WEEK 4

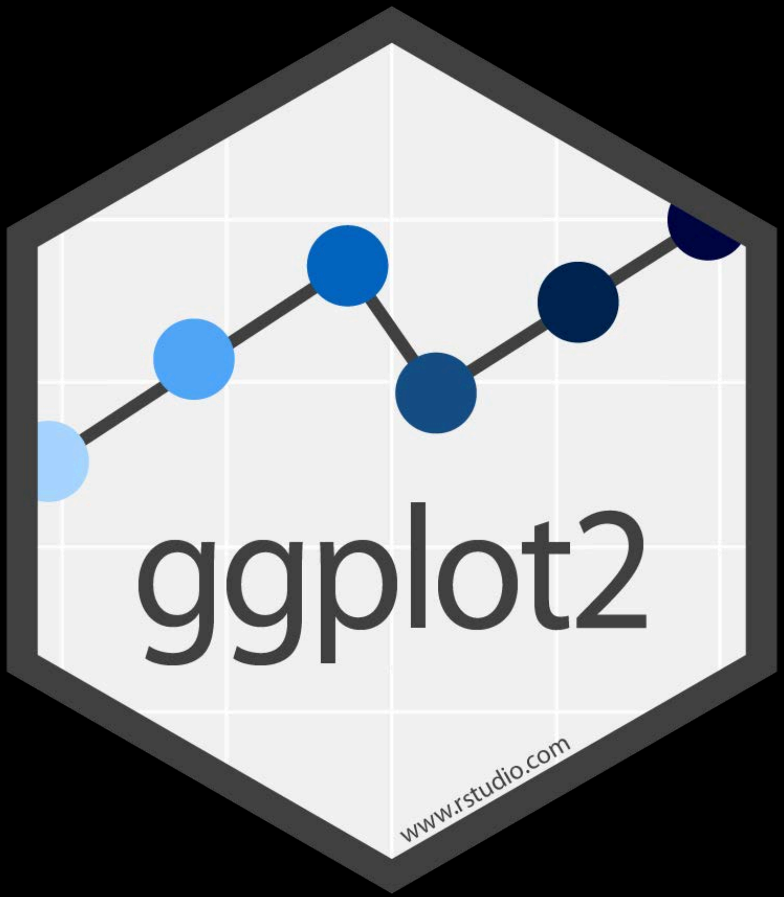
VISUALIZING AMOUNTS AND PARTS OF A WHOLE

TODAY'S TOPICS

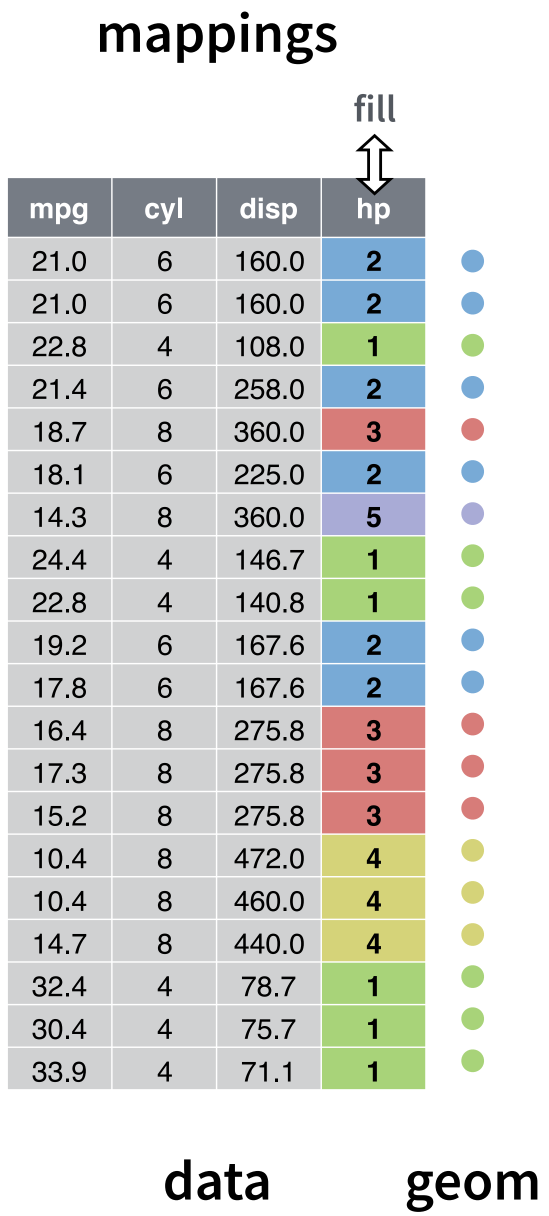
- Visualizing amounts and proportions with pie charts, bar charts, and variations
- Fine-tuning ggplot2

GGPLOT2

A GRAMMAR OF GRAPHICS



REVIEW



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	●

1. Pick a **data** set

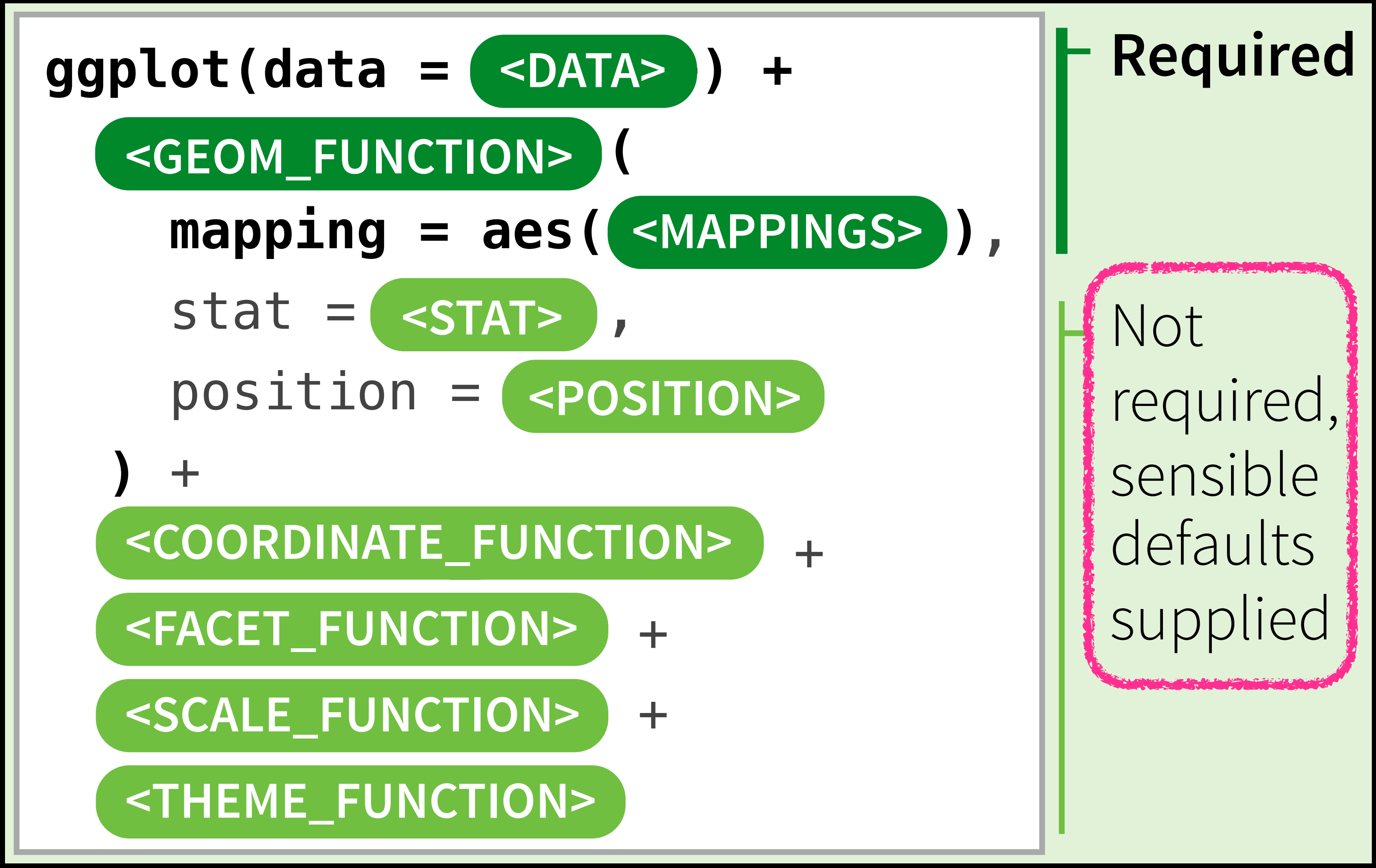
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

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

3. **Map** aesthetic properties to variables

WHAT ELSE?

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes



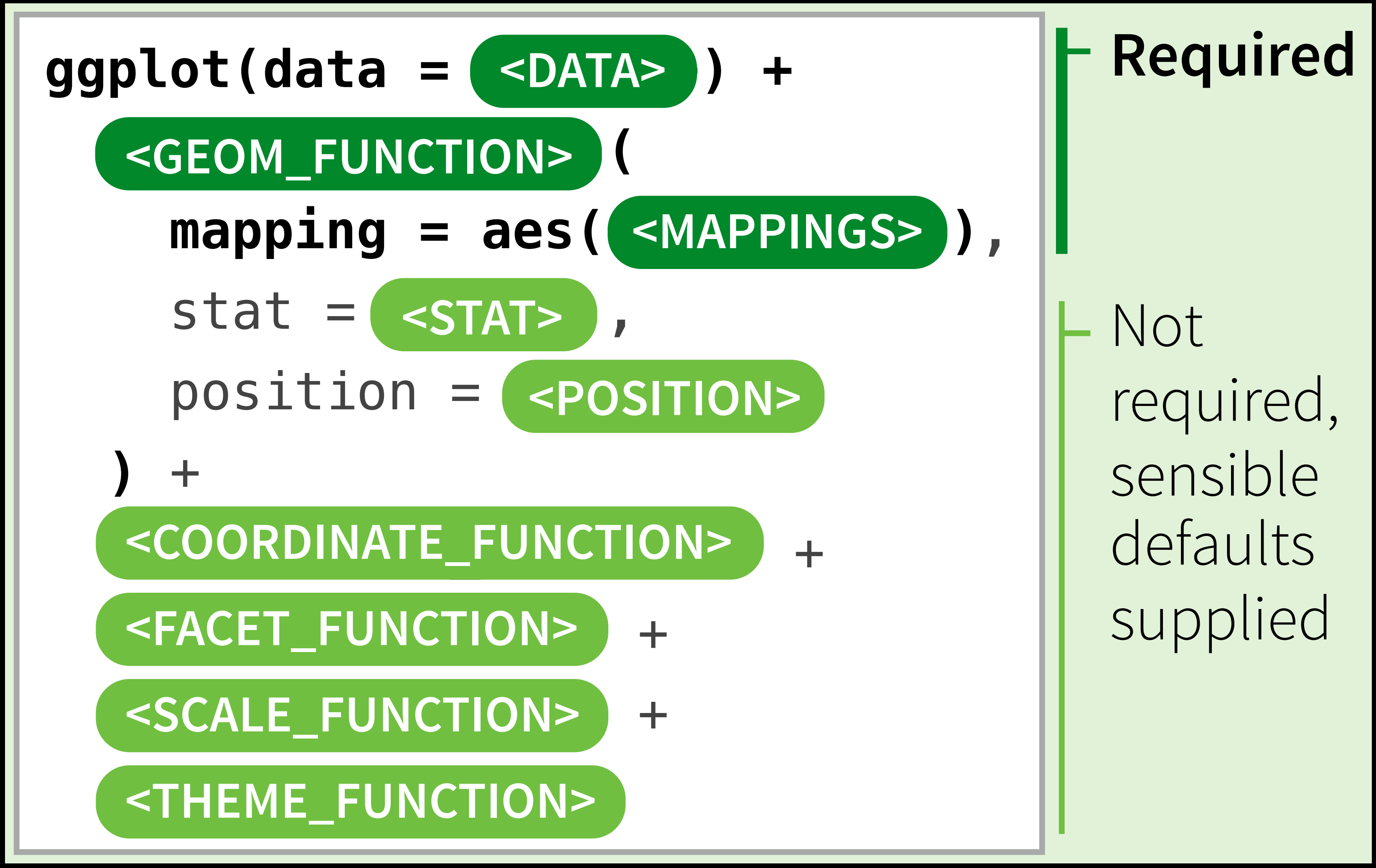
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

Required

Not required, sensible defaults supplied

WHAT ELSE?

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes



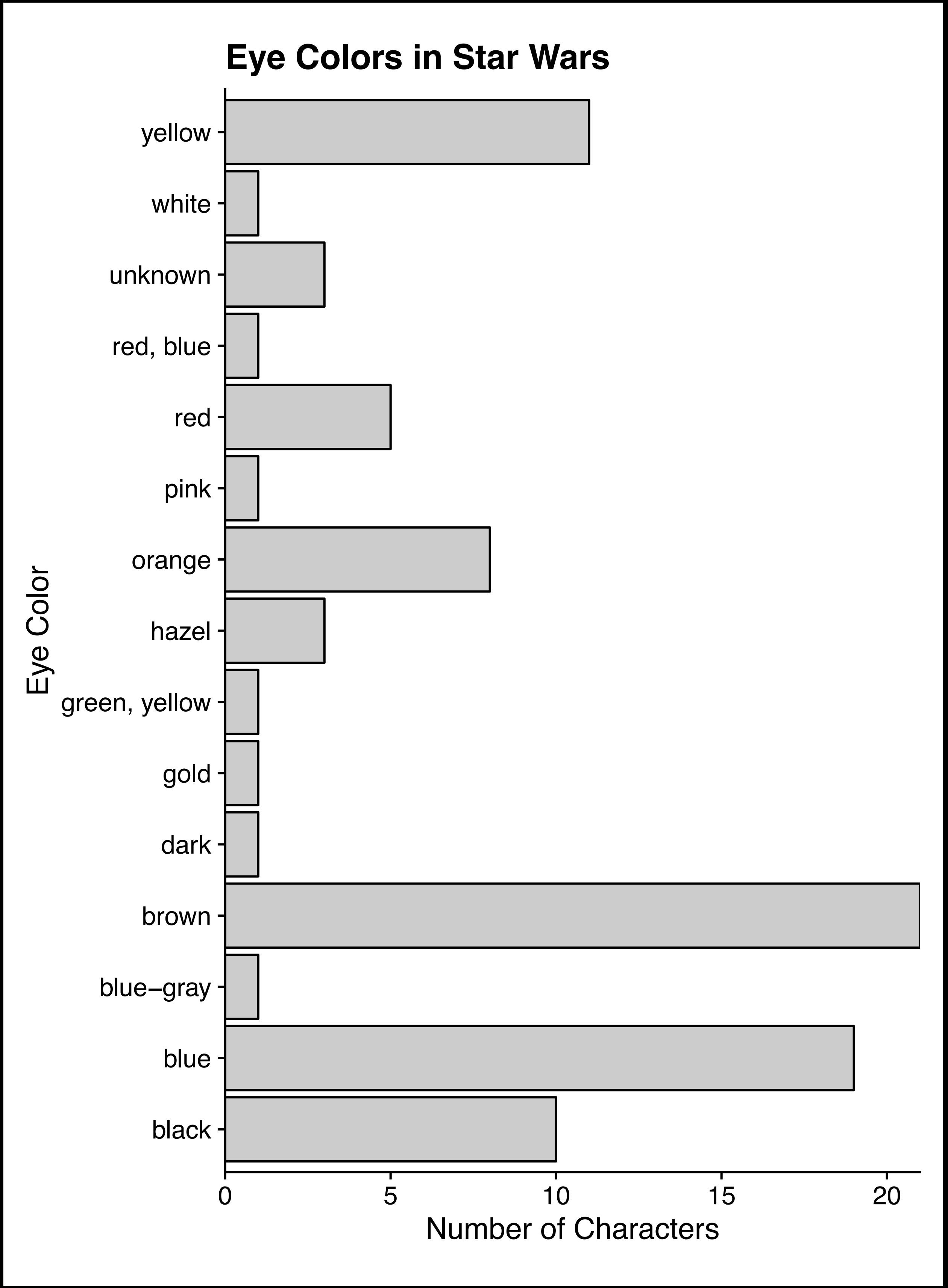
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

STATS AND POSITIONS

- Each geom_ function has a default stat and position, *so you can usually omit it!*
- But there are some situations where they are useful.
- Let’s explore how stats and positions work using bar charts.

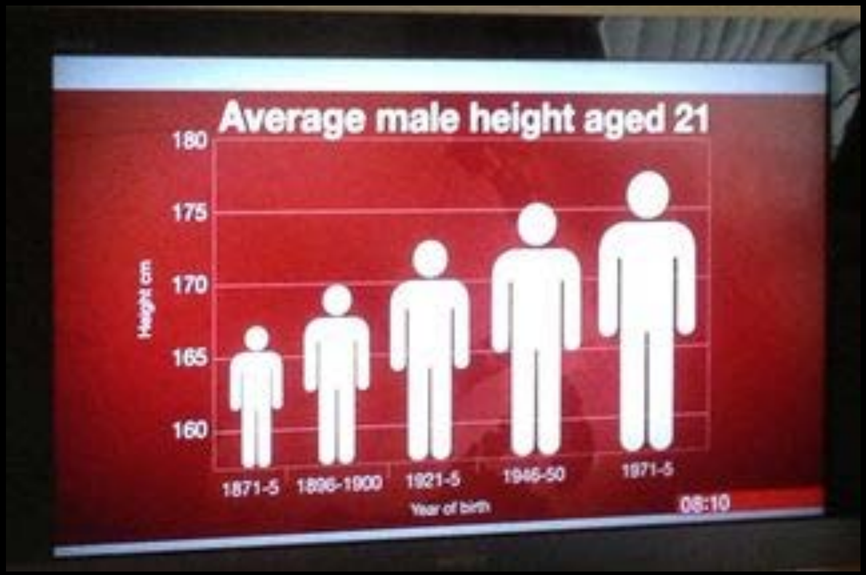
BAR CHARTS

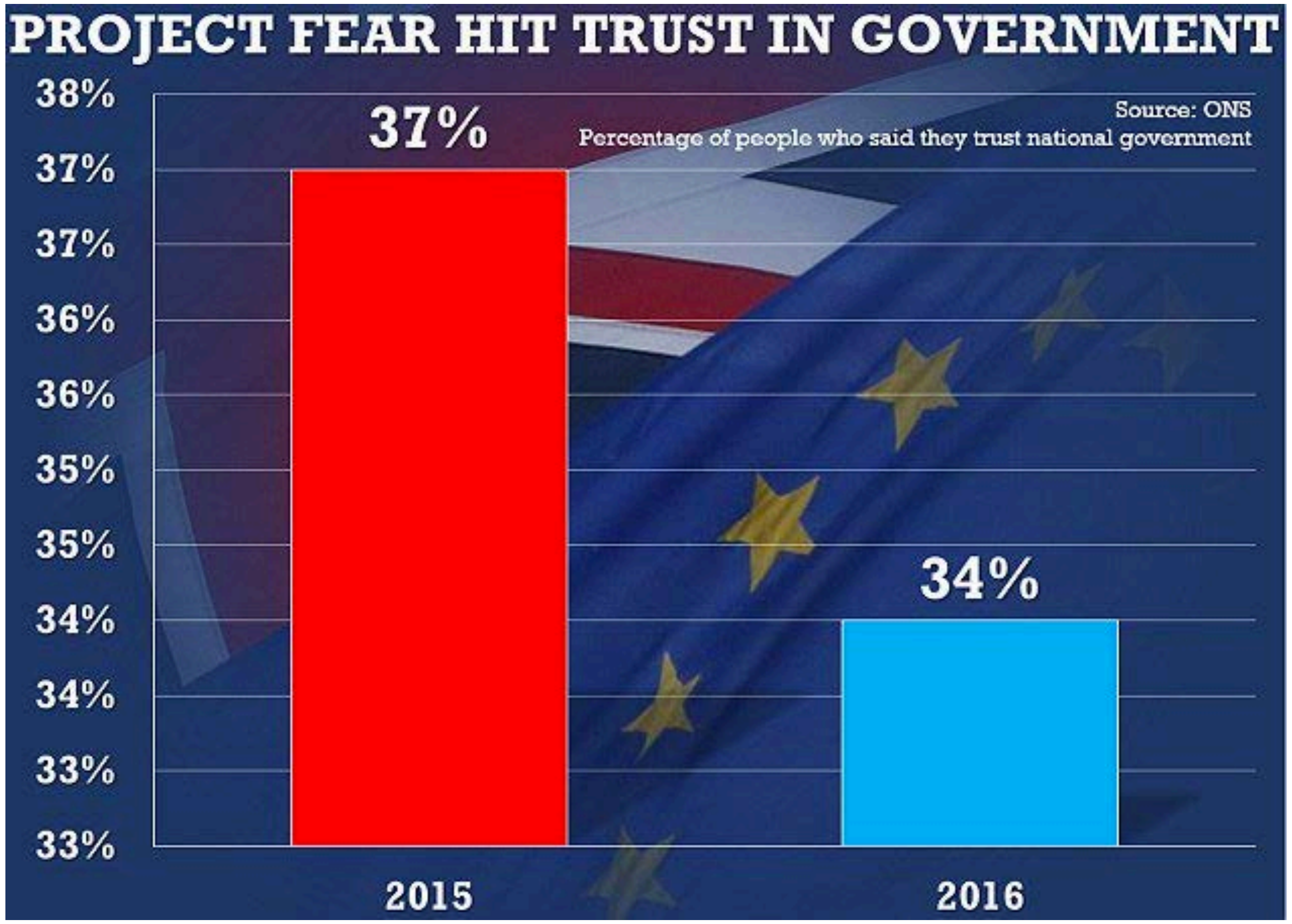
- Used for discrete groups or categories
- Y-axis should always include zero!

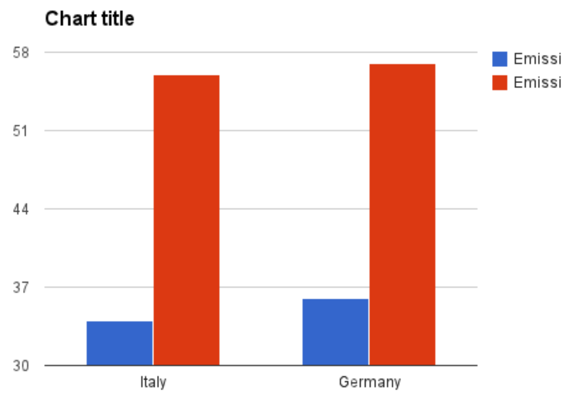


BAR CHARTS

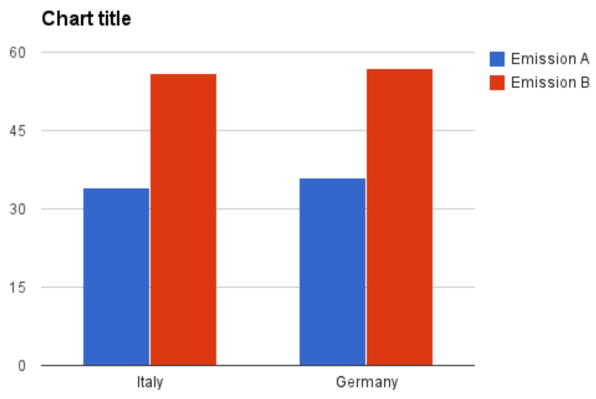
- Used for discrete groups or categories
- Y-axis should always include zero!







No

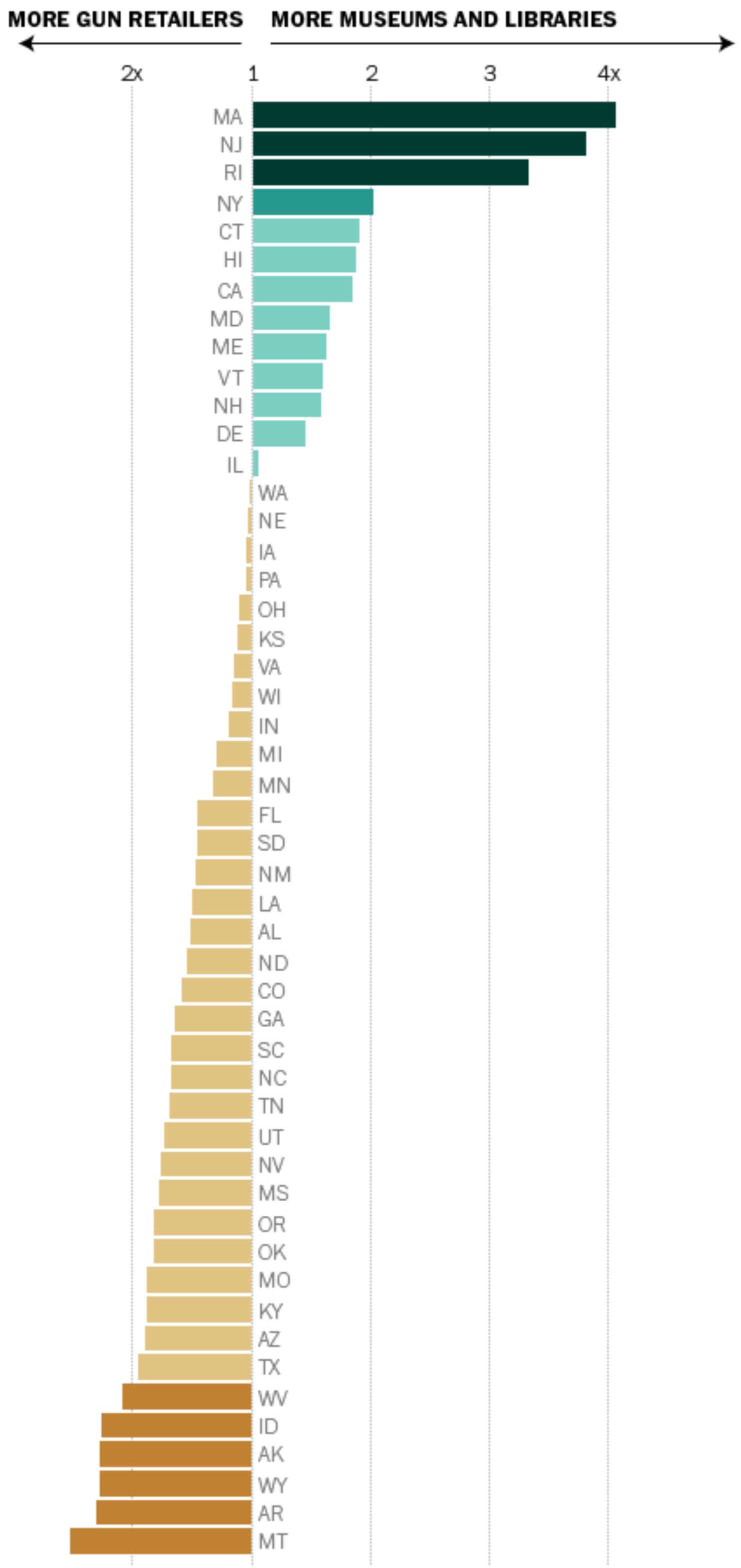


Yes

BAR CHARTS

- Used for discrete groups or categories
- Y-axis should always include zero!
  - Very few exceptions

In 37 states, gun dealers outnumber museums and libraries

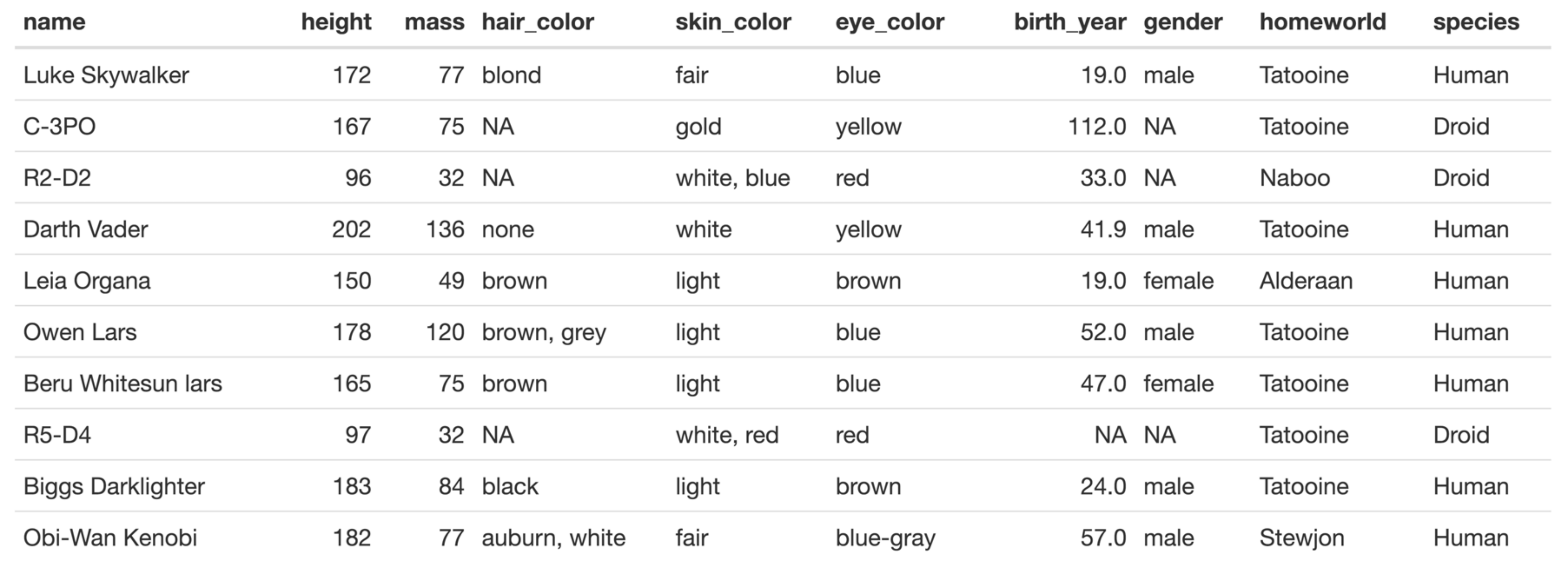


SOURCE: Institute of Museum and Library Sciences; Bureau of Alcohol, Tobacco and Firearms.
GRAPHIC: The Washington Post. Published June 17, 2014

YOUR TURN

- Let's create a bar in ggplot2 using the `starwars` dataset that comes with `tidyverse`

- First 10 rows:



name	height	mass	hair_color	skin_color	eye_color	birth_year	gender	homeworld	species
Luke Skywalker	172	77	blond	fair	blue	19.0	male	Tatooine	Human
C-3PO	167	75	NA	gold	yellow	112.0	NA	Tatooine	Droid
R2-D2	96	32	NA	white, blue	red	33.0	NA	Naboo	Droid
Darth Vader	202	136	none	white	yellow	41.9	male	Tatooine	Human
Leia Organa	150	49	brown	light	brown	19.0	female	Alderaan	Human
Owen Lars	178	120	brown, grey	light	blue	52.0	male	Tatooine	Human
Beru Whitesun lars	165	75	brown	light	blue	47.0	female	Tatooine	Human
R5-D4	97	32	NA	white, red	red	NA	NA	Tatooine	Droid
Biggs Darklighter	183	84	black	light	brown	24.0	male	Tatooine	Human
Obi-Wan Kenobi	182	77	auburn, white	fair	blue-gray	57.0	male	Stewjon	Human

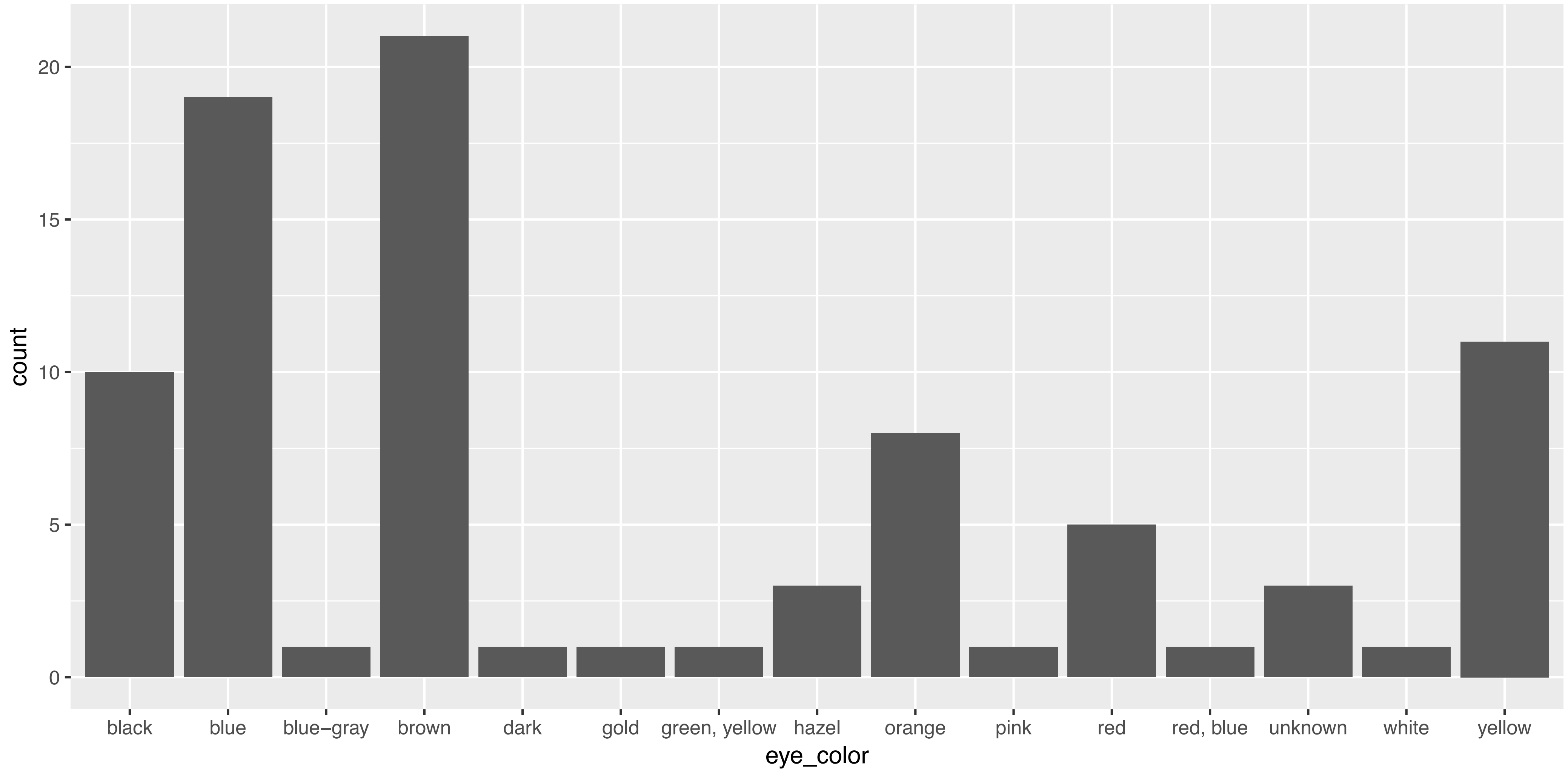
- ***How many*** characters have each different eye color?

YOUR TURN

- Create a new folder for this week's files
- Create a new R markdown file and clear out the extra stuff
- Create a new R chunk and load the `tidyverse` package
- Try to create a bar chart of eye_color using `geom_bar()`
- Hint: `?geom_bar()`

3:00

YOUR TURN



```
ggplot(starwars, aes(x = eye_color)) +
  geom_bar()
```

STATS AND POSITIONS

- Note that counts were calculated automatically! This happened because the default stat of `geom_bar()` is *count*.

- Look at the help function for `geom_bar()` and see if you can find this information.

```
?geom_bar()
```

- For most geoms, the default stat is "identity," meaning leave the data as is.

STATS AND POSITIONS

- What if our data are already summarized?
- In other words, what if we need to use a different stat?
  - We can provide it.

YOUR TURN

- Let's summarize the starwars data set by eye color using the `dplyr` function `count()`.

- Run this code, look at the result, and describe what happened.

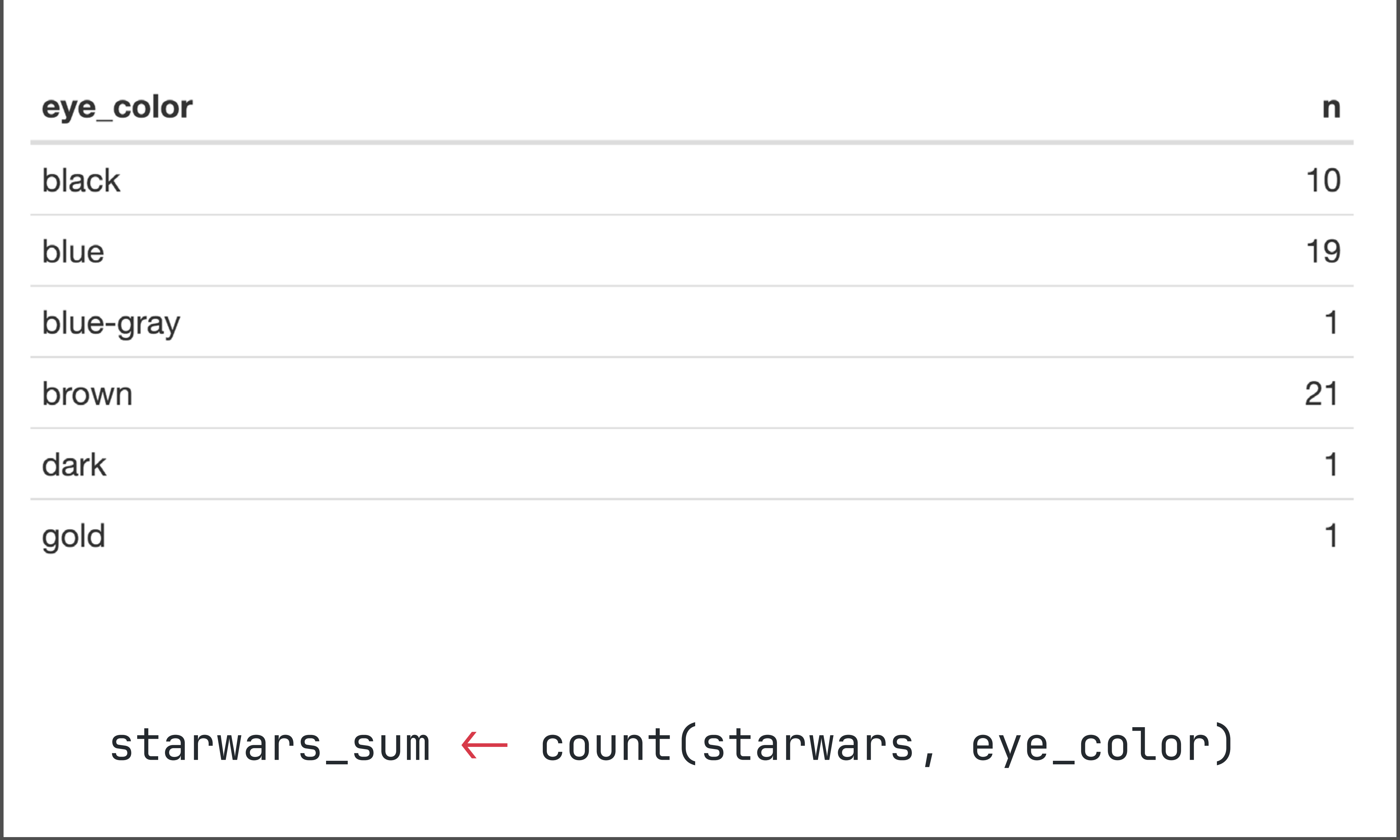
```
starwars_sum <- count(starwars, eye_color)
```

Data set

Category to count

1:00

YOUR TURN



eye_color	n
black	10
blue	19
blue-gray	1
brown	21
dark	1
gold	1

```
starwars_sum <- count(starwars, eye_color)
```

YOUR TURN

- Try to predict what will happen for each of the following, then run it (in the console).

```r
ggplot(starwars, aes(x = eye_color)) +
  geom_bar()

ggplot(starwars, aes(x = eye_color)) +
  geom_col()

ggplot(starwars_sum, aes(x = eye_color, y = n)) +
  geom_col()

ggplot(starwars_sum, aes(x = eye_color, y = n)) +
  geom_bar()

ggplot(starwars_sum, aes(x = eye_color, y = n)) +
  geom_bar(stat = "identity")
```

4:00

STATS AND POSITIONS

- Bad combinations of stats and mappings produce errors
  - `geom_bar()`'s count stat *calculates* a y value by counting, so there's a conflict it you also try mapping something to y.
  - `geom_col()`'s identity stat *requires* a y value, so there's an error if you don't provide one with a mapping.
- You can *override* the default aesthetics for `geom_` functions if you really want to (this is often a bad idea).
- Bottom line: if the data are already summarized, then use `geom_col()`.

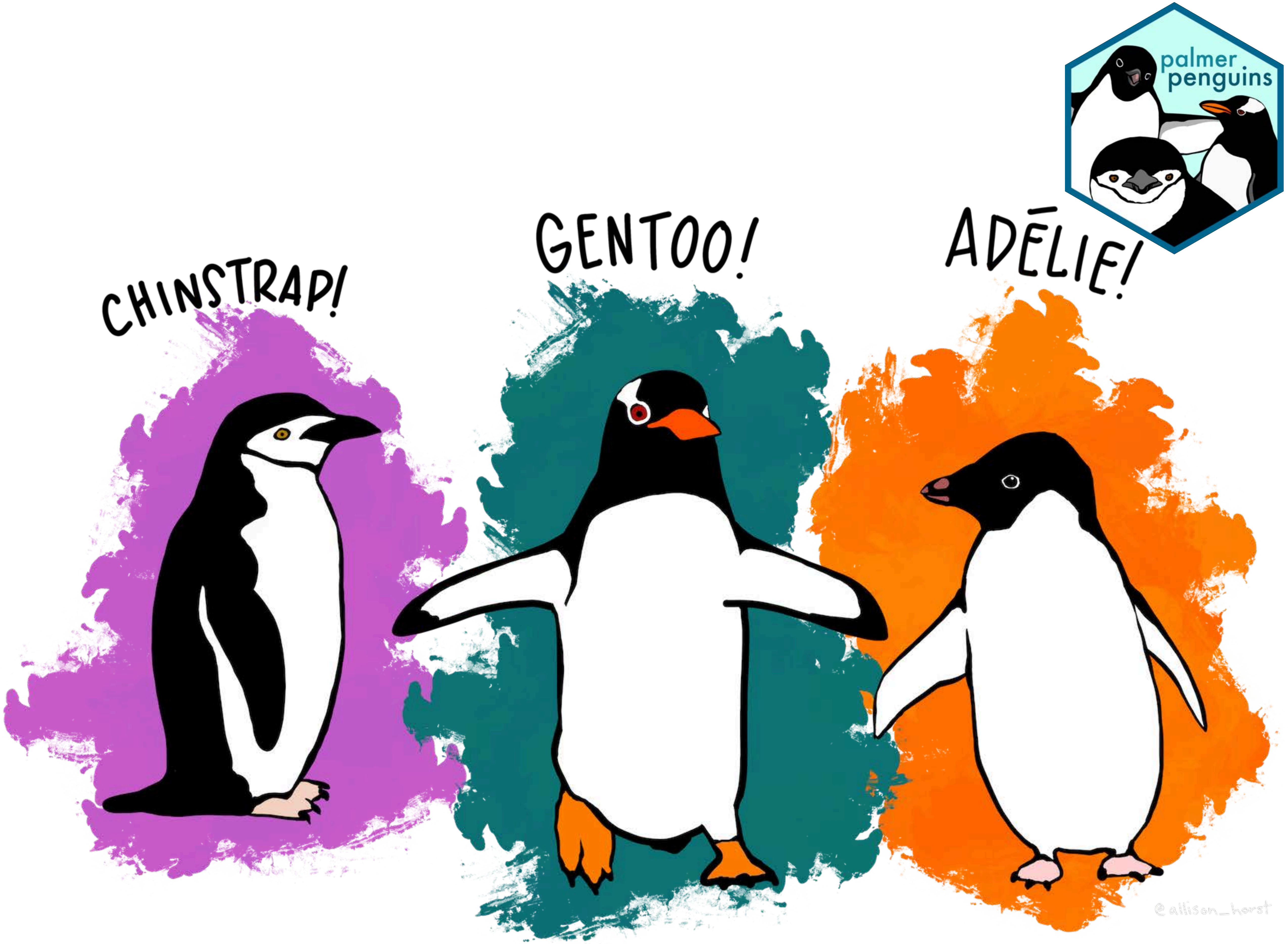
STATS AND POSITIONS

- Position adjustments are used to control the behavior of overlapping geoms.

- In most cases, the default is "identity" meaning, don't adjust position.

- What is the default position for `geom_bar()`?

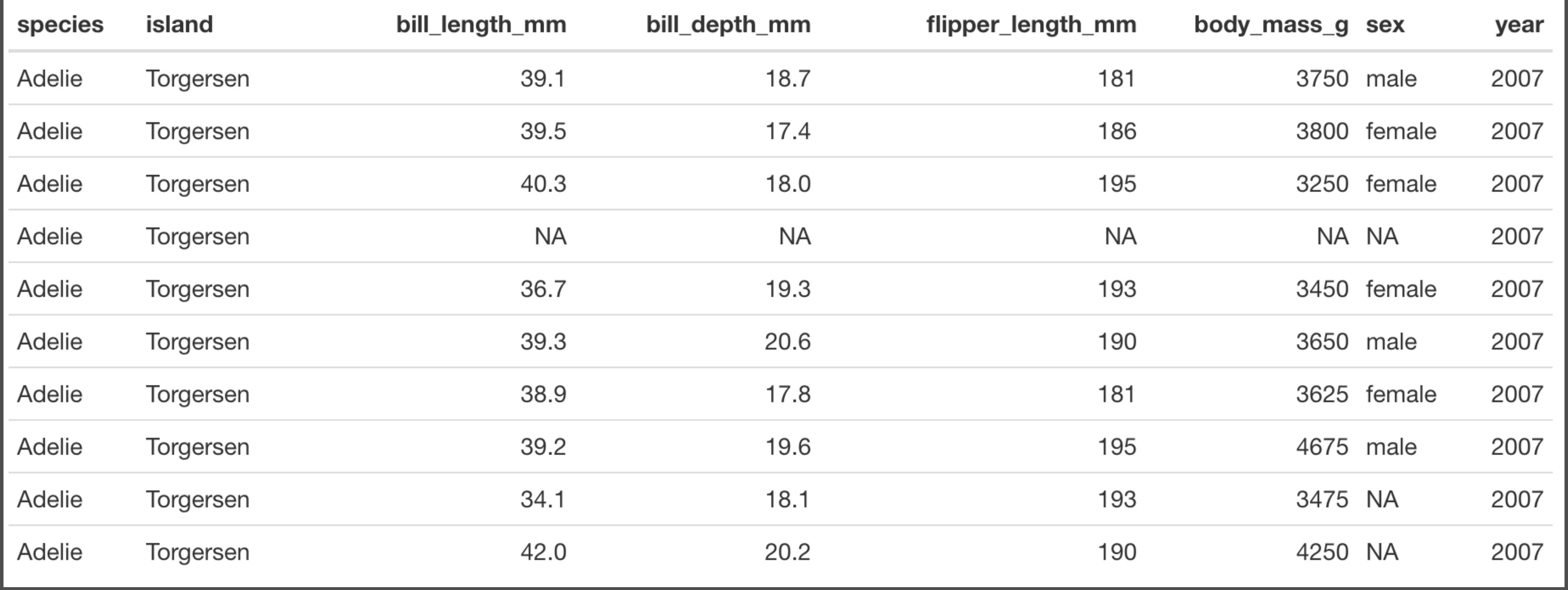
- Let's see what other positions we can use…



```
install.packages("palmerpenguins")
```

YOUR TURN

- First 10 rows of the `penguins` data set:

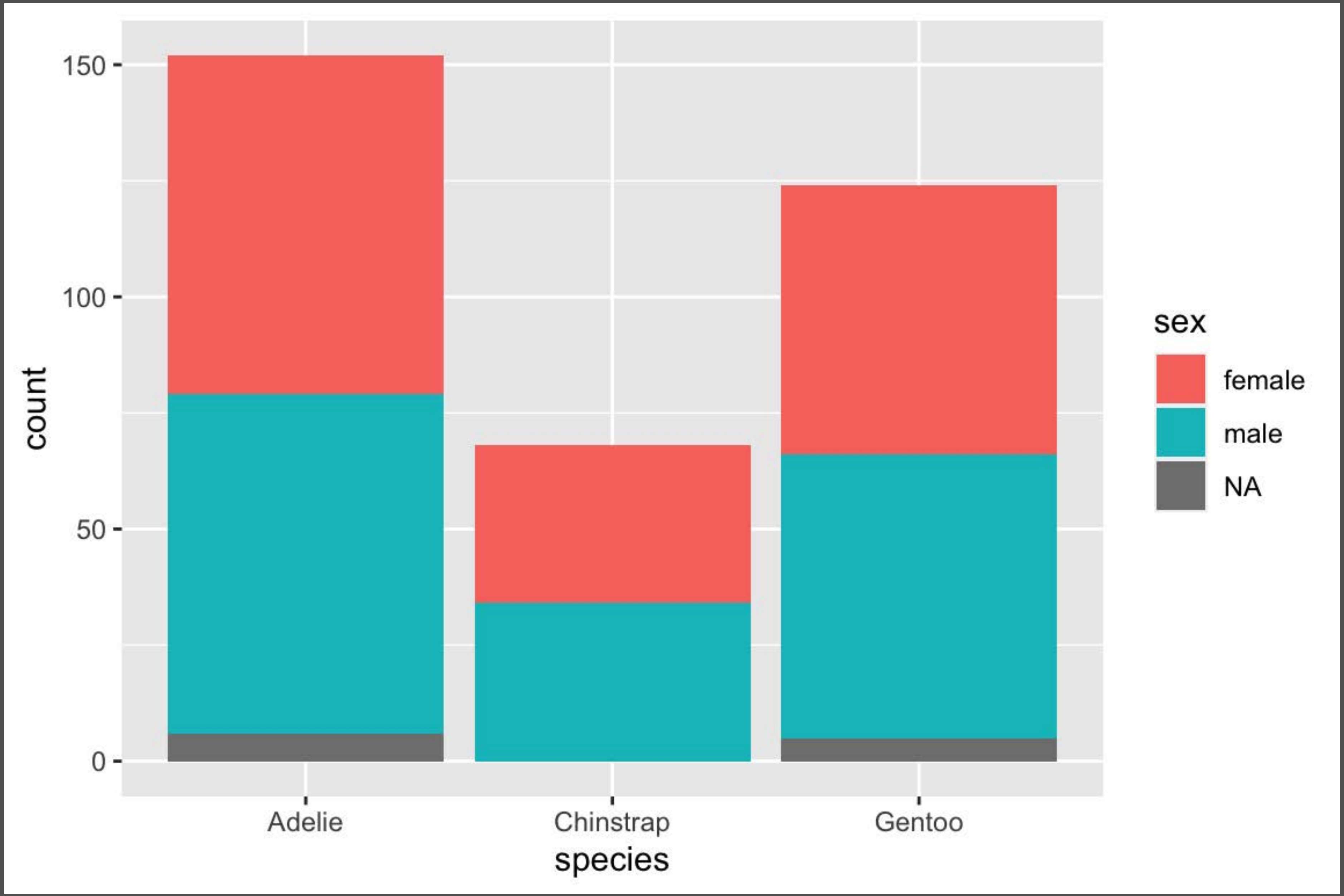


species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

- ***How many*** penguins are there of each sex in each species?

YOUR TURN

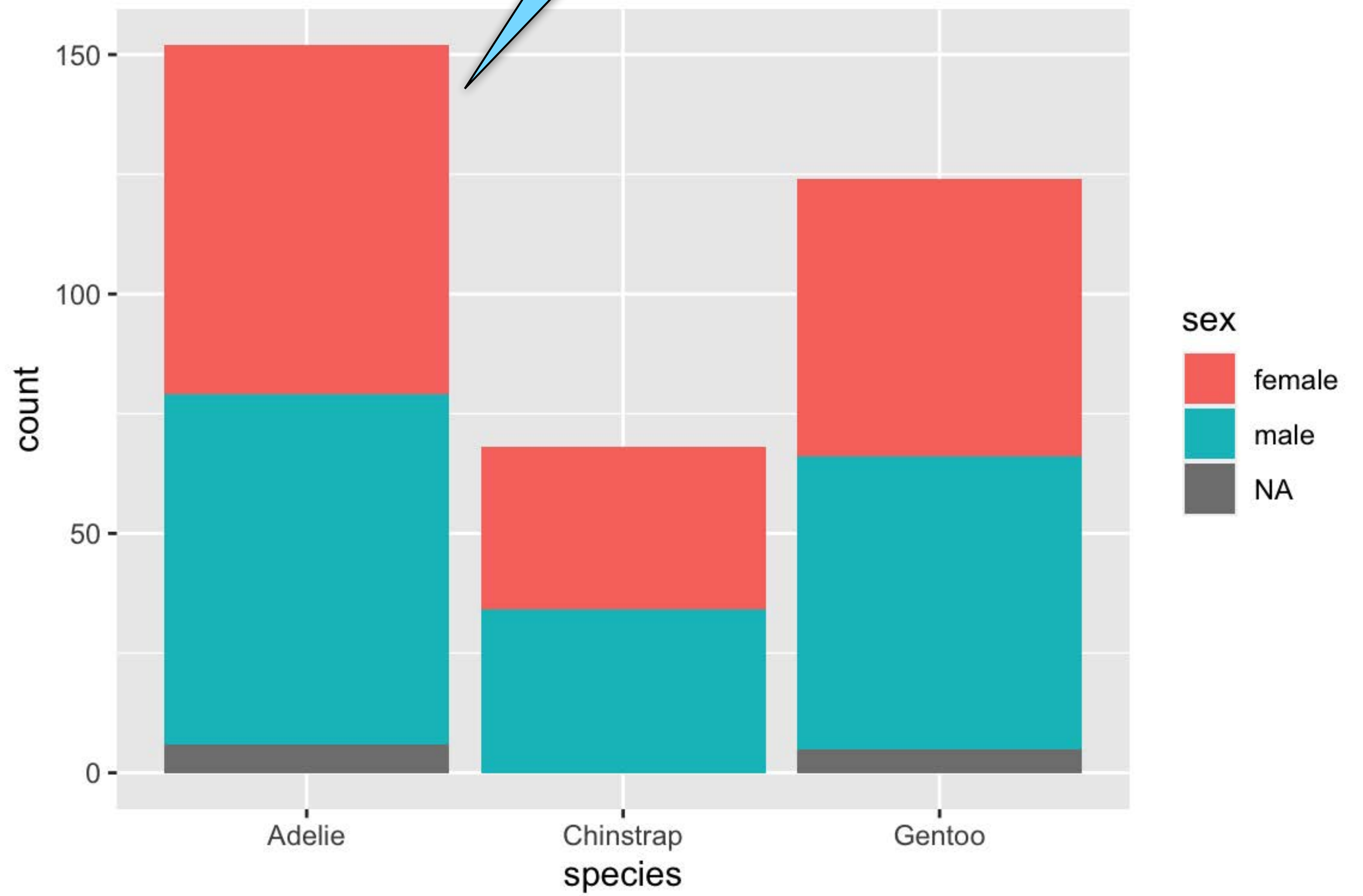
- Let's make a bar chart of the penguins to visualize this. Try to produce this plot:



2:00

YOUR TURN

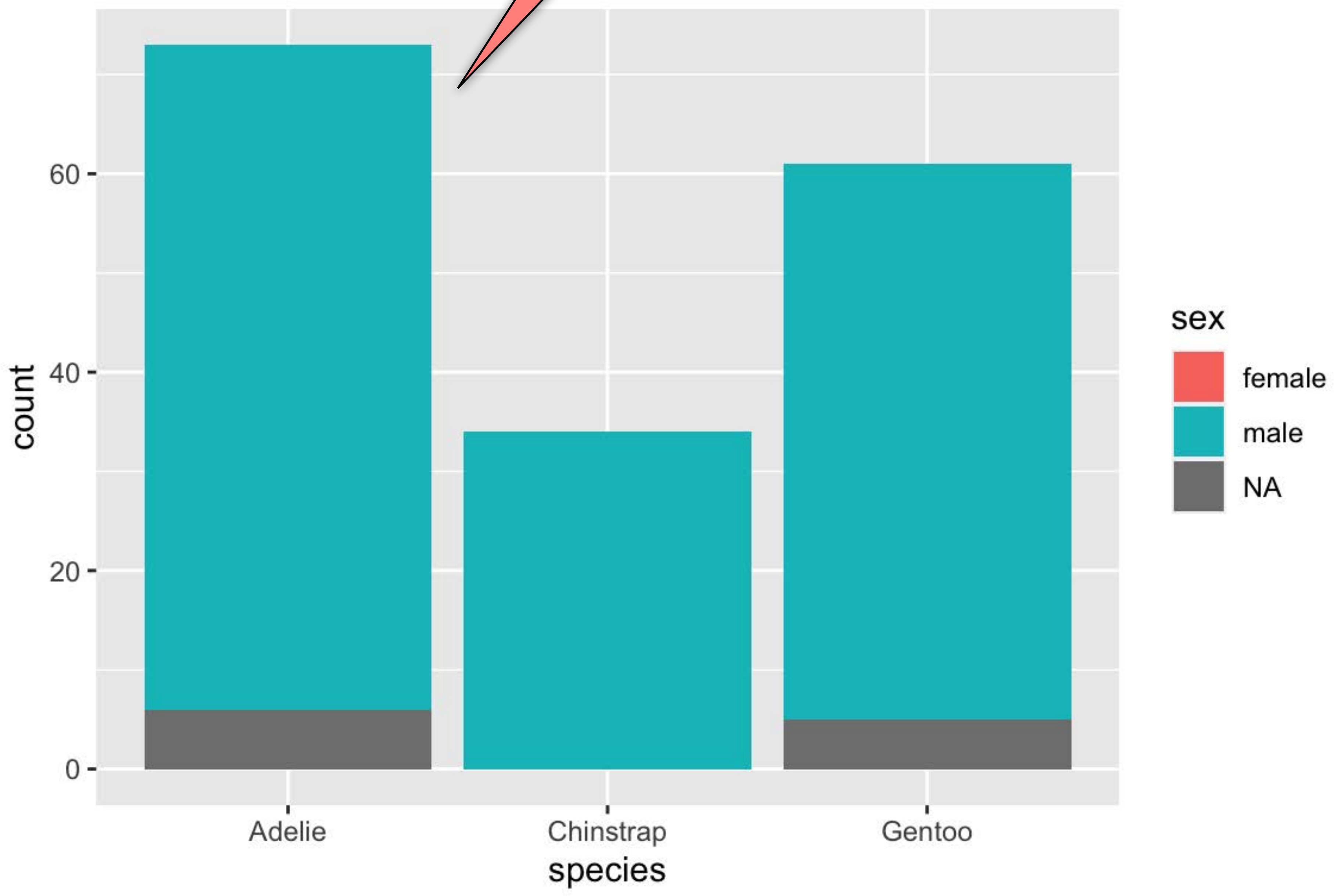
Instead of overlapping, bars are stacked



```
ggplot(penguins, aes(x = species, fill = sex)) +
  geom_bar()
```

YOUR TURN

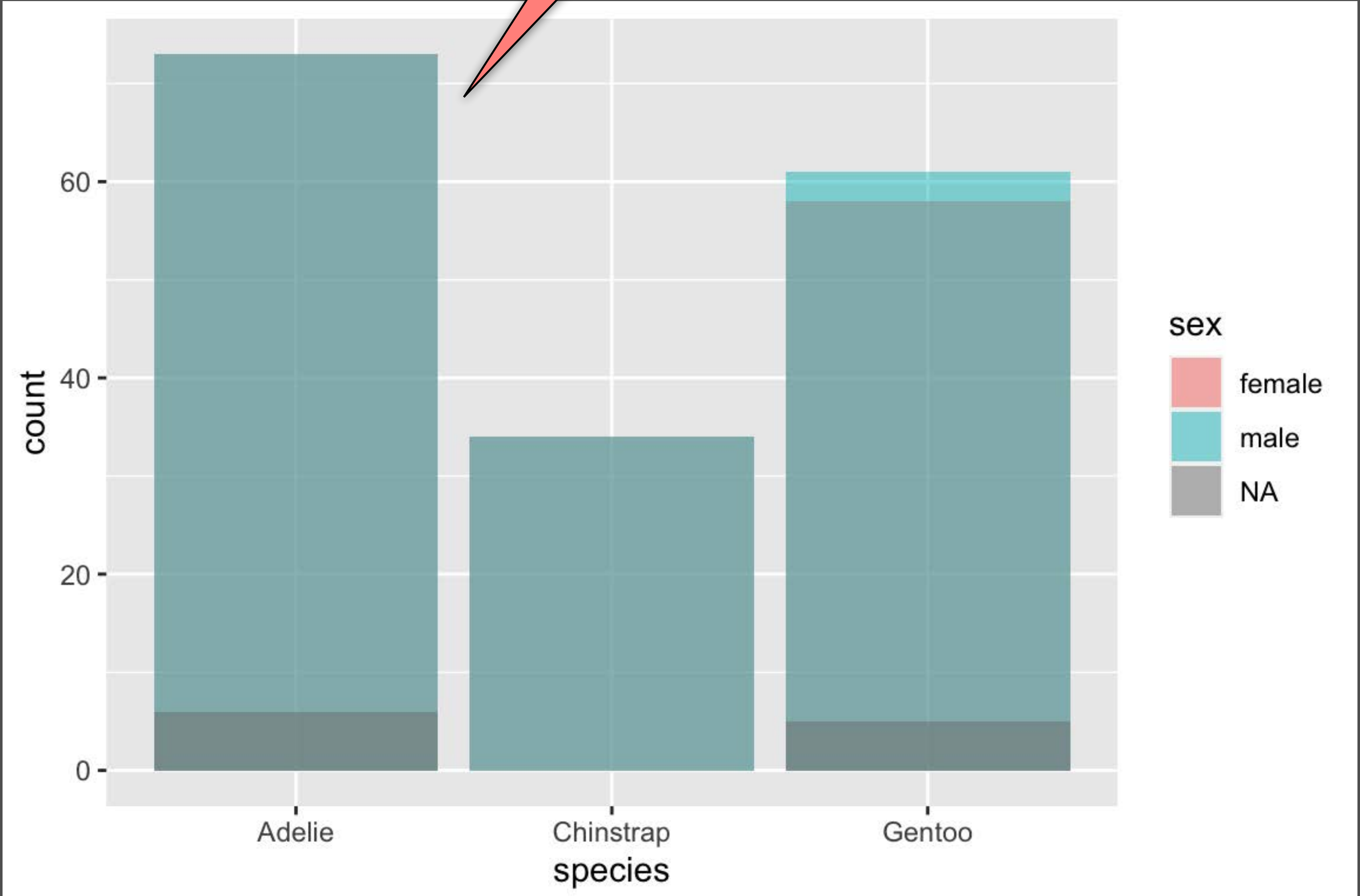
Bars are drawn over each other



```r
ggplot(penguins, aes(x = species, fill = sex)) +
  geom_bar(position = "identity")
```

YOUR TURN

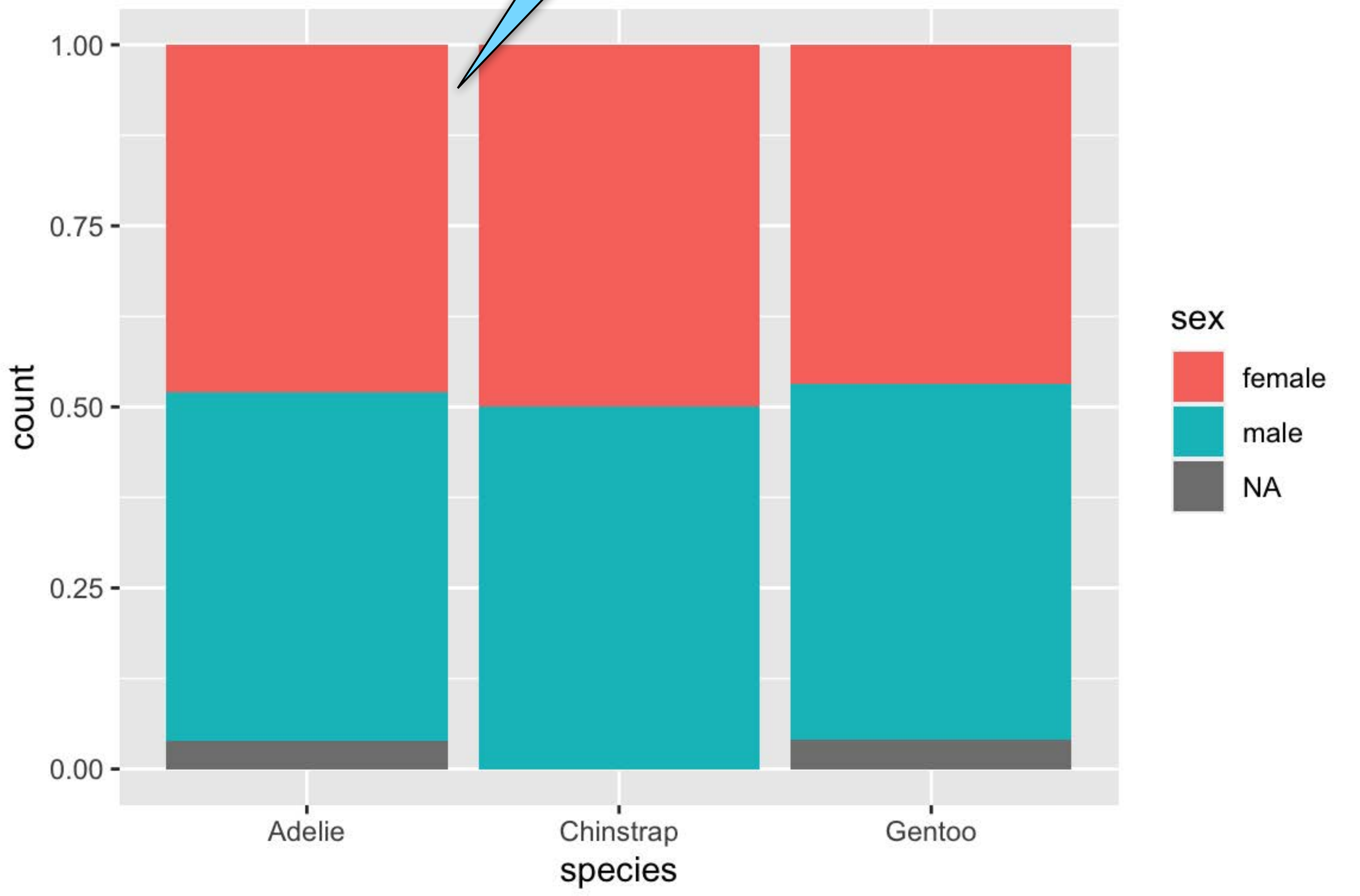
Bars are drawn over each other



```
ggplot(penguins, aes(x = species, fill = sex)) +
  geom_bar(position = "identity", alpha = 0.5)
```

YOUR TURN

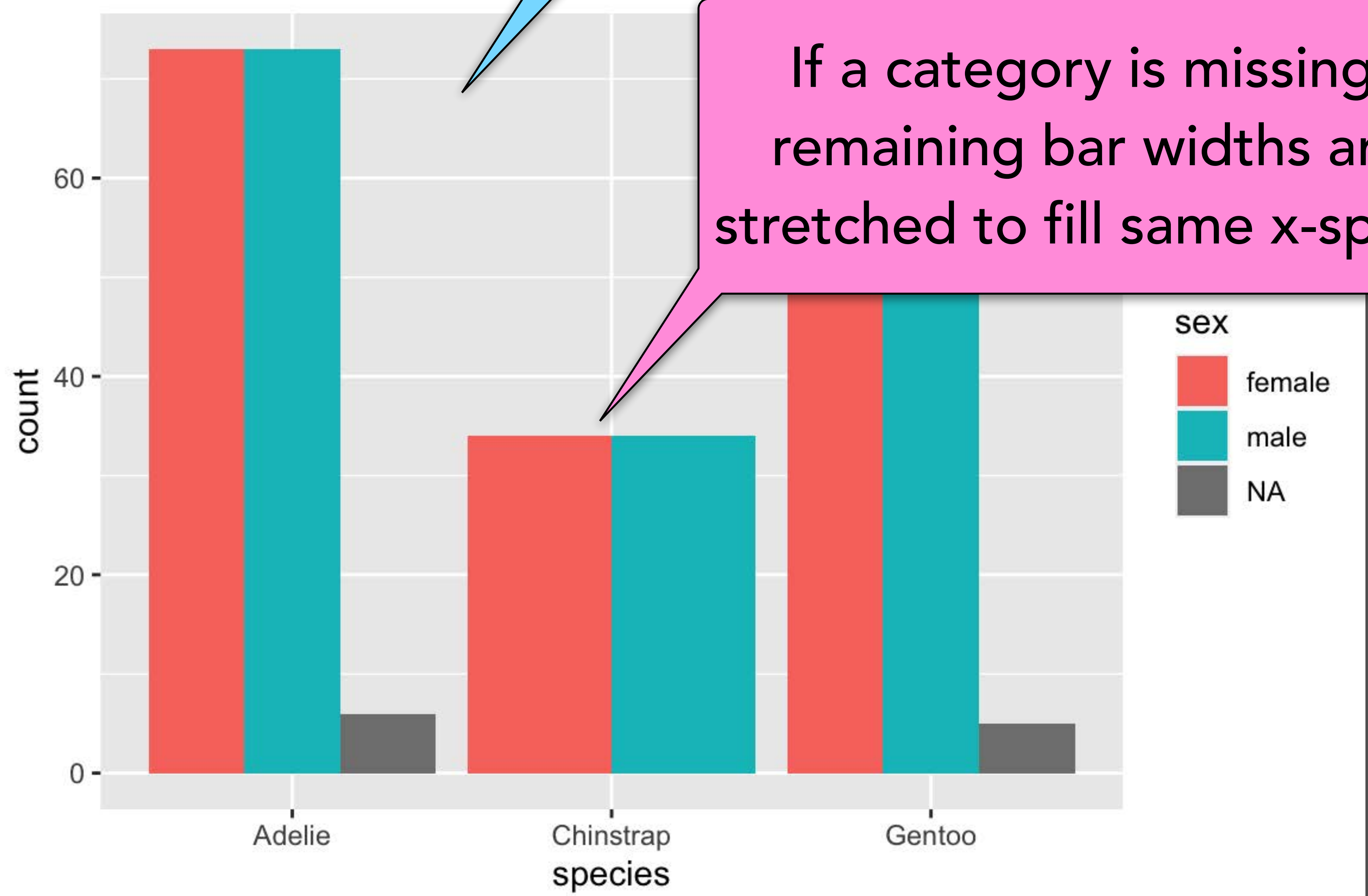
Bars "fill" the same y-space, showing proportions



```
ggplot(penguins, aes(x = species, fill = sex)) +
  geom_bar(position = "fill")
```

3:00

YOUR TURN



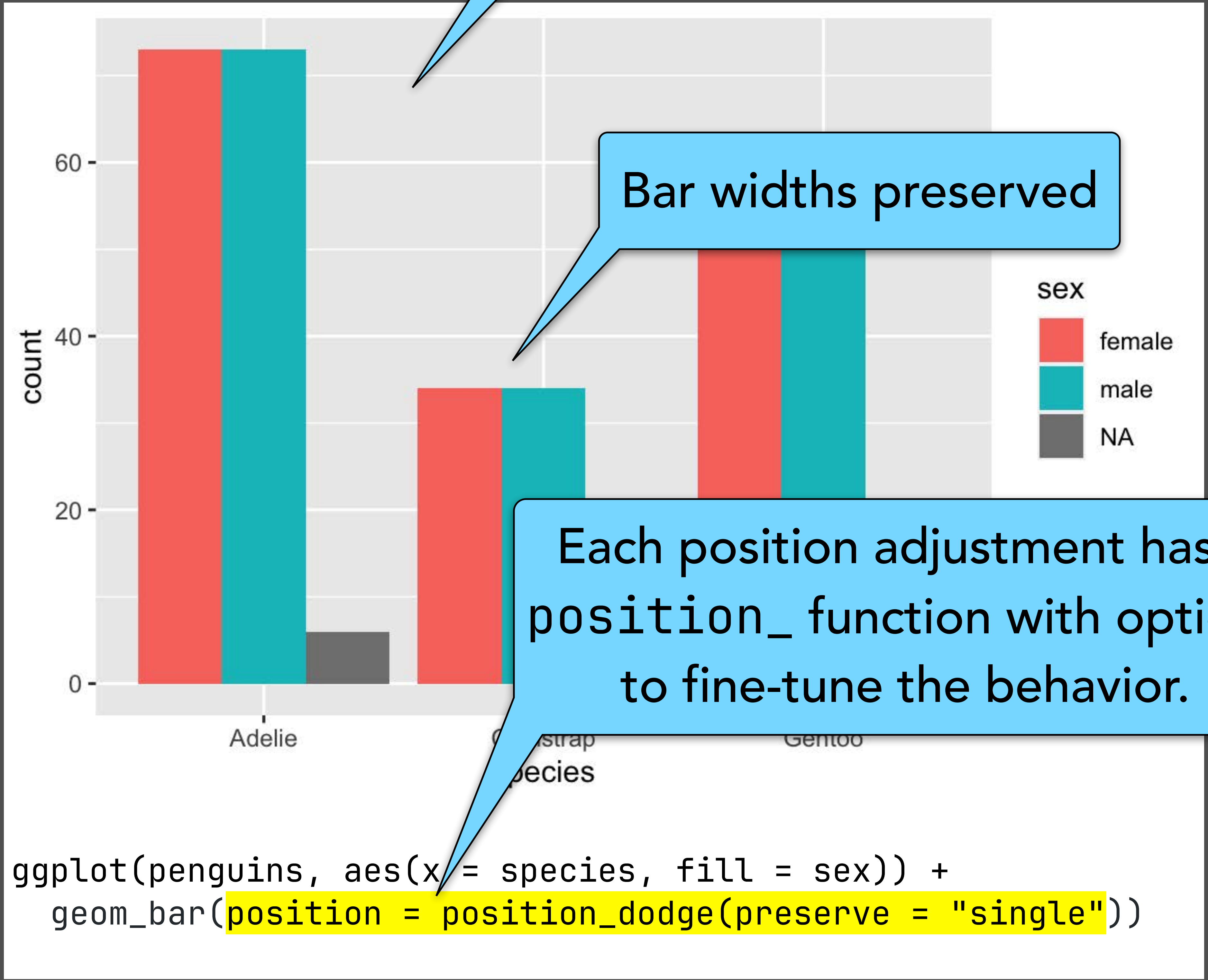
Bars dodge each other side-to-side

If a category is missing, remaining bar widths are stretched to fill same x-space

```
ggplot(penguins, aes(x = species, fill = sex)) +
  geom_bar(position = "dodge")
```

3:00

YOUR TURN



Bars dodge each other side-to-side

Bar widths preserved

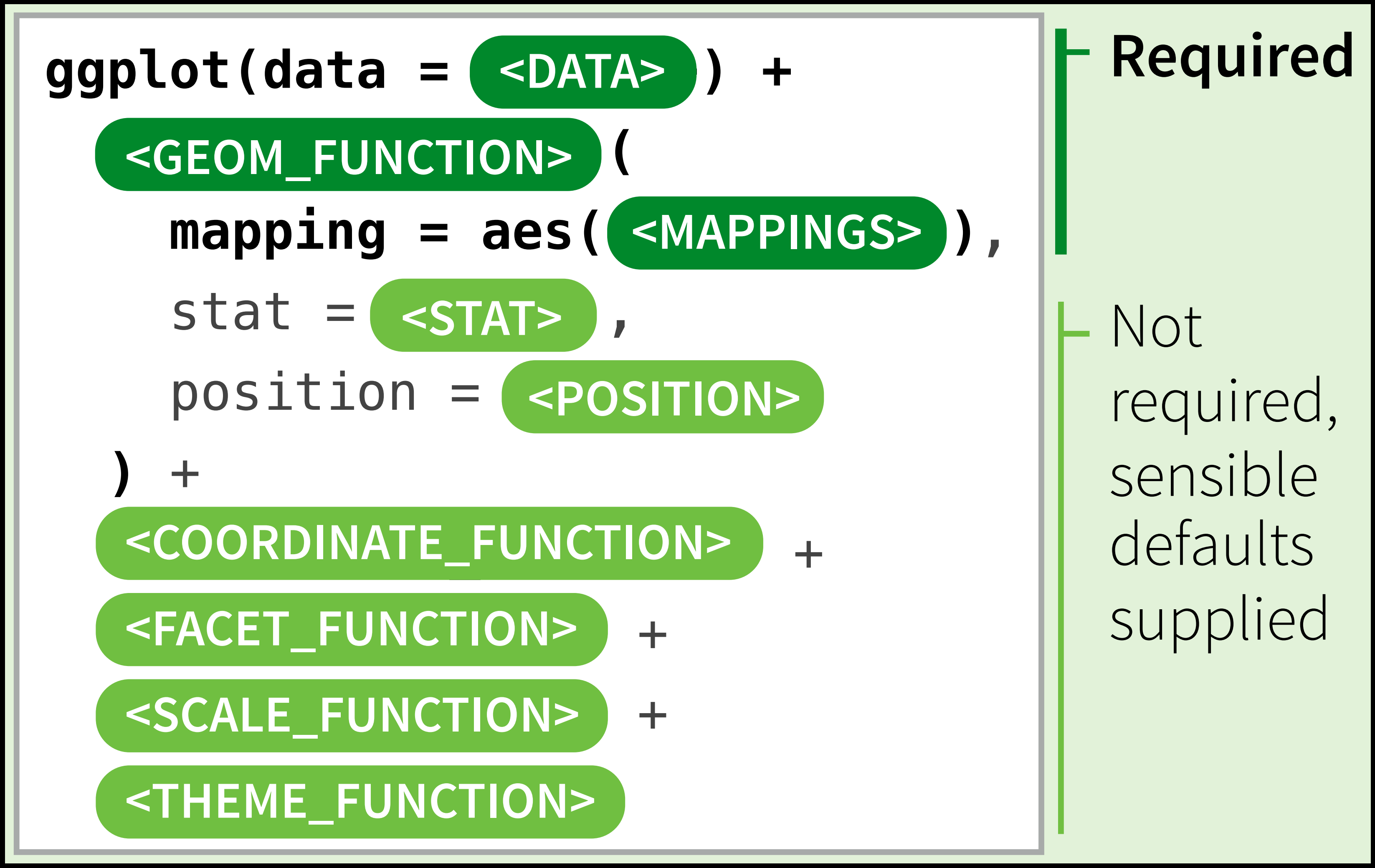
Each position adjustment has a `position_` function with options to fine-tune the behavior.

```
ggplot(penguins, aes(x = species, fill = sex)) +
  geom_bar(position = position_dodge(preserve = "single"))
```

3:00

WHAT ELSE?

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes



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

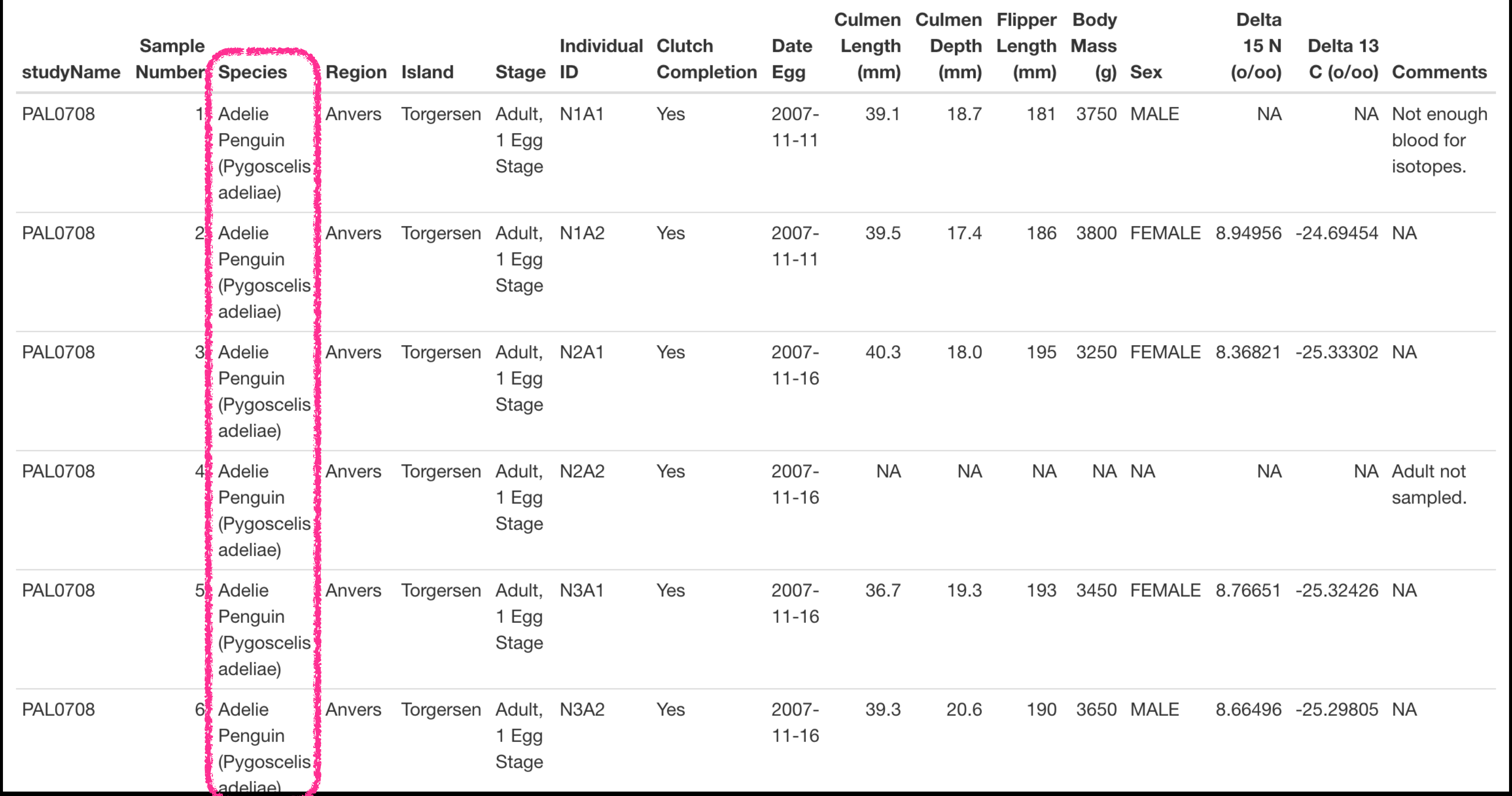
COORDINATE FUNCTIONS

- Coordinate functions determine how x and y in the plot are related to each other

- There are only a few situations when this is useful:
  - Flipping x and y axes with `coord_flip()`
  - Making circular plots with `coord_polar()`
  - Using a fixed ratio between x and y with `coord_fixed()`
  - Using certain map projections `coord_map()`

EXAMPLE: LONG CATEGORY LABELS

- First 6 rows of the `penguins_raw` data set:



studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)	Culmen Depth (mm)	Flipper Length (mm)	Body Mass (g)	Sex	Delta 15 N (o/oo)	Delta 13 C (o/oo)	Comments
PAL0708	1	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N1A1	Yes	2007-11-11	39.1	18.7	181	3750	MALE	NA	NA	Not enough blood for isotopes.
PAL0708	2	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N1A2	Yes	2007-11-11	39.5	17.4	186	3800	FEMALE	8.94956	-24.69454	NA
PAL0708	3	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N2A1	Yes	2007-11-16	40.3	18.0	195	3250	FEMALE	8.36821	-25.33302	NA
PAL0708	4	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N2A2	Yes	2007-11-16	NA	NA	NA	NA	NA	NA	NA	Adult not sampled.
PAL0708	5	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N3A1	Yes	2007-11-16	36.7	19.3	193	3450	FEMALE	8.76651	-25.32426	NA
PAL0708	6	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N3A2	Yes	2007-11-16	39.3	20.6	190	3650	MALE	8.66496	-25.29805	NA

YOUR TURN

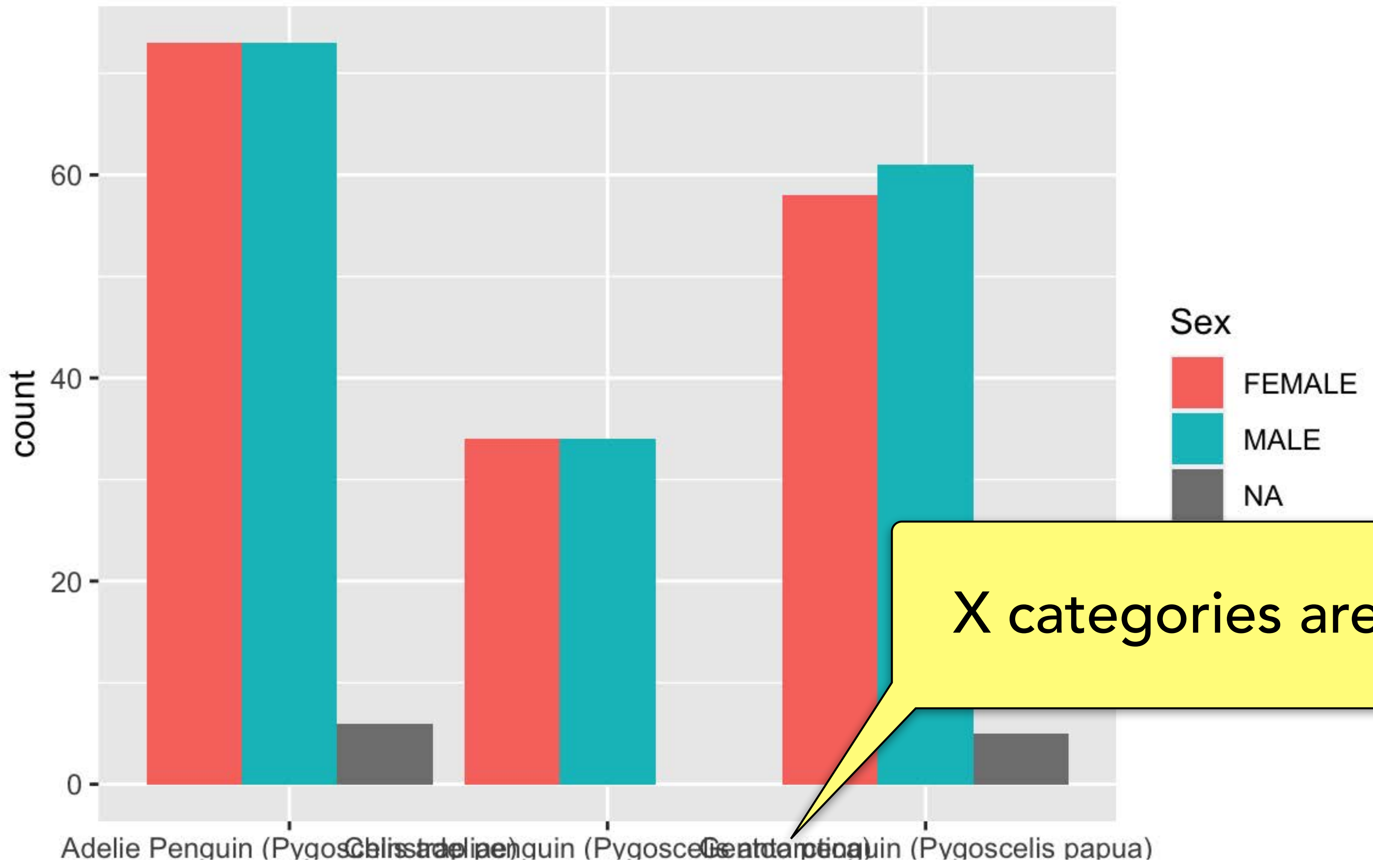
- Make the same kind of bar chart as before with the `penguins_raw` dataset.

- Like our last plot, map x to the Species column, map fill to the Sex column, and use dodge position adjustment.

- What problem does this plot have?

2:00

YOUR TURN

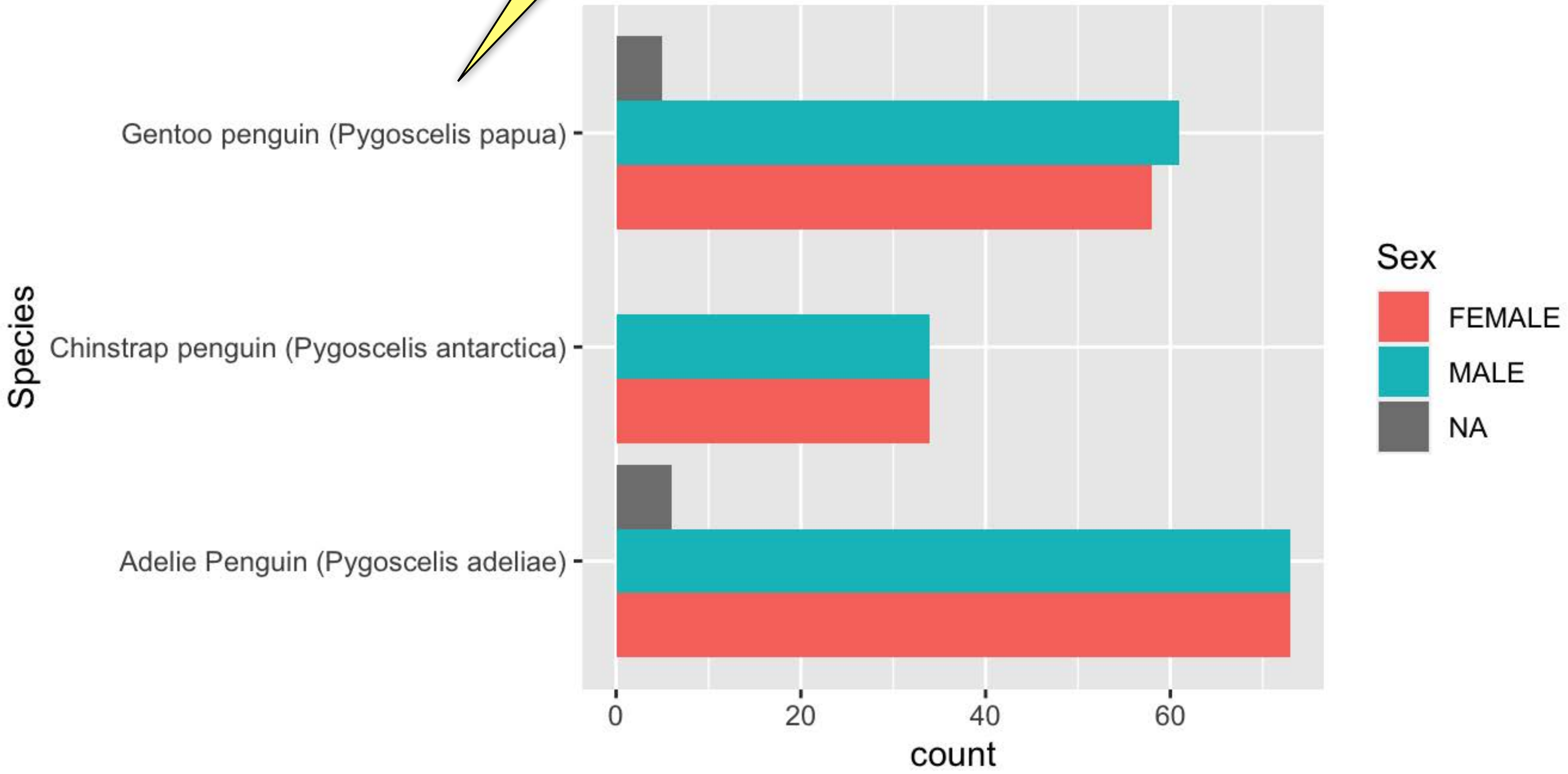


X categories are too long

```
ggplot(penguins_raw, aes(x = Species, fill = Sex)) +
  geom_bar(position = position_dodge(preserve = "single"))
```

YOUR TURN

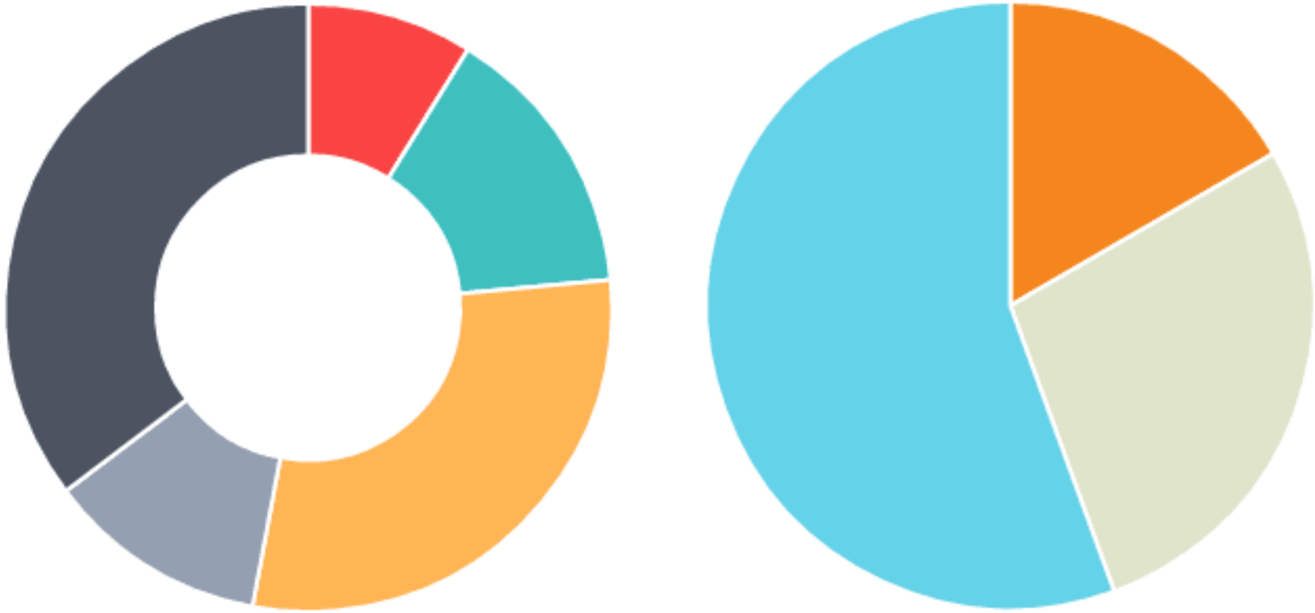
Long categories fit better with flipped axes



```
ggplot(penguins_raw, aes(x = Species, fill = Sex)) +
  geom_bar(position = position_dodge(preserve = "single")) +
  coord_flip()
```

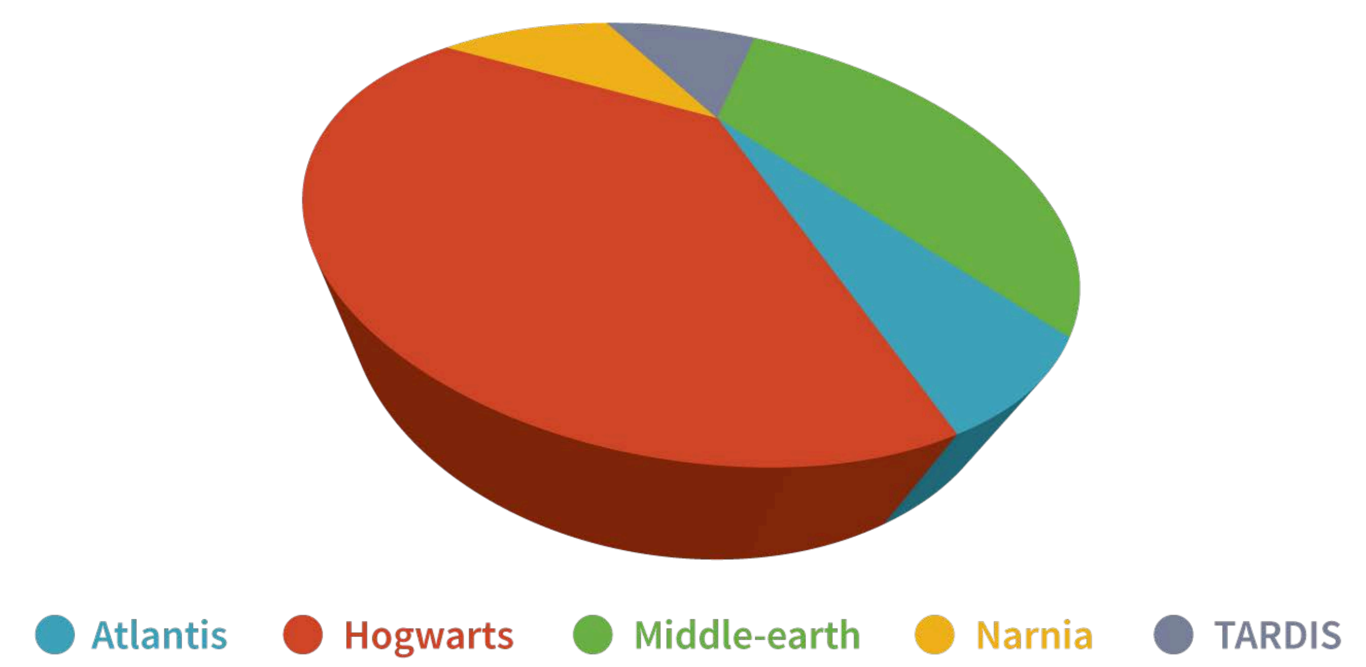
PIE CHARTS AND DONUT CHARTS

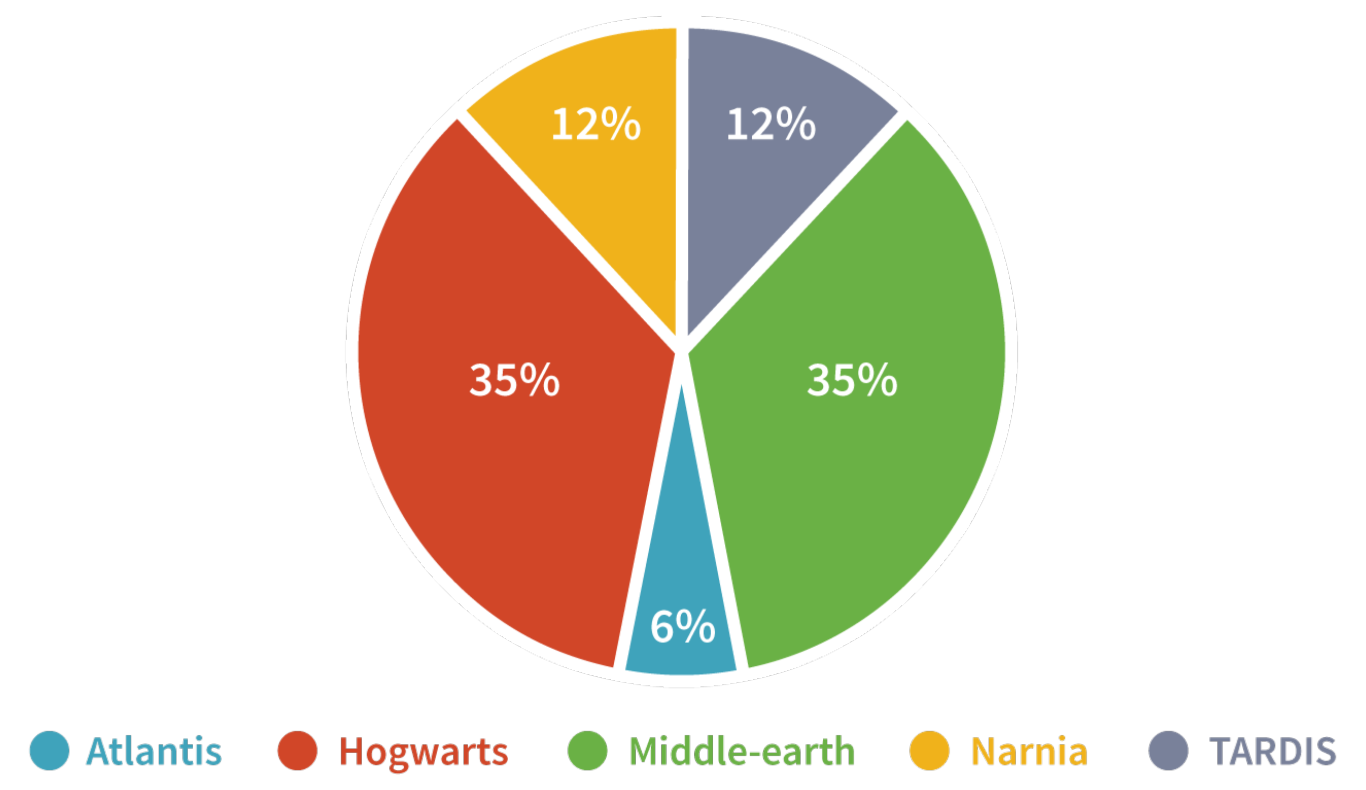
- Also used with categorical variables
- Probably most misused type of graph



PIE CHARTS AND DONUT CHARTS

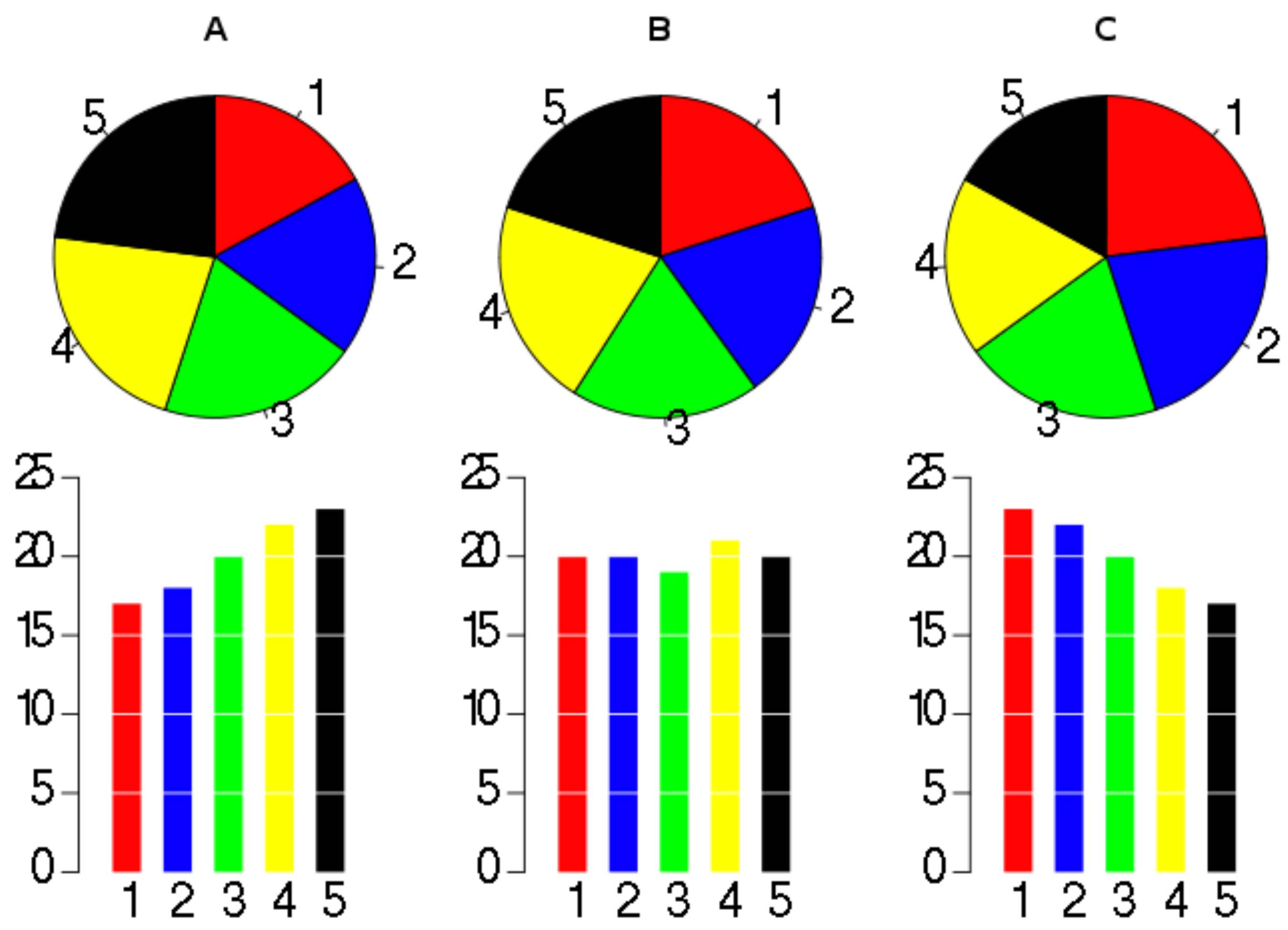
- Also used with categorical variables
- Probably most misused type of graph
- Perceptual problems—no 3D!





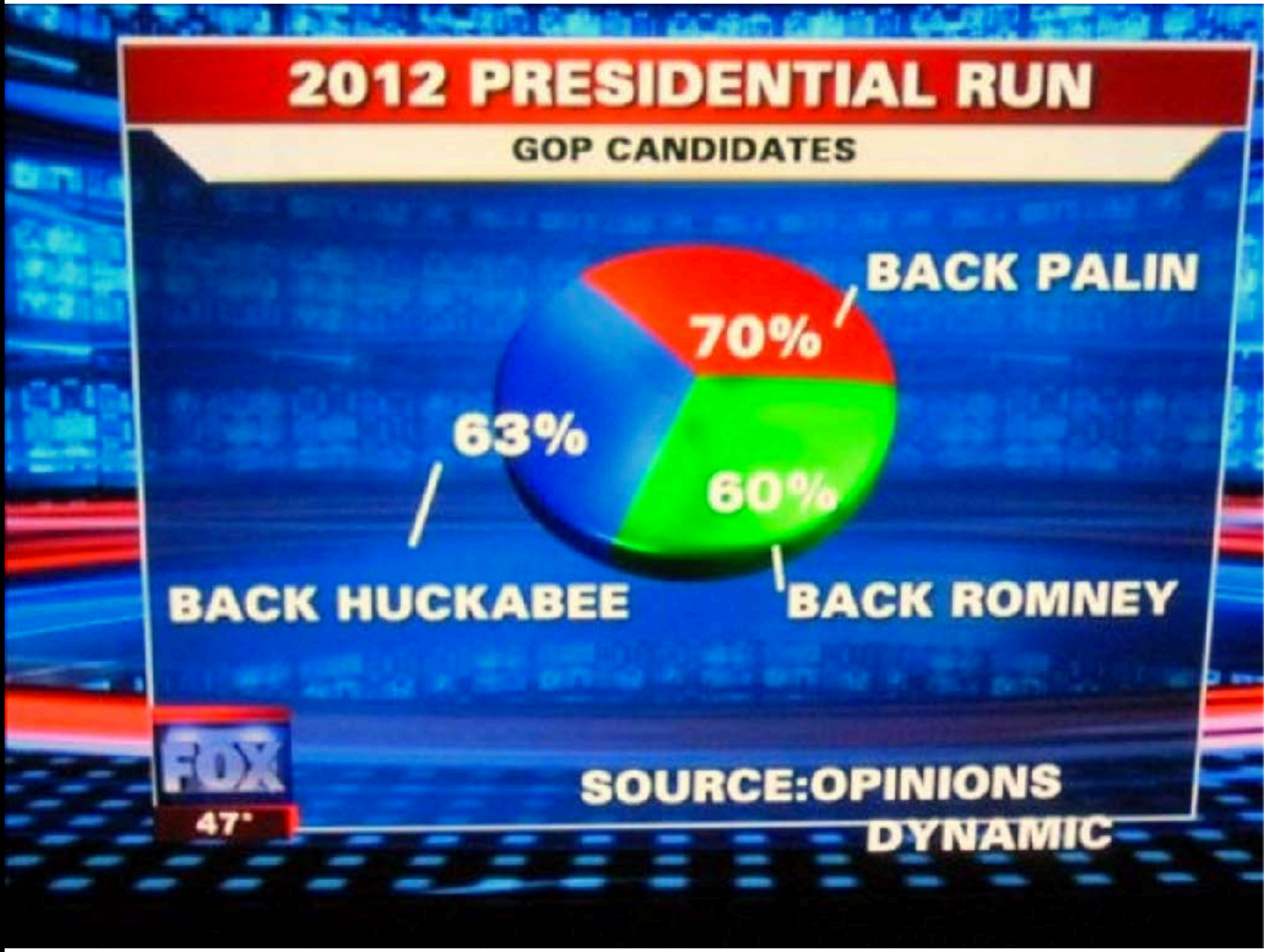
PIE CHARTS AND DONUT CHARTS

- Also used with categorical variables
- Probably most misused type of graph
- Perceptual problems—no 3D!



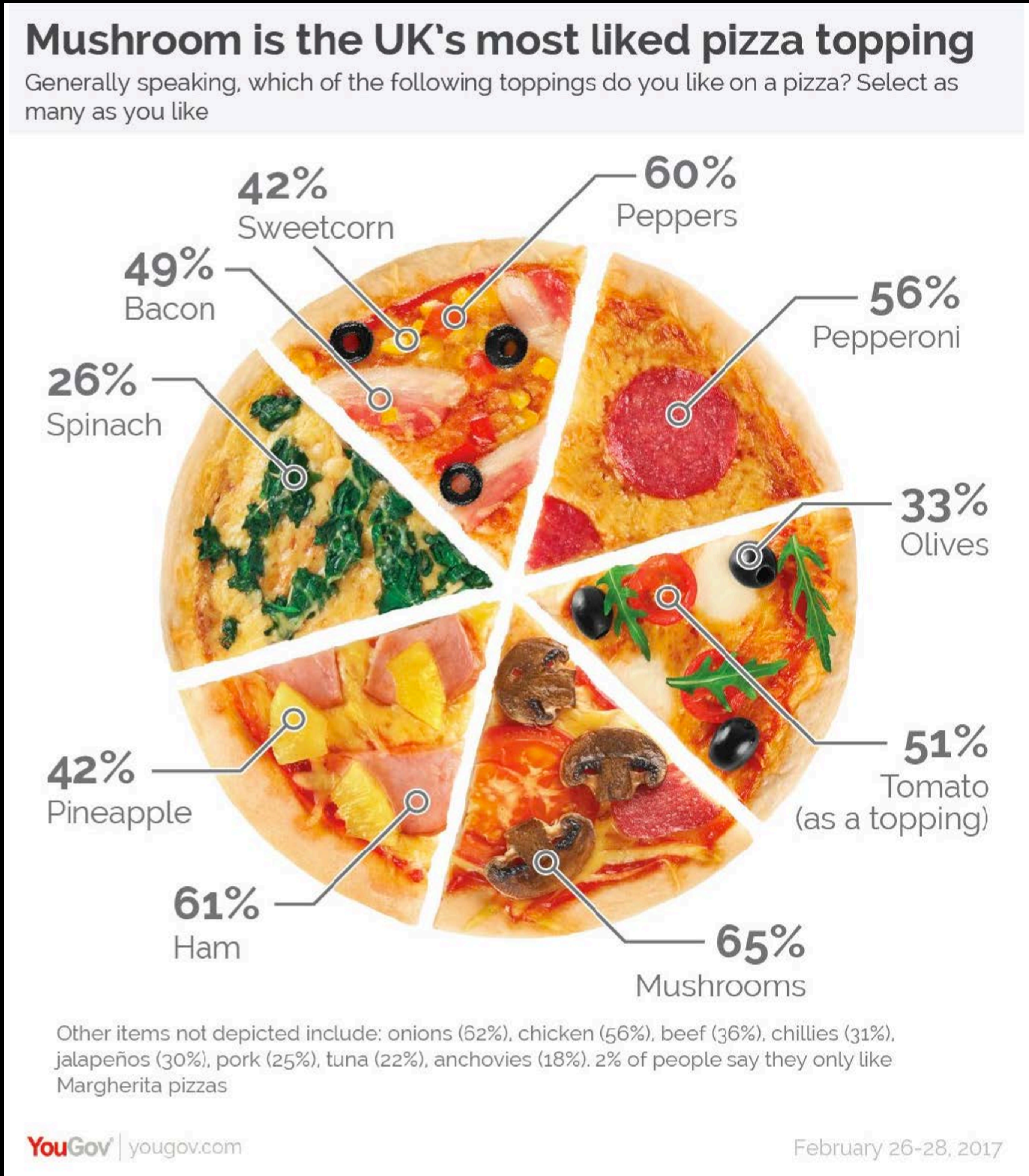
PIE CHARTS AND DONUT CHARTS

- Also used with categorical variables
- Probably most misused type of graph
- Perceptual problems—no 3D!
- Can pie charts be used effectively? Yes, in limited cases, when:
  - The parts sum to a meaningful whole



PIE CHARTS AND DONUT CHARTS

- Also used with categorical variables
- Probably most misused type of graph
- Perceptual problems—no 3D!
- Can pie charts be used effectively? Yes, in limited cases, when:
  - The parts sum to a meaningful whole



Mushroom is the UK's most liked pizza topping

Generally speaking, which of the following toppings do you like on a pizza? Select as many as you like

Other items not depicted include: onions (62%), chicken (56%), beef (36%), chillies (31%), jalapeños (30%), pork (25%), tuna (22%), anchovies (18%). 2% of people say they only like Margherita pizzas

YouGov | yougov.com

February 26-28, 2017

PIE CHARTS AND DONUT CHARTS

- Also used with categorical variables

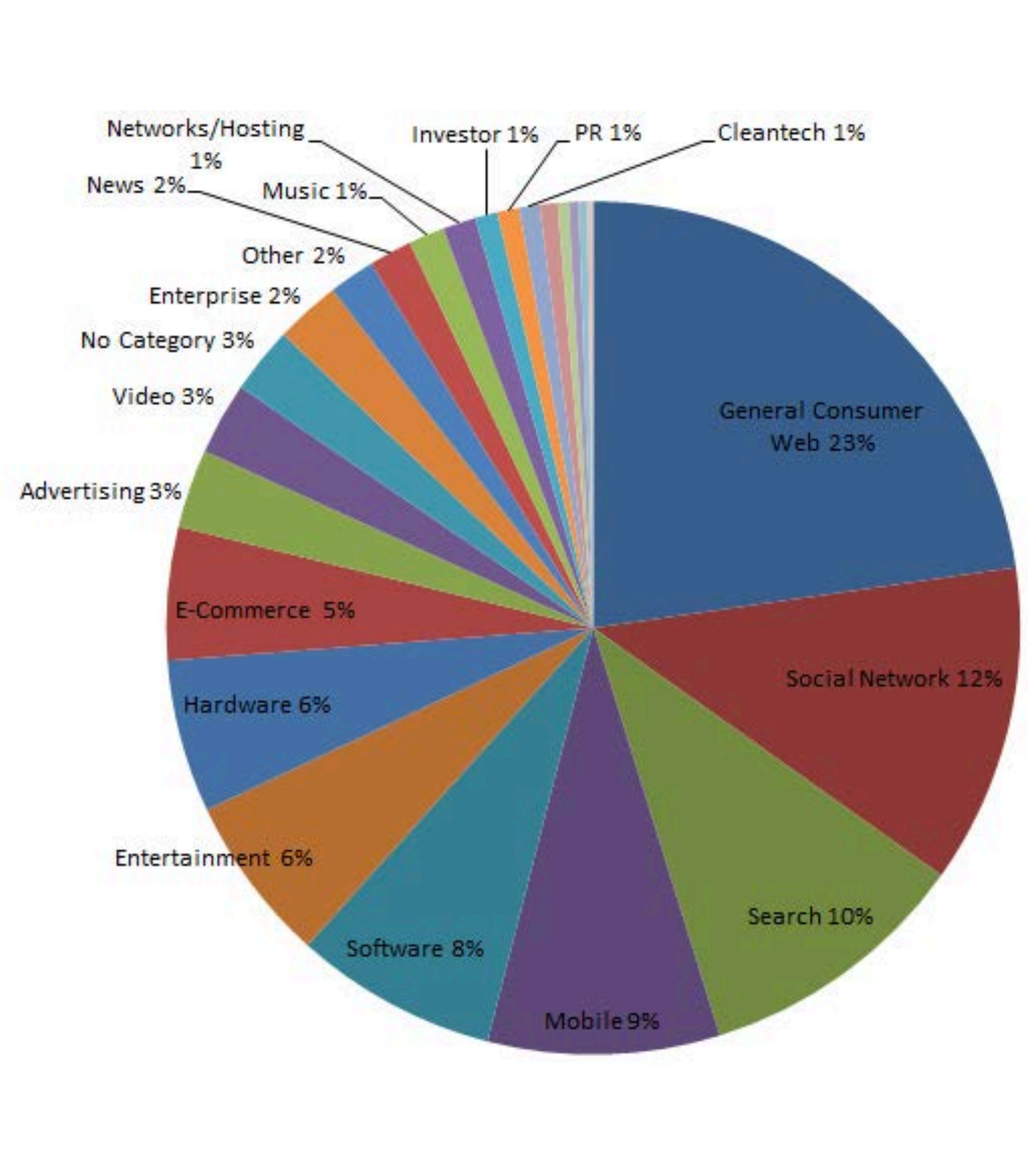
- Probably most misused type of graph

- Perceptual problems—no 3D!

- Can pie charts be used effectively? Yes, in limited cases, when:

  - The parts sum to a meaningful whole

  - There are few categories (≤3)



YOUR TURN

- I mentioned that pie charts are used for categorical data, just like bar charts.

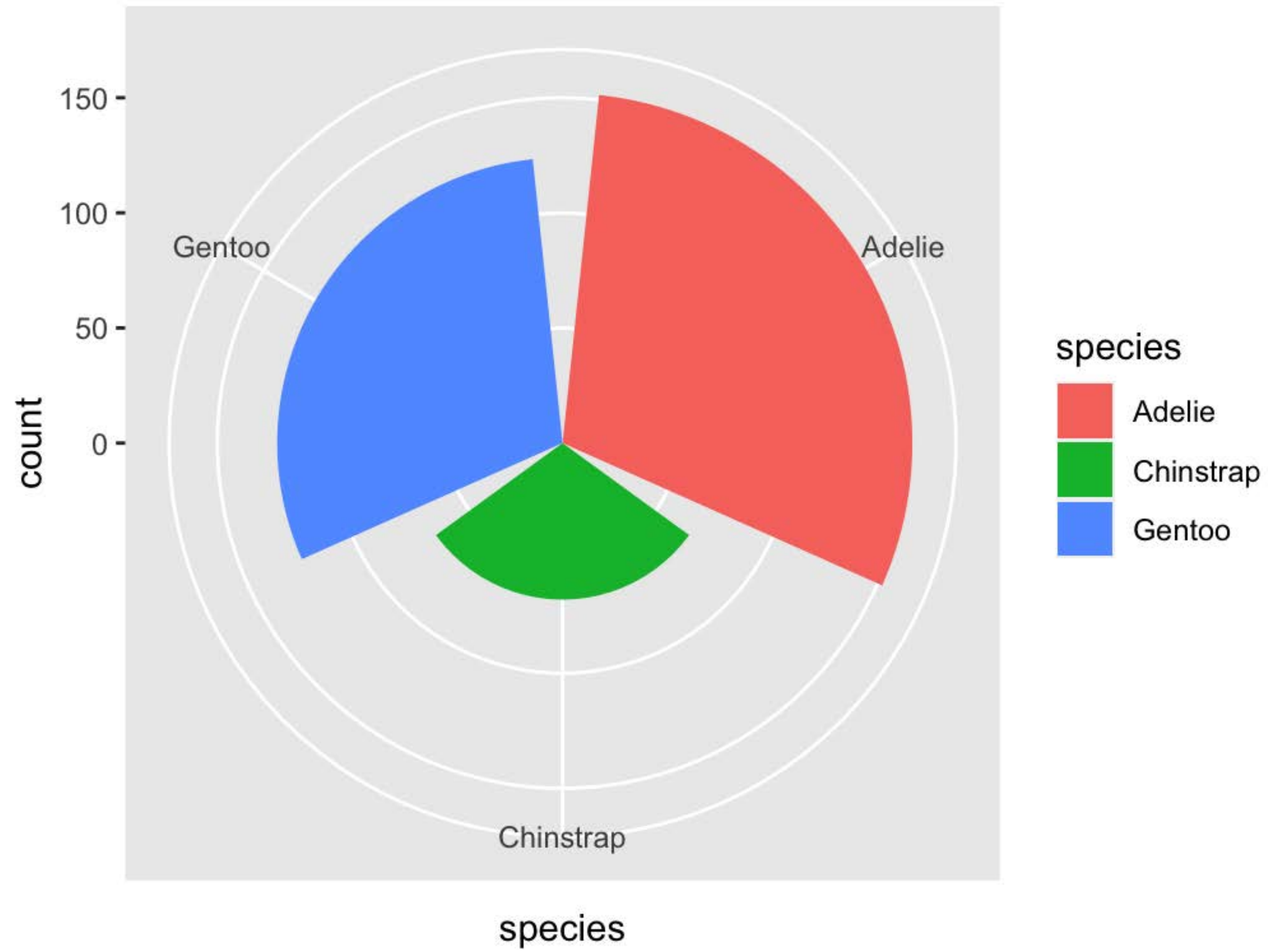
- I also mentioned that `coord_polar()` is used for making circular plots (like pies and donuts).

- What happens if we take a penguins bar plot and just add `coord_polar()`?

```
ggplot(penguins, aes(x = species, fill = species)) +
  geom_bar(position = "dodge") +
  coord_polar()
```

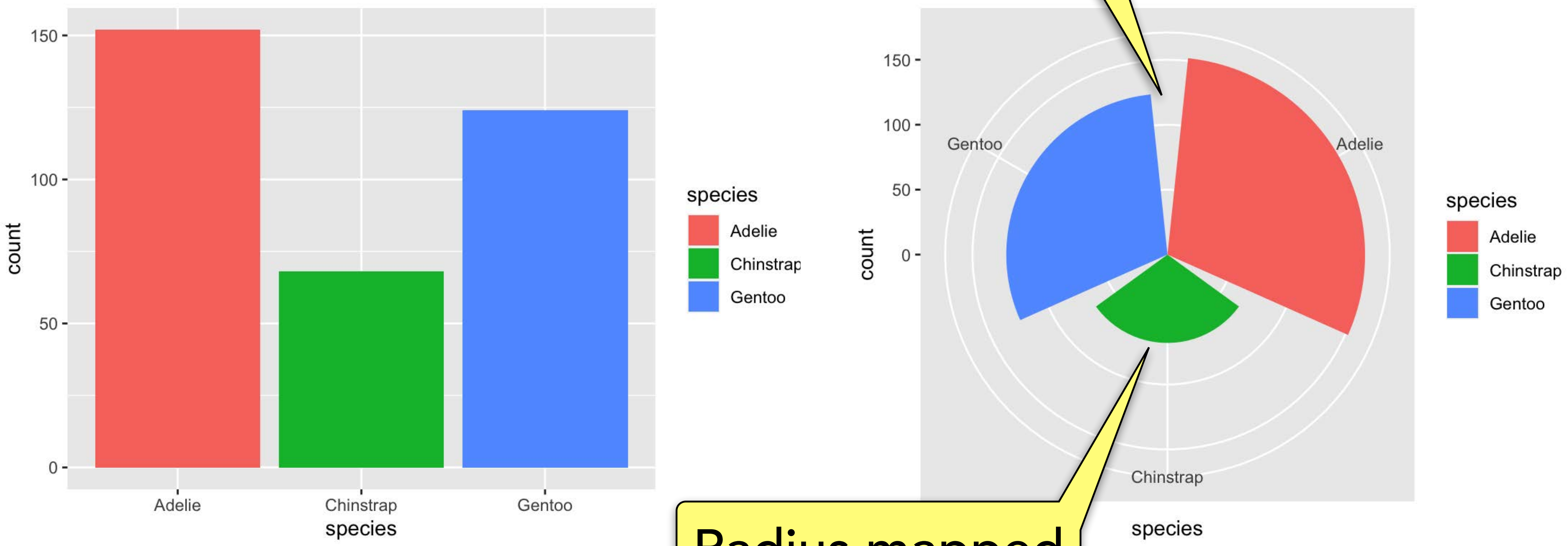
2:00

NOPE



```
ggplot(penguins, aes(x = species, fill = species)) +
  geom_bar(position = "dodge") +
  coord_polar()
```

WHAT HAPPENED?

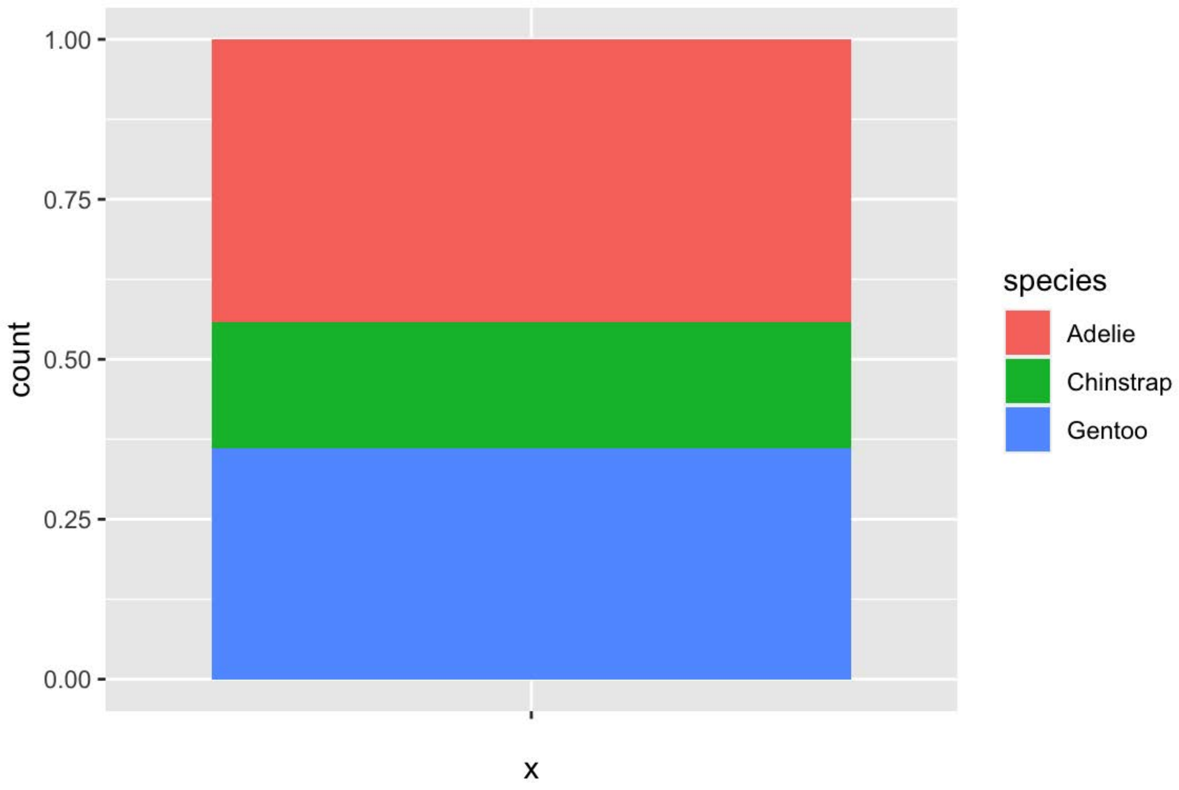


Angle (theta) mapped to x variable

Radius mapped to y (count)

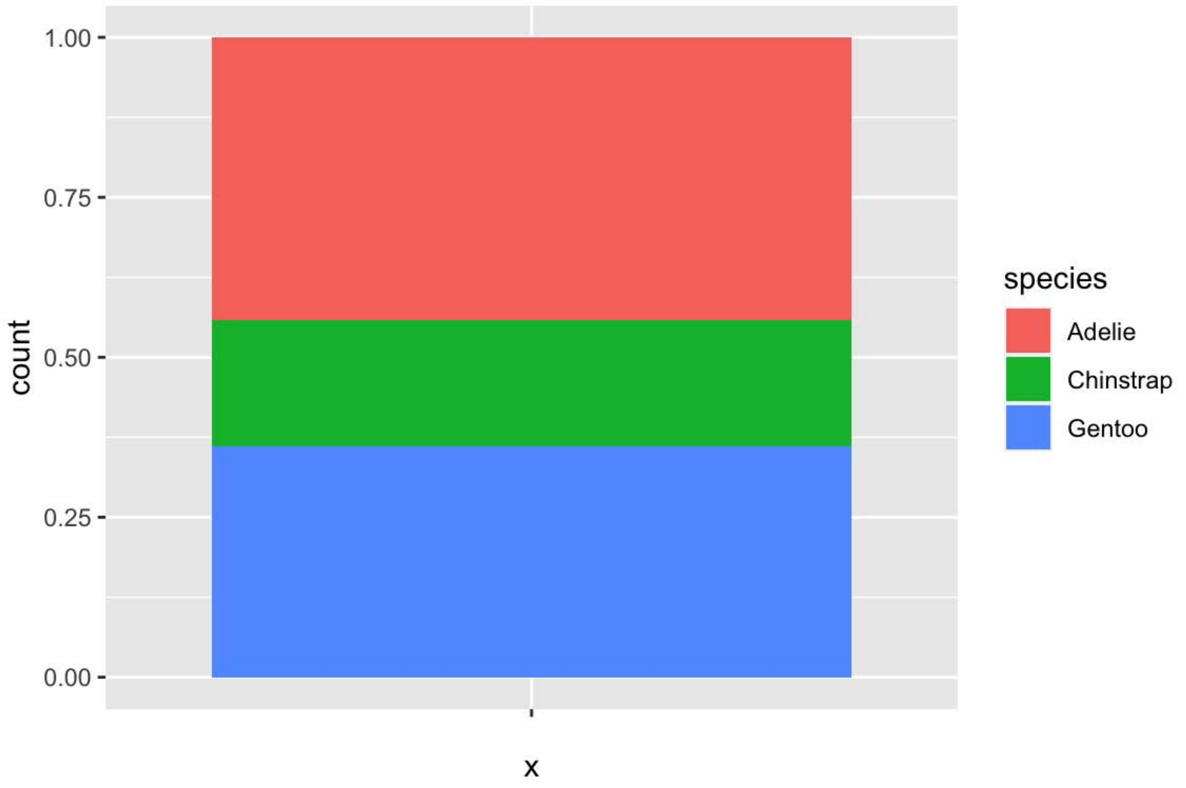
```
ggplot(penguins, aes(x = species, fill = species)) +
  geom_bar(position = "dodge") +
  coord_polar()
```

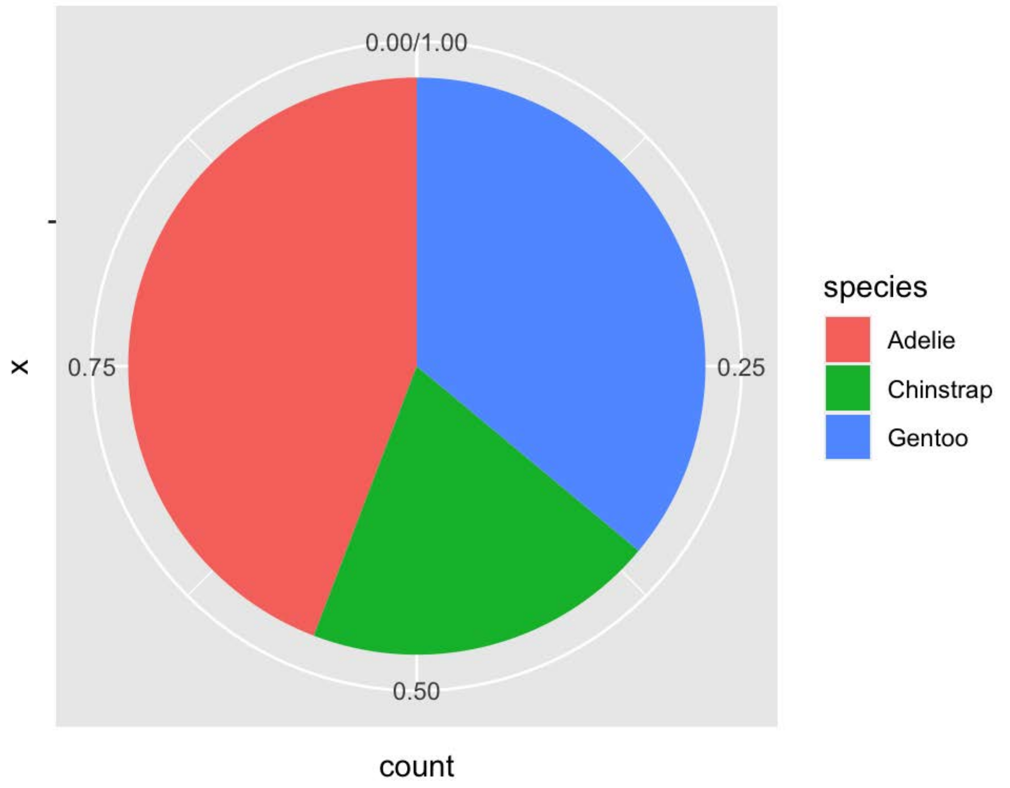
HOW TO MAKE A PIE



```
ggplot(penguins, aes(x = "", fill = species)) +
  geom_bar(position = "fill")
```

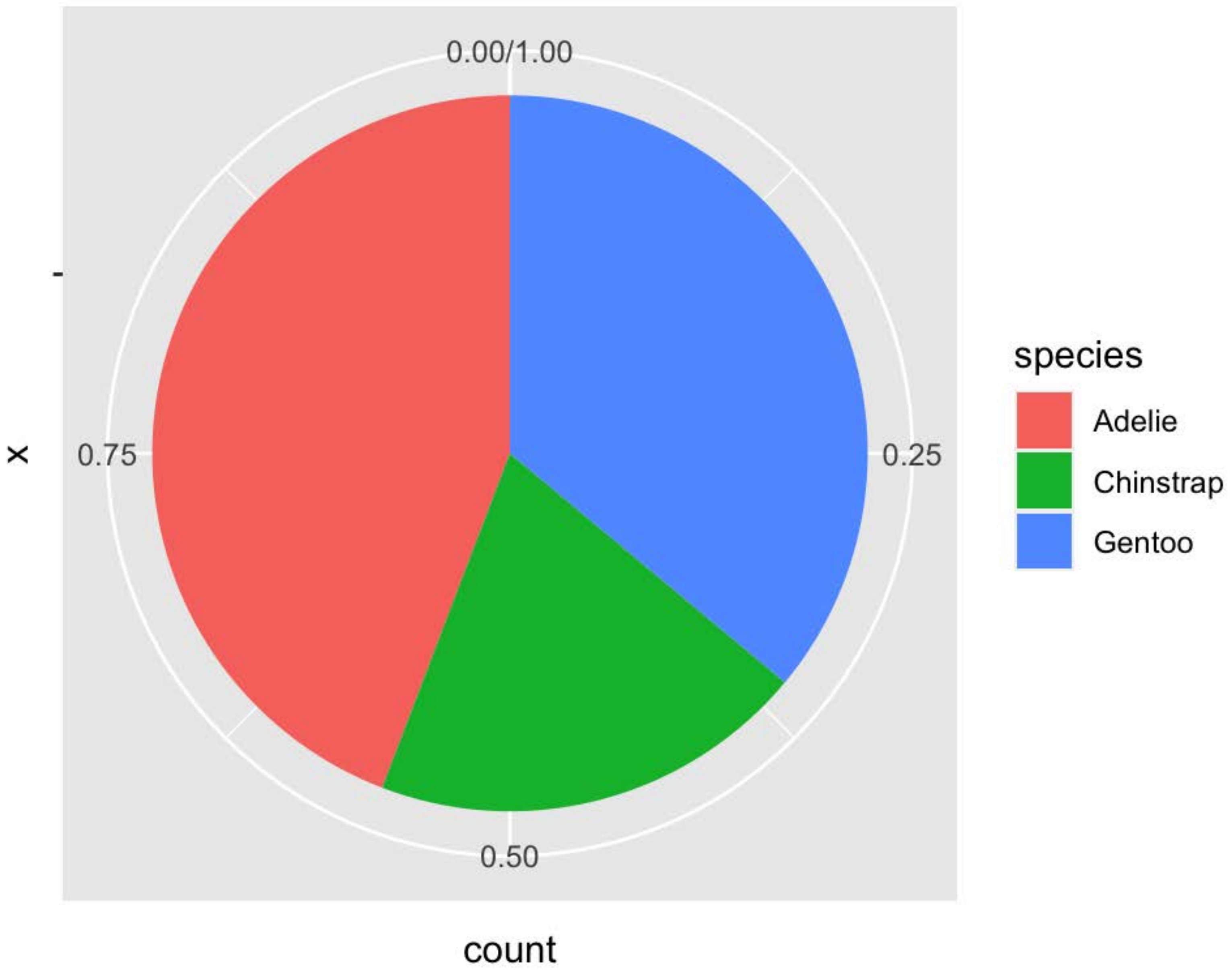
HOW TO MAKE A PIE





```
ggplot(penguins, aes(x = "", fill = species)) +
  geom_bar(position = "fill") +
  coord_polar(theta = "y")
```

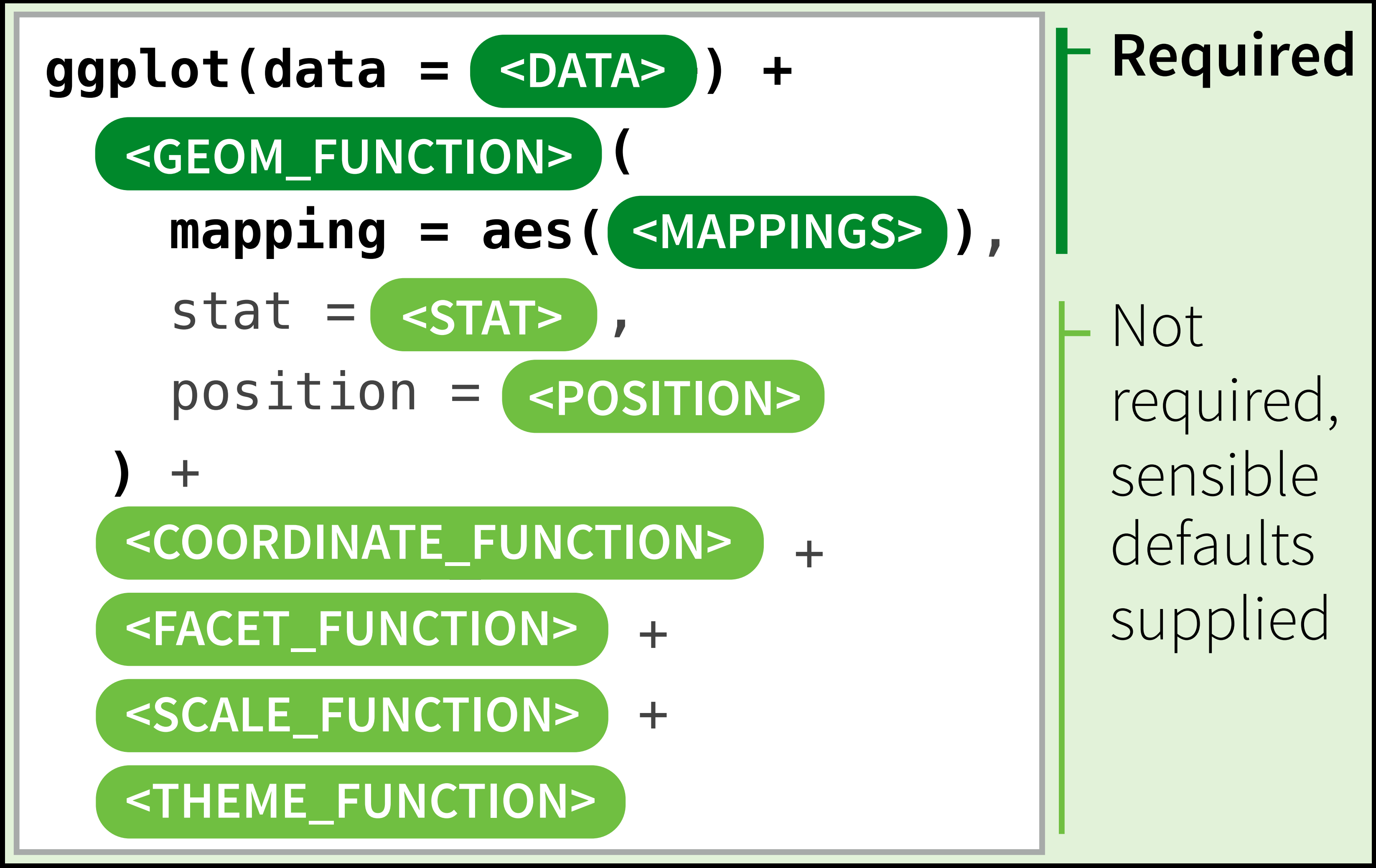
HOW DO WE MAKE THIS LESS UGLY?



```
ggplot(penguins, aes(x = "", fill = species)) +
  geom_bar(position = "fill") +
  coord_polar(theta = "y")
```

WHAT ELSE?

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes



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

THEMES

- Themes control the appearance of all the non-data elements of the plot

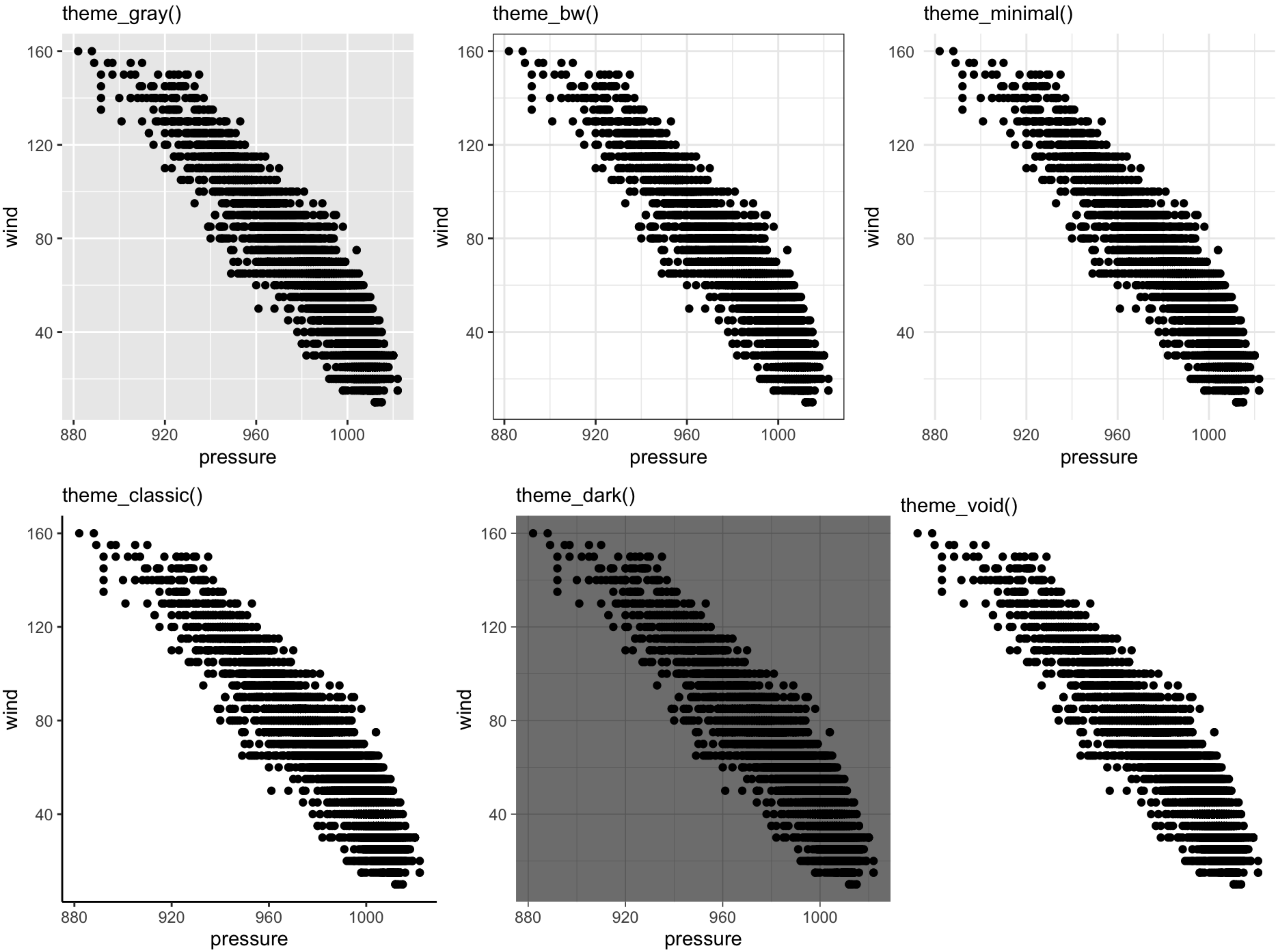
- ggplot2 includes several complete built-in themes.

- In addition, the appearance of just about every non-data element of the plot can be customized using the `theme()` function.

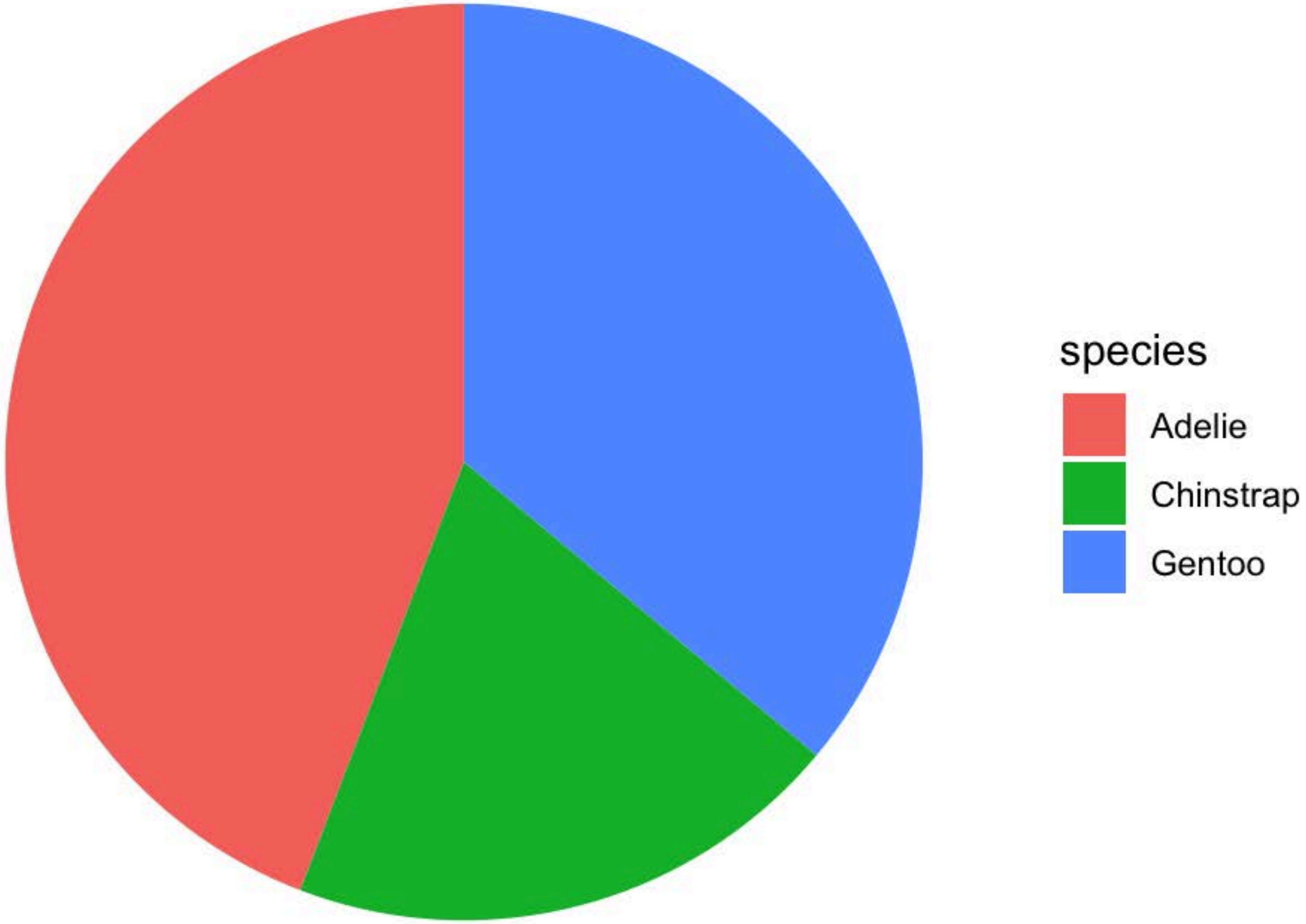
    - We will not talk much about customizing the built-in themes.

    - Lots of examples in the online documentation: https://ggplot2.tidyverse.org/reference/theme.html

    - Warning: tons of options, sort of tedious to learn


theme_gray()
theme_bw()
theme_minimal()
theme_classic()
theme_dark()
theme_void()
wind
pressure
160
120
80
40
880
920
960
1000

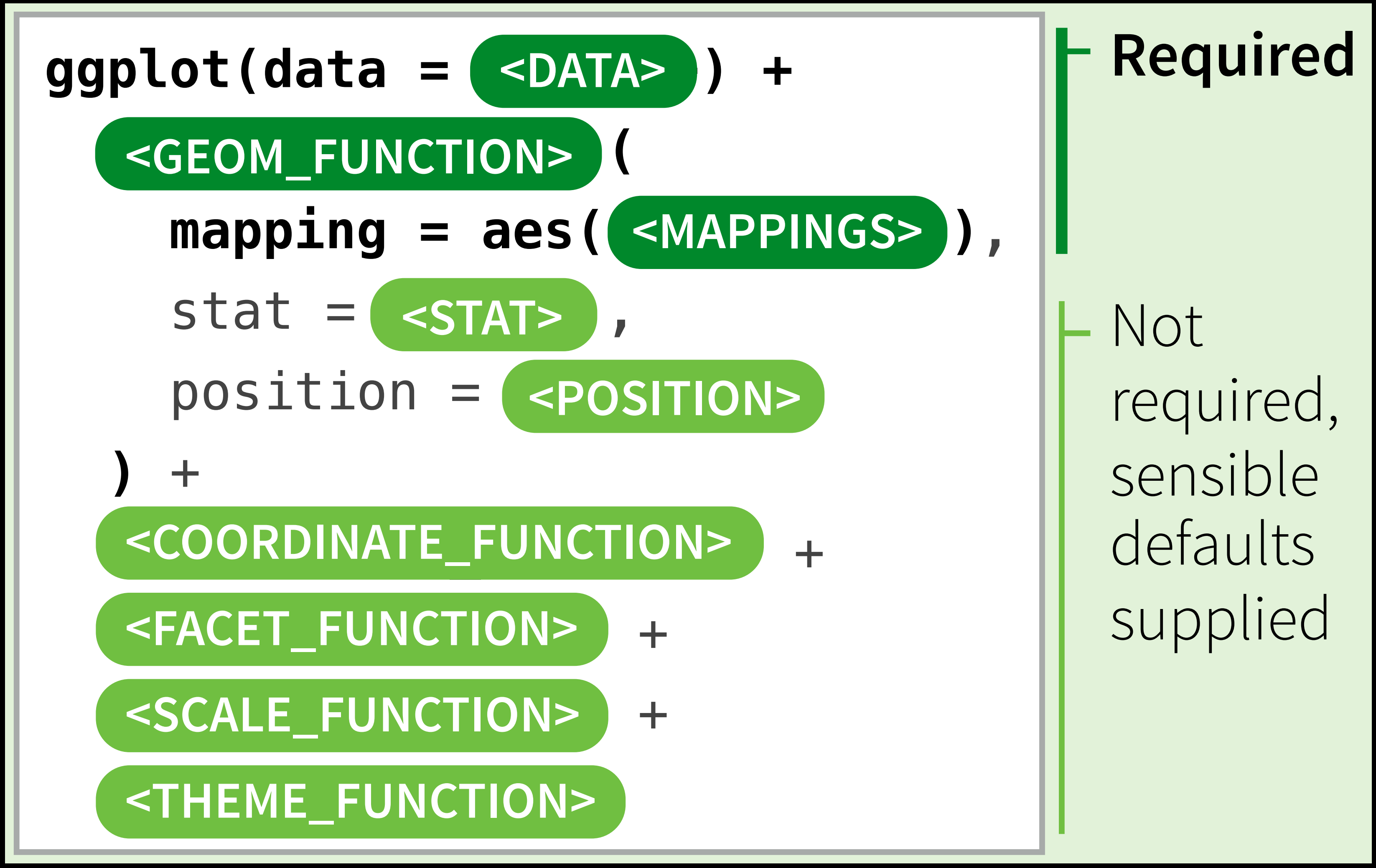
HOW DO WE MAKE THIS LESS UGLY?



```
ggplot(penguins, aes(x = "", fill = species)) +
  geom_bar(position = "fill") +
  coord_polar(theta = "y") +
  theme_void()
```

WHAT ELSE?

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes

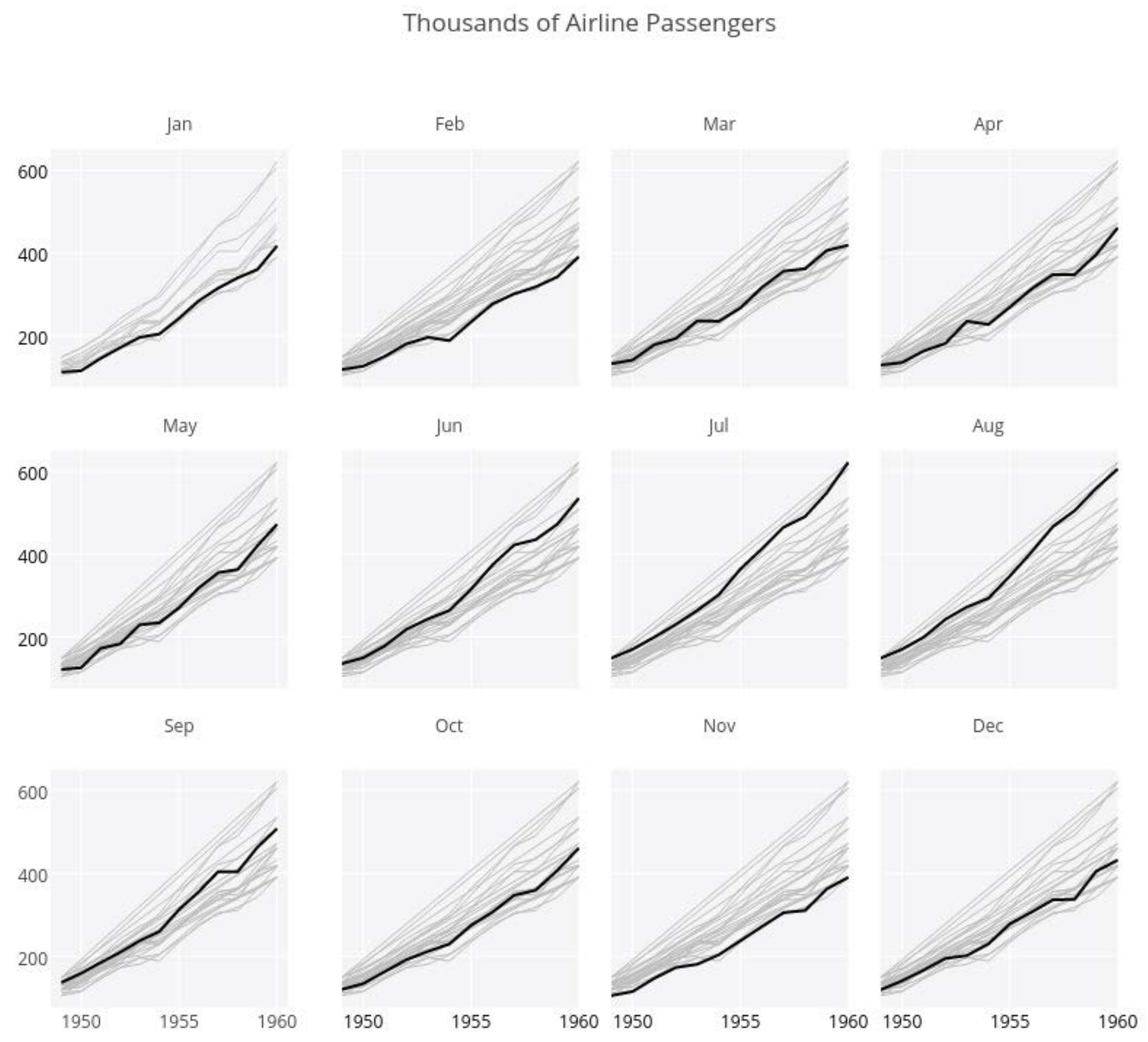


```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

Required
Not required, sensible defaults supplied

SMALL MULTIPLES

- Use the same basic graphic or chart to display different **slices** of a data set.
  - Indexed by category, time period, or some other variable not shown in chart
- Great way of showing complex data.
- In ggplot2, these are called *facets*.

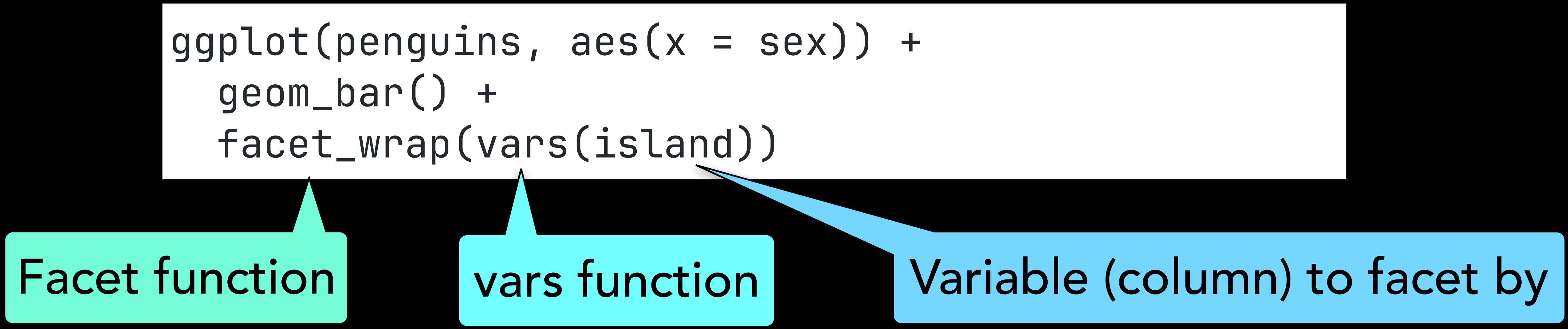


FACET FUNCTIONS

- Two facet functions:

1. `facet_wrap()`: lay out a sequence of small multiples, usually by one discrete variable

   - Control number of rows or columns with options `ncol` or `nrow`



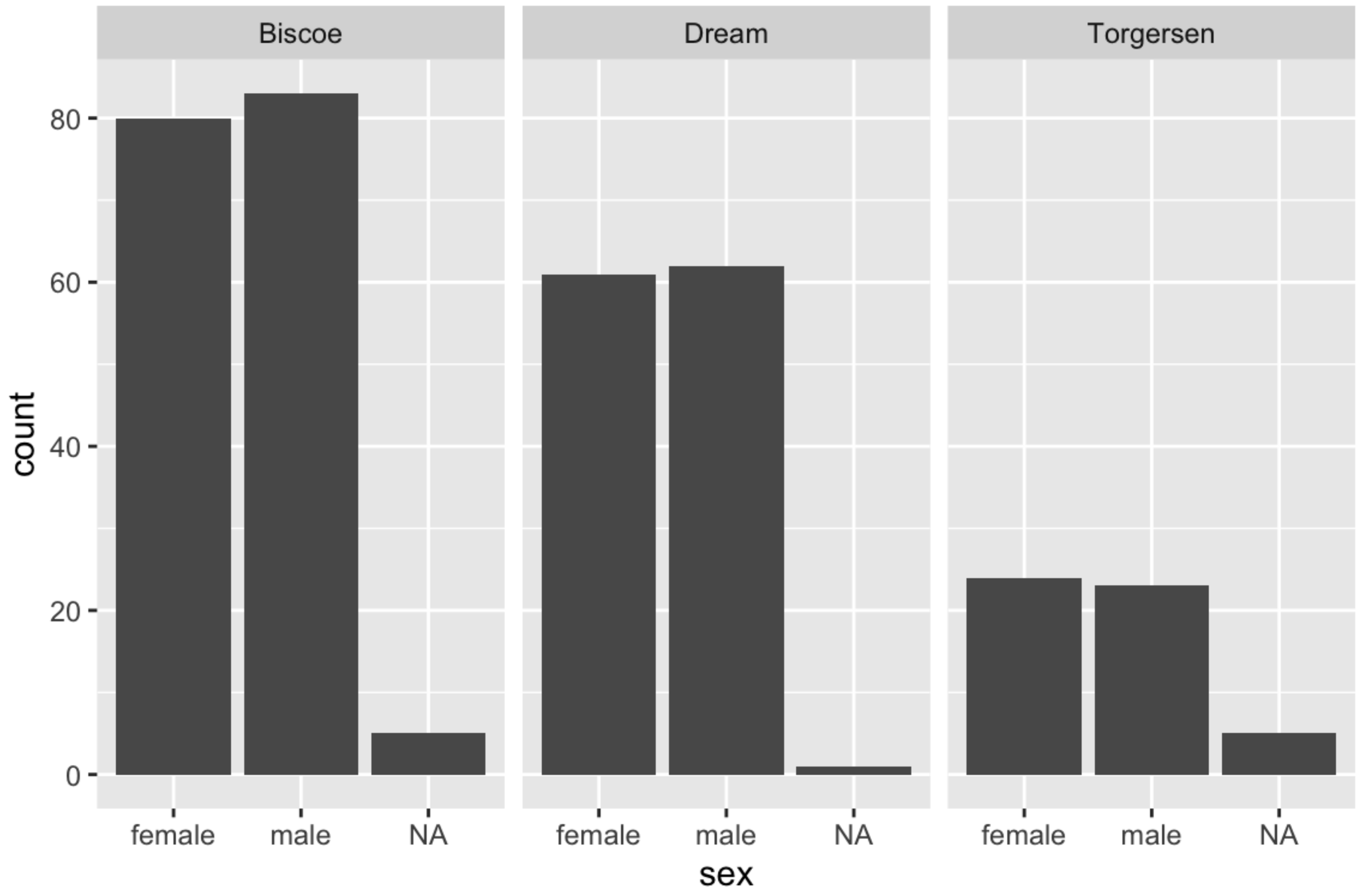
```
ggplot(penguins, aes(x = sex)) +
  geom_bar() +
  facet_wrap(vars(island))
```

Facet function

vars function

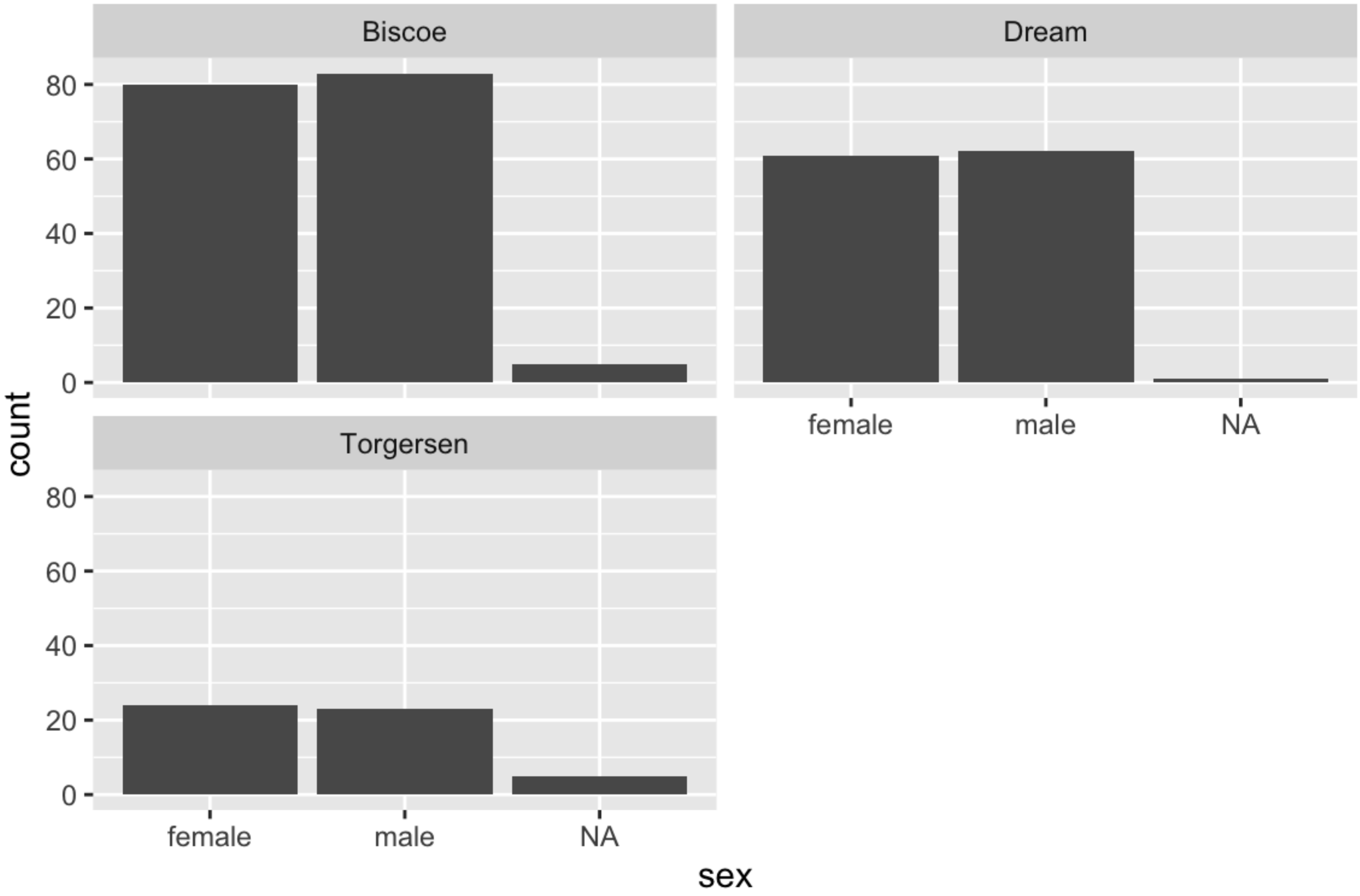
Variable (column) to facet by

FACET FUNCTIONS



```
ggplot(penguins, aes(x = sex)) +
  geom_bar() +
  facet_wrap(vars(island))
```

FACET FUNCTIONS



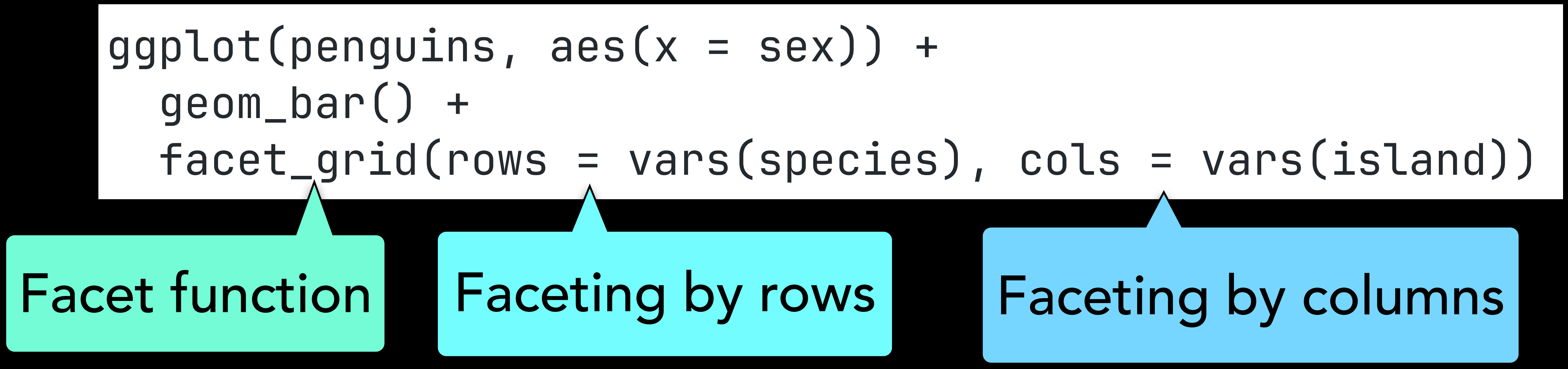
```
ggplot(penguins, aes(x = sex)) +
  geom_bar() +
  facet_wrap(vars(island), ncol = 2)
```

FACET FUNCTIONS

- Two facet functions:

  2. `facet_grid()`: lay out a grid of small multiples using two discrete variables

     - Provide faceting variables for grid using **`rows`** and **`cols`**



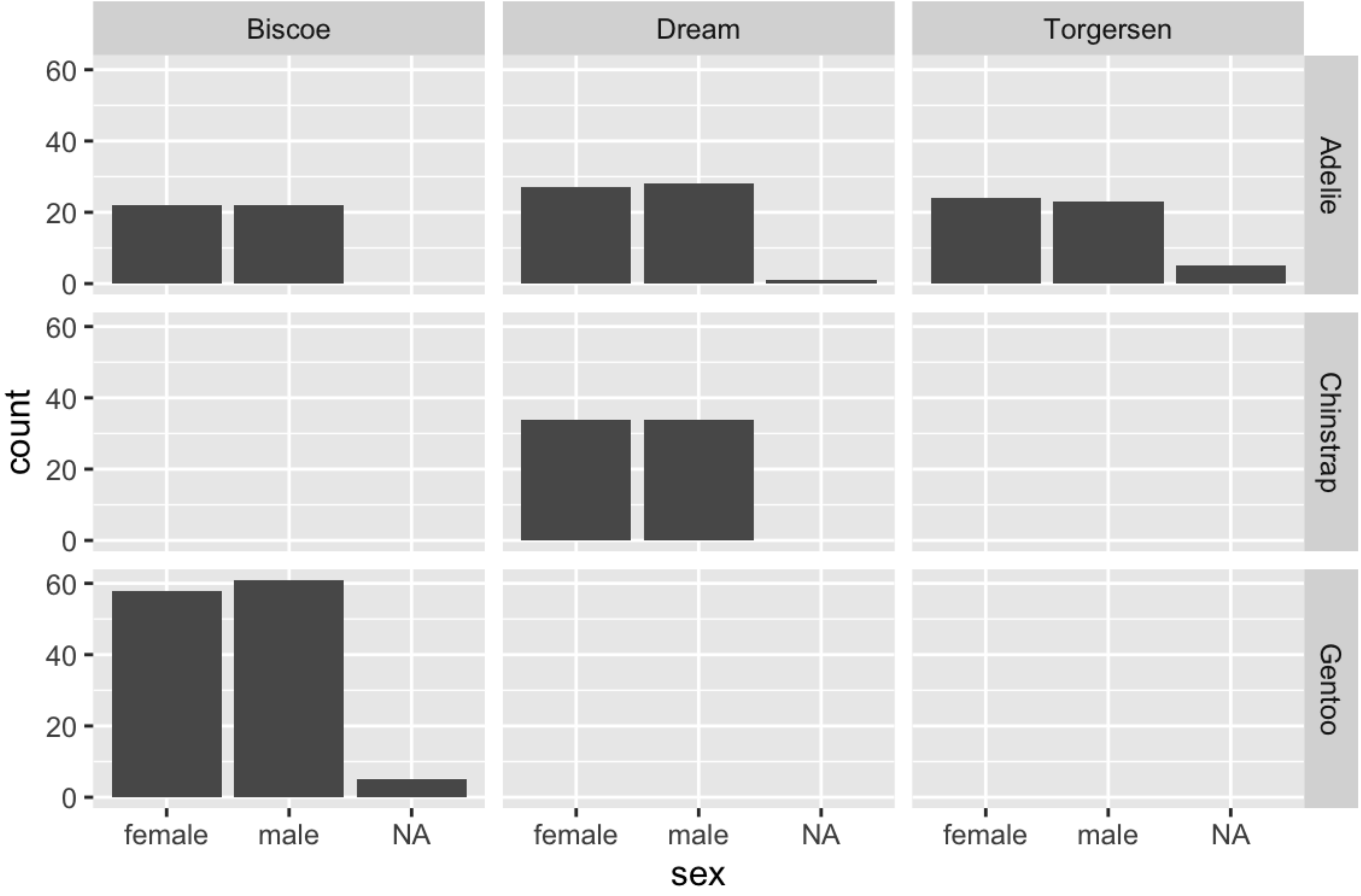
```
ggplot(penguins, aes(x = sex)) +
  geom_bar() +
  facet_grid(rows = vars(species), cols = vars(island))
```

Facet function

Faceting by rows

Faceting by columns

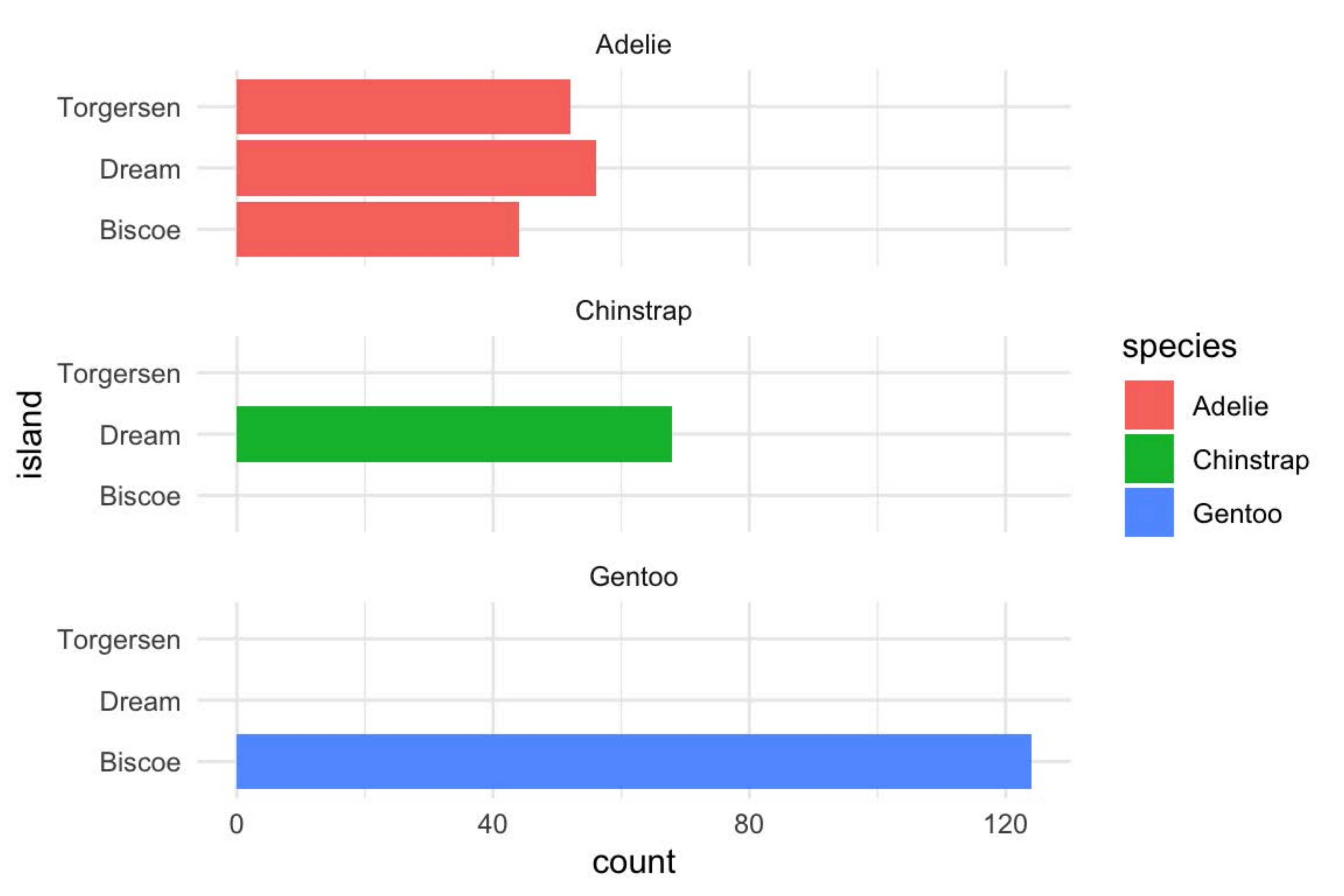
FACET FUNCTIONS



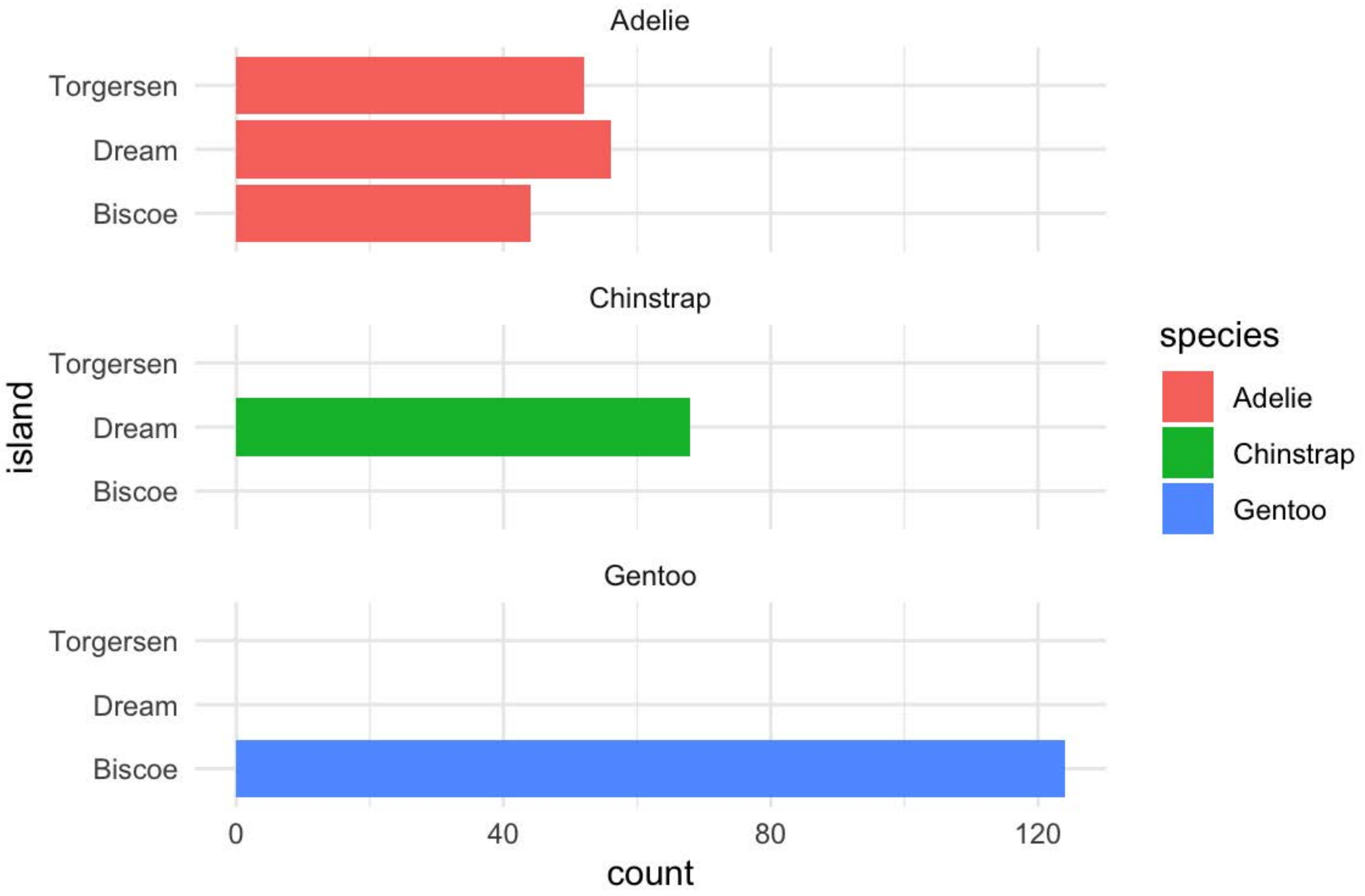
```
ggplot(penguins, aes(x = sex)) +
  geom_bar() +
  facet_grid(rows = vars(species), cols = vars(island))
```

YOUR TURN

- Try to make this plot:



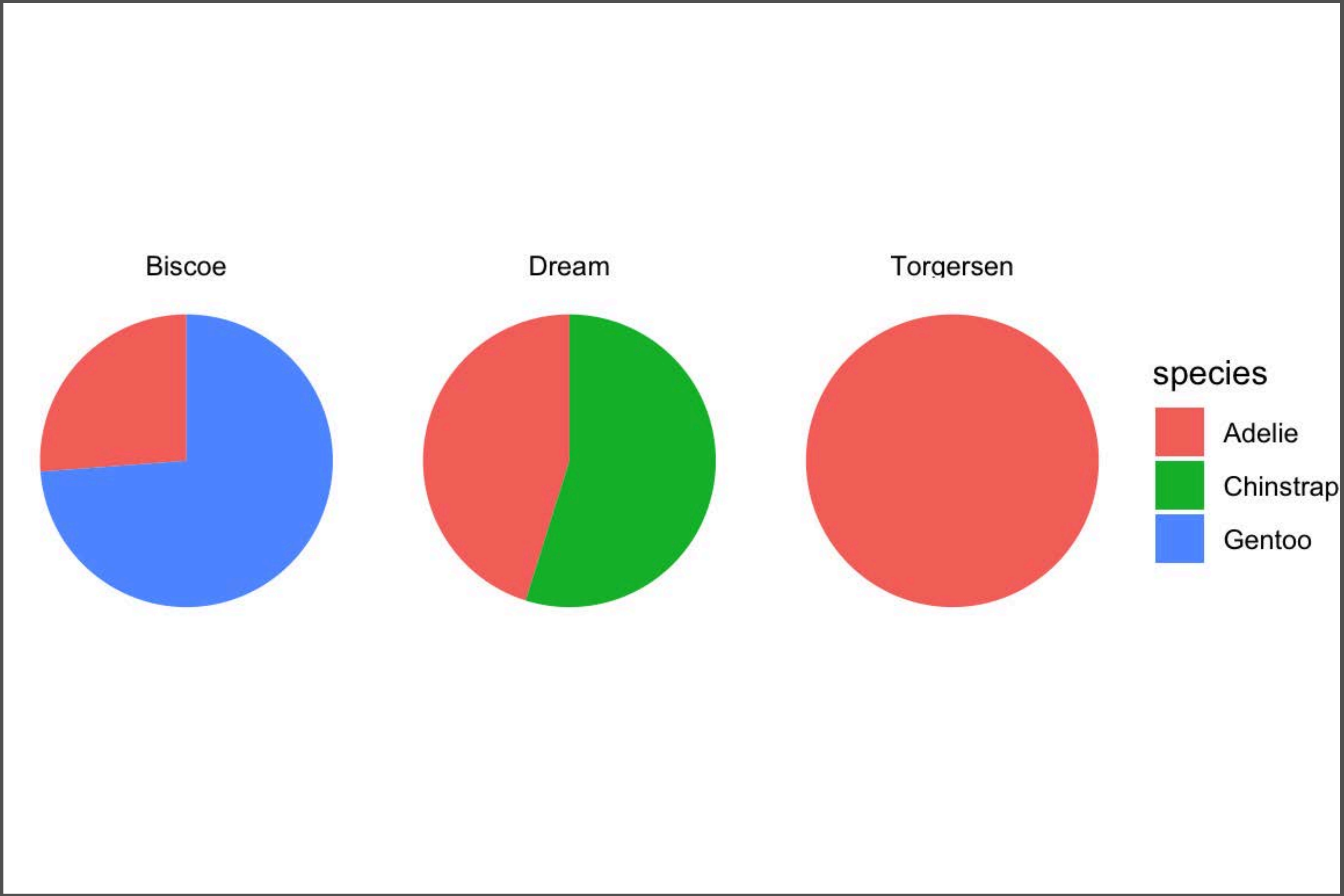
4:00



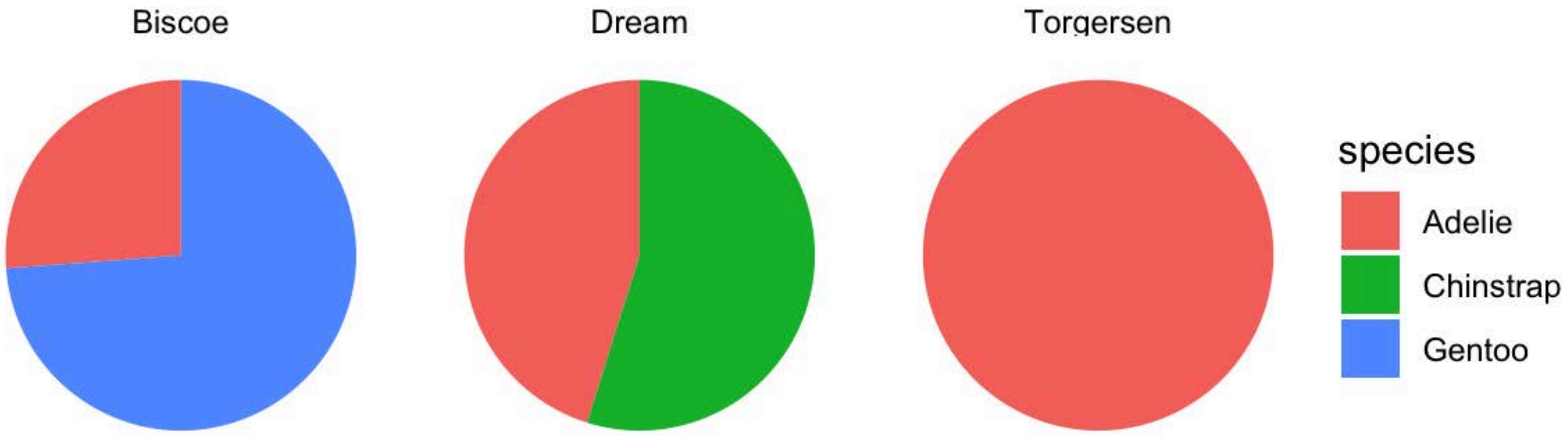
```
ggplot(penguins, aes(x = island, fill = species)) +
  geom_bar() +
  coord_flip() +
  facet_wrap(vars(species), ncol = 1) +
  theme_minimal()
```

YOUR TURN

- Try to make this plot:



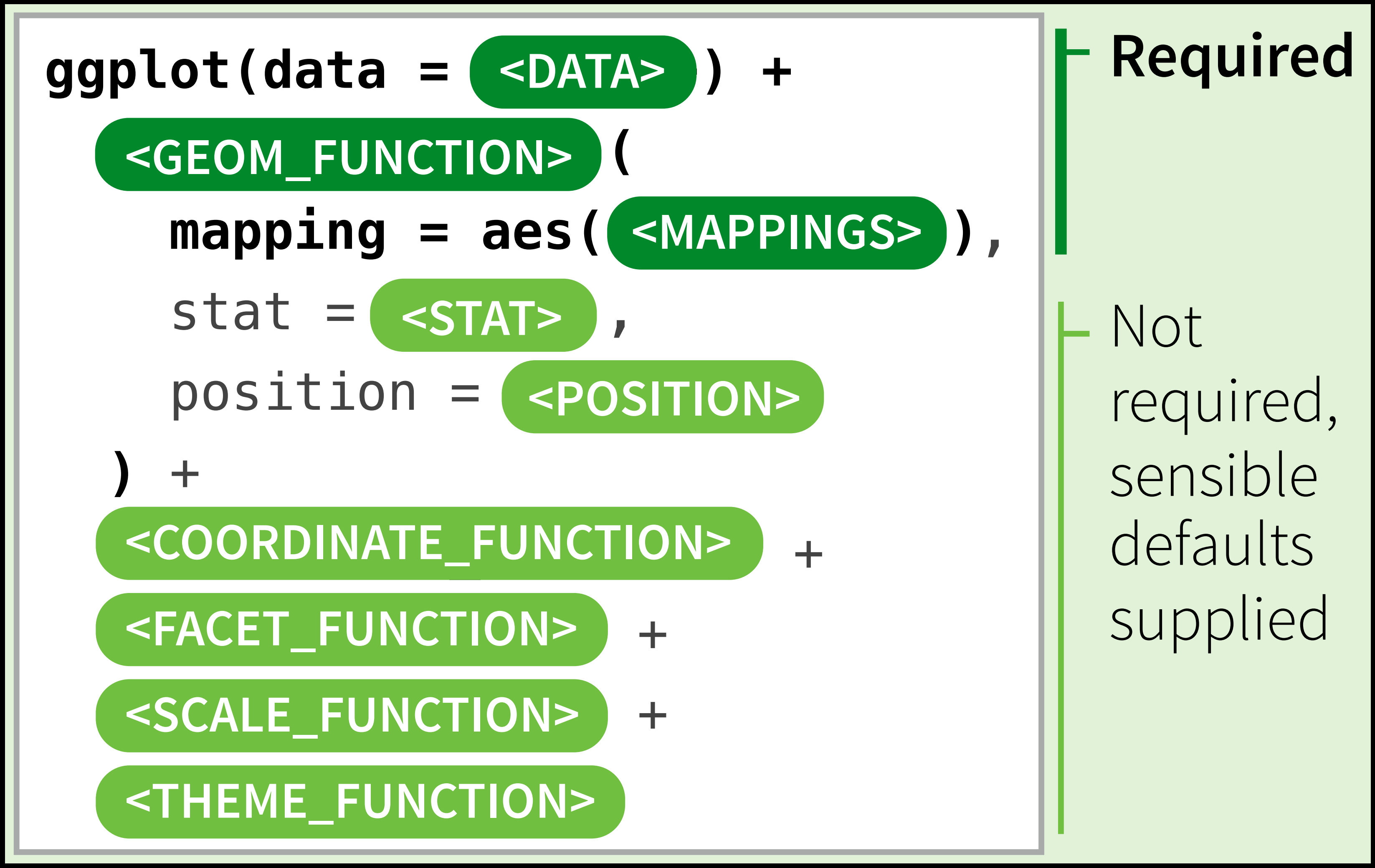
4:00



```
ggplot(penguins, aes(x = "", fill = species)) +
  geom_bar(position = "fill") +
  coord_polar(theta = "y") +
  facet_wrap(vars(island)) +
  theme_void()
```

WHAT ELSE?

- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes



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

Required

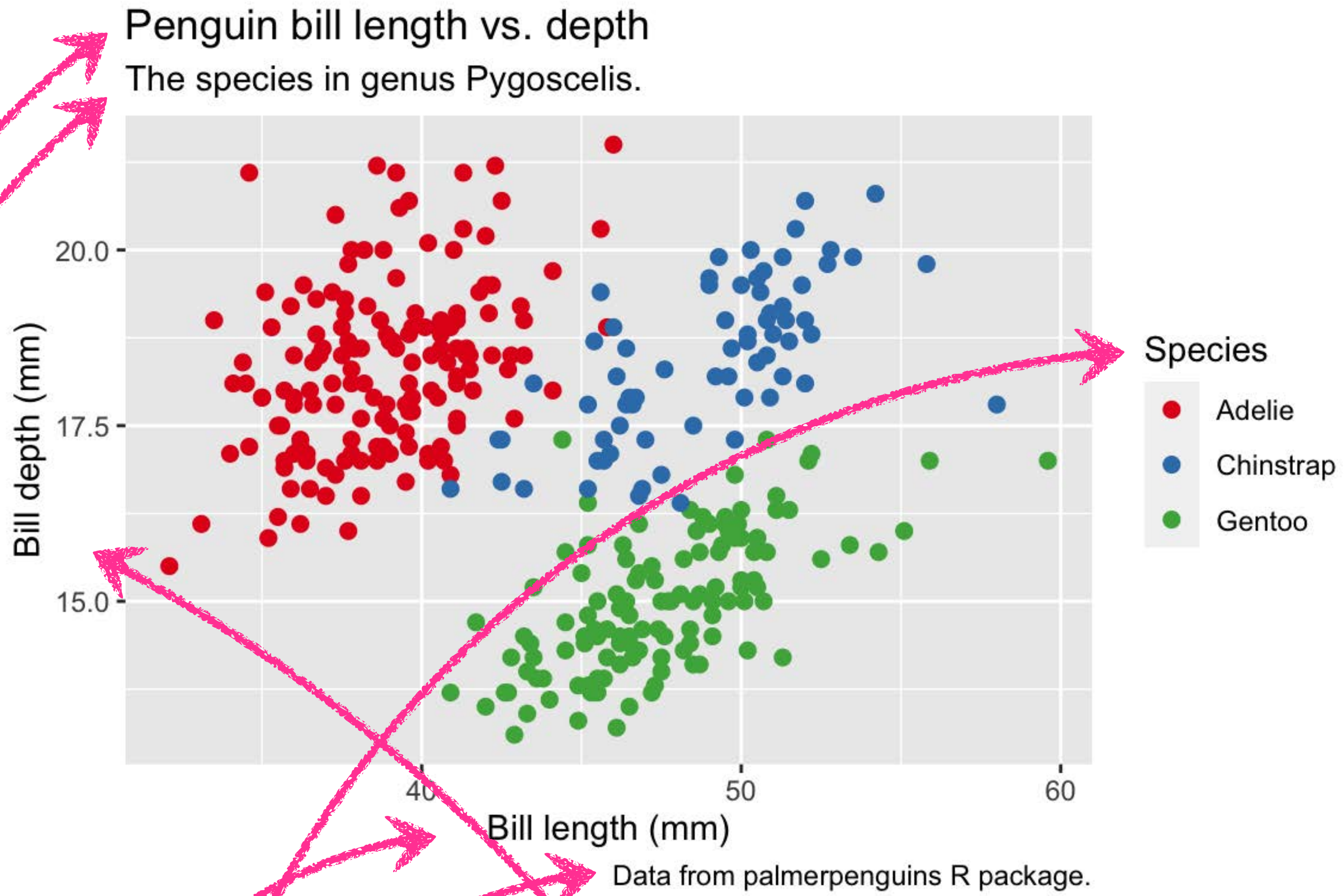
Not required, sensible defaults supplied

SCALE FUNCTIONS

- Scales control how data values are translated to visual properties.

- These are usually set to sensible defaults, but details often need to be tweaked.

- Override the default scales to modify things like the axis labels, legend keys, color palettes, x & y position, etc.



```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm,
                     color = species)) +
  geom_point(size = 2) +
  scale_color_brewer(palette = "Set1") +
  labs(x = "Bill length (mm)", y = "Bill depth (mm)",
       color = "Species",
       title = "Penguin bill length vs. depth",
       subtitle = "The species in genus Pygoscelis.",
       caption = "Data from palmerpenguins R package.")
```

YOUR TURN

ggplot2

www.rstudio.com

- Go to this week's assignments on the course website.
- Download the **baboon activities** R Markdown file.
- Download the data file: baboon_acts_2000.csv
- Follow the instructions to visualize baboon activity budgets using pies, bars, and other types of charts
- Also learn some ways to fine-tune your plot's appearance.

45:00