ANT 6973: DATA VISUALIZATION AND EXPLORATION

VISUALIZING RELATIONSHIPS AND CHANGE OVER TIME

TODAY'S TOPICS

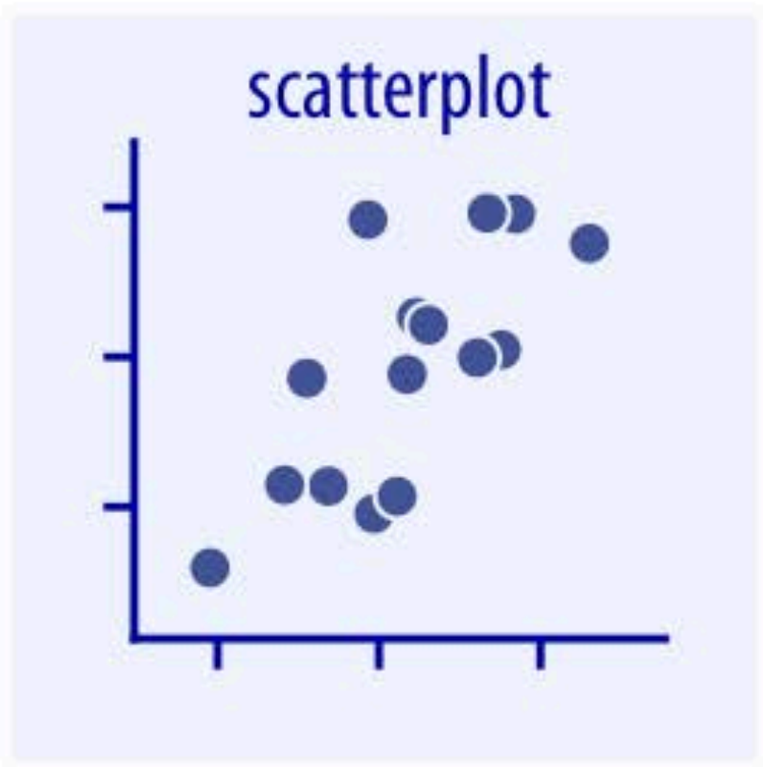
- Visualizing relationships between variables
- Visualizing change over time
- Activity:
  - Baby names

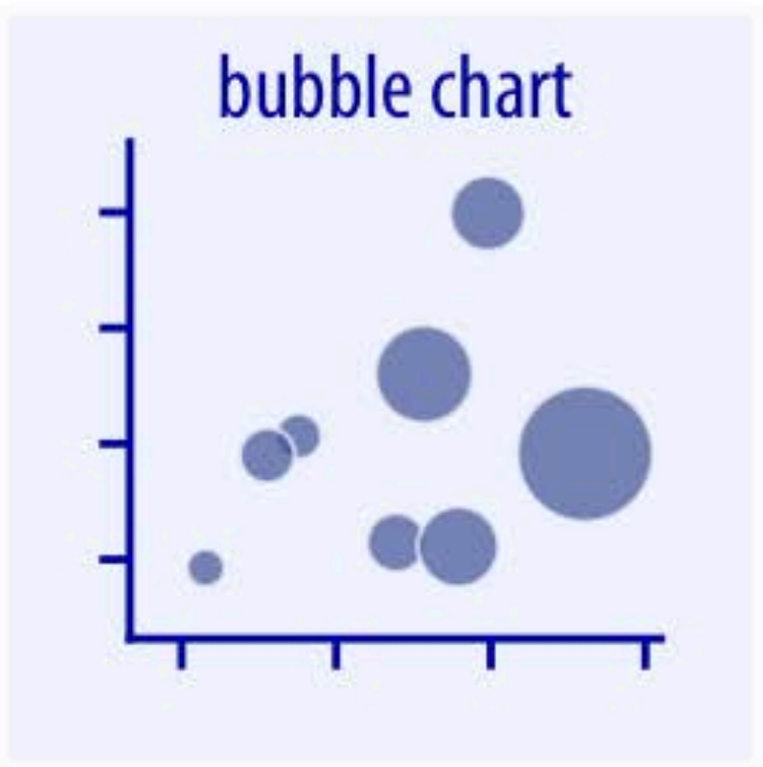
NEW SKILLS

- More practice filtering: multiple and nested logical tests
- Annotations, including positioning and justifying text and other marks
- Extend the axes with `expand_limits()`
- Axis scales: breaks and labels

VISUALIZING RELATIONSHIPS BETWEEN TWO VARIABLES

VISUALIZING RELATIONSHIPS BETWEEN TWO VARIABLES

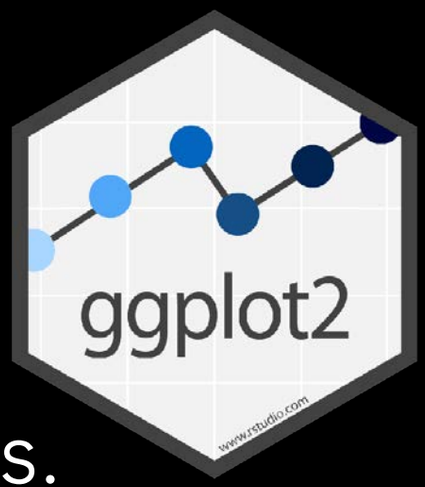




SCATTERPLOTS

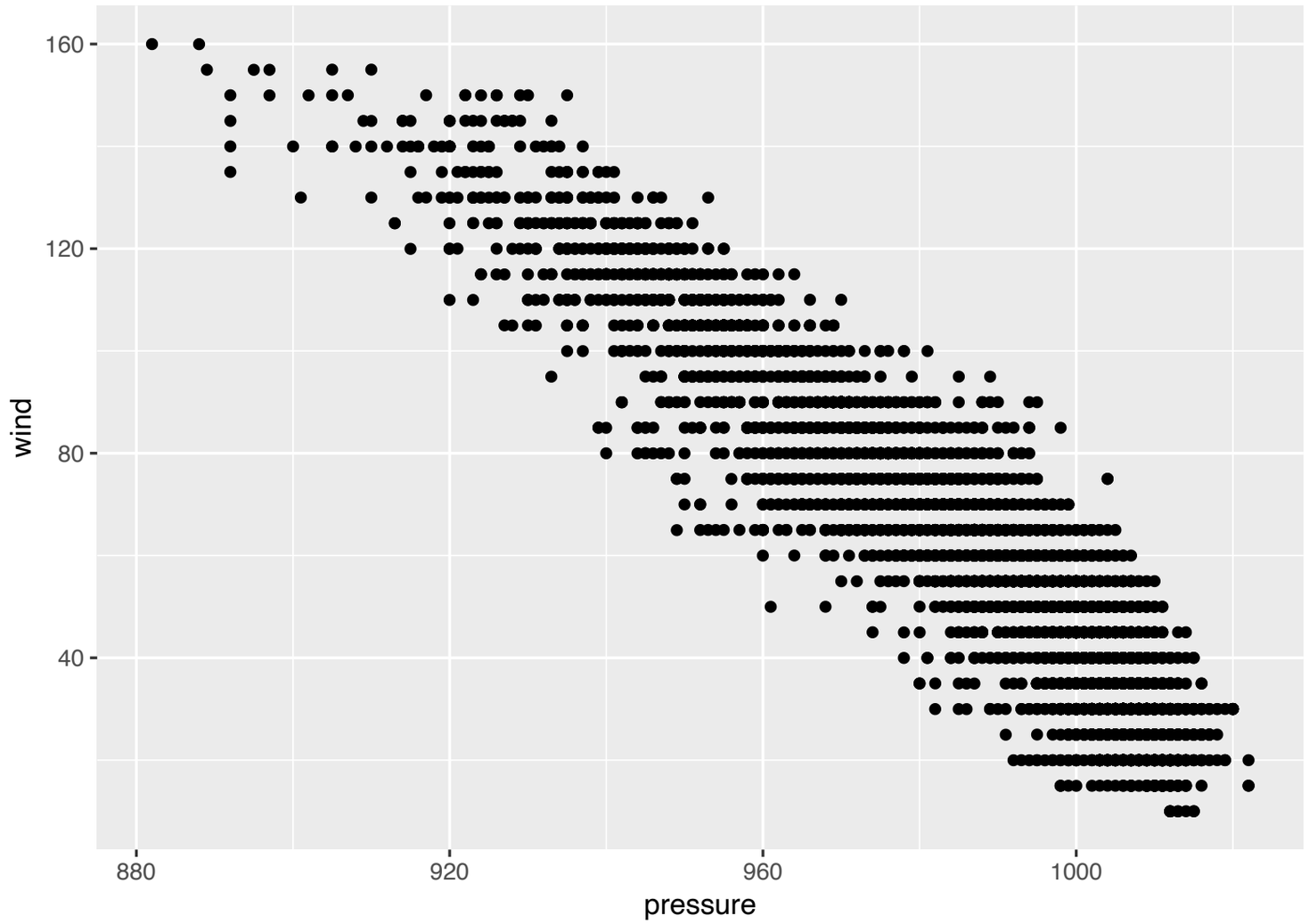
- Scatterplots are the first step for examining relationships between two numerical variables.

SCATTERPLOTS

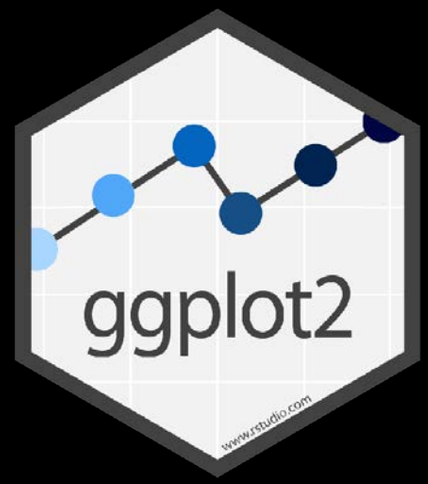


- We have already practiced making scatterplots.

```
ggplot(storms, aes(x = pressure, y = wind)) +
  geom_point()
```

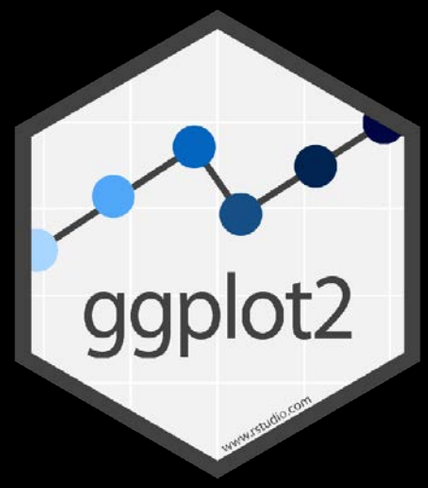


SCATTERPLOTS

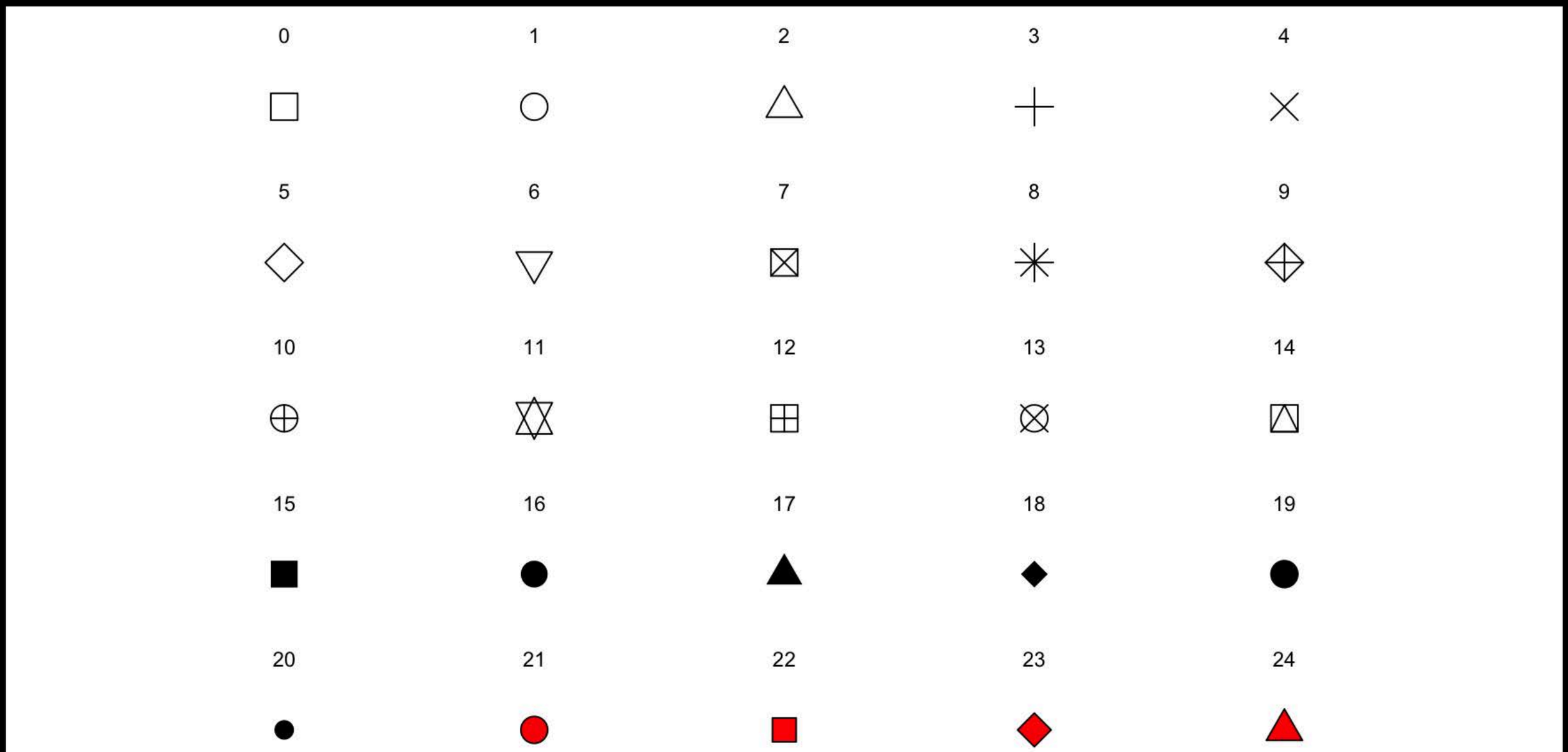


- Third *categorical* variable can be represented using color or shape.

SCATTERPLOTS

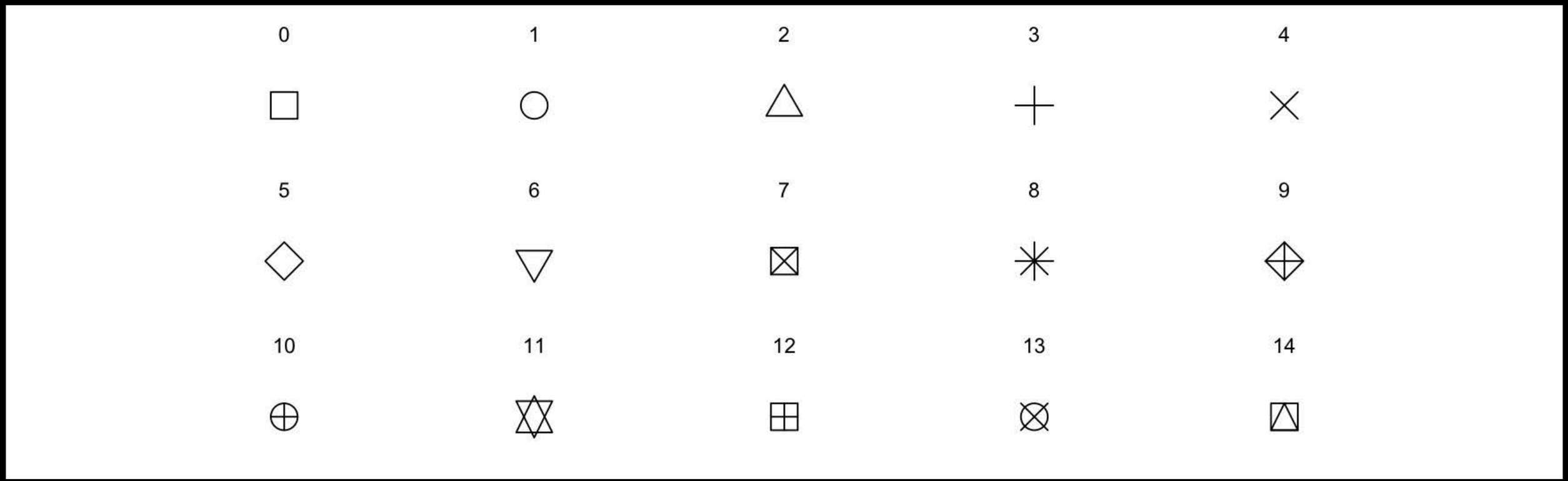


- The options for point shapes:



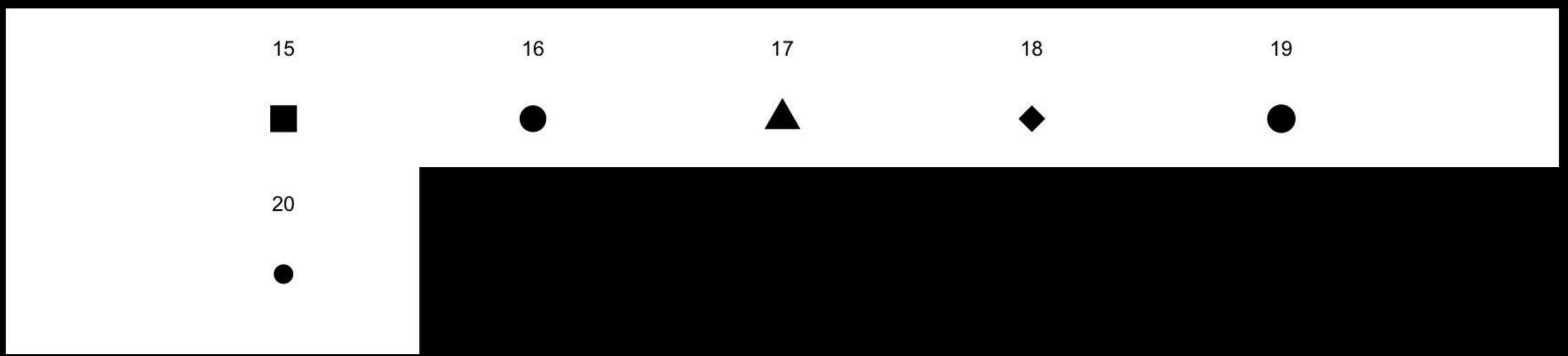
SCATTERPLOTS

- Hollow shapes (use 'color')

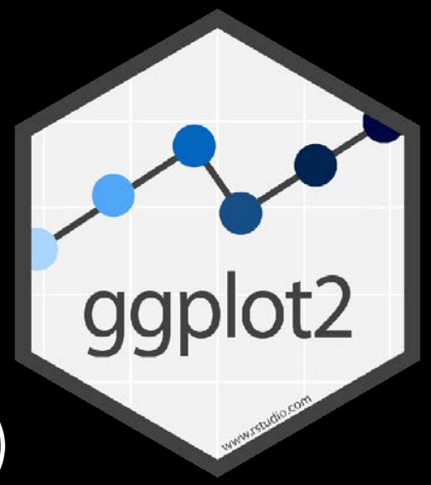


SCATTERPLOTS

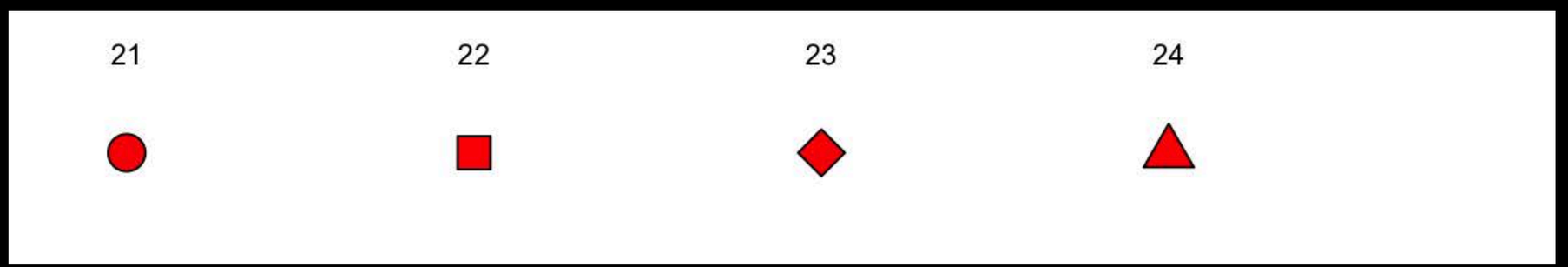
- Solid shapes (use 'color')



SCATTERPLOTS

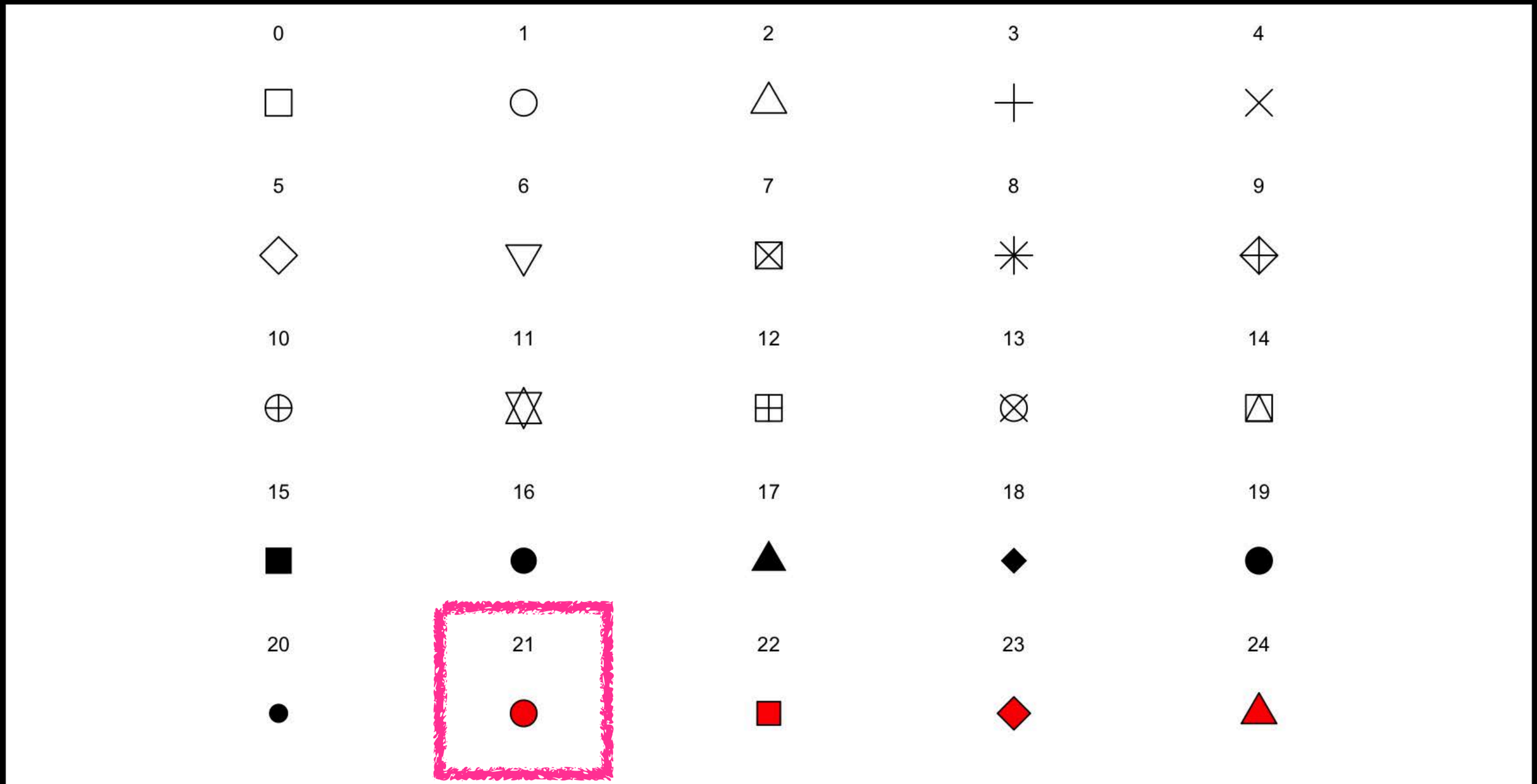


- Outlined shapes (use 'color' for outline, 'fill' for interior)
- The 'stroke' argument controls the thickness of the outline

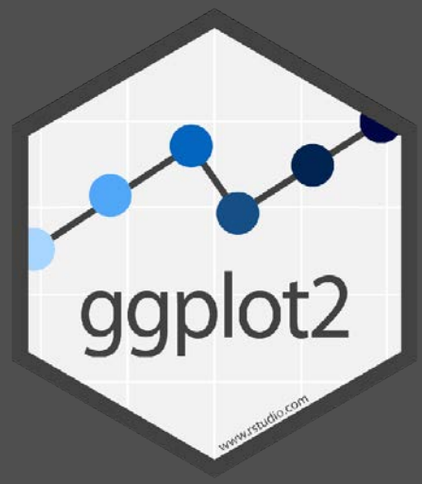


SCATTERPLOTS

- Shape 21 is really useful (it's the only one I remember)



YOUR TURN

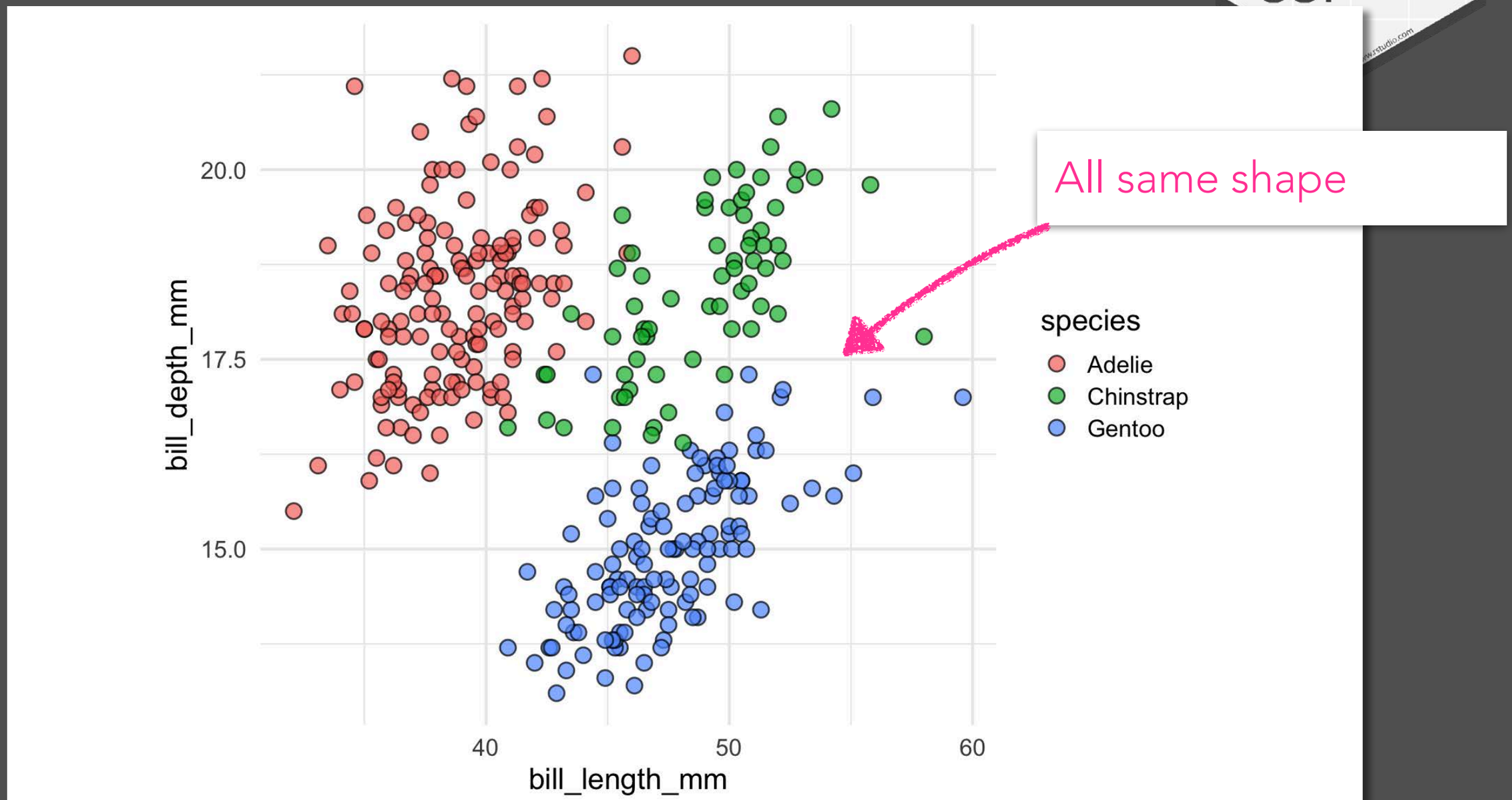


- Create a new folder for the week and a new R markdown file.

- Load the packages `tidyverse` and `palmerpenguins`

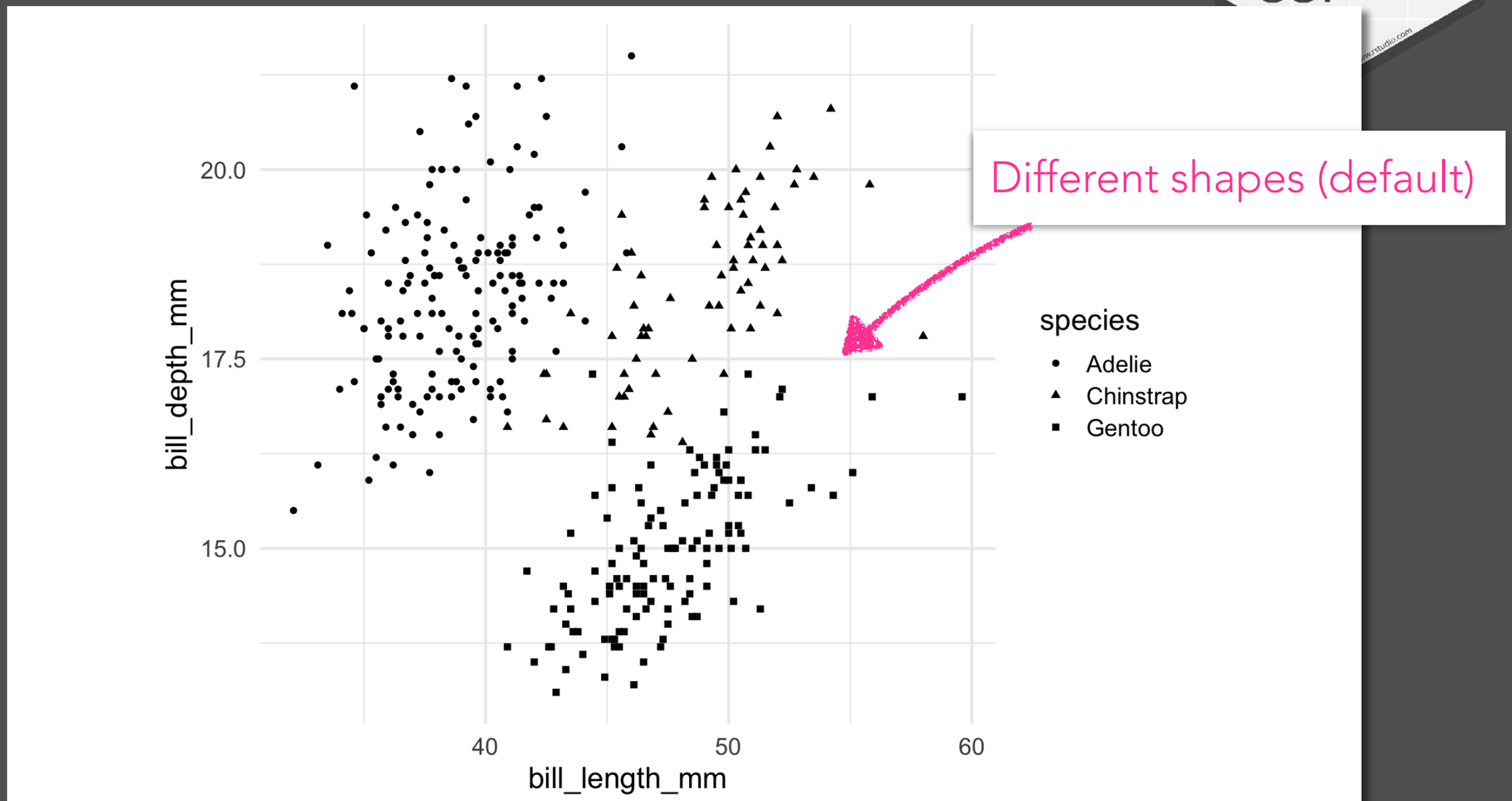
- Look at the `penguins` data set

YOUR TURN

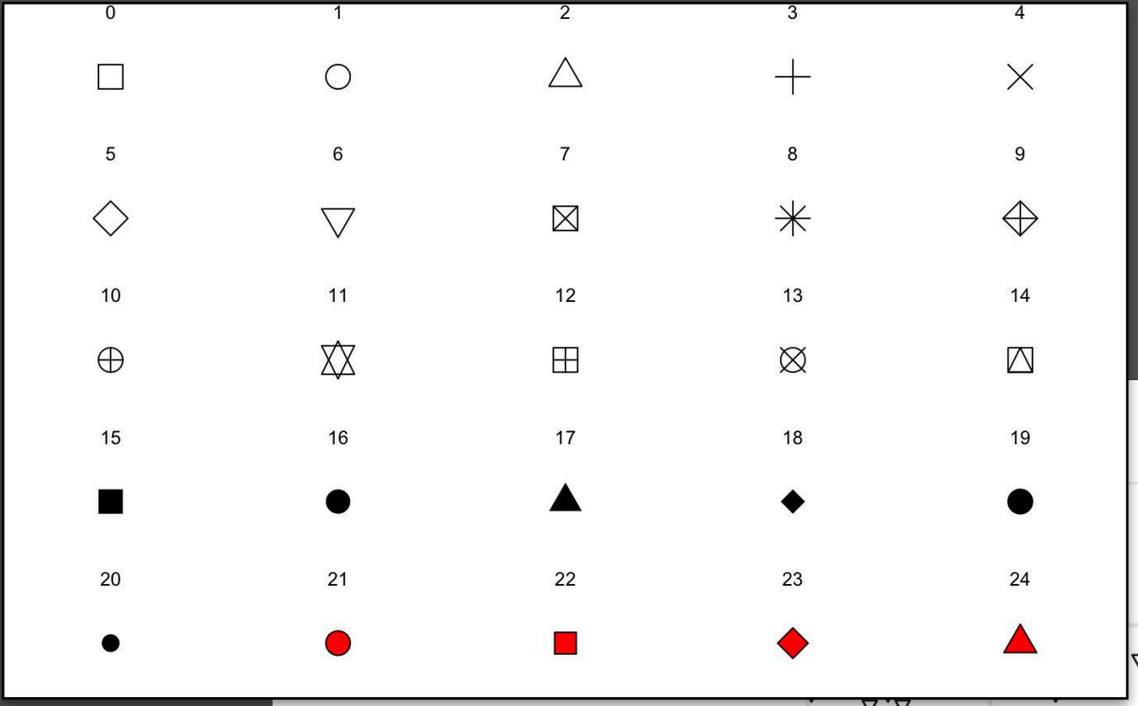


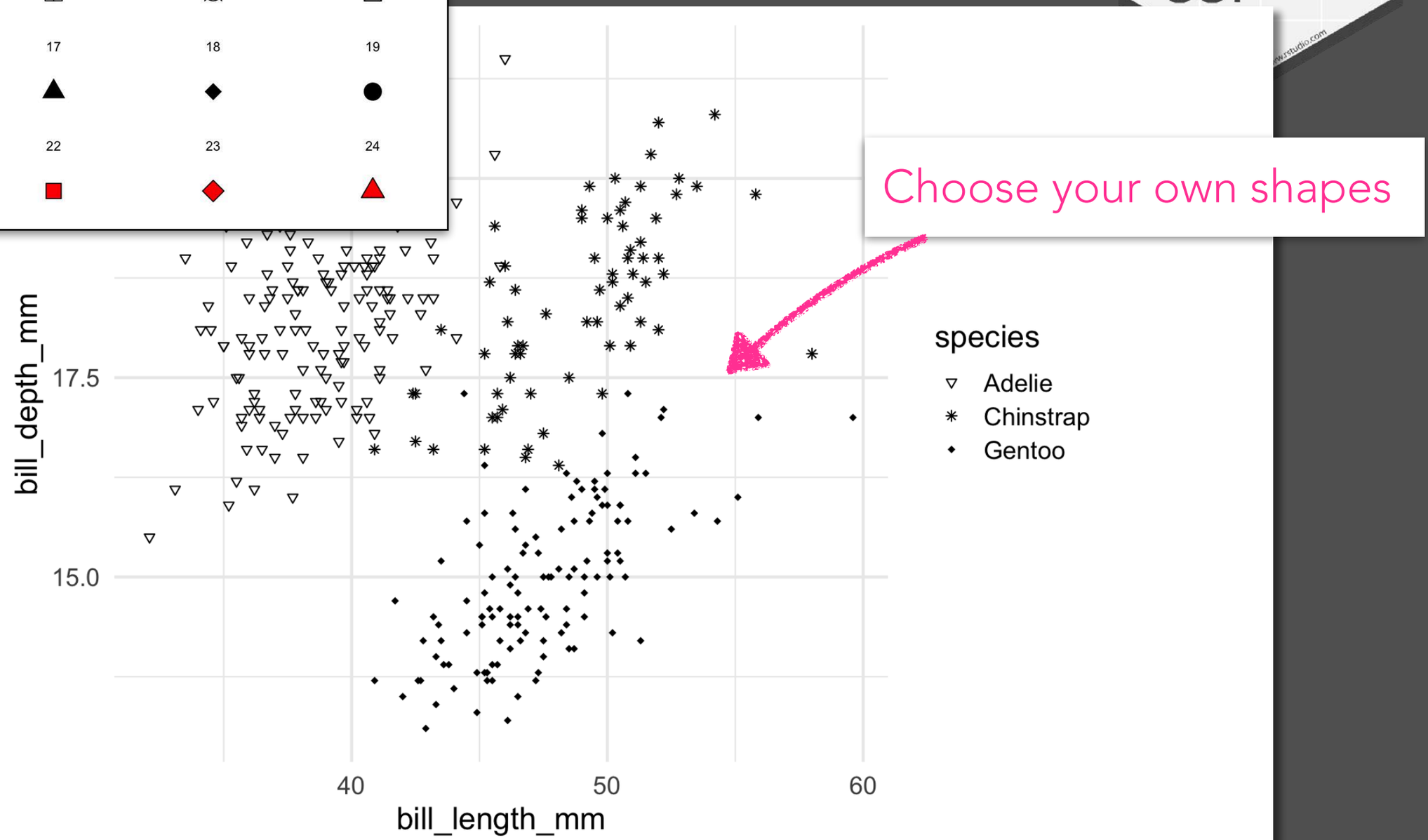

```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
                     fill = species)) +
  geom_point(color = "black", size = 3.5, stroke = 0.75,
             shape = 21, alpha = 0.7)
```

YOUR TURN

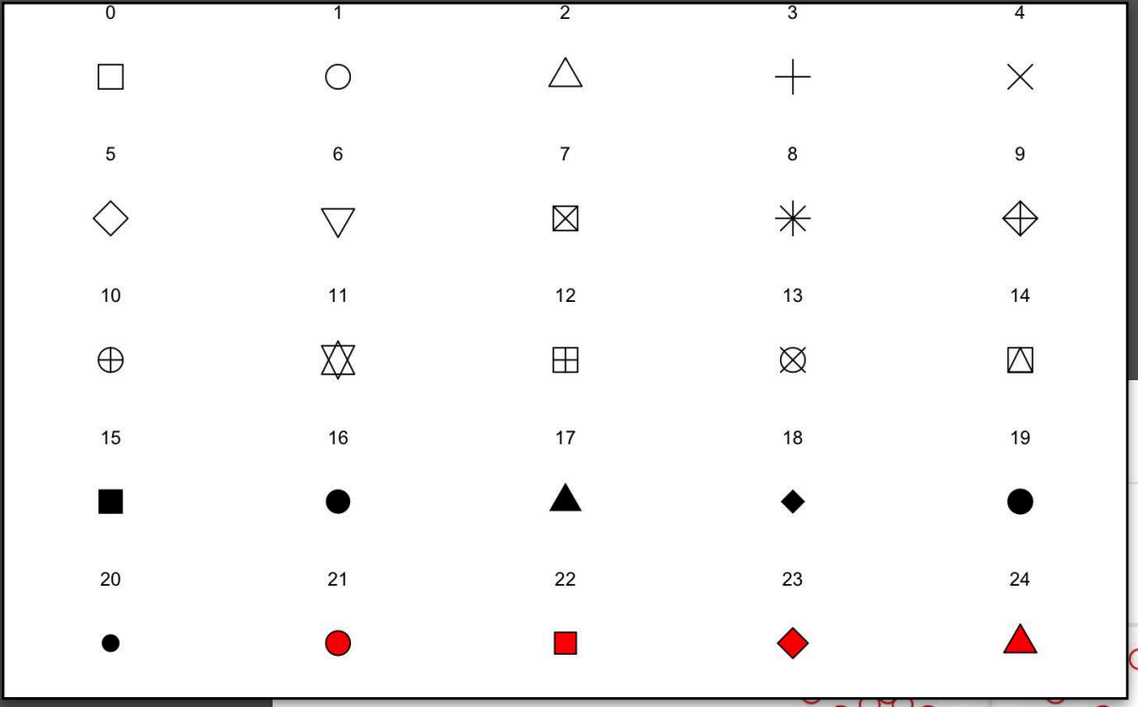


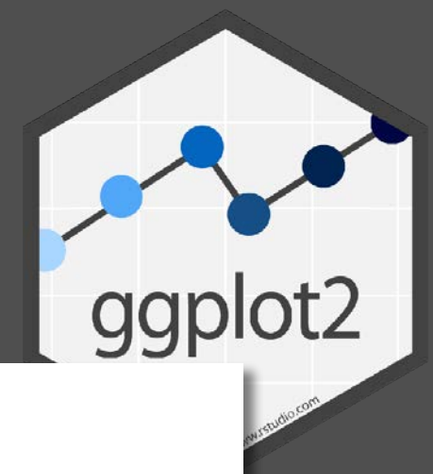

```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
                     shape = species)) +
  geom_point()
```



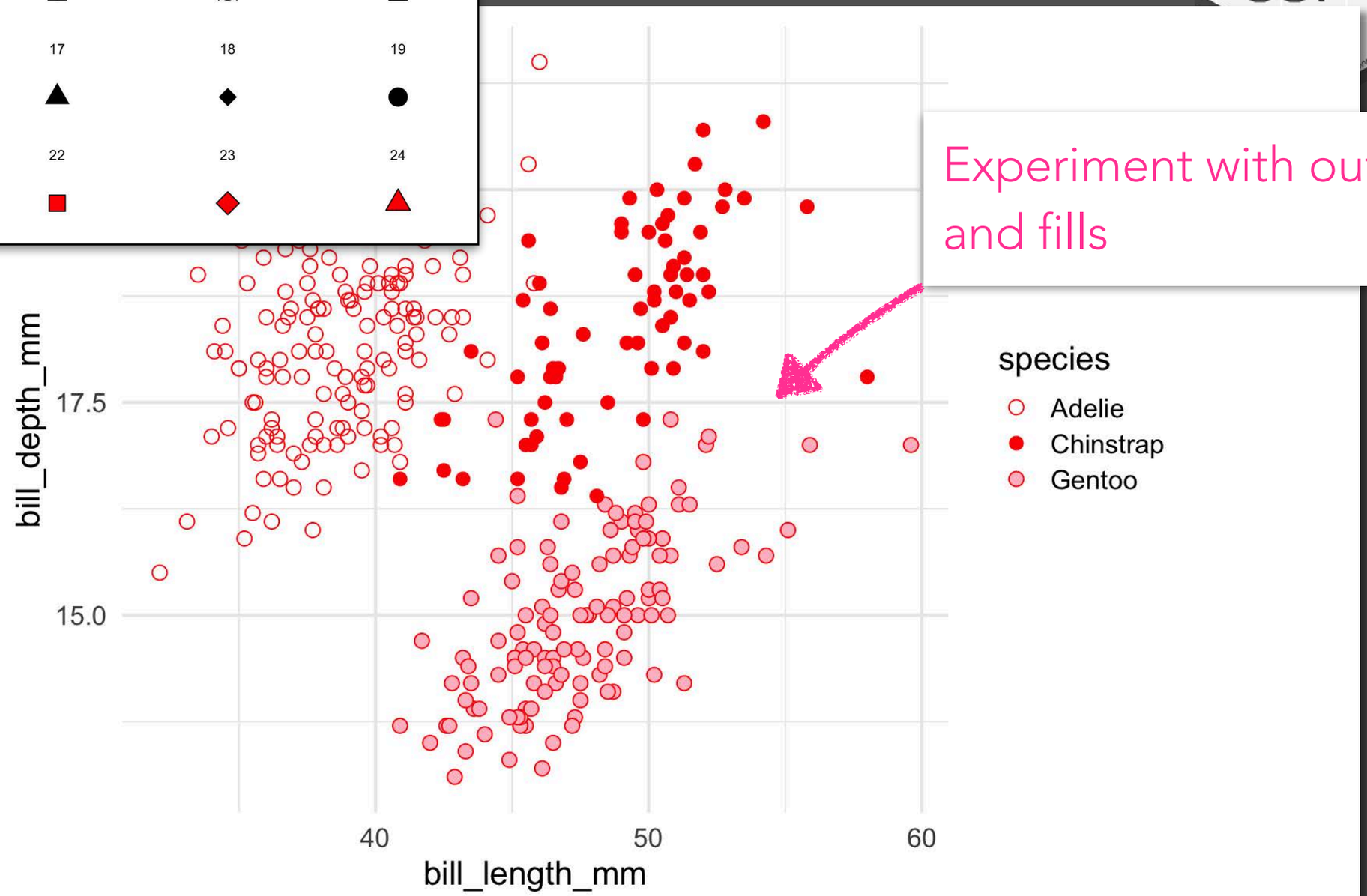



```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
                     shape = species)) +
  geom_point() +
  scale_shape_manual(values = c(6, 8, 18))
```



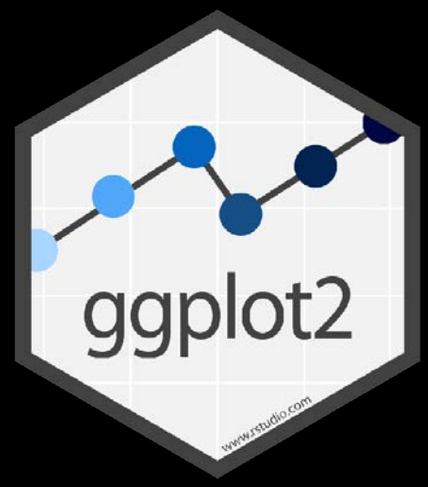


Experiment with outlines and fills




```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
                     shape = species)) +
  geom_point(color = "red", fill = "pink", size = 3) +
  scale_shape_manual(values = c(1, 16, 21))
```

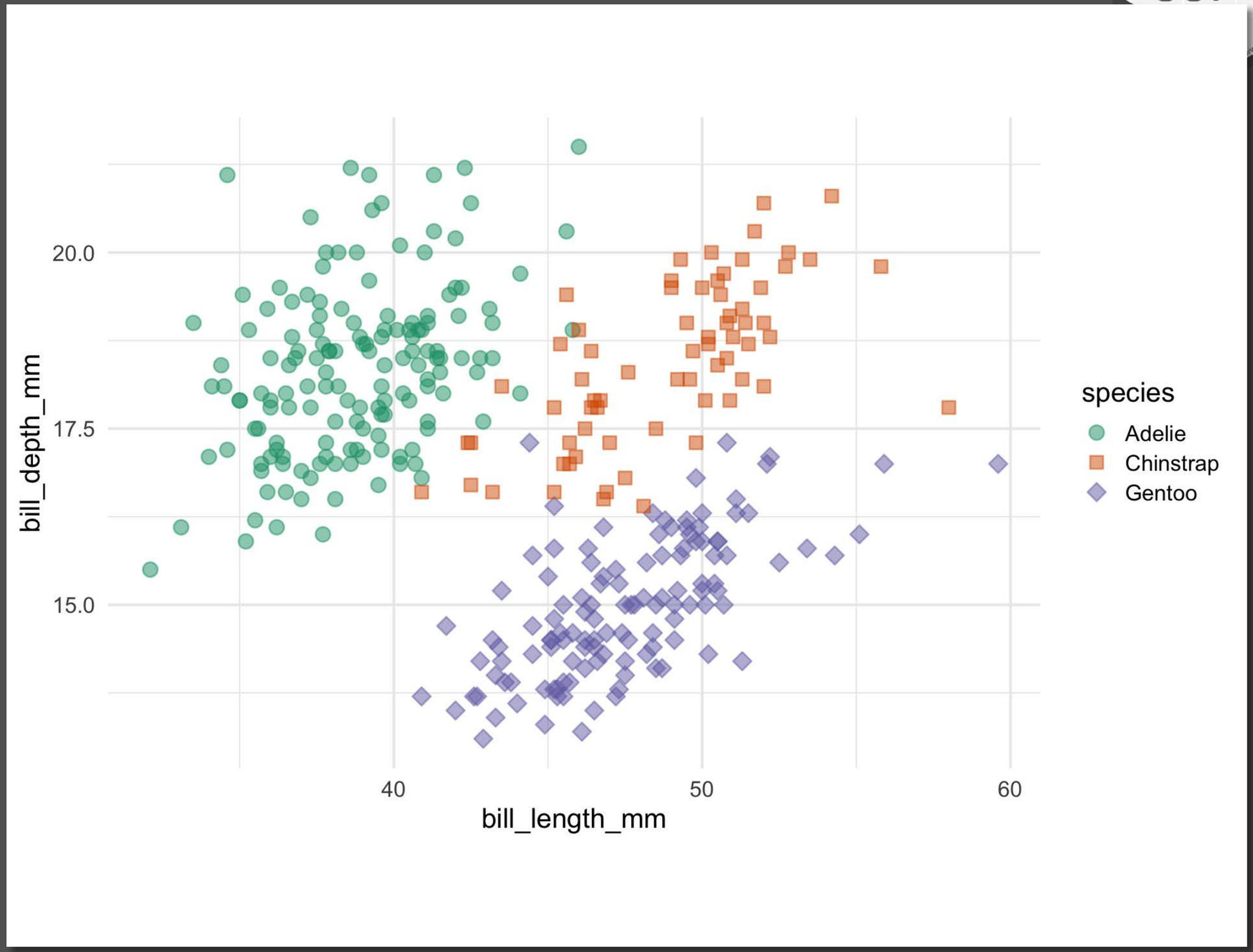
SCATTERPLOTS



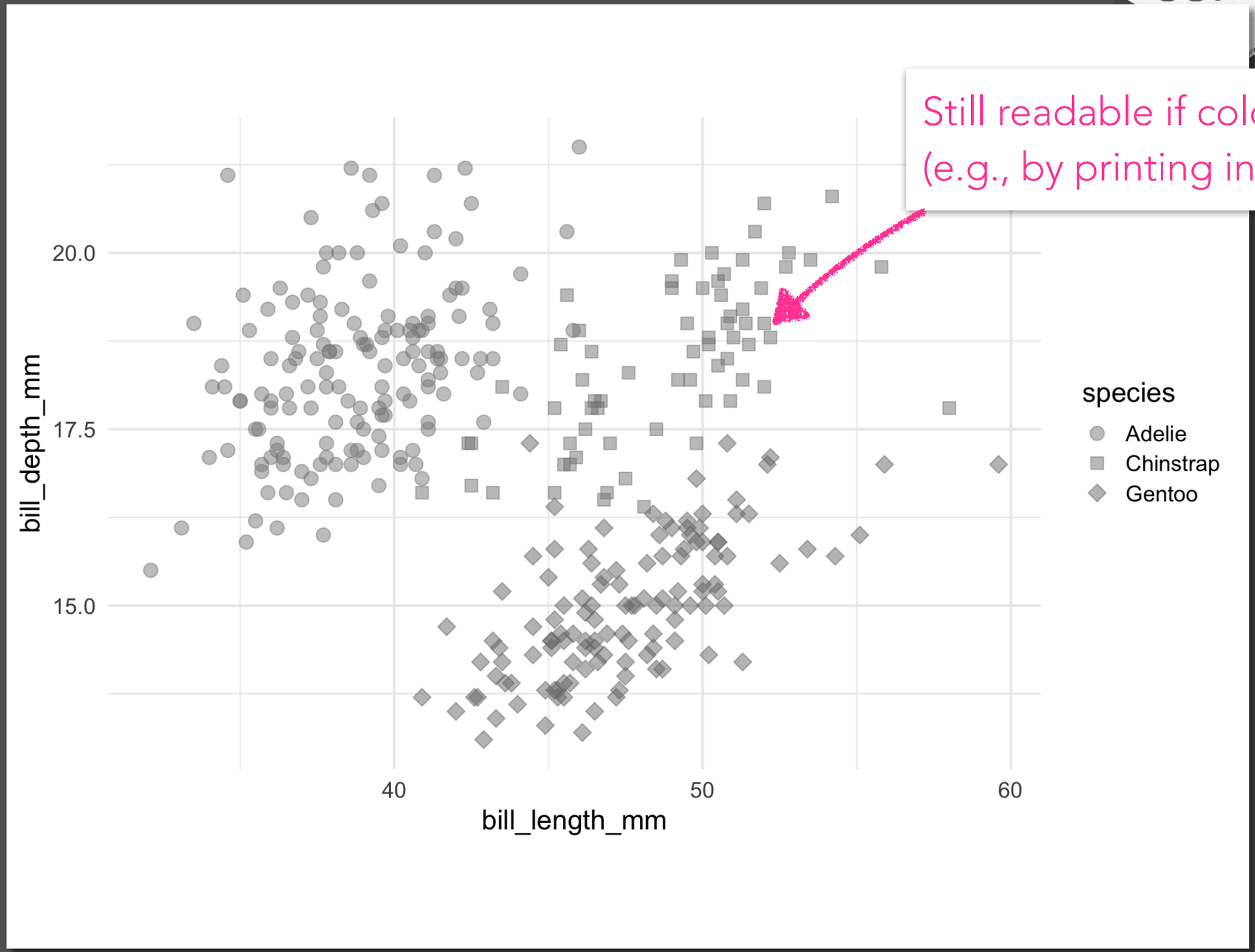
- Guidelines for shapes:

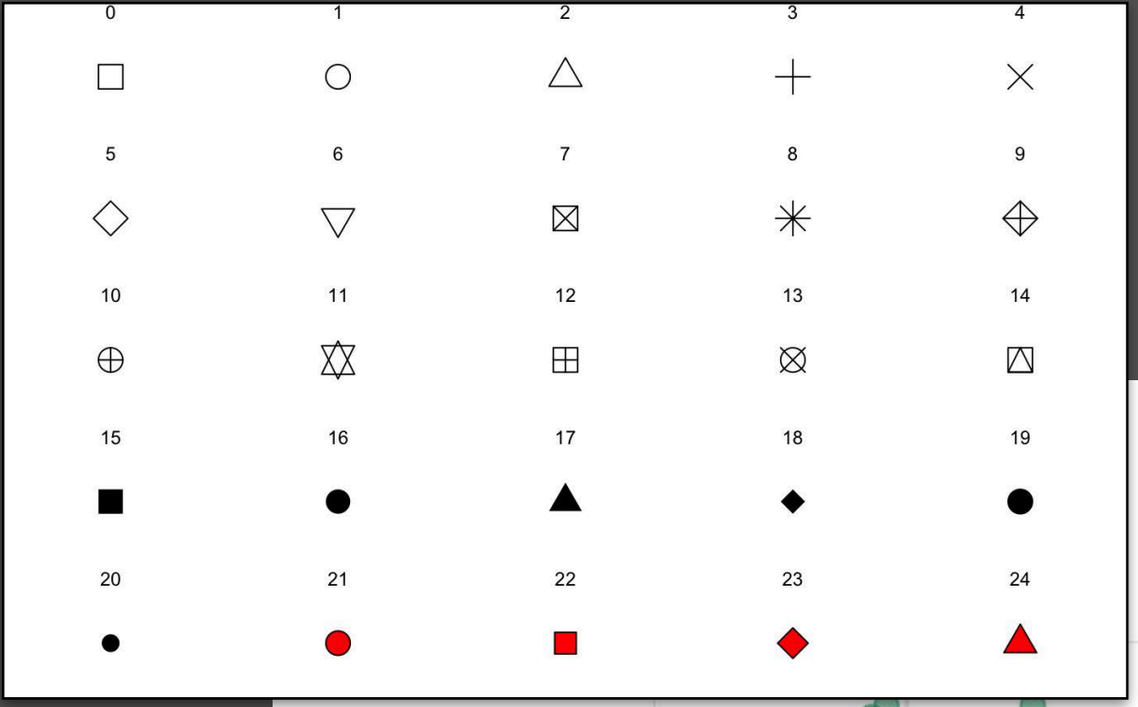
1. The main reason to use shapes is when colors can't be used for some reason.
2. Using shape alone to encode a variable is usually a bad decision—shapes are hard for people to distinguish.
3. If using different shapes, use no more than 6.
4. Redundant encoding can be useful (e.g., map both color and shape to the same variable)

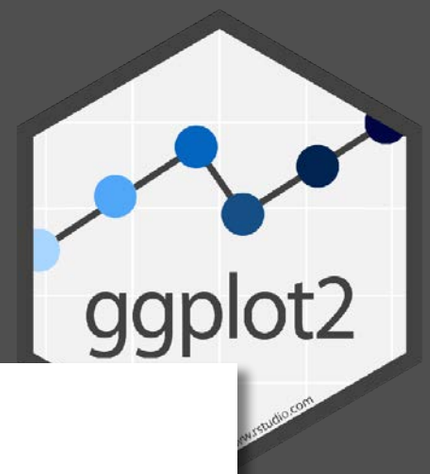
YOUR TURN

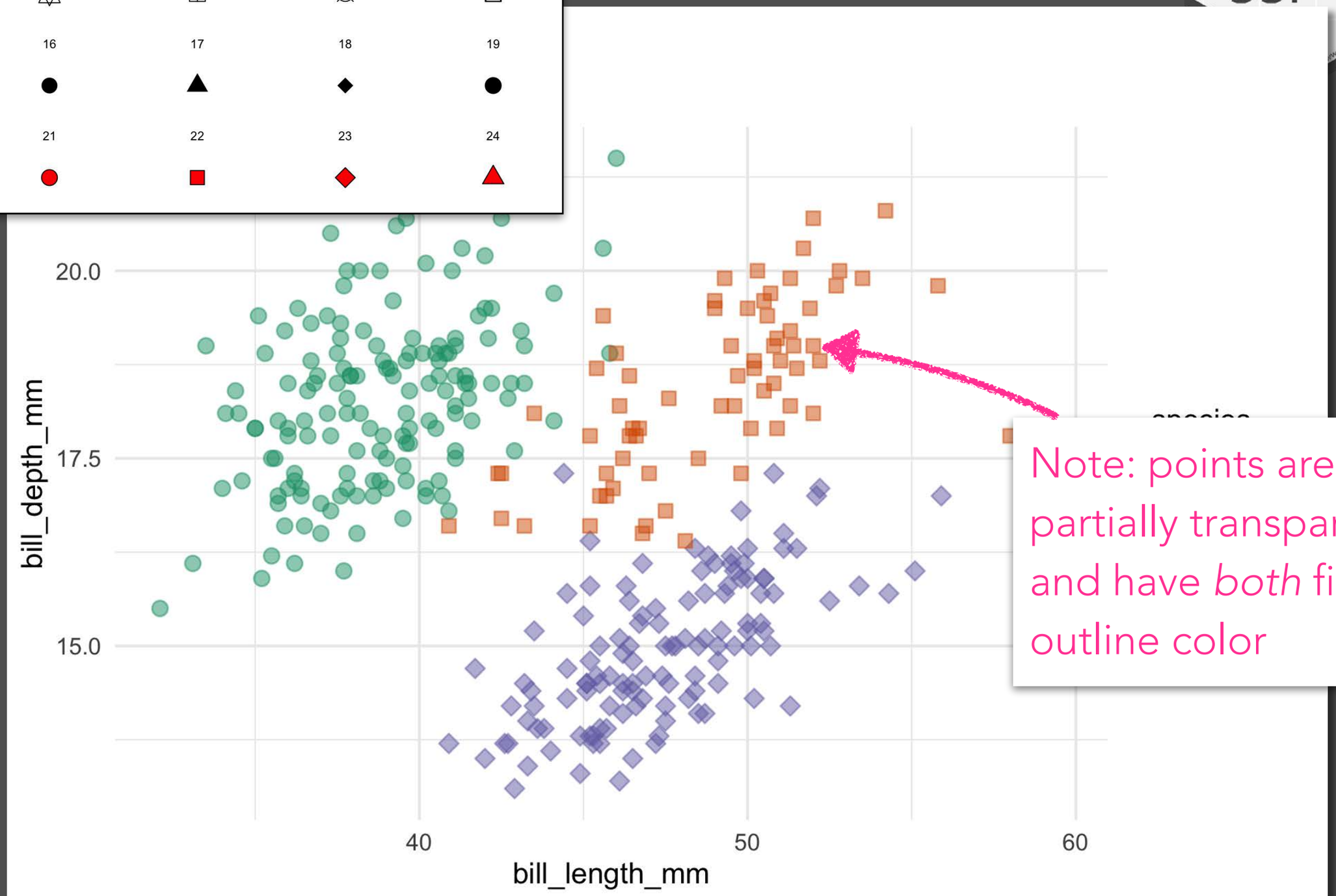

20.0
17.5
15.0
bill_depth_mm
40
50
60
bill_length_mm
species
Adelie
Chinstrap
Gentoo

YOUR TURN


Still readable if color is lost
(e.g., by printing in B&W)
20.0
17.5
15.0
bill_depth_mm
40
50
60
bill_length_mm
species
Adelie
Chinstrap
Gentoo

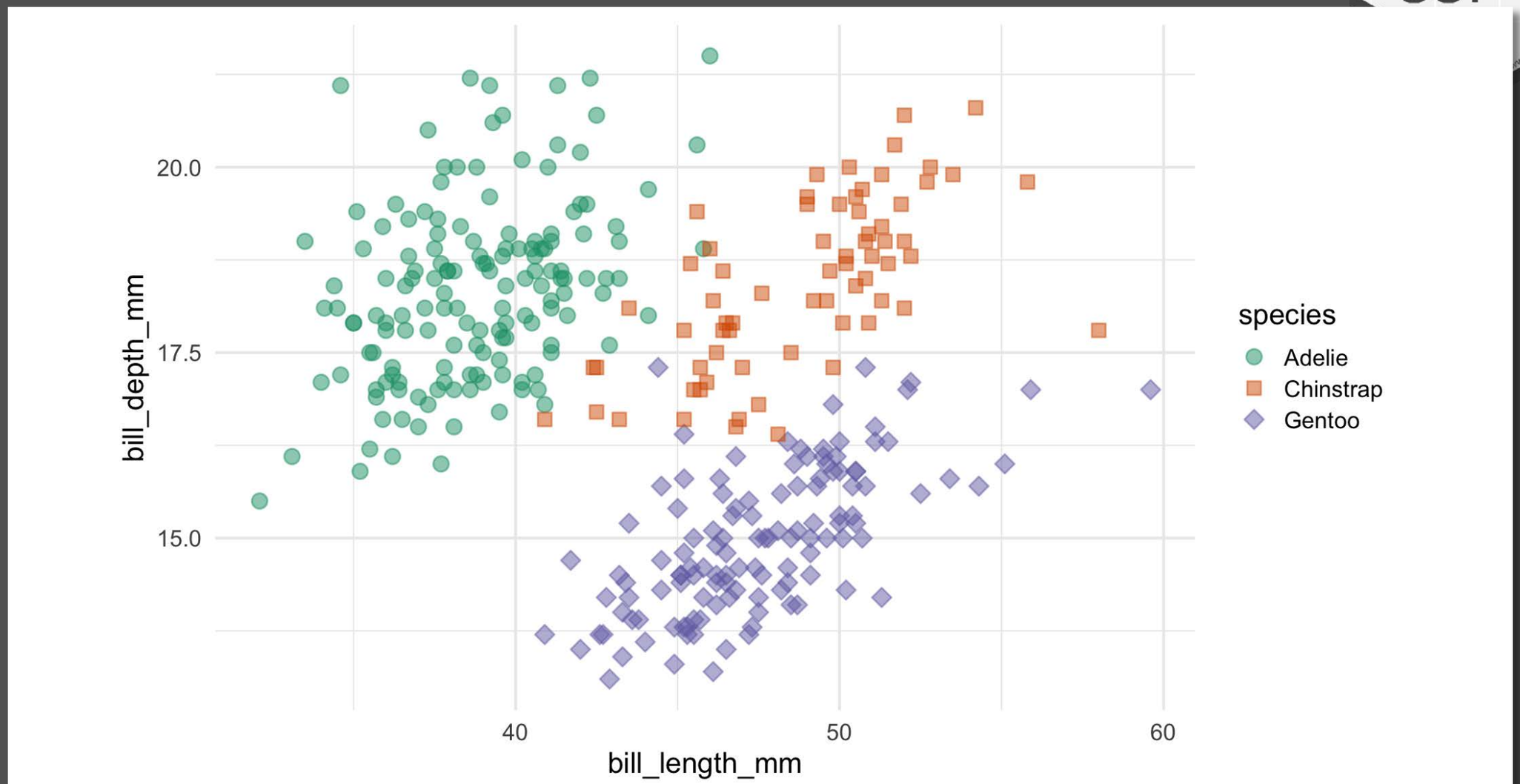






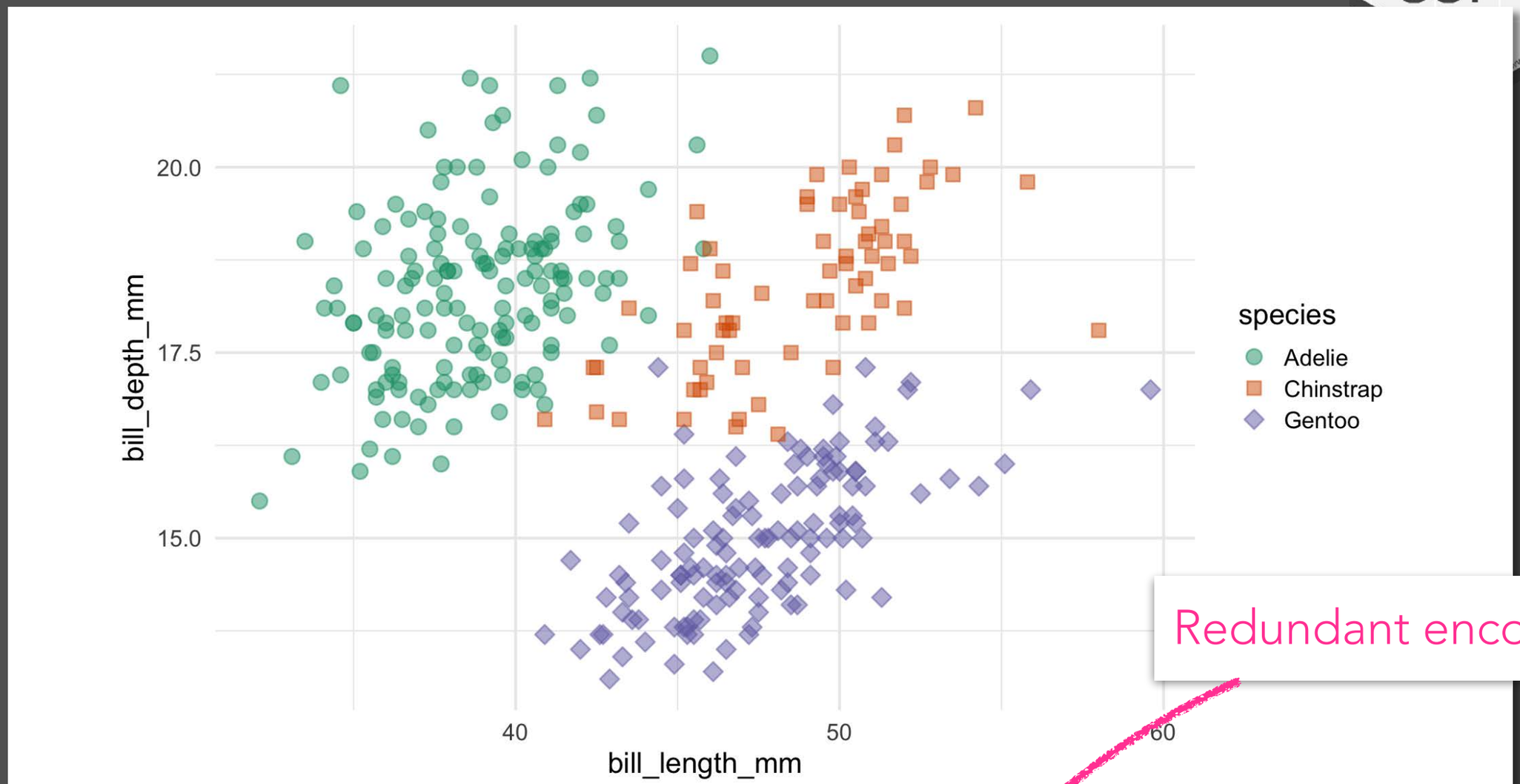
Note: points are partially transparent and have *both* fill and outline color

YOUR TURN




```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
       fill = species, shape = species, color = species)) +
  geom_point(size = 3.5, alpha = 0.5) +
  scale_fill_brewer(palette = "Dark2") +
  scale_color_brewer(palette = "Dark2") +
  scale_shape_manual(values = c(21, 22, 23))
```

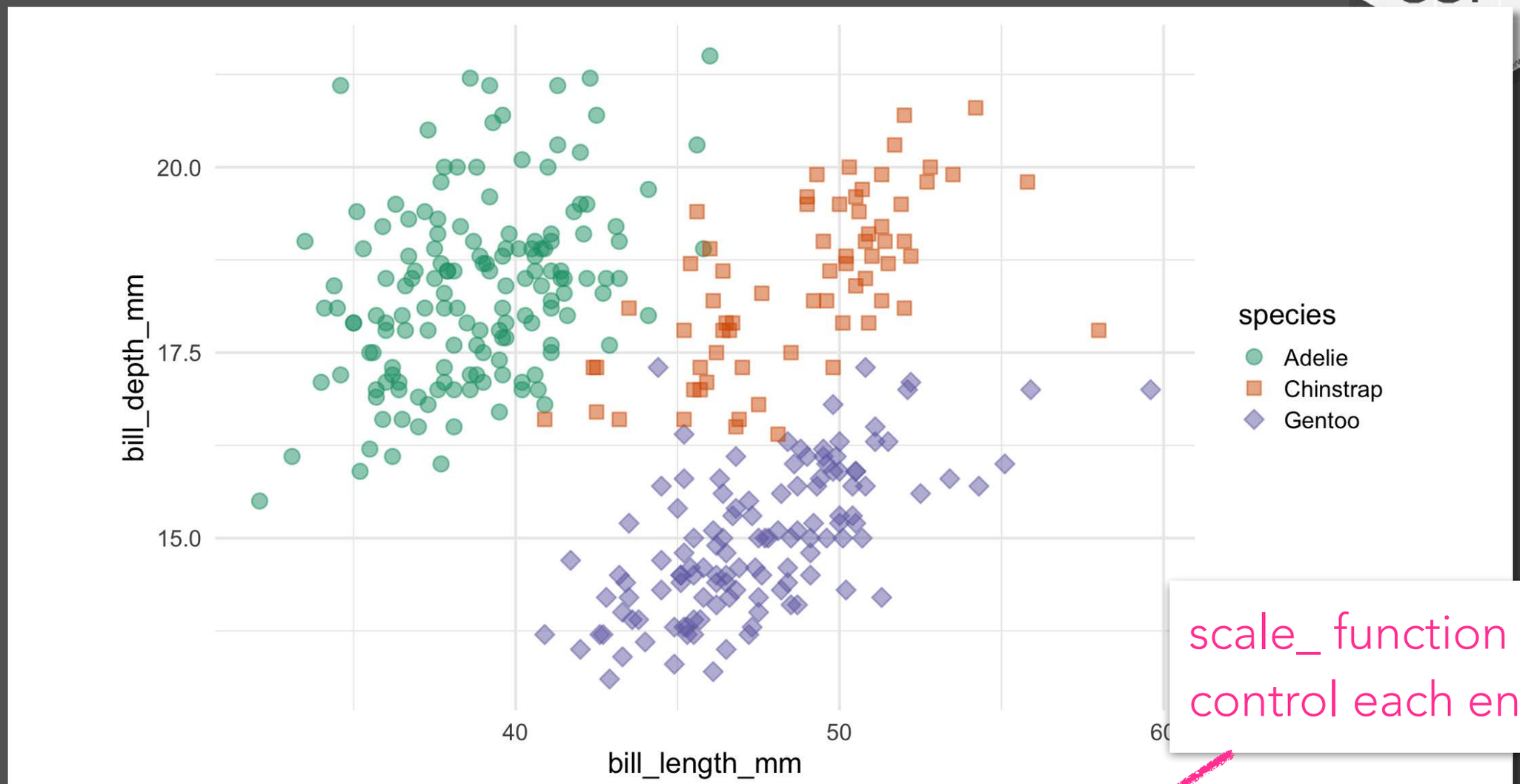
YOUR TURN



Redundant encodings

```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
       fill = species, shape = species, color = species)) +
  geom_point(size = 3.5, alpha = 0.5) +
  scale_fill_brewer(palette = "Dark2") +
  scale_color_brewer(palette = "Dark2") +
  scale_shape_manual(values = c(21, 22, 23))
```

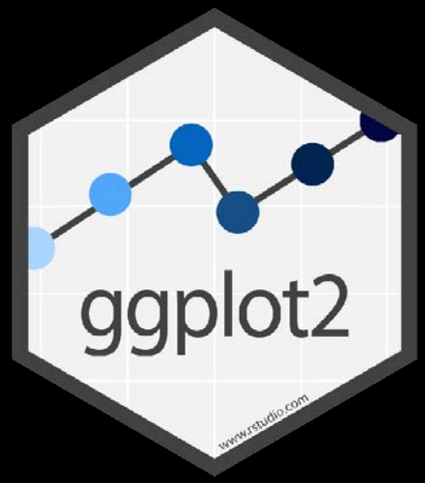
YOUR TURN



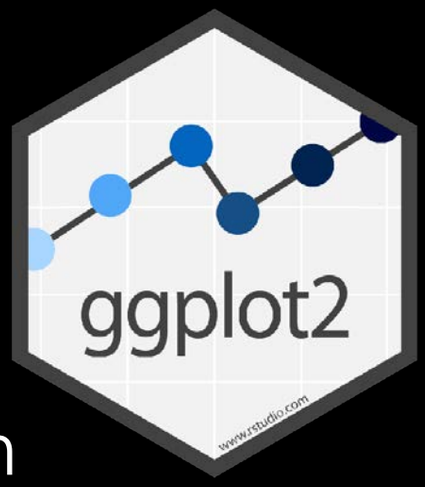
scale_ function to control each encoding

```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
       fill = species, shape = species, color = species)) +
  geom_point(size = 3.5, alpha = 0.5) +
  scale_fill_brewer(palette = "Dark2") +
  scale_color_brewer(palette = "Dark2") +
  scale_shape_manual(values = c(21, 22, 23))
```

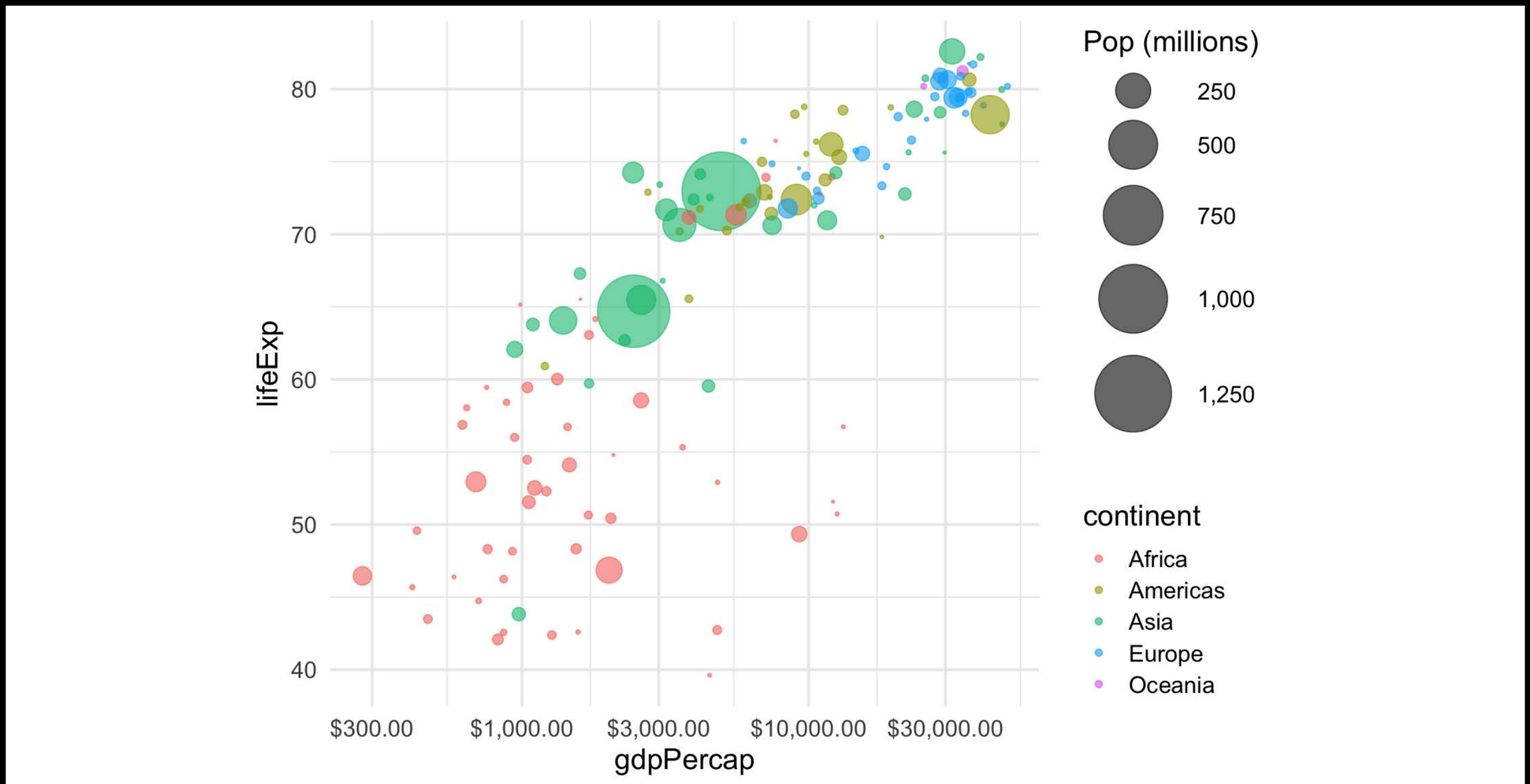
BUBBLE CHARTS



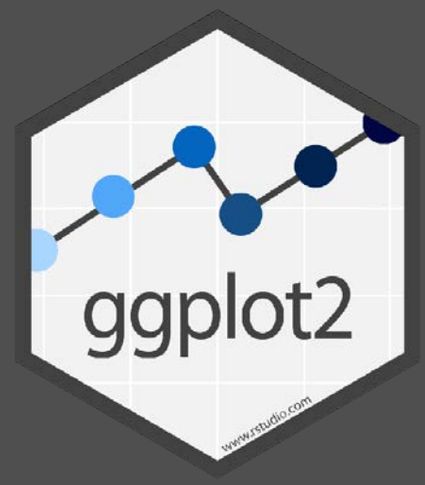
BUBBLE CHARTS



- Third (and fourth) *numerical* variable can be shown by color or size.



YOUR TURN



- Create a new folder for the week and a new R markdown file.

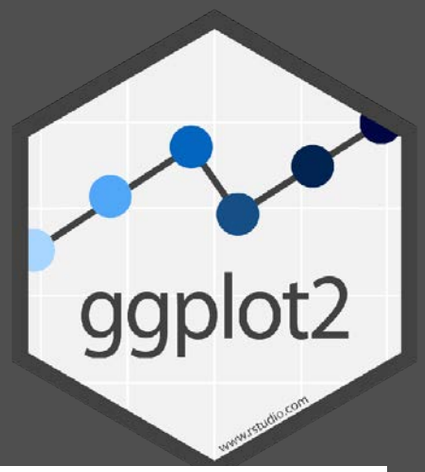
- Create a new R chunk and load the tidyverse package.

- Install the packages `gapminder` and `scales`

- Load the packages `tidyverse`, `gapminder`, and `scales`

- Look at the gapminder data set

YOUR TURN

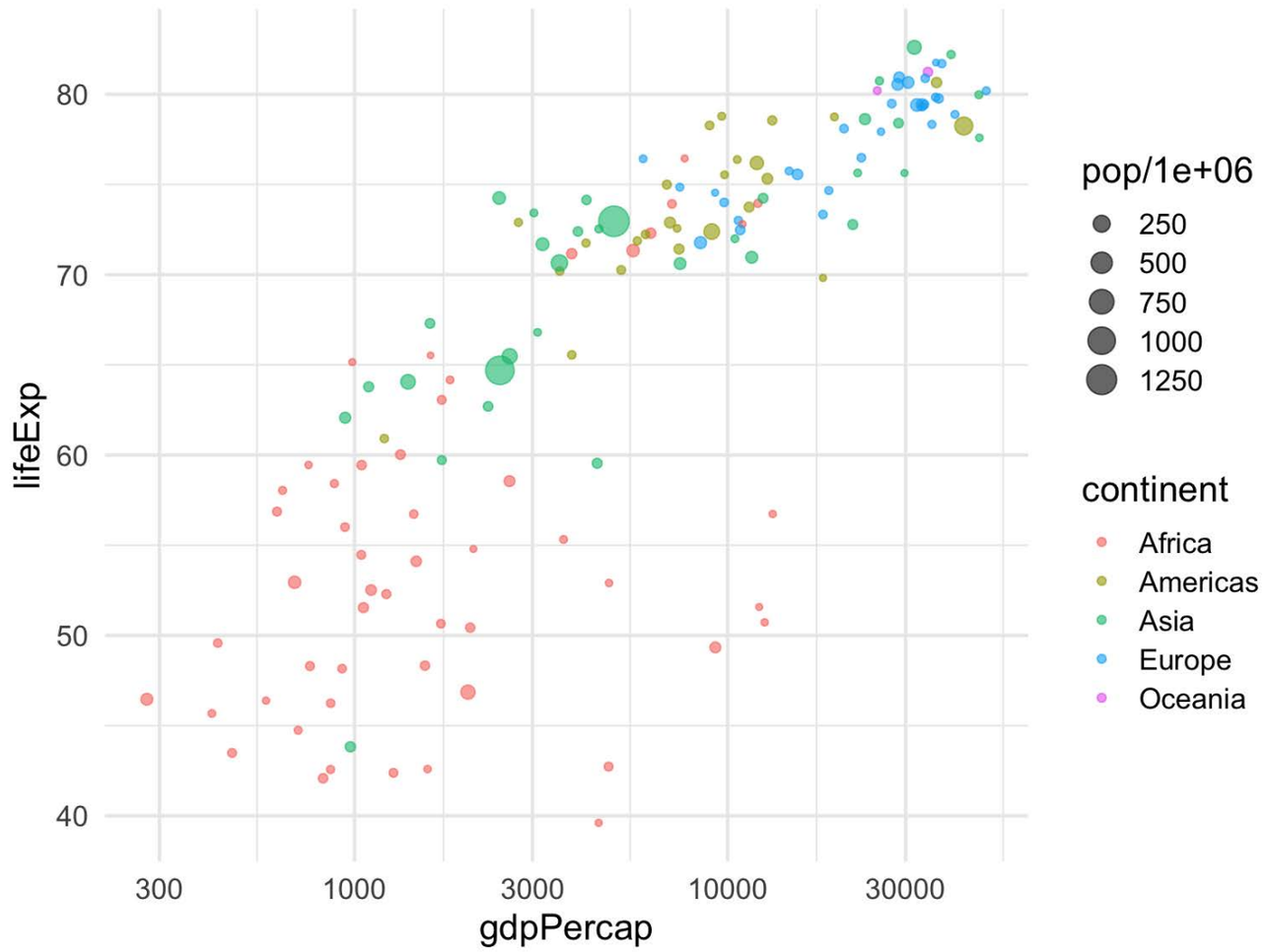


country	continent	year	lifeExp	pop	gdpPercap
Afghanistan	Asia	1952	28.801	8425333	779.4453
Afghanistan	Asia	1957	30.332	9240934	820.8530
Afghanistan	Asia	1962	31.997	10267083	853.1007
Afghanistan	Asia	1967	34.020	11537966	836.1971
Afghanistan	Asia	1972	36.088	13079460	739.9811
Afghanistan	Asia	1977	38.438	14880372	786.1134
Afghanistan	Asia	1982	39.854	12881816	978.0114
Afghanistan	Asia	1987	40.822	13867957	852.3959
Afghanistan	Asia	1992	41.674	16317921	649.3414
Afghanistan	Asia	1997	41.763	22227415	635.3414

- Plot data lifeExp (y) vs. gdpPerCap (x) for 2007 only
- Divide pop by 1 million and encode as point size; color points by continent
- Use log 10 for x axis

YOUR TURN

```
ggplot(filter(gapminder, year == 2007),
       aes(x = gdpPercap, y = lifeExp,
           size = pop / 1e6, color = continent)) +
  geom_point(alpha = 0.6) +
  scale_x_log10()
```



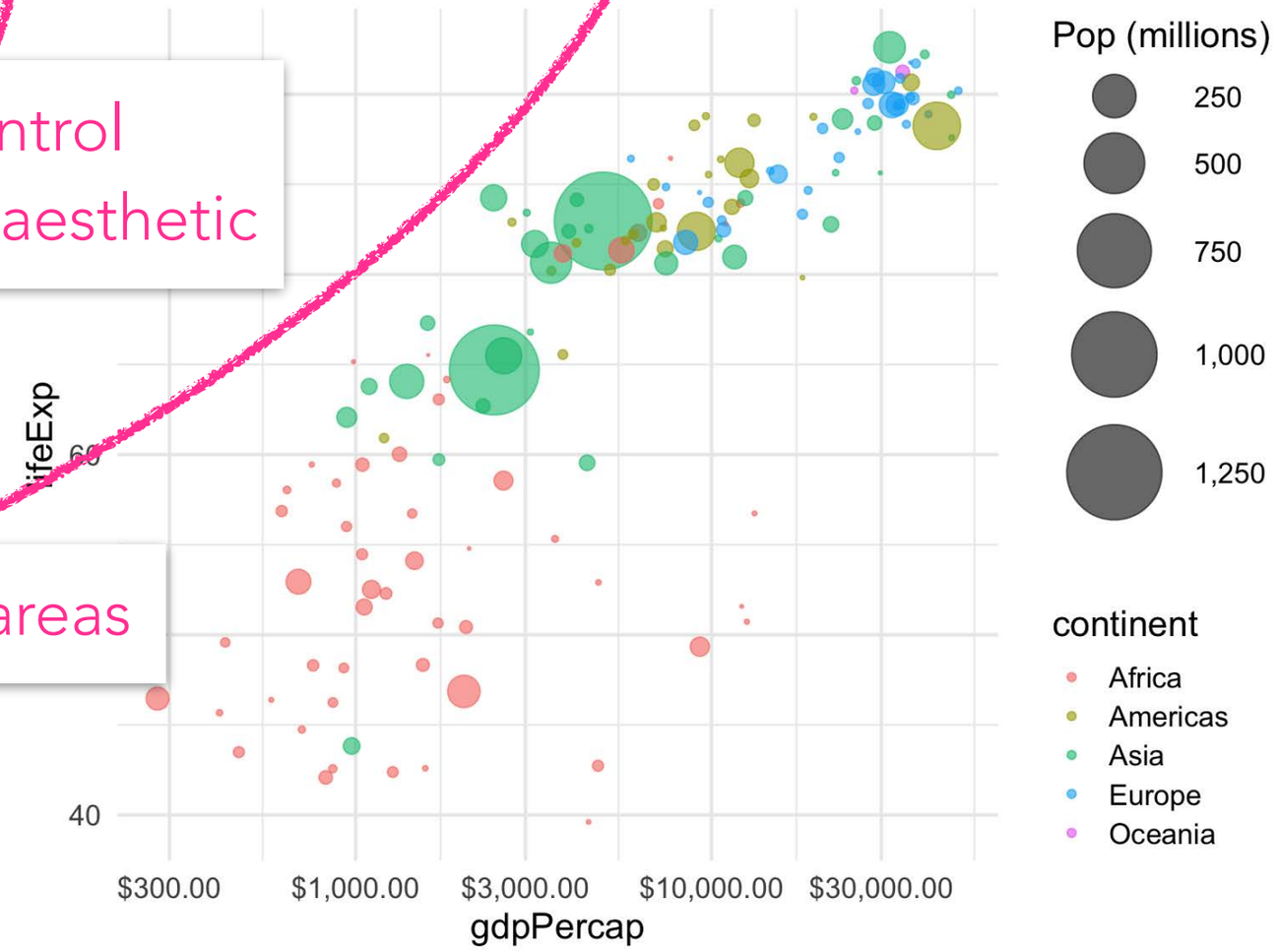
YOUR TURN

Convenient labelling functions provided by scales package

```
ggplot(filter(gapminder, year == 2007),
       aes(x = gdpPercap, y = lifeExp,
           size = pop / 1e6, color = continent)) +
  geom_point(alpha = 0.6) +
  scale_x_log10(labels = label_dollar()) +
  scale_size_area(max_size = 20, labels = label_comma(),
                  name = "Pop (millions)")
```

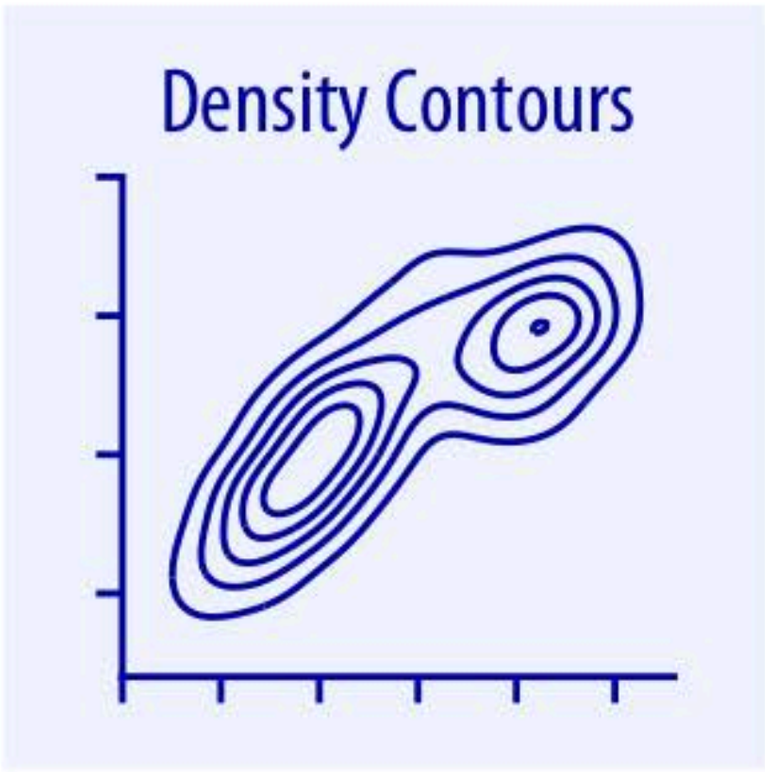
Scale function to control appearance of area aesthetic

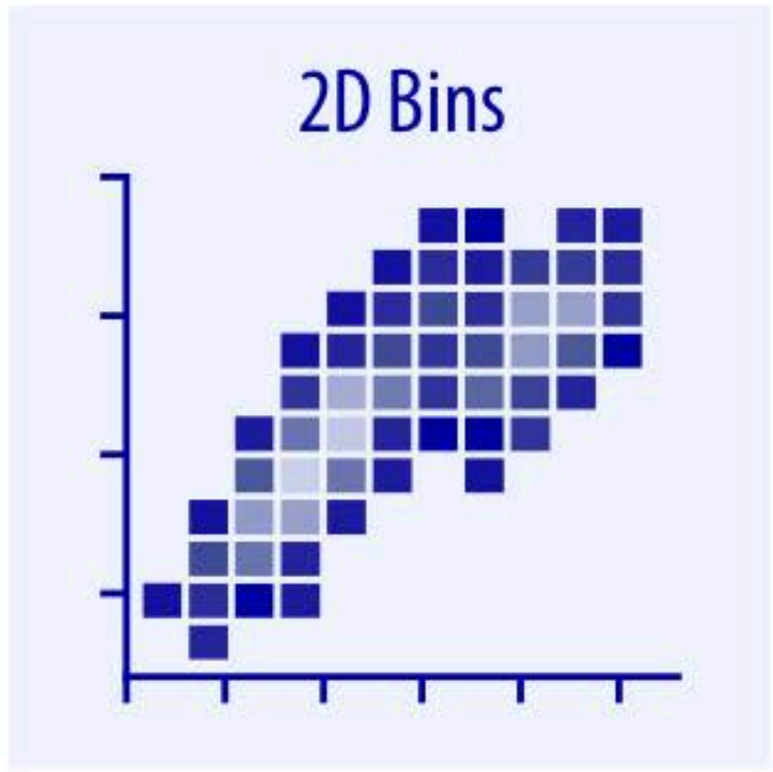
Larger max size for areas

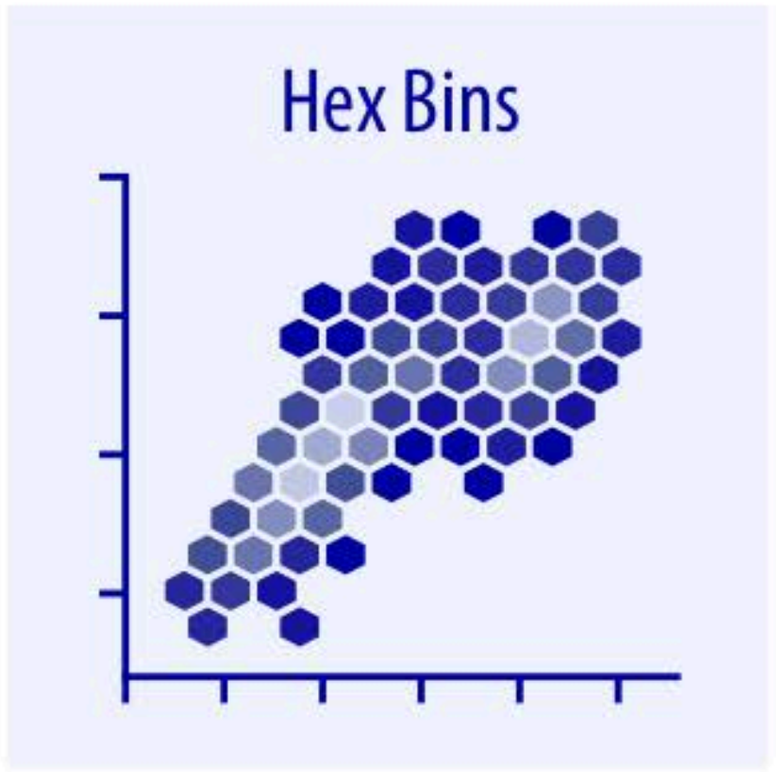


WHAT IF YOU HAVE TOO MANY OVERLAPPING POINTS TO DISPLAY?

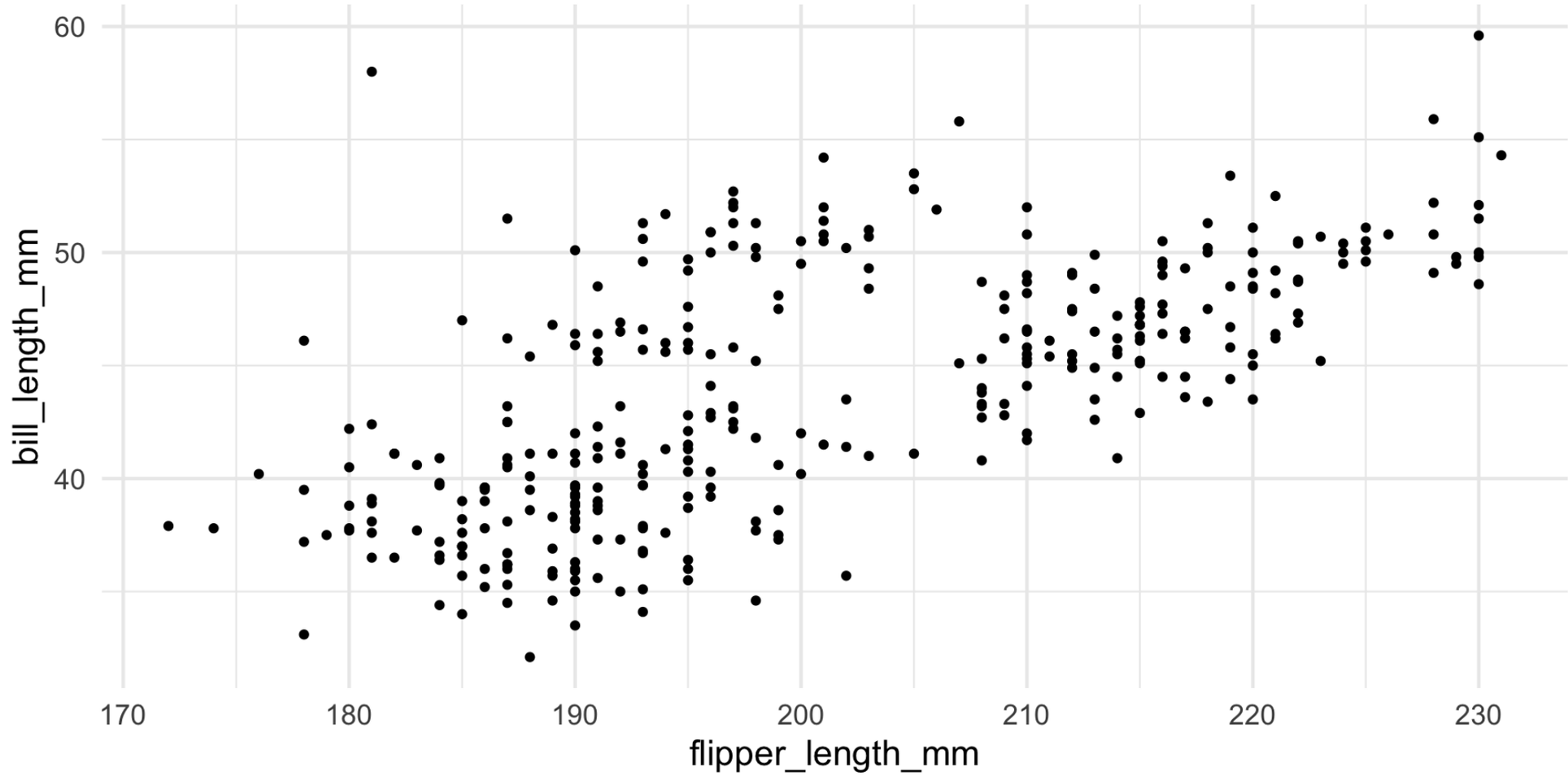
WHAT IF YOU HAVE TOO MANY OVERLAPPING POINTS TO DISPLAY?





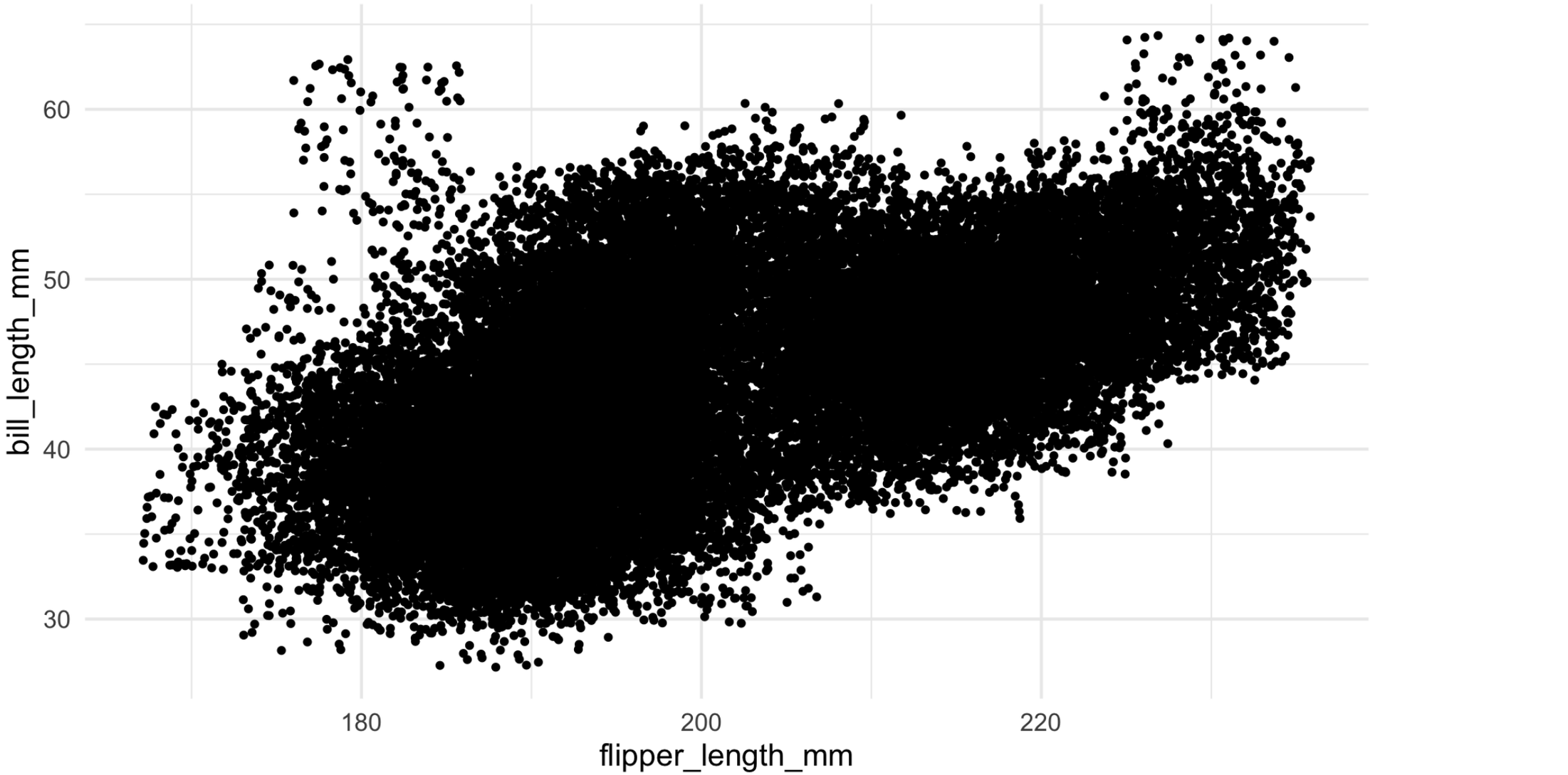


REASONABLE # OF POINTS



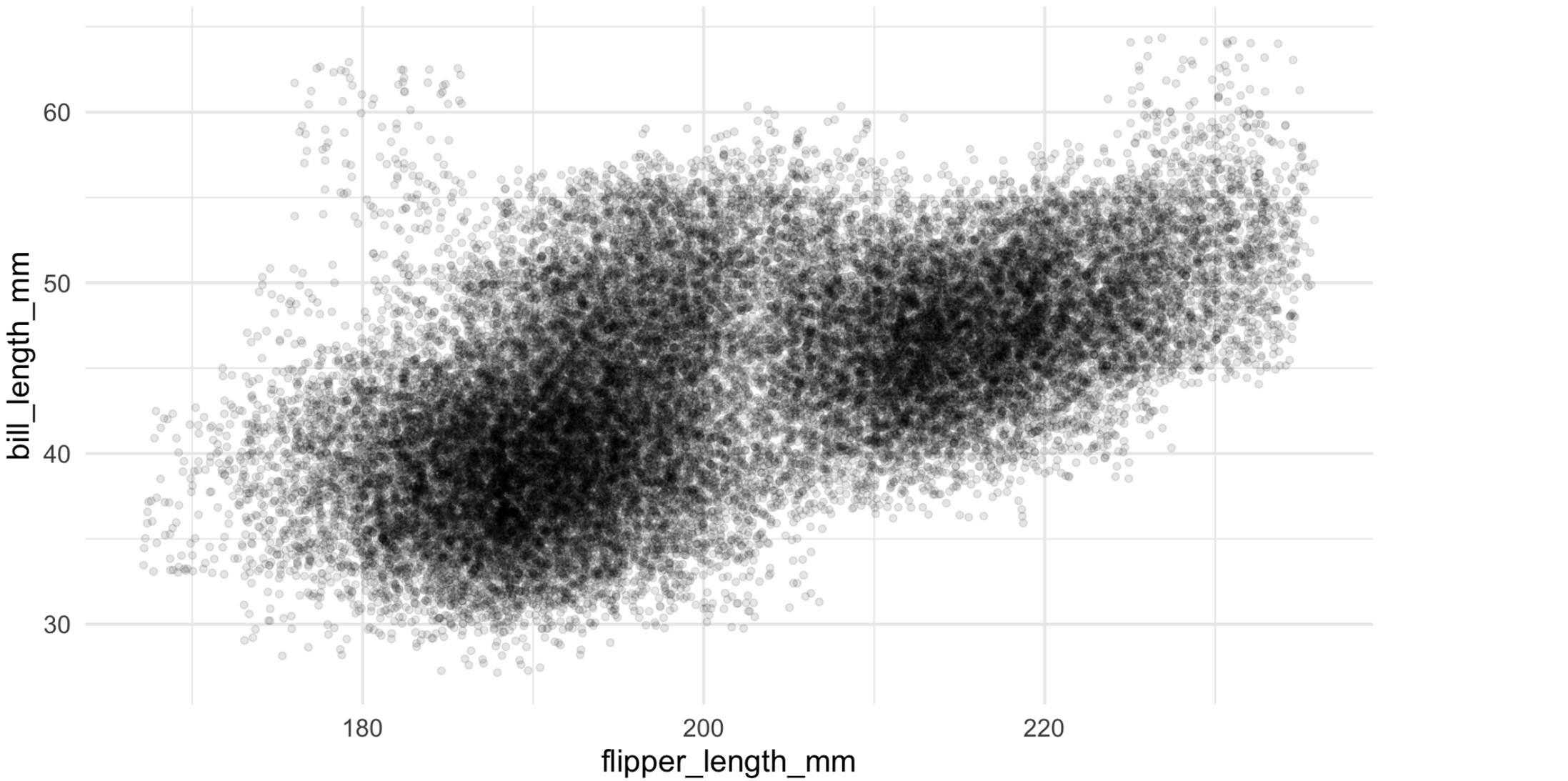

```
ggplot(penguins, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_point() +
  coord_fixed(ratio = 1)
```

TOO MANY POINTS



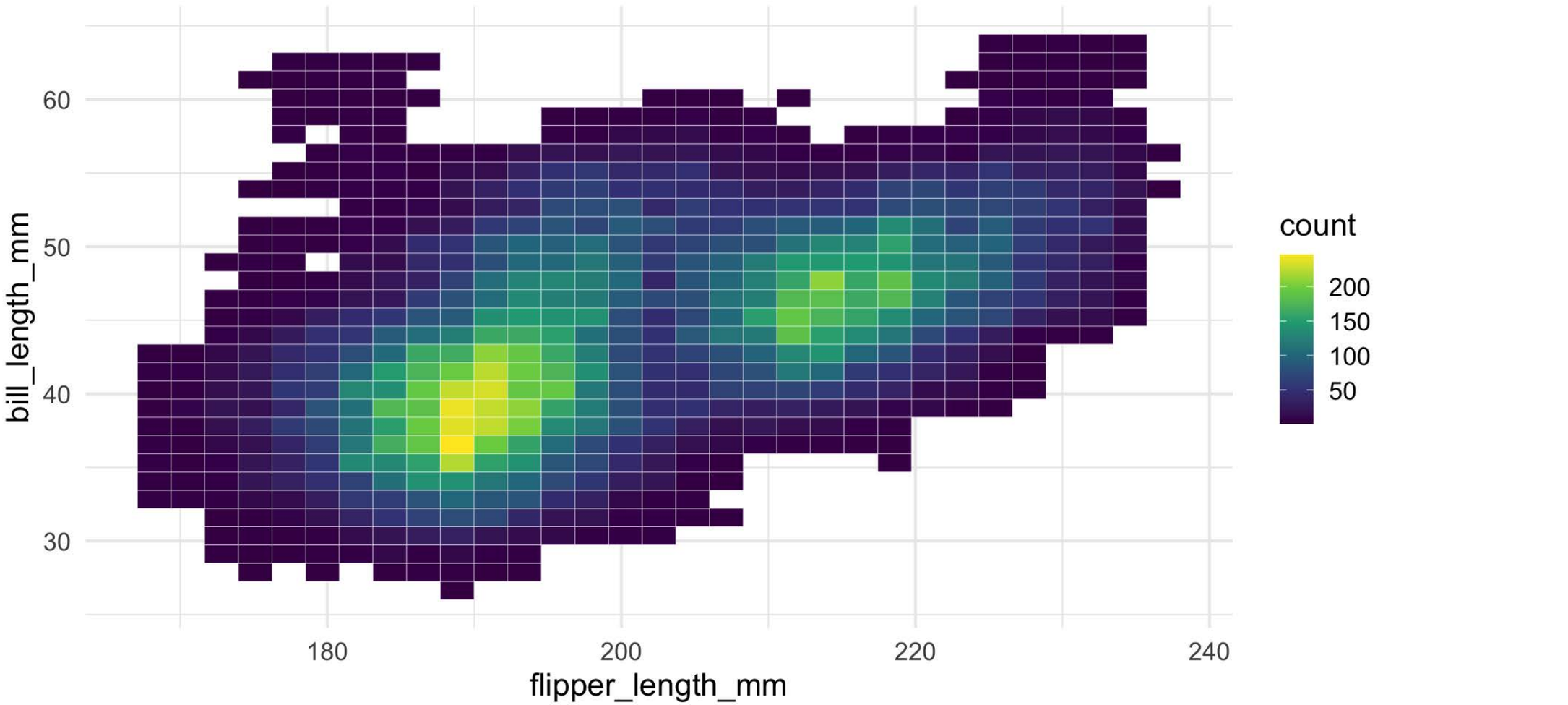

```
ggplot(penguins_rep, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_point() +
  coord_fixed(ratio = 1)
```

TRANSPARENCY



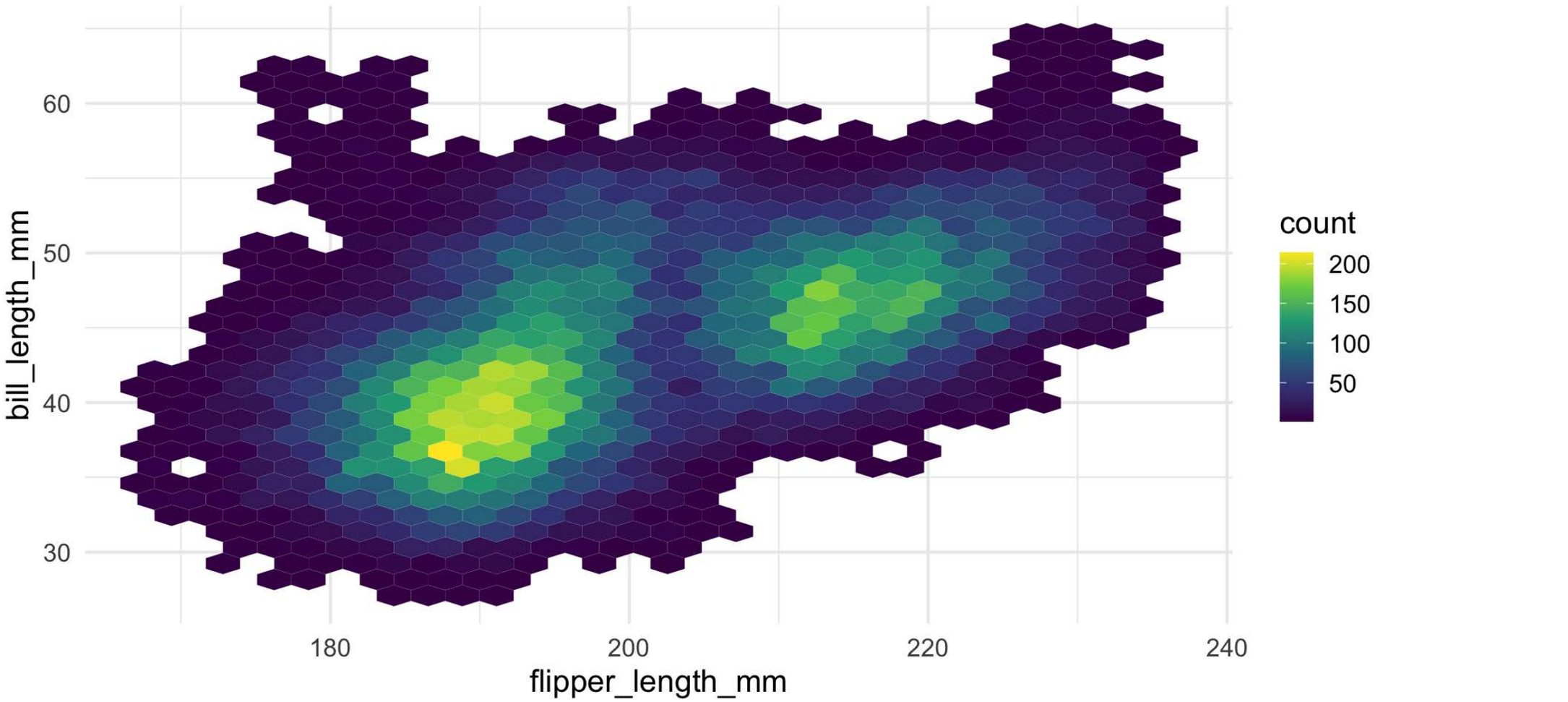

```
ggplot(penguins_rep, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_point(alpha = 0.1) +
  coord_fixed(ratio = 1)
```

2-D HISTOGRAM



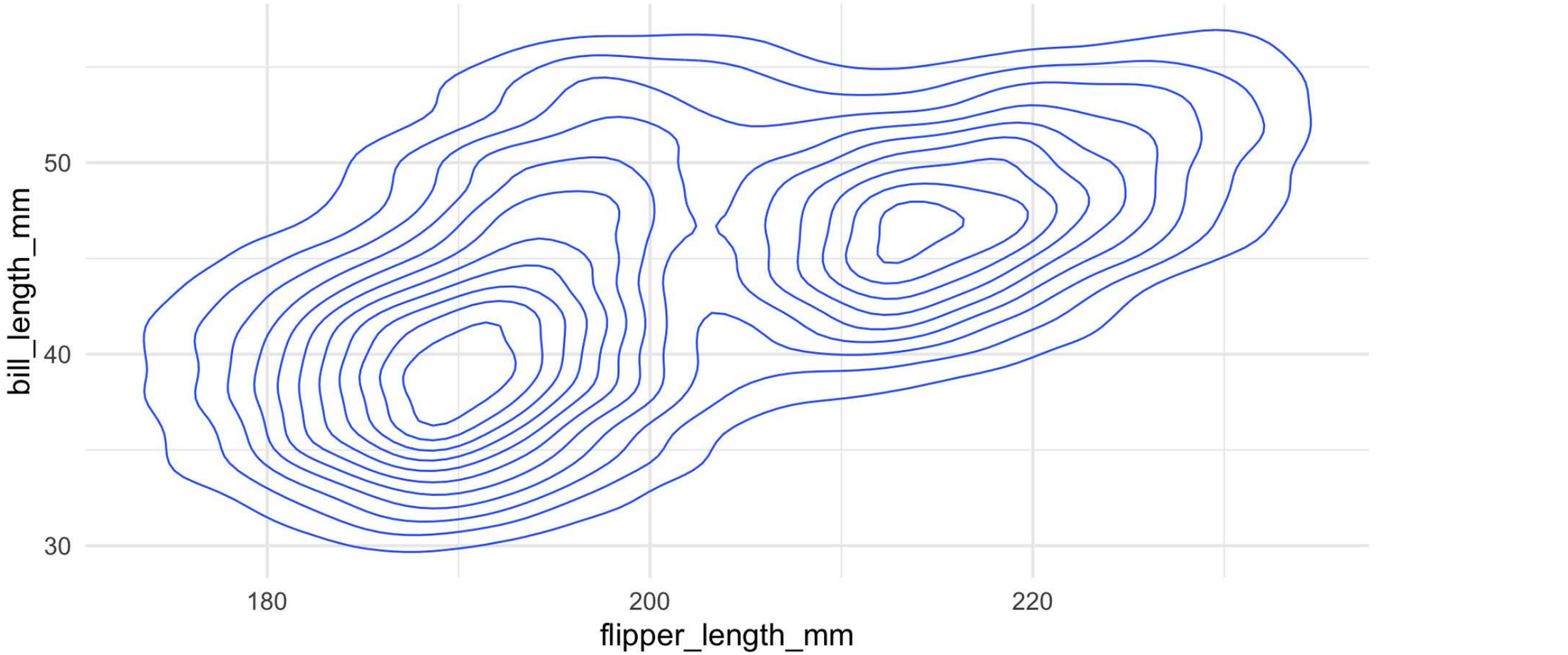

```
ggplot(penguins_rep, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_bin2d(color = "white") +
  scale_fill_viridis_c() +
  coord_fixed(ratio = 1)
```

HEXAGONAL BINS



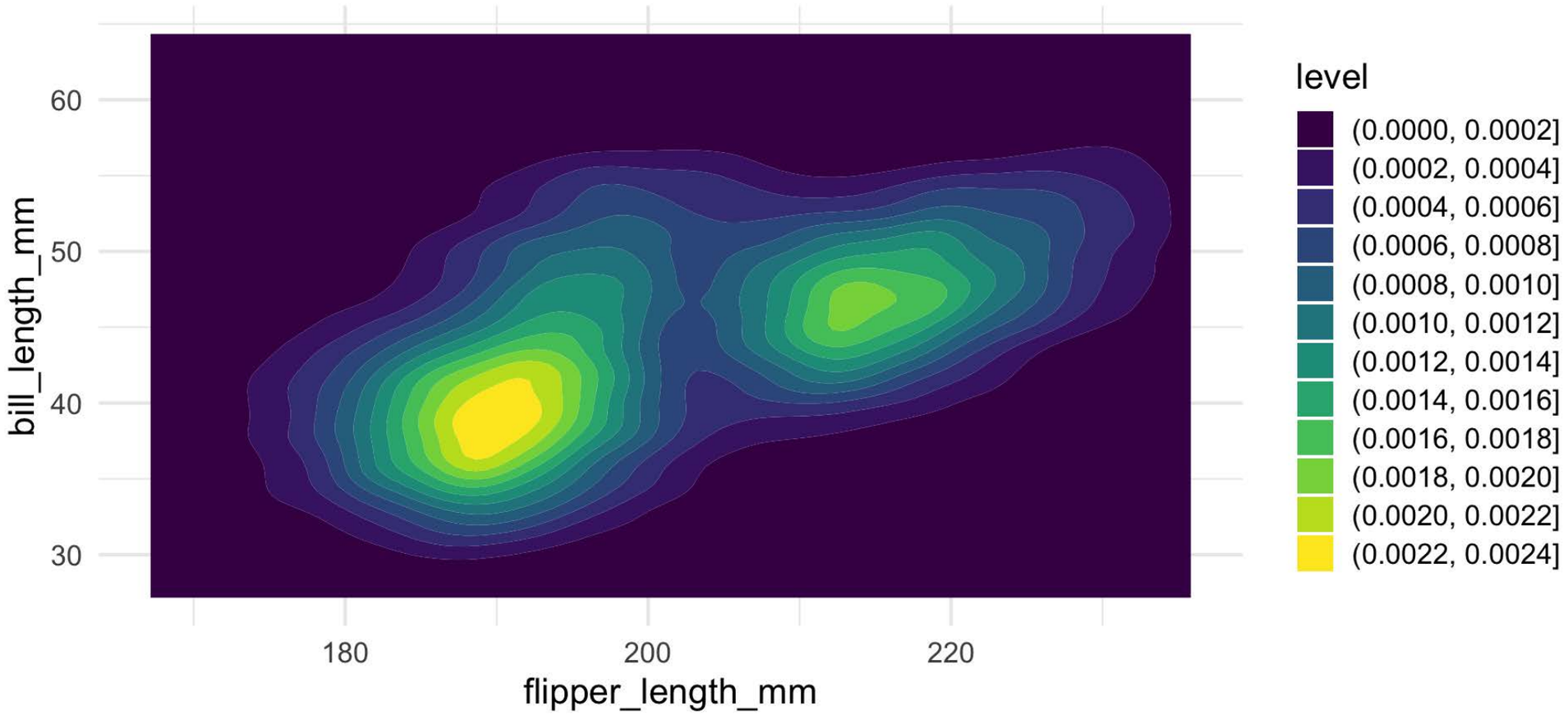

```
ggplot(penguins_rep, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_hex() +
  scale_fill_viridis_c() +
  coord_fixed(ratio = 1)
```

CONTOUR LINES




```
ggplot(penguins_rep, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_density2d() +
  coord_fixed(ratio = 1)
```

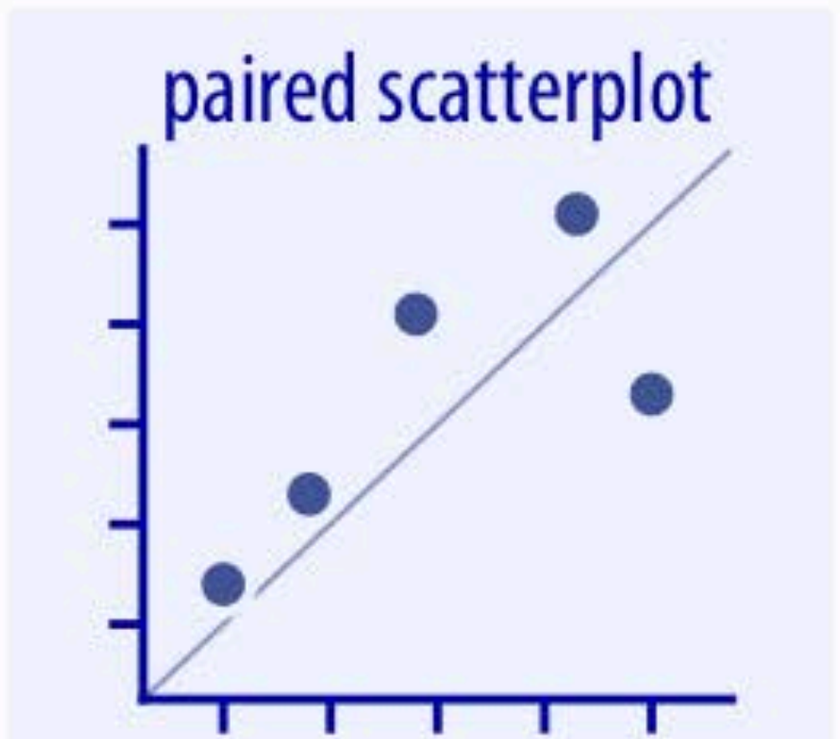
CONTOUR BANDS

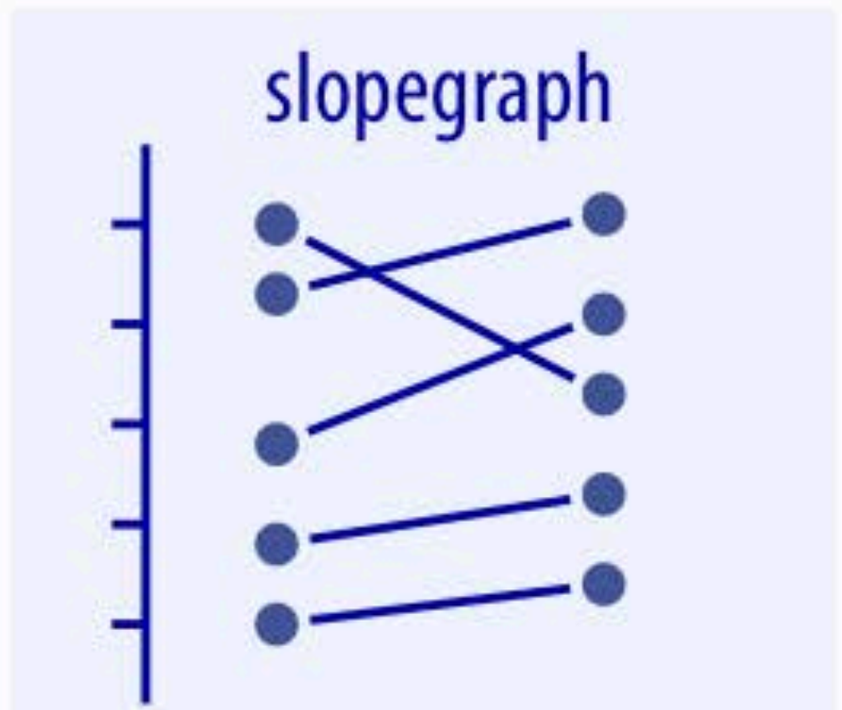



```
ggplot(penguins_rep, aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_density2d_filled() +
  scale_fill_viridis_d() +
  coord_fixed(ratio = 1)
```

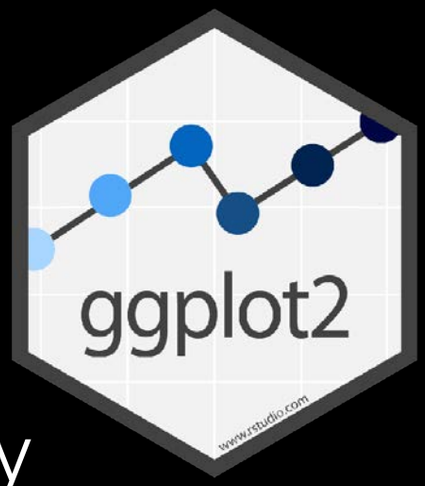
VISUALIZING DIFFERENCES IN PAIRED DATA

VISUALIZING DIFFERENCES IN PAIRED DATA





EXAMPLE: GENDER PAY GAP

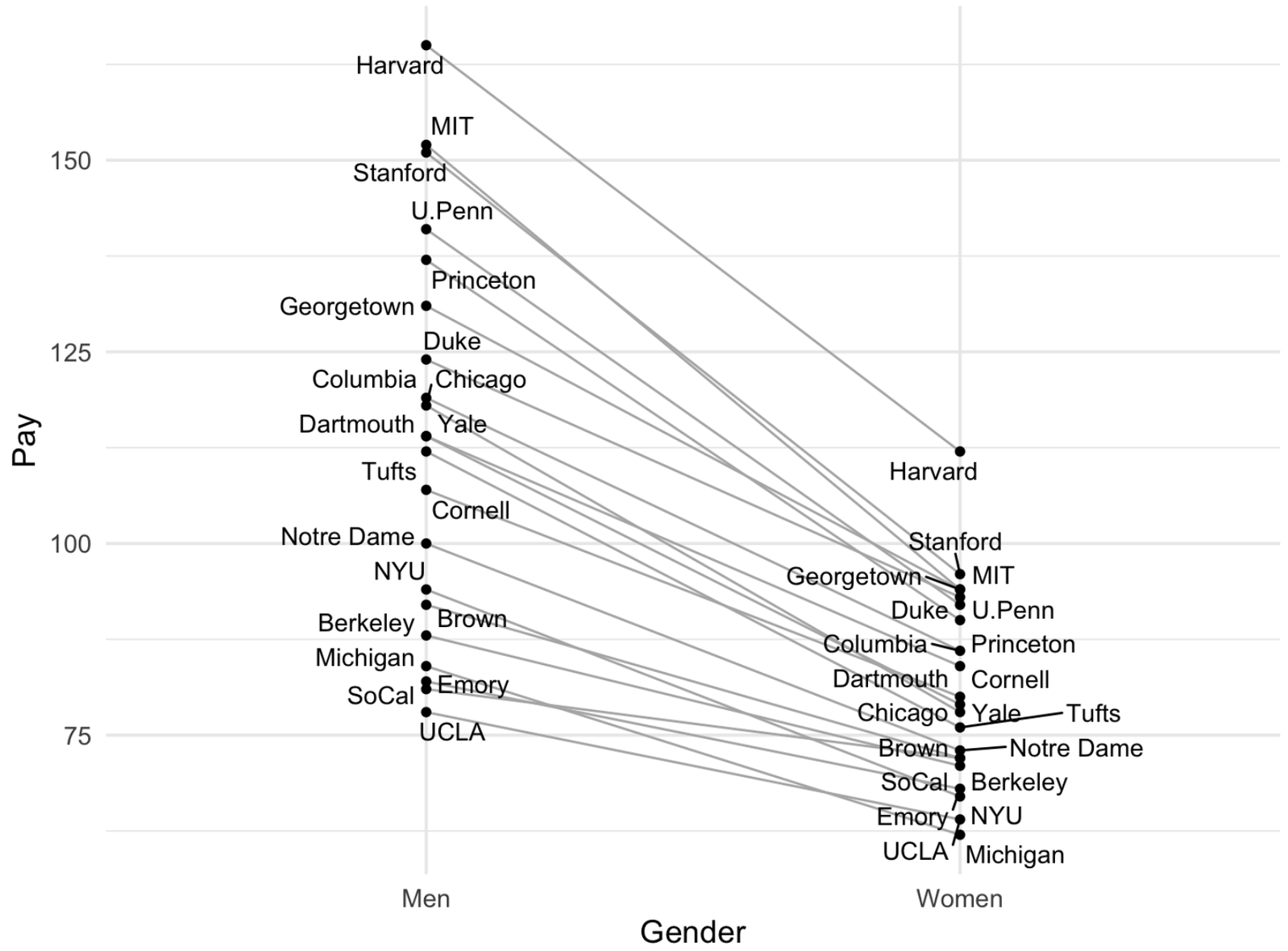


- The data show median mid-career annual salary (in thousands) for men and women who graduated from various elite universities.

- Task: visualize the gender pay gap for each school.

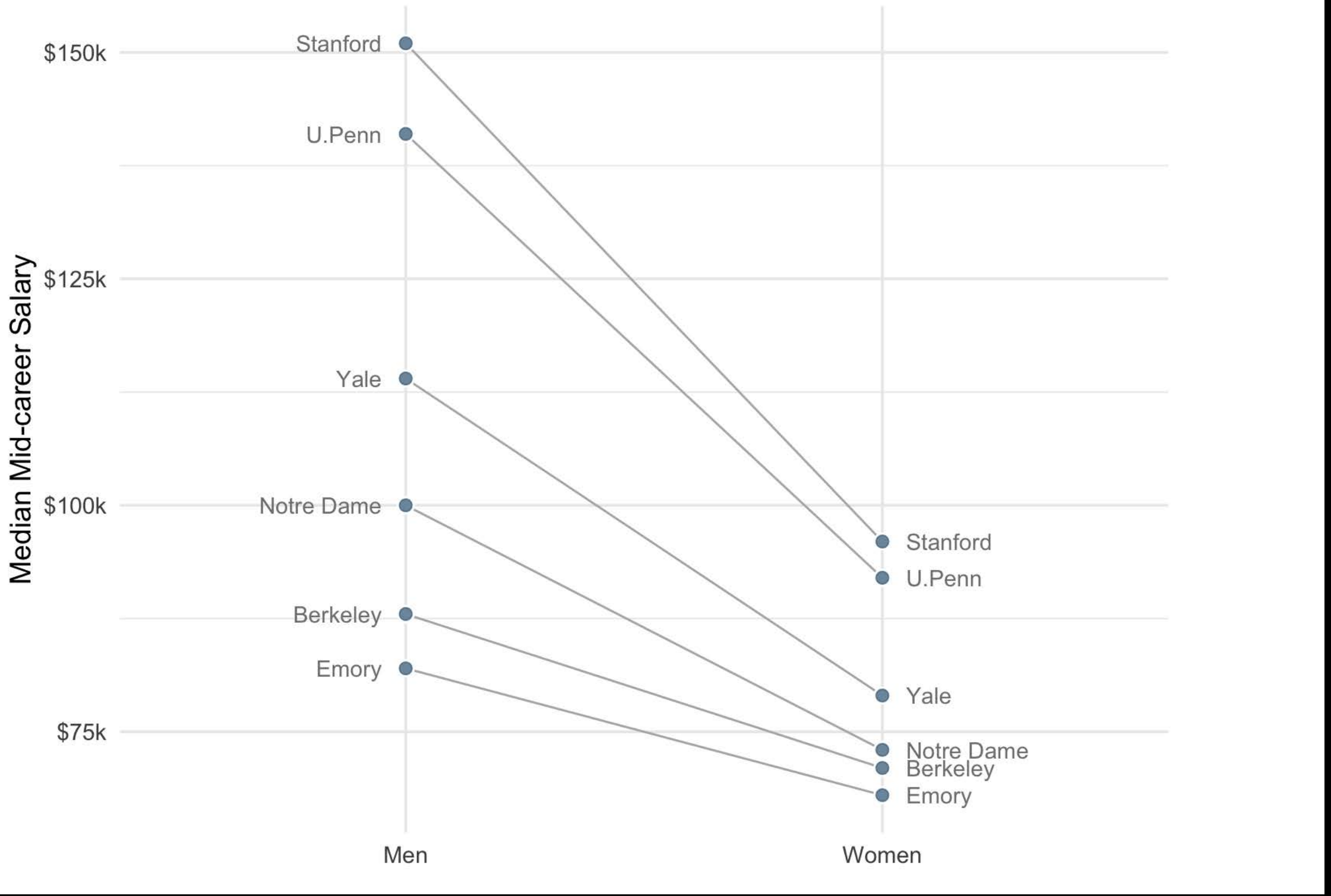
School	Gap	Gender	Pay
Berkeley	17	Men	88
Berkeley	17	Women	71
Brown	20	Men	92
Brown	20	Women	72
Chicago	40	Men	118
Chicago	40	Women	78
Columbia	33	Men	119
Columbia	33	Women	86
Cornell	27	Men	107
Cornell	27	Women	80

SLOPEGRAPH (BASICS)

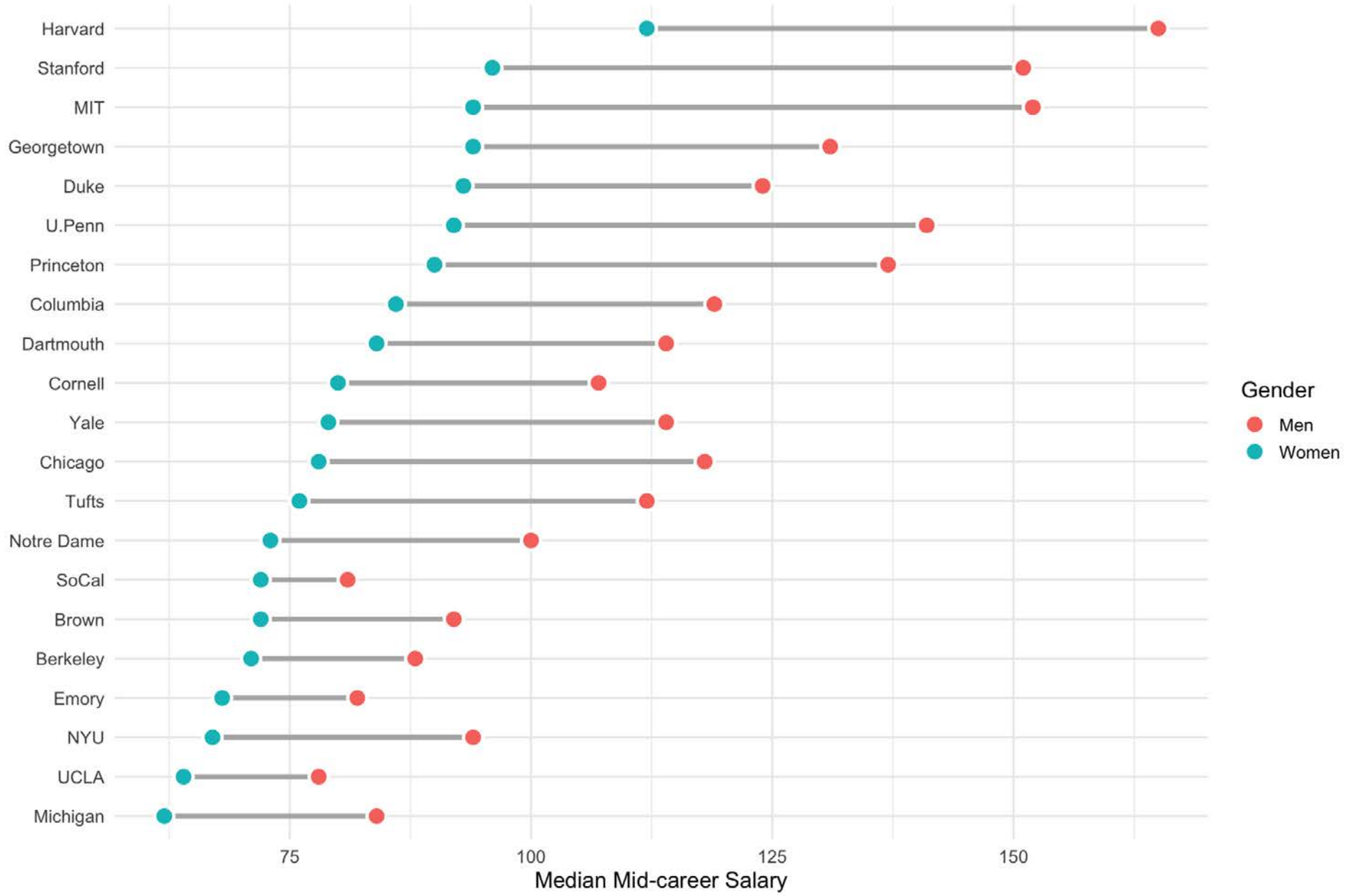



```
ggplot(earnings, aes(x = Gender, y = Pay)) +
  geom_line(aes(group = School)) +
  geom_point() +
  geom_text_repel(aes(label = School))
```

SLOPEGRAPH

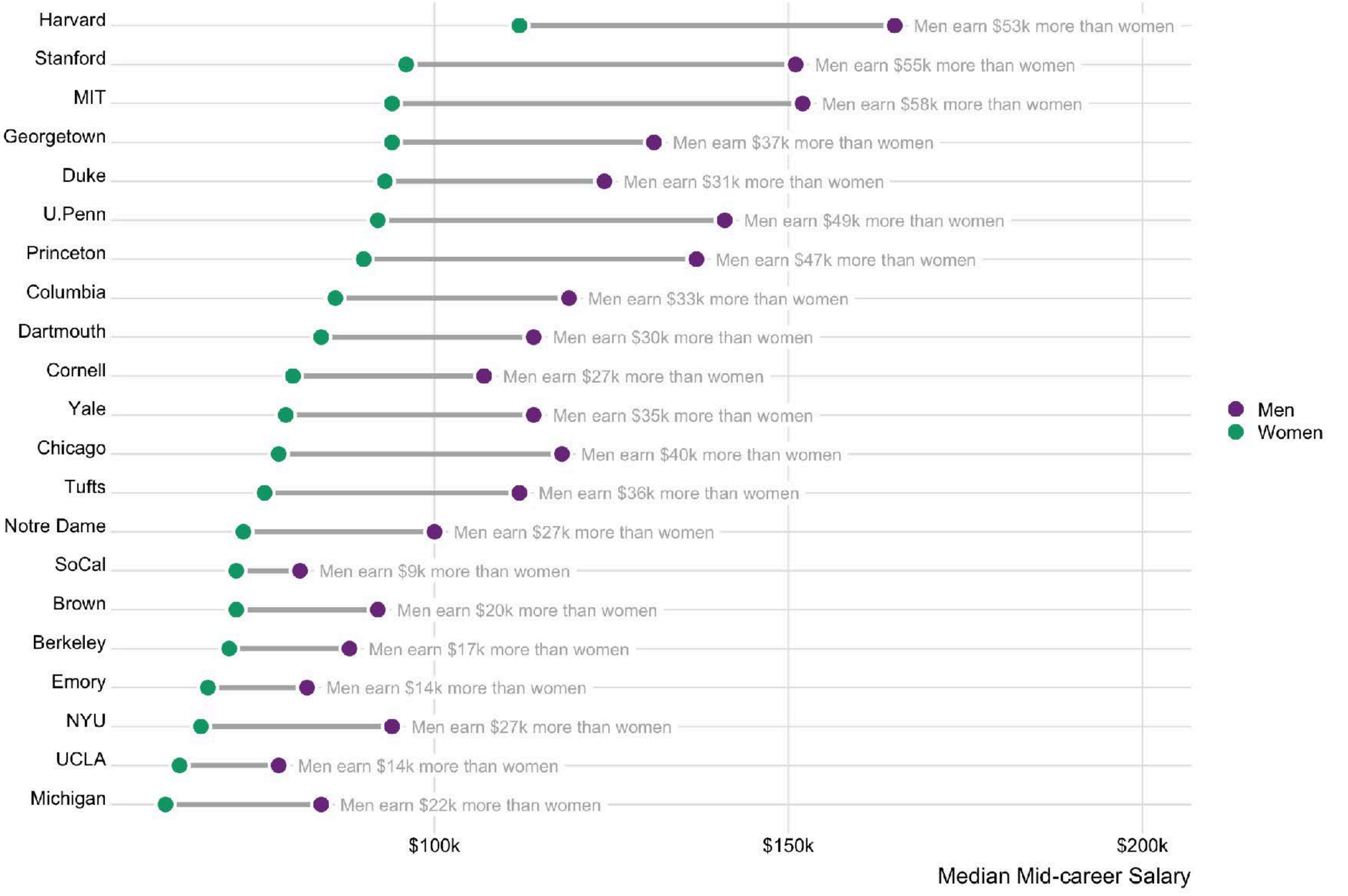

$150k
$125k
$100k
$75k
Median Mid-career Salary
Stanford
U.Penn
Yale
Notre Dame
Berkeley
Emory
Men
Women

DUMBELL PLOT

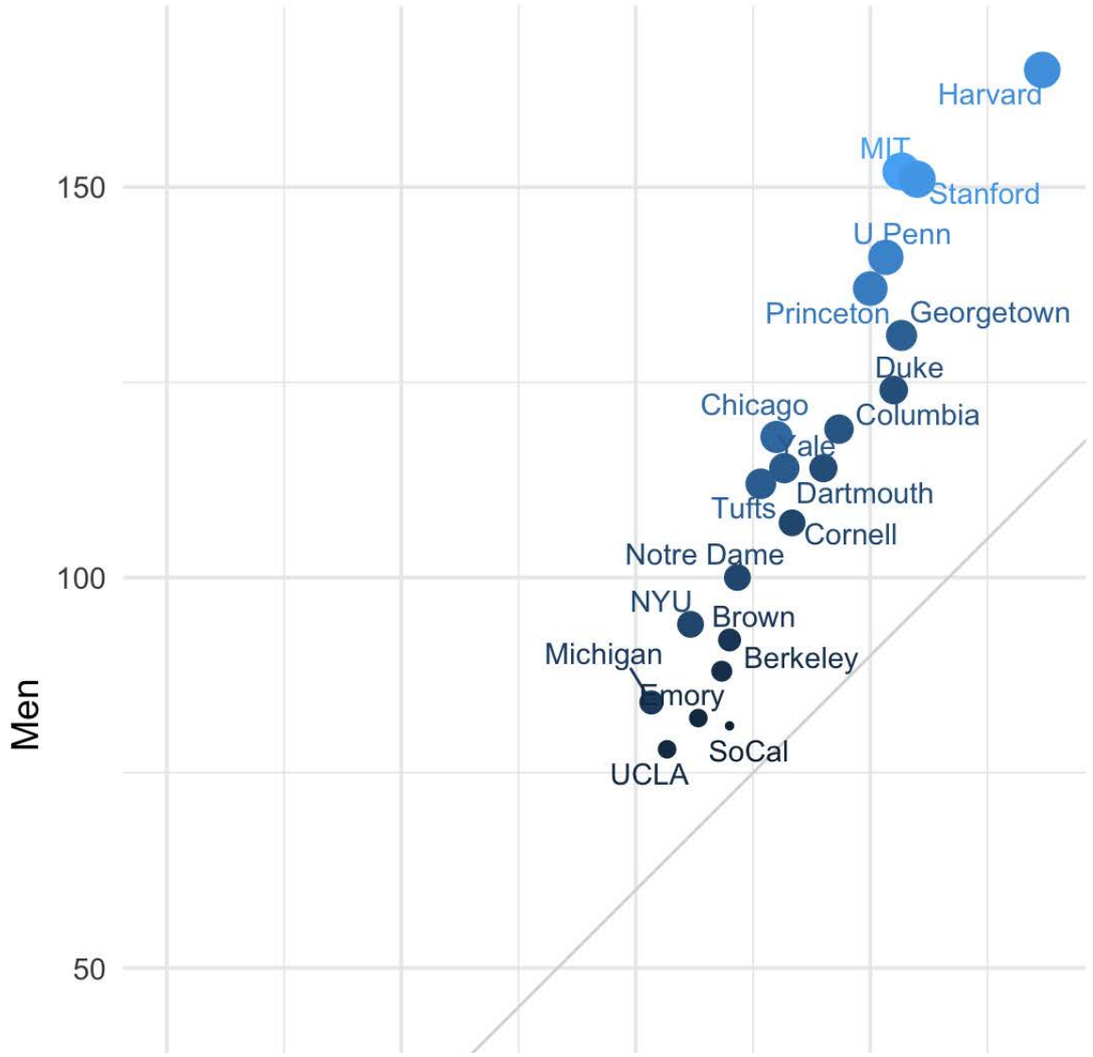



```
ggplot(earnings,
  aes(x = Pay, y = fct_reorder(School, Pay, .fun = "min"))) +
  geom_line(size = 1.5, color = "gray70") +
  geom_point(aes(color = Gender), size = 4)
```

DUMBELL PLOT

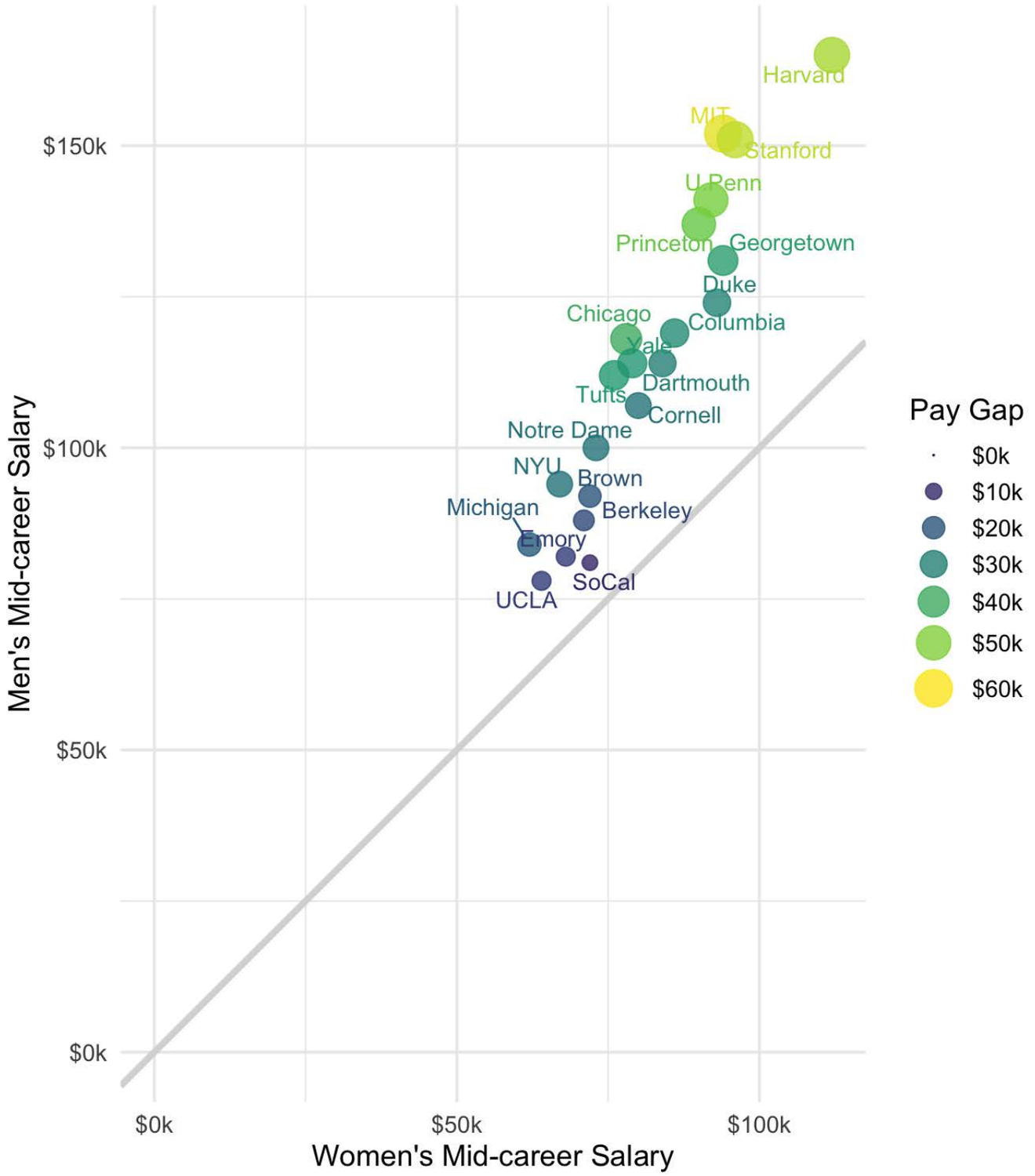


SLOPEGRAPH (BASICS)




```
ggplot(earnings_wide, aes(x = Women, y = Men)) +
  geom_abline(aes(slope = 1, intercept = 0)) +
  geom_point(aes(size = Gap, color = Gap)) +
  geom_text_repel(aes(color = Gap, label = School)) +
  coord_fixed(ratio = 1) +
  expand_limits(x = 0, y = 0)
```

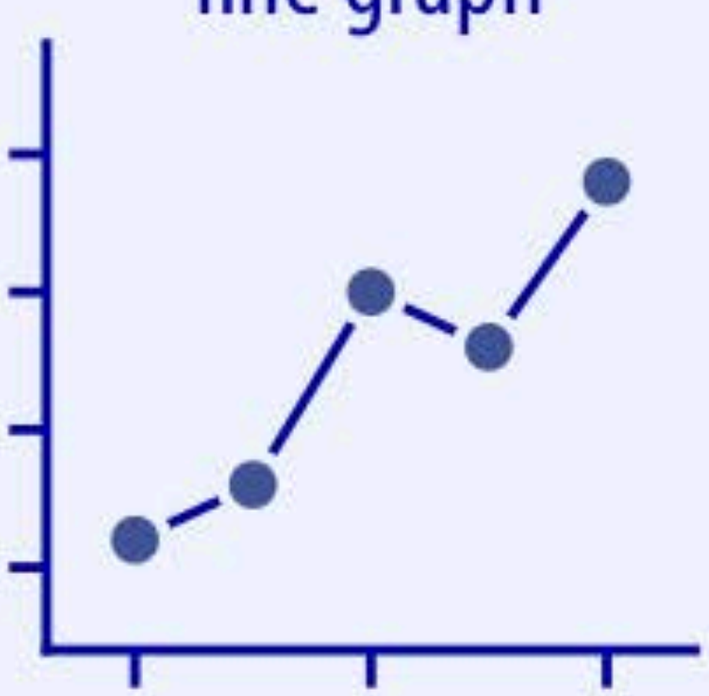
PAIRED SCATTERPLOT



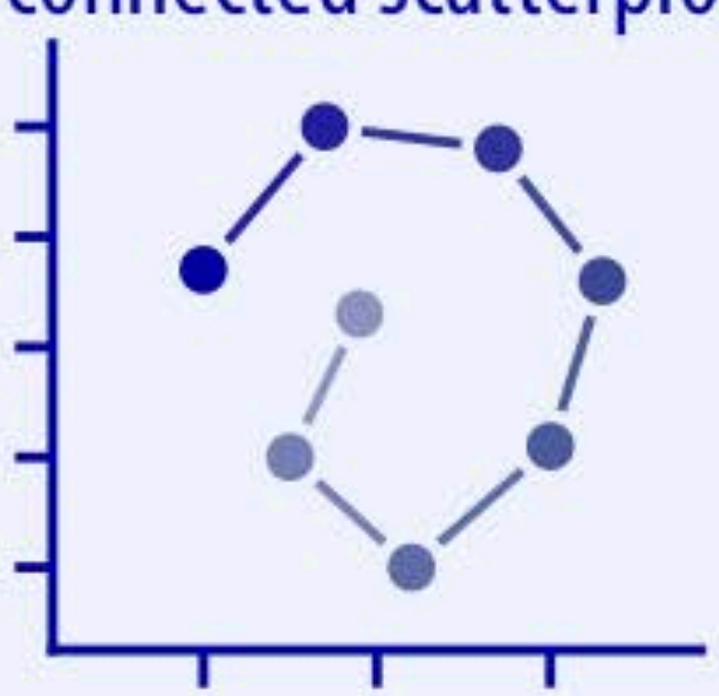
VISUALIZING CHANGE OVER TIME

VISUALIZING CHANGE OVER TIME

line graph



connected scatterplot



LINE GRAPHS

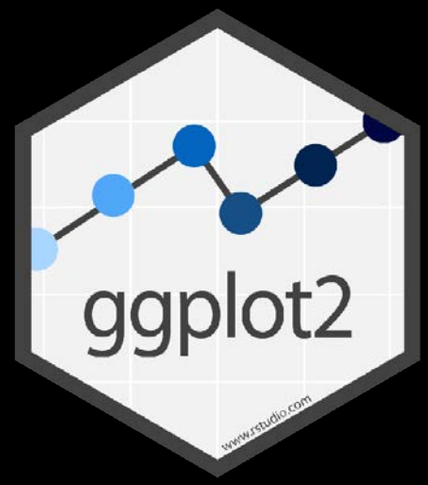
- Typically used when the x-axis represents date/time and the y-axis represents some other numerical variable.

- This is called a *time series*.

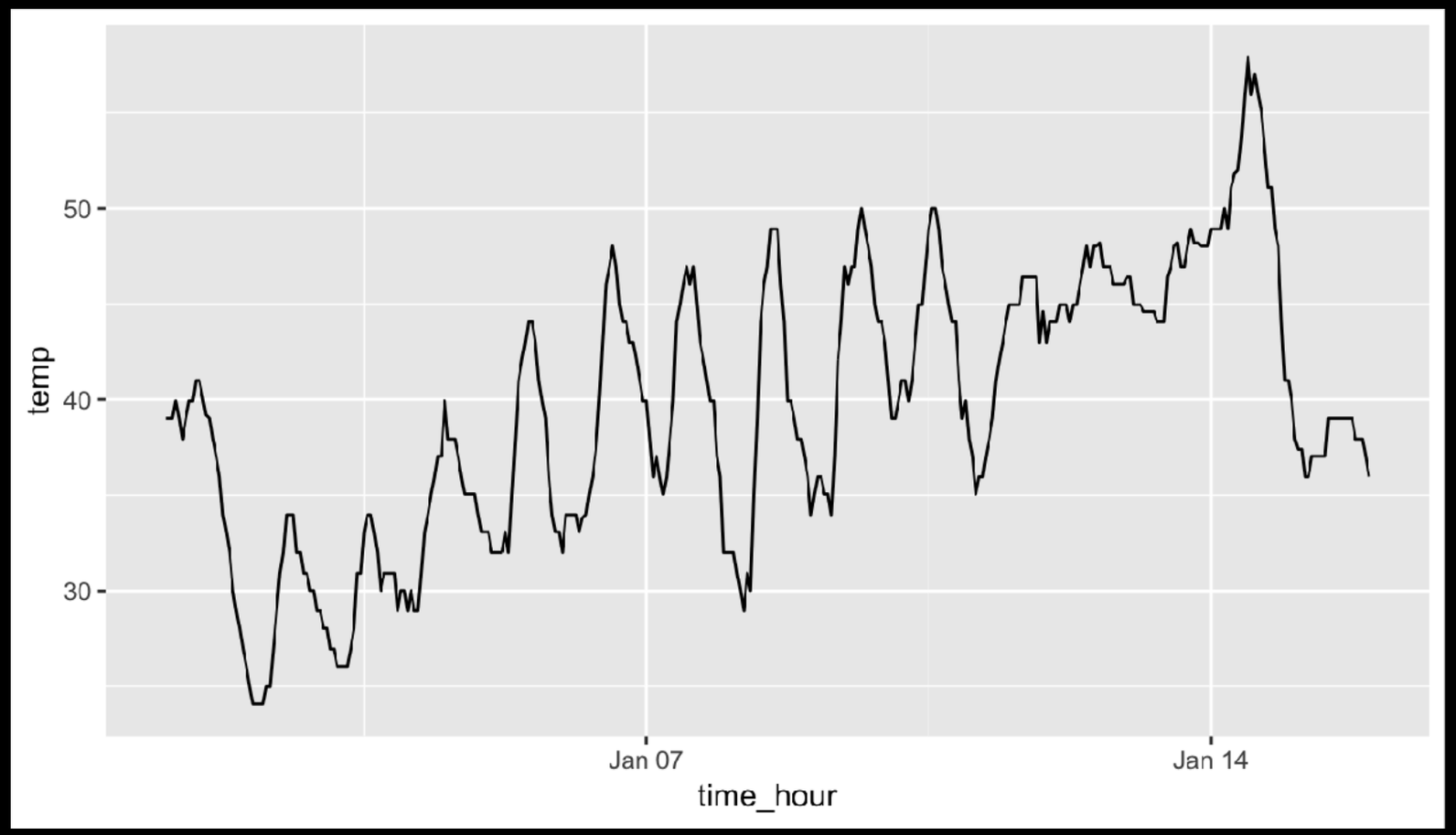
- The line between points implies that they are connected through some defined order.

- Line graphs should not be used when there is not a natural sequential ordering to the data!

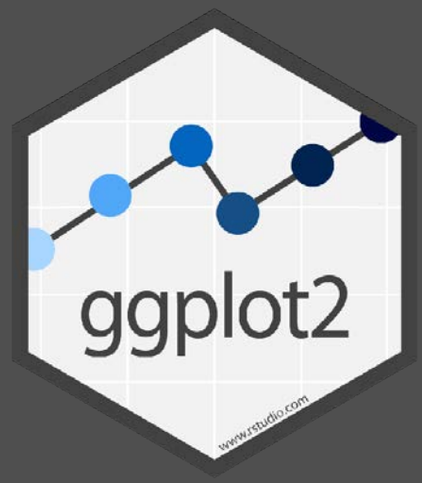
LINE GRAPHS



- Geometric object is `geom_line()`
- One numeric variable to mapped to x
- Another numeric variable to mapped to y



YOUR TURN



- Go to this week's assignments on the course website.

- Install the `babynames` package.

- Follow the instructions in the babynames.Rmd file, and answer the questions about trends in baby names.