

Visualisation: Miscellaneous useful stuff

Research Methods for Human Inquiry
Andrew Perfors

Putting figures in Markdown

Pretty straightforward - if it's in the code chunk, it will end up in your compiled document. The main thing to know is that you can change the figure size (and other factors, like alignment) using the arguments in the code chunk:

```
```{r boxplotpalette, warning=FALSE, message=FALSE, fig.width=10, fig.height=7}
dl %>%
 ggplot(mapping = aes(x = question, y = rating,
 fill = question, colour = question)) +
 geom_boxplot(alpha=0.4,show.legend=FALSE) +
 geom_jitter(alpha=0.7,show.legend=FALSE) +
 scale_fill_brewer(palette="Set1") +
 scale_color_brewer(palette="Set1") +
 facet_wrap(~species) +
 theme_bw() +
 labs(
 title = "Boxplot plot of ratings by species and type",
 y = "Rating",
 x = "Question type"
)
```

```

Saving figures as images

Suppose you want to save your figure as a png or other kind of file? Use the function `ggsave()`

`ggsave(filename="file.pdf", plot=plot_name, device="pdf")`

The name of the file you're saving. Defaults to saving it in your current directory. Add path information if you want to change that.

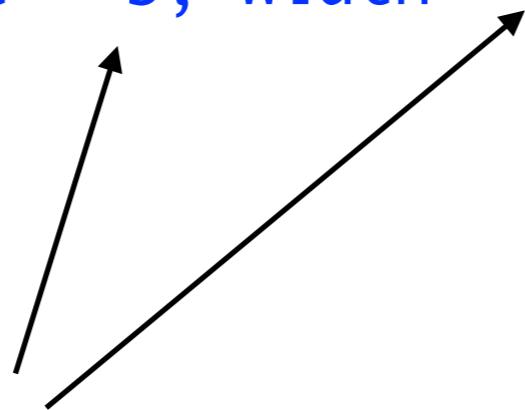
The name of the plot to save. If you don't specify this, it will default to the last plot you've made.

What kind of figure you want (e.g., pdf, jpeg, png, etc)

Saving figures as images

Suppose you want to save your figure as a png or other kind of file? Use the function `ggsave()`

```
ggsave( filename="file.jpg", plot=plot_name, device="jpeg",  
        height = 5, width = 8)
```

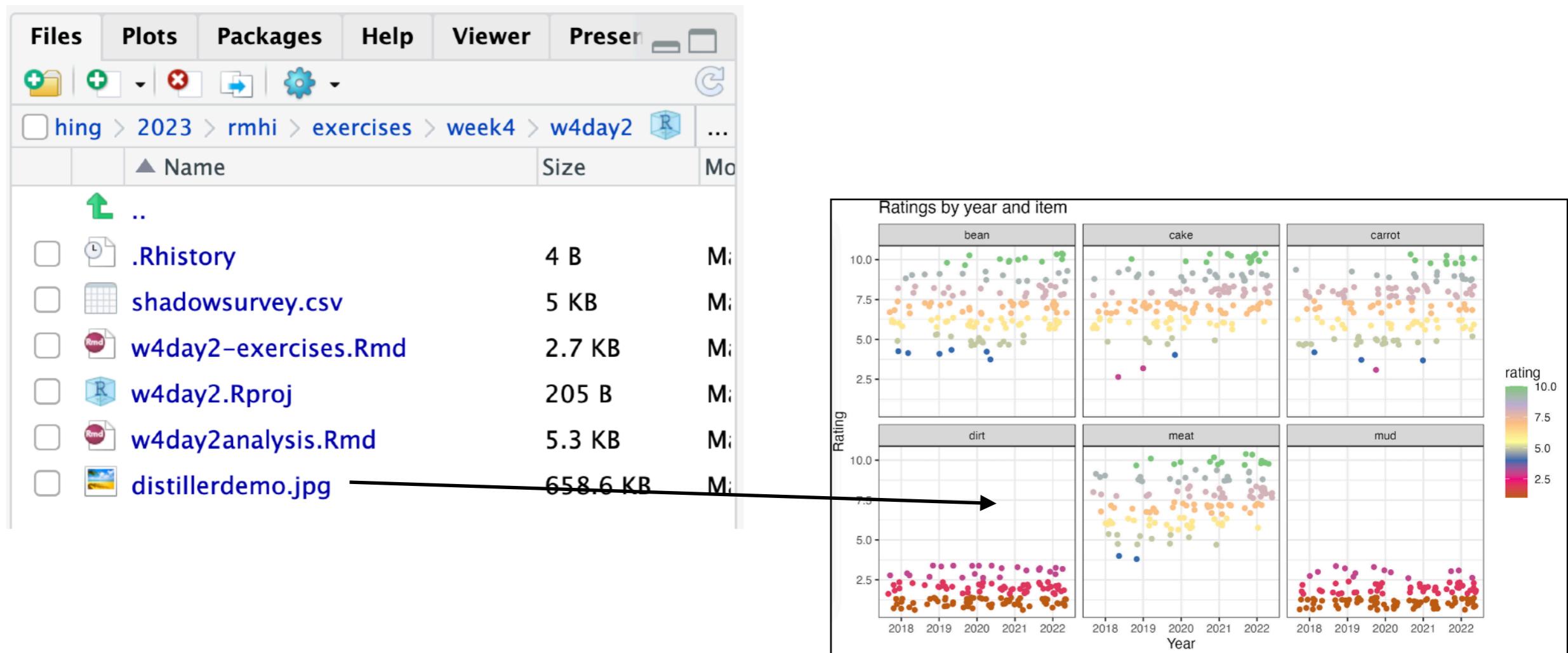


Nice to specify the size of
the figure. Units are
whatever is the default on
your device, but can specify
with the `units` argument

Saving figures as images

```
```{r savingplot, warning=FALSE, message=FALSE}
ggsave(filename="distillerdem.jpg", device="jpeg", height=5, width=8)
```

Saves the last plot as “distillerdem.jpg” in the current directory

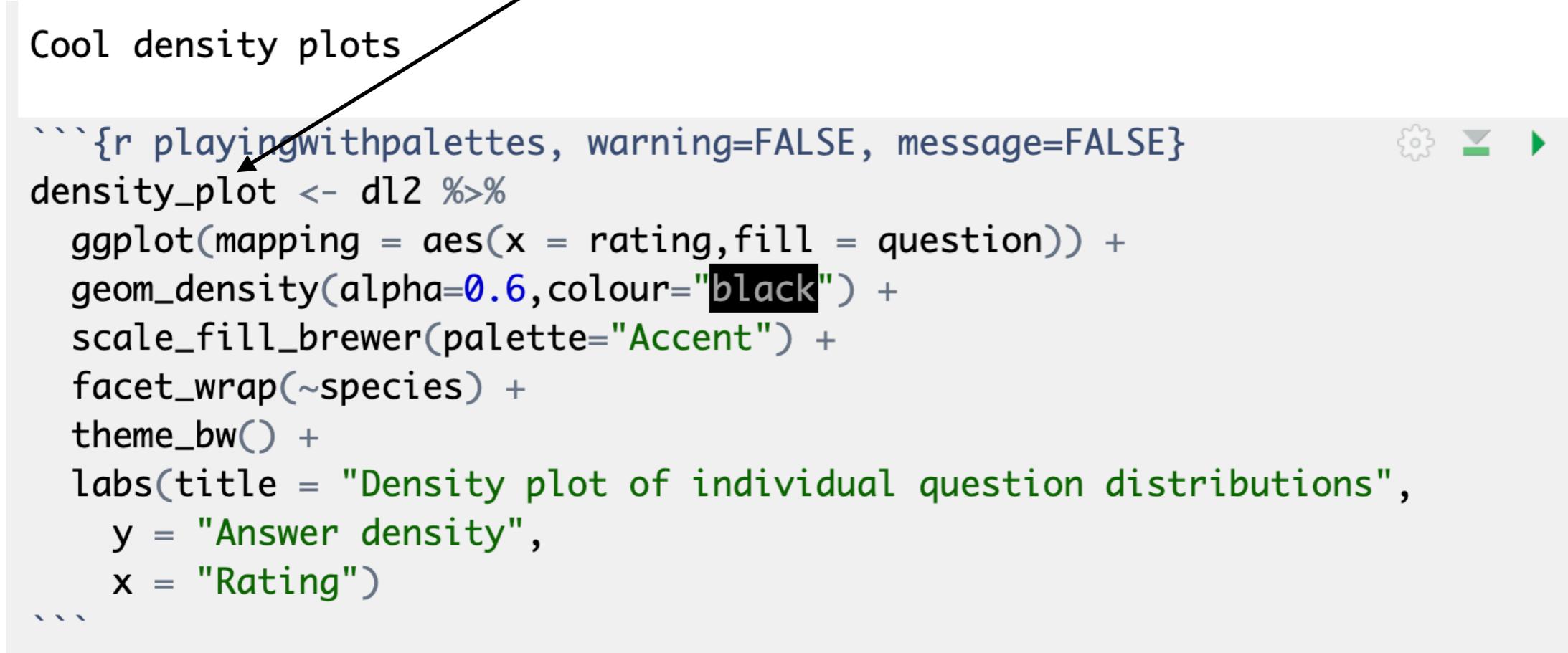


# Saving figures as images

It's usually a good idea to assign plots to variables so you can keep track of them and save them without confusion!

Assigning this to  
the density\_plot  
variable

Cool density plots

```
```{r playingwithpalettes, warning=FALSE, message=FALSE}
density_plot <- dl2 %>%
  ggplot(mapping = aes(x = rating, fill = question)) +
  geom_density(alpha=0.6, colour="black") +
  scale_fill_brewer(palette="Accent") +
  facet_wrap(~species) +
  theme_bw() +
  labs(title = "Density plot of individual question distributions",
       y = "Answer density",
       x = "Rating")
...```

```

Saving figures as images

It's usually a good idea to assign plots to variables so you can keep track of them and save them without confusion!

Now we can
save it!

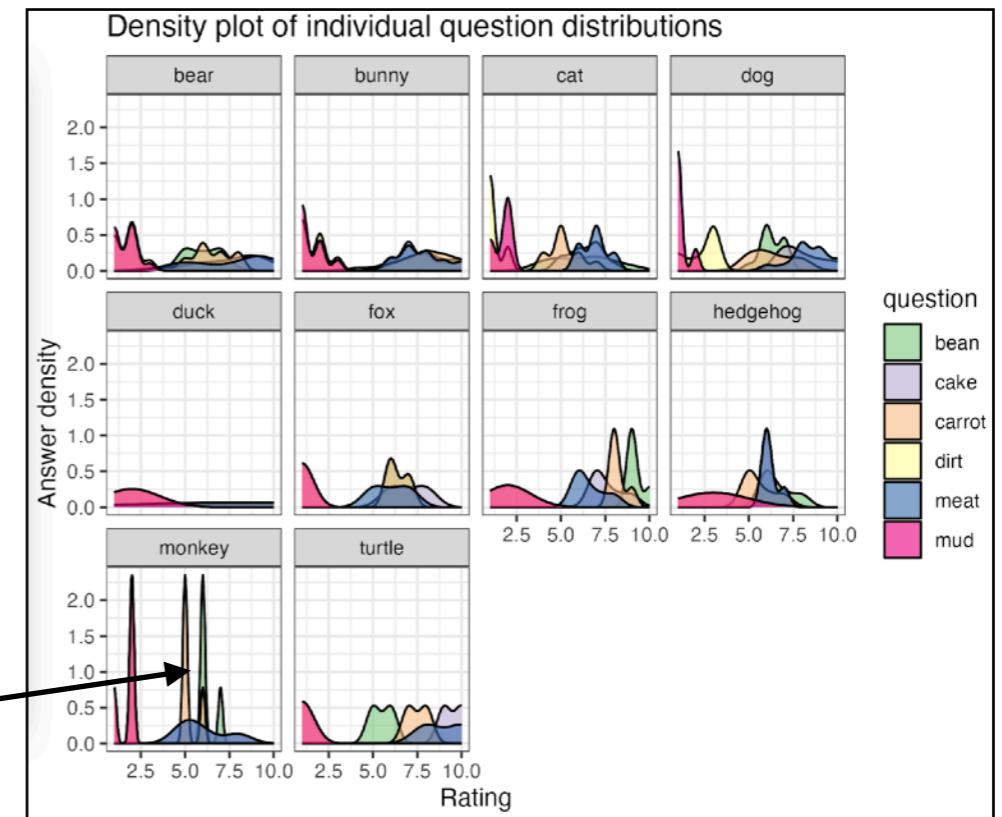
```
```{r savingplot, warning=FALSE, message=FALSE}
ggsave(filename="distillerdemo.jpg", device="jpeg", height=5, width=8)
ggsave(filename="densityplot.png", device="png", plot=density_plot,
 height=5, width=6)
...```

```

The screenshot shows the RStudio interface. On the left is the file browser with the following contents:

Name	Size	Modif.
..		
.Rhistory	4 B	Mon
shadowsurvey.csv	5 KB	Mon
w4day2-exercises.Rmd	2.7 KB	Mon
w4day2.Rproj	205 B	Mon
w4day2analysis.Rmd	5.3 KB	Mon
distillerdemo.jpg	805.8 KB	Mon
densityplot.png	225.8 KB	Mon

A black arrow points from the 'densityplot.png' entry in the file browser to the corresponding density plot shown on the right.



# Making bar plots with error bars

There's a geom for this, but it's a little different because instead of taking the entire dataset it needs the summary statistic directly (e.g. if you want to plot the means, it needs the means). Luckily we have the tools we need to do this!

```
dl_sum <- dl %>%
 group_by(question) %>%
 summarise(mean = mean(rating),
 sd = sd(rating),
 n = n(),
 sdErr = sd/sqrt(n)) %>%
ungroup()

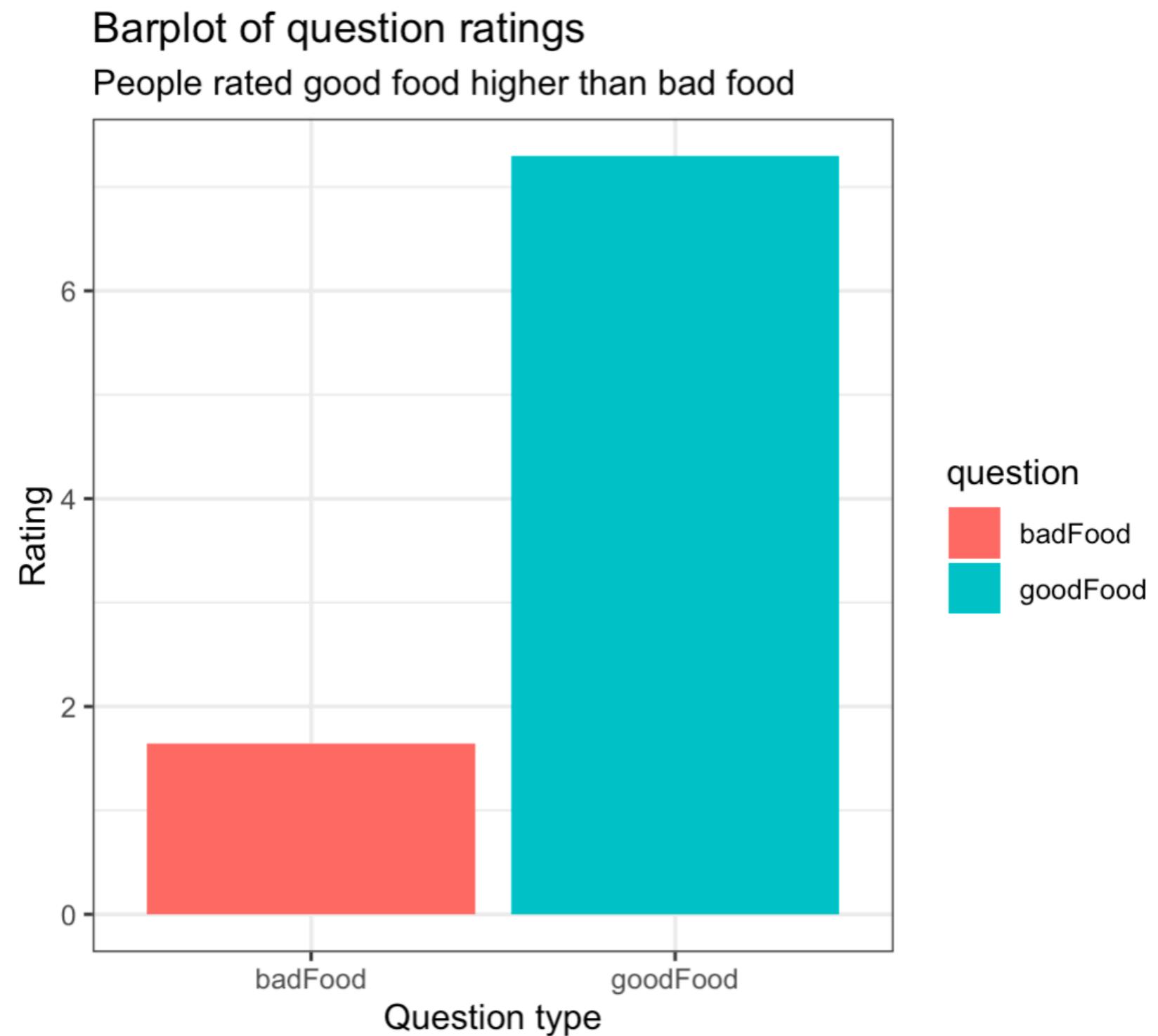
> dl_sum
A tibble: 2 x 5
 question mean sd n sdErr
 <chr> <dbl> <dbl> <int> <dbl>
1 badFood 1.64 0.604 135 0.0520
2 goodFood 7.29 1.06 135 0.0909
```

Bar plot uses `geom_col()` where the y value is the thing you just calculated

```
dl_sum %>%
 ggplot(mapping = aes(x = question,y = mean,fill = question)) +
 theme_bw() +
 geom_col() +
 labs(title = "Barplot of question ratings",
 subtitle = "People rated good food higher than bad food",
 y = "Rating",
 x = "Question type"
)
```

```
> dl_sum
A tibble: 2 x 5
 question mean sd n sdErr
 <chr> <dbl> <dbl> <int> <dbl>
1 badFood 1.64 0.604 135 0.0520
2 goodFood 7.29 1.06 135 0.0909
```

Bar plot uses `geom_col()` where the y value is the thing you just calculated



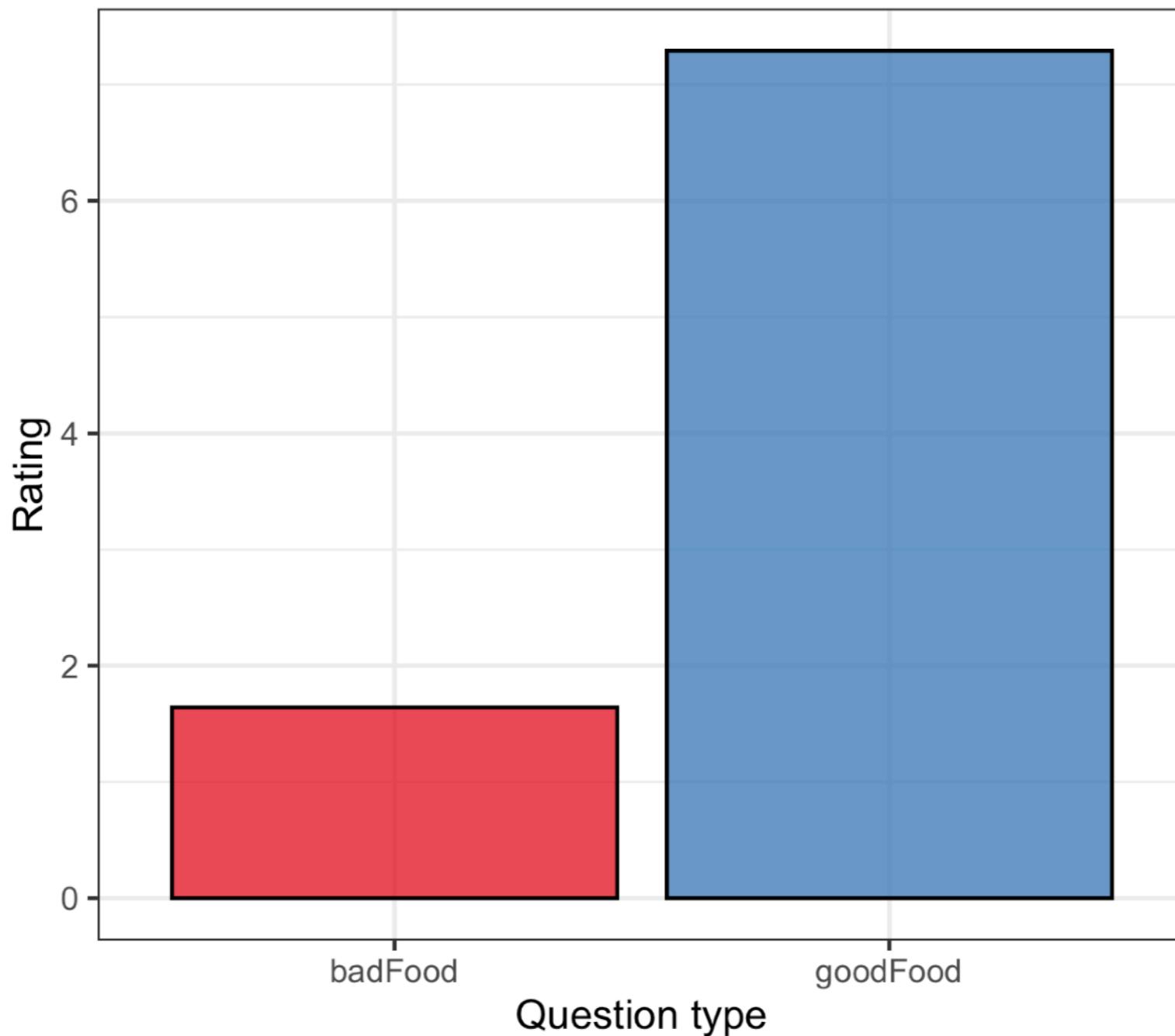
# Let's make this a little nicer...

```
dl_sum %>%
 ggplot(mapping = aes(x = question,y = mean,fill = question)) +
 geom_col(alpha=0.8,show.legend=FALSE,colour="black") +
 scale_fill_brewer(palette="Set1") +
 scale_colour_brewer(palette="Set1") +
 theme_bw() +
 labs(title = "Barplot of question ratings",
 subtitle = "People rated good food higher than bad food",
 y = "Rating",
 x = "Question type")
```

# Let's make this a little nicer...

Barplot of question ratings

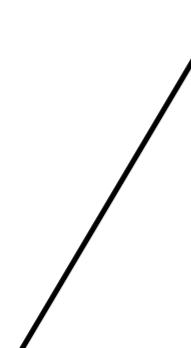
People rated good food higher than bad food



# But what about the error bars??

These use a different geom:

```
geom_errorbar(mapping = aes(ymin = mean - sdErr, ymax = mean + sdErr),
width=0.2) +
```



These are the values we calculated just now using the `summarise()` function!

Specifies that the width of the error bar is only 20% of the width of the bar

```
> dL_sum
A tibble: 2 × 5
 question mean sd n sdErr
 <chr> <dbl> <dbl> <int> <dbl>
1 badFood 1.64 0.604 135 0.0520
2 goodFood 7.29 1.06 135 0.0909
```

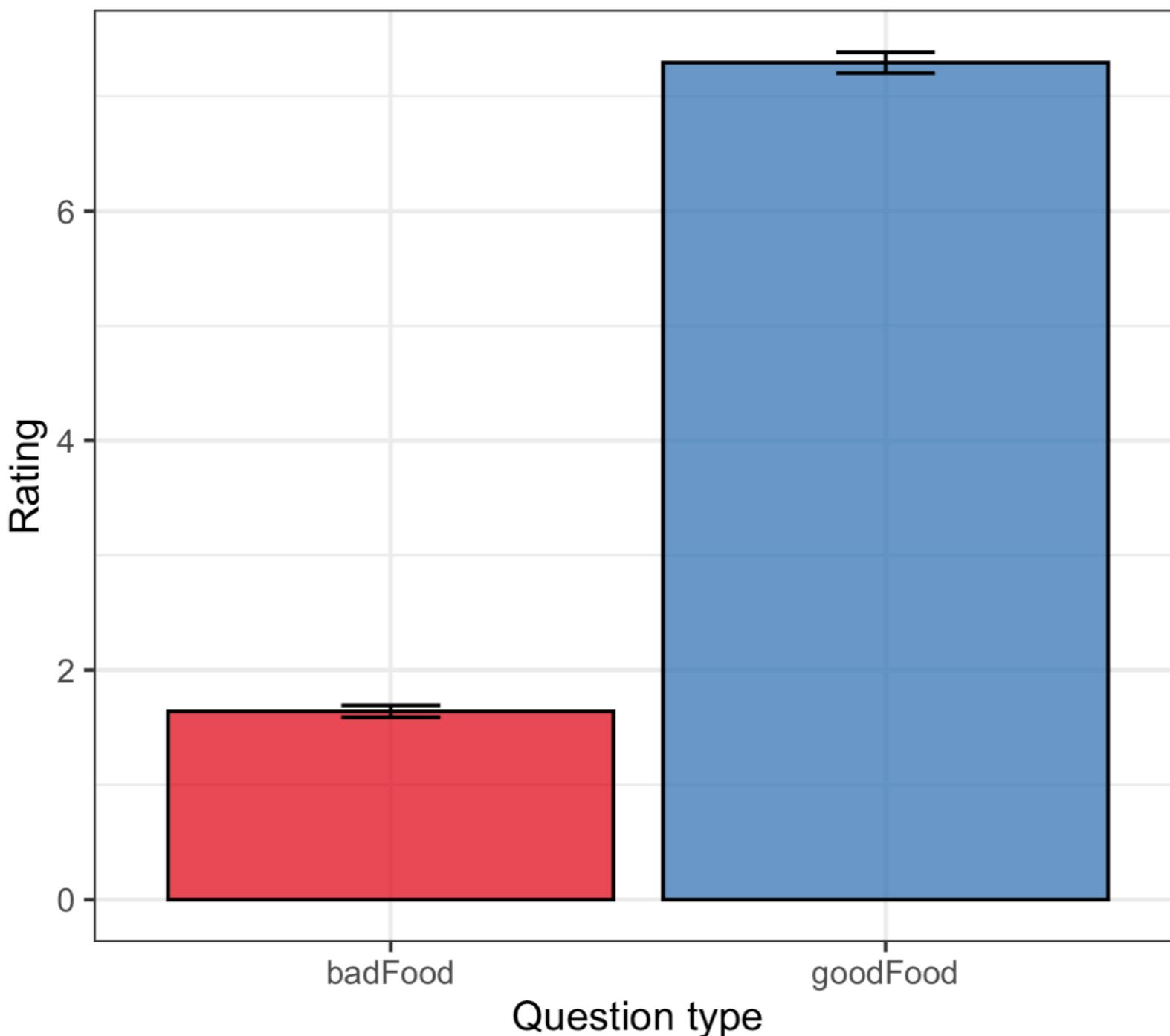
# But what about the error bars??

```
dl_sum %>%
 ggplot(mapping = aes(x = question,y = mean,fill = question)) +
 geom_col(alpha=0.8,show.legend=FALSE,colour="black") +
 geom_errorbar(mapping=aes(ymin = mean - sdErr,
 ymax = mean + sdErr), width=0.2) +
 scale_fill_brewer(palette="Set1") +
 scale_colour_brewer(palette="Set1") +
 theme_bw() +
 labs(title = "Barplot of question ratings",
 subtitle = "People rated good food higher than bad food",
 y = "Rating",
 x = "Question type")
```

# But what about the error bars??

Barplot of question ratings

People rated good food higher than bad food



# What if we want to layer multiple geoms?

This gets a little more complicated because they might be using slightly different data

`geom_col` and  
`geom_errorbar` are using  
the summary statistics we  
calculated:

```
> dl_sum
A tibble: 2 × 5
 question mean sd n sdErr
 <chr> <dbl> <dbl> <int> <dbl>
1 badFood 1.64 0.604 135 0.0520
2 goodFood 7.29 1.06 135 0.0909
```

Other geoms need the full  
dataset:

```
> head(dl)
A tibble: 6 × 6
 name gender species year question rating
 <chr> <chr> <chr> <fct> <chr> <dbl>
1 foxy female fox 2021 goodFood 7.25
2 foxy female fox 2021 badFood 1
3 bunny female bunny 2021 goodFood 9
4 bunny female bunny 2021 badFood 1
5 doggie male dog 2021 goodFood 8.75
6 doggie male dog 2021 badFood 1
```

# What if we want to layer multiple geoms?

For any geom you can manually give it a different dataset if you need to:

```
dl_sum %>%
 ggplot(mapping = aes(x = question,y = mean,fill = question)) +
 geom_col(alpha=0.8,show.legend=FALSE,colour="black") +
 geom_jitter(data=dl,
 mapping=aes(x=question,y=rating,colour=question),
 alpha=0.7,show.legend=FALSE) +
 geom_errorbar(mapping=aes(ymin = mean - sdErr,
 ymax = mean + sdErr), width=0.2) +
 scale_fill_brewer(palette="Set1") +
 scale_colour_brewer(palette="Set1") +
 theme_bw() +
 labs(title = "Barplot of question ratings",
 subtitle = "People rated good food higher than bad food",
 y = "Rating",
 x = "Question type")
```

# What if we want to layer multiple geoms?

For any geom you can manually give it a different dataset if you need to:

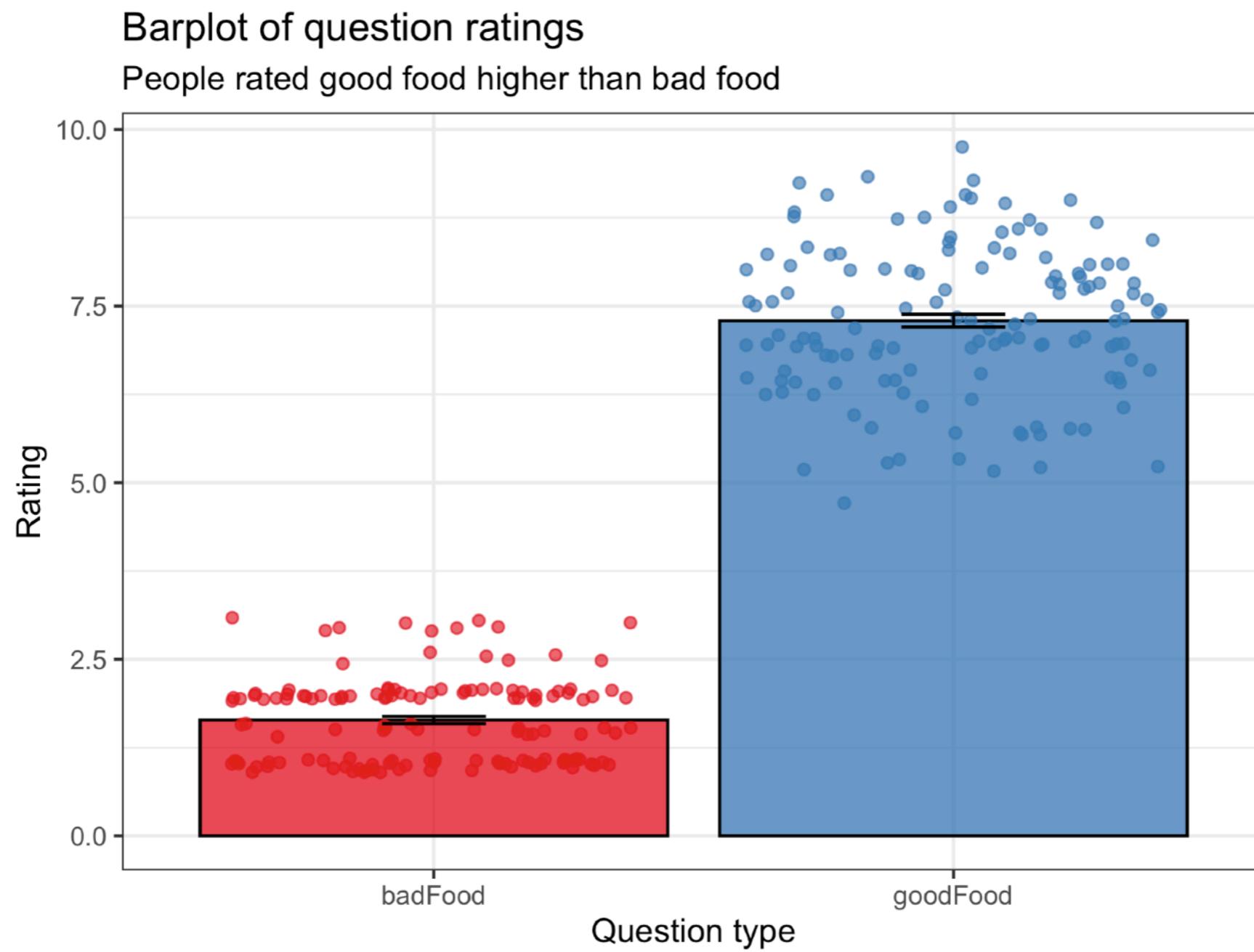
```
dl_sum %>%
 ggplot(mapping = aes(x = question,y = mean,fill = question)) +
 geom_col(alpha=0.8,show.legend=FALSE,colour="black") +
 geom_jitter(data=dl,
 mapping=aes(x=question,y=rating,colour=question),
 alpha=0.7,show.legend=FALSE) +
```

Overrides just for this  
geom the tibble `dl_sum`  
which is otherwise being  
passed to all of the other  
geoms

Note that it refers to the  
variables in the `dl` tibble,  
not `dl_sum`

# What if we want to layer multiple geoms?

For any geom you can manually give it a different dataset if you need to:





Do these plots help us to understand what's going on over time though?

Maybe not, but we now know (from this and the last video) that individual species vary as you'd expect for different foods, which suggests that it's not just about somebody's cooking getting better.



# How about this?

Calculates the mean and  
error bars for ratings  
broken down by question  
*and* year

```
dl_sum2 <- dl %>%
 group_by(year, question) %>%
 summarise(mean = mean(rating),
 sd = sd(rating),
 n = n(),
 sdErr = sd/sqrt(n)) %>%
 ungroup()
```

```
> dl_sum2
A tibble: 10 × 6
 year question mean sd n sdErr
 <fct> <chr> <dbl> <dbl> <int> <dbl>
1 2017 badFood 1.61 0.654 18 0.154
2 2017 goodFood 6.67 0.845 18 0.199
3 2018 badFood 1.67 0.701 23 0.146
4 2018 goodFood 6.87 1.06 23 0.222
5 2019 badFood 1.59 0.642 29 0.119
6 2019 goodFood 6.81 0.868 29 0.161
7 2020 badFood 1.61 0.535 32 0.0945
8 2020 goodFood 7.51 0.932 32 0.165
9 2021 badFood 1.71 0.559 33 0.0974
10 2021 goodFood 8.14 0.815 33 0.142
```

# How about this?

Calculates the mean and  
error bars for ratings  
broken down by question  
*and* year

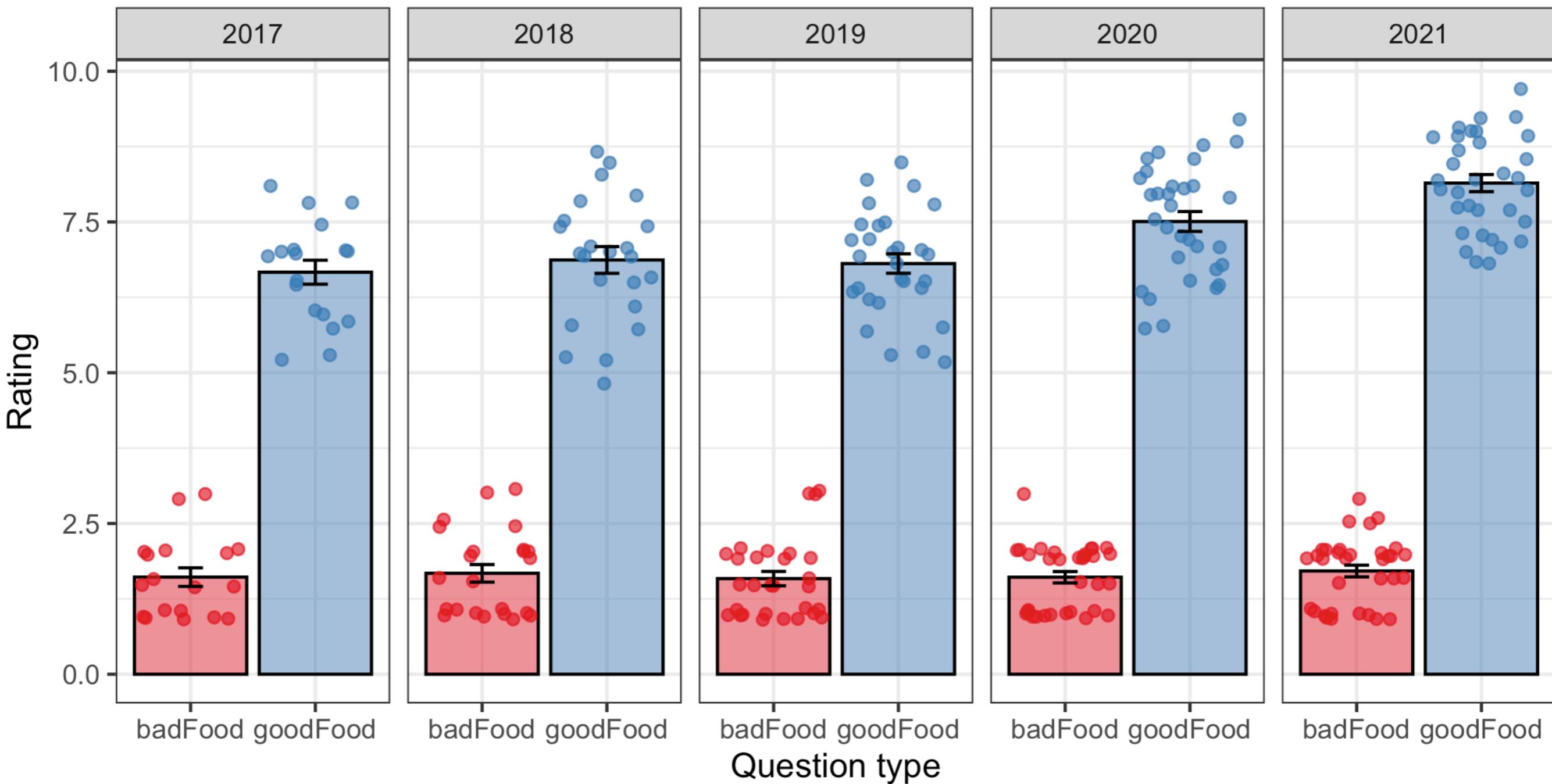
```
dl_sum2 <- dl %>%
 group_by(year, question) %>%
 summarise(mean = mean(rating),
 sd = sd(rating),
 n = n(),
 sdErr = sd/sqrt(n)) %>%
 ungroup()
```

```
dl_sum2 %>%
 ggplot(mapping = aes(x = question, y = mean, fill = question)) +
 geom_col(alpha=0.5, show.legend=FALSE, colour="black") +
 geom_jitter(data=dl, mapping=aes(x=question, y=rating, colour=question),
 alpha=0.7, show.legend=FALSE) +
 geom_errorbar(mapping=aes(ymin = mean - sdErr,
 ymax = mean + sdErr), width=0.2, show.legend=FALSE) +
 scale_fill_brewer(palette="Set1") +
 scale_colour_brewer(palette="Set1") +
 facet_wrap(~year, ncol=5) +
 theme_bw() +
 labs(title = "Barplot of question ratings",
 subtitle = "People rated good food higher than bad food",
 y = "Rating",
 x = "Question type")
```

# How about this?

## Barplot of question ratings

Only good food ratings got higher over time



# How about this?

These ratings do seem to be changing,  
especially on the higher end.

Hmmm.



See the `w4day2exercises.Rmd` file for  
the exercises!