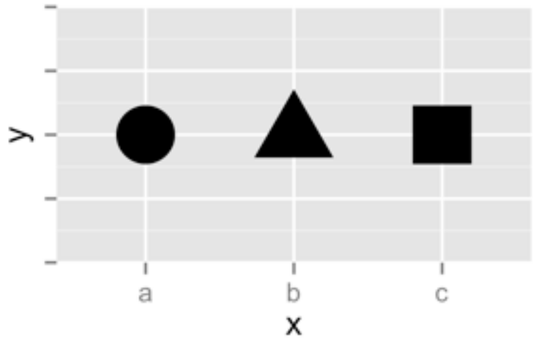
Legend added automatically

Your Turn 2

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

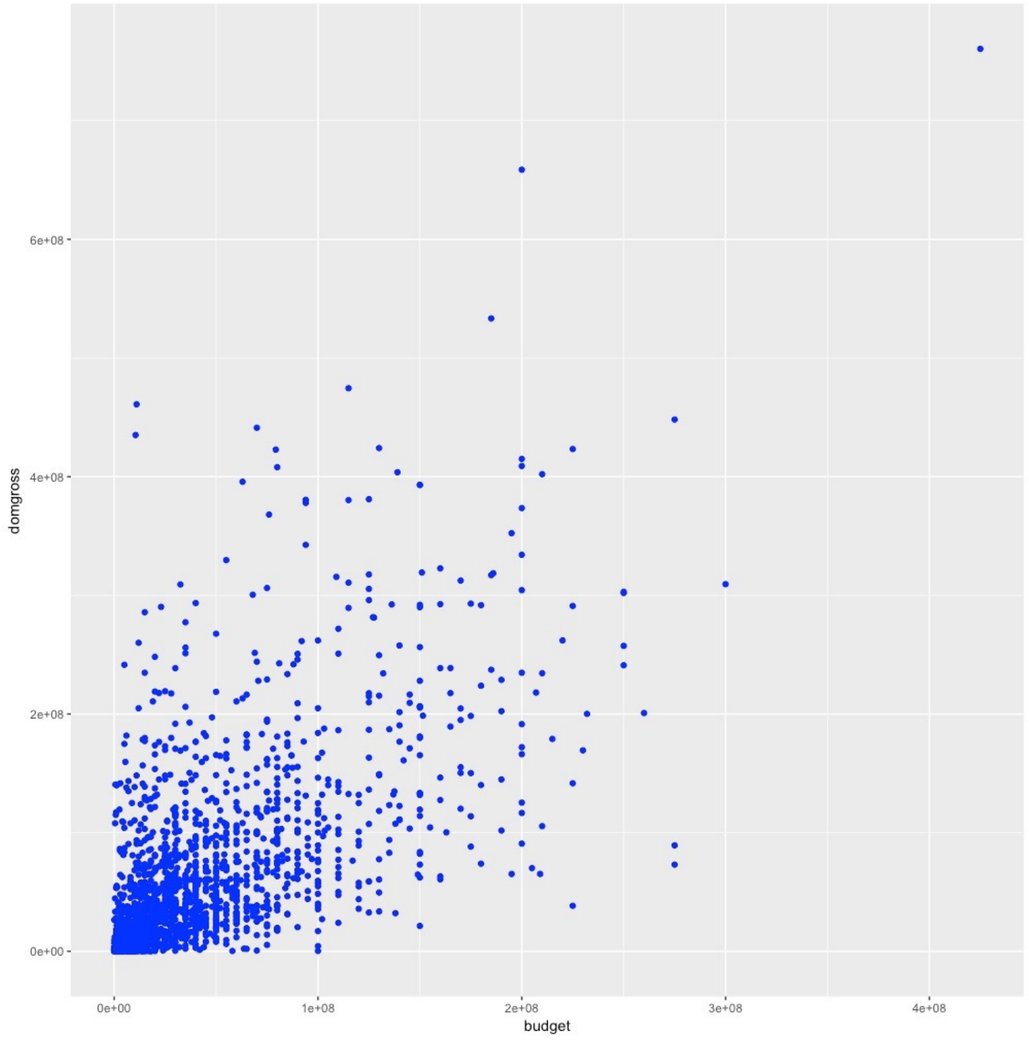
- Do different things happen when you map aesthetics to discrete and continuous variables?
- What happens when you use more than one aesthetic?

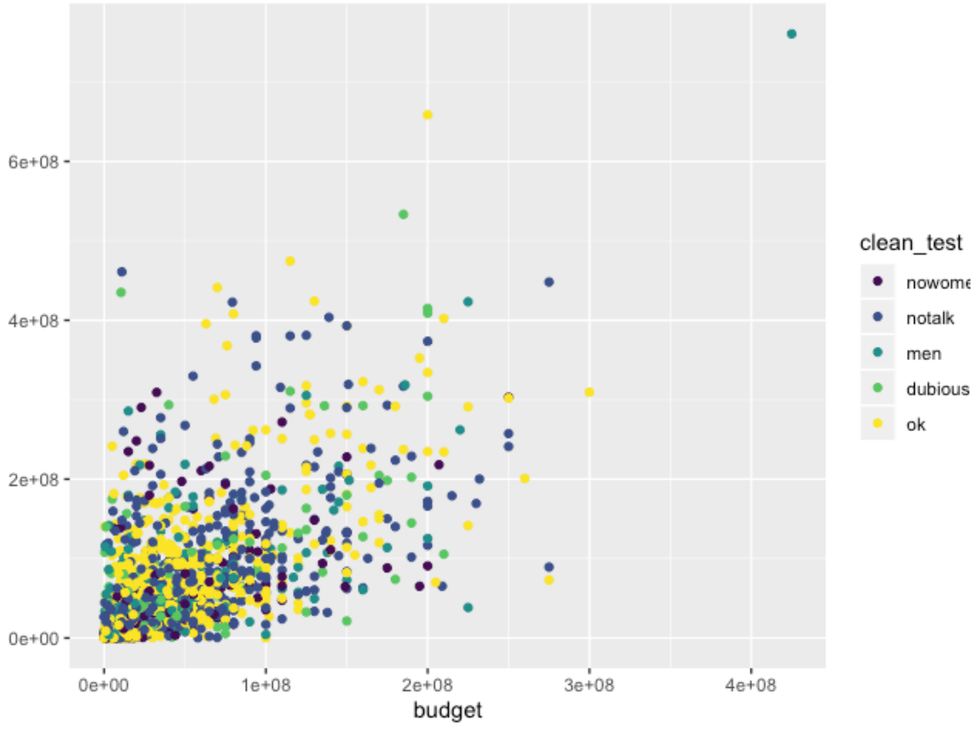
05:00

	Discrete	Continuous
Color		
Size		
Shape		

set vs. map

How would you make this plot?

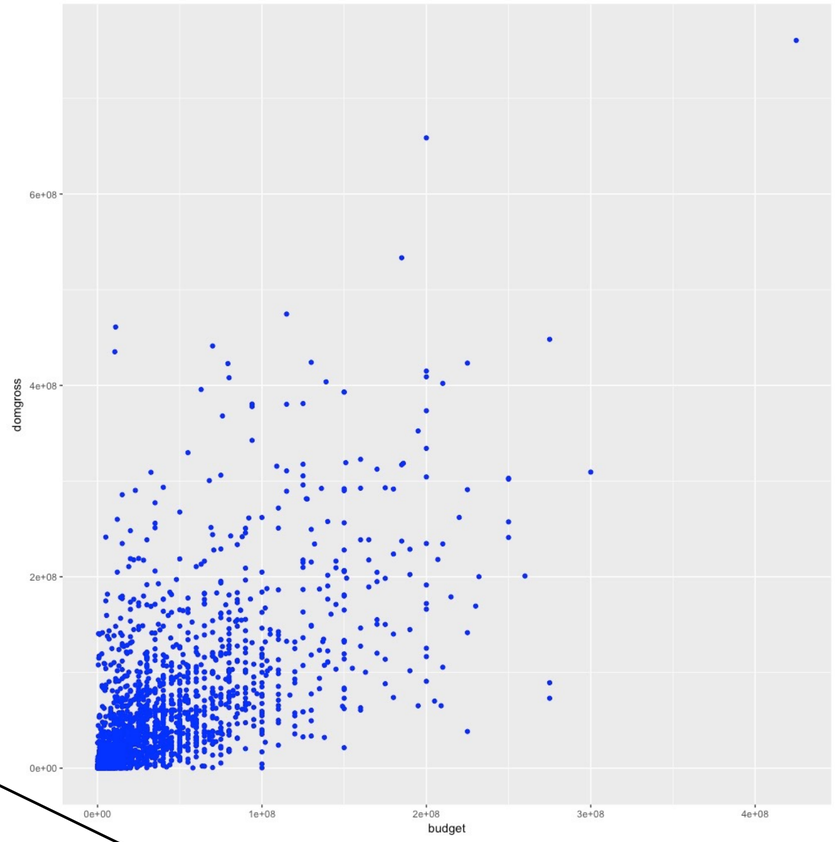




Inside of aes(): maps an aesthetic to a variable

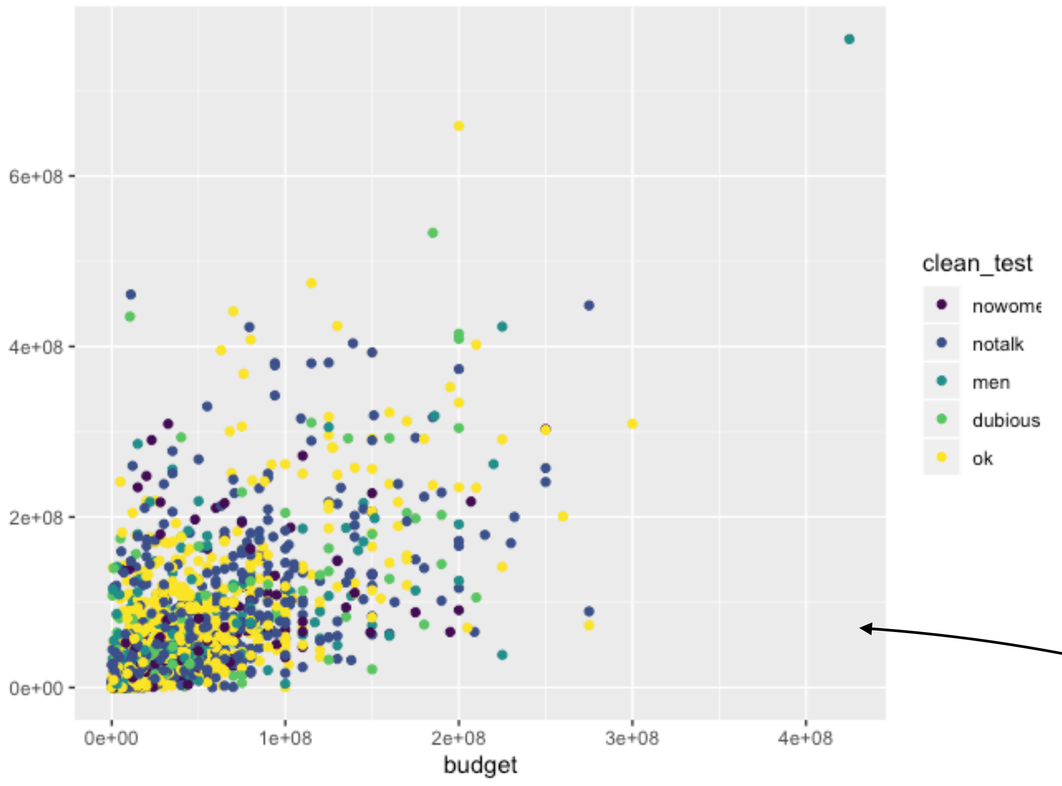
```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, color=clean_test))
```

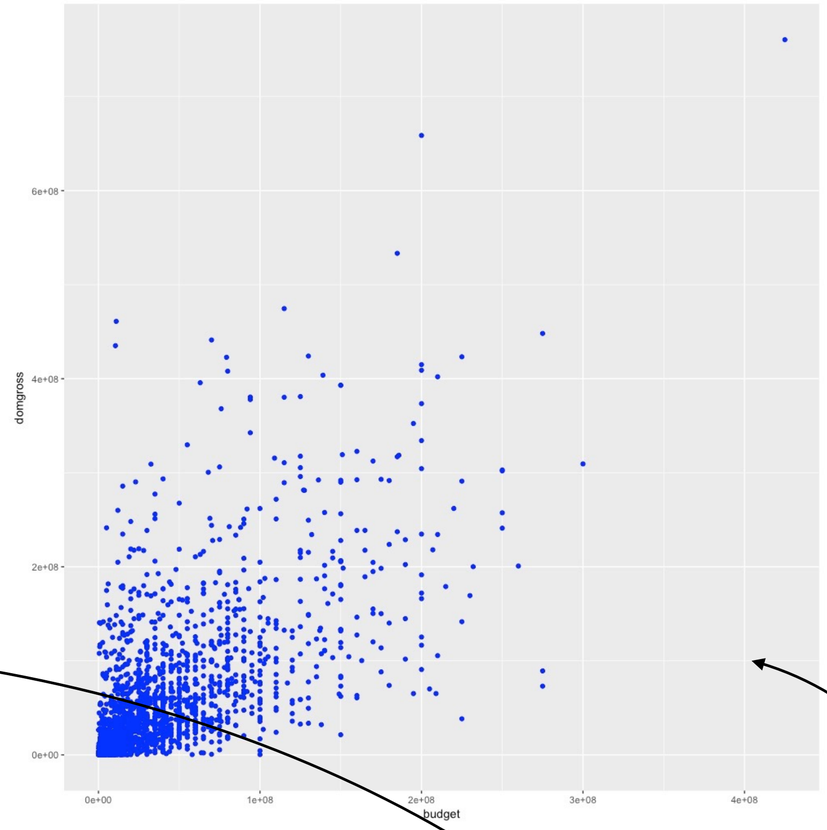
Outside of aes(): sets an aesthetic to a value




```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, color=clean_test)
```

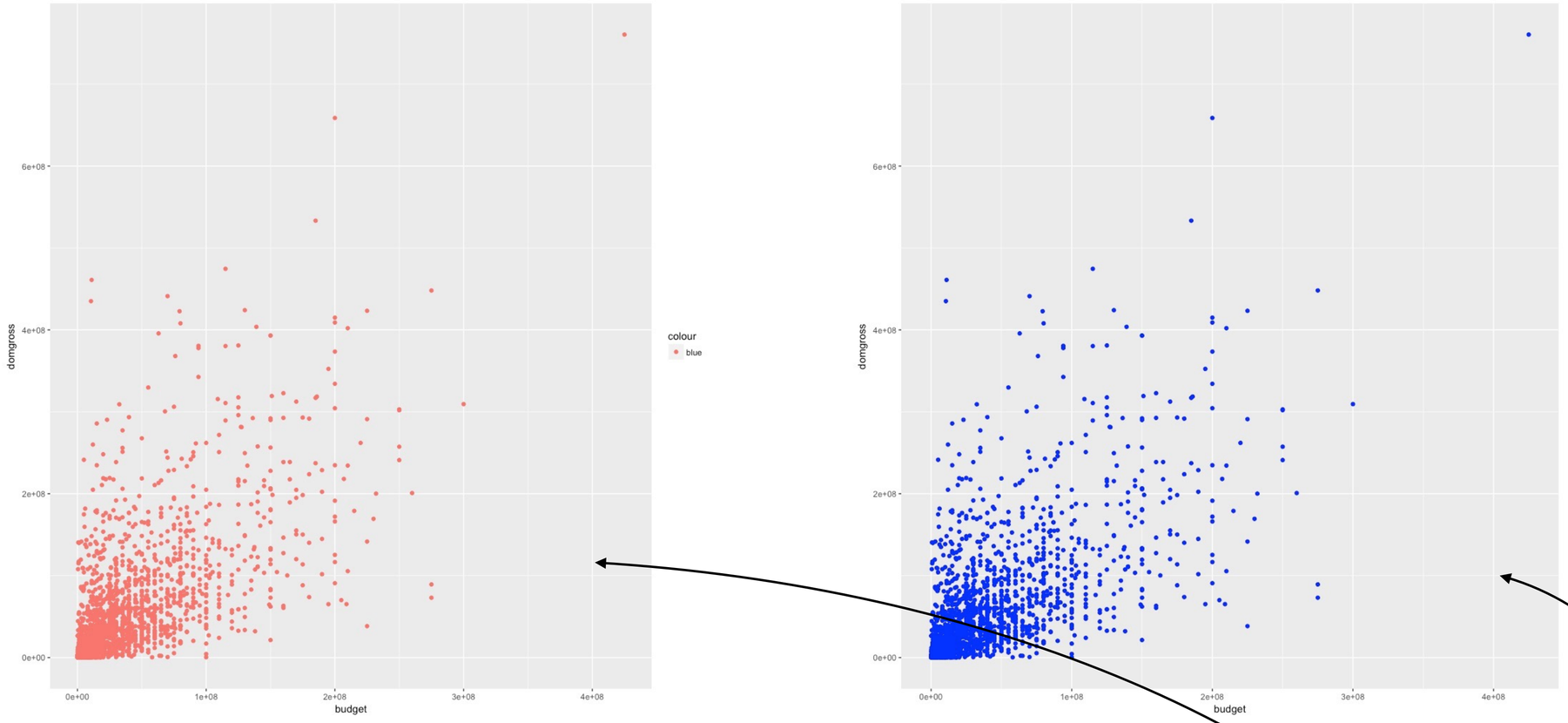
```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross), color="blue")
```






```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, color=clean_test))
```

```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross), color="blue")
```



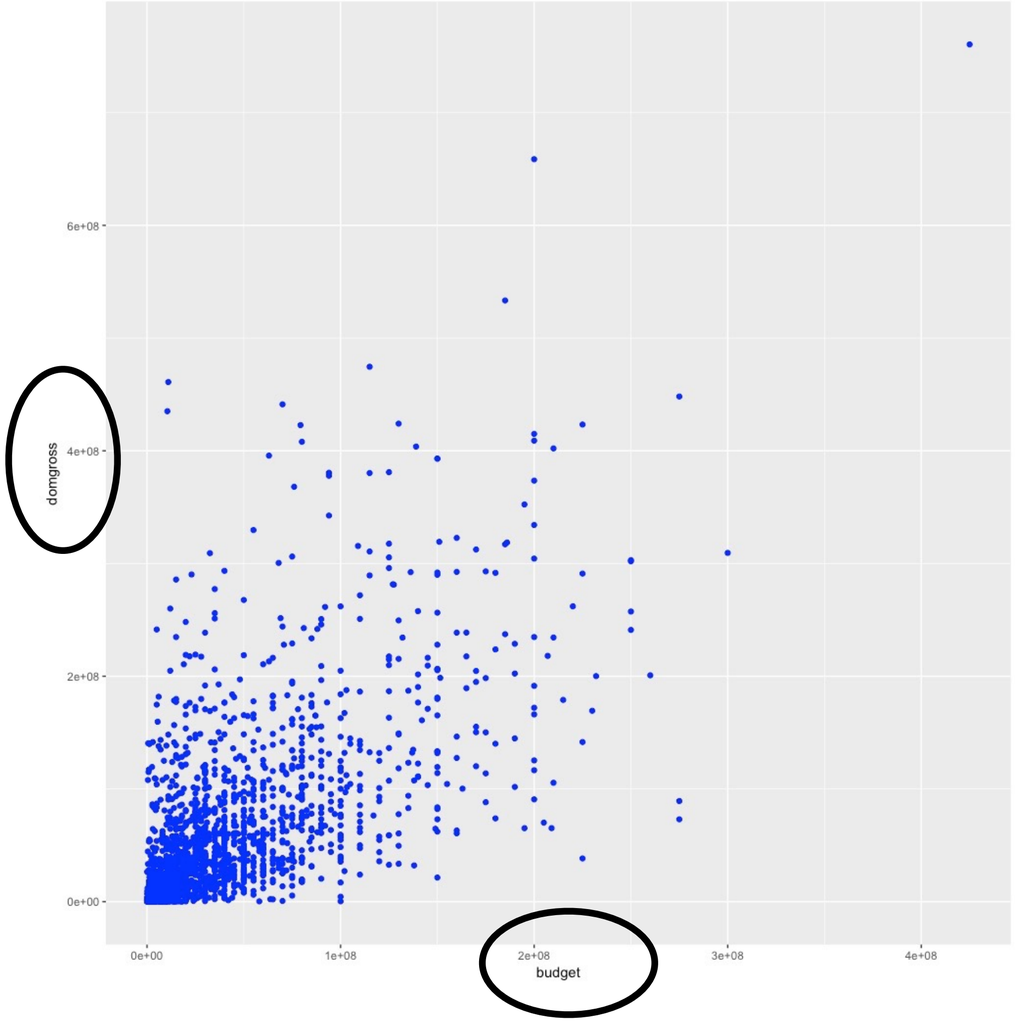

```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross, color="blue"))
```

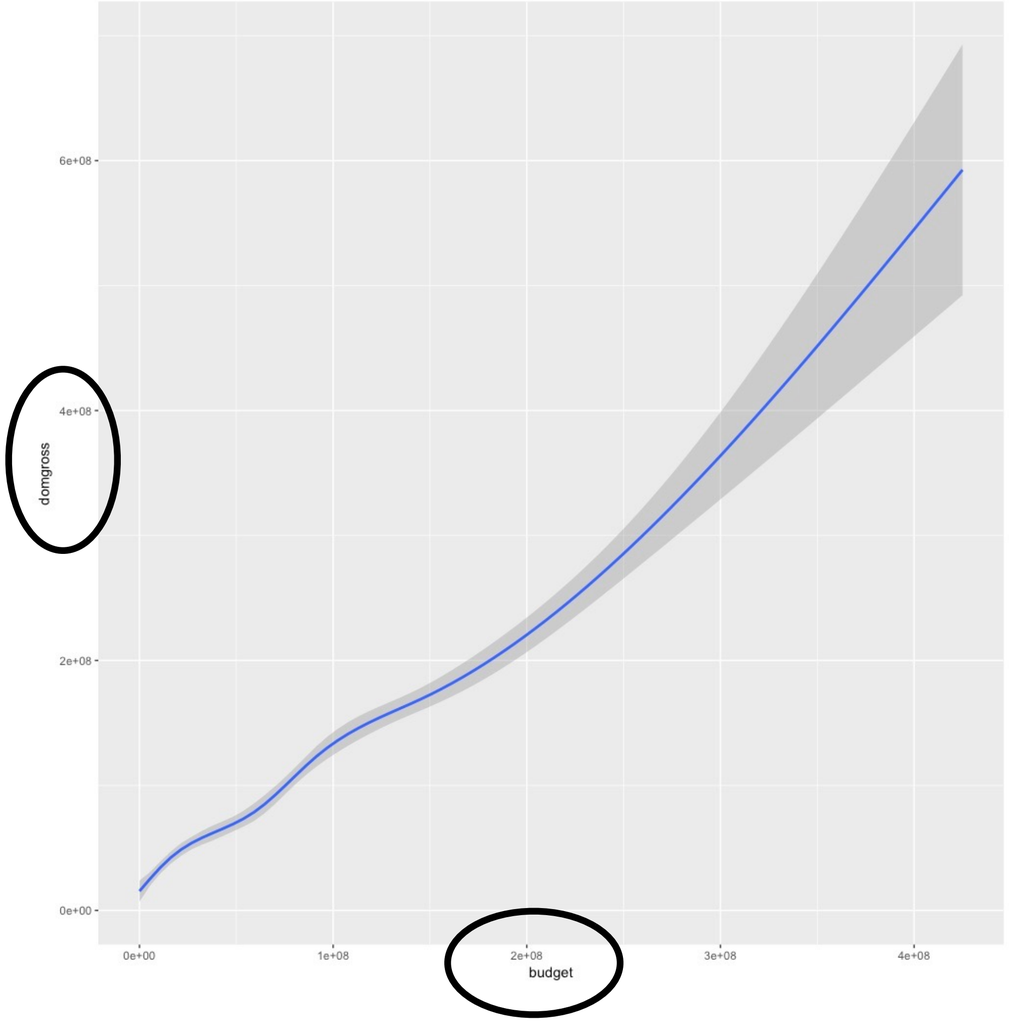
```
ggplot(bechdel) +
    geom_point(mapping = aes(x = budget, y = domgross), color="blue")
```

geoms

How are these plots similar?

Same: x var , y var , data

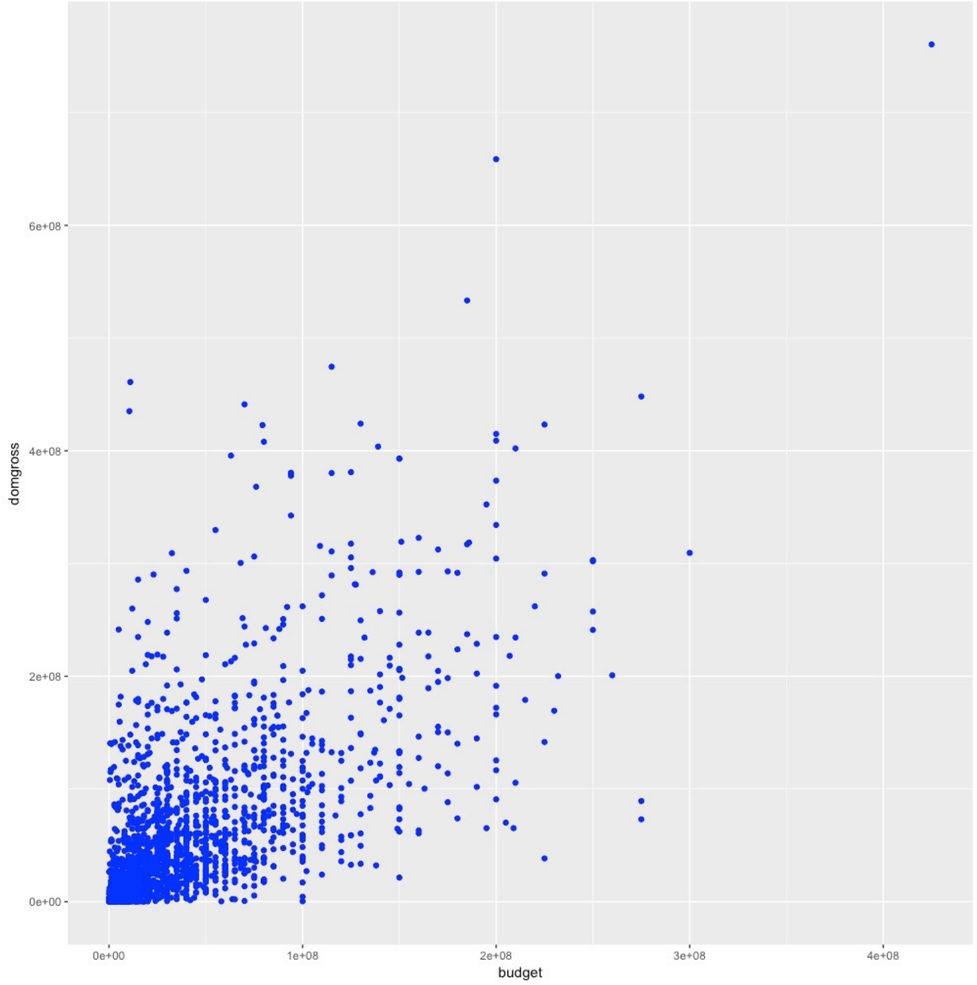


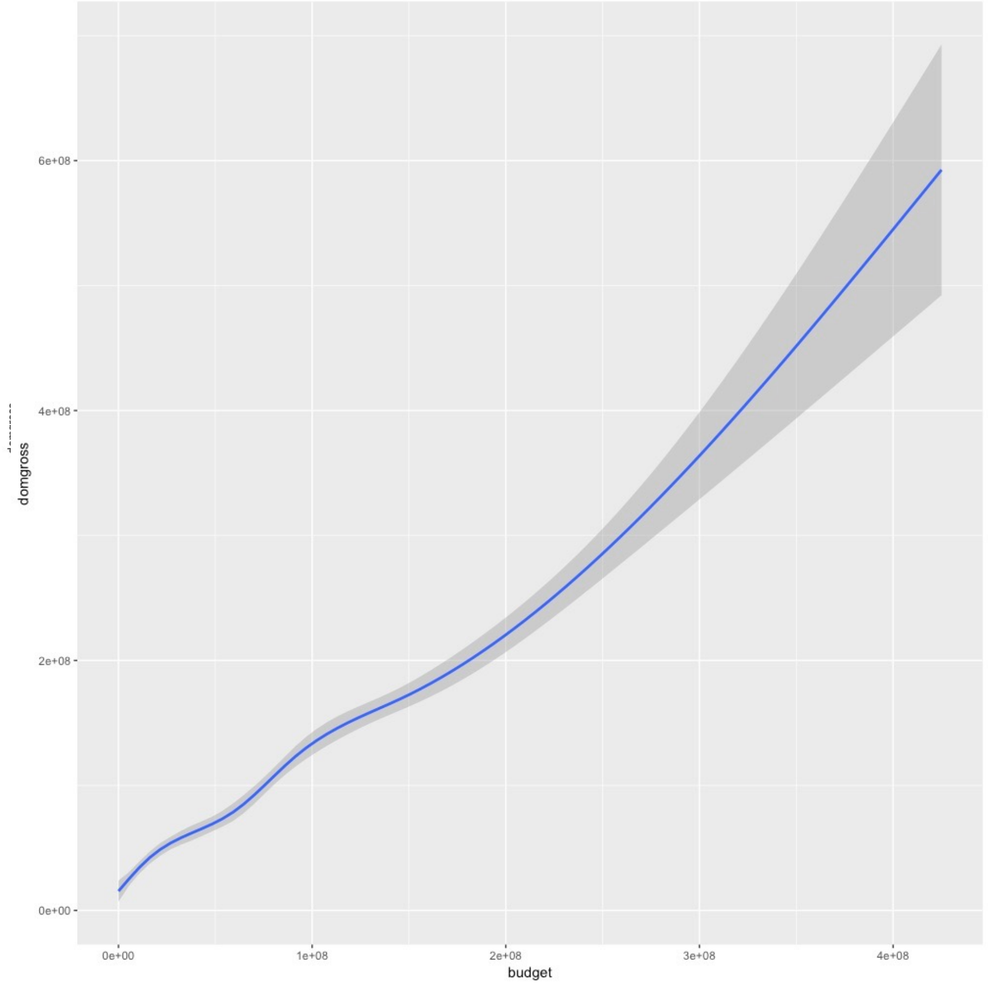


How are these plots different?

Different: geometric object (geom),

e.g. the visual object used to represent the data



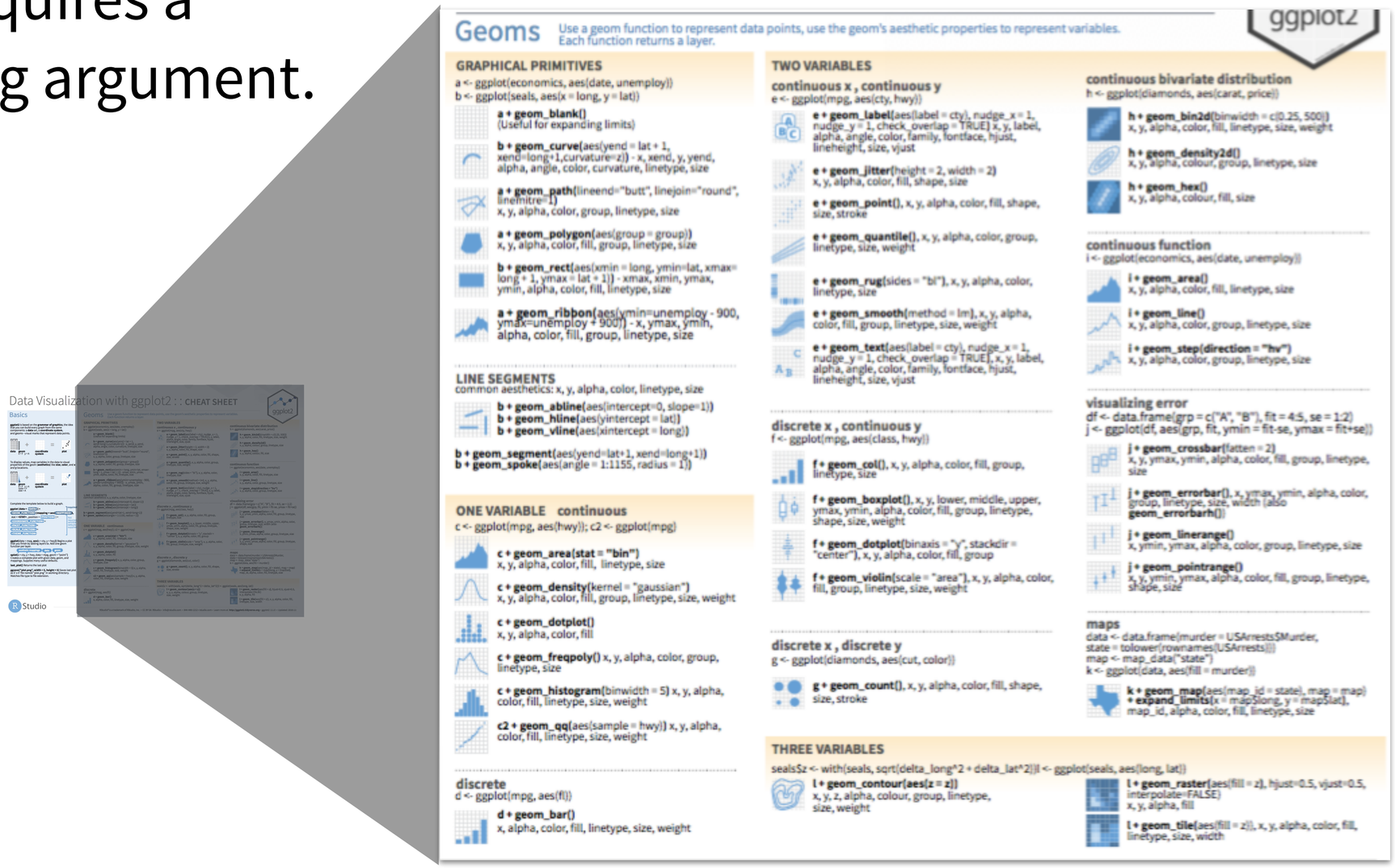


geoms

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

geom_ functions

Each requires a mapping argument.

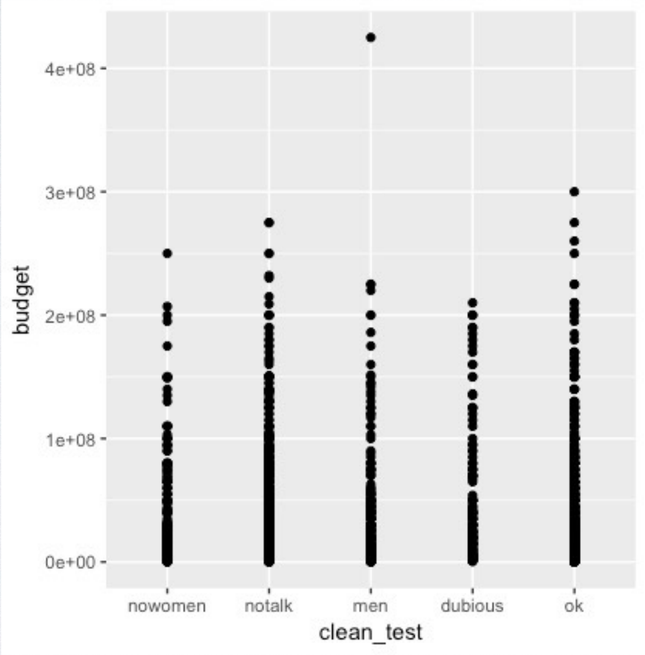


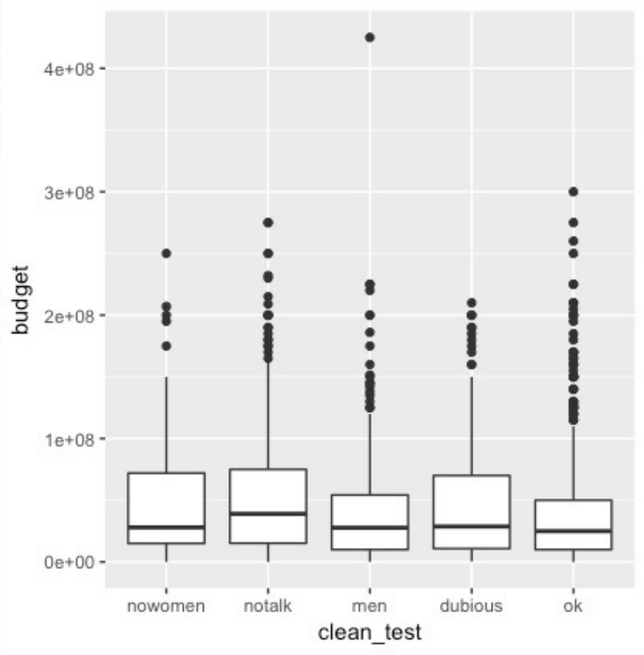
Your Turn

Pair up.

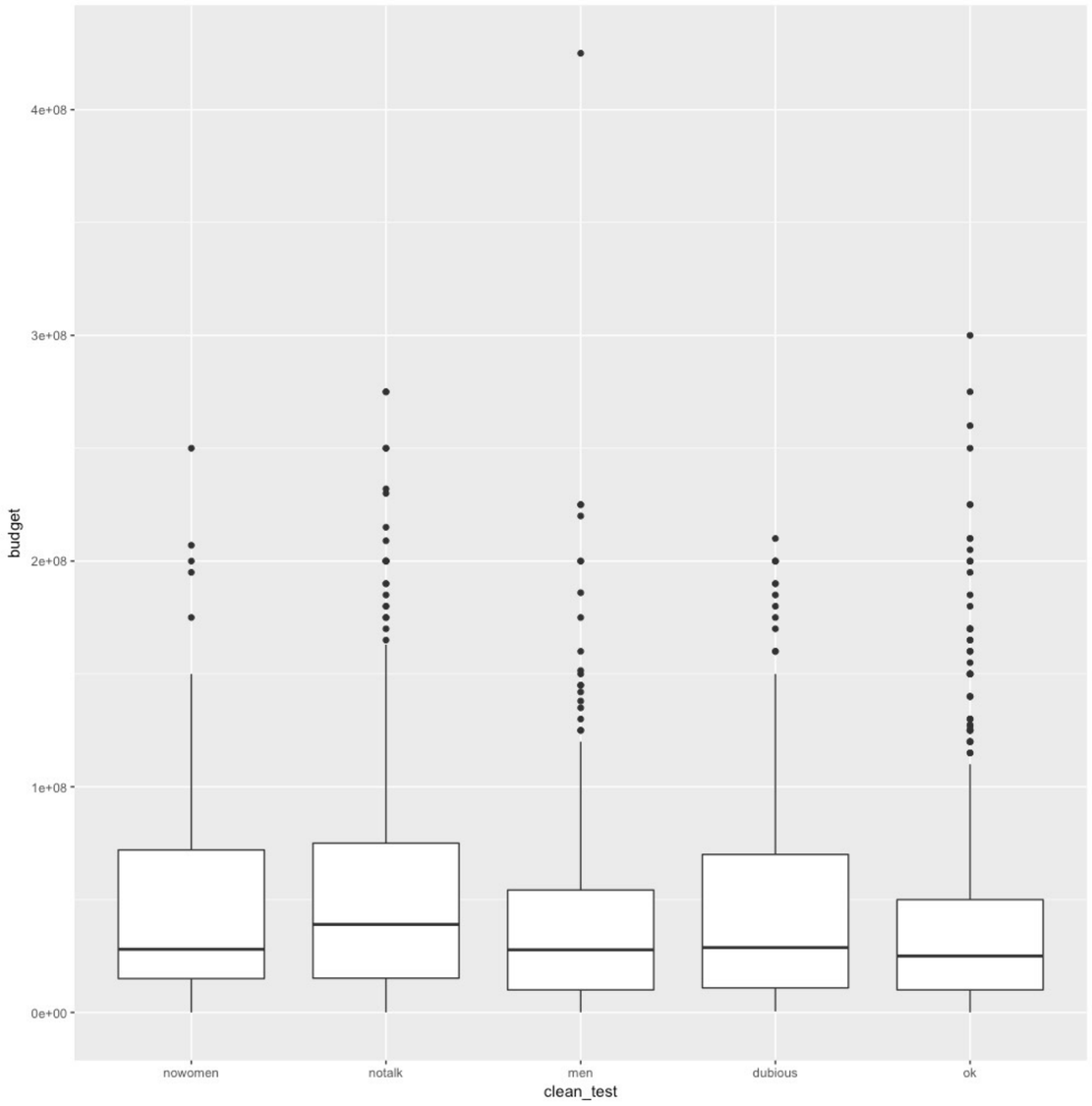
Your Turn 3

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





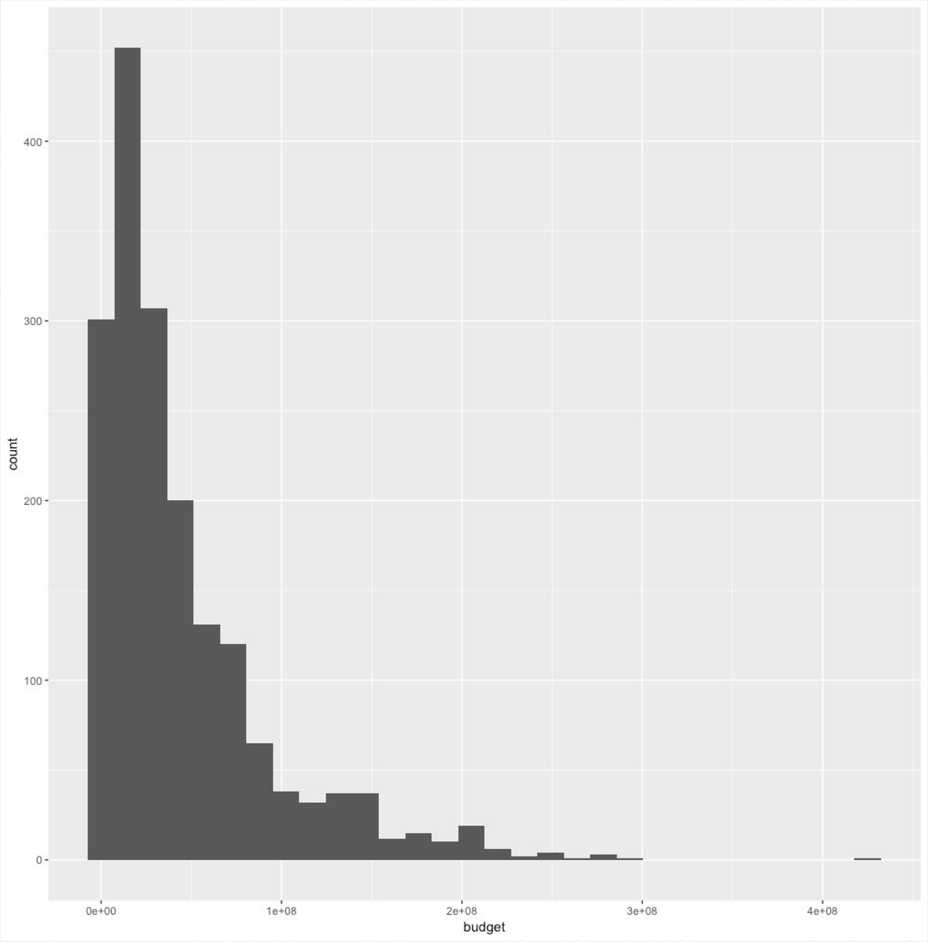

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = clean_test, y = budget))
```

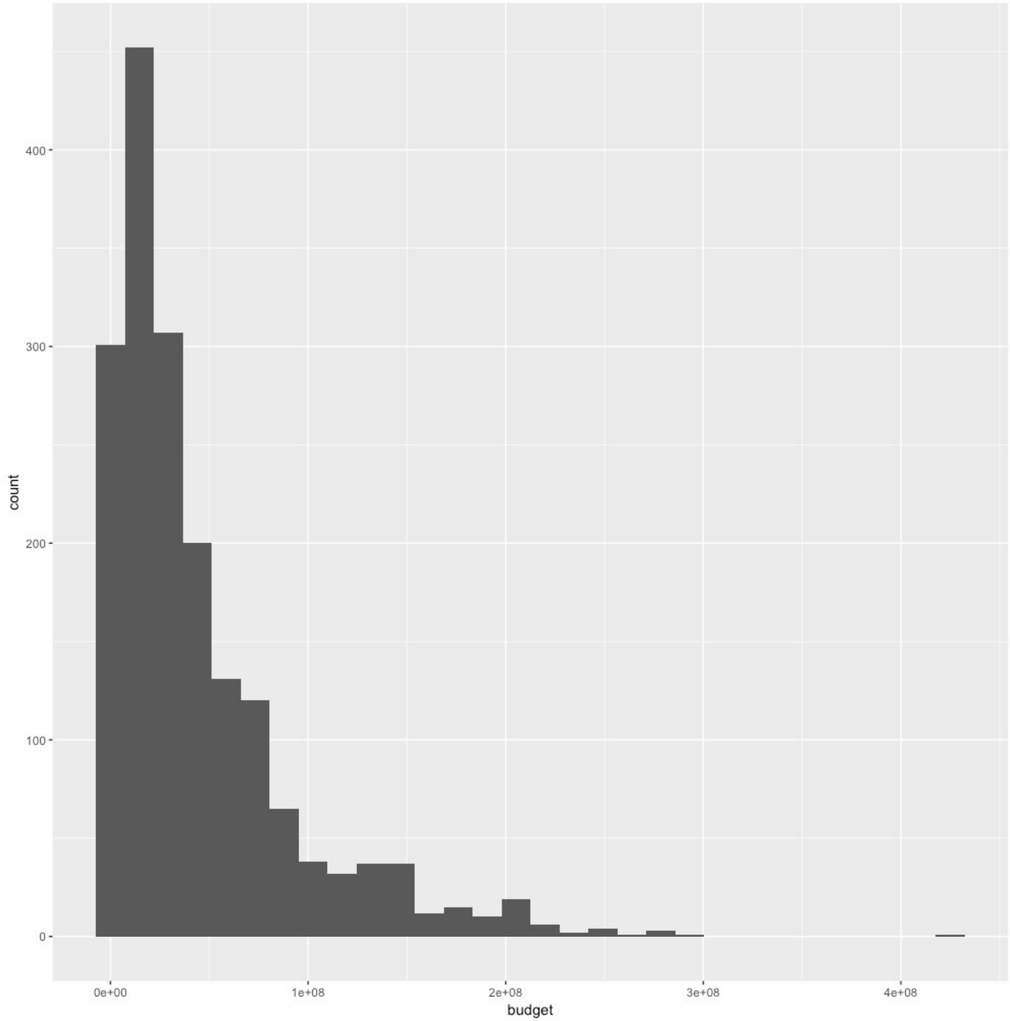



```
ggplot(data = bechdel) +
  geom_boxplot(mapping = aes(x = clean_test, y = budget))
```

Your Turn 4

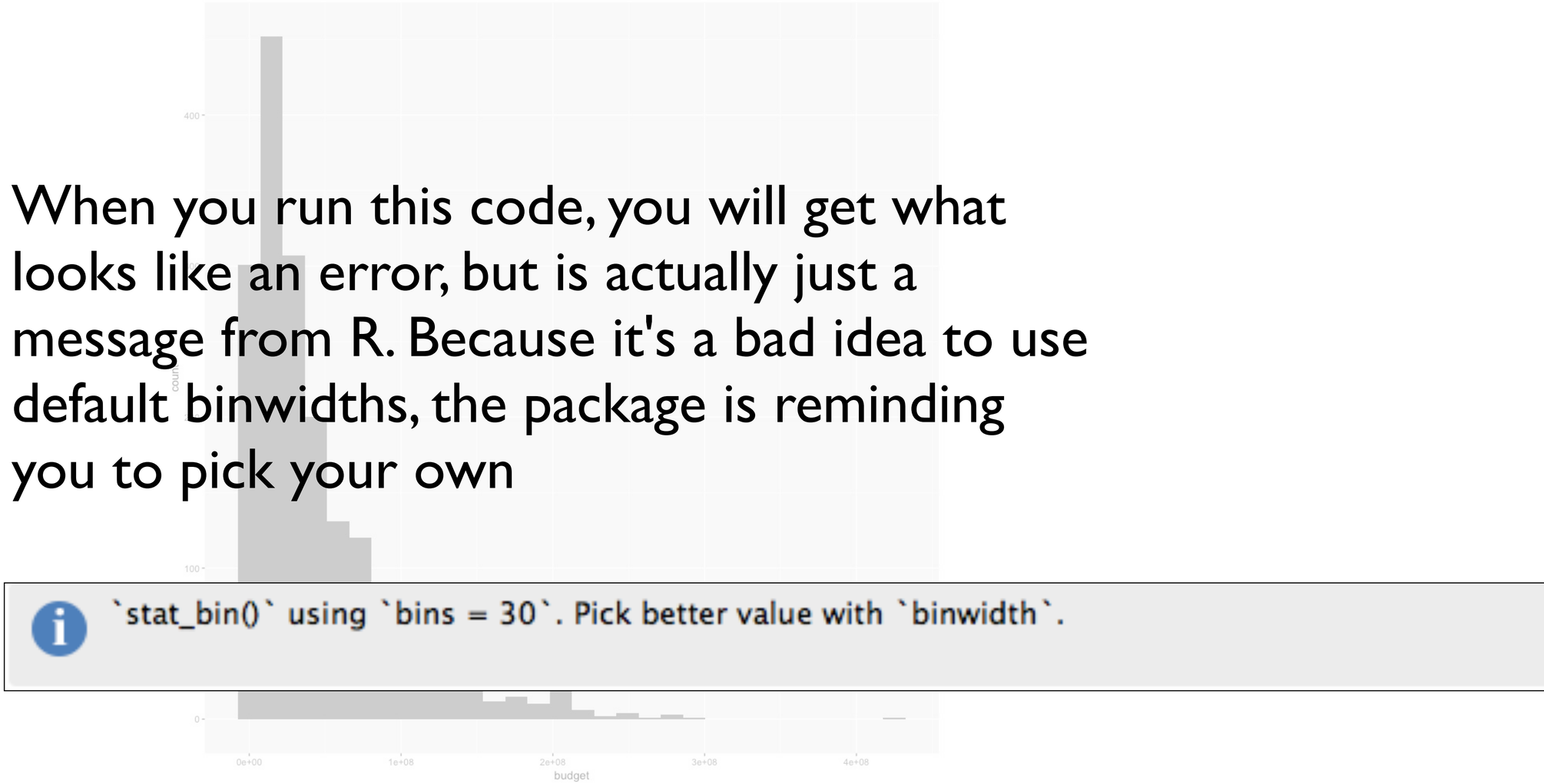
With your partner, make the histogram of **budget** below.
Use the cheatsheet. Hint: do not supply a **y** variable.






```
ggplot(data = bechdel) +
  geom_histogram(mapping = aes(x = budget))
```

When you run this code, you will get what looks like an error, but is actually just a message from R. Because it's a bad idea to use default binwidths, the package is reminding you to pick your own

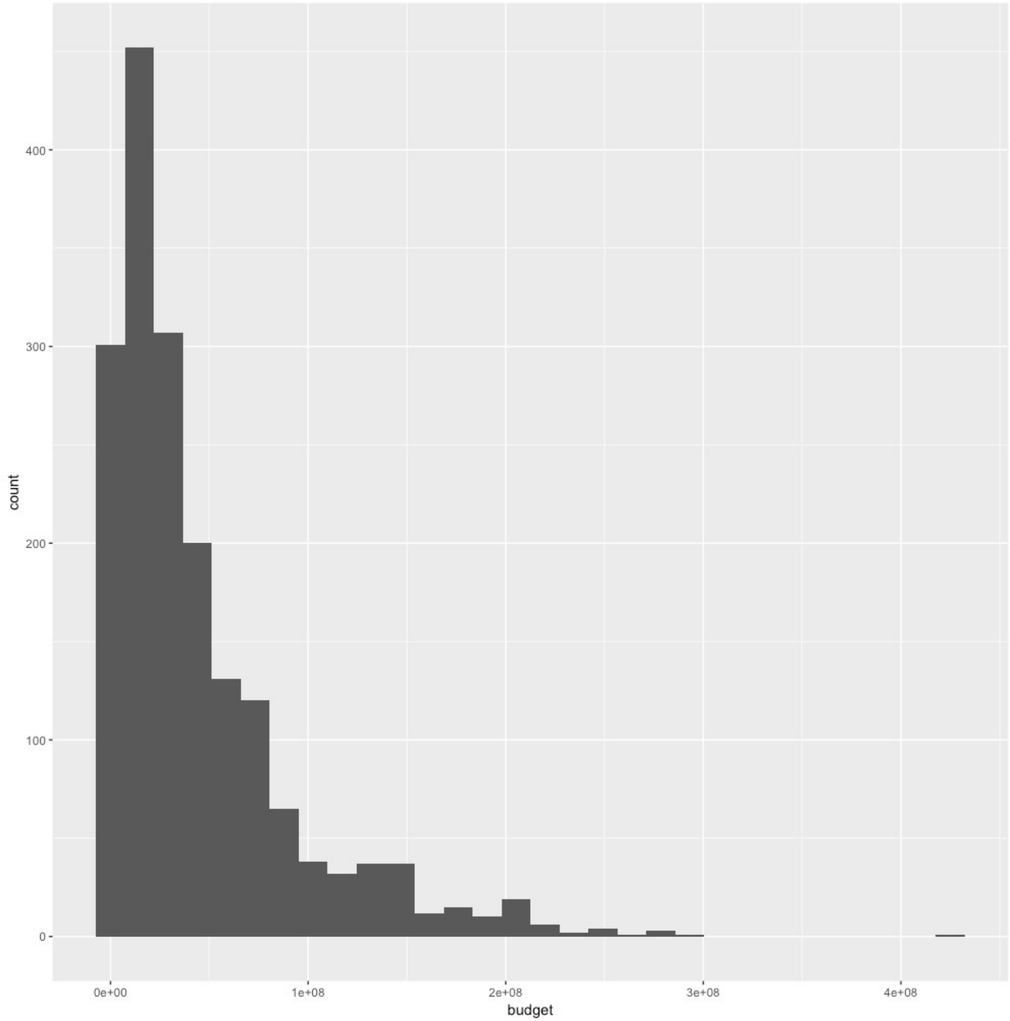



```
ggplot(data = bechdel) +
  geom_histogram(mapping = aes(x = budget))
```

Your Turn 5

What would be a reasonable `binwidth` for `budget`?

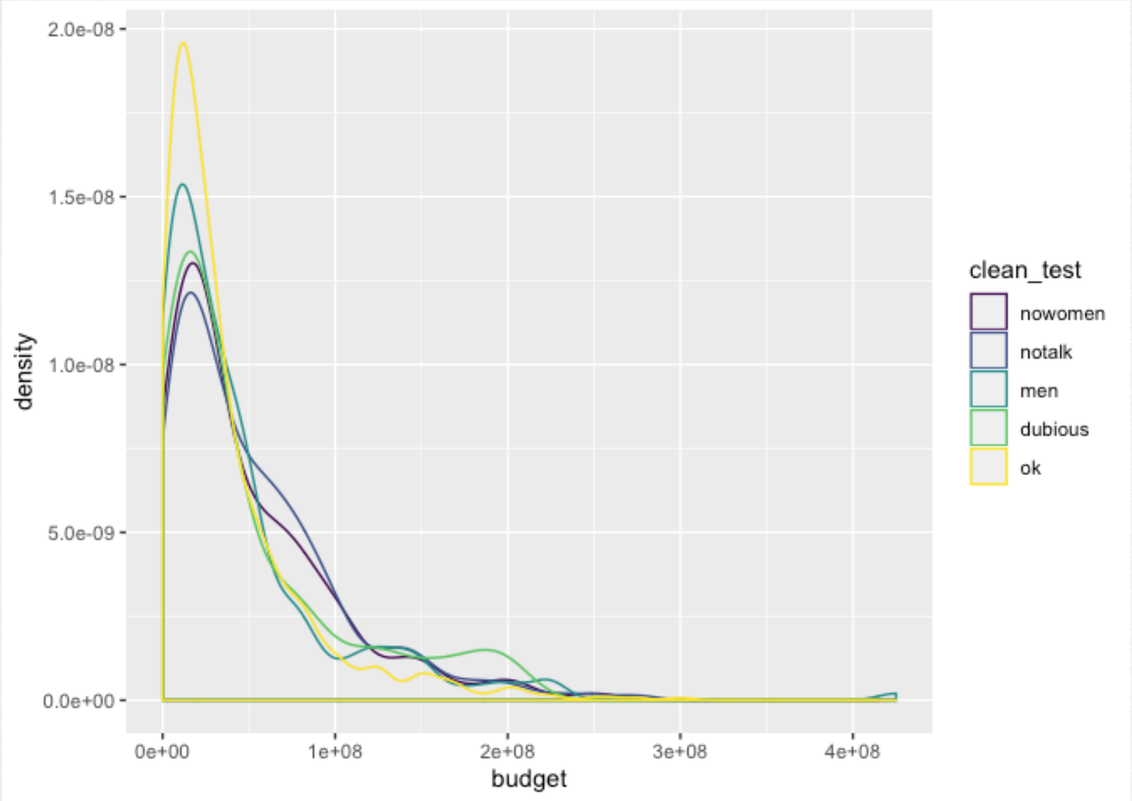
Try it out

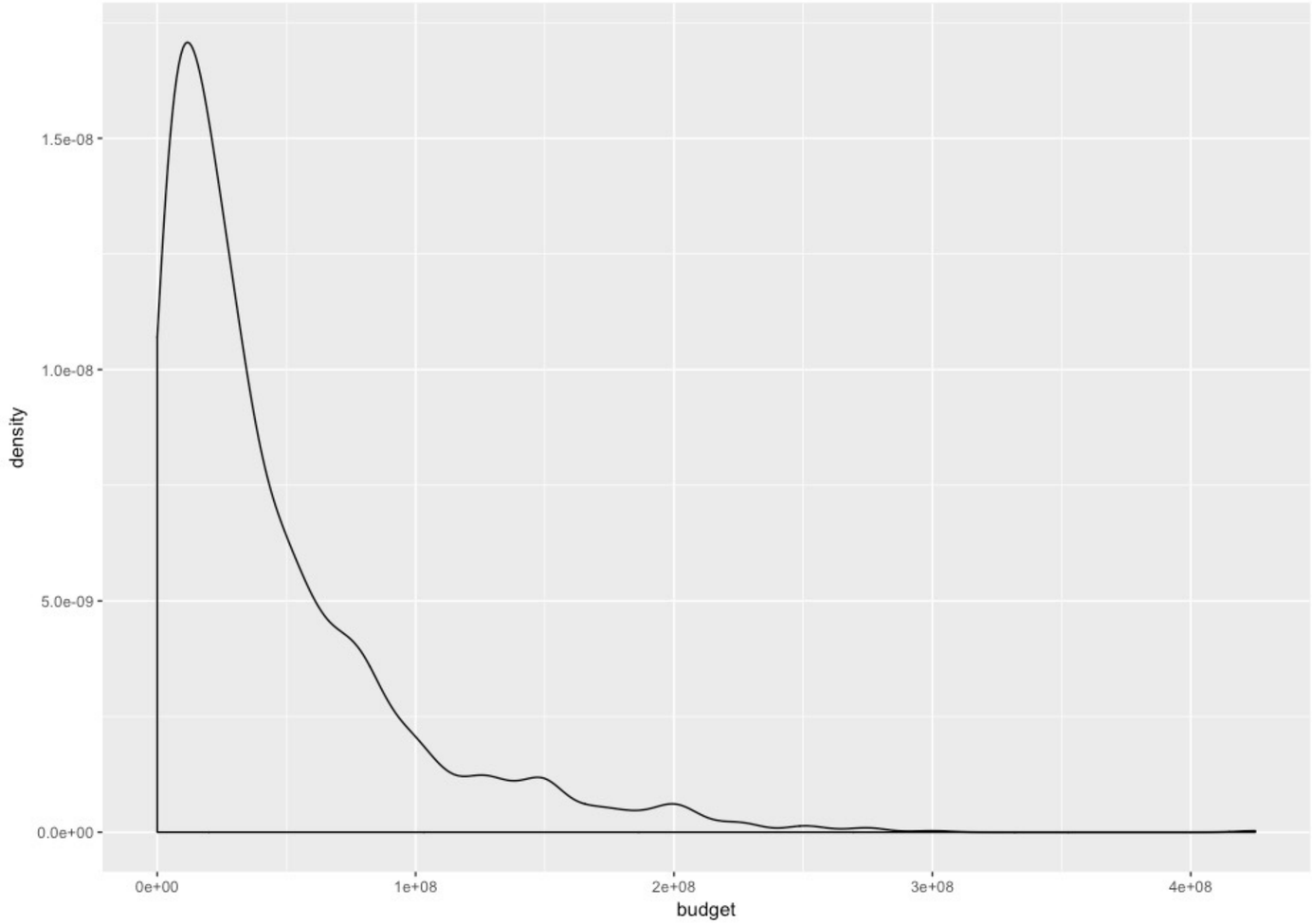



```
ggplot(data = bechdel) +
  geom_histogram(mapping = aes(x = budget), binwidth=10000000)
```

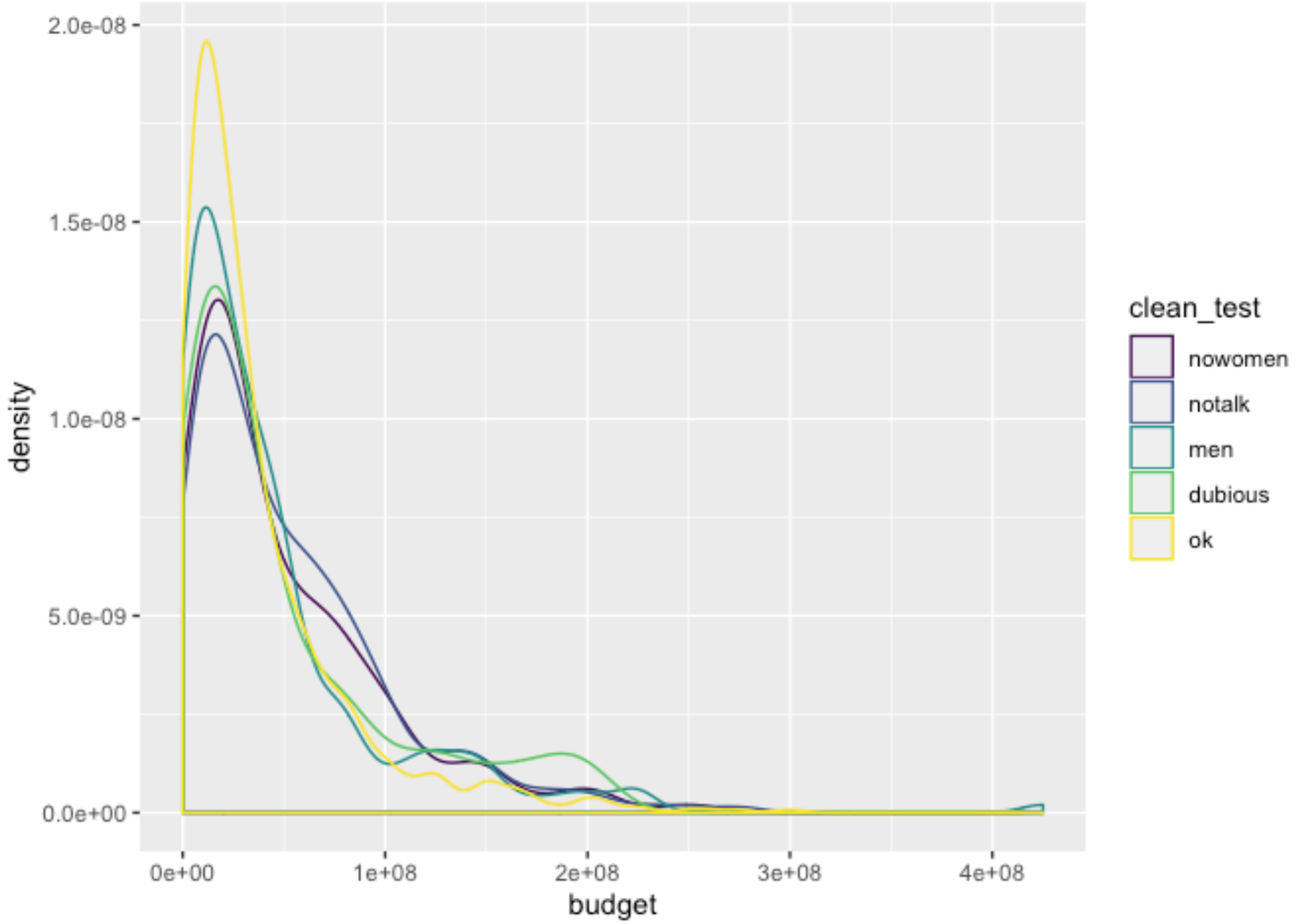
Your Turn 6

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





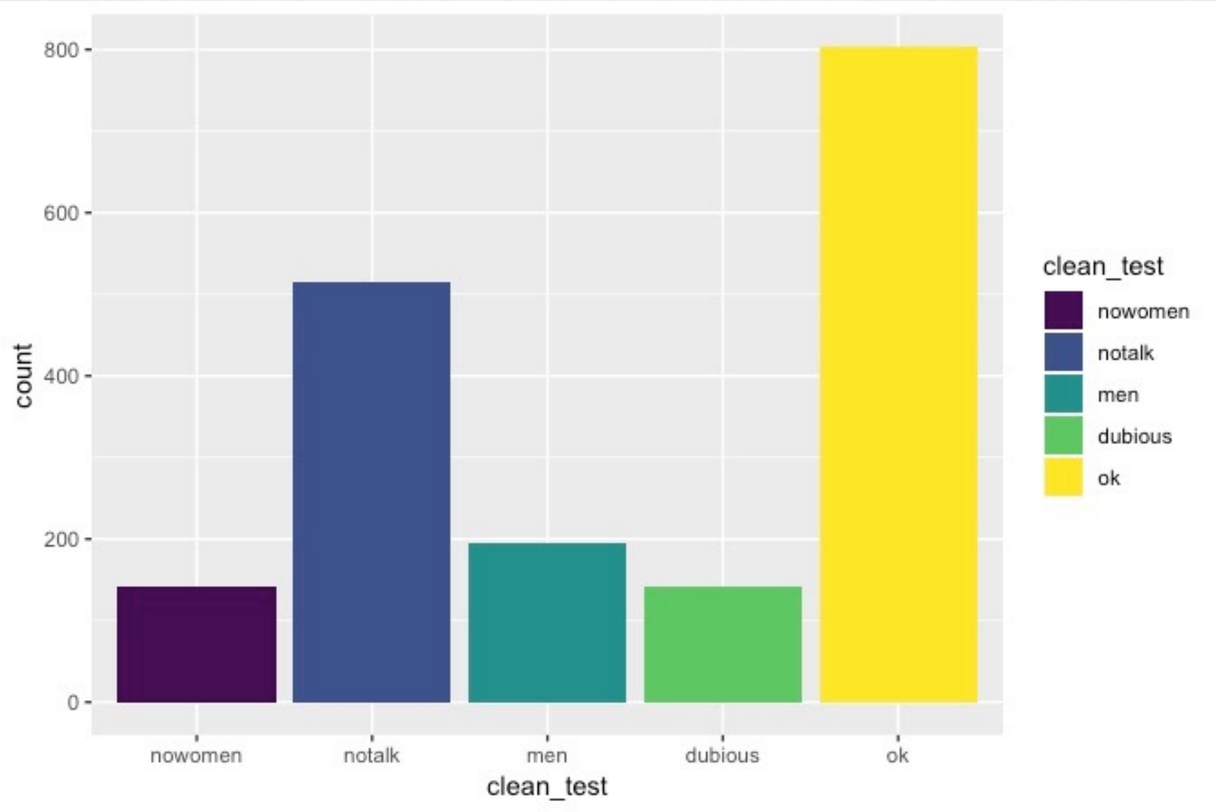

```
ggplot(data = bechdel) +
  geom_density(mapping = aes(x = budget))
```

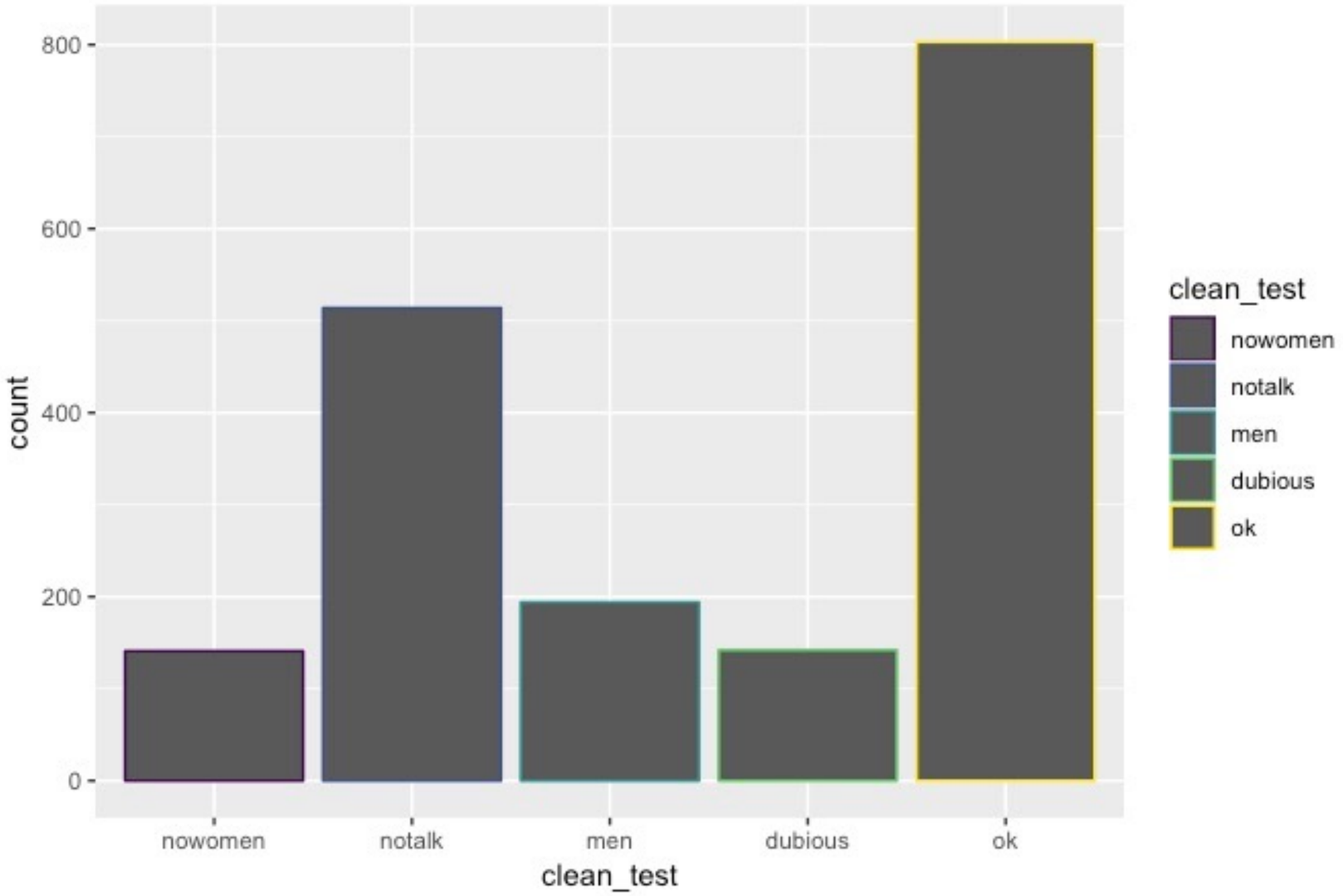



```
ggplot(data = bechdel) +
  geom_density(mapping = aes(x = budget, color=clean_test))
```

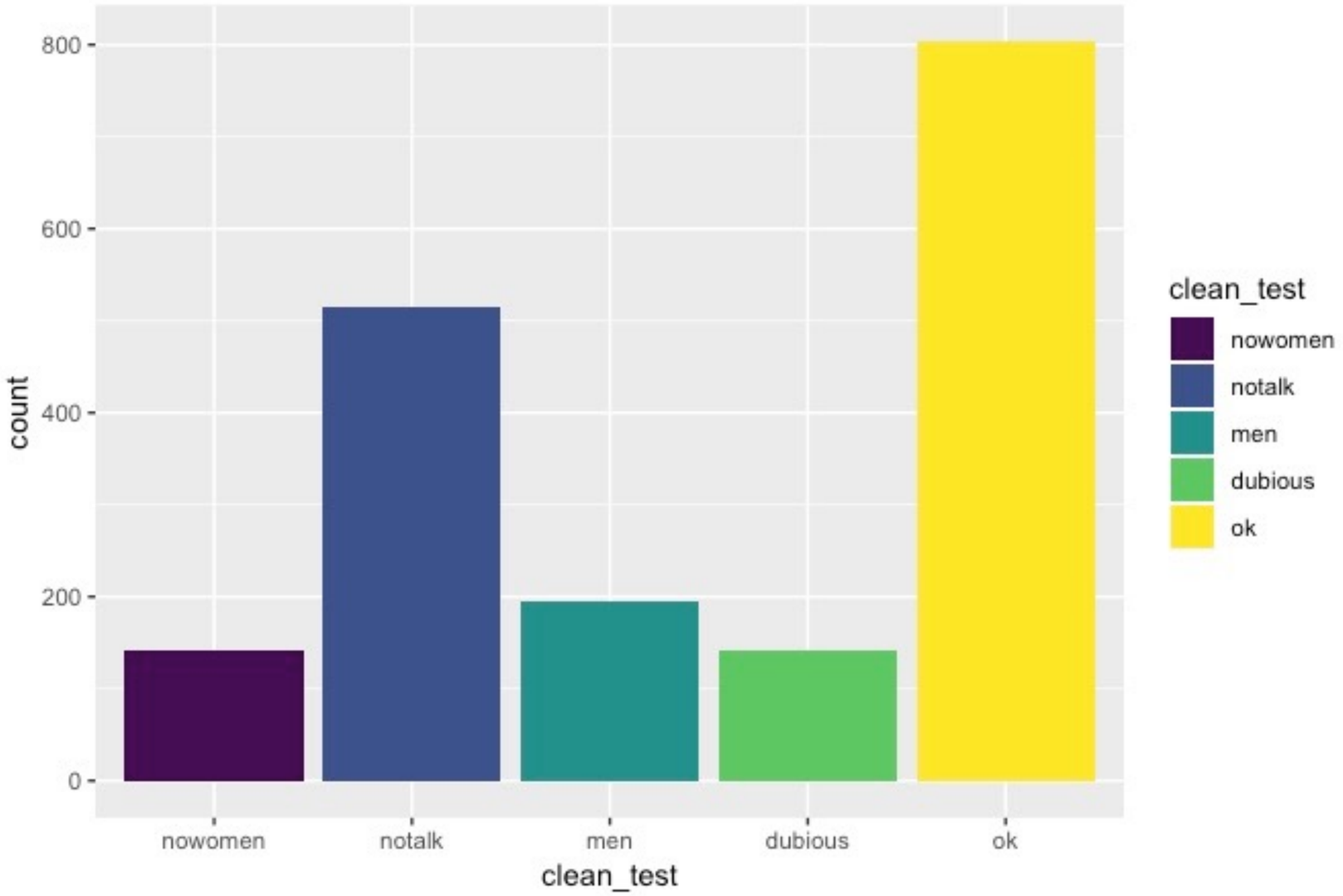
Your Turn 7

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






```
ggplot(data=bechdel) +
    geom_bar(aes(x=clean_test, color=clean_test))
```



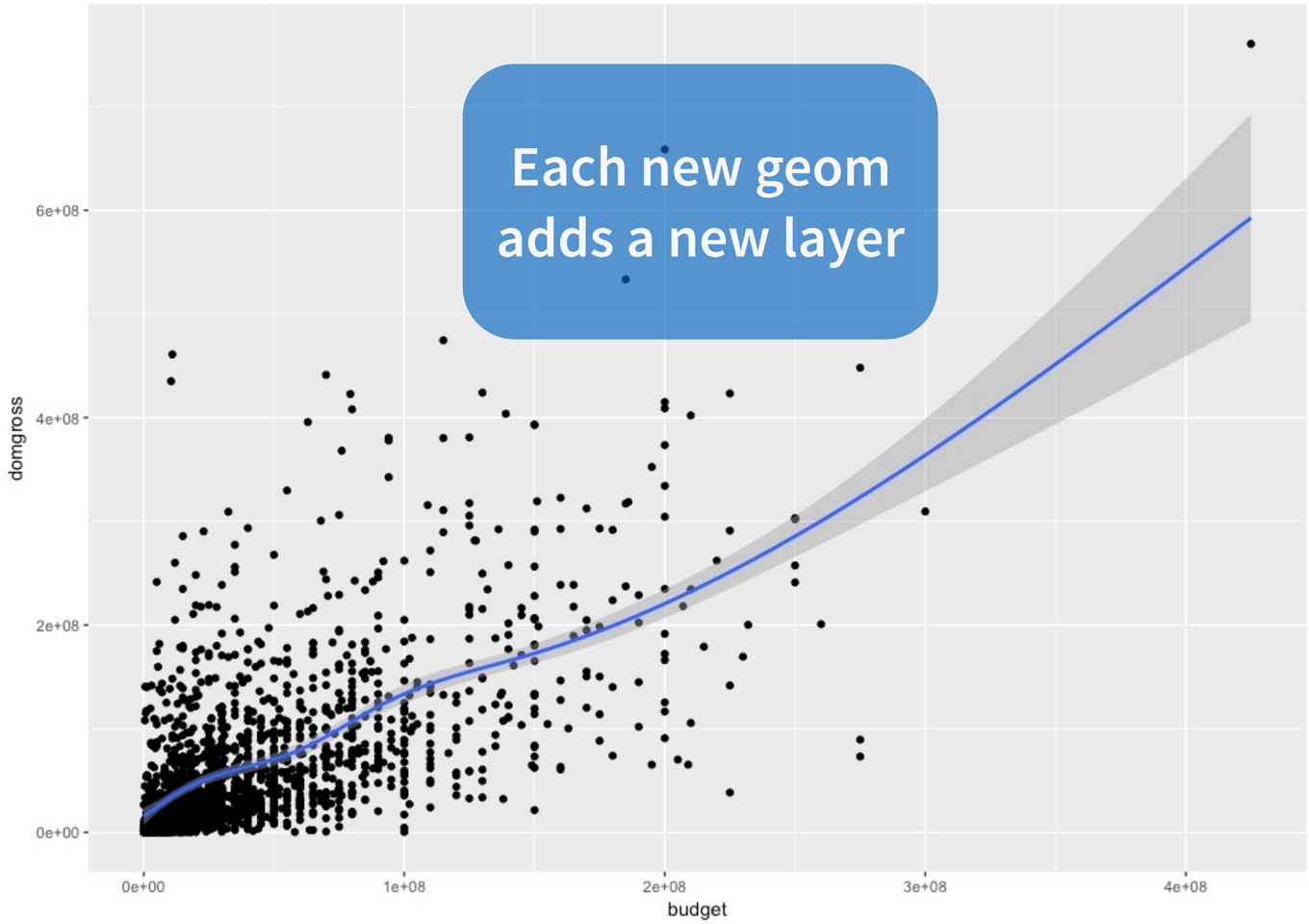

```
ggplot(data=bechdel) +
    geom_bar(aes(x=clean_test, fill=clean_test))
```

Your Turn 8

With a partner, predict what this code will do.

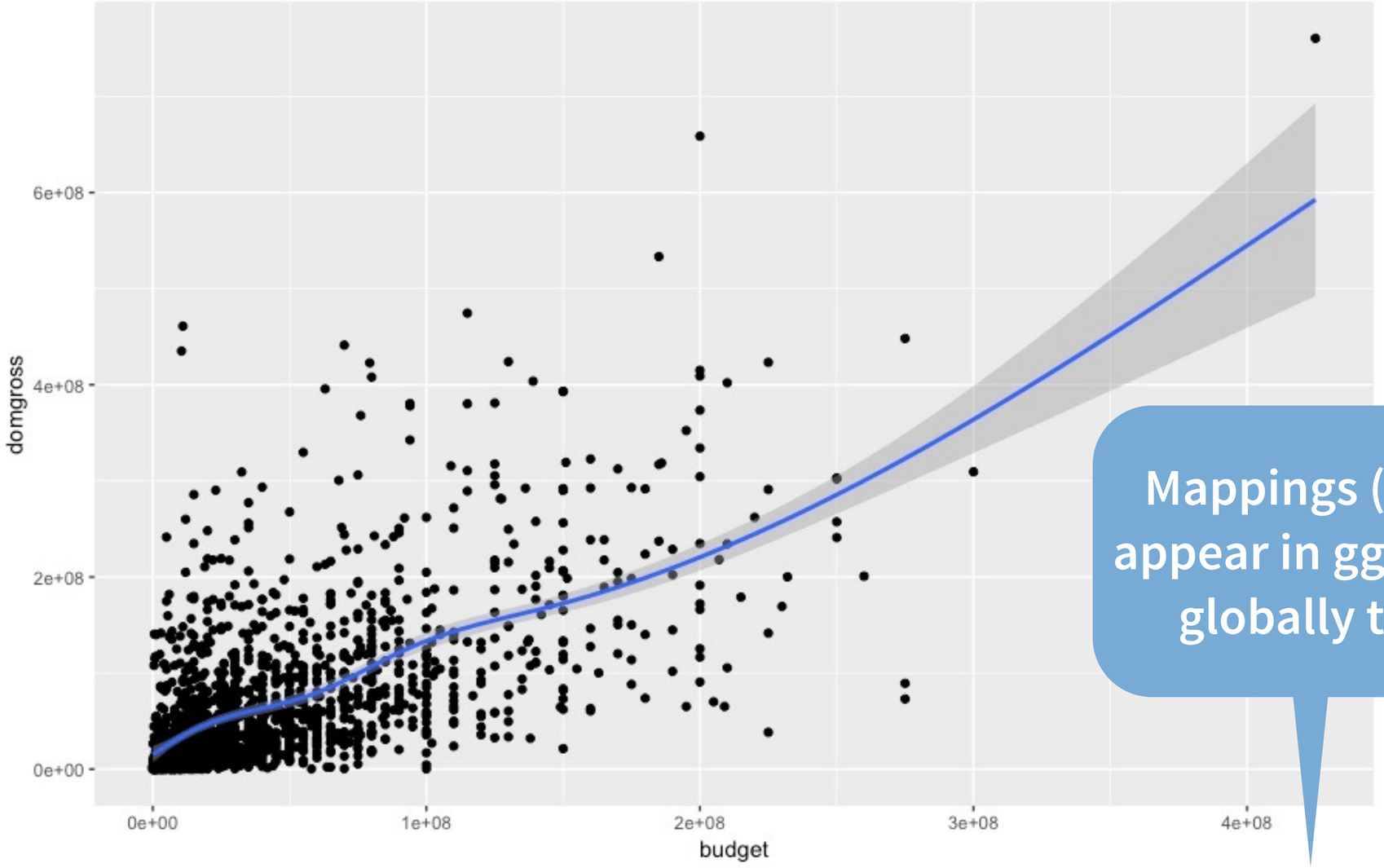
Then run it.

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross)) +
  geom_smooth(mapping = aes(x = budget, y = domgross))
```



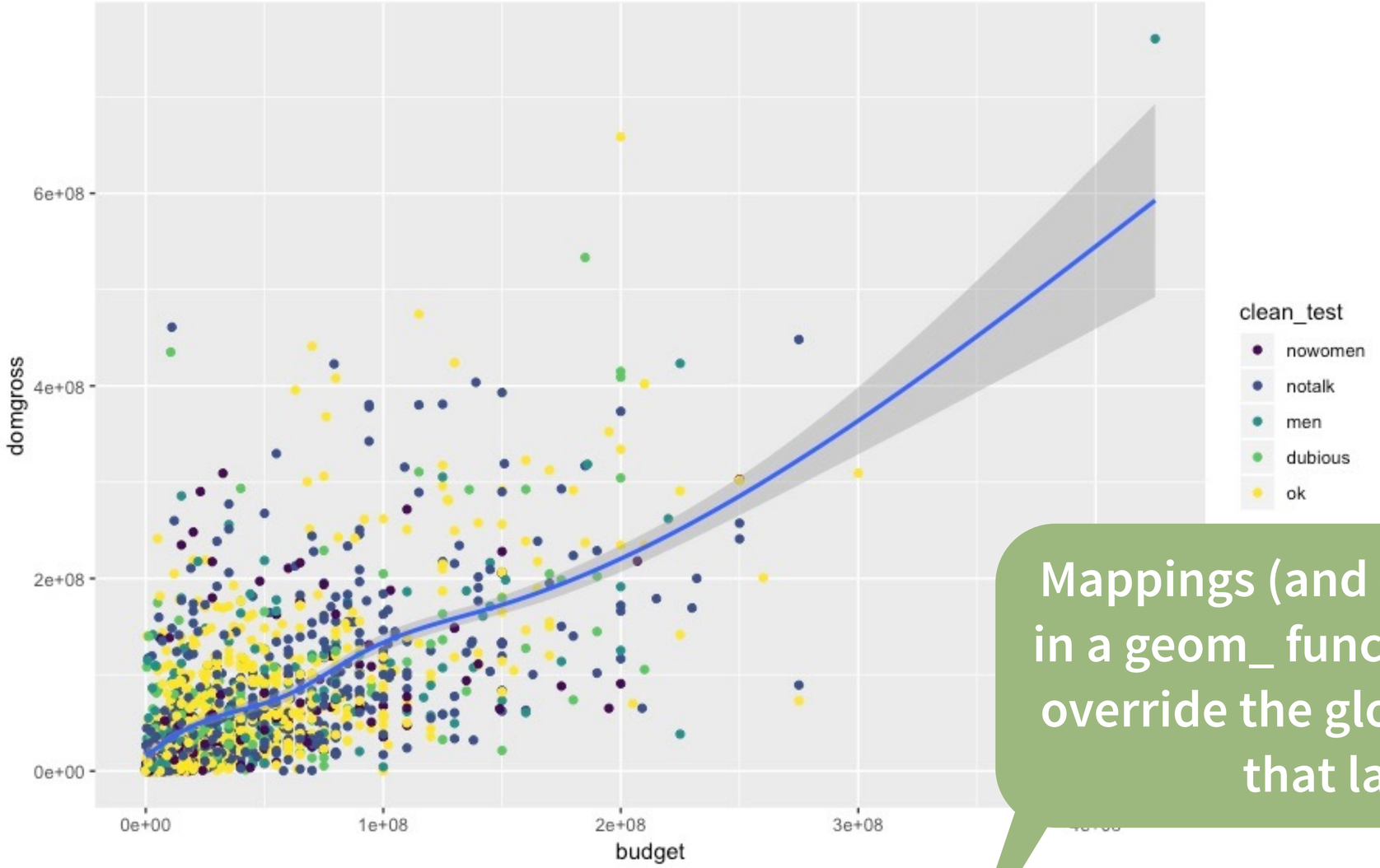

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross)) +
  geom_smooth(mapping = aes(x = budget, y = domgross))
```

global vs. local



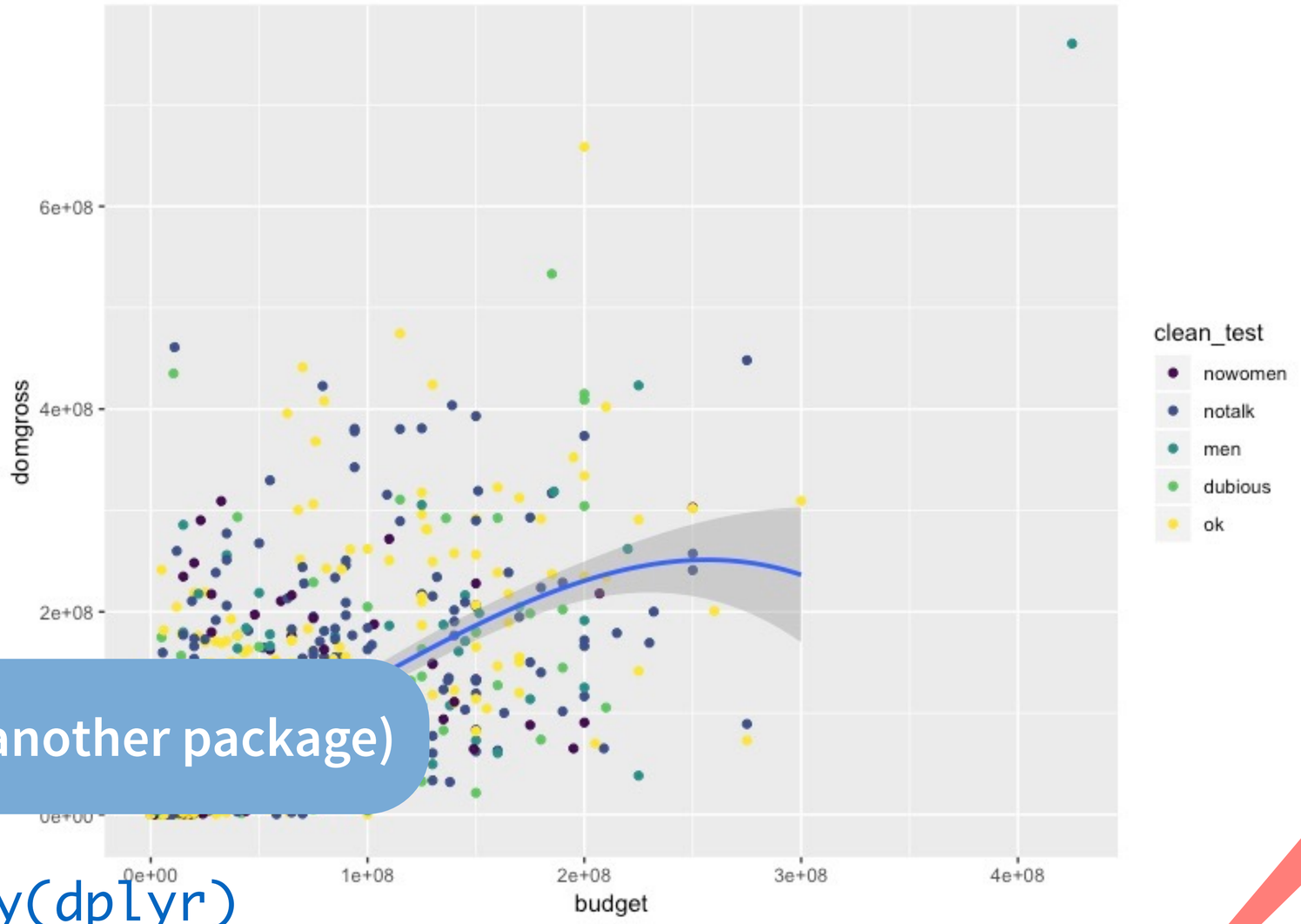
Mappings (and data) that appear in ggplot() will apply globally to every layer

```
ggplot(data = bechdel, mapping = aes(x = budget, y = domgross)) +
  geom_point() +
  geom_smooth()
```



Mappings (and data) that appear in a geom_ function will add to or override the global mappings for that layer only

```
ggplot(data = bechdel, mapping = aes(x = budget, y = domgross)) +
  geom_point(mapping = aes(color = clean_test)) +
  geom_smooth()
```



(requires another package)

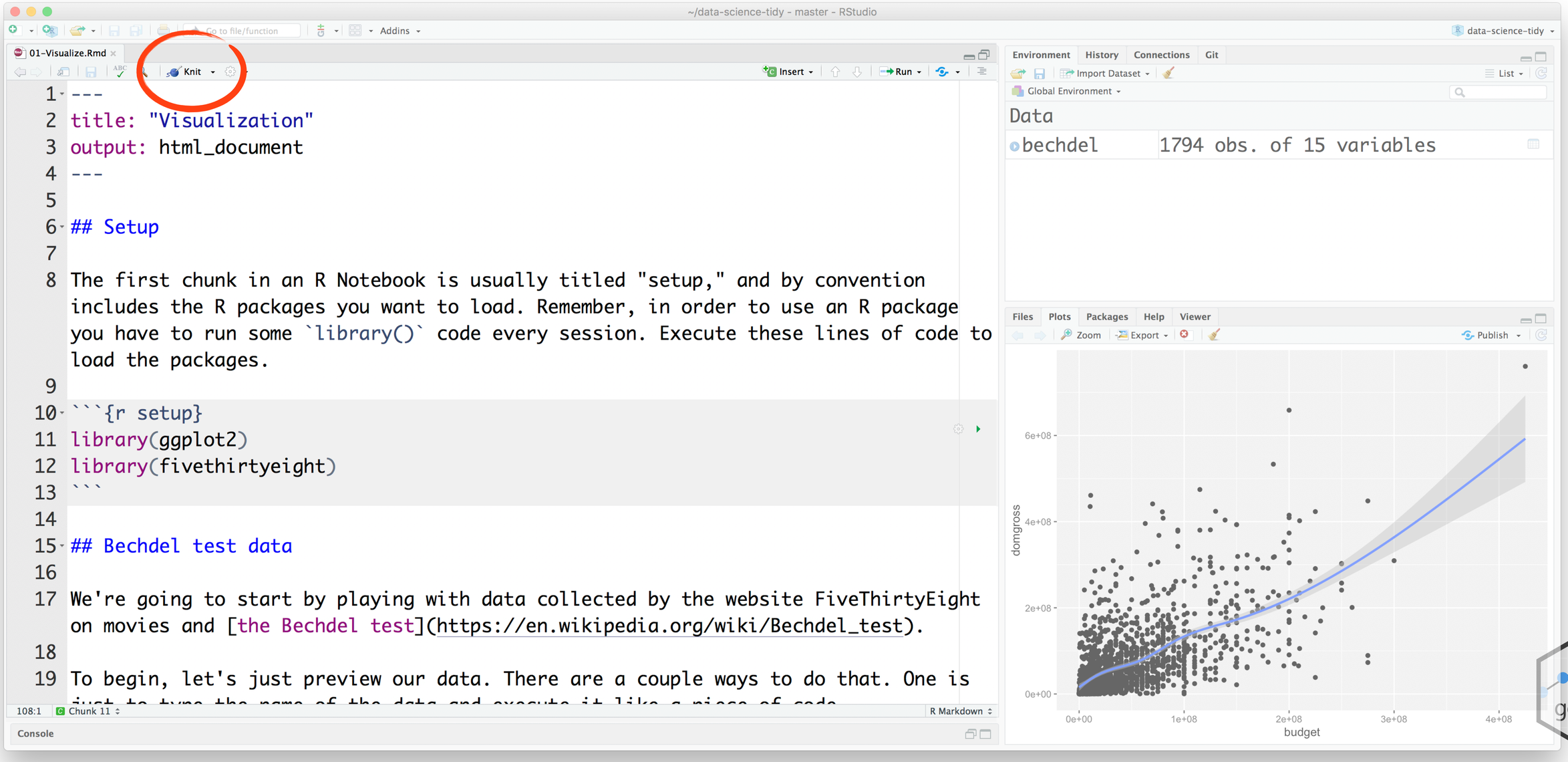
data can also be set locally or globally

```
library(dplyr)
ggplot(data = bechdel, mapping = aes(x = budget, y = domgross)) +
  geom_point(mapping = aes(color = clean_test)) +
  geom_smooth(data = filter(bechdel, clean_test == "ok"))
```

Saving graphs

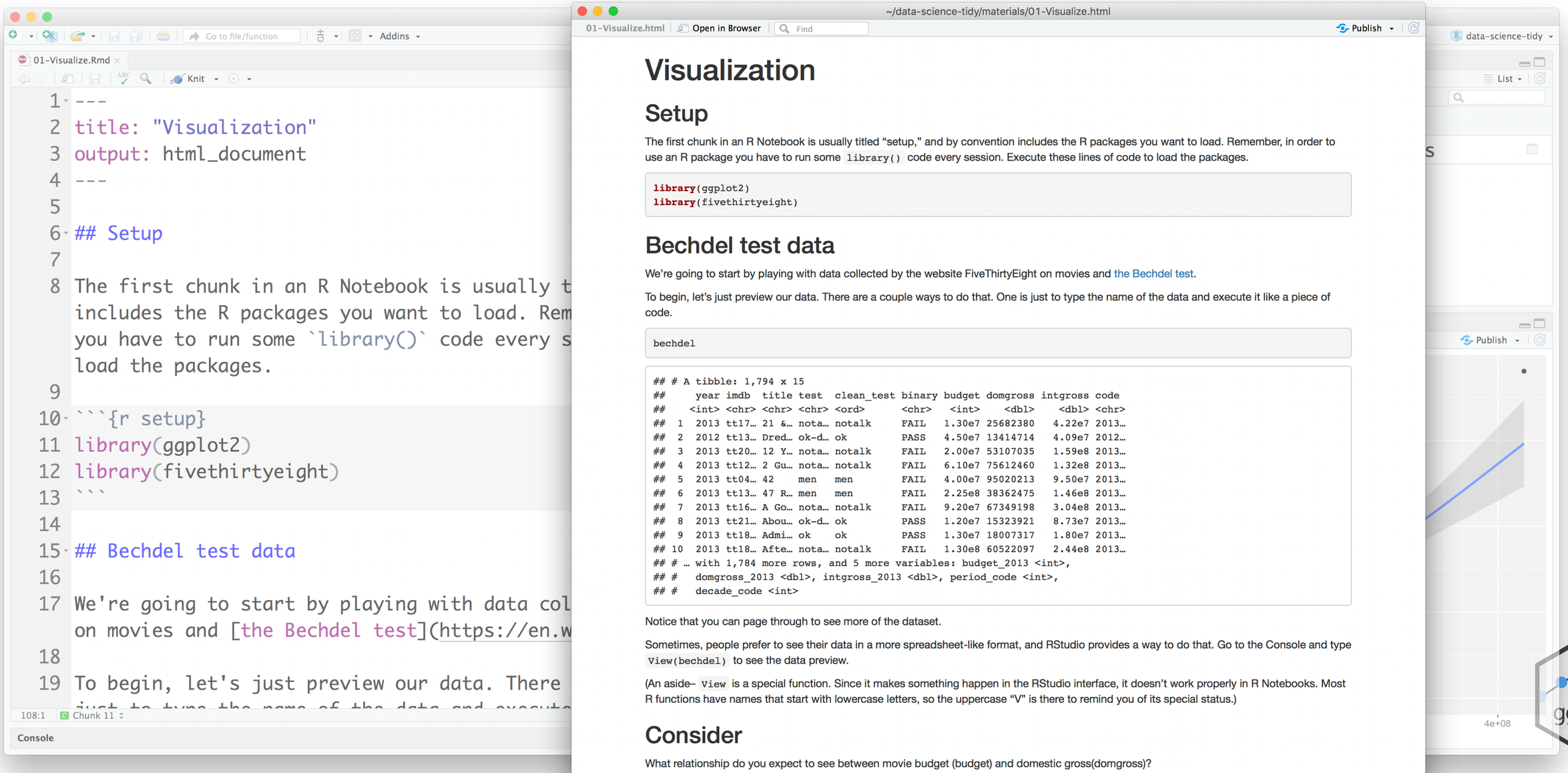
“knitting” an RMarkdown document

The easiest way to save all your work (including graphs) is to include it in an RMarkdown document. While you're working, you can “knit” your document by clicking “Knit.”



HTML preview

Now, you'll see a beautifully typeset version of what you've done!



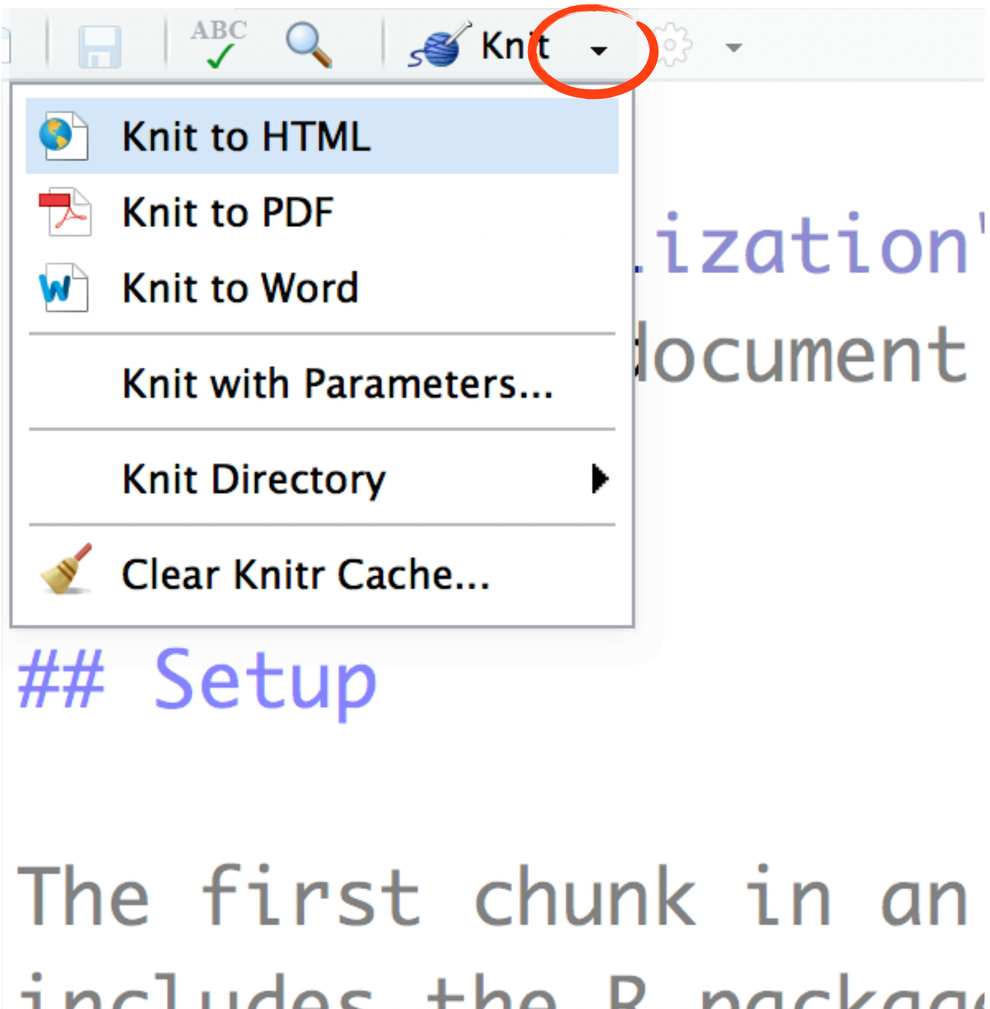
Sharing your work

The preview is something that pops up for you to look at as you work, but RStudio automatically creates a file you can share with others when you knit an RMarkdown document. By default, it creates an HTML file, and the file's name corresponds to the name of your RMarkdown document.

You can email this HTML document to anyone you'd like to share your work with!

Other formats

While RStudio automatically generates an HTML file of your work, you might want a different format. Clicking the down arrow next to Knit lets you see other options.



Your Turn

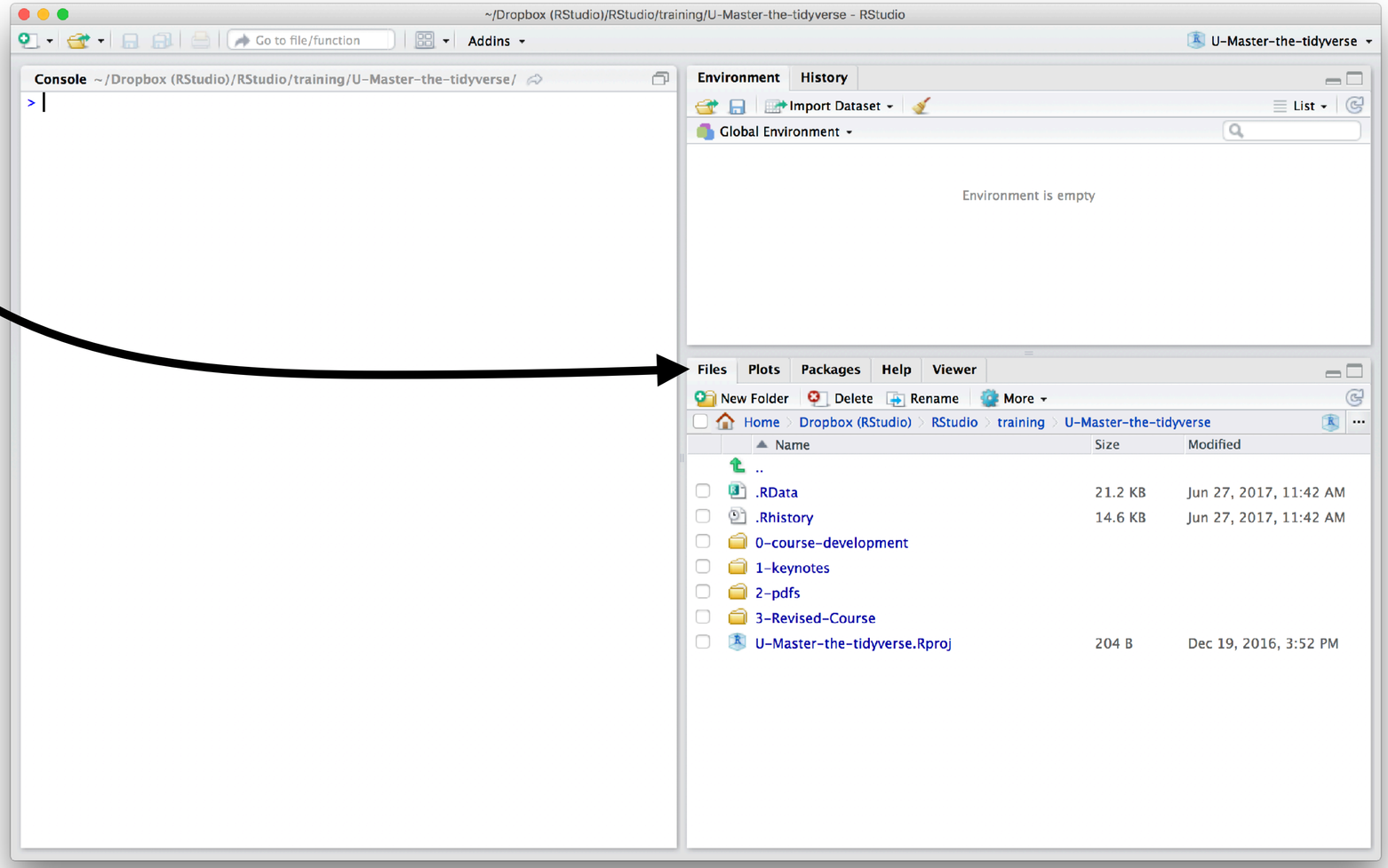
Locate the 01-Visualize.html file in your Files pane. It will be located in your **working directory**

Working directory

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

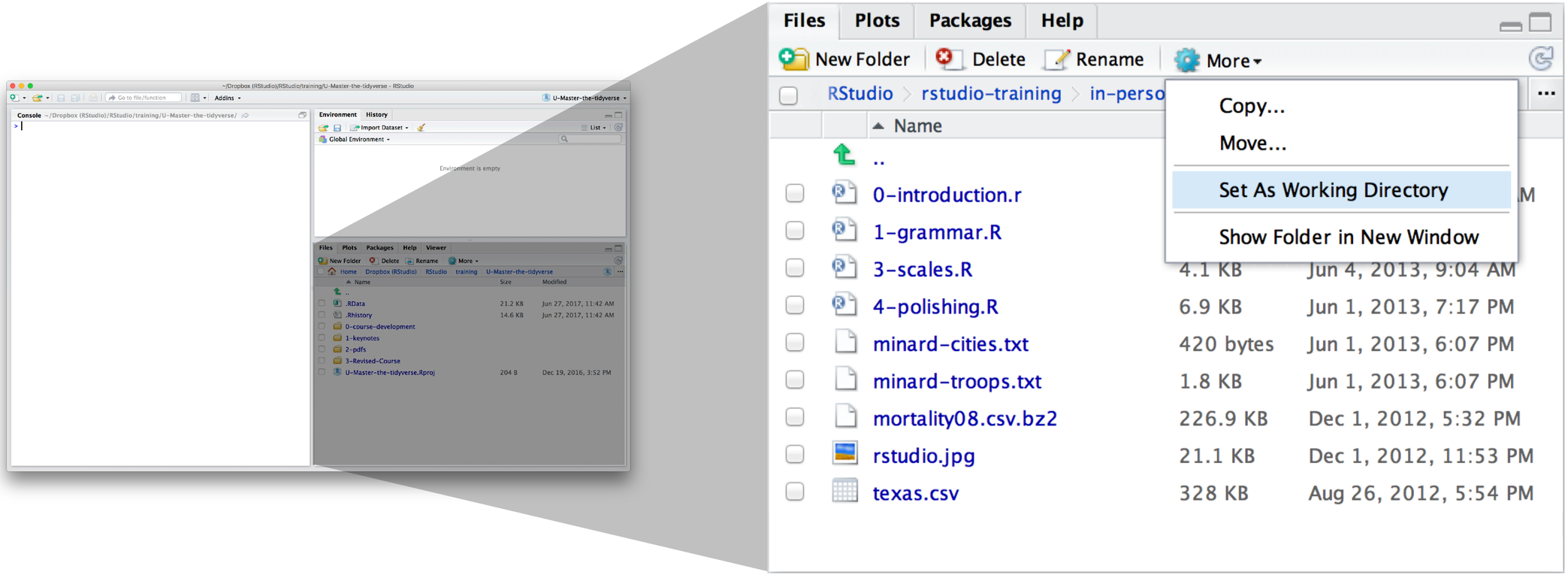
- This folder is known as your "working directory"
- When you save files, R will save them here
- When you load files, R will look for them here

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



Changing the Working directory

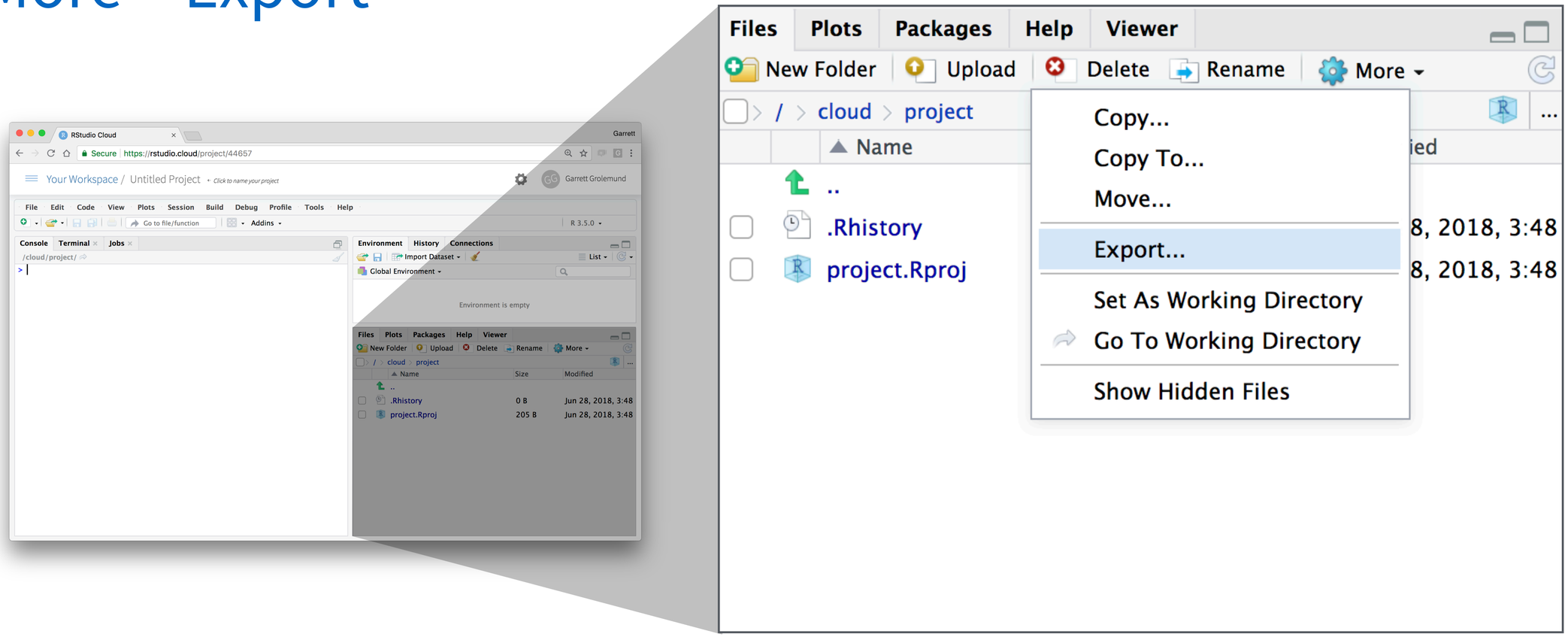
Navigate in the files pane to a new directory. Click More > Set As Working Directory



Download files

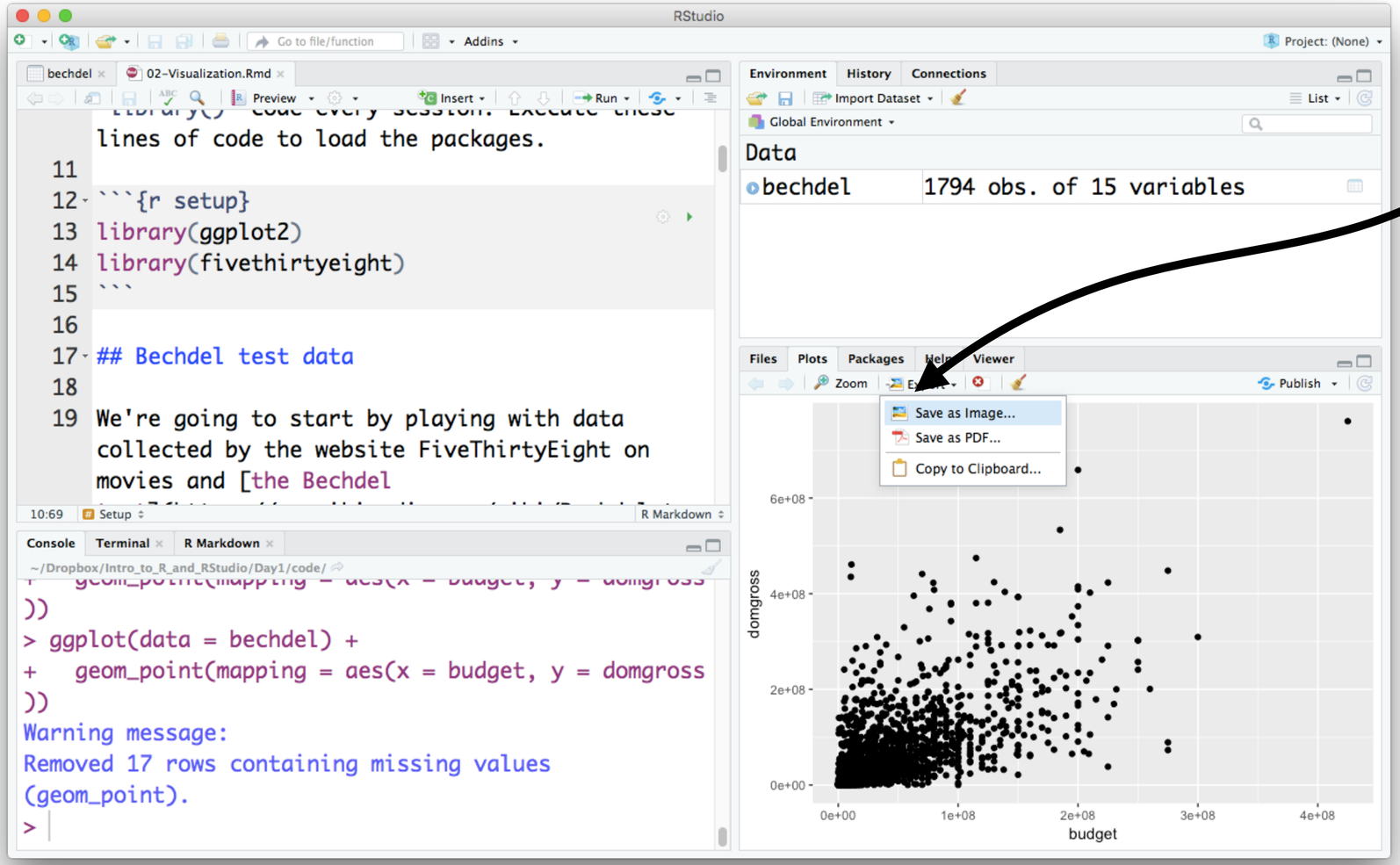
In the files pane, check next to the file(s) to download

More > Export



Manually saving plots

Save plots manually with the export menu



Saving plots

ggsave() saves the last plot.
Uses size on screen:

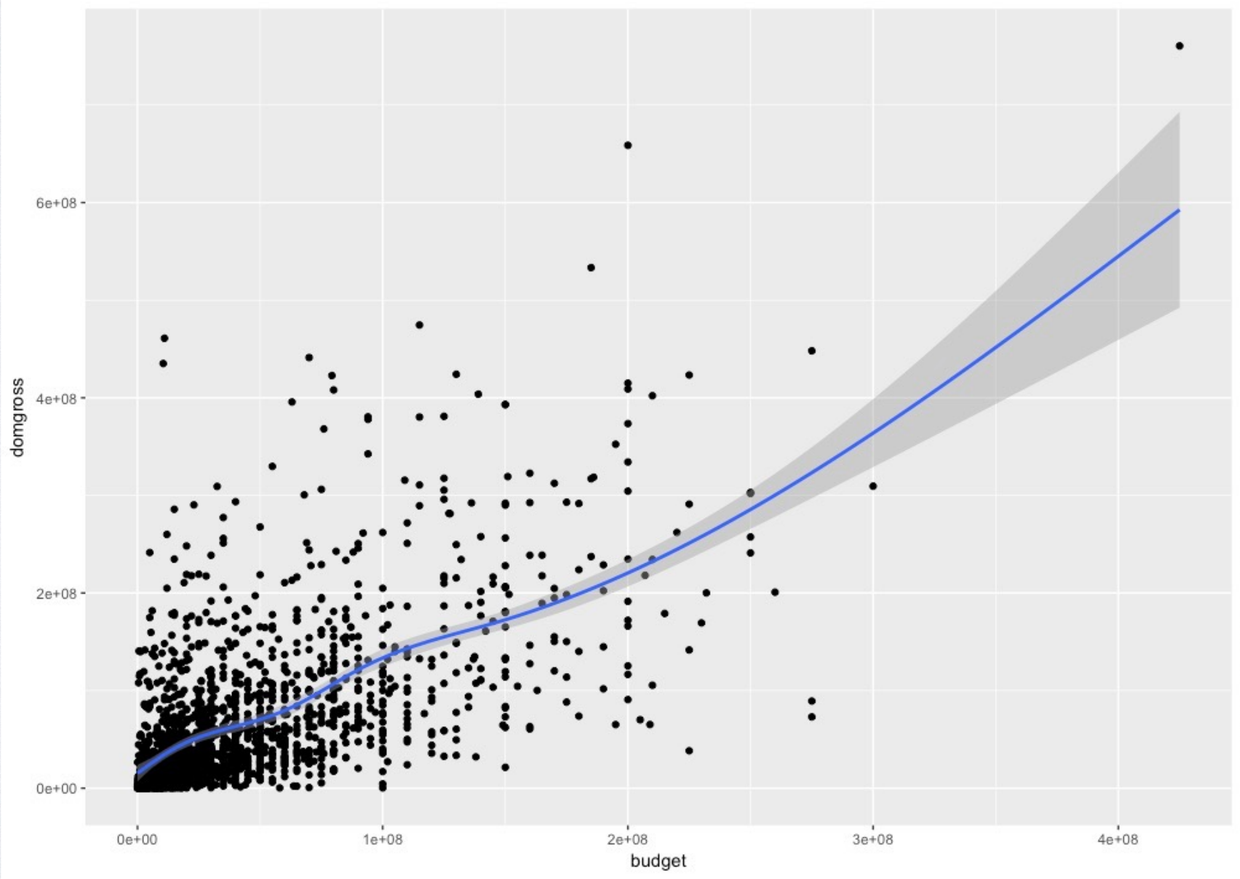
```
ggsave("my-plot.pdf")
ggsave("my-plot.png")
```

Specify size in inches

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

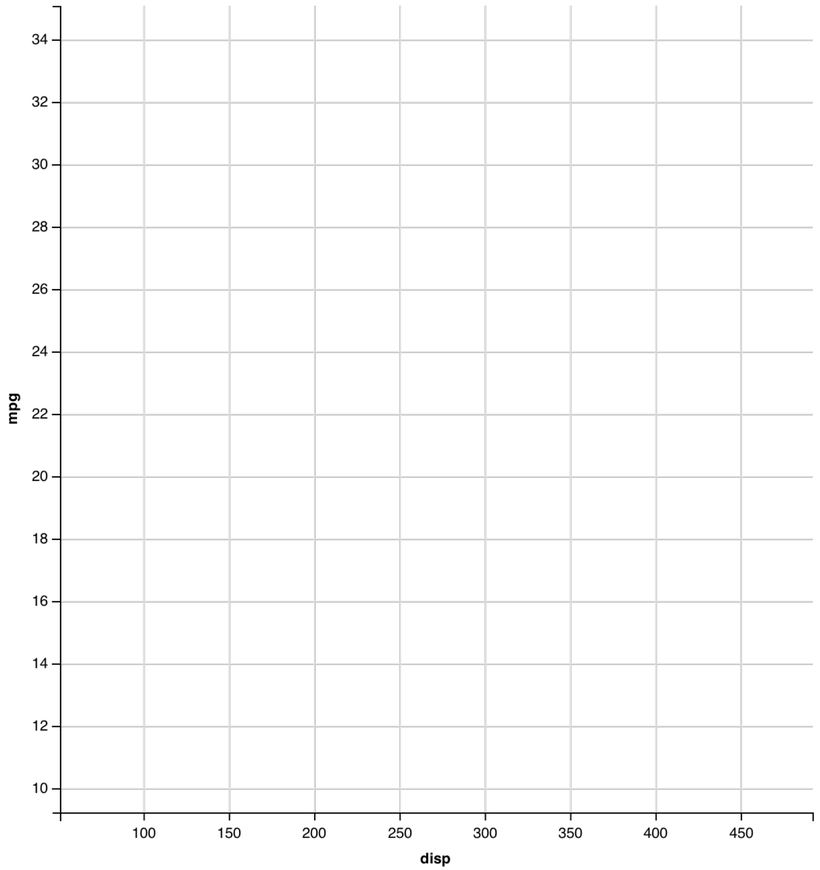
Your Turn 9

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



Grammar of Graphics

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

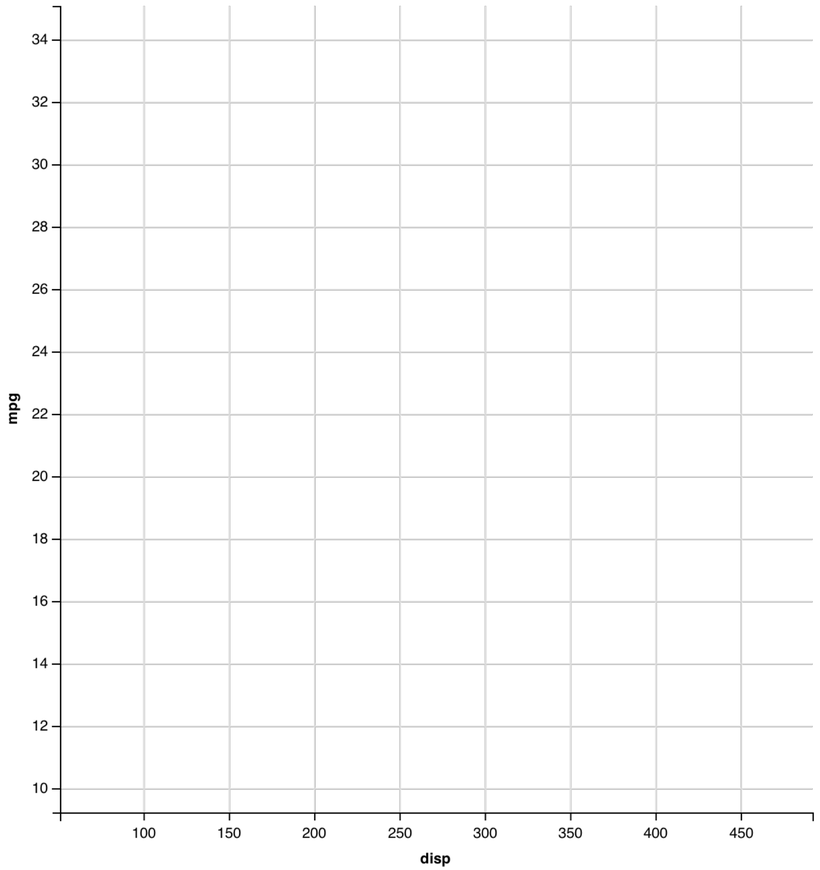


data geom

mappings

fill

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

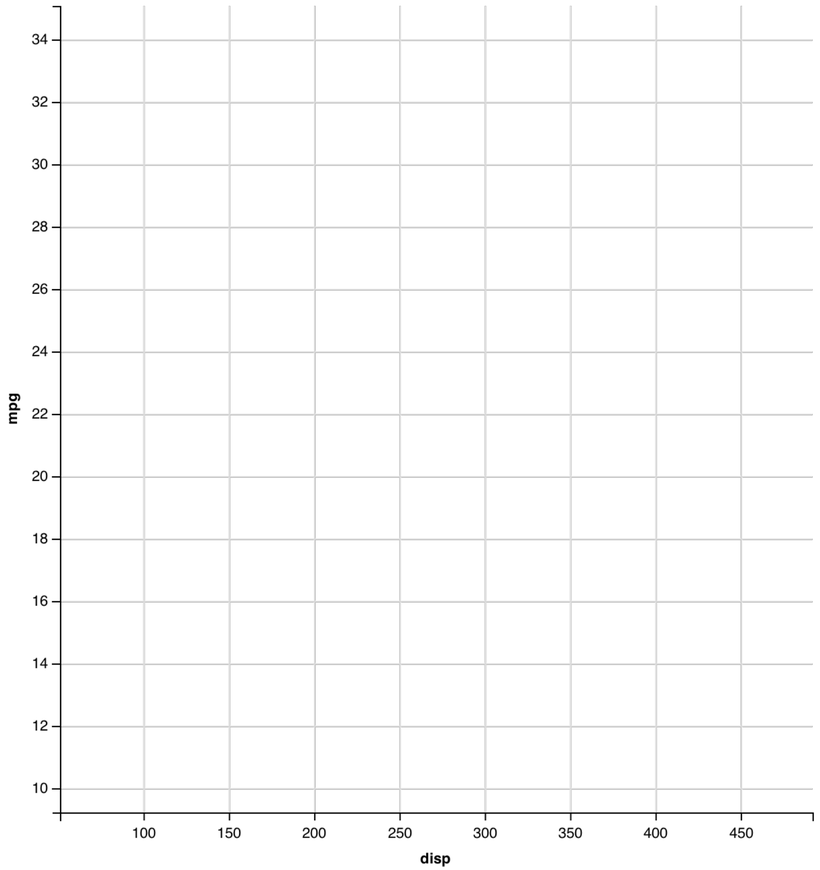


data geom

mappings

shape fill

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

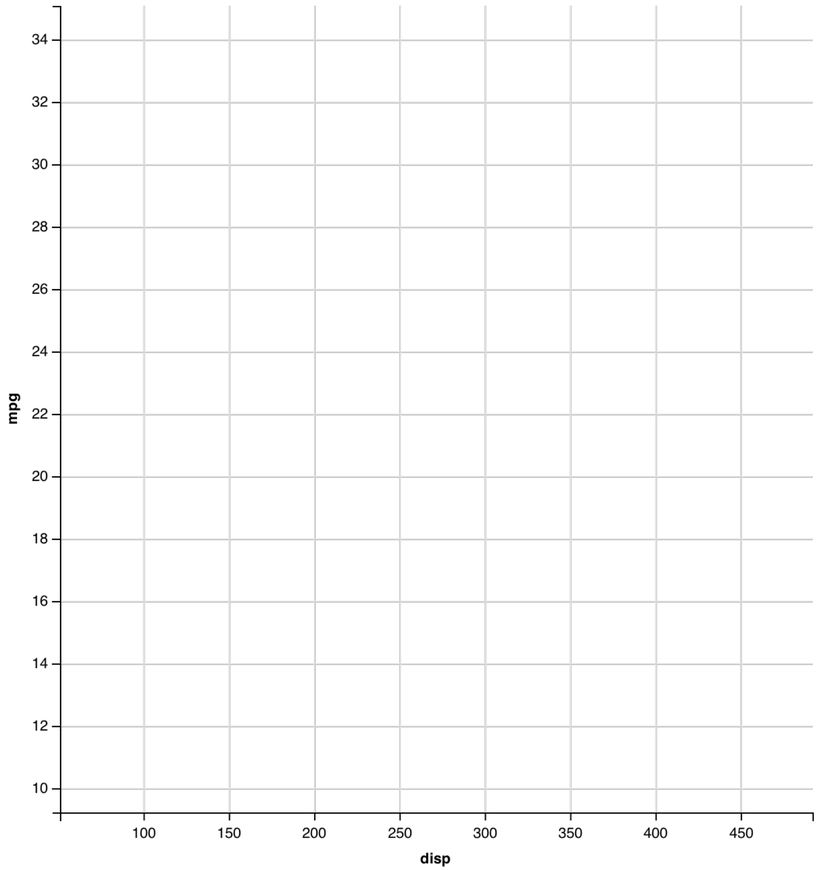


data geom

mappings

shape x fill

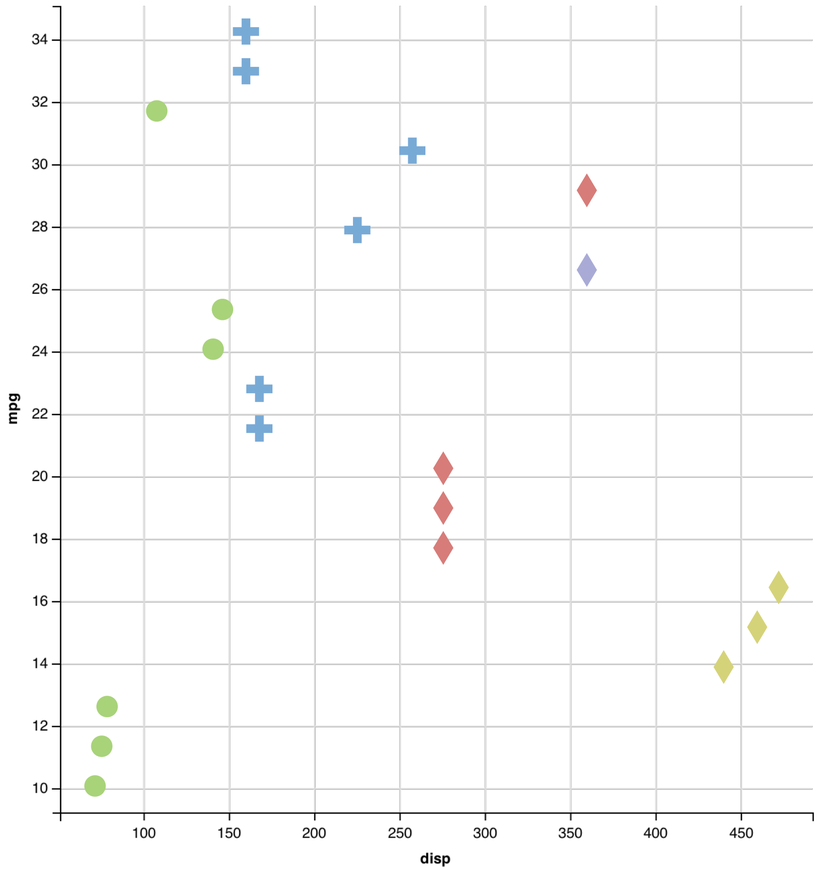
mpg	cyl	disp	hp
21.0	6	160.0	2
21.0	6	160.0	2
22.8	4	108.0	1
21.4	6	258.0	2
18.7	8	360.0	3
18.1	6	225.0	2
14.3	8	360.0	5
24.4	4	146.7	1
22.8	4	140.8	1
19.2	6	167.6	2
17.8	6	167.6	2
16.4	8	275.8	3
17.3	8	275.8	3
15.2	8	275.8	3
10.4	8	472.0	4
10.4	8	460.0	4
14.7	8	440.0	4
32.4	4	78.7	1
30.4	4	75.7	1
33.9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



data geom

mappings

y	shape	x	fill
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

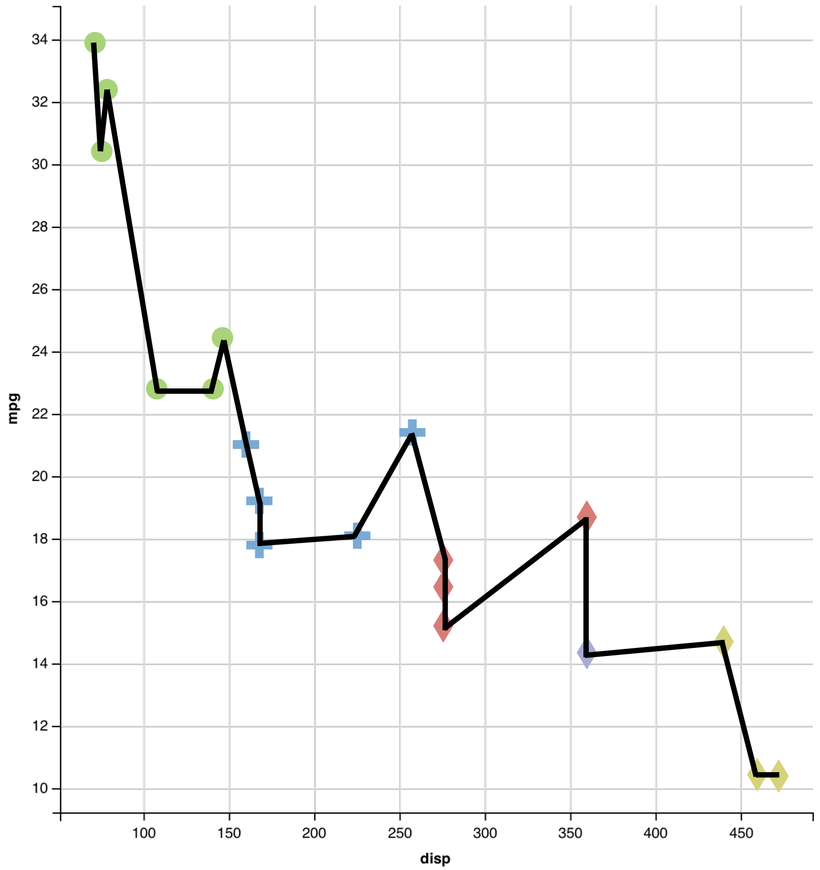


data **geom**

mappings

y shape x fill

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



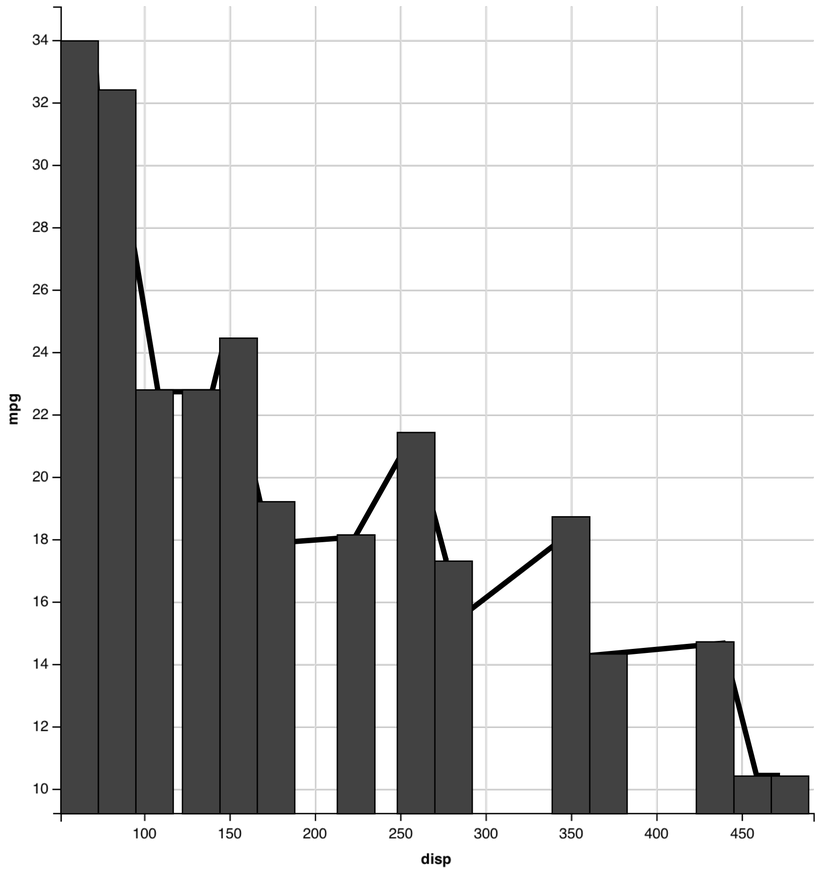
data

geom
points
lines

mappings

y x

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

points

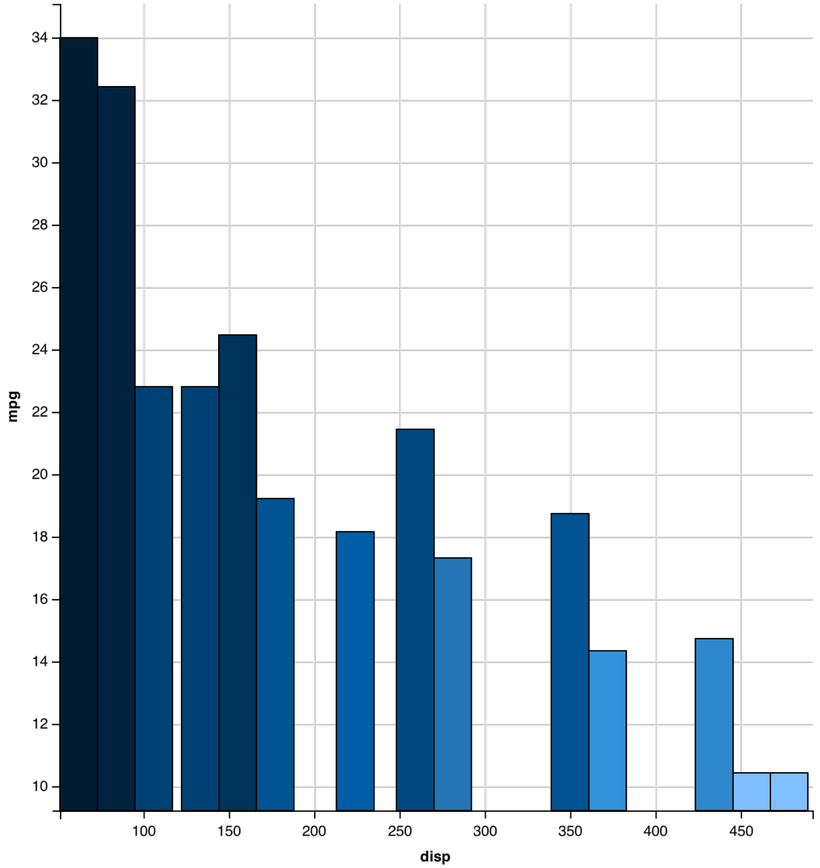
lines

bars

mappings

y fill

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

points

lines

bars

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

mappings

fill

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

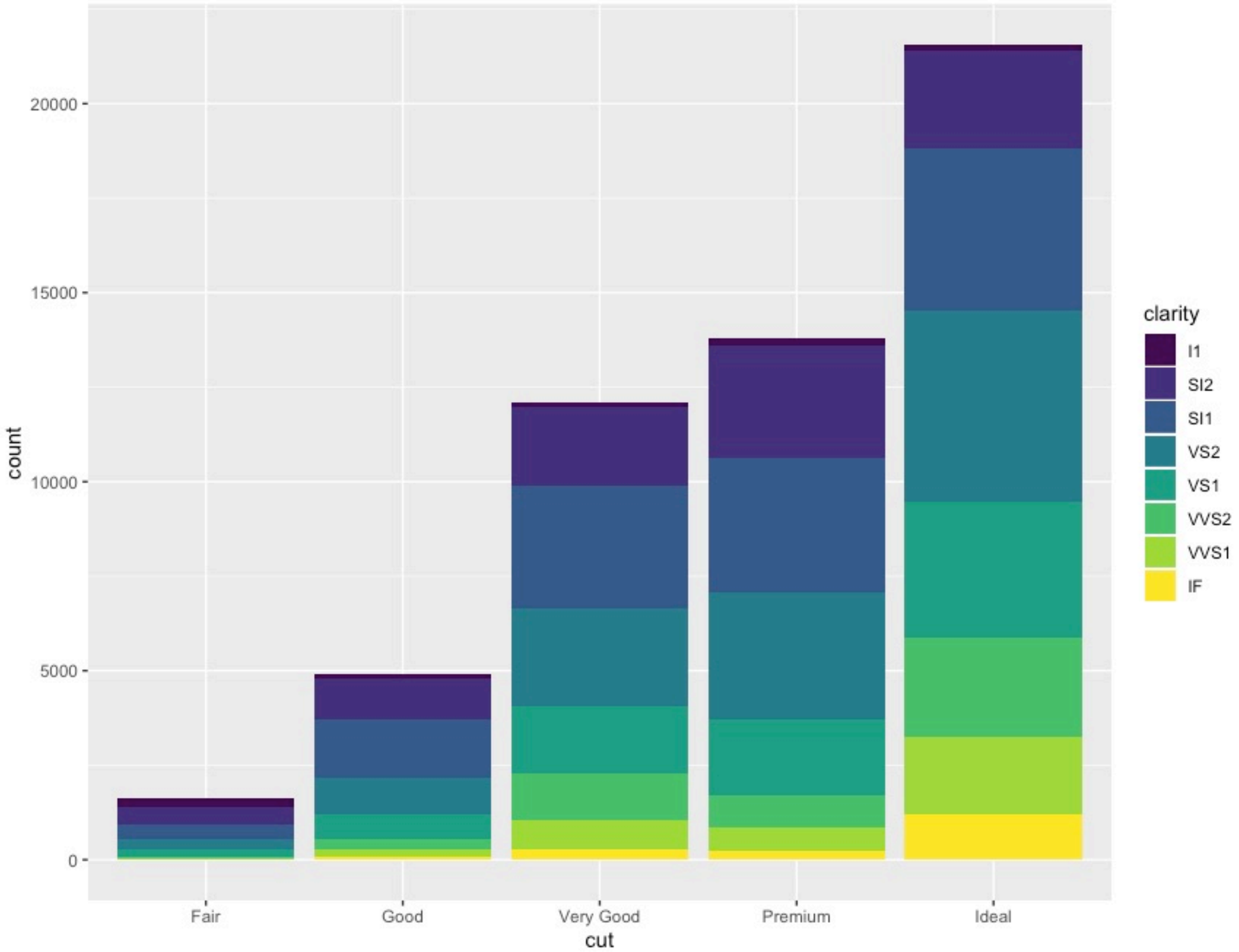
1. Pick a **data** set

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

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

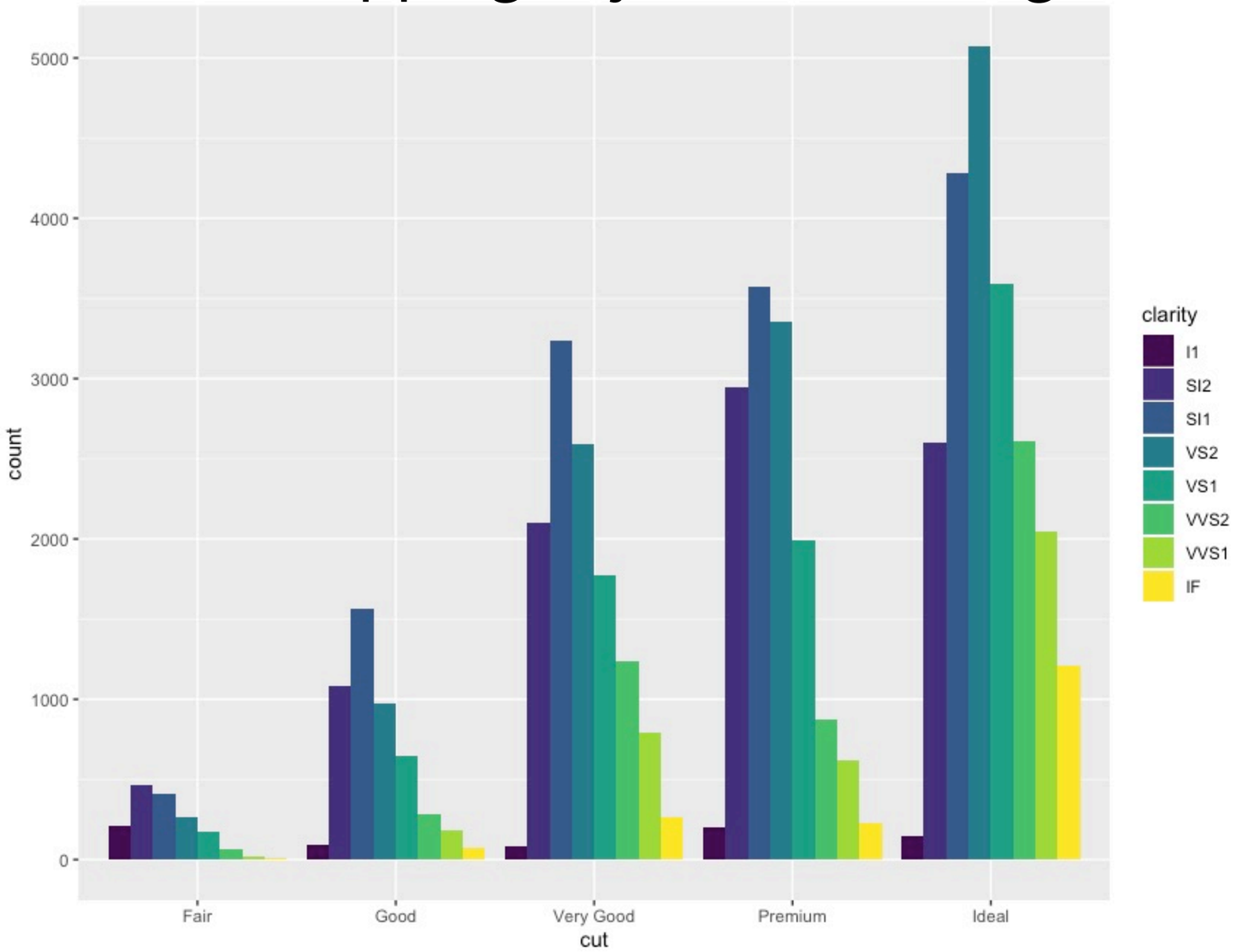
3. **Map** aesthetic properties to variables

What else?


20000
15000
10000
5000
0
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

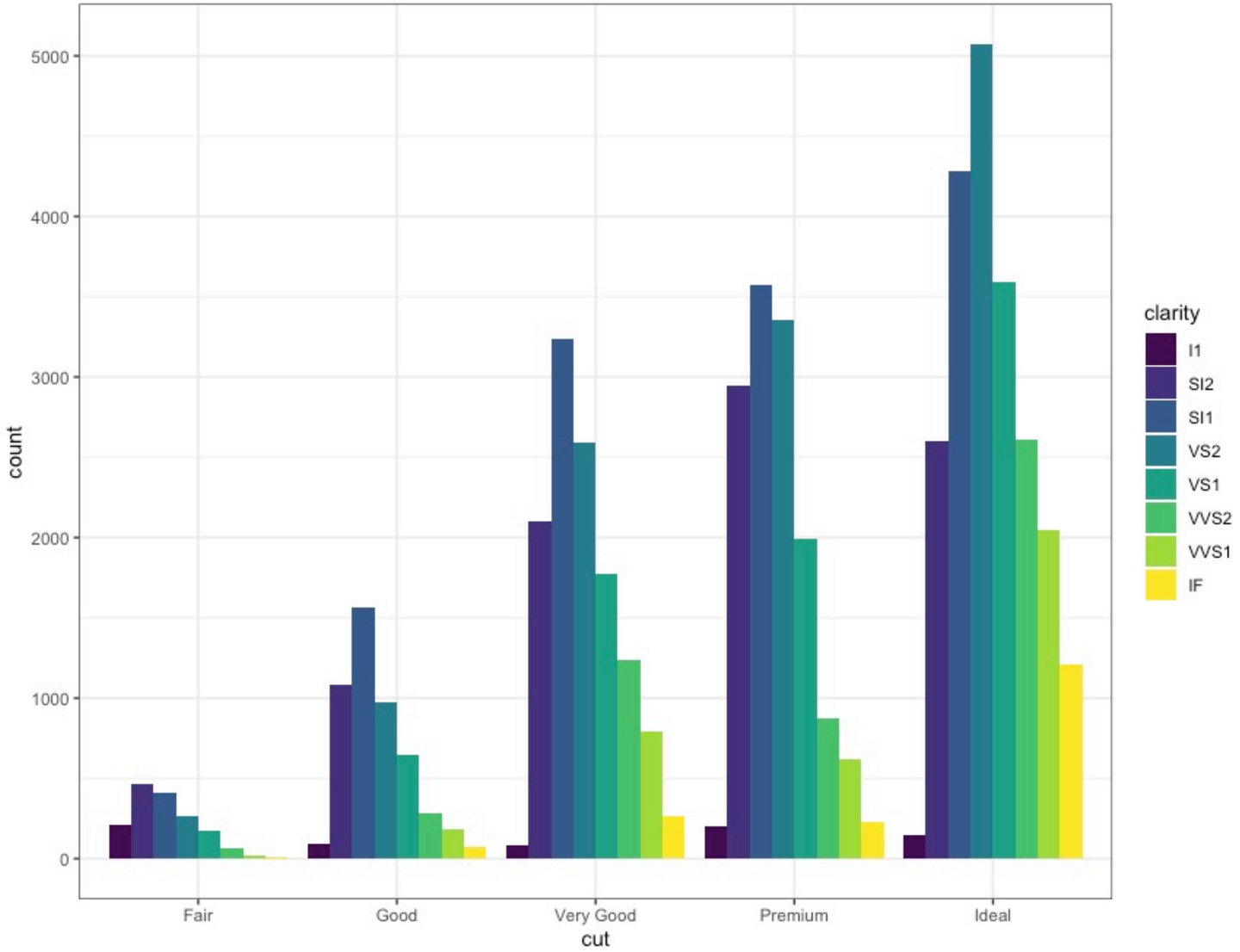
Position Adjustments

How overlapping objects are arranged



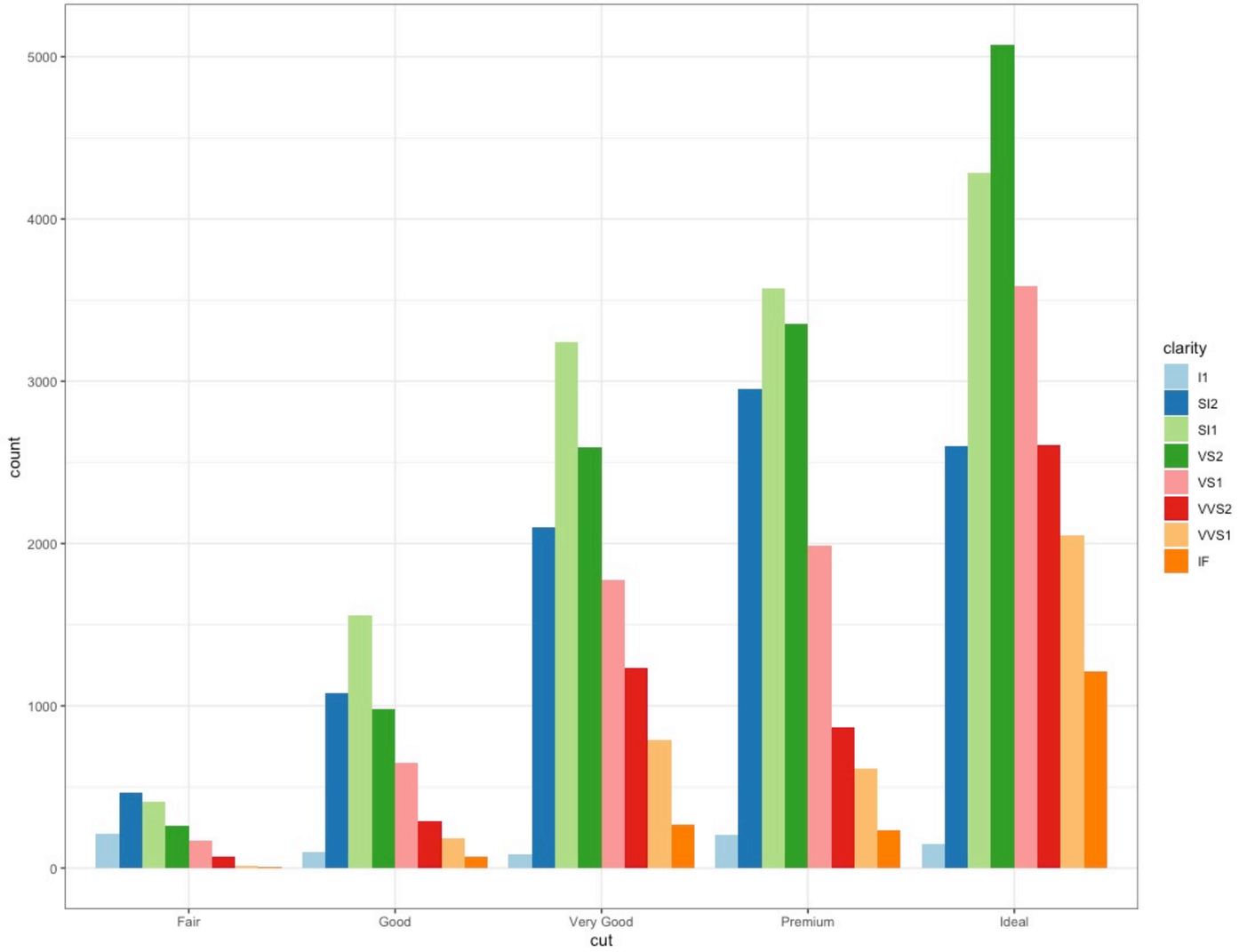
Themes

Visual appearance of non-data elements



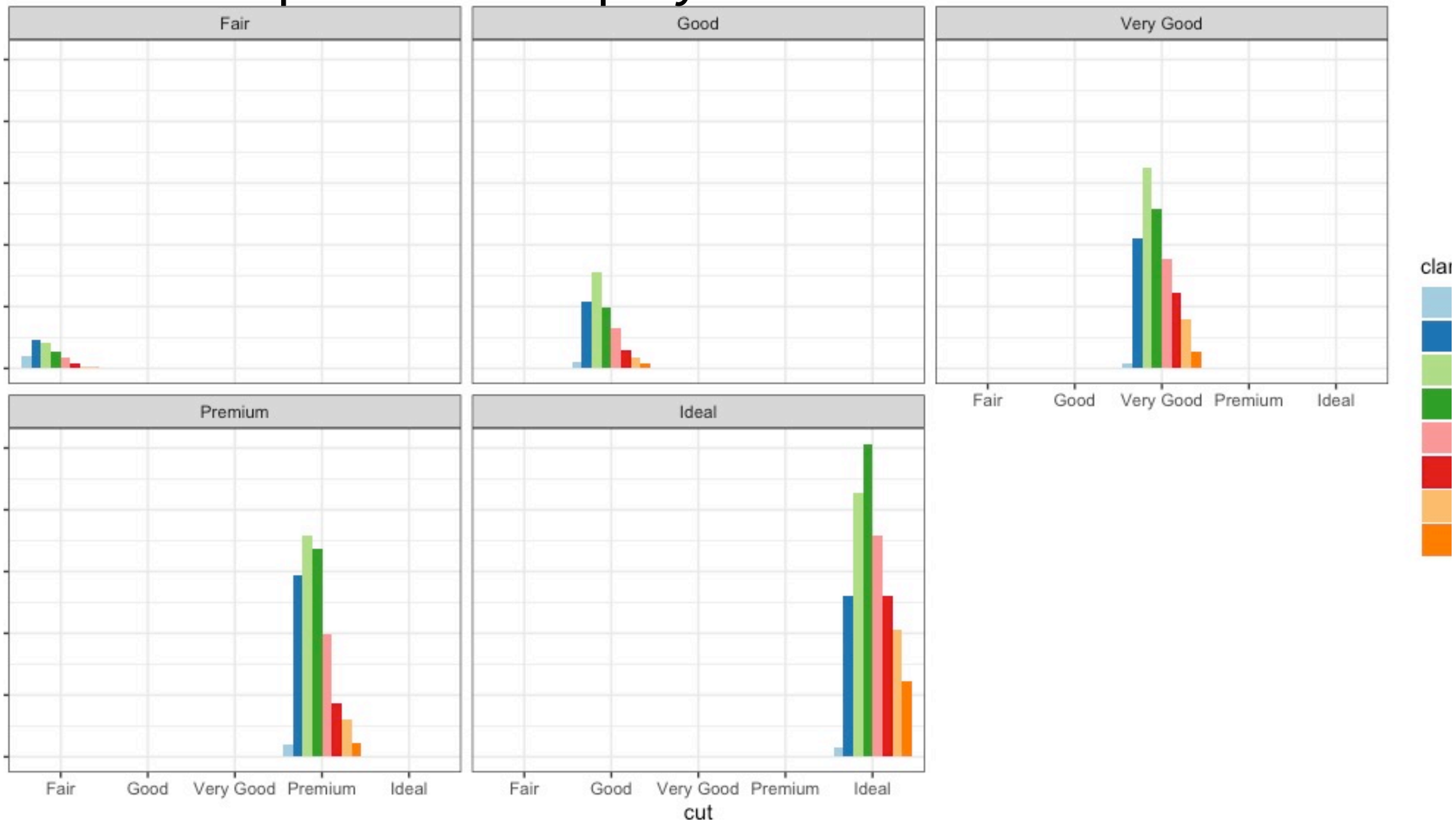
Scales

Customize color scales, other mappings

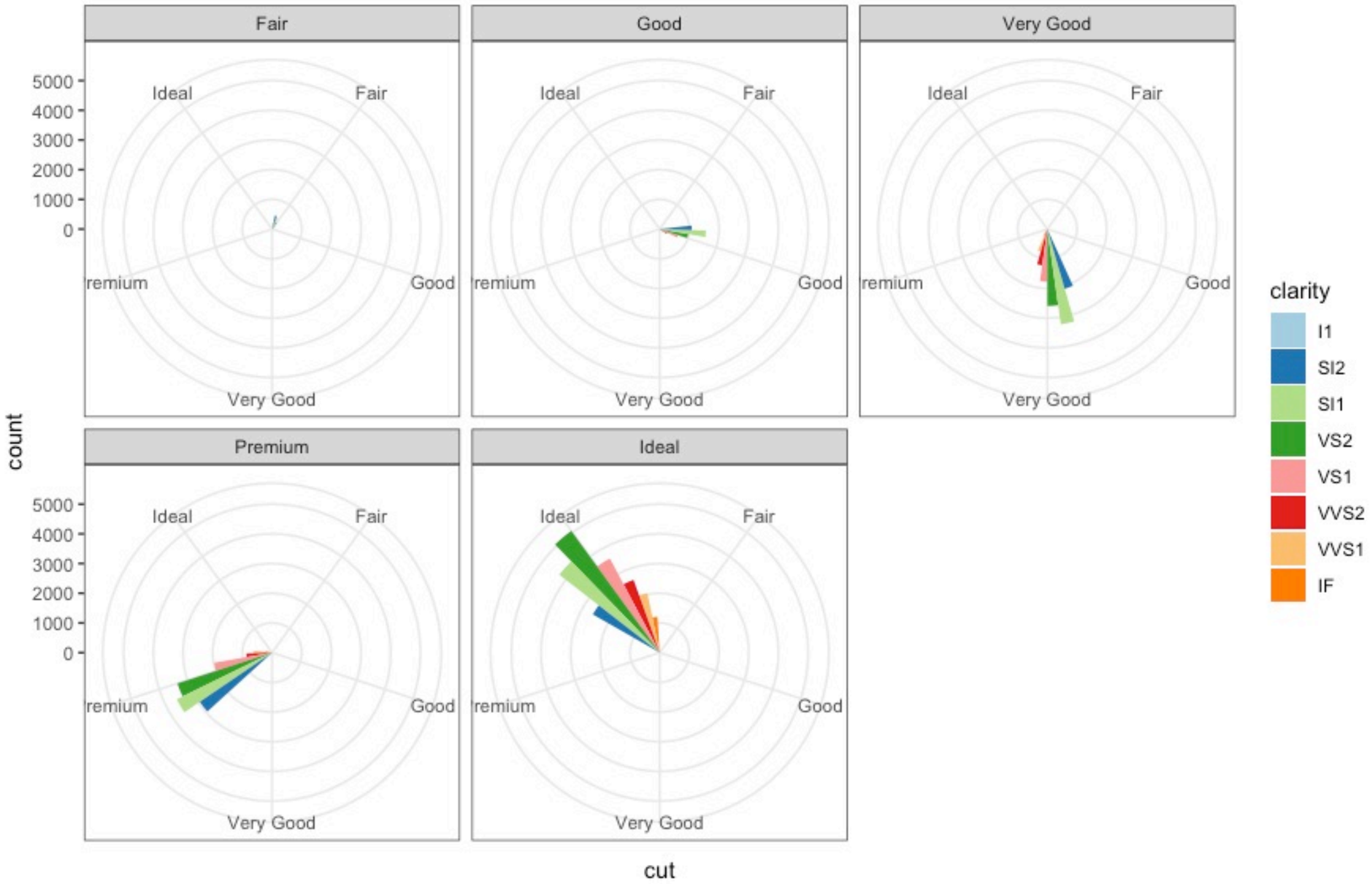


Facets

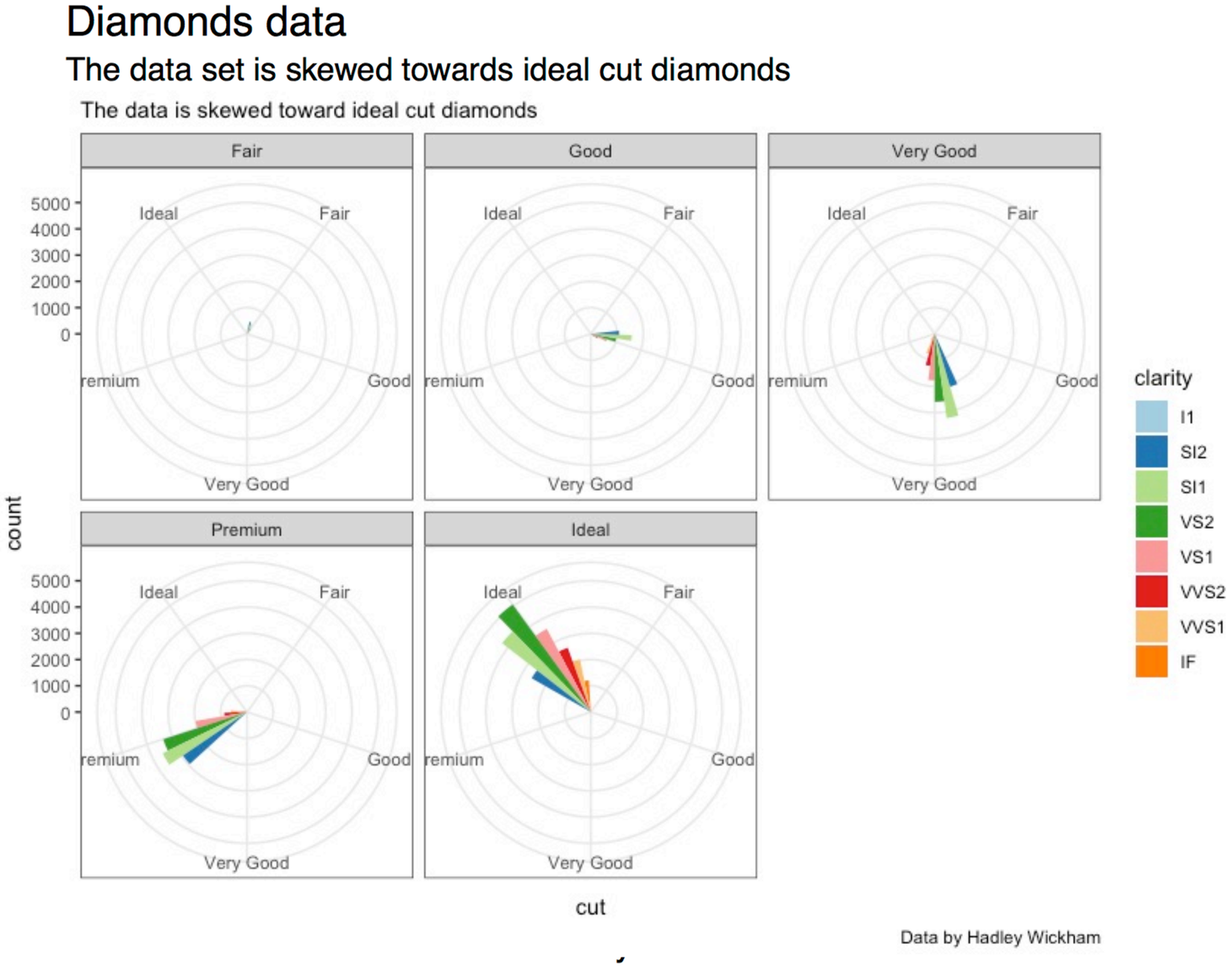
Subplots that display subsets of the data.



Coordinate systems



Titles and captions



Data by Hadley Wickham

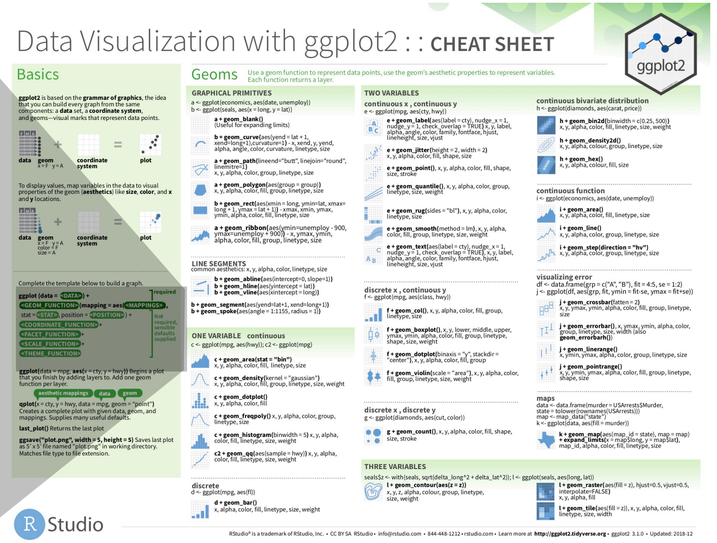
A ggplot2 template

Make any plot by filling in the parameters of this template

```
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



ggplot2.tidyverse.org

Create Elegant Data Visualisatic

https://ggplot2.tidyverse.org

ggplot2 part of the tidyverse 3.2.1

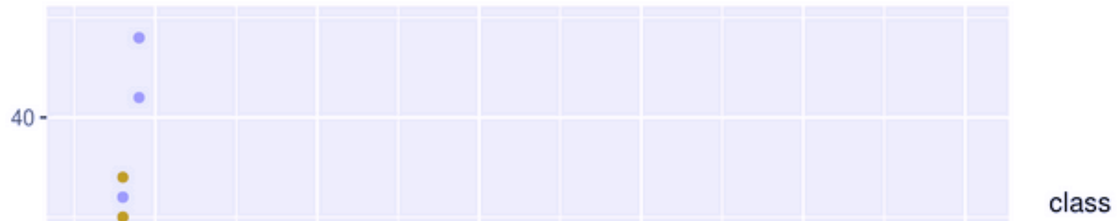
Reference Articles News Extensions

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()
```



Developers

Hadley Wickham
Author, maintainer

Winston Chang
Author

Lionel Henry
Author

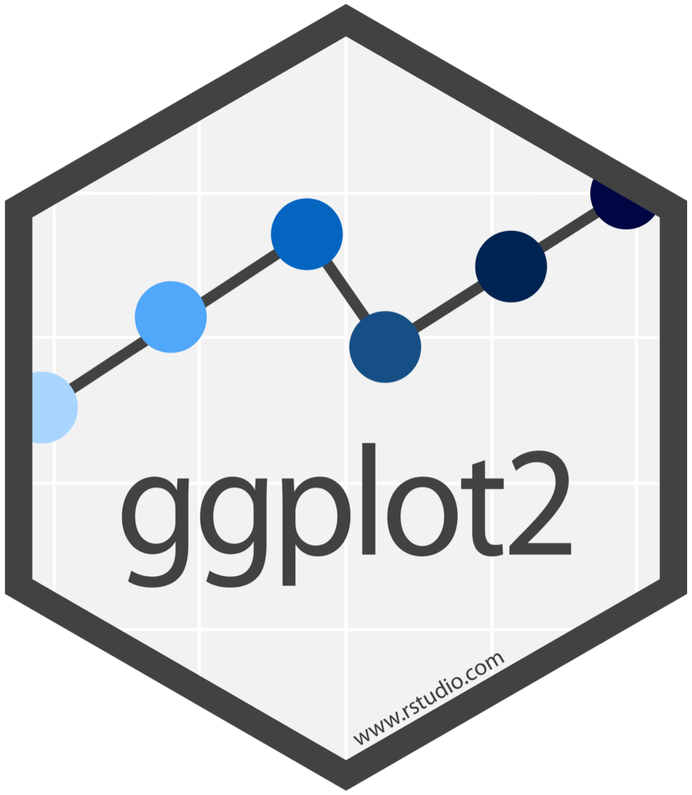
Thomas Lin Pedersen
Author

Kohske Takahashi
Author

Claus Wilke
Author

Kara Woo
Author

Hiroaki Yutani
Author

All authors...

Visualize Data with