Visualisation:
A grammar of graphics

Research Methods for Human Inquiry
Andrew Perfors

Today's story…

The animals are starting to fight about food

Today's story…

We're running out! There's a big problem!

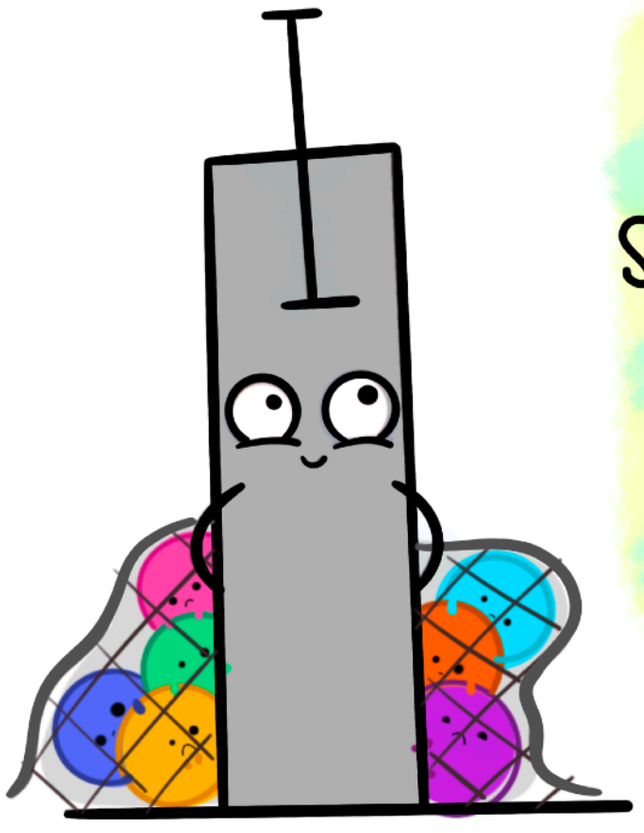
You guys are freaking out over nothing

Today's story…

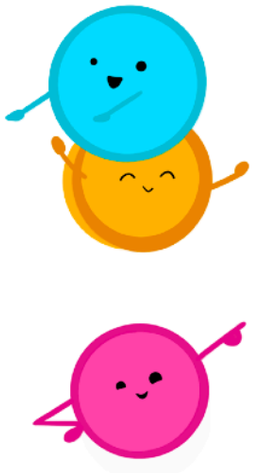
The data analysis we've done so far has been rather coarse:

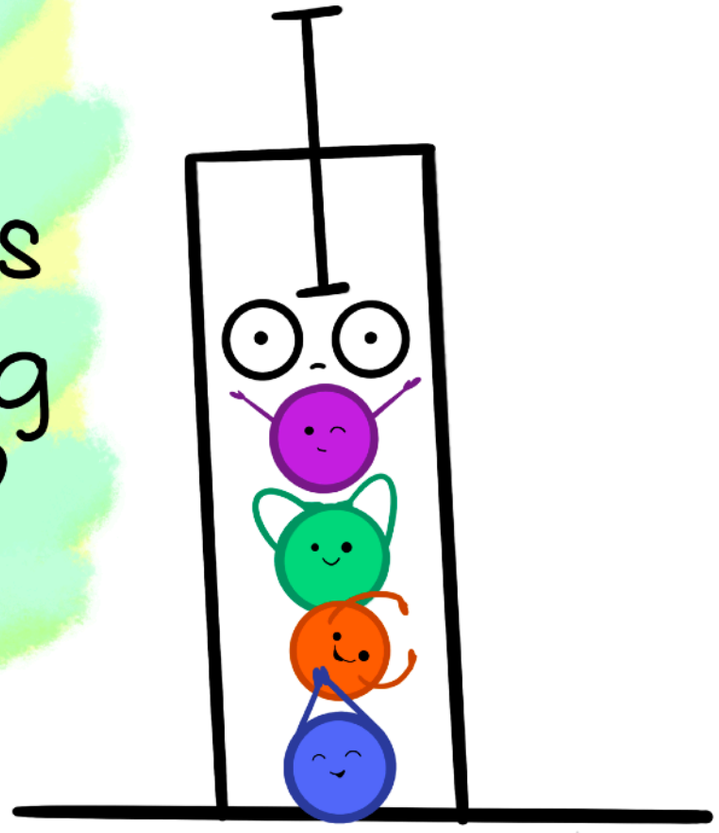
1. We don't know if anything is "significantly different" (what does that mean anyway?)

2. We're only getting rough summary statistics and have no real sense of the data and what's going on



are your summary statistics hiding something interesting?





Solution: Graphics

```
install.packages("ggplot2")
```

We're going to plot a bunch of things from Shadow's survey

I *highly highly* recommend you get in the habit of doing this thoroughly before running any tests on your data

Some good graphics principles

Most of this week is about how to make useful figures in R, but first let's talk about what kinds of things make a good figure in general

WARNING: RANT AHEAD

Nobody graphs things enough!

(If your study is well-designed and your effects are real and large enough, I think you can learn 90% of what you need to know **just from the figures**)

ALWAYS DRAW PICTURES
BEFORE YOU DO
ANYTHING ELSE

Example: how to stop a cholera outbreak

Person	Age	Occupation	Household size	Address	Sex	Cholera?
Mary Smith	12	child	8	7 Cross St	F	yes
Robert Plank	48	unemployed	5	12 King St	fair	yes
John WIlliams	7	child	12	16 Main St	good	no
Henry Locke	23	dockworker	9	24 King St	poor	yes
Elizabeth Gates	3	child	5	32 Banks St	poor	no
Jane Potter	29	homemaker	7	35 Cross St	fair	no

Pictures are a good idea

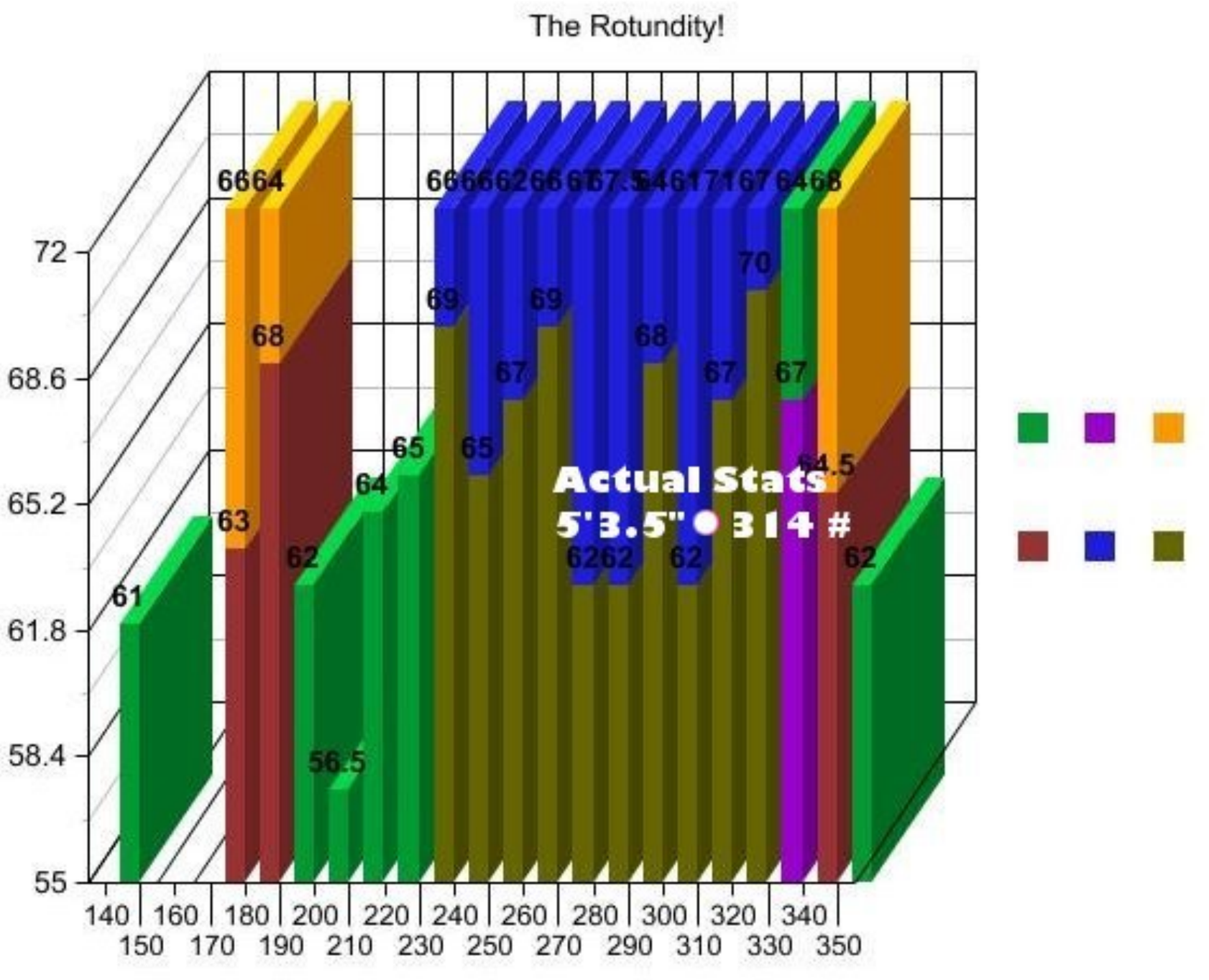
- Water pump
- Cholera case

Remove the handle from this pump

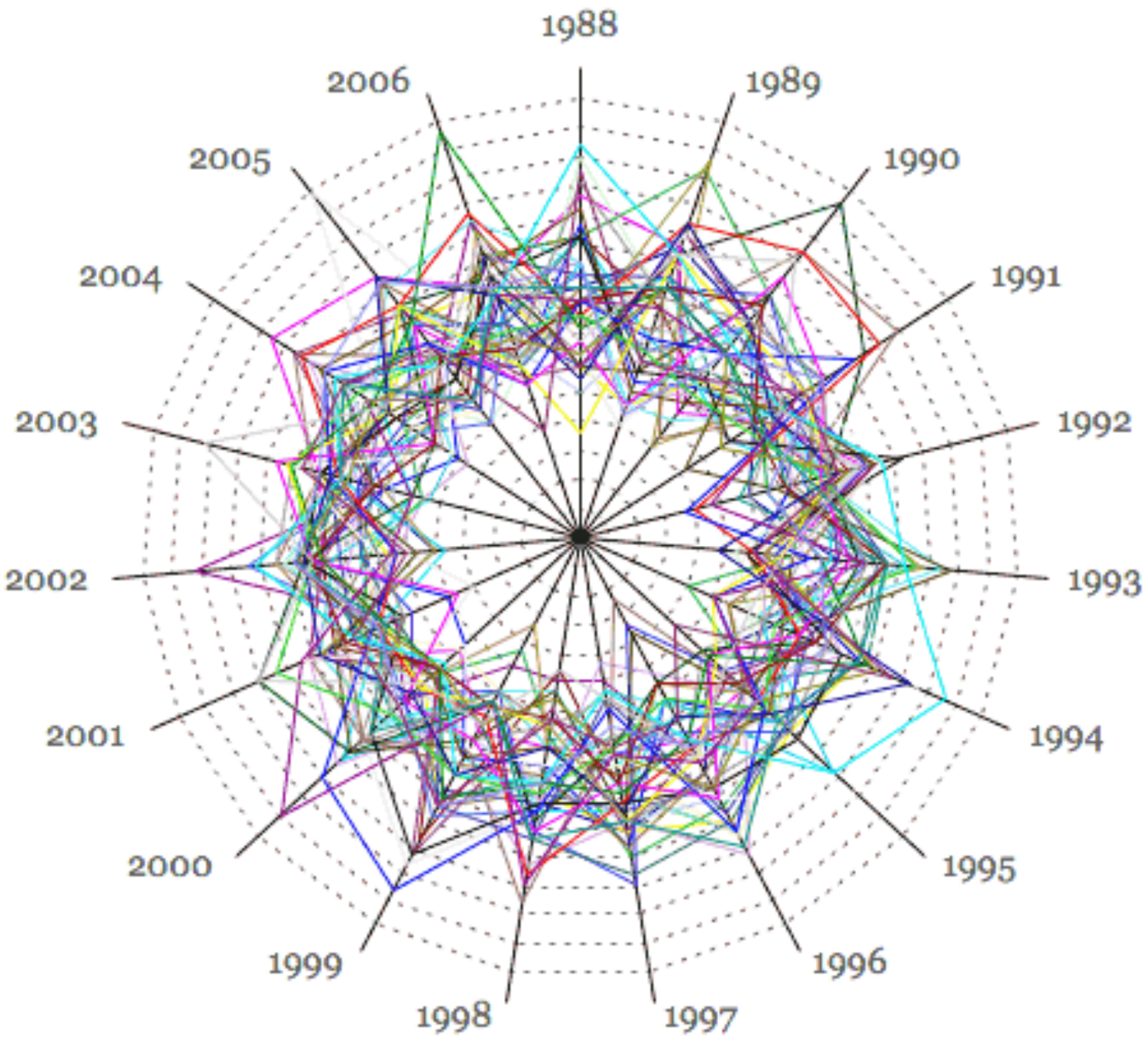
John Snow 1854

It’s easy to do it badly

I have no idea what this is saying



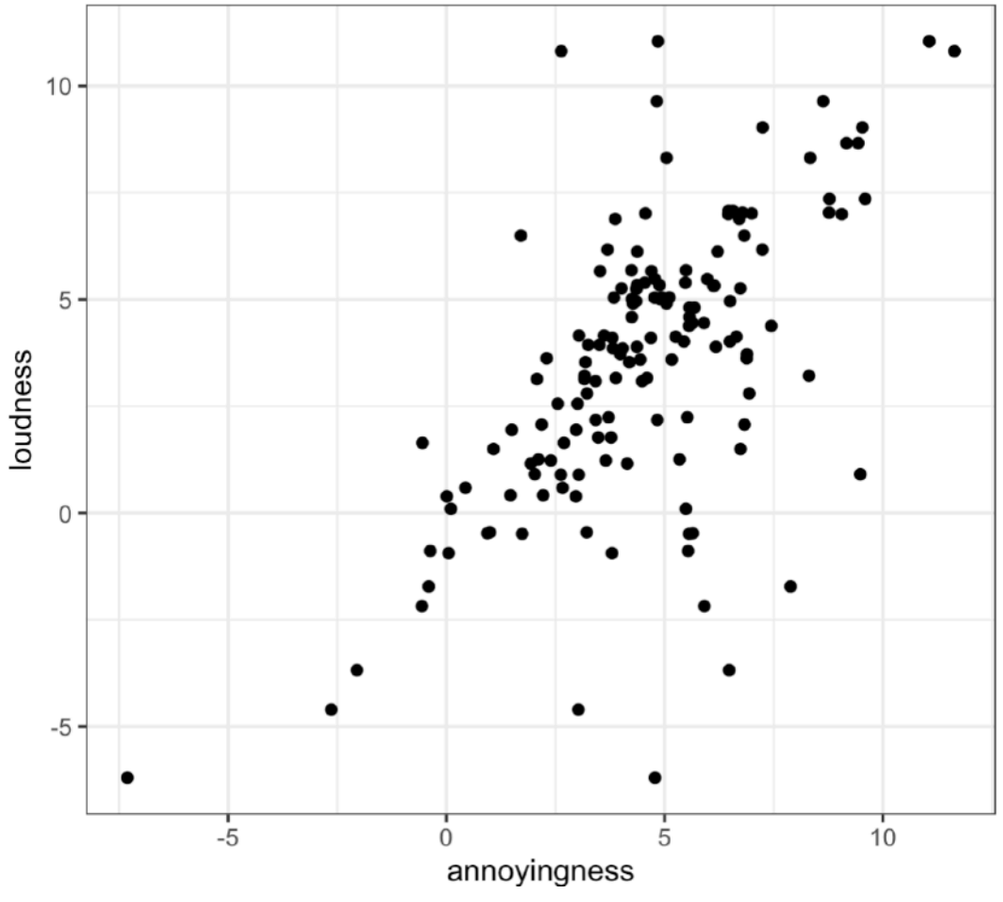
Unreadable, and implies time is stuck in a loop between 1988 and 2006



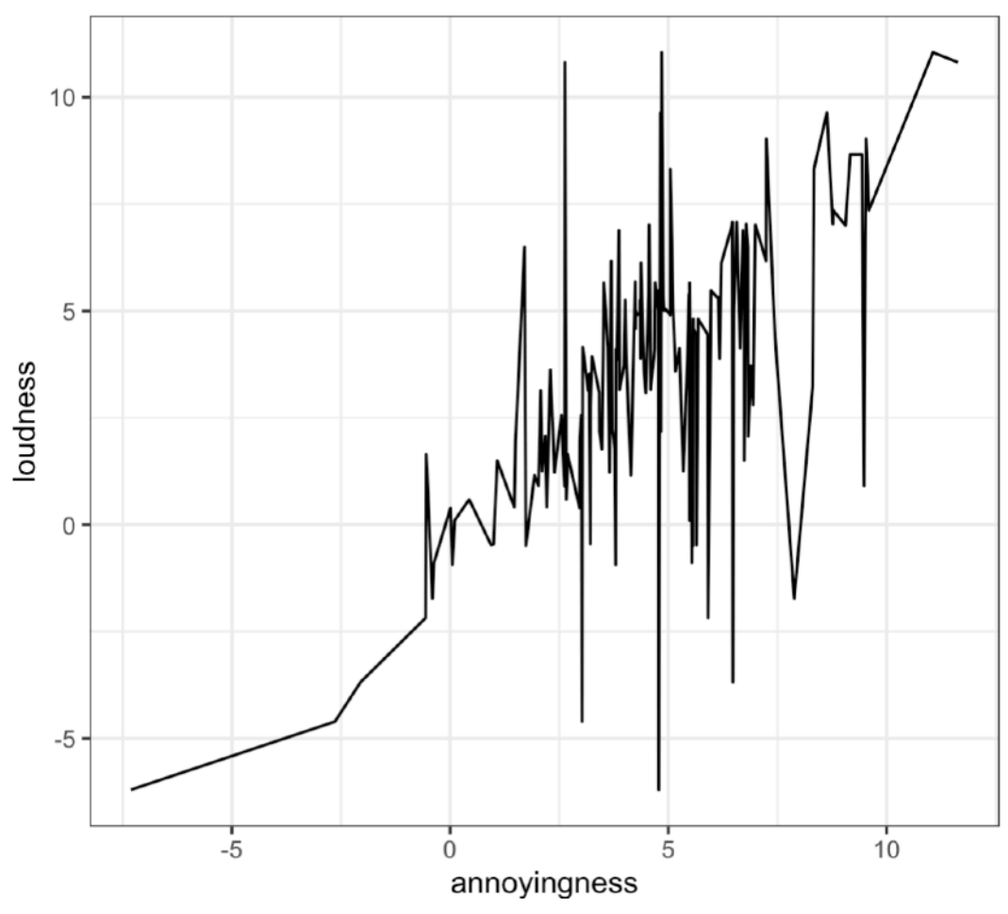
What properties does a good figure have?

- Figure type is chosen to isolate the thing you care about without being misleading

Good

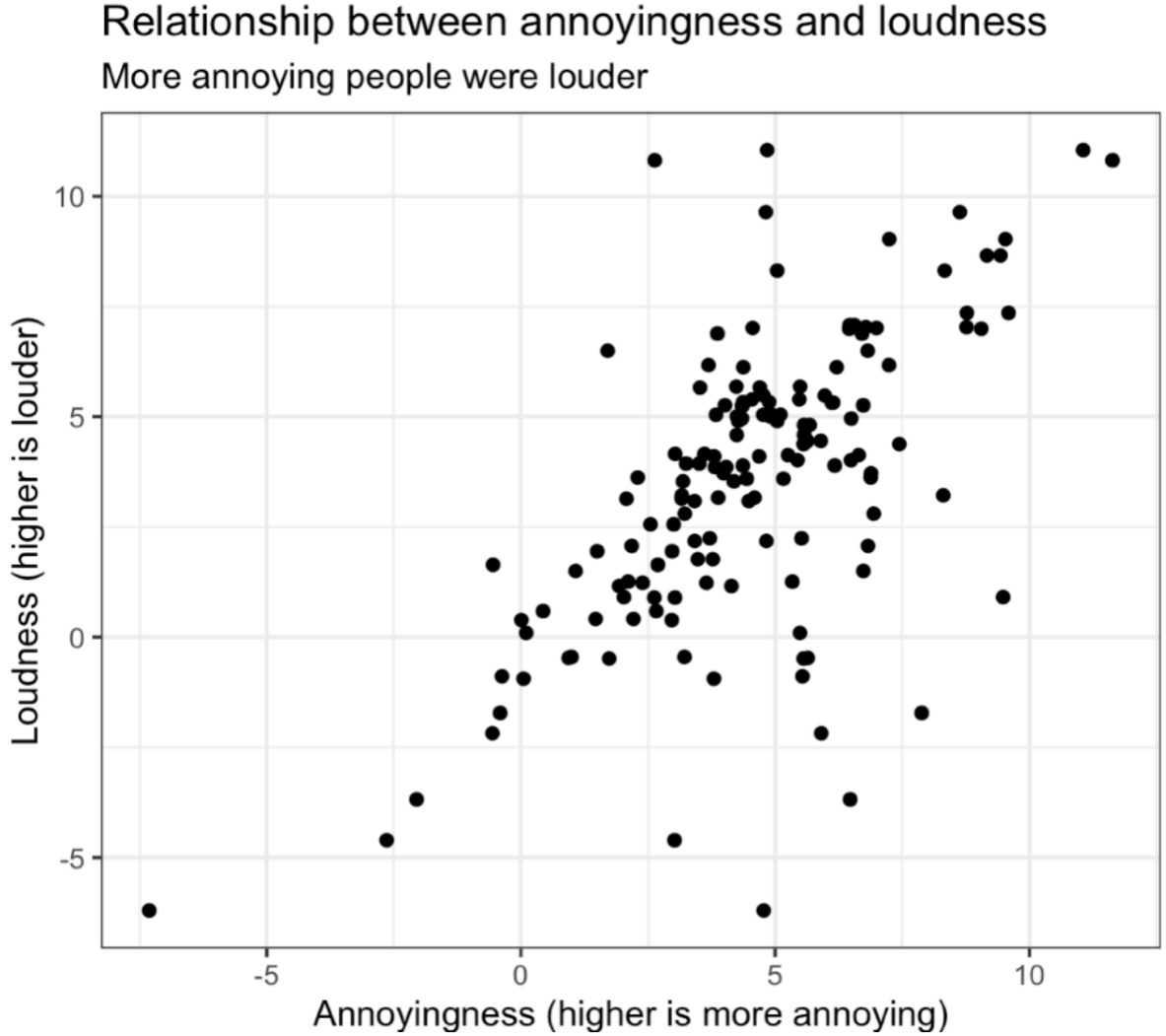


Implies nearby points are related



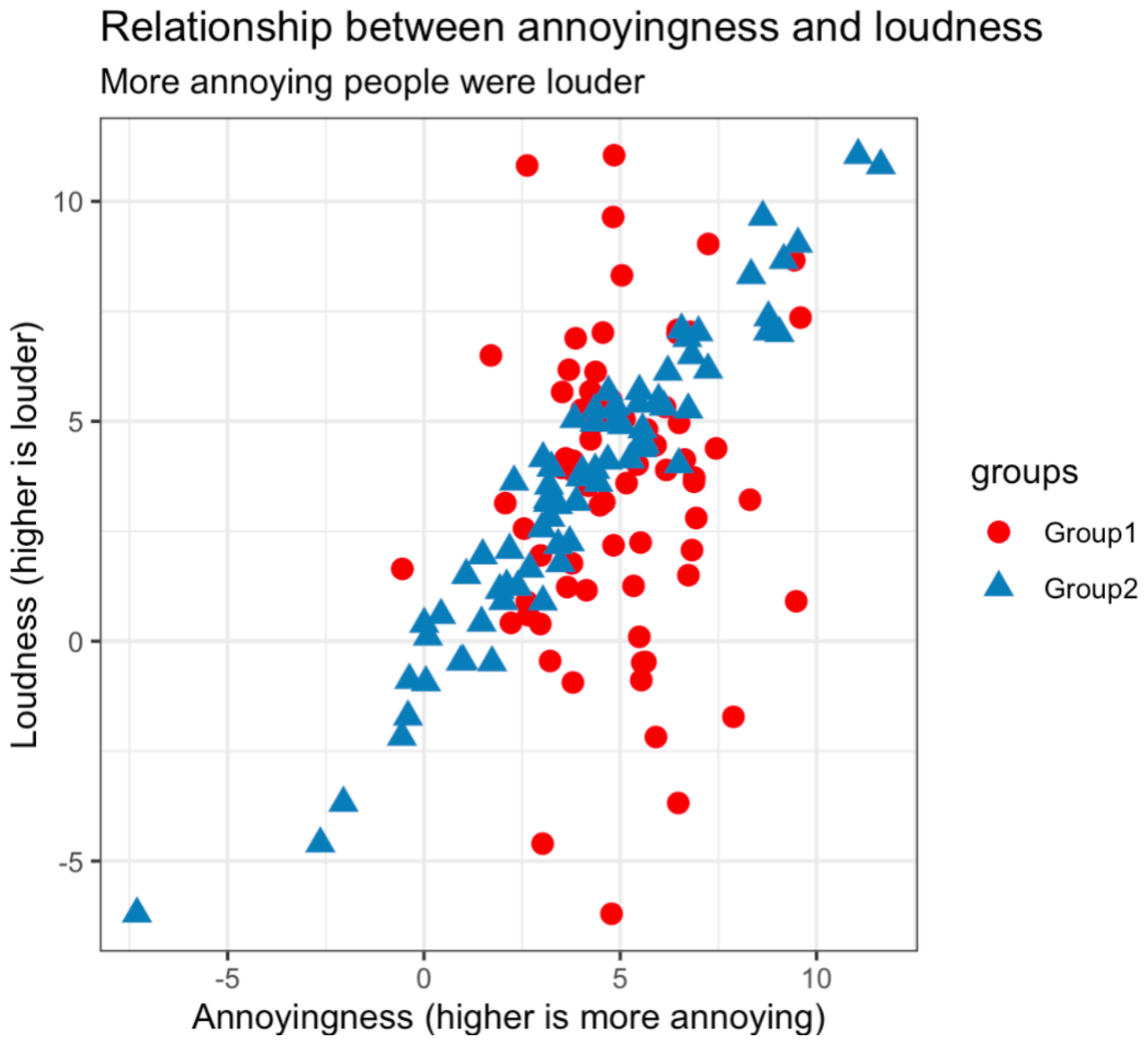
What properties does a good figure have?

- Axes are labeled clearly and informatively, title is descriptive, and fonts are large enough
  - I like using subtitles to explain in words the main finding



What properties does a good figure have?

- Important things are indicated in multiple ways where possible (e.g., colour, shape, hue) and in such a way as to be colour-blind and print-friendly



What properties does a good figure have?

aps
ASSOCIATION FOR PSYCHOLOGICAL SCIENCE

Psychological Science in the Public Interest
2021, Vol. 22(3) 110–161

DOI: 10.1177/15291006211051956
www.psychologicalscience.org/PSPI
SAGE

The Science of Visual Data Communication: What Works

Steven L. Franconeri[1], Lace M. Padilla[2], Priti Shah[3], Jeffrey M. Zacks[4], and Jessica Hullman[5]

[1]Department of Psychology, Northwestern University; [2]Department of Cognitive and Information Sciences, University of California, Merced; [3]Department of Psychology, University of Michigan; [4]Department of Psychological & Brain Sciences, Washington University in St. Louis; and [5]Department of Computer Science, Northwestern University

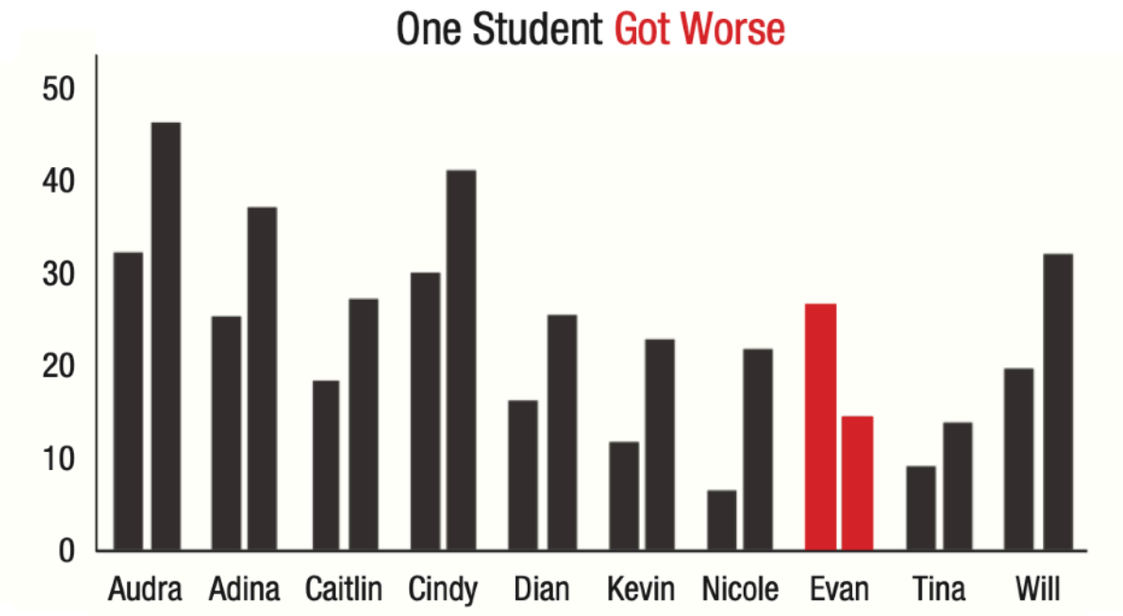
PNAS RESEARCH ARTICLE | PSYCHOLOGICAL AND COGNITIVE SCIENCES OPEN ACCESS Check for updates

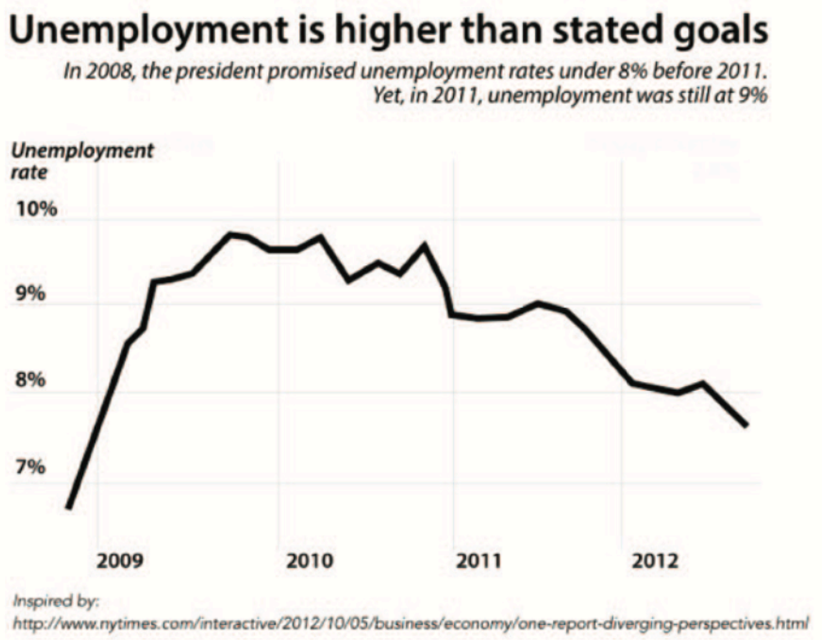
An illusion of predictability in scientific results: Even experts confuse inferential uncertainty and outcome variability

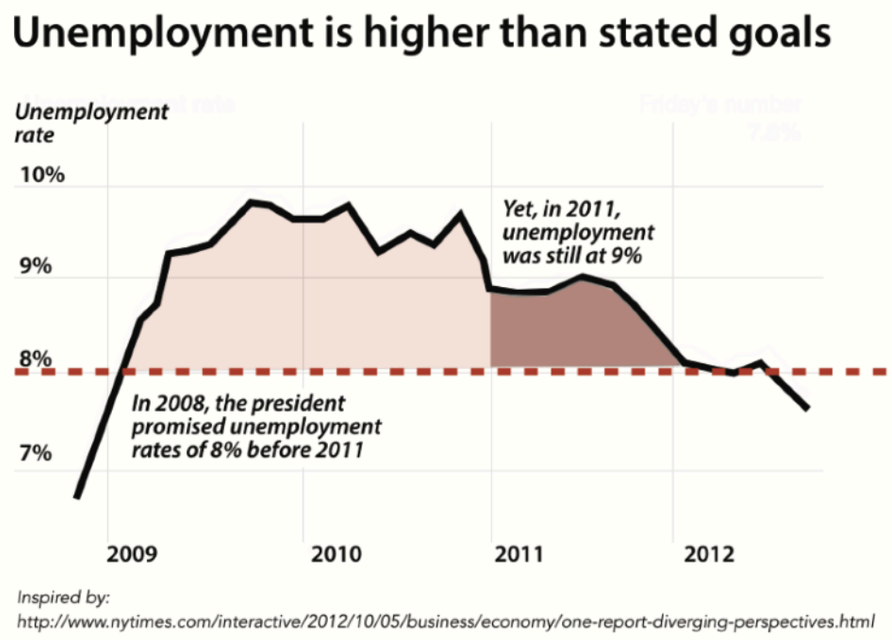
Sam Zhang[a,1], Patrick R. Heck[b], Michelle N. Meyer[c], Christopher F. Chabris[c], Daniel G. Goldstein[d], and Jake M. Hofman[d,1]

Edited by Elke Weber, Princeton University, Princeton, NJ; received February 22, 2023; accepted June 26, 2023

Use perceptual grouping and text to guide the eye to important things

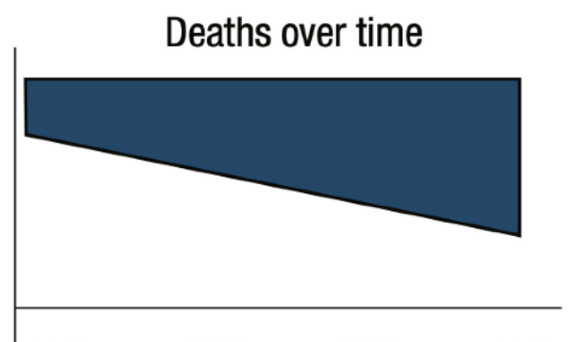




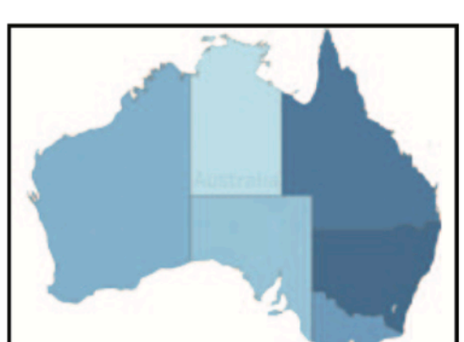


Use common shorthands

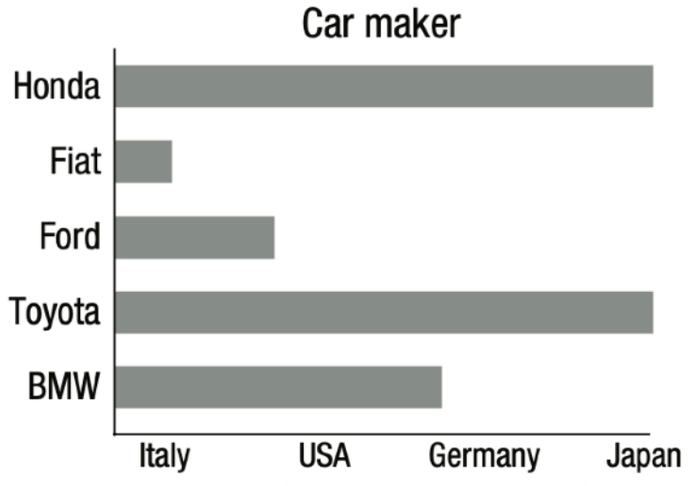
Common Confusions Caused by How Data Are Mapped to Visuals



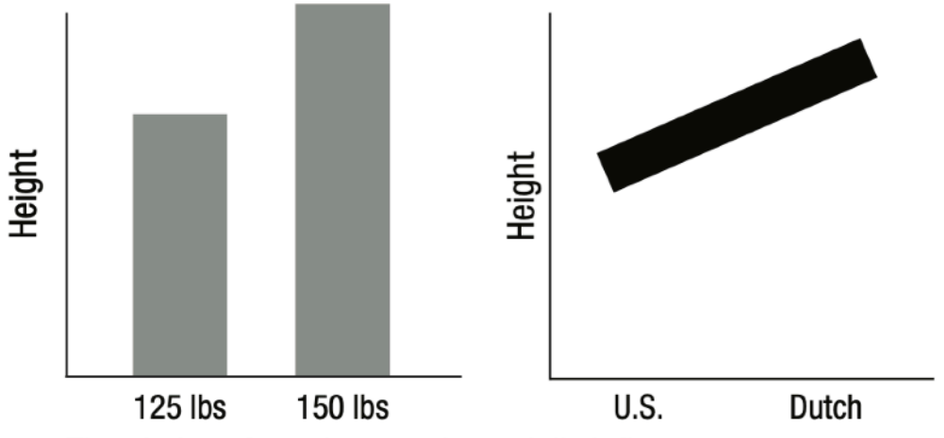
It can be confusing to map increases downward



For light backgrounds, darker colors clearly map to higher values. For dark backgrounds, it's not so clear.

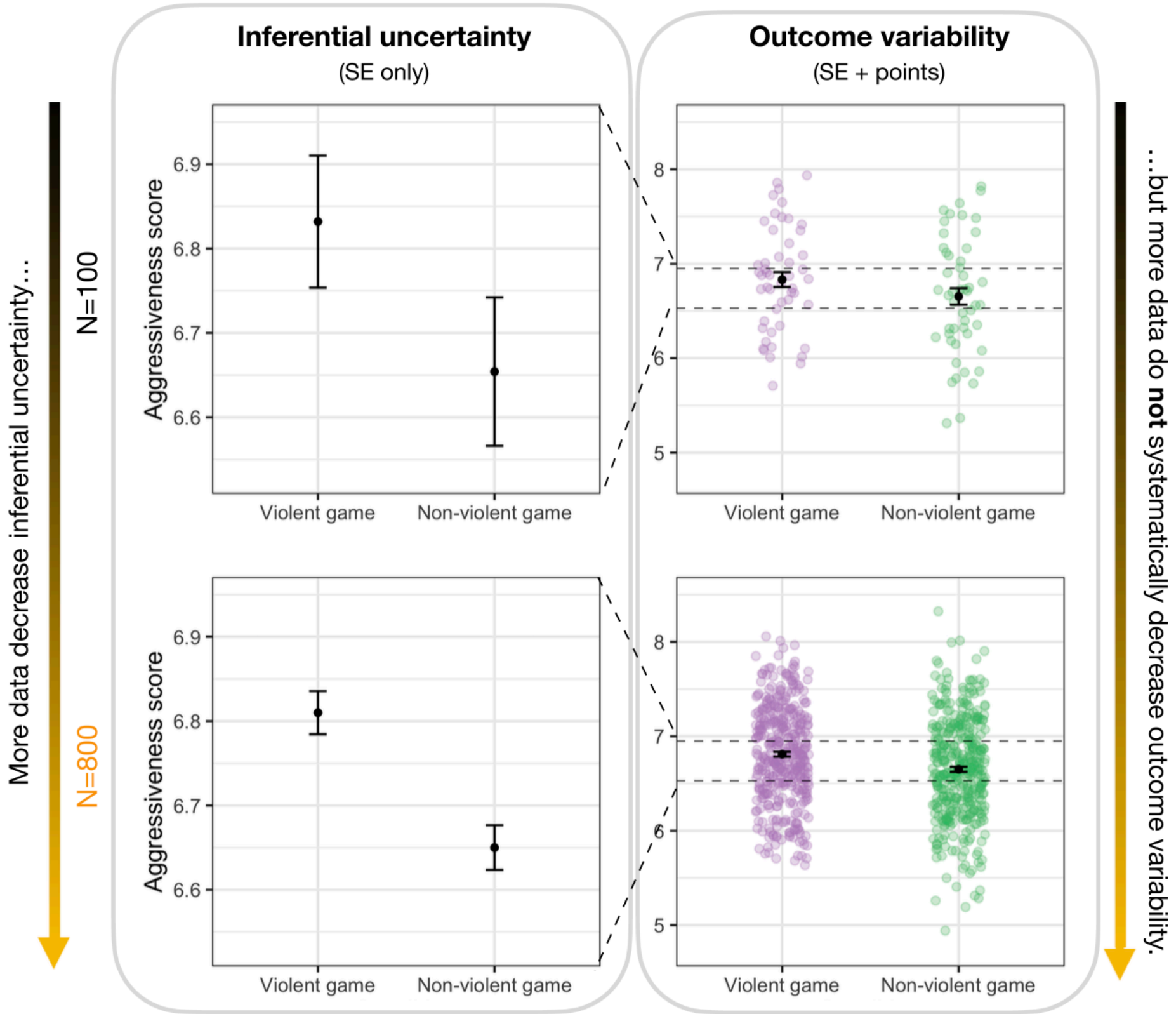


It can be confusing to map nominal values to magnitudes

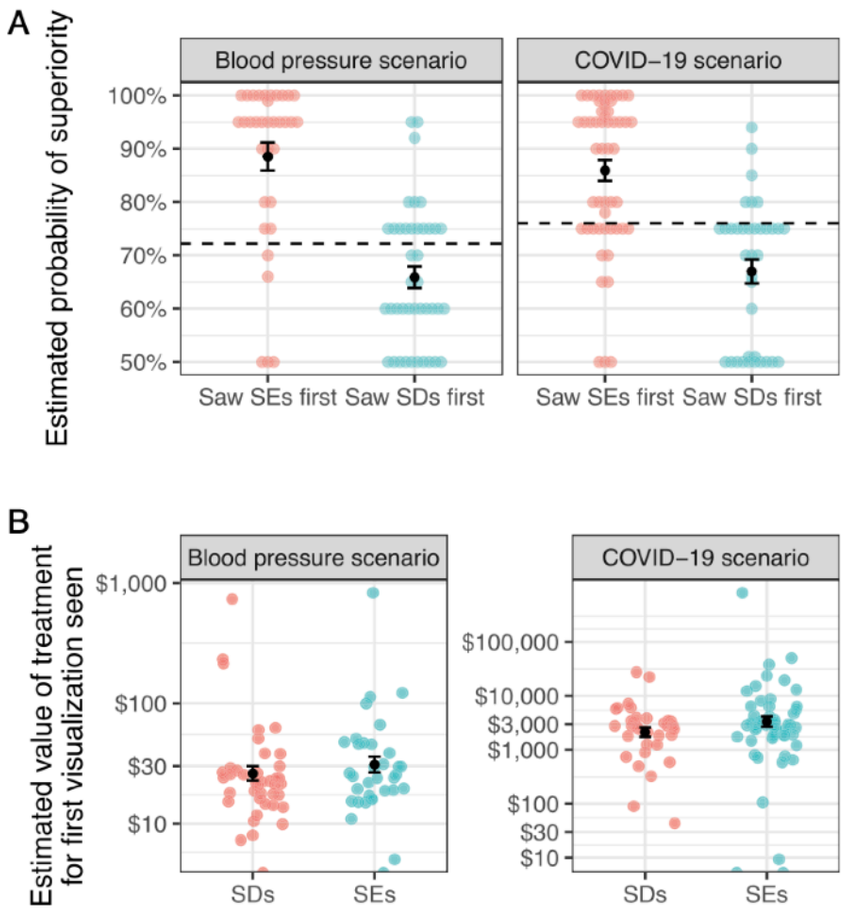


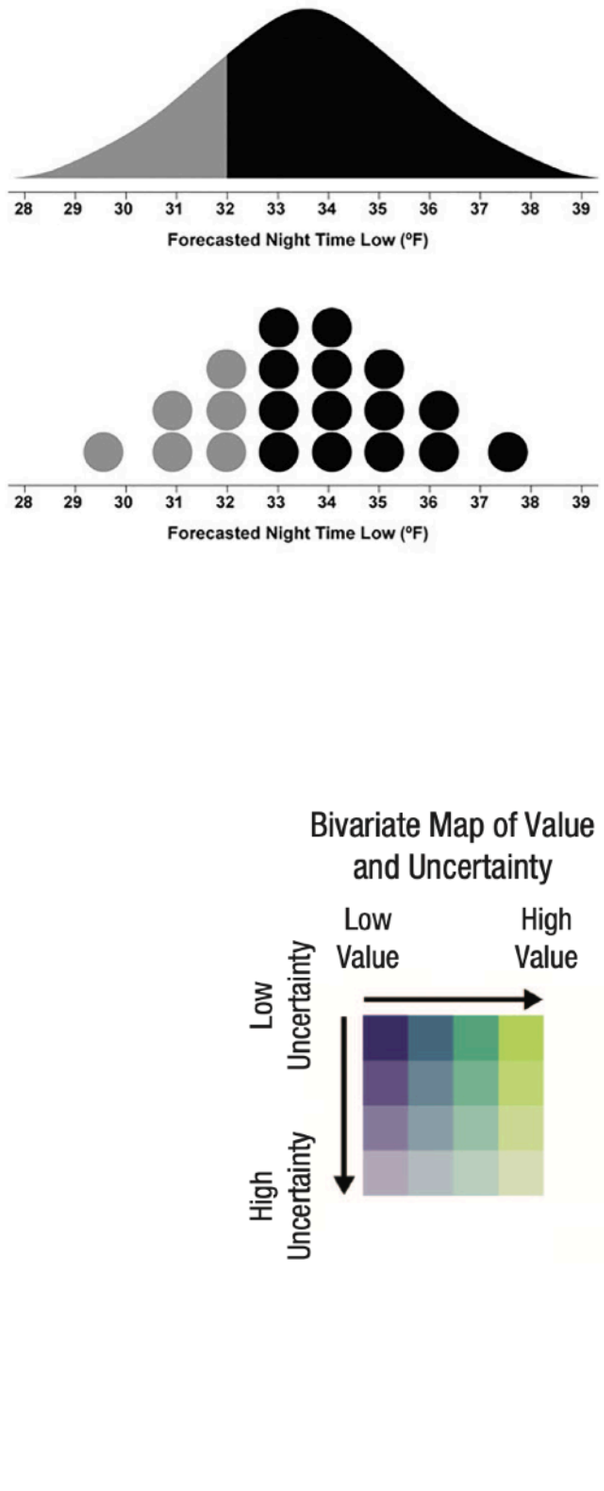
The choice of graph can substantially influence conclusions made from the same data

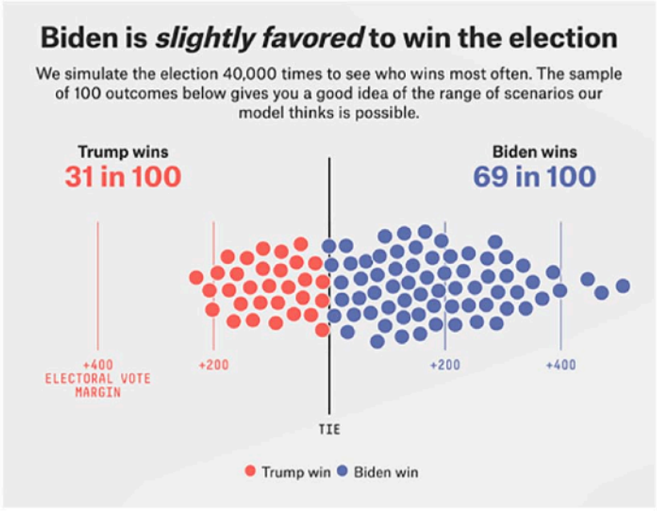
Make sure to visualise entire distributions!

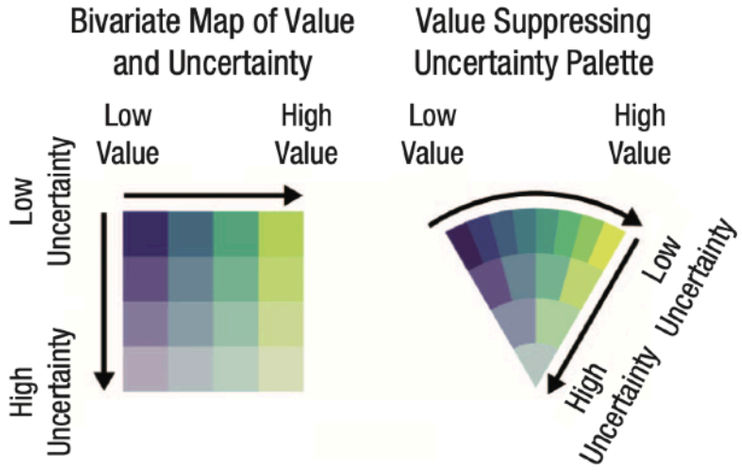


Many ways to do this









How, then?!?!

ggplot2: Visual Data Exploration

Solution: Graphics

Let's start by loading up our data. Click on the w4day1 project icon in the w4day1 zipped file you downloaded from Canvas. Then open `w4daylanalysis.Rmd`:

It loads up the libraries we need (including ggplot2) as well as the dataset (same as in last week's tutes)

````
```{r setup, include=FALSE}
# We'll begin by loading up the libraries and data we need, as always.
knitr::opts_chunk$set(echo = TRUE)

# loading the libraries
library(tidyverse)
library(here)
library(ggplot2)

loc <- here("shadowsurvey.csv")
d <- read_csv(file=loc)
d$year <- as.factor(d$year)
```
````

Solution: Graphics

We'll also organise the data. First let's create a goodFood and badFood variable, like we did in last weeks' tutes:

```
dnew <- d %>%
  mutate(goodFood = (carrot+bean+cake+meat)/4) %>%
      mutate(badFood = (mud+dirt)/2)
```

```
> dnew
# A tibble: 135 × 12
   name        gender species year  carrot  bean  cake  meat   mud  dirt goodFood badFood
   <chr>       <chr>  <chr>   <fct>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>   <dbl>
 1 foxy        female fox     2021       7     7     8     7     1     1     7.25     1
 2 bunny       female bunny   2021      10     7    10     9     1     1     9        1
 3 doggie      male   dog     2021       8     7    10    10     1     1     8.75     1
 4 flopsy      nb     bunny   2021      10    10     9    10     1     1     9.75     1
 5 fluffy      female cat     2021       7     9     8     8     2     1     8        1.5
 6 furry       male   bunny   2021       9     8     8     8     1     1     8.25     1
 7 giganticky  male   bunny   2021       6     8    10    10     3     1     8.5      2
 8 grey        female bunny   2021       9     8    10     9     2     1     9        1.5
 9 panda       male   bear    2021       6     6     9    10     1     1     7.75     1
10 pink bear   male   bear    2021       8     8     6     7     1     1     7.25     1
# ℹ 125 more rows
```

Solution: Graphics

For the plots we need to make, we need this to be in long form — we want there to be one column called “question” with different question subtypes (carrot, bean, mud, goodFood, etc).

```
dl <- dnew %>%
  pivot_longer(-c(name,gender,species,year),names_to=“question",
               values_to="rating")
```

```
> dl
# A tibble: 1,080 × 6
   name  gender species year  question rating
   <chr> <chr>  <chr>   <fct> <chr>     <dbl>
 1 foxy  female fox     2021  carrot     7
 2 foxy  female fox     2021  bean       7
 3 foxy  female fox     2021  cake       8
 4 foxy  female fox     2021  meat       7
 5 foxy  female fox     2021  mud        1
 6 foxy  female fox     2021  dirt       1
 7 foxy  female fox     2021  goodFood   7.25
 8 foxy  female fox     2021  badFood    1
 9 bunny female bunny   2021  carrot    10
10 bunny female bunny   2021  bean       7
# ℹ 1,070 more rows
```

Solution: Graphics

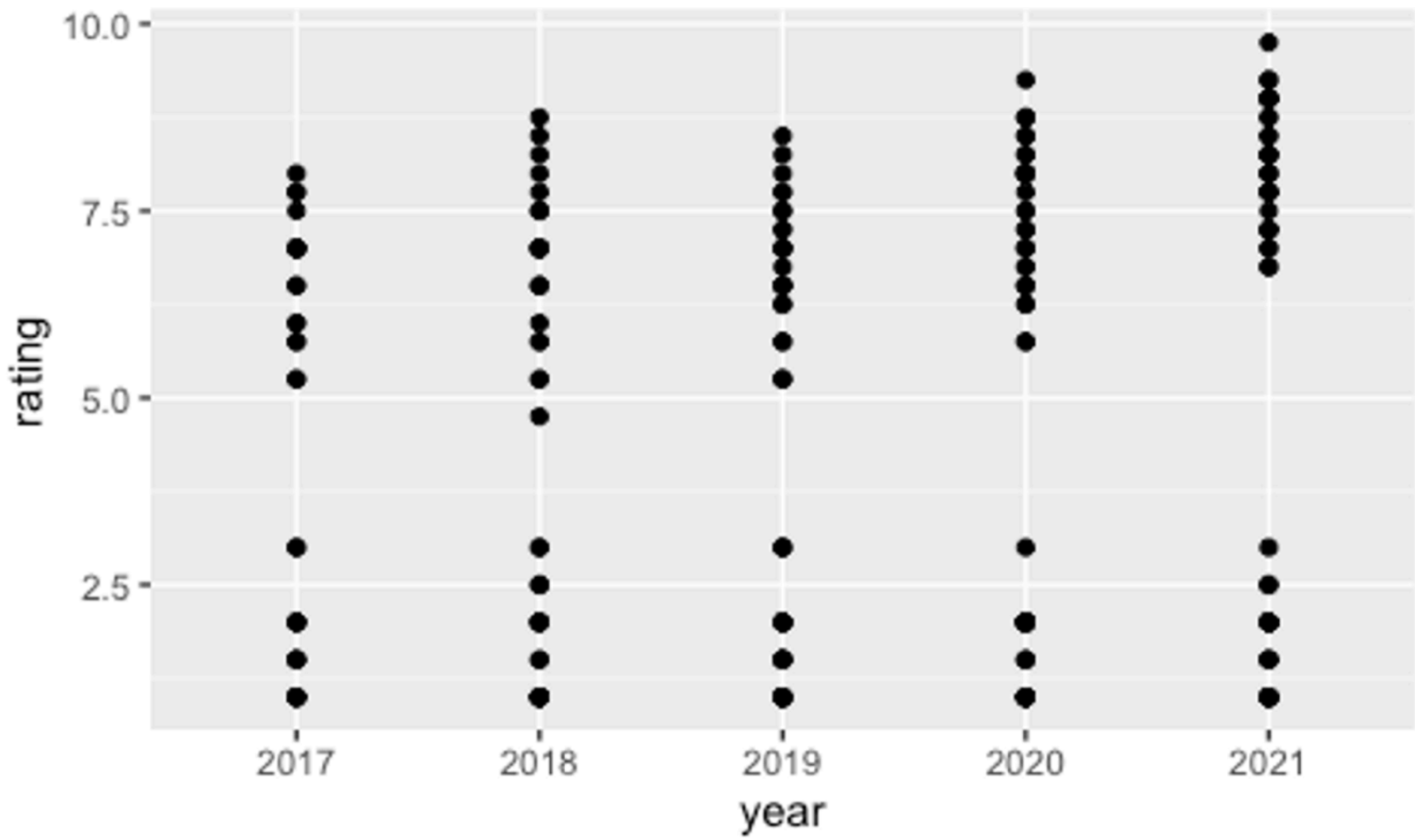
We probably don't need all of the individual sub questions, so let's filter out all of them except goodFood and badFood:

```
dl <- dnew %>%
  pivot_longer(-c(name,gender,species,year),names_to="question",
               values_to="rating") %>%
  filter(question=="goodFood" | question=="badFood")
```

```
> dl
# A tibble: 270 × 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
 7 flopsy nb     bunny   2021  goodFood   9.75
 8 flopsy nb     bunny   2021  badFood    1
 9 fluffy female cat     2021  goodFood   8
10 fluffy female cat     2021  badFood    1.5
# ℹ 260 more rows
```

Now let's make a plot!

```
dl %>%
  ggplot(mapping = aes(x = year, y = rating)) +
  geom_point()
```



Now let's make a plot!

No offence, but that is a lot of work for a kind of ugly plot. What's that all about?

ggplot2 is…

A grammar

- Compose and reuse smaller parts
- Create complex structure from simpler units

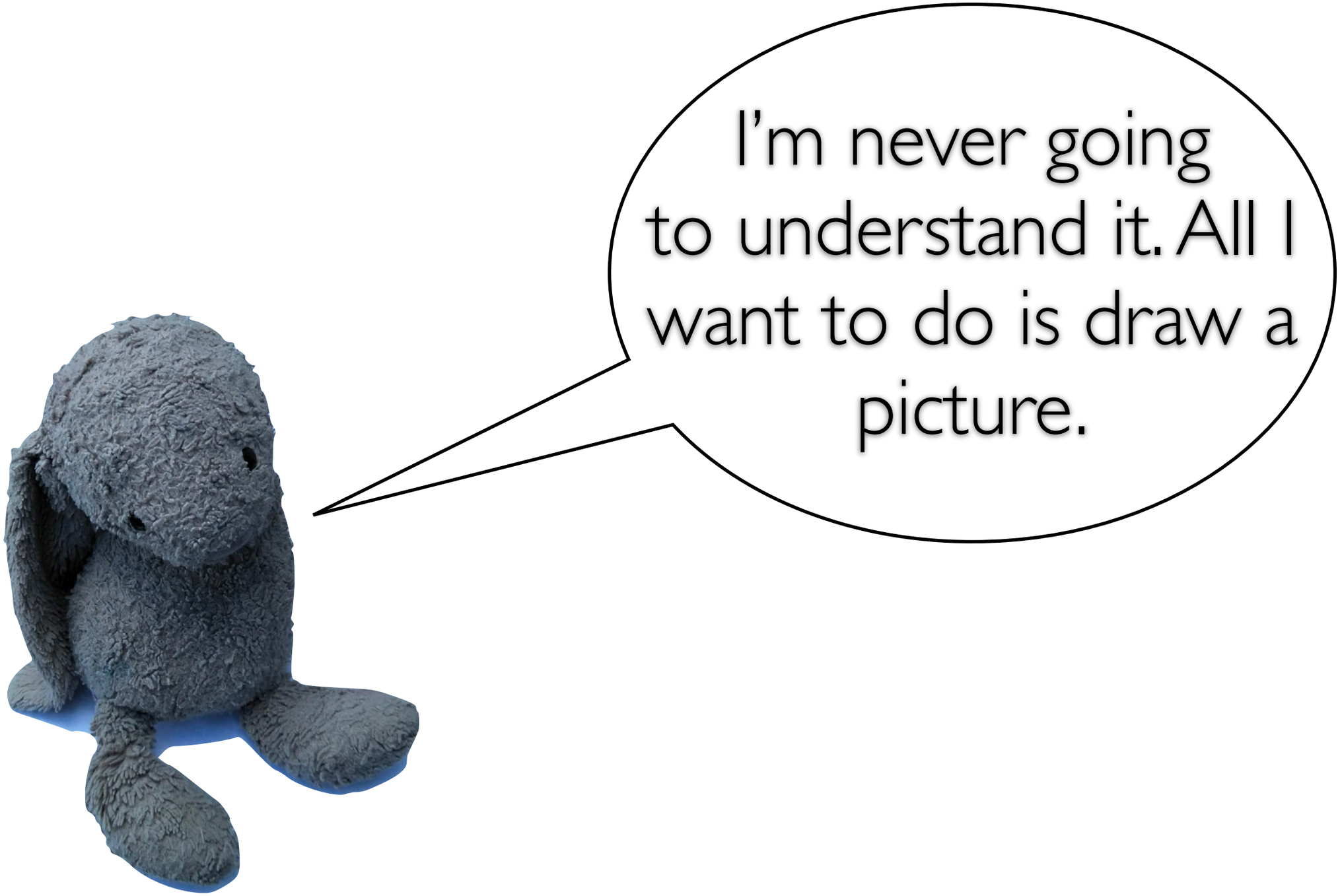
Of graphics

- Uses the “painter” model
- A plot is built in layers
- Each layer is drawn on top of the last

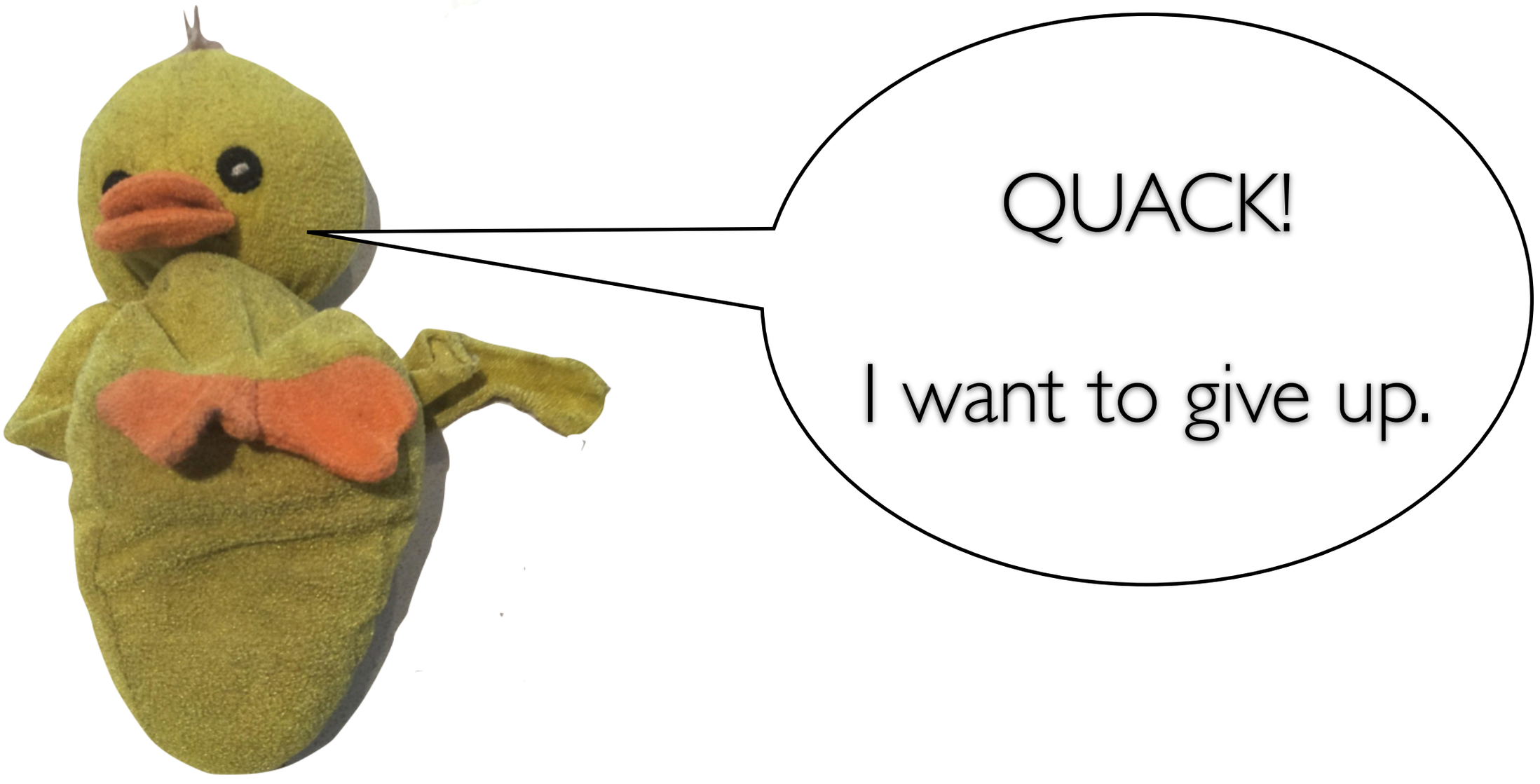
ggplot2 is…

That's all unnecessarily complicated, isn't it?

ggplot2 is…


I'm never going to understand it. All I want to do is draw a picture.

ggplot2 is…



Just take it one step at a time

```
dl %>%
```


Specifies the data (but it doesn’t know what to do with it yet)

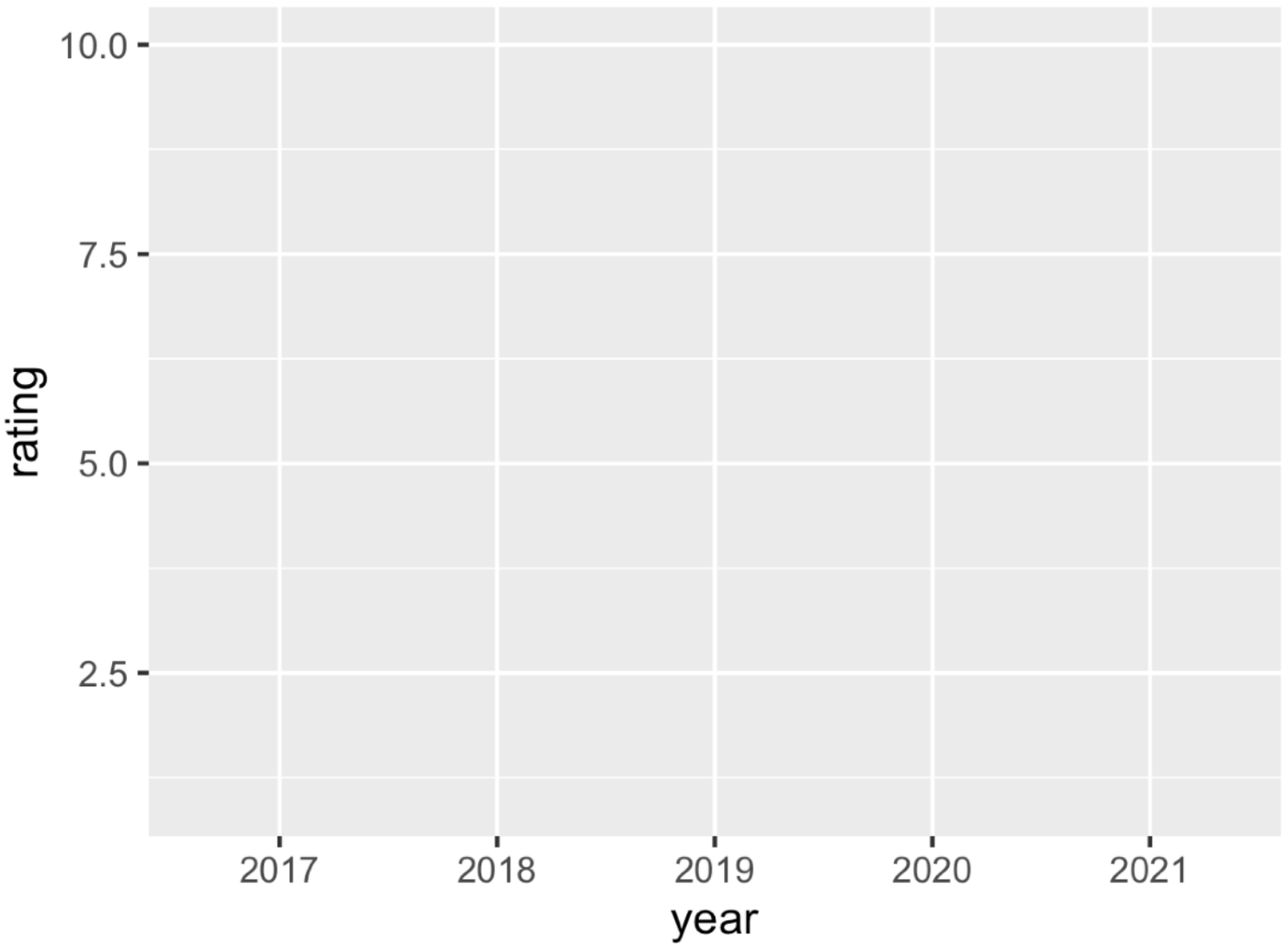
Just take it one step at a time

```
dl %>%
  ggplot()
```

Sets a blank canvas

Just take it one step at a time

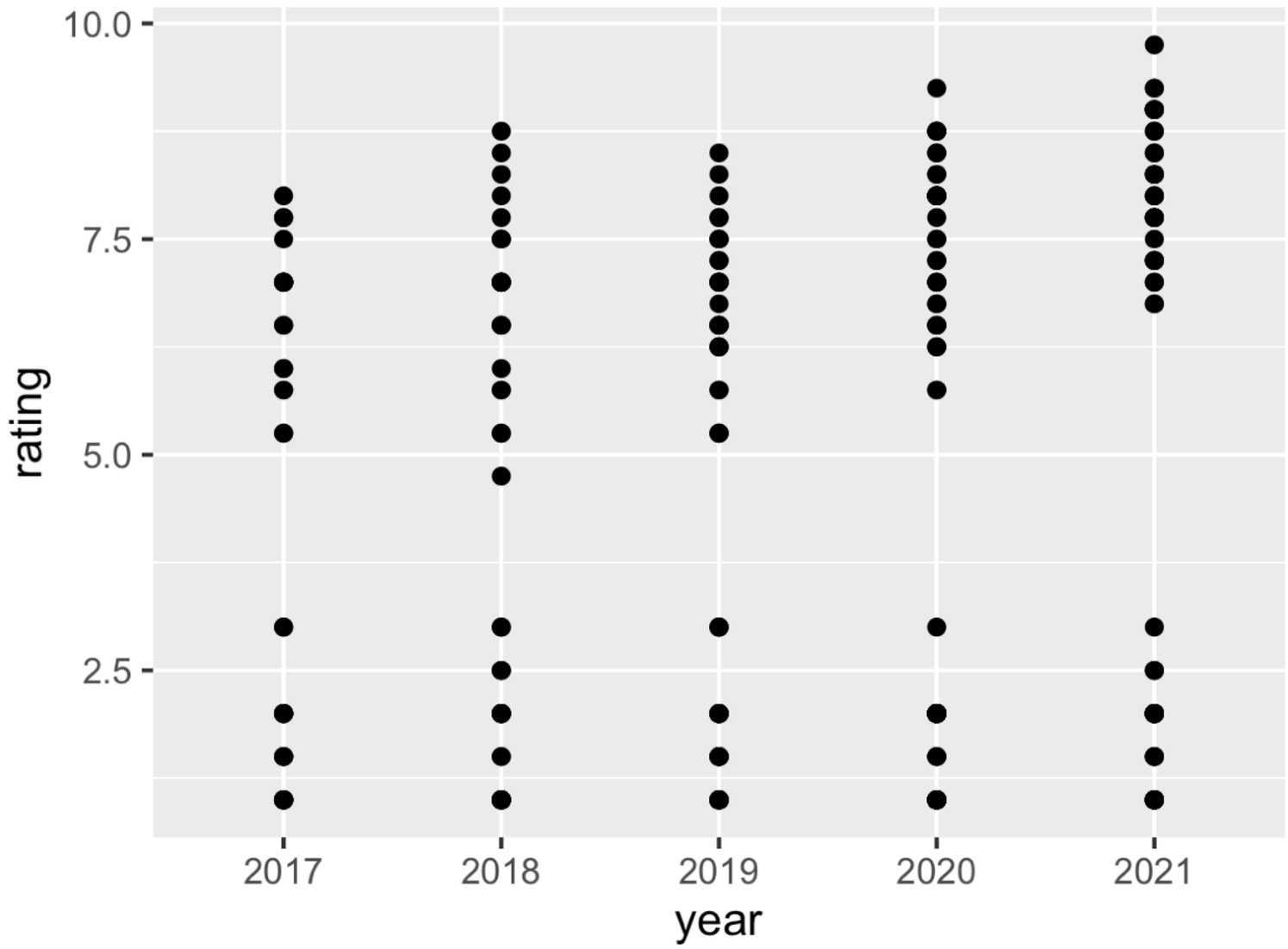
```
dl %>%
  ggplot(mapping = aes(x = year, y = rating))
```



Specifies a **mapping** from the data to the plot **aes**thetics (in this case, the x and y axis locations)

Just take it one step at a time

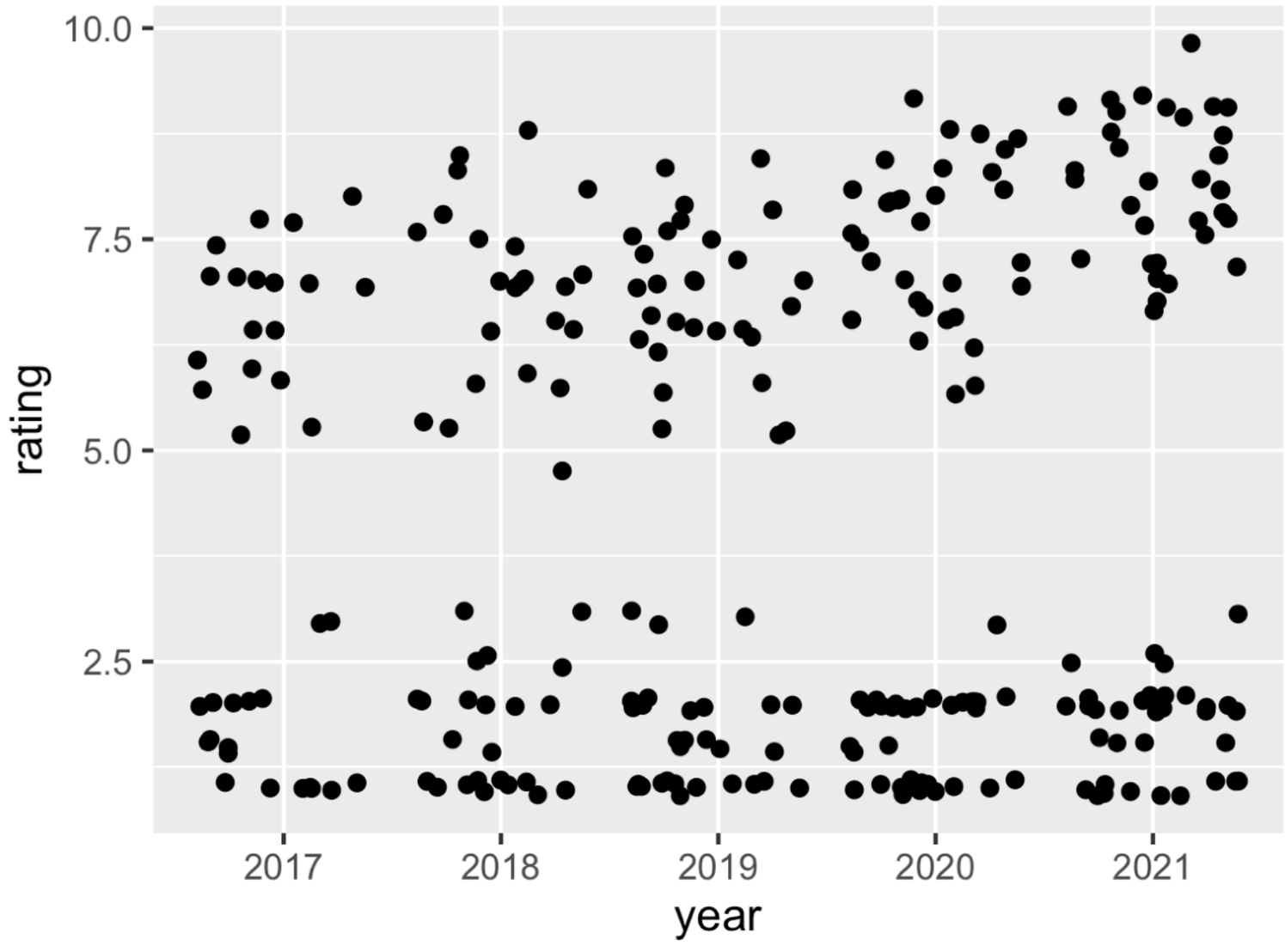
```
dl %>%
  ggplot(mapping = aes(x = year, y = rating)) +
  geom_point()
```



Add a **plot** layer (the points, lines, histograms, etc)

Now we can change all sorts of things!

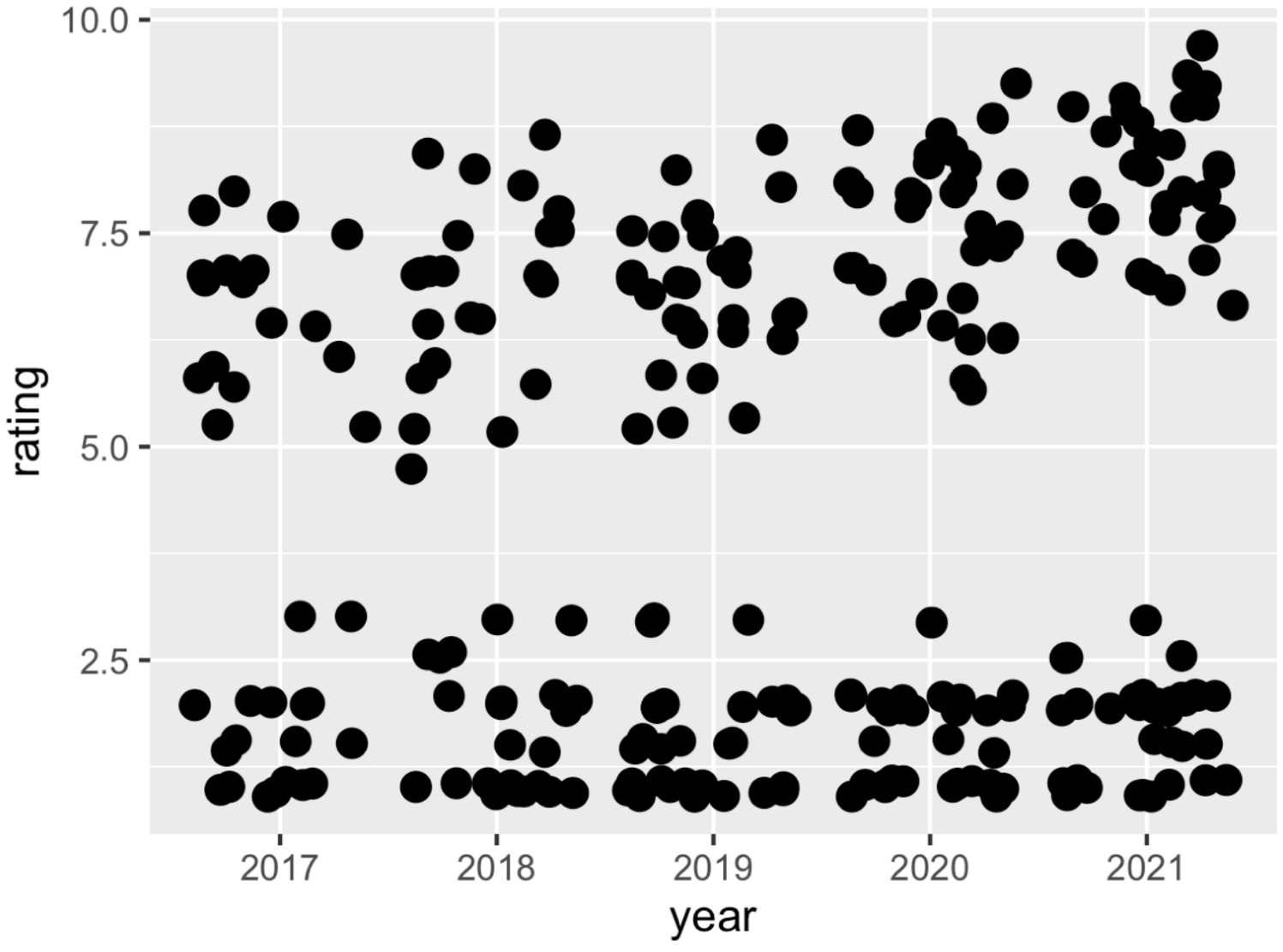
```
dl %>%
  ggplot(mapping = aes(x = year, y = rating)) +
  geom_jitter()
```



Uses a different plot layer to **jitter** the points

Now we can change all sorts of things!

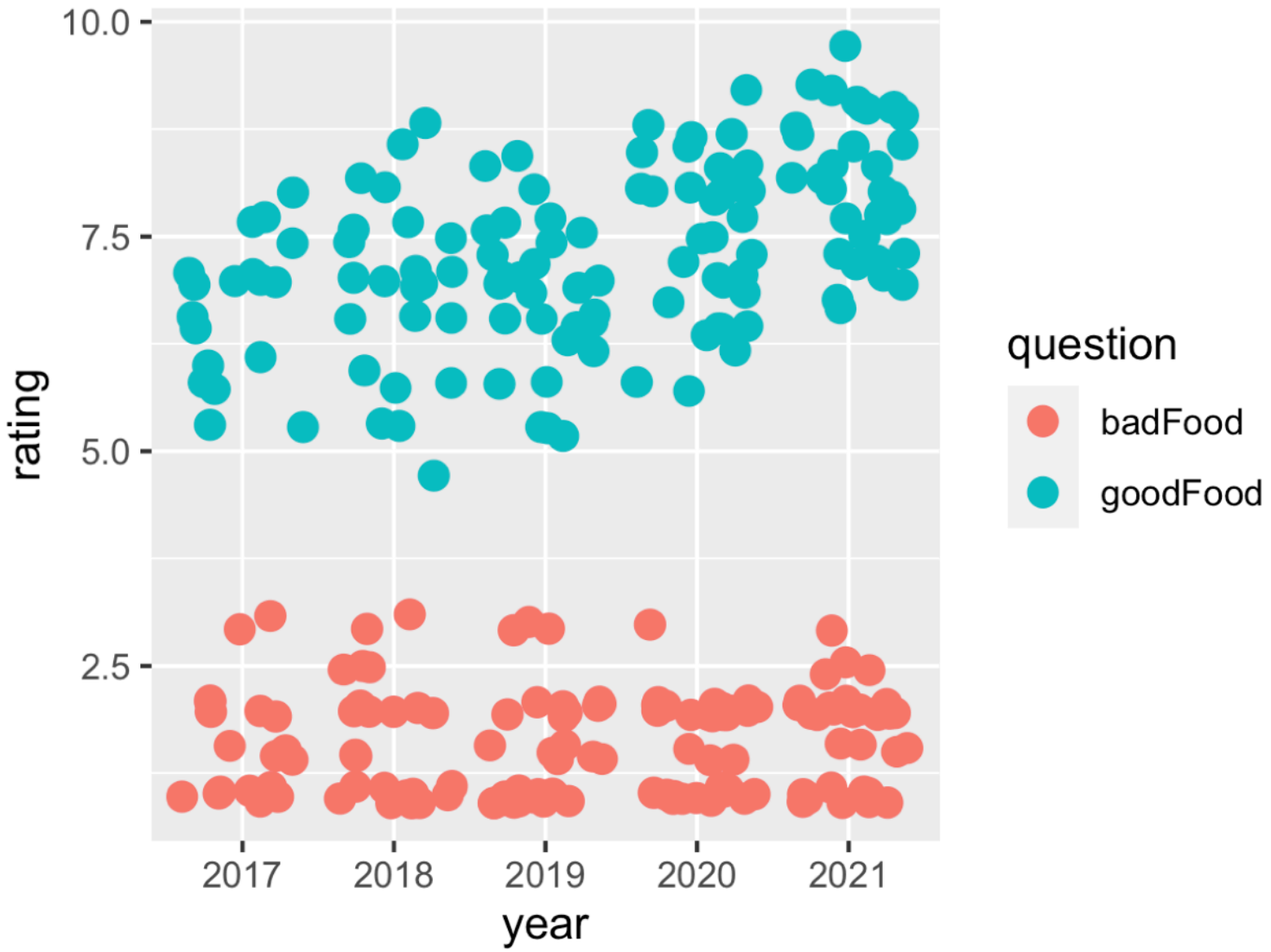
```
dl %>%
  ggplot(mapping = aes(x = year, y = rating)) +
  geom_jitter(size=3)
```



Add layer-specific parameters like dot size

Now we can change all sorts of things!

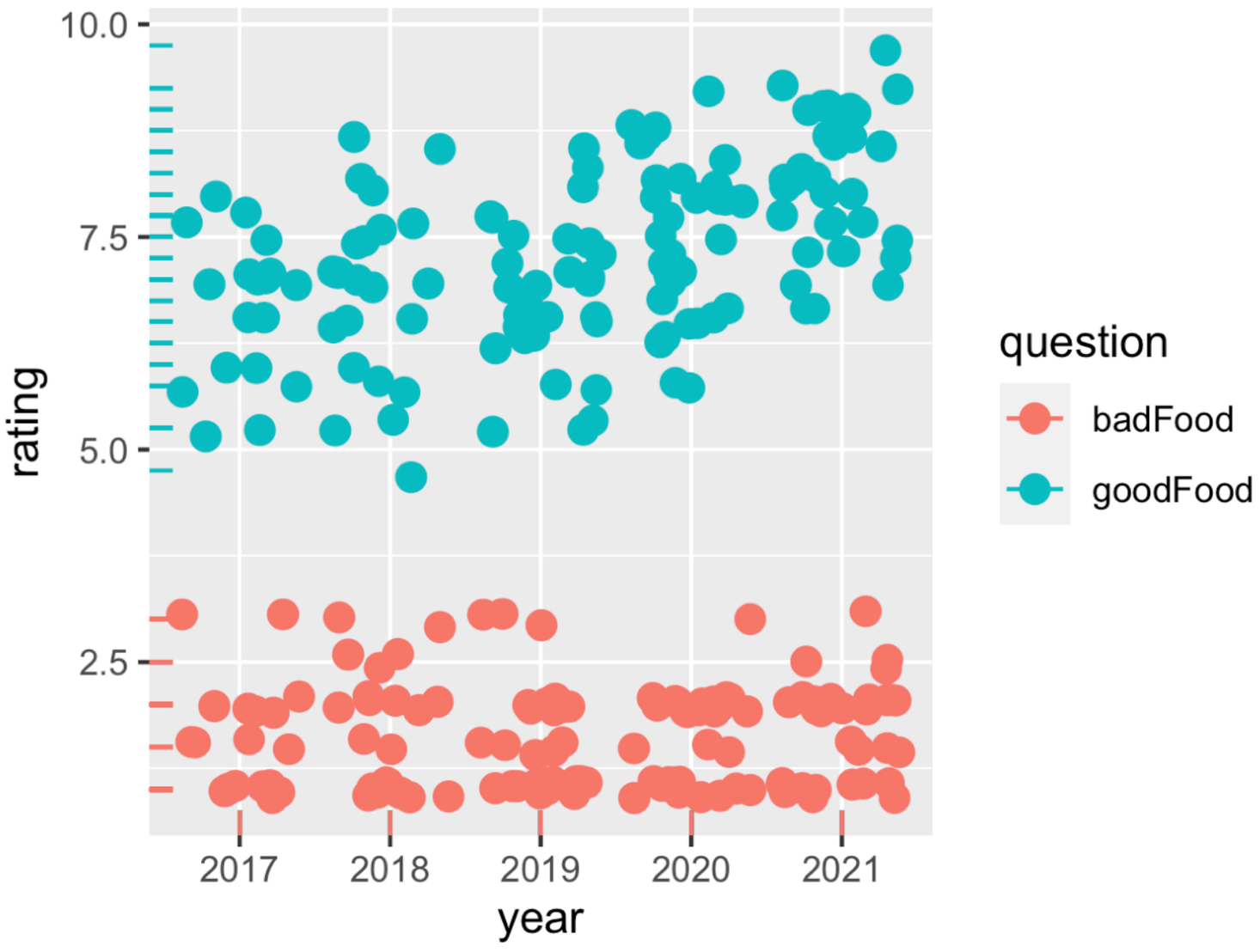
```
dl %>%
  ggplot(mapping = aes(x = year, y = rating, colour=question)) +
  geom_jitter(size=3)
```



Add additional **aes**thetics to the mappings

Now we can change all sorts of things!

```
dl %>%
  ggplot(mapping = aes(x = year, y = rating, colour=question)) +
  geom_jitter(size=3) +
  geom_rug()
```



Plot layers can go on top of each other!

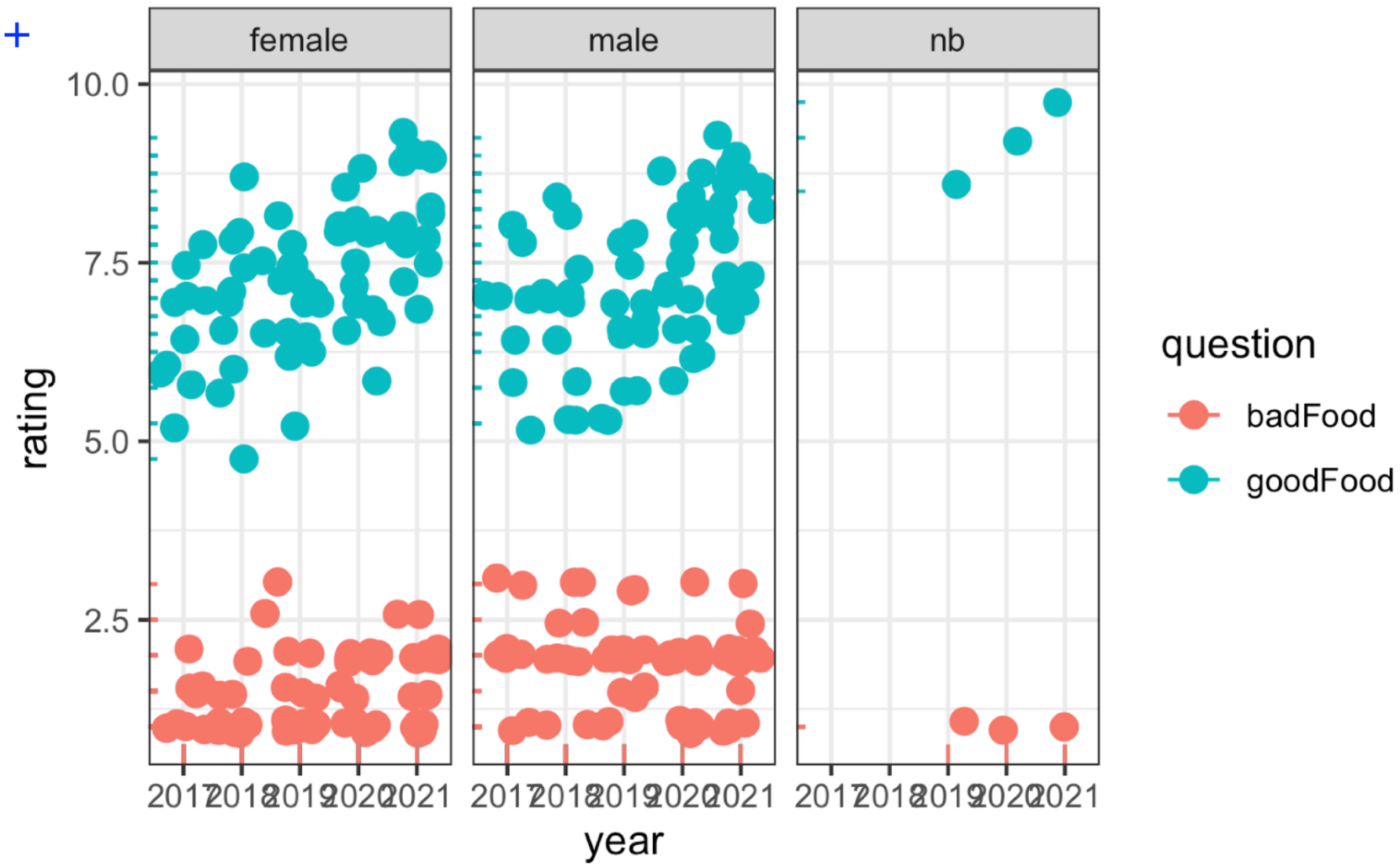
Now we can change all sorts of things!

```
dl %>%
  ggplot(mapping = aes(x = year, y = rating, colour=question)) +
  geom_jitter(size=3) +
  geom_rug() +
  facet_wrap(~gender)
```

We can split it into facets/panels

Now we can change all sorts of things!

```
dl %>%
  ggplot(mapping = aes(x = year, y = rating, colour=question)) +
  geom_jitter(size=3) +
  geom_rug() +
  facet_wrap(~gender) +
  theme_bw()
```



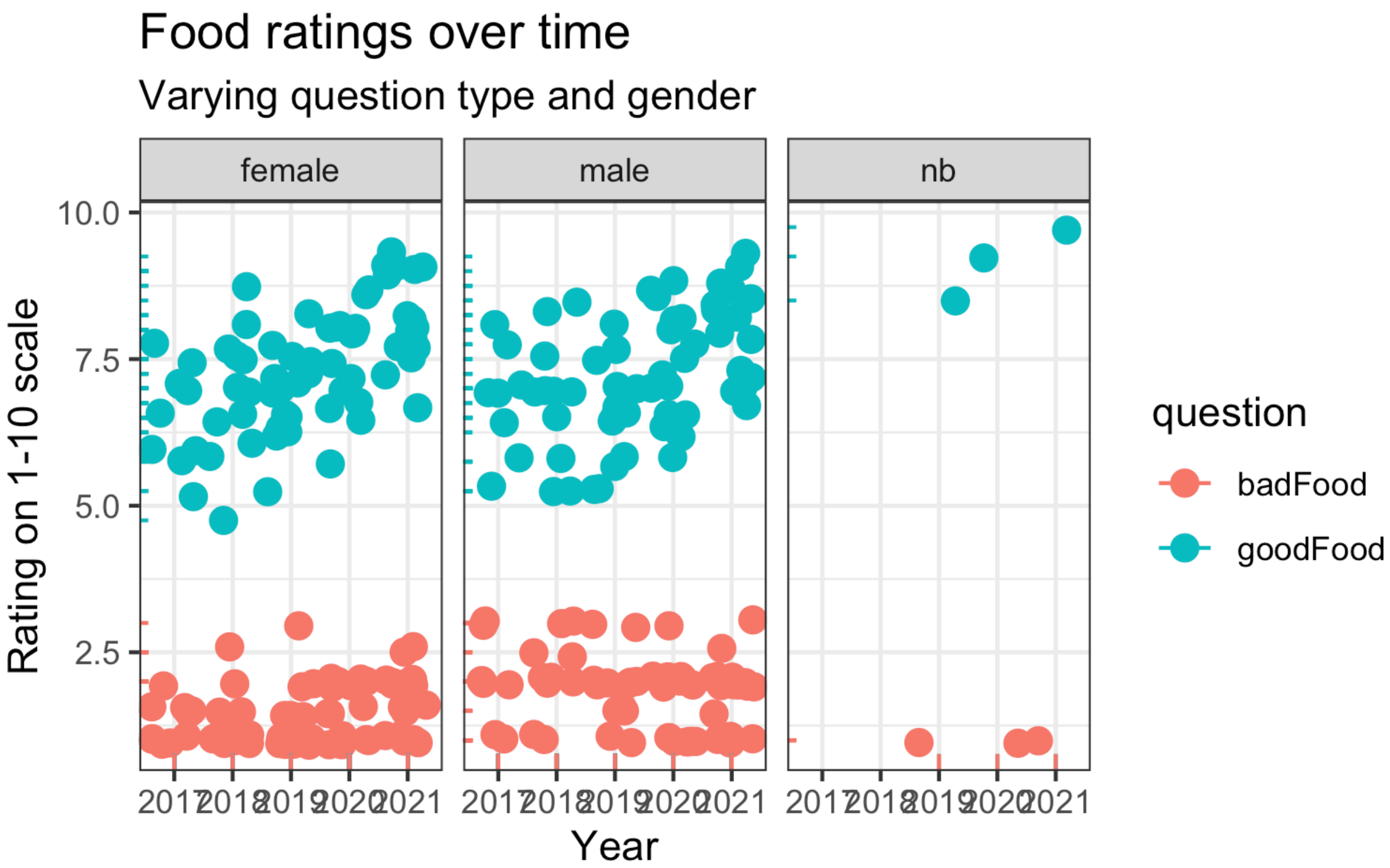
And modify the design theme…

Now we can change all sorts of things!

```
dl %>%
  ggplot(mapping = aes(x = year, y = rating, colour=question)) +
  geom_jitter(size=3) +
  geom_rug() +
  facet_wrap(~gender) +
  theme_bw() +
  labs(title = "Food ratings over time",
    subtitle = "Varying question type and gender",
  x = "Year",
  y = "Rating on 1-10 scale")
```

And add title, subtitle, axis labels, etc

Now our plot is beautiful!



See the `w4day1exercises.Rmd` file for the exercises!