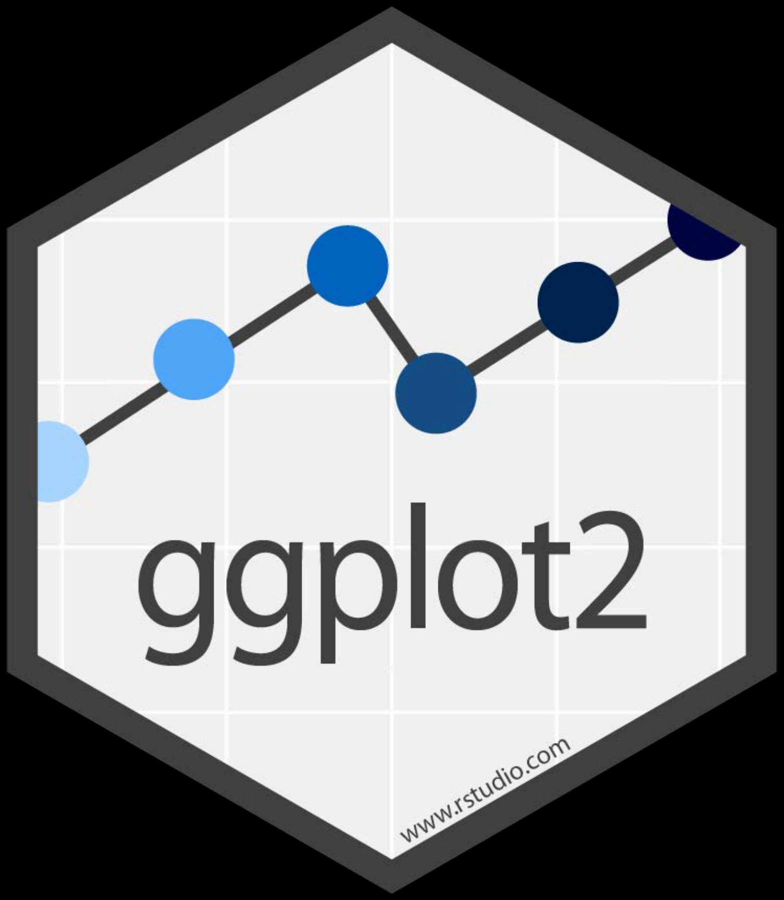
TODAY'S TOPICS

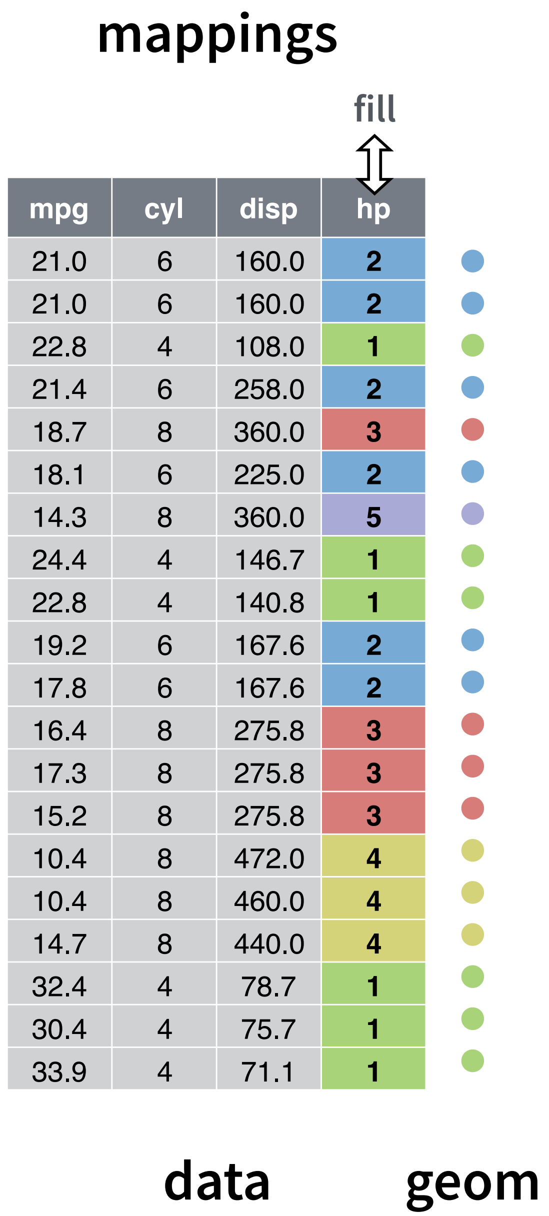
- Visualizing amounts and proportions with pie charts, bar charts, and variations
- Fine-tuning ggplot2
  - facet functions
  - scale functions
  - coord functions
  - themes
- Activities: pies and bars

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

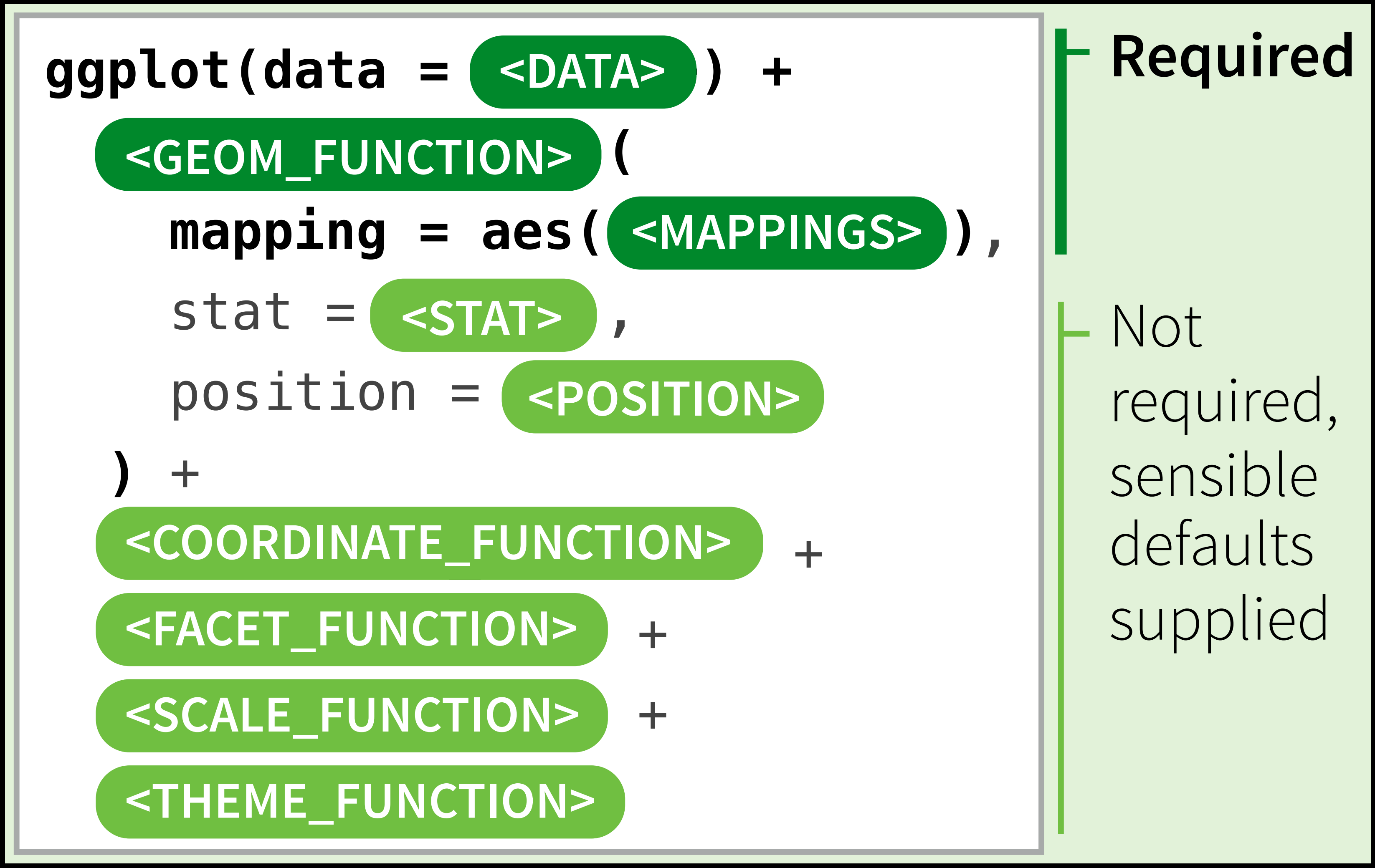
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

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

3. **Map** aesthetic properties to variables

WHAT ELSE?

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



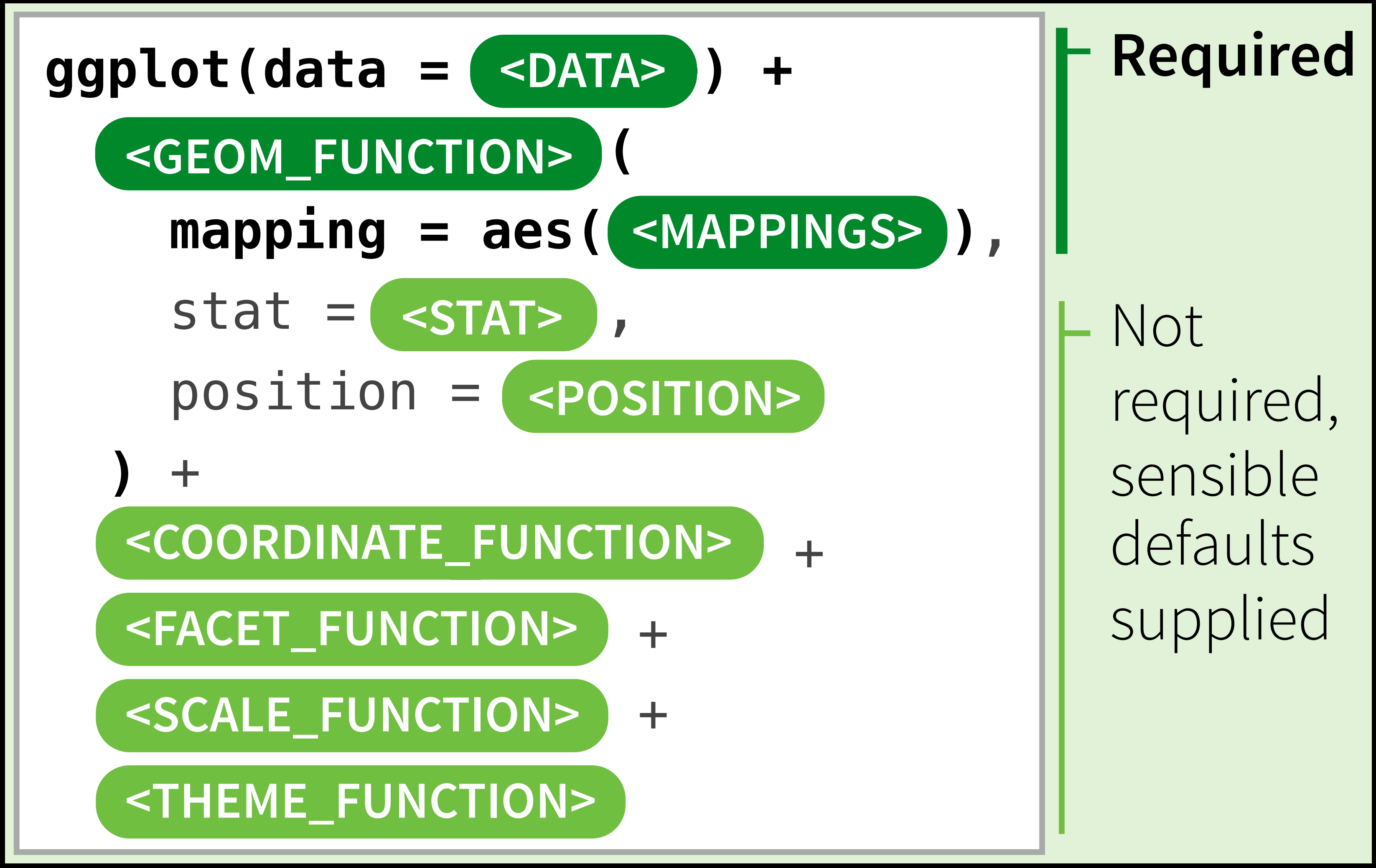
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
     mapping = aes(<MAPPINGS>),
     stat = <STAT>,
     position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

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>
```

Required

Not required, sensible defaults supplied

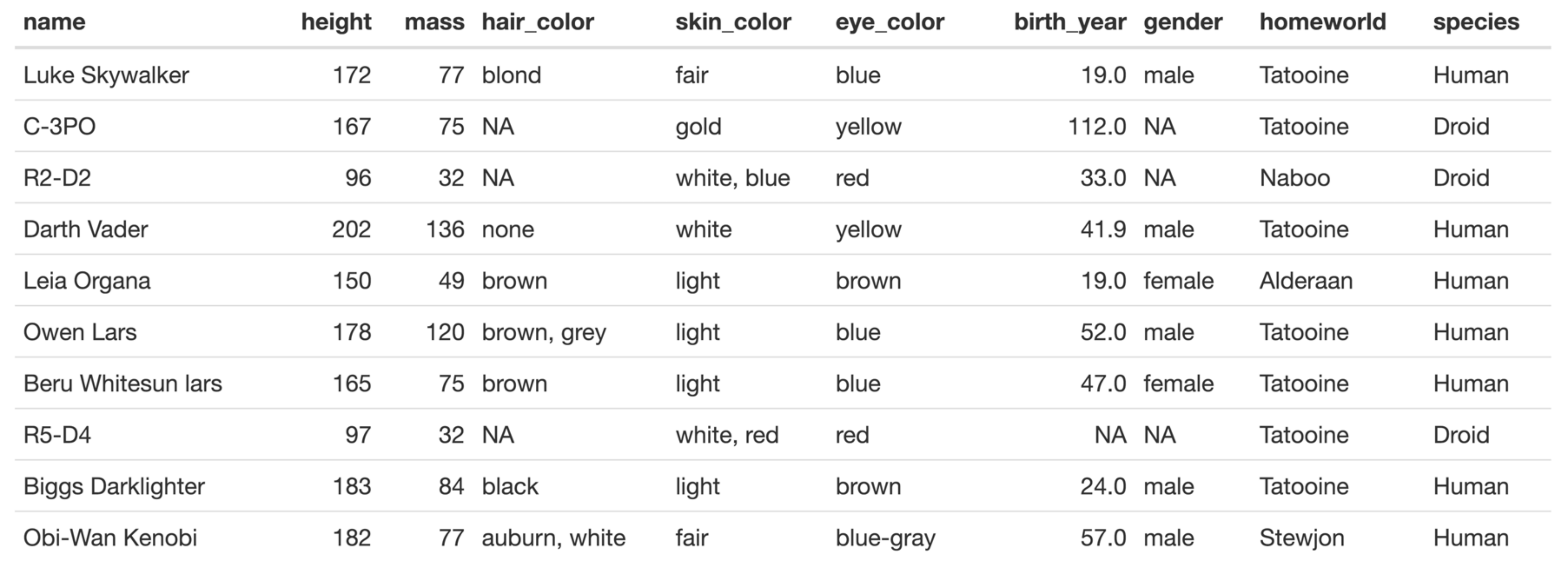
STATS

- Each geom_ function has a default stat, *so you can usually omit it!*
- Let's explore how stats work using bar charts.

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

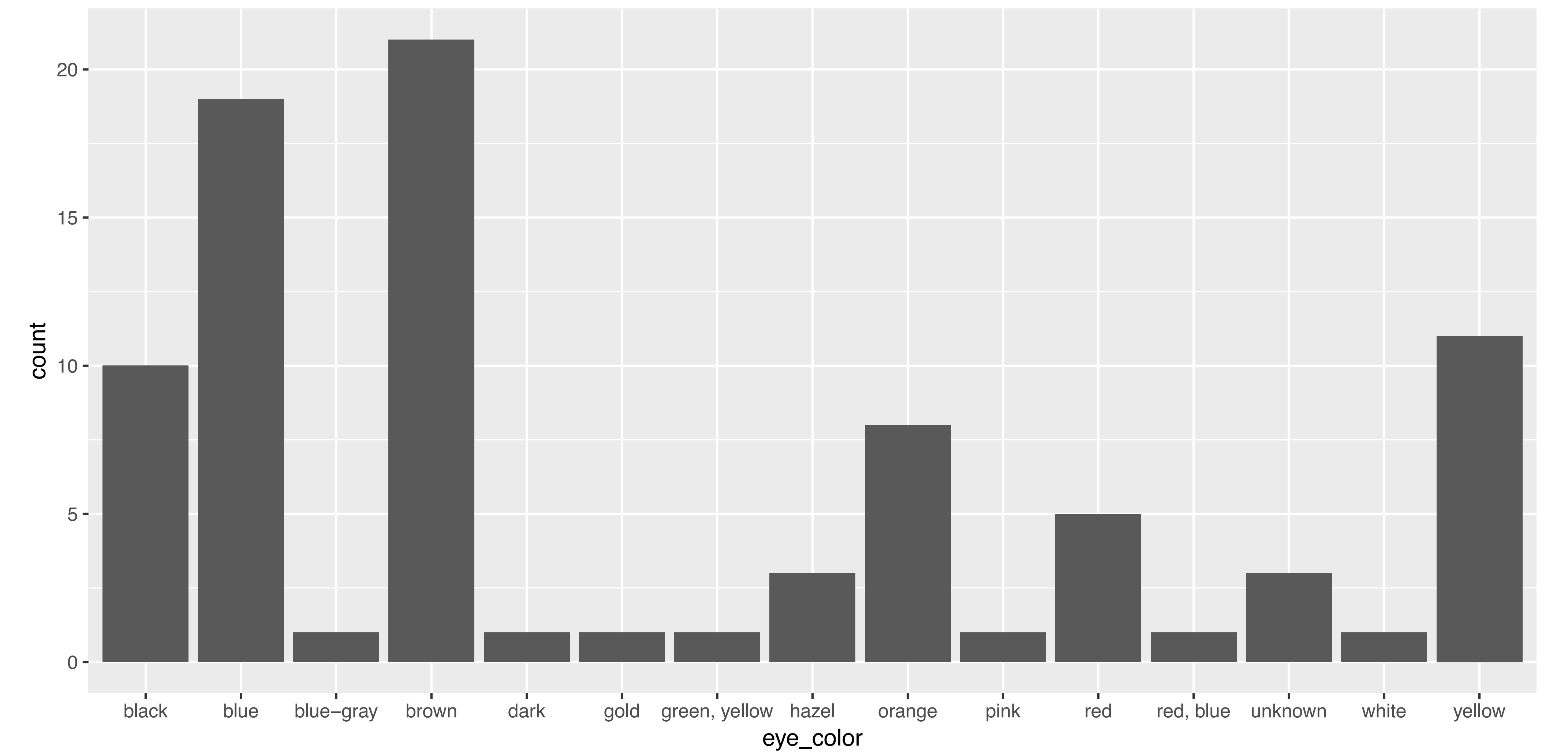
- Create a new R markdown file and clear out the extra stuff

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

- Try to create a bar chart of eye color using `geom_bar()`

3:00

YOUR TURN



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

STATS

- 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()
```

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

STATS

- What if our data set is already summarized?
- In other words, what if we need to use a different stat?

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.

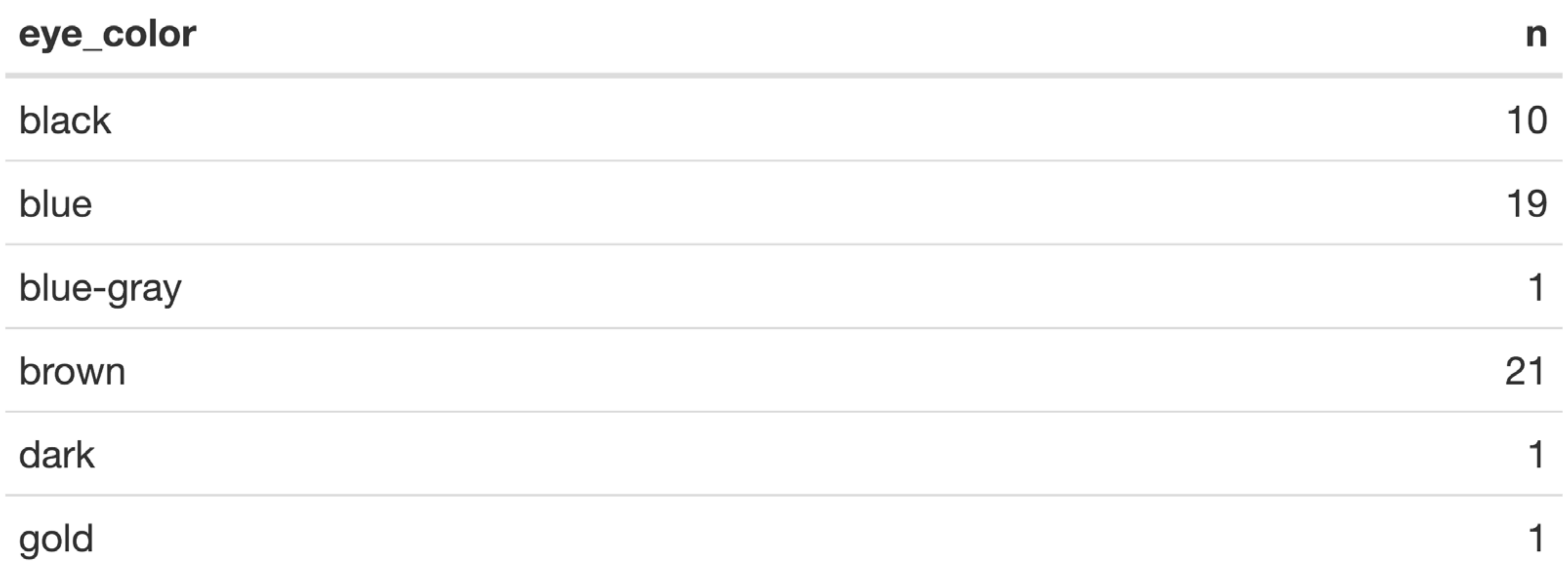
```
starwars_sum <- count(starwars, eye_color)
```

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

```r
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).

```
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")
```

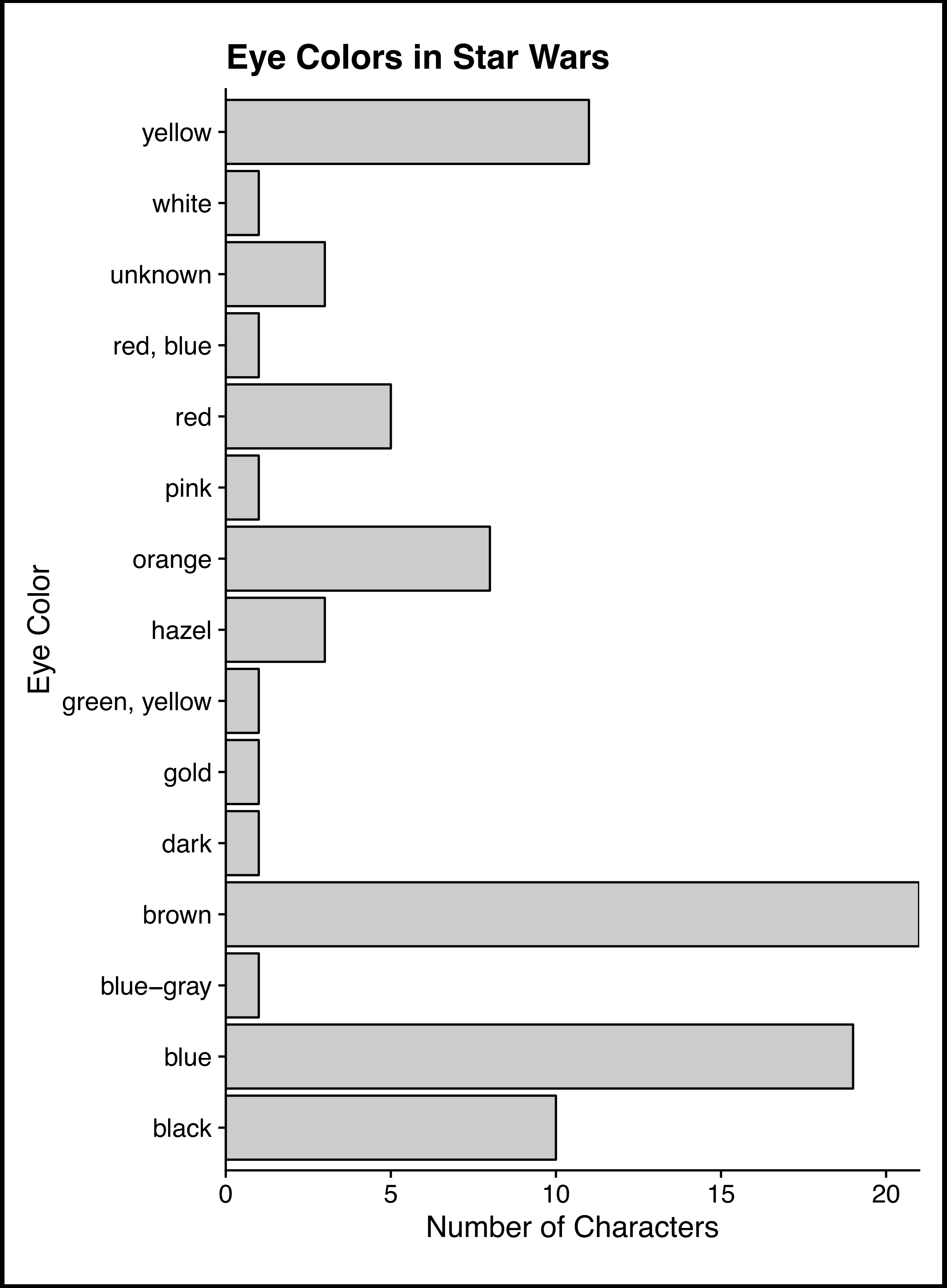
4:00

STATS

- 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 aesthetic for `geom_` functions if you really want to (tip: this is rarely a good idea).
- Bottom line: if the data are already summarized, then use `geom_col()`.

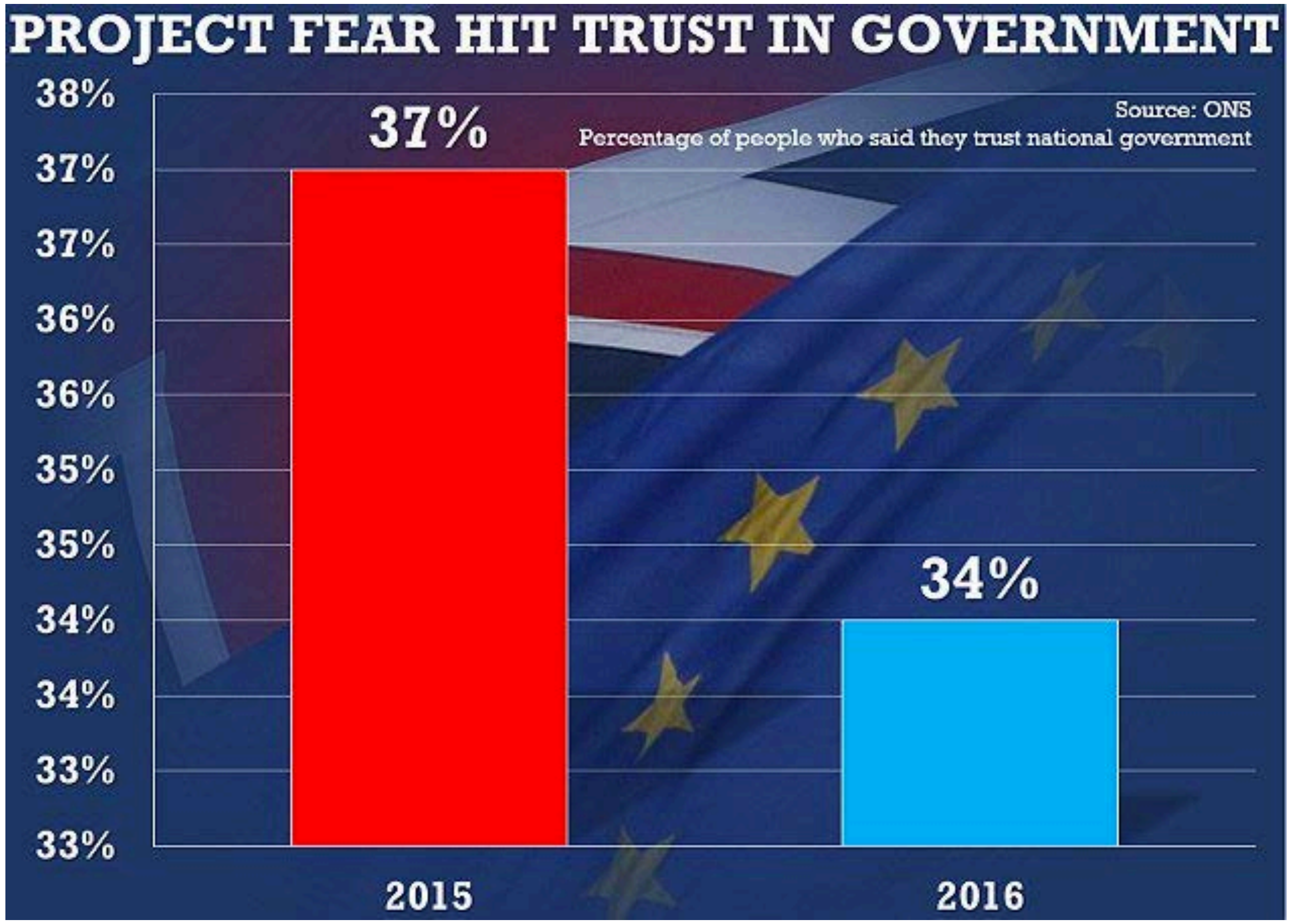
BAR CHARTS

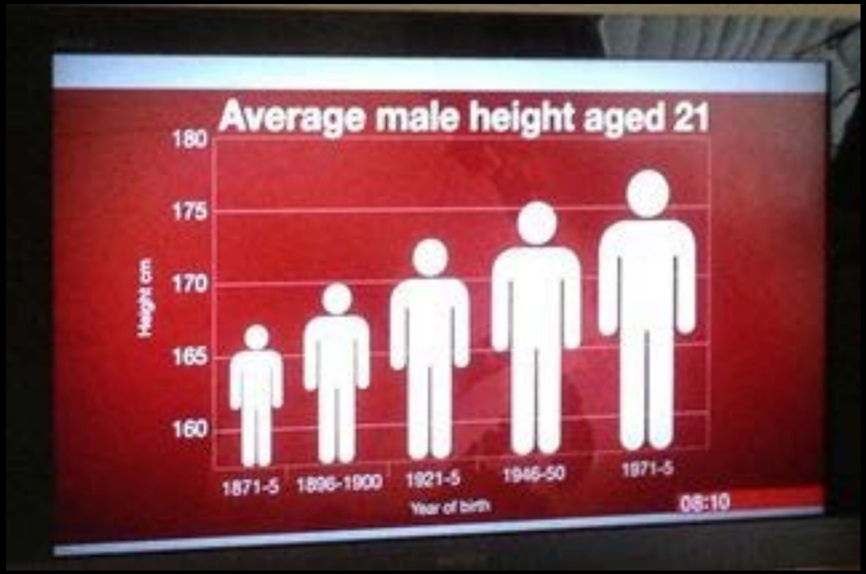
- Used for discrete groups or categories
- Y-axis should always include zero!

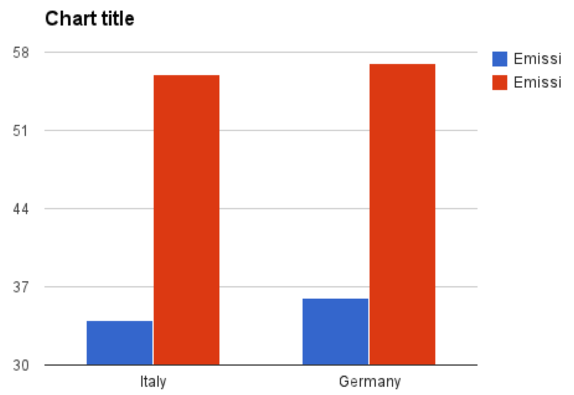


BAR CHARTS

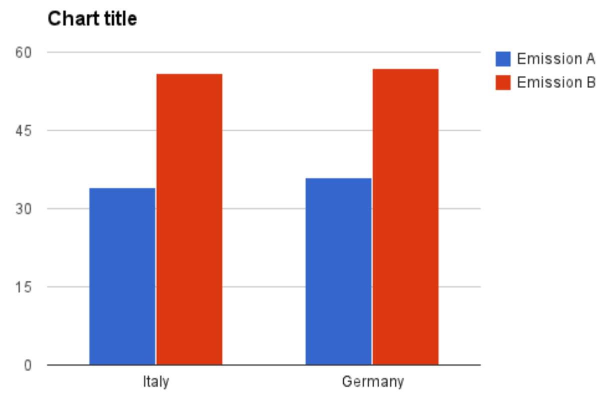
- Used for discrete groups or categories
- Y-axis should always include zero!







No

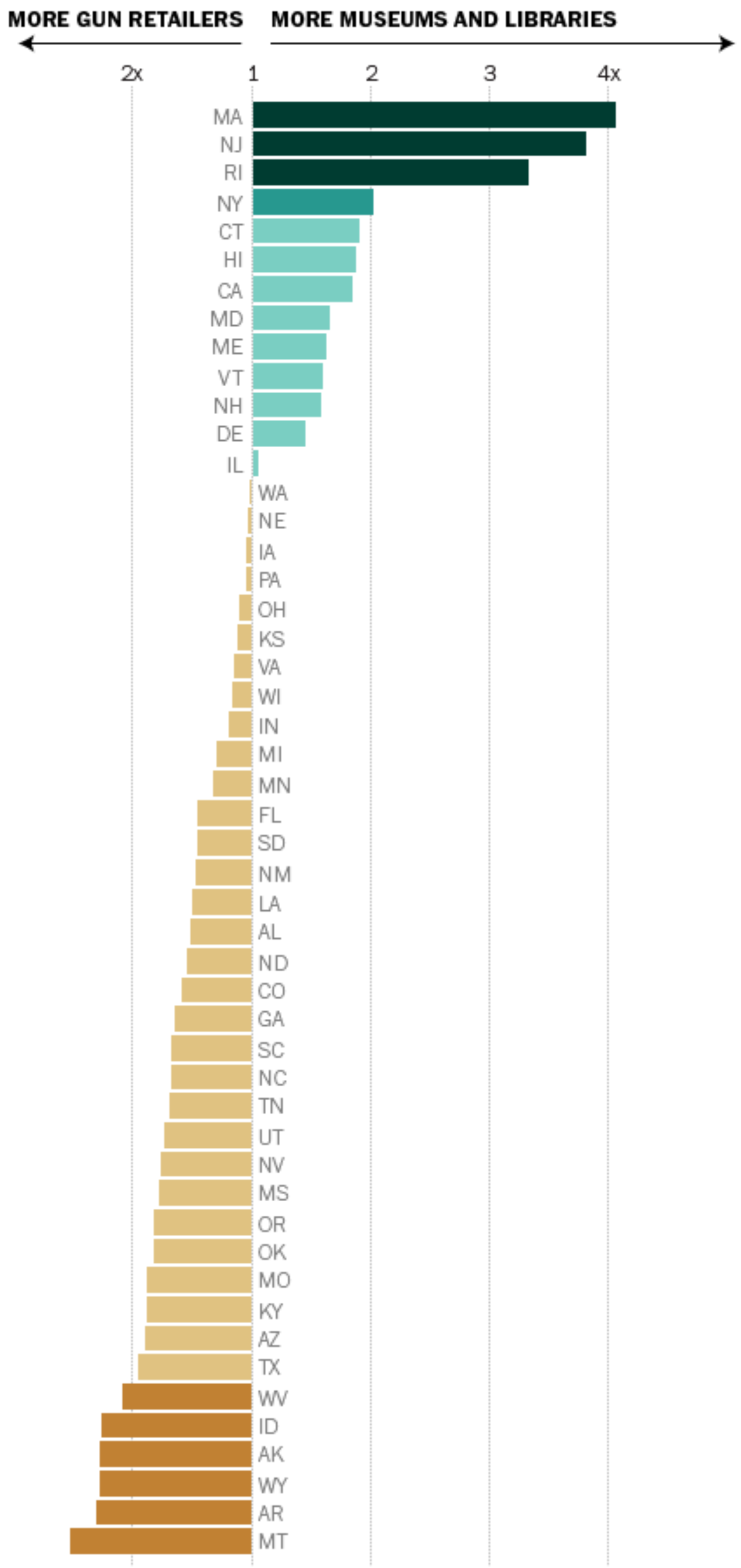


Yes

BAR CHARTS

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

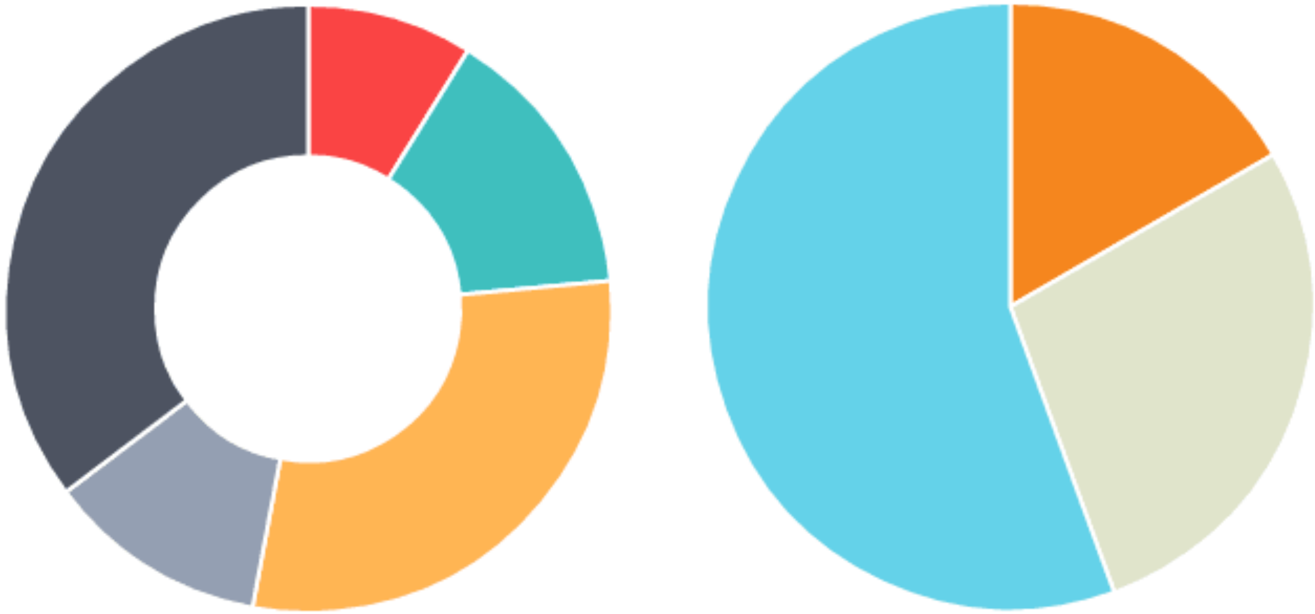
In 37 states, gun dealers outnumber museums and libraries



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

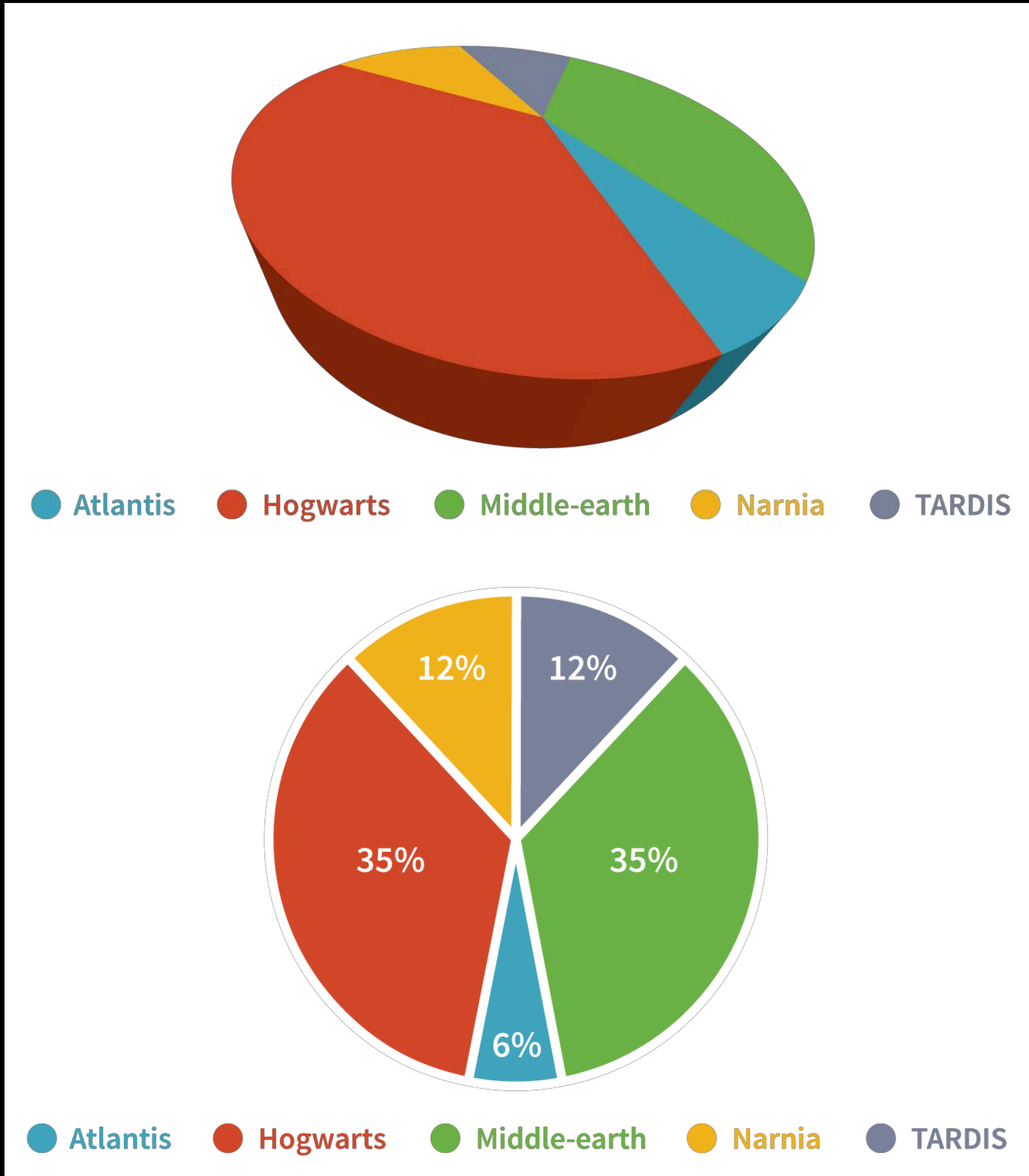
PIE CHARTS AND DONUT CHARTS

- Categorical variables
- Probably most misused type of graph



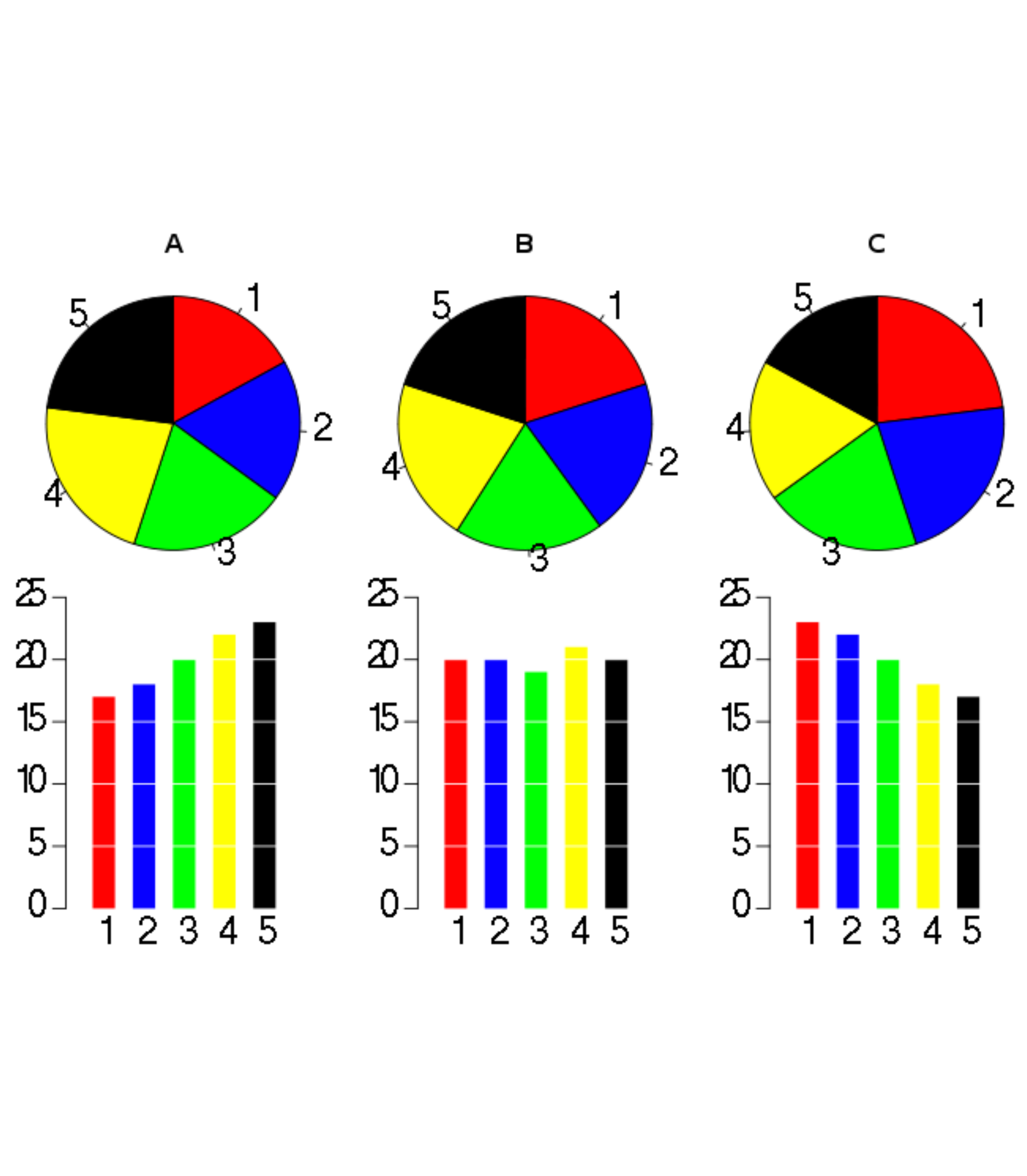
PIE CHARTS AND DONUT CHARTS

- Categorical variables
- Probably most misused type of graph
- Perceptual problems—no 3D!



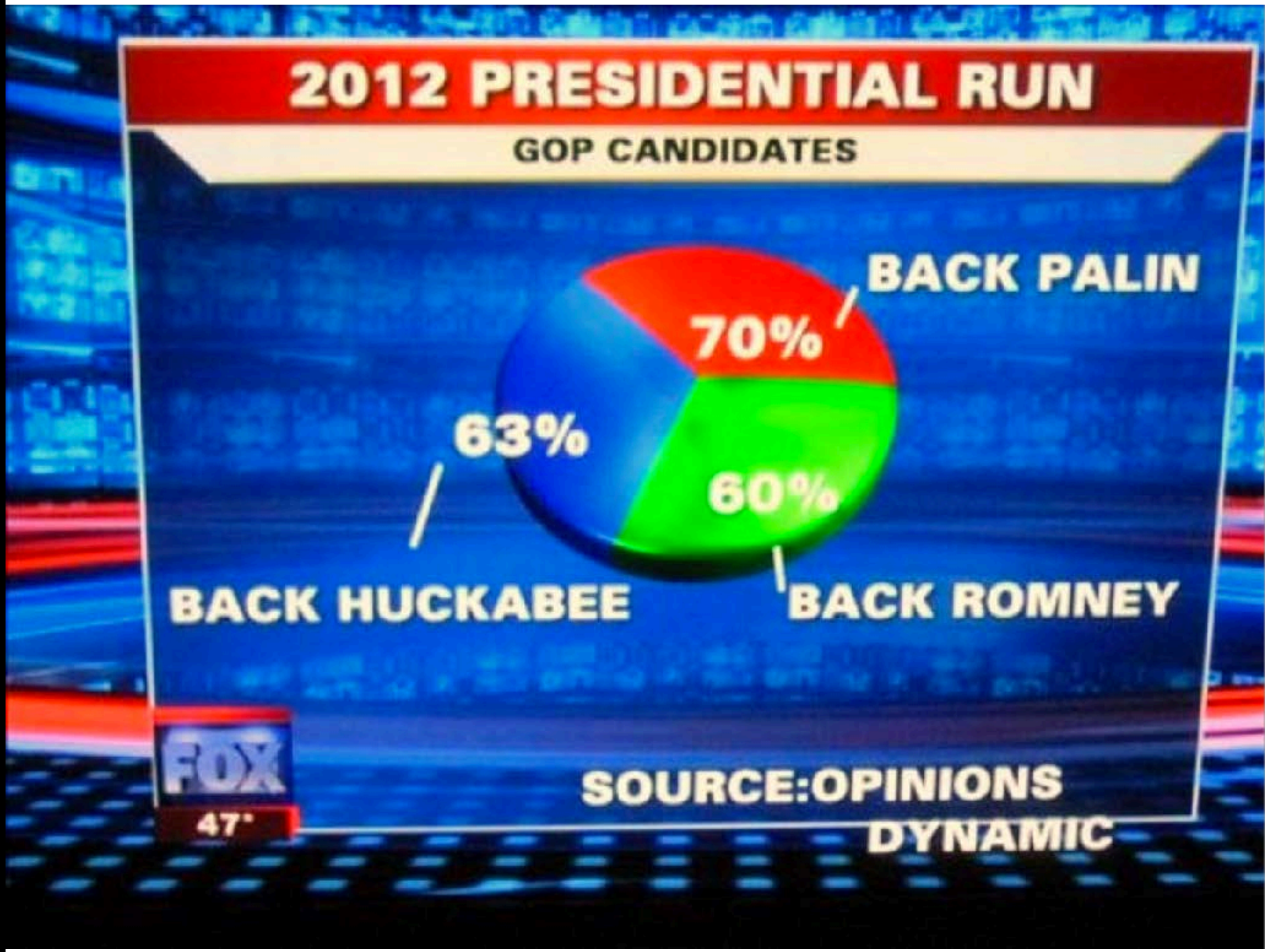
PIE CHARTS AND DONUT CHARTS

- Categorical variables
- Probably most misused type of graph
- Perceptual problems—no 3D!



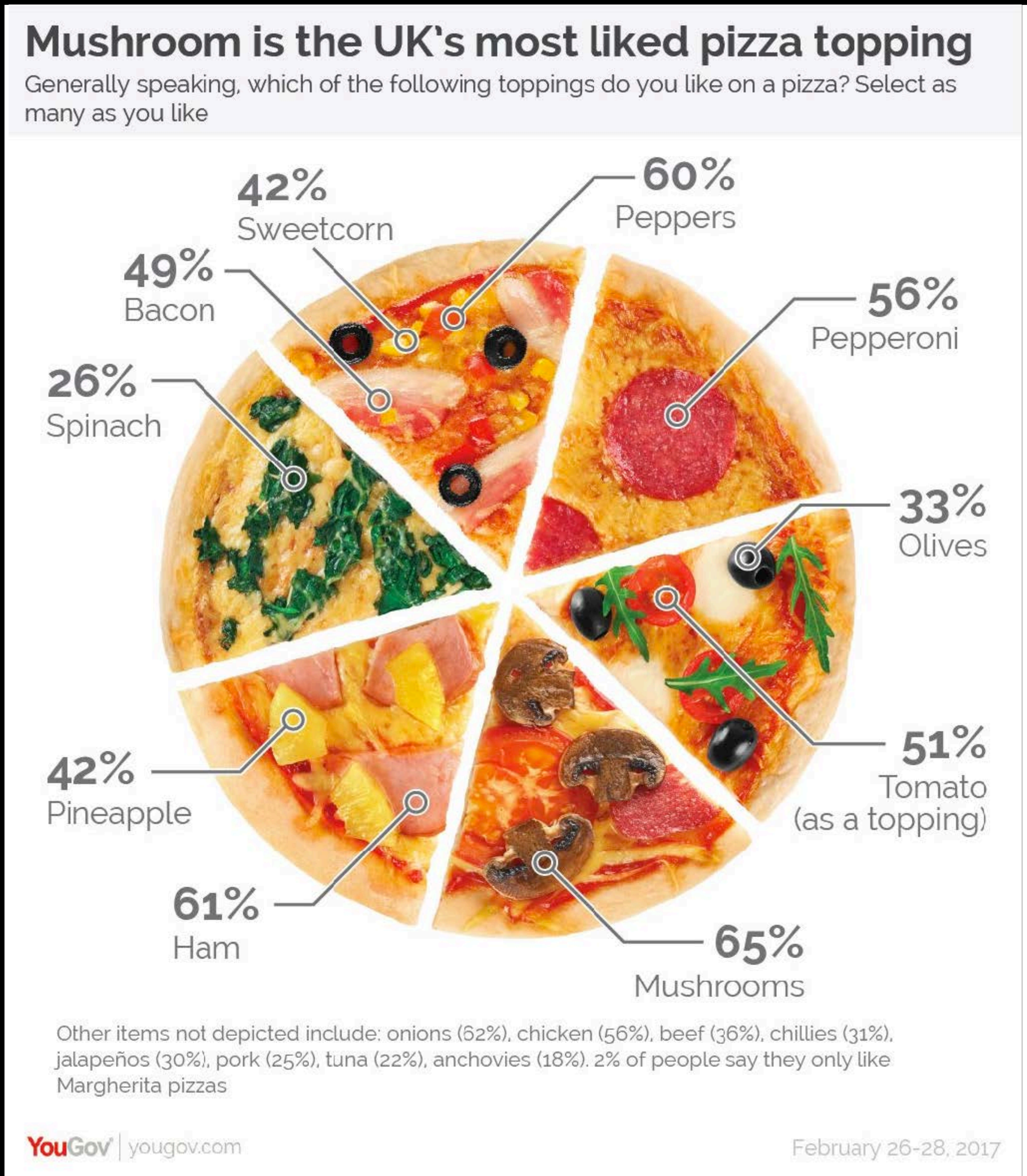
PIE CHARTS AND DONUT CHARTS

- 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

- 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

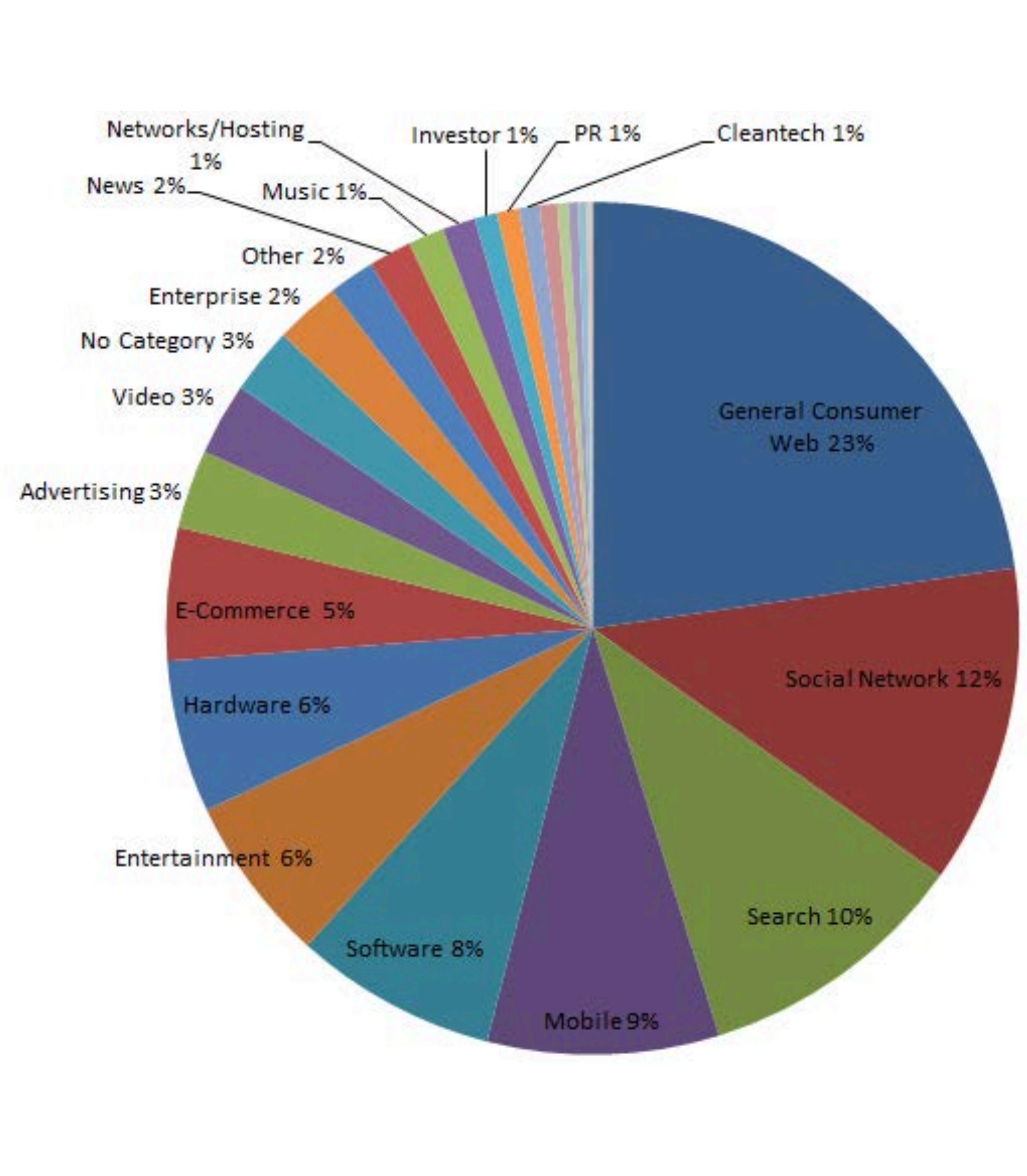
Other items not depicted include: onions (62%), chicken (56%), beef (36%), chillies (31%), jalapeños (30%), pork (25%), tuna (22%), anchovies (18%). 2% of people say they only like Margherita pizzas

YouGov | yougov.com

February 26-28, 2017

PIE CHARTS AND DONUT CHARTS

- 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

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

- Download the **pies** R Markdown file (save it in this week's folder in your class activities R project).

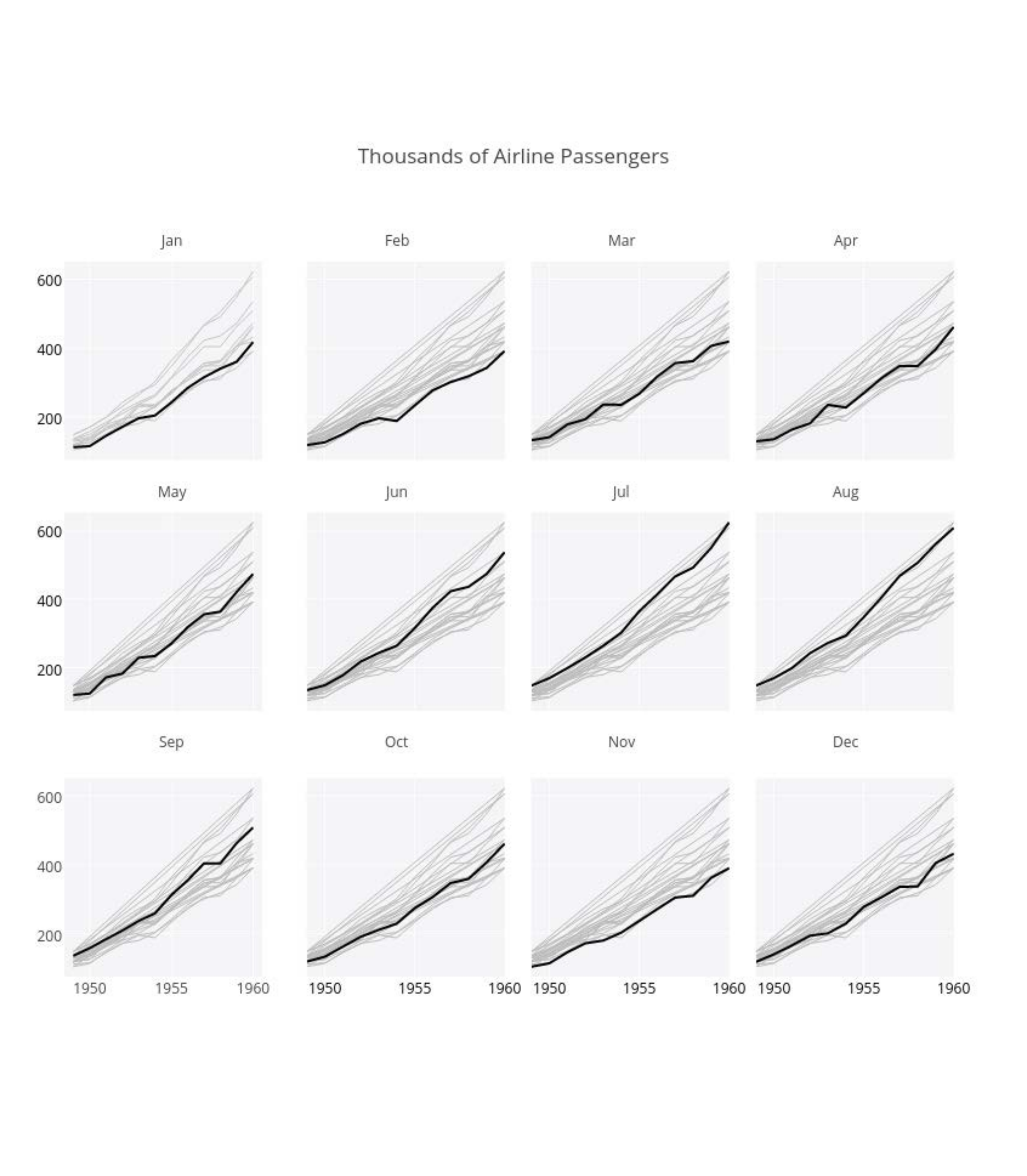
- Open the R Markdown file in R Studio.

- Follow the instructions to visualize answers from a multiple choice exam using pie charts.

20:00

SMALL MULTIPLES

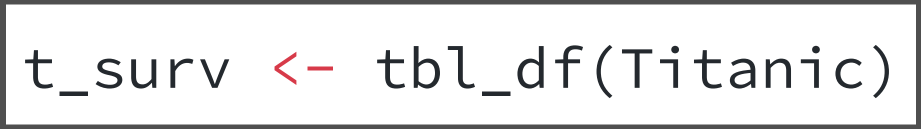
- Use the same basic graphic or chart to display different **slices** of a data set.
  - Indexed by category, time period, or some other variable not shown in chart
- Great way of showing complex data.



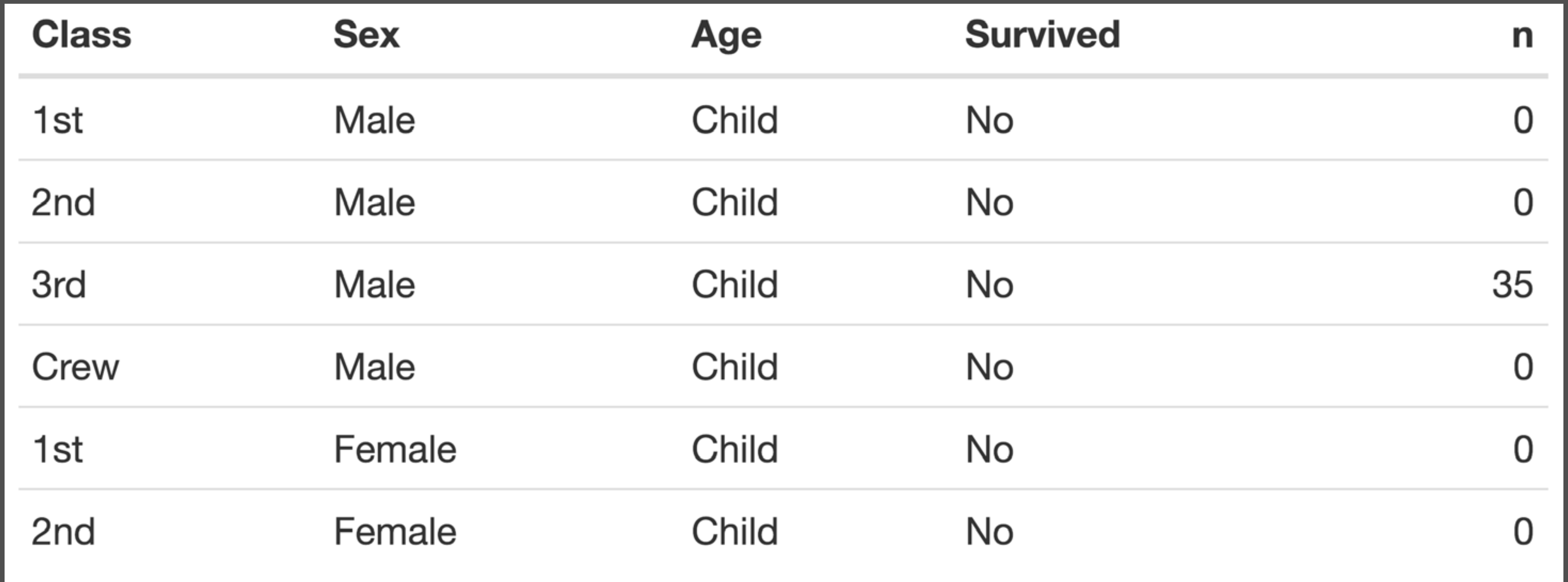
YOUR TURN

- Let's make small multiples out of the `Titanic` passenger survival dataset that comes with base R.

- Unfortunately, it's provided in a weird format. We can coerce it to a rectangular table with `tbl_df()`.



```
t_surv <- tbl_df(Titanic)
```

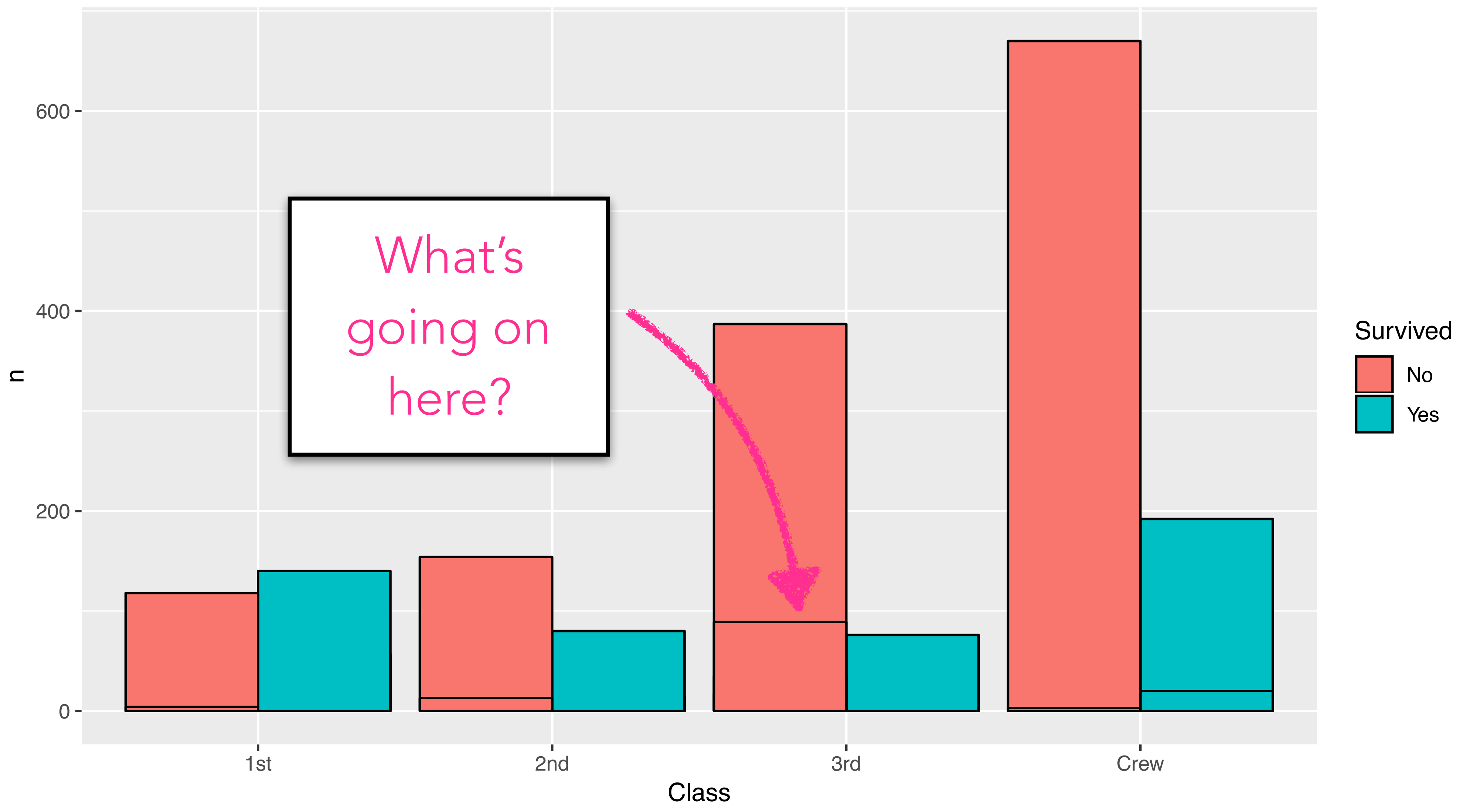


Class	Sex	Age	Survived	n
1st	Male	Child	No	0
2nd	Male	Child	No	0
3rd	Male	Child	No	35
Crew	Male	Child	No	0
1st	Female	Child	No	0
2nd	Female	Child	No	0

Note: summarized already!

- How did passenger survival vary with class? *Did this differ among age/sex categories?*

YOUR TURN



What's going on here?

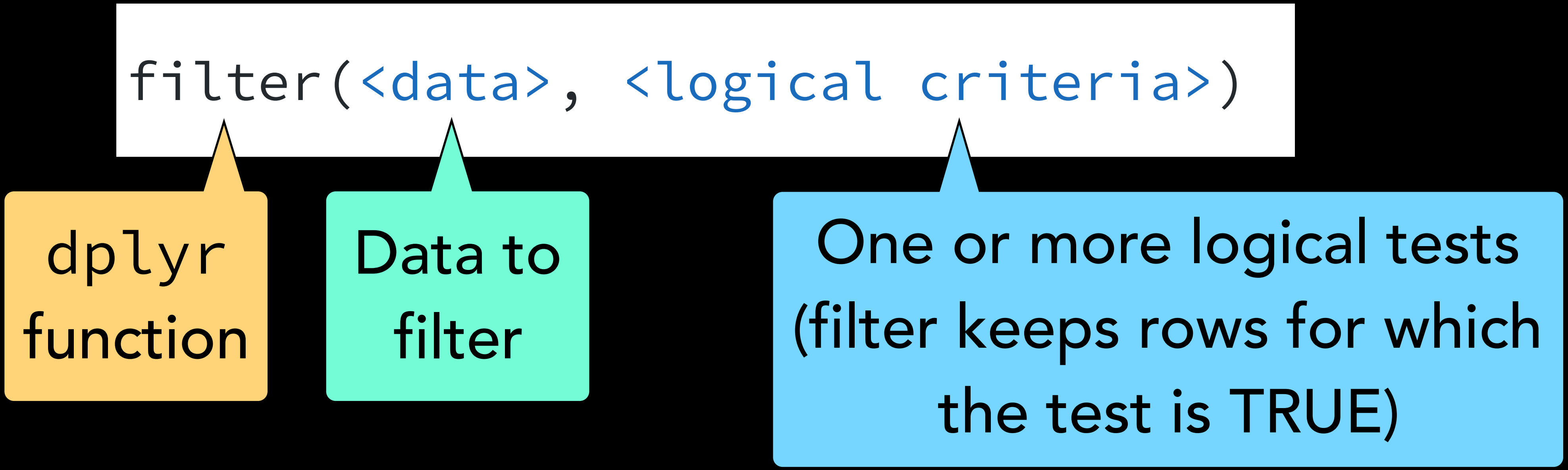
```
ggplot(t_surv, aes(x = Class, y = n, fill = Survived)) +
  geom_col(position = "dodge", color = "black")
```

SUBSETTING

- We need to show slices or subsets of data.
- Two approaches:
  - Filter the data so that fewer categories are plotted
  - Separate the plot into small multiples

SUBSETTING

- Filter the data so that fewer categories are plotted:



```
filter(<data>, <logical criteria>)
```

dplyr function

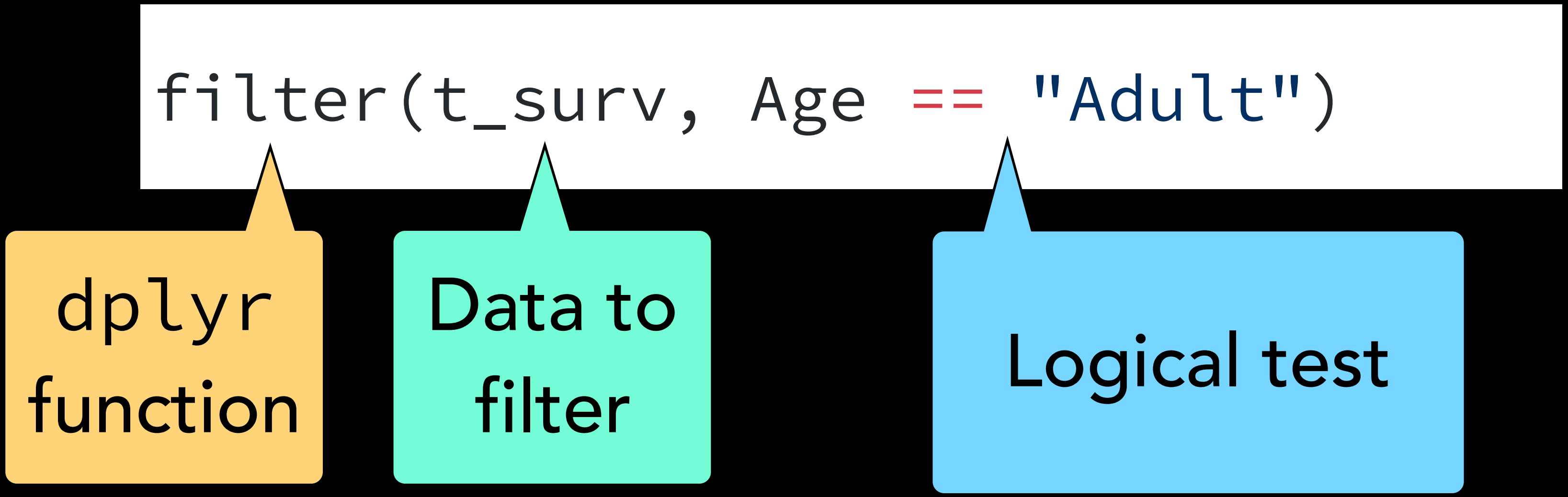
Data to filter

One or more logical tests (filter keeps rows for which the test is TRUE)

- Much more about this later. For now…

SUBSETTING

- Filter the data so that fewer categories are plotted:



```
filter(t_surv, Age == "Adult")
```

dplyr function

Data to filter

Logical test

SUBSETTING

- Filter the data so that fewer categories are plotted:

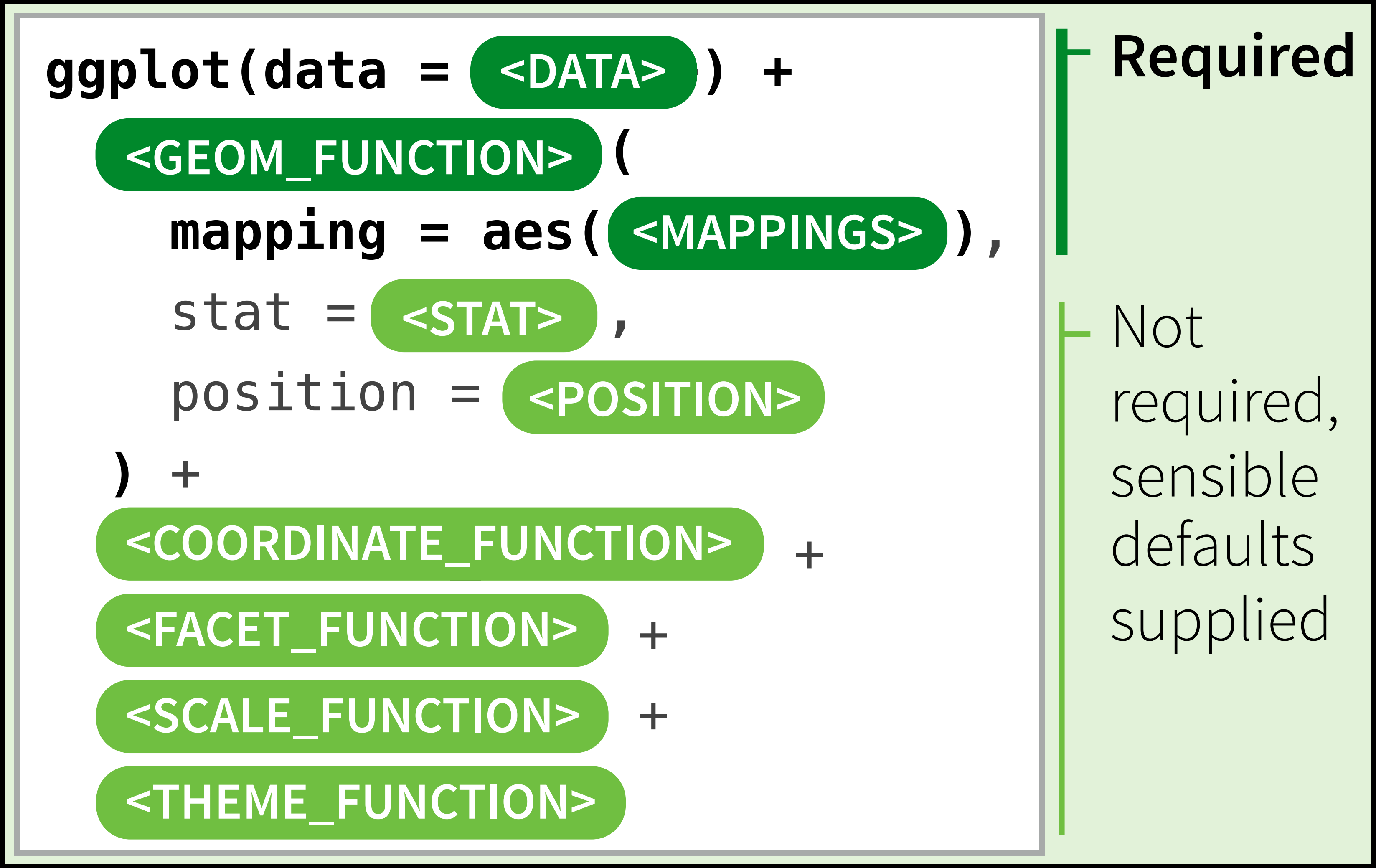
```
filter(t_surv, Age == "Adult")
```

= sets something
(returns nothing)

== tests if equal
(returns TRUE or FALSE)

SUBSETTING

- Separate the plot into small multiples (facets)



```
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

SUBSETTING

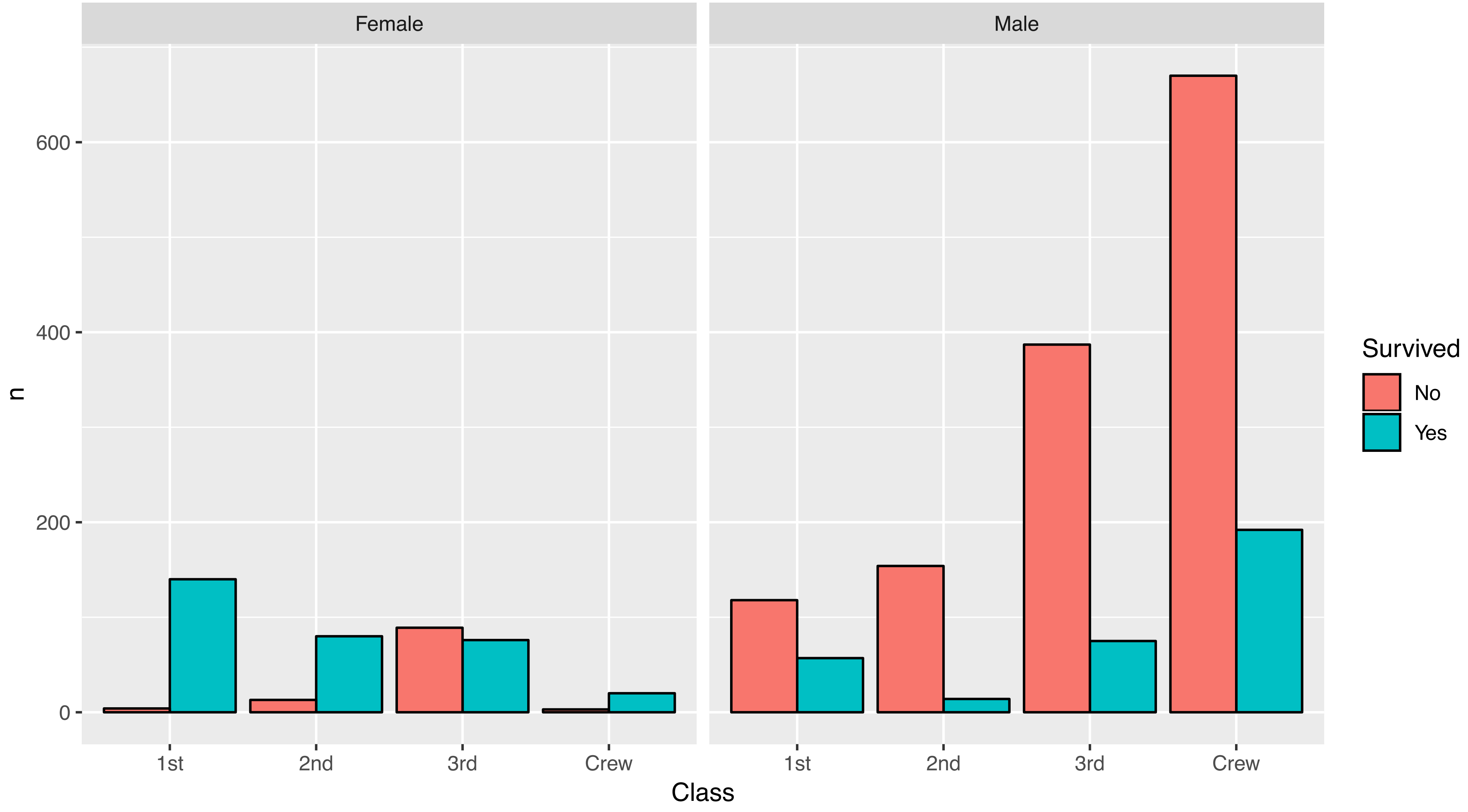
- Separate the plot into small multiples (facets)
  - `facet_wrap()`: creates a “ribbon” of panels (best used when splitting by one discrete variable)
    - `facet_wrap(~my_variable)`
  - `facet_grid()`: creates a “grid” of panels split by two discrete one variables
    - `facet_grid(var1 ~ var2)`

YOUR TURN

- Plot the titanic survival data again, but show only data for adults and facet by Sex

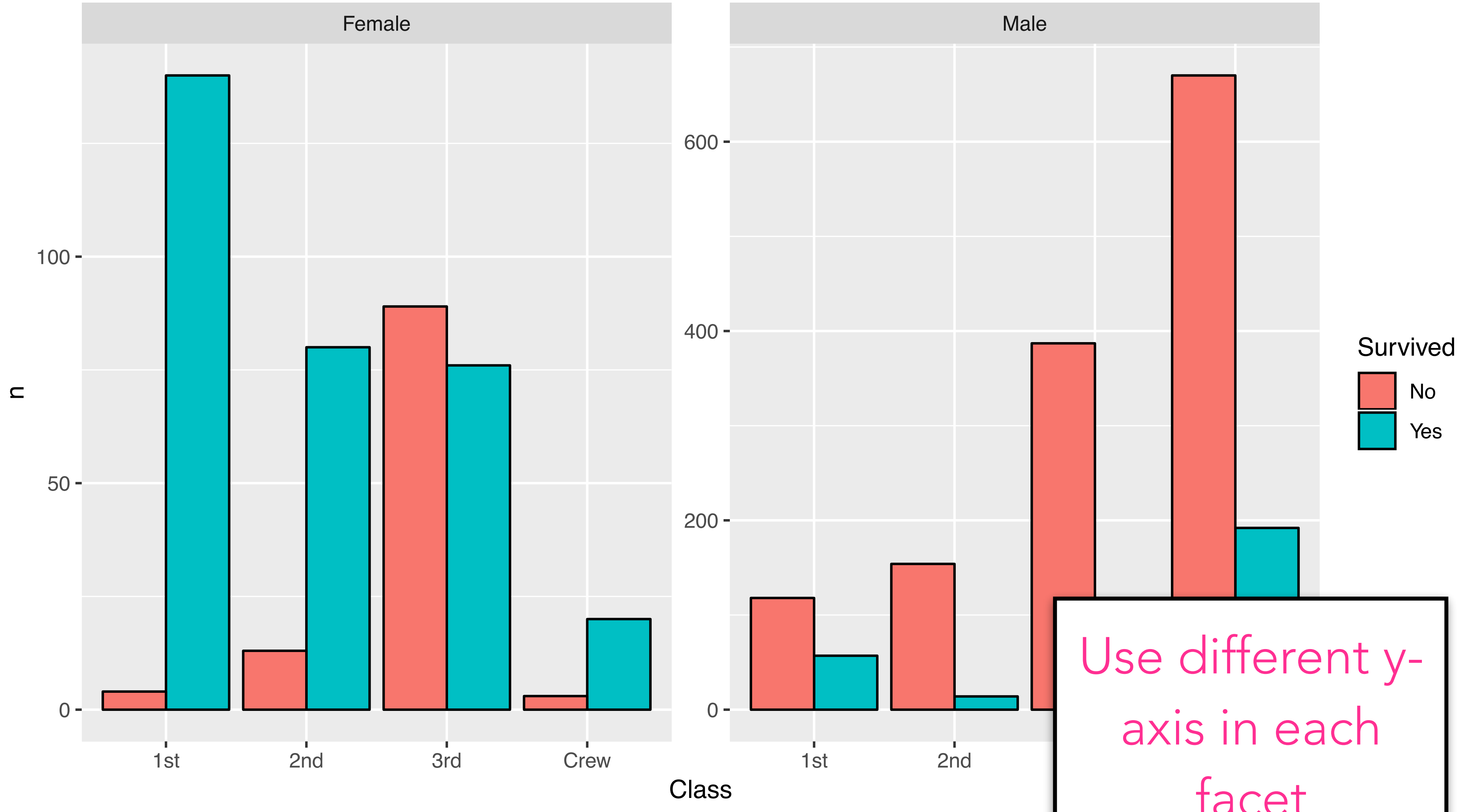
2:00

YOUR TURN



```
ggplot(filter(t_surv, Age == "Adult"),
       aes(x = Class, y = n, fill = Survived)) +
  geom_col(position = “dodge”, color = "black") +
  facet_wrap(~Sex)
```

YOUR TURN



Use different y-axis in each facet

```
ggplot(filter(t_surv, Age == "Adult"),
       aes(x = Class, y = n, fill = Survived)) +
  geom_col(position = "dodge", color = "black") +
  facet_wrap(~Sex, scales = "free_y")
```

YOUR TURN

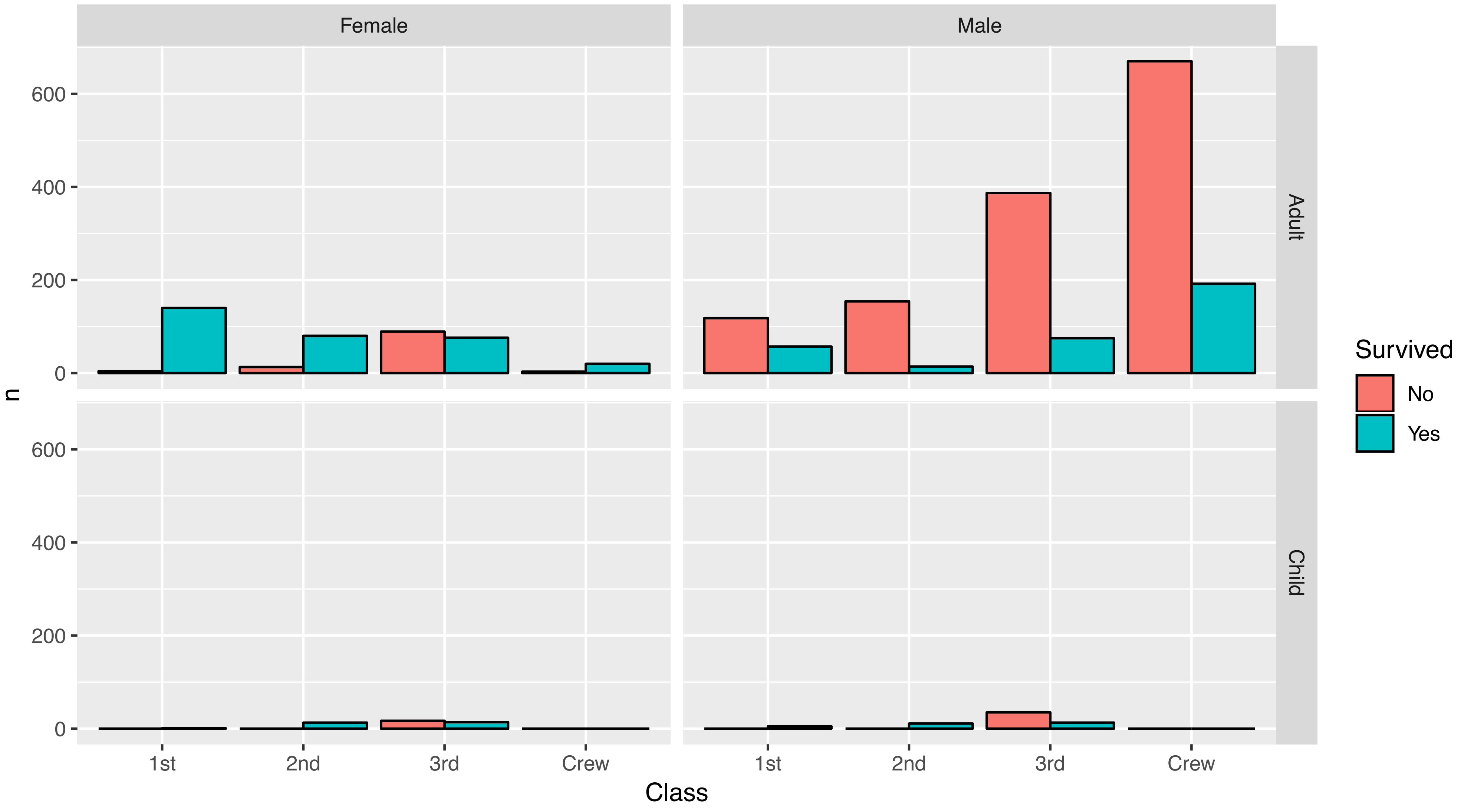
ggplot2

www.rstudio.com

- Plot **all** the titanic survival data again, but use facet grid to split apart the data by Age and by Sex

2:00

YOUR TURN



```
ggplot(t_surv, aes(x = Class, y = n, fill = Survived)) +
  geom_col(position = "dodge", color = "black") +
  facet_grid(Age ~ Sex)
```

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