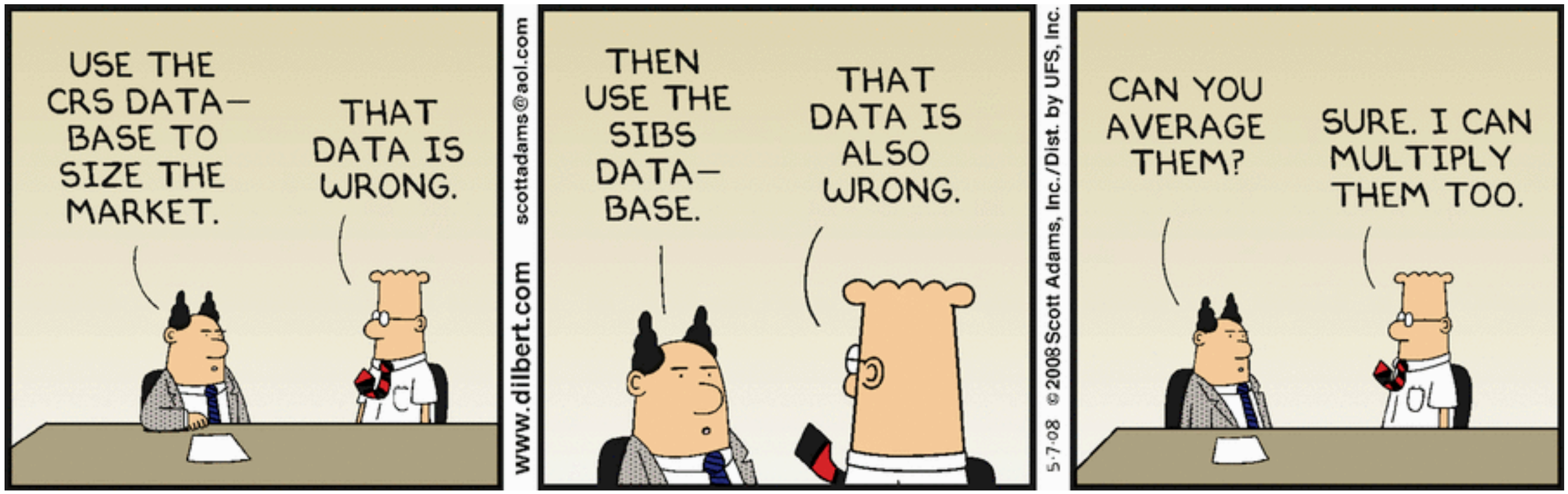
Simulation 2

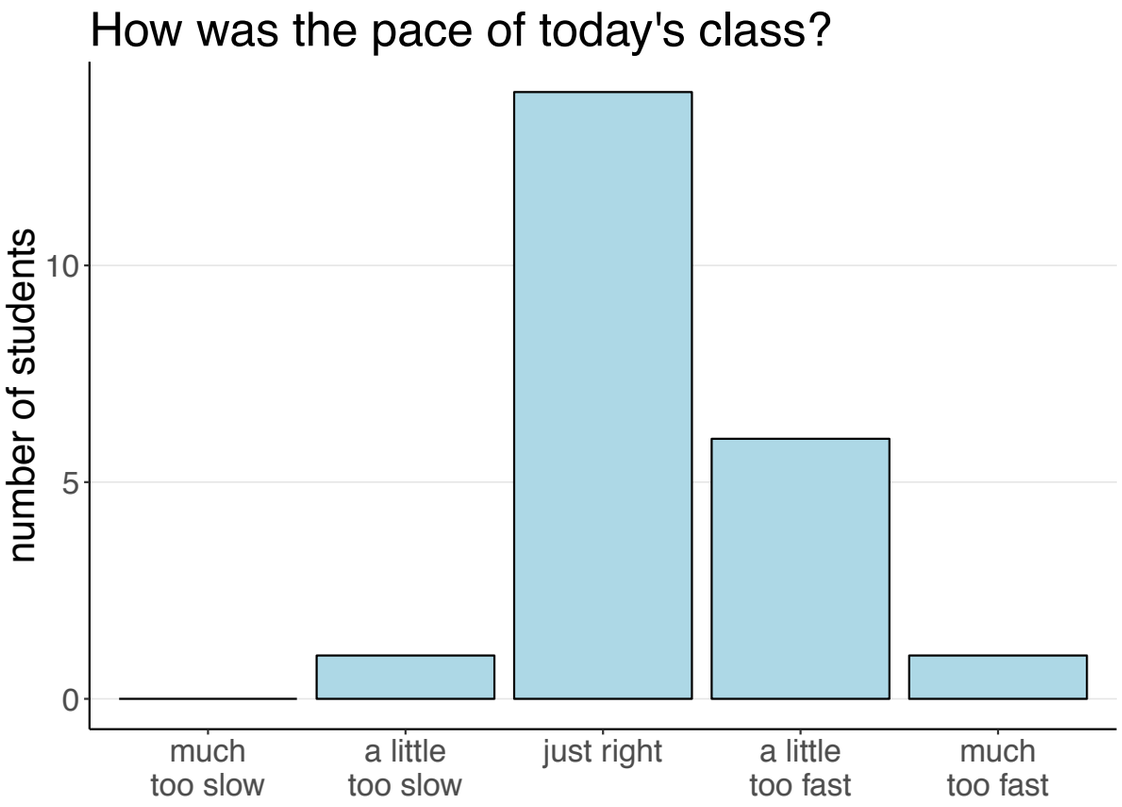


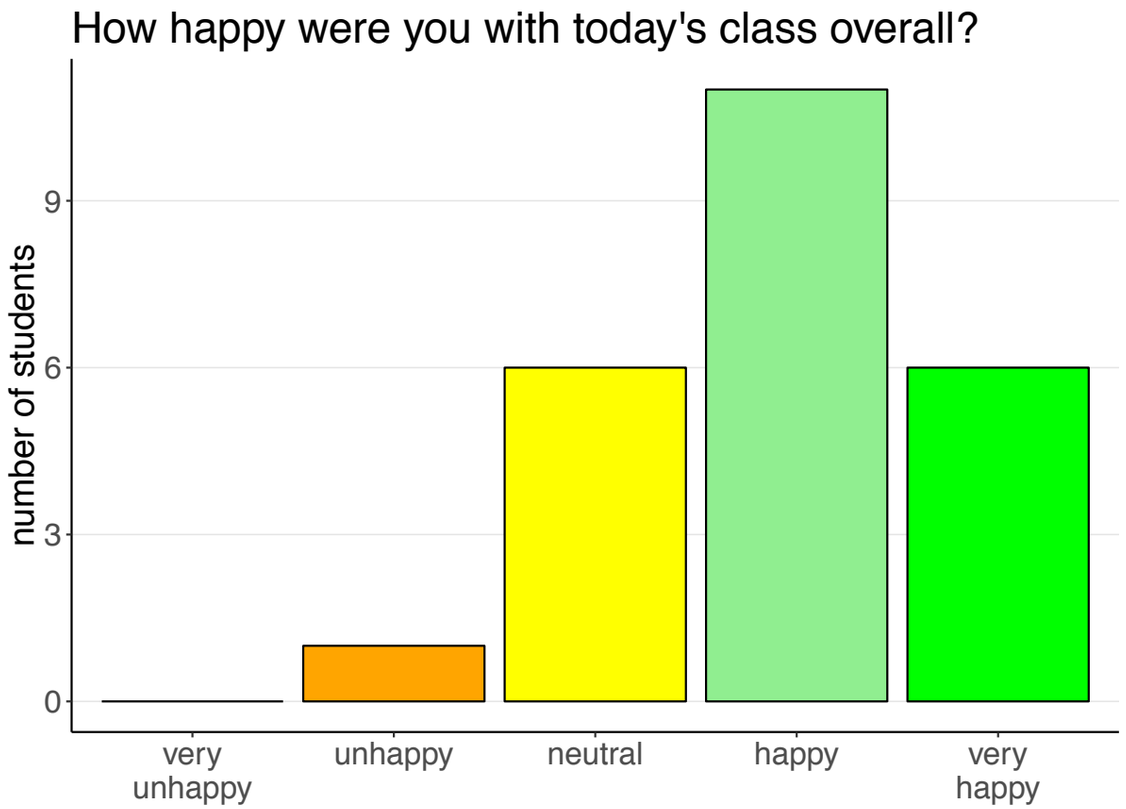
01/25/2019

Logistics

Your feedback

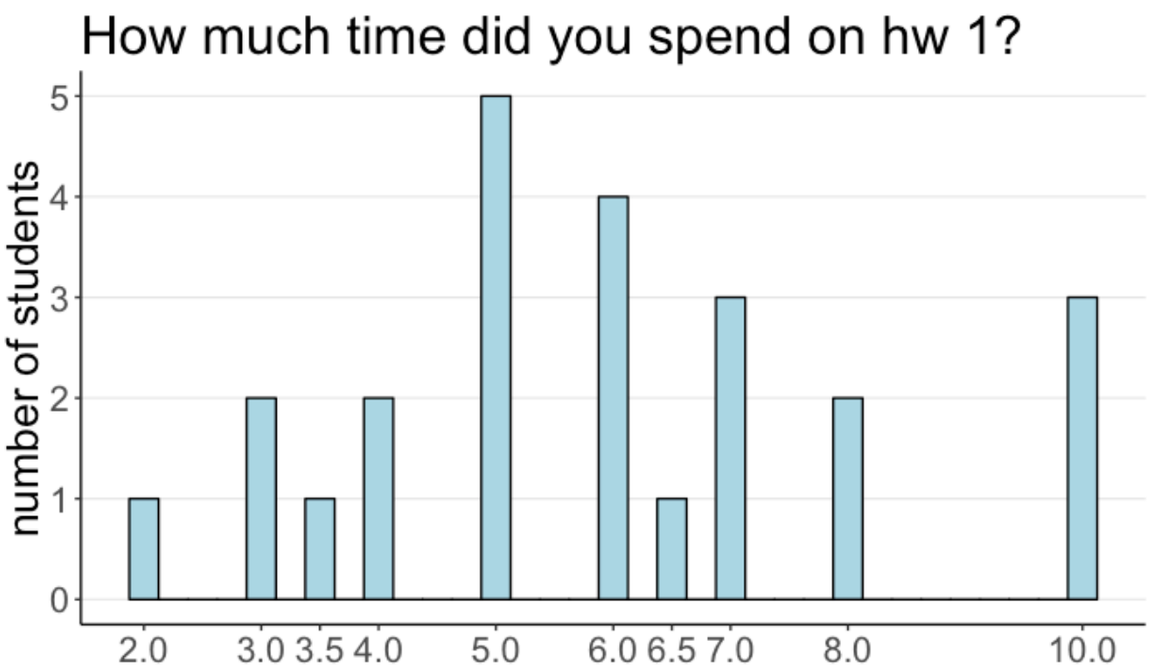
Your feedback



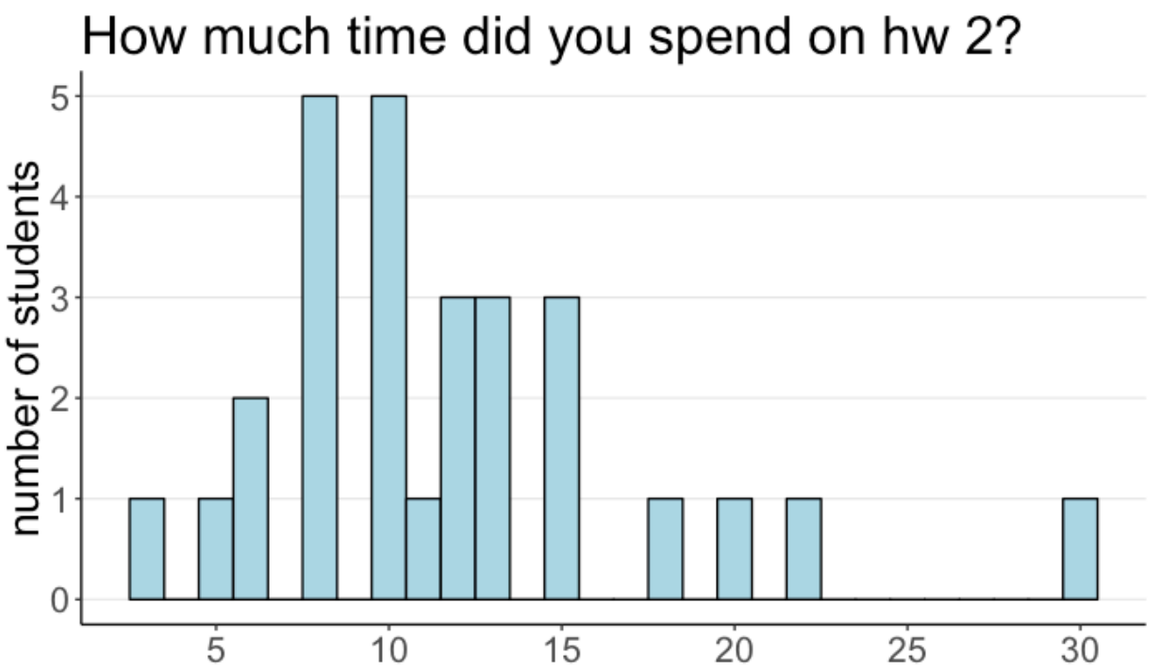


I'll try to go slower and use more examples

Your feedback



Average of **5.92** hours (SD = 2.21).



Average of **11.84** hours (SD = 5.62).

Your feedback

We don’t have a **formal opportunity to provide feedback on HW**, which is arguably very important!

Can you also take open ended feedback about homework or course in general? like maybe an anonymous feedback form on the website or something

ways to provide feedback on homework:

- using the open-ended feedback after class
- using the anonymous feedback form at the bottom of the website

Your feedback

The class is okay. Right now it is difficult to map what's learn to what we will use it later. So hopefully for later classes, it will be okay. **The major problem for me is that I was not able to make notes on the lecture notes.** I understand that it is difficult to prepare notes several days in advance, but I hope you can at least post the notes 10 min before the class, so students can download it and make notes when attending the class

I will post slides now before class

Your feedback

My comment is about the homework. **I am a bit concerned about how the questions were answered on piazza.** I got most of my answers from stackoverflow and I felt very upset when I couldn't receive any hint on how to fix the error in my code, when I spent countless of hours on it.

I'll give some Piazza hints

Piazza

Piazza

- best way to get help is by posting a **reprex**
- **reprex =** reproducible example

reprex

CRAN 0.2.1 | build passing | build passing | codecov 78% | lifecycle stable

Overview

Prepare reprexes for posting to GitHub issues, StackOverflow, or Slack snippets. What is a `reprex`? It's a **repr**oducible **ex**ample, as coined by Romain Francois.

Given R code on the clipboard, selected in RStudio, as an expression (quoted or not), or in a file ...

- run it via `rmarkdown::render()`,
- with deliberate choices re: arguments and setup chunk.

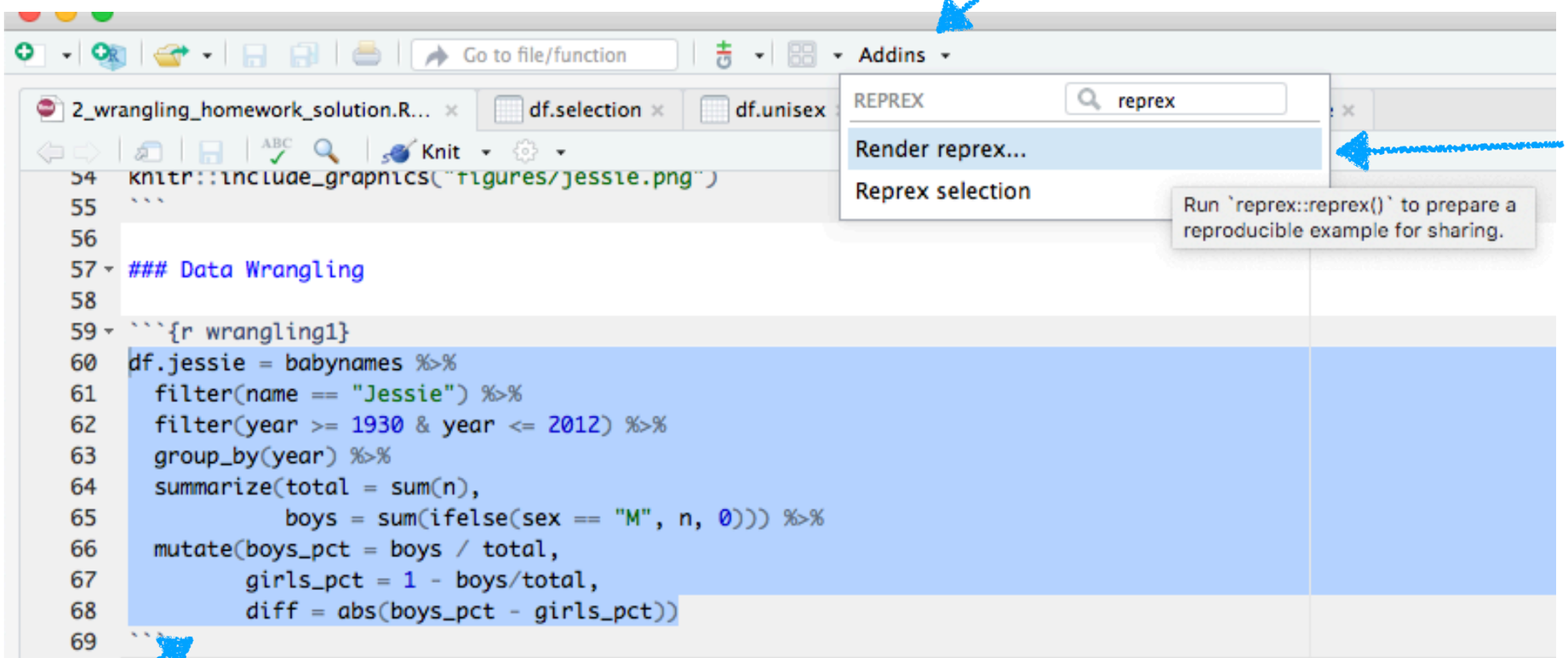
Get resulting runnable code + output as

- Markdown, formatted for target venue, e.g. `gh` or `so`, or as
- R code, augmented with commented output.

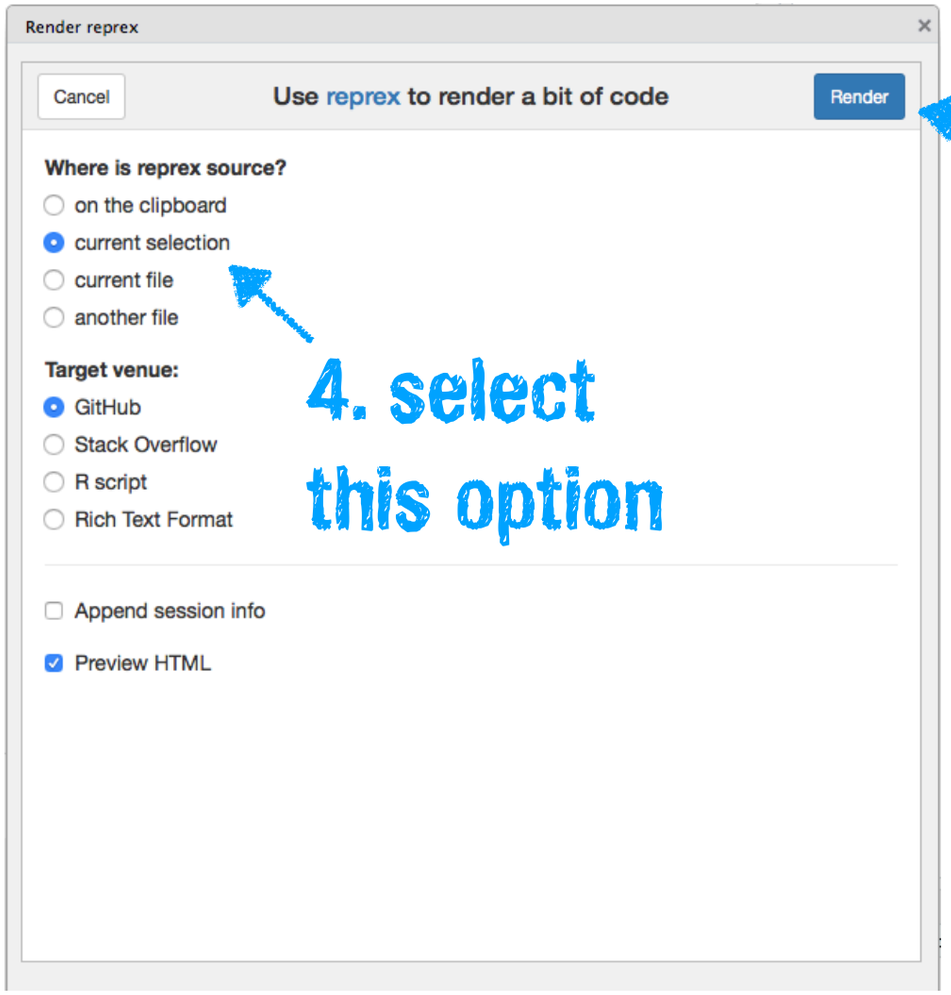
Result is returned invisibly, placed on the clipboard, and written to a file. Preview an HTML version in RStudio viewer or default browser.

Piazza

- `install.package("reprex")`



Piazza



6. copy and paste from the viewer

Piazza

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish

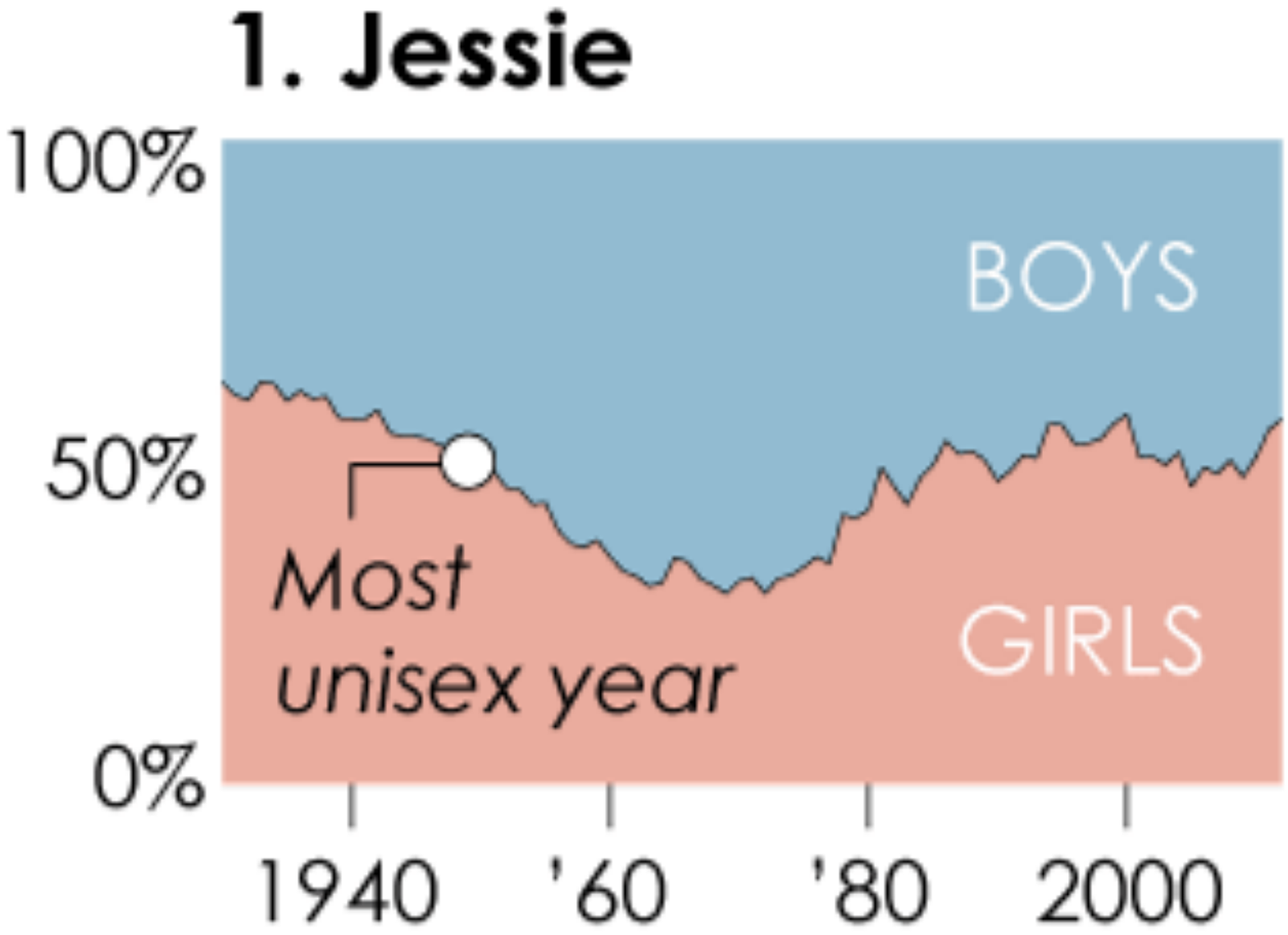
```
library("babynames")
library("tidyverse")
df.jessie = babynames %>%
  filter(name == "Jessie") %>%
  filter(year >= 1930 & year <= 2012) %>%
  group_by(year) %>%
  summarize(total = sum(n),
            boys = sum(ifelse(sex == "M", n, 0))) %>%
  mutate(boys_pct = boys / total,
         girls_pct = 1 - boys/total,
         diff = abs(boys_pct - girls_pct)) %>%
  print()
#> # A tibble: 83 x 6
#>     year total  boys boys_pct girls_pct  diff
#>    <dbl> <int> <dbl>    <dbl>     <dbl> <dbl>
#>  1  1930  3525  1329    0.377     0.623 0.246
#>  2  1931  3196  1267    0.396     0.604 0.207
#>  3  1932  3178  1282    0.403     0.597 0.193
#>  4  1933  2886  1079    0.374     0.626 0.252
#>  5  1934  2883  1090    0.378     0.622 0.244
#>  6  1935  2721  1103    0.405     0.595 0.189
#>  7  1936  2599  1012    0.389     0.611 0.221
#>  8  1937  2589  1041    0.402     0.598 0.196
#>  9  1938  2446   970    0.397     0.603 0.207
#> 10  1939  2454  1058    0.431     0.569 0.138
#> # … with 73 more rows
```

7. make sure to load necessary packages in your reprex

Created on 2019-01-24 by the reprex package (v0.2.1)

Homework 2 solution

Homework 2: Exercise 1



Homework 2: Exercise 1

Wrangling

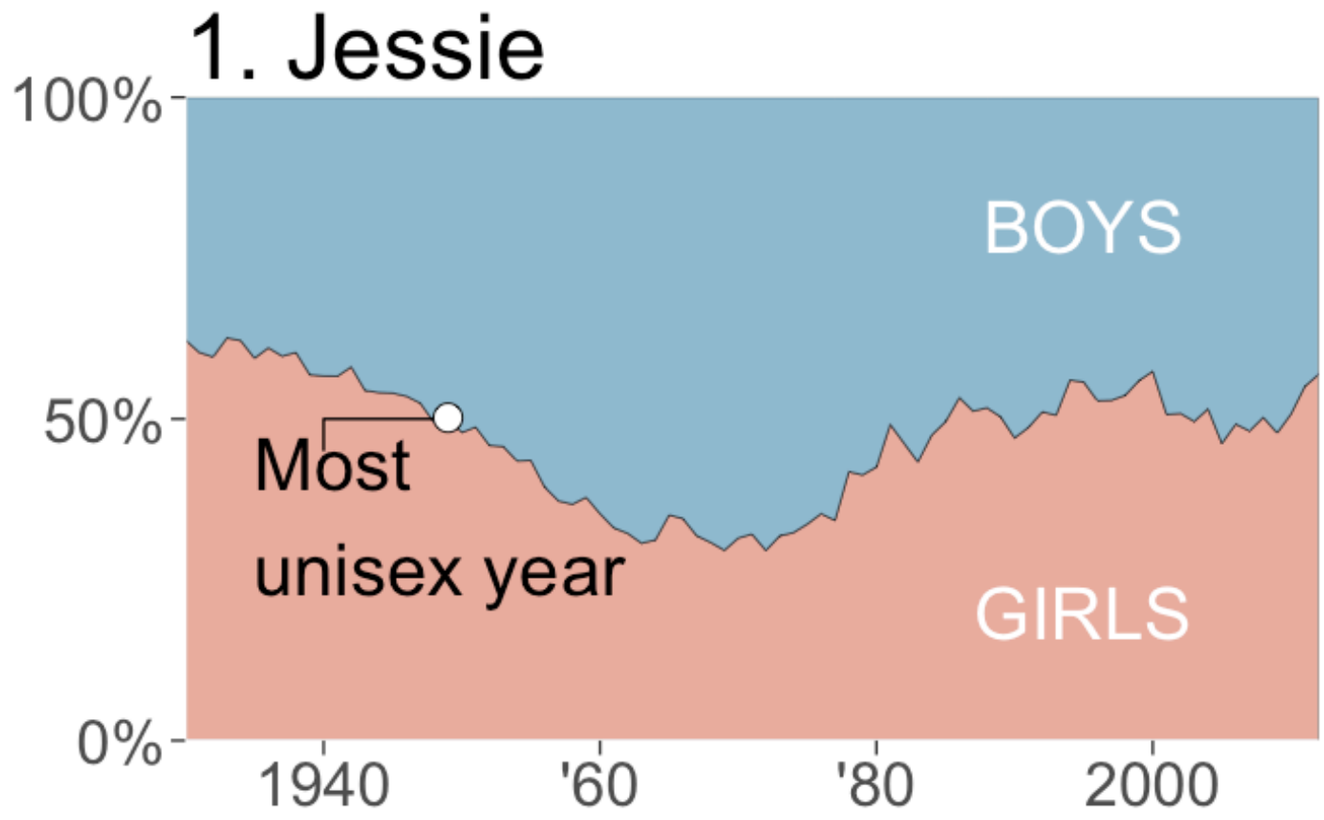
```
df.jessie = babynames %>%
  filter(name == "Jessie") %>%
  filter(year >= 1930 & year <= 2012) %>%
  group_by(year) %>%
  summarize(total = sum(n),
            boys = sum(ifelse(sex == "M", n, 0))) %>%
  mutate(boys_pct = boys / total,
         girls_pct = 1 - boys/total,
         diff = abs(boys_pct - girls_pct))
```

year	total	boys	boys_pct	girls_pct	diff
1930	3525	1329	0.38	0.62	0.25
1931	3196	1267	0.40	0.60	0.21
1932	3178	1282	0.40	0.60	0.19
1933	2886	1079	0.37	0.63	0.25
1934	2883	1090	0.38	0.62	0.24
1935	2721	1103	0.41	0.59	0.19
1936	2599	1012	0.39	0.61	0.22
1937	2589	1041	0.40	0.60	0.20
1938	2446	970	0.40	0.60	0.21
1939	2454	1058	0.43	0.57	0.14

Homework 2: Exercise 1

Plotting

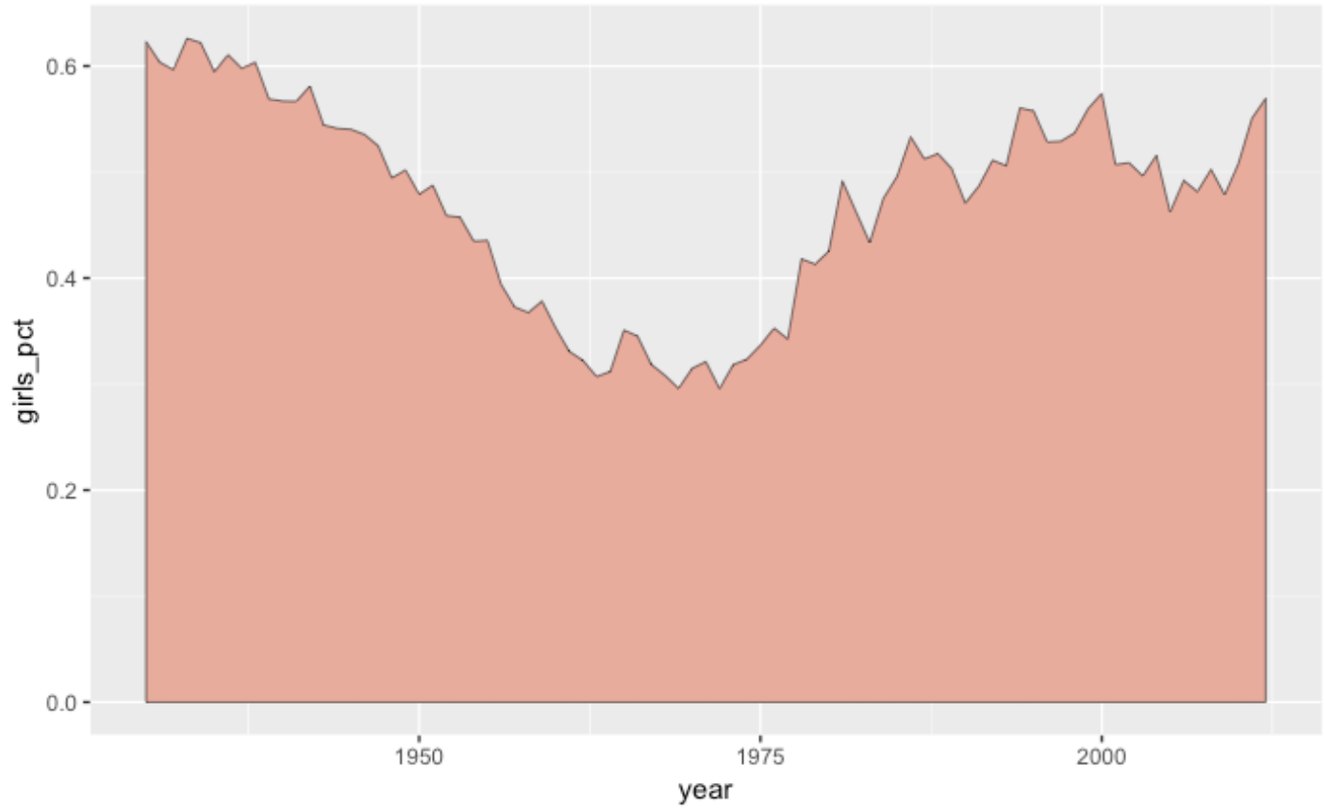
```
ggplot(data = df.jessie,
       mapping = aes(x = year, y = girls_pct)) +
  geom_area(fill = "#eaac9e",
            color = "black",
            size = 0.2) +
  geom_ribbon(mapping = aes(ymin = girls_pct, ymax = 1),
              fill = "#92bbd1",
              color = "black",
              size = 0.2) +
  geom_point(data = df.jessie %>%
               arrange(diff) %>%
               filter(row_number() == 1),
             shape = 21,
             fill = "white",
             size = 5) +
  geom_path(data = tibble(year = c(1940, 1940, 1949),
                          girls_pct = c(0.45, 0.5, 0.5))) +
  annotate(geom = "text",
           x = 1995,
           y = 0.8,
           label = "BOYS",
           color = "white",
           size = 10) +
  annotate(geom = "text",
           x = 1995,
           y = 0.2,
           label = "GIRLS",
           color = "white",
           size = 10) +
  annotate(geom = "text",
           x = 1935,
           y = 0.35,
           label = "Most\nunisex year",
           hjust = "left",
           size = 10) +
  scale_y_continuous(name = NULL,
                     limits = c(0, 1),
                     breaks = c(0, 0.5, 1),
                     labels = c("0%", "50%", "100%"),
                     expand = c(0, 0)) +
  scale_x_continuous(name = NULL,
                     breaks = seq(from = 1940, to = 2000, by = 20),
                     labels = c(1940, "'60", "'80", 2000),
                     expand = c(0, 0)) +
  labs(title = "1. Jessie") +
  theme(axis.ticks.length = unit(0.2, "cm"),
        axis.line = element_line(color = "white"),
        panel.background = element_rect(fill = "white"),
        text = element_text(size = 30,  family = "ArialMT"))
```



Homework 2: Exercise 1

Plotting

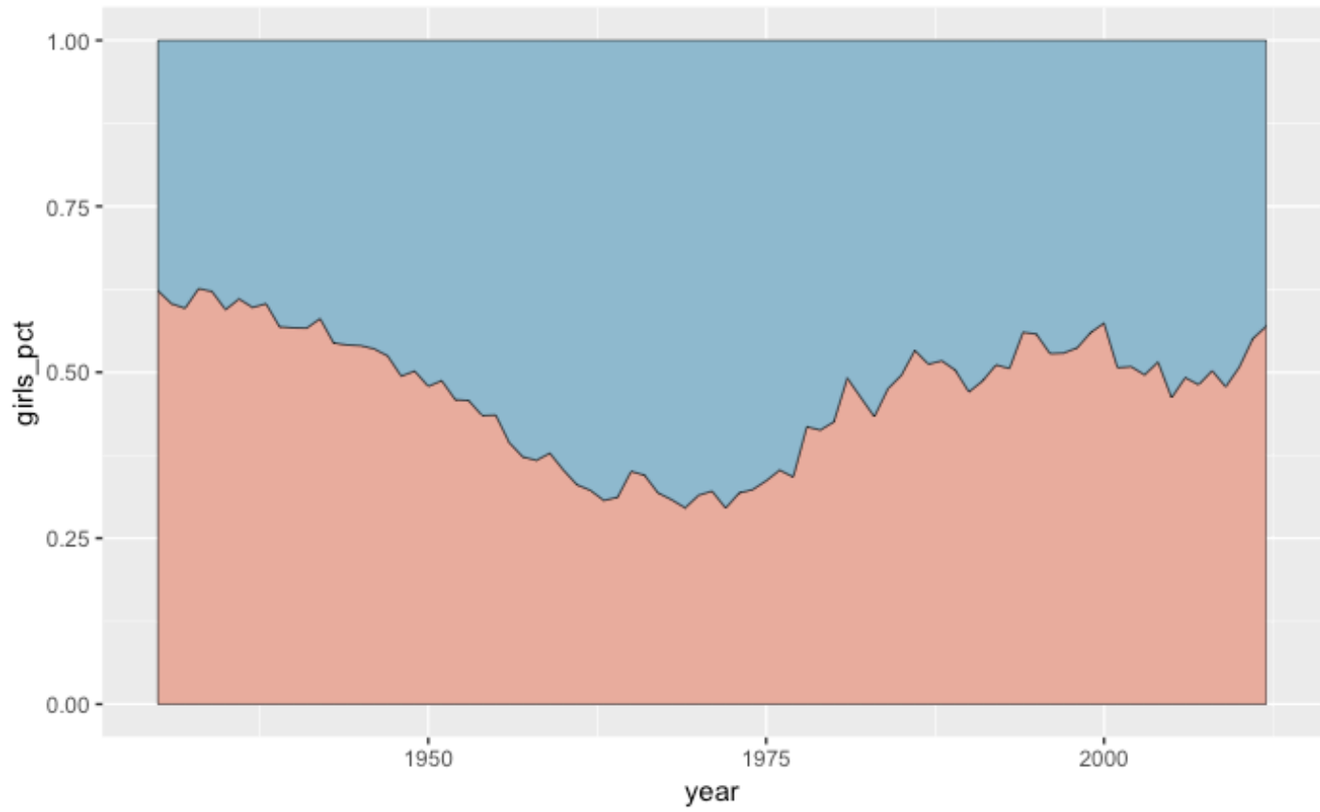
```
1 ggplot(data = df.jessie,
2        mapping = aes(x = year, y = girls_pct)) +
3   geom_area(fill = "#eaac9e",
4             color = "black",
5             size = 0.2)
```



Homework 2: Exercise 1

Plotting

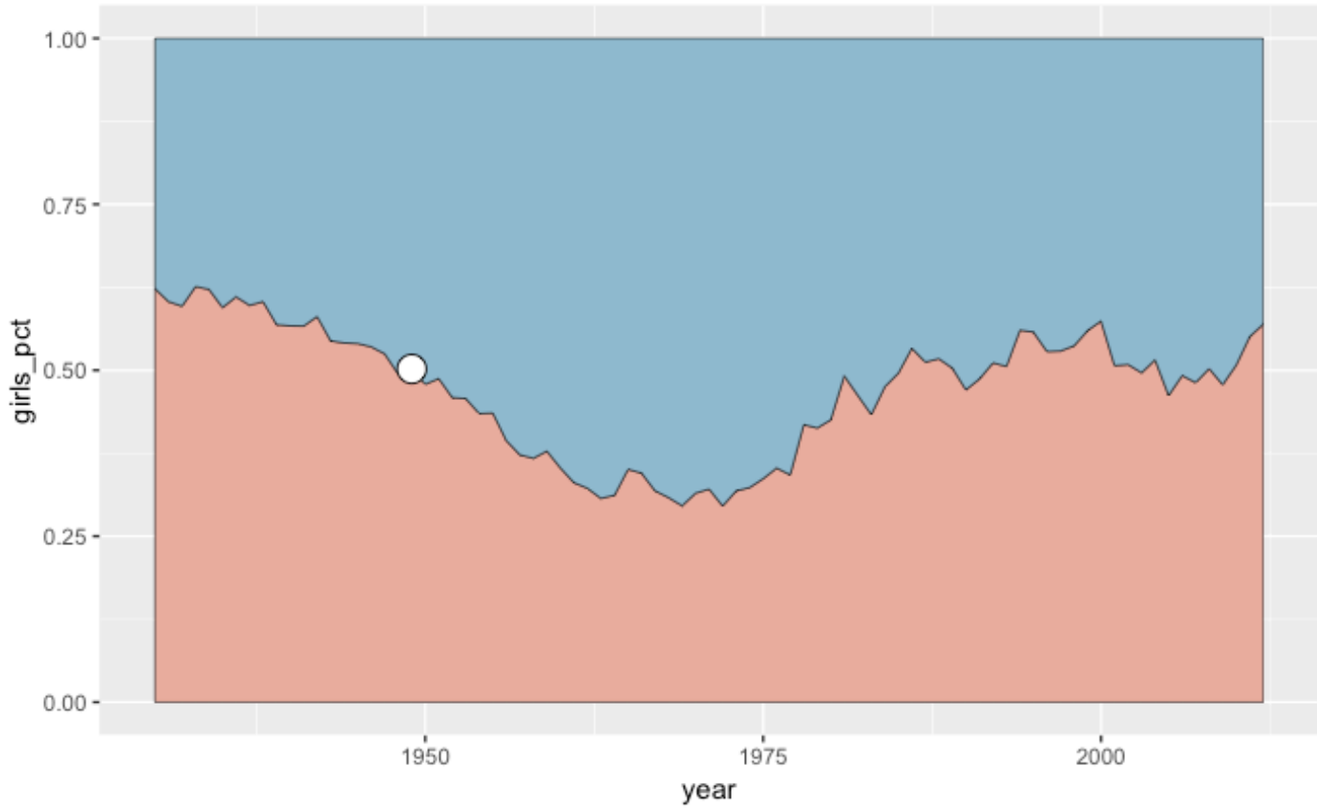
```
ggplot(data = df.jessie,
       mapping = aes(x = year, y = girls_pct)) +
  geom_area(fill = "#eaac9e",
            color = "black",
            size = 0.2) +
  geom_ribbon(mapping = aes(ymin = girls_pct, ymax = 1),
              fill = "#92bbd1",
              color = "black",
              size = 0.2)
```



Homework 2: Exercise 1

Plotting

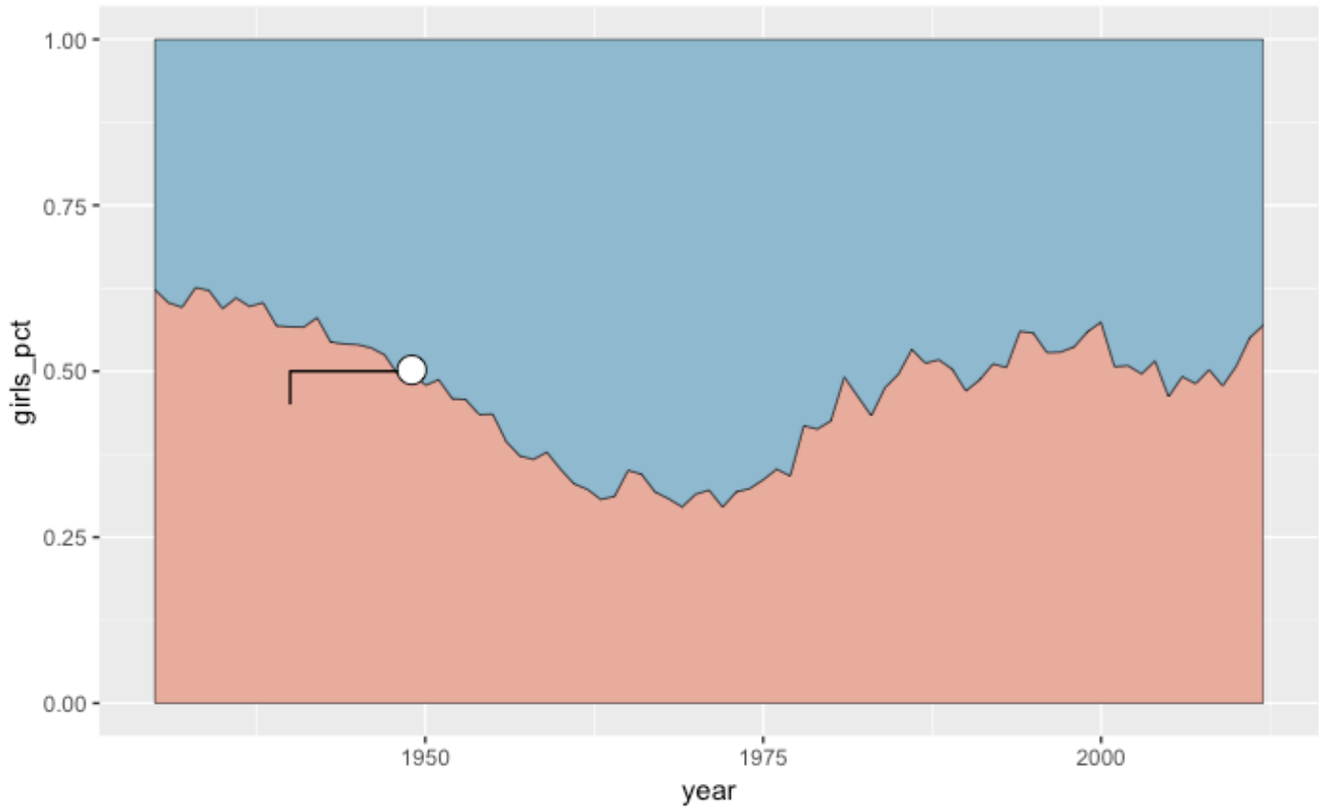
```
ggplot(data = df.jessie,
       mapping = aes(x = year, y = girls_pct)) +
  geom_area(fill = "#eaac9e",
            color = "black",
            size = 0.2) +
  geom_ribbon(mapping = aes(ymin = girls_pct, ymax = 1),
              fill = "#92bbd1",
              color = "black",
              size = 0.2) +
  geom_point(data = df.jessie %>%
               arrange(diff) %>%
               filter(row_number() == 1),
             shape = 21,
             fill = "white",
             size = 5) +
```



Homework 2: Exercise 1

Plotting

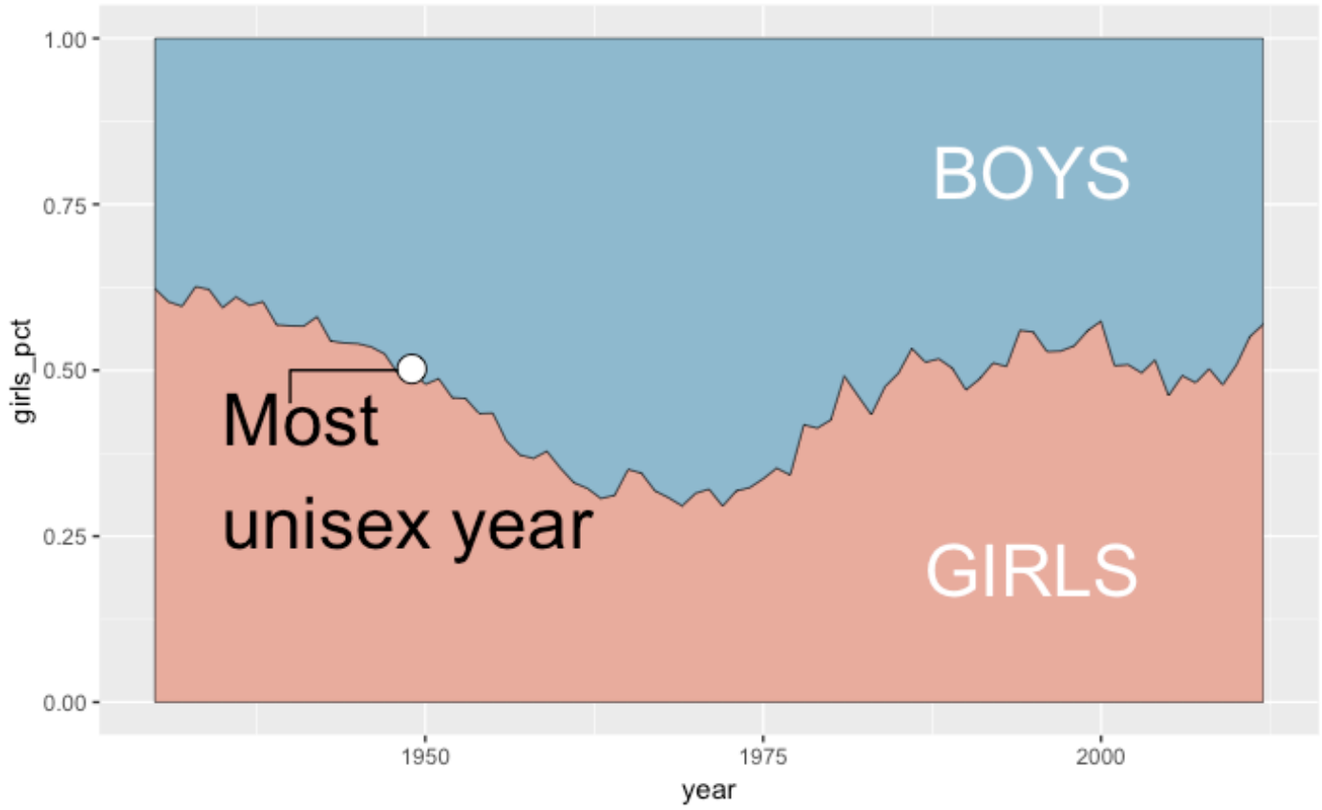
```
ggplot(data = df.jessie,
       mapping = aes(x = year, y = girls_pct)) +
  geom_area(fill = "#eaac9e",
            color = "black",
            size = 0.2) +
  geom_ribbon(mapping = aes(ymin = girls_pct, ymax = 1),
              fill = "#92bbd1",
              color = "black",
              size = 0.2) +
  geom_point(data = df.jessie %>%
               arrange(diff) %>%
               filter(row_number() == 1),
             shape = 21,
             fill = "white",
             size = 5) +
  geom_path(data = tibble(year = c(1940, 1940, 1948),
                          girls_pct = c(0.45, 0.5, 0.5)))
```



Homework 2: Exercise 1

Plotting

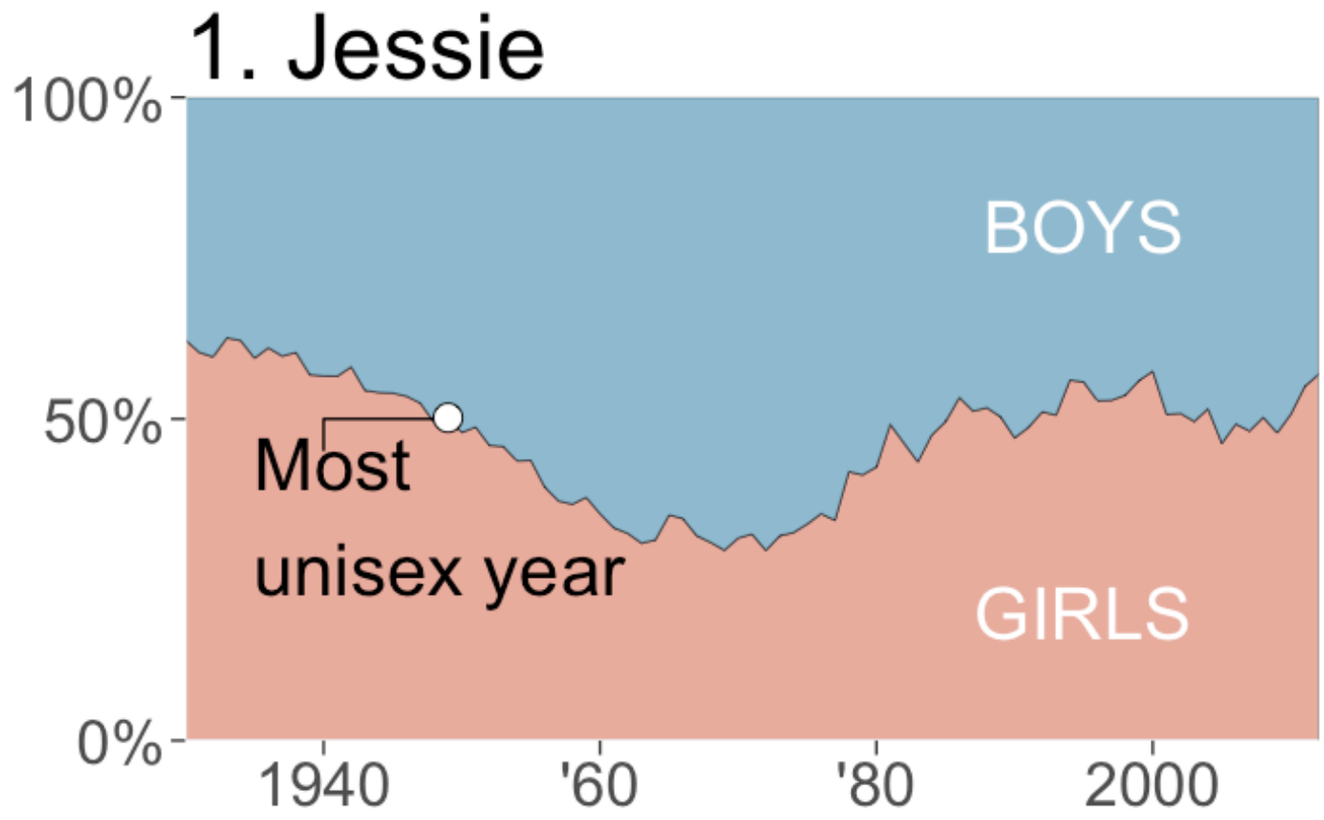
```
ggplot(data = df.jessie,
       mapping = aes(x = year, y = girls_pct)) +
  geom_area(fill = "#eaac9e",
            color = "black",
            size = 0.2) +
  geom_ribbon(mapping = aes(ymin = girls_pct, ymax = 1),
              fill = "#92bbd1",
              color = "black",
              size = 0.2) +
  geom_point(data = df.jessie %>%
               arrange(diff) %>%
               filter(row_number() == 1),
             shape = 21,
             fill = "white",
             size = 5) +
  geom_path(data = tibble(year = c(1940, 1940, 1949),
                          girls_pct = c(0.45, 0.5, 0.5))) +
  annotate(geom = "text",
           x = 1995,
           y = 0.8,
           label = "BOYS",
           color = "white",
           size = 10) +
  annotate(geom = "text",
           x = 1995,
           y = 0.2,
           label = "GIRLS",
           color = "white",
           size = 10) +
  annotate(geom = "text",
           x = 1935,
           y = 0.35,
           label = "Most\nunisex year",
           hjust = "left",
           size = 10)
```



Homework 2: Exercise 1

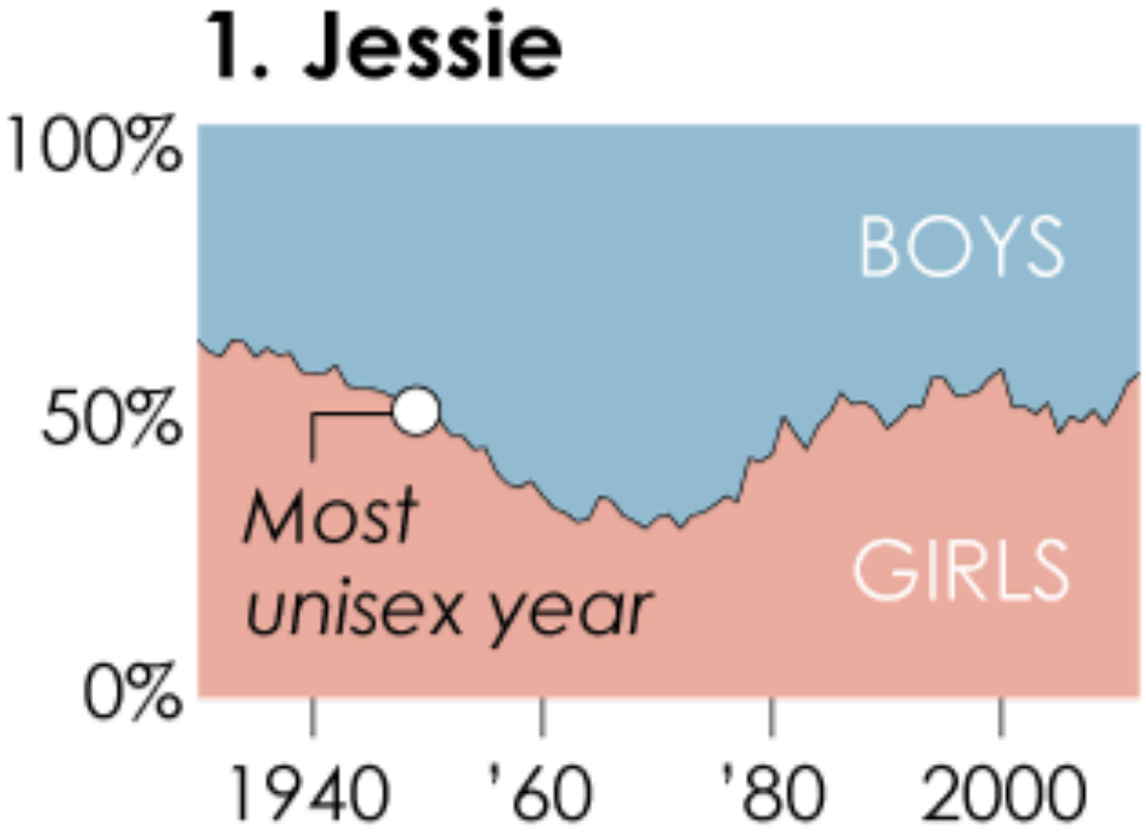
Plotting

```r
ggplot(data = df.jessie,
       mapping = aes(x = year, y = girls_pct)) +
  geom_area(fill = "#eaac9e",
            color = "black",
            size = 0.2) +
  geom_ribbon(mapping = aes(ymin = girls_pct, ymax = 1),
              fill = "#92bbd1",
              color = "black",
              size = 0.2) +
  geom_point(data = df.jessie %>%
               arrange(diff) %>%
               filter(row_number() == 1),
             shape = 21,
             fill = "white",
             size = 5) +
  geom_path(data = tibble(year = c(1940, 1940, 1949),
                          girls_pct = c(0.45, 0.5, 0.5))) +
  annotate(geom = "text",
           x = 1995,
           y = 0.8,
           label = "BOYS",
           color = "white",
           size = 10) +
  annotate(geom = "text",
           x = 1995,
           y = 0.2,
           label = "GIRLS",
           color = "white",
           size = 10) +
  annotate(geom = "text",
           x = 1935,
           y = 0.35,
           label = "Most\nunisex year",
           hjust = "left",
           size = 10) +
  scale_y_continuous(name = NULL,
                     limits = c(0, 1),
                     breaks = c(0, 0.5, 1),
                     labels = c("0%", "50%", "100%"),
                     expand = c(0, 0)) +
  scale_x_continuous(name = NULL,
                     breaks = seq(from = 1940, to = 2000, by = 20),
                     labels = c(1940, "'60", "'80", 2000),
                     expand = c(0, 0)) +
  labs(title = "1. Jessie") +
  theme(axis.ticks.length = unit(0.2, "cm"),
        axis.line = element_line(color = "white"),
        panel.background = element_rect(fill = "white"),
        text = element_text(size = 30,  family = "ArialMT"))
```

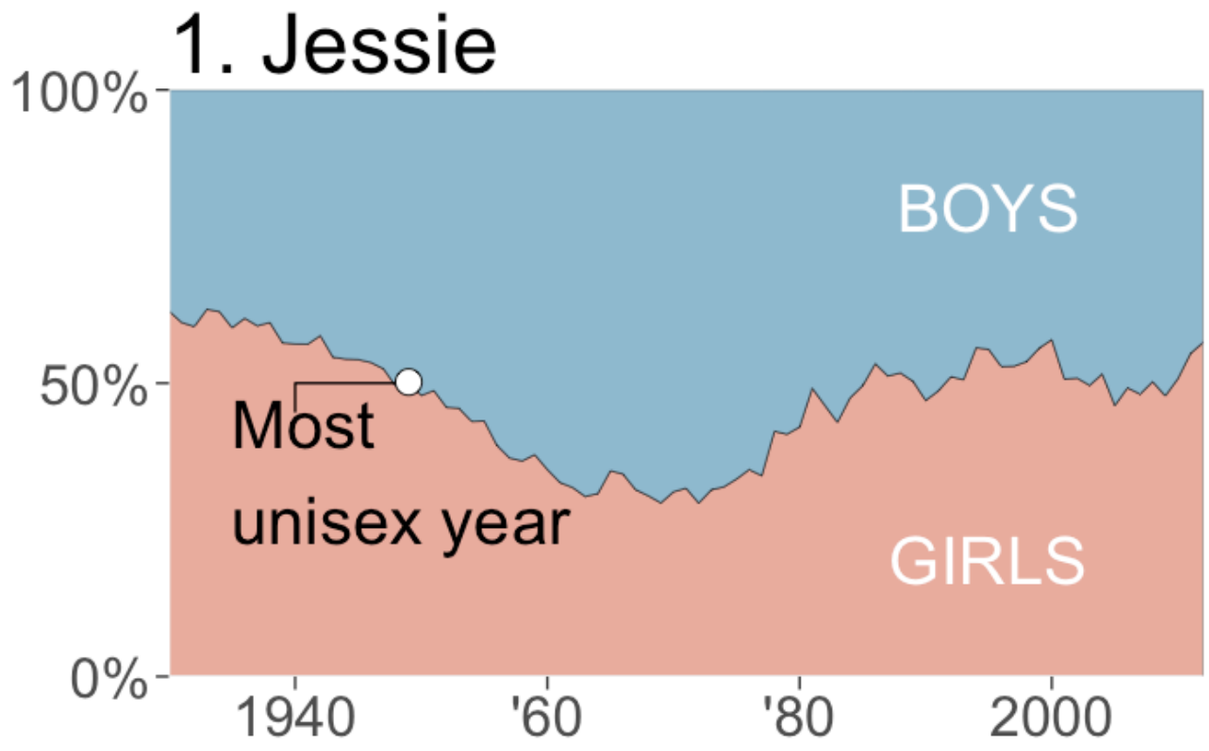


Homework 2: Exercise 1

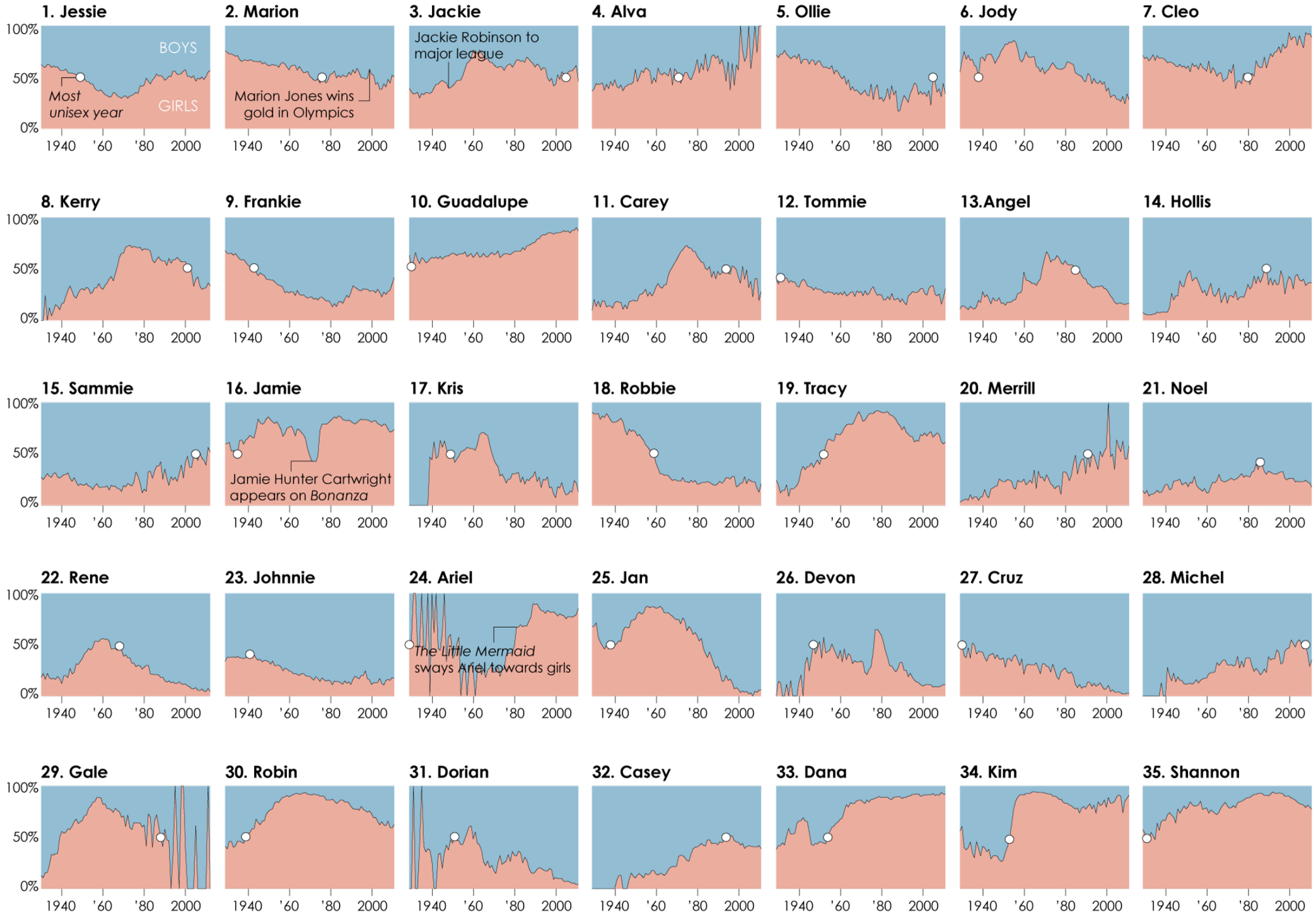
Target



Reproduction



Homework 2: Exercise 2



Homework 2: Exercise 2

Wrangling

Step 1: Filter out the correct names

```
1 # filter out common and consistent names
2 df.names = babynames %>%
3   filter(year >= 1930 & year <= 2012,
4          name != "Unknown") %>%
5   group_by(name) %>%
6   summarize(num_years = n_distinct(year),
7             num_babies = sum(n)) %>%
8   filter(num_years > 75,
9          num_babies > 9000)
```

Homework 2: Exercise 2

Wrangling

Step 2: Compute mean squared error

```
1 # compute se for each name and year
2 df.se = df.names %>%
3   left_join(babynames, by = "name") %>%
4   filter(year >= 1930 & year <= 2012) %>%
5   group_by(name, year) %>%
6   summarize(num_babies = sum(n),
7             pct_girls = sum(ifelse(sex == "F", n, 0)) / num_babies) %>%
8   mutate(se = (pct_girls - 0.5)^2)
```

The first charts rank names based on mean squared error from the 50-50 line. This chart on the other hand goes by count and a simple percent

Homework 2: Exercise 2

Wrangling

Step 2: Compute mean squared error

```
1 # compute mse for each name and select the top 35 names
2 df.selection = df.se %>%
3   group_by(name) %>%
4   summarize(mse = mean(se)) %>%
5   arrange(mse) %>%
6   head(35)
```

Homework 2: Exercise 2

Wrangling

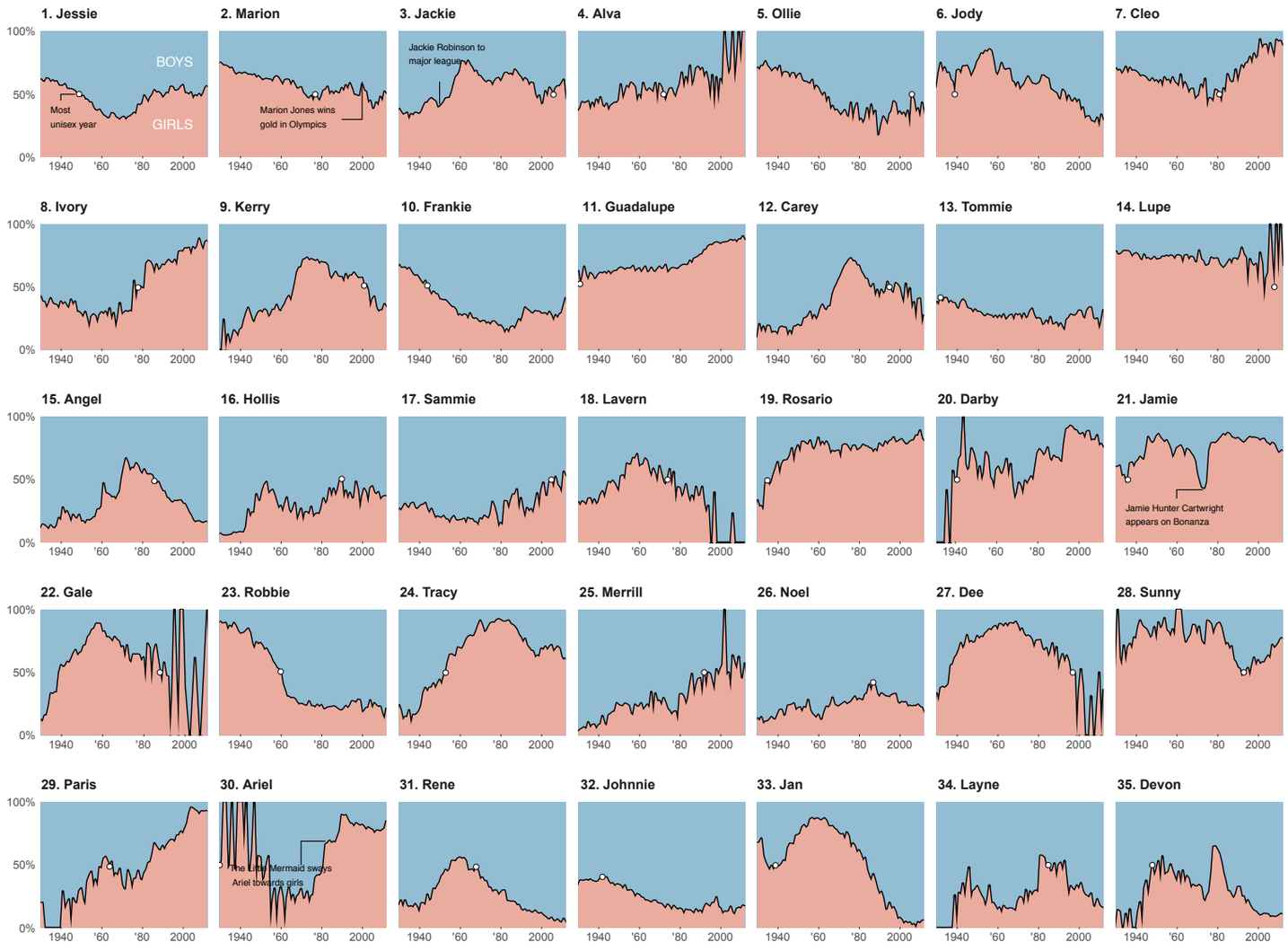
Step 3: Find most unisex year

(same as for Exercise 1)

Homework 2: Exercise 2

Plotting

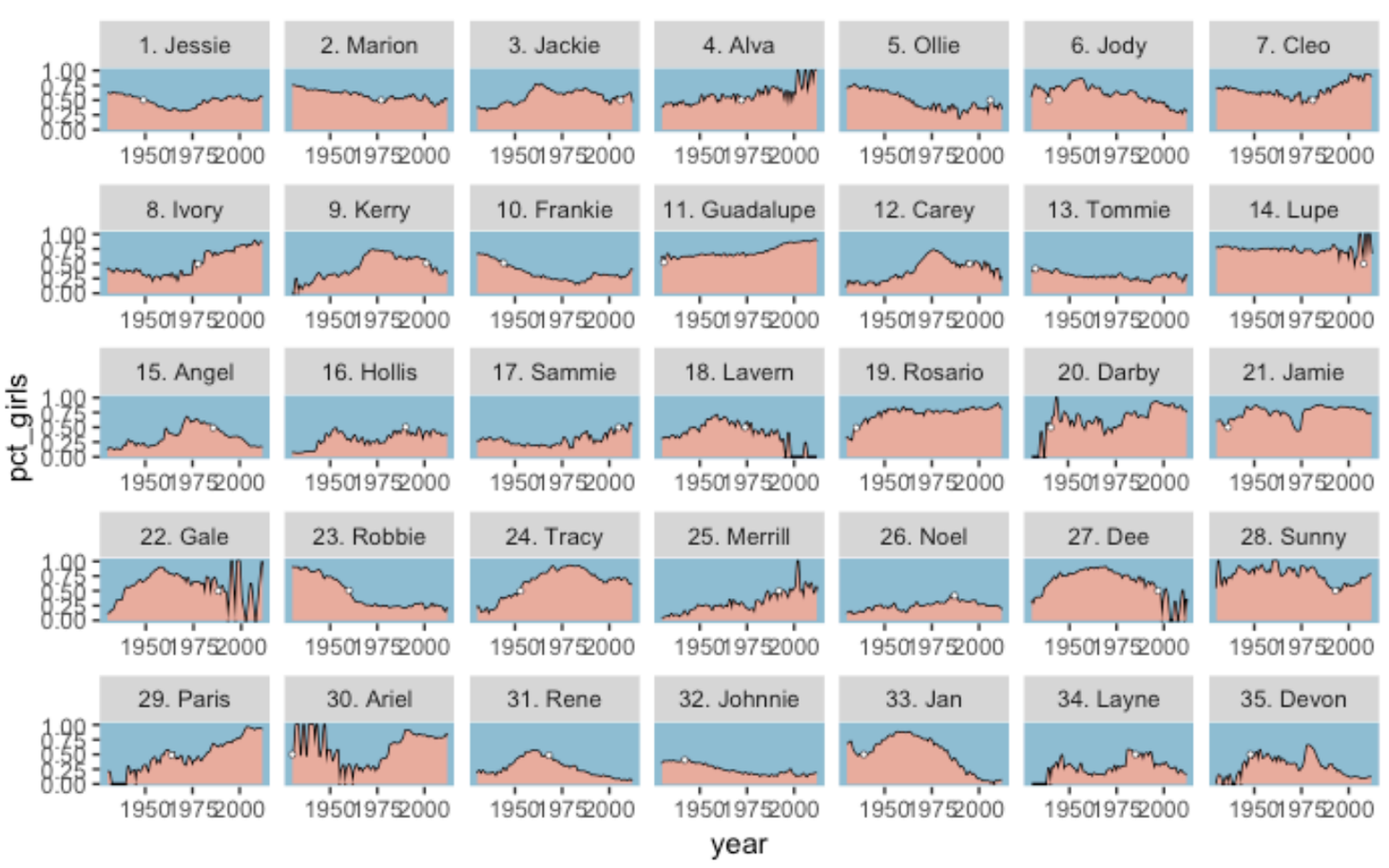
```r
ggplot(df.unisex, aes(x = year, y = pct_girls)) +
  geom_line() +
  geom_area(fill = "#eaac9e") +
  geom_point(data = df.most_unisex_year,
             fill = "white",
             pch = 21,
             size = 1,
             stroke = 0.2) +
  geom_text(data = df.text,
            mapping = aes(x = x, y = y, label = text),
            size = 1.5,
            hjust = 0,
            vjust = 1) +
  geom_text(data = tibble(name_label = as.factor("1. Jessie"),
                          text = "GIRLS",
                          x = 2005,
                          y = 0.3),
            mapping = aes(x = x, y = y, label = text),
            size = 2,
            hjust = 1,
            vjust = 1,
            color = "white") +
  geom_text(data = tibble(name_label = as.factor("1. Jessie"),
                          text = "BOYS",
                          x = 2005,
                          y = 0.8),
            mapping = aes(x = x, y = y, label = text),
            size = 2,
            hjust = 1,
            vjust = 1,
            color = "white") +
  geom_path(data = tibble(year = c(1940, 1940, 1947),
                          pct_girls = c(0.45, 0.5, 0.5),
                          name_label = as.factor("1. Jessie")),
            size = 0.2) +
  geom_path(data = tibble(year = c(1990, 2000, 2000),
                          pct_girls = c(0.3, 0.3, 0.6),
                          name_label = as.factor("2. Marion")),
            size = 0.2) +
  geom_path(data = tibble(year = c(1950, 1950),
                          pct_girls = c(0.6, 0.42),
                          name_label = as.factor("3. Jackie")),
            size = 0.2) +
  geom_path(data = tibble(year = c(1960, 1960, 1973),
                          pct_girls = c(0.35, 0.42, 0.42),
                          name_label = as.factor("21. Jamie")),
            size = 0.2) +
  geom_path(data = tibble(year = c(1970, 1970, 1982),
                          pct_girls = c(0.5, 0.69, 0.69),
                          name_label = as.factor("30. Ariel")),
            size = 0.2) +
  facet_wrap(vars(name_label), ncol = 7, scales = "free_x") +
  scale_y_continuous(NULL,
                     breaks = 0:2/2,
                     labels = scales::percent,
                     expand = c(0, 0)) +
  scale_x_continuous(NULL,
                     breaks = seq(from = 1940, to = 2000, by = 20),
                     labels = c(1940, "'60", "'80", 2000),
                     expand = c(0, 0)) +
  theme(panel.background = element_rect(fill = "#92bdd3"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        strip.background = element_blank(),
        strip.text = element_text(hjust = -0.1, face = "bold", size = 6),
        axis.text.y = element_text(hjust = 1),
        axis.ticks.y = element_blank(),
        axis.ticks.length = unit(0, "cm"),
        text = element_text(size = 6, family = "ArialMT"))
```



Homework 2: Exercise 2

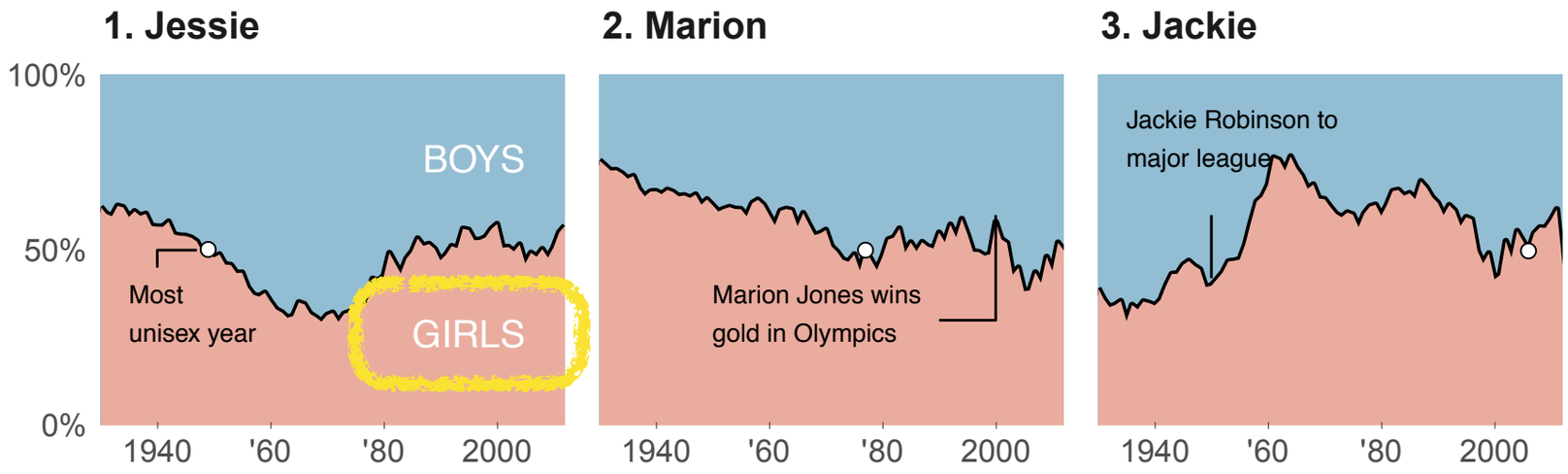
Plotting

```
ggplot(df.unisex, aes(x = year, y = pct_girls)) +
 geom_line() +
 geom_area(fill = "#eaac9e") +
 geom_point(data = df.most_unisex_year,
            fill = "white",
            shape = 21,
            size = 1,
            stroke = 0.2) +
 facet_wrap(vars(name_label), ncol = 7, scales = "free_x") +
 theme(panel.background = element_rect(fill = "#92bdd3"),
       panel.grid.major = element_blank(),
       panel.grid.minor = element_blank())
```



Homework 2: Exercise 2

Plotting

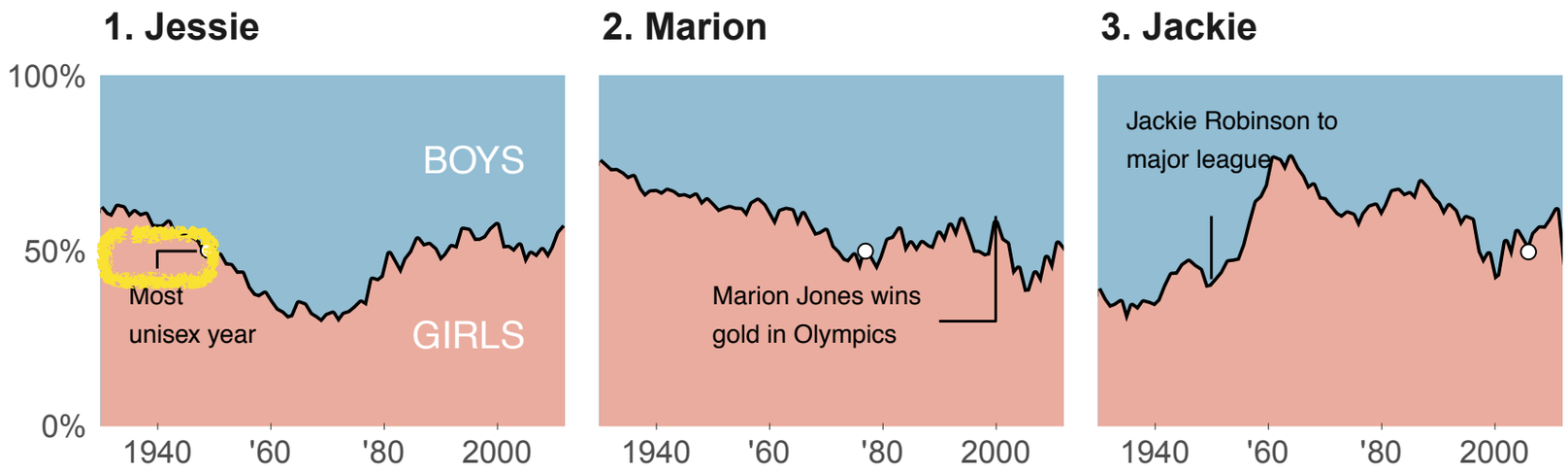


Adding **text** to a particular facet

```
1 geom_text(data = tibble(name_label = as.factor("1. Jessie"),
2                         text = "GIRLS",
3                         x = 2005,
4                         y = 0.3),
5             mapping = aes(x = x, y = y, label = text),
6             size = 2,
7             hjust = 1,
8             vjust = 1,
9             color = "white")
```

Homework 2: Exercise 2

Plotting



Adding **a path** to a particular facet

```
1 geom_path(data = tibble(year = c(1940, 1940, 1948),
2                         pct_girls = c(0.45, 0.5, 0.5),
3                         name_label = as.factor("1. Jessie")),
4             size = 0.2)
```

Outline

- Statistical inference
- Central limit theorem
- Sampling distributions
- p-values
- Confidence intervals

Statistical inference

Statistical inference

The process of making claims about a population based on information from a sample.

Life would be easy if we were able to observe the whole population -- we could simply do descriptive analyses!

Key question:

What can we infer about the population from our sample?

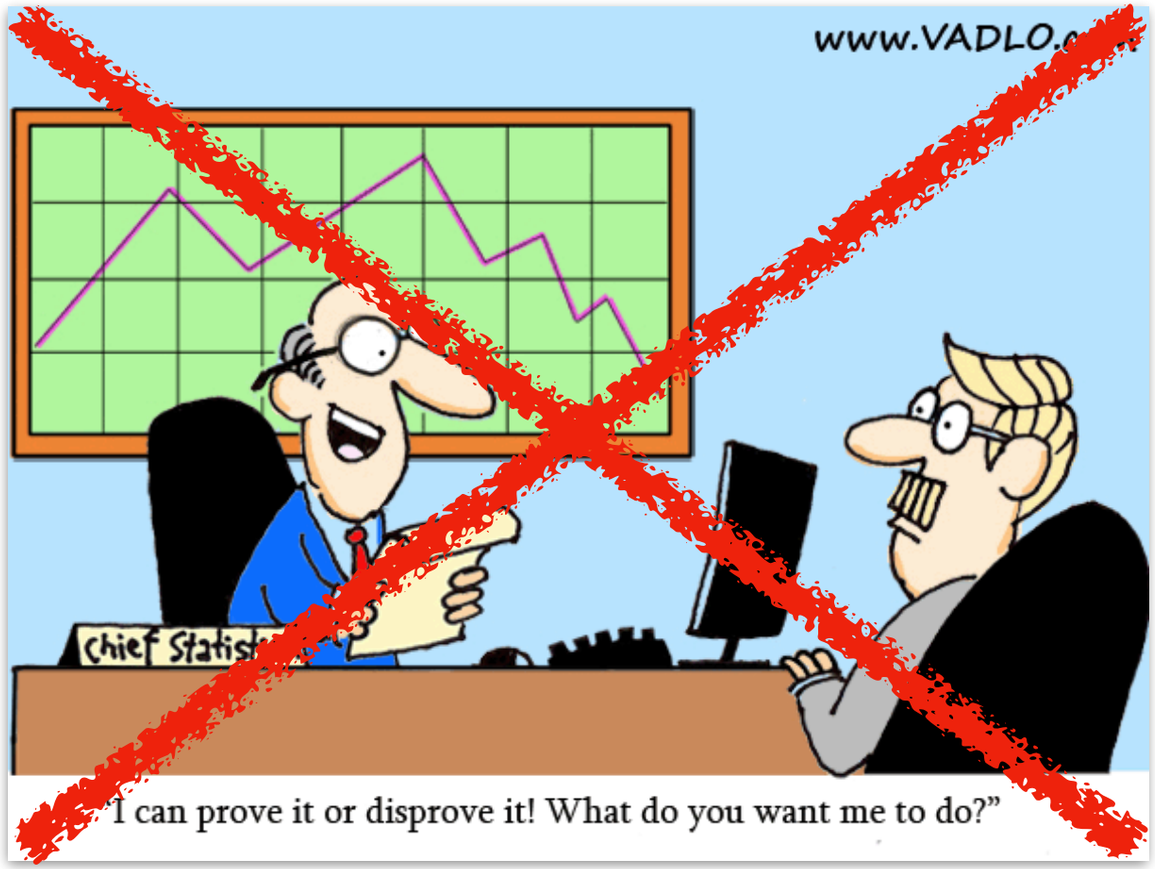
Statistical inference

Key question:

What can we infer about the population from our sample?

Answer:

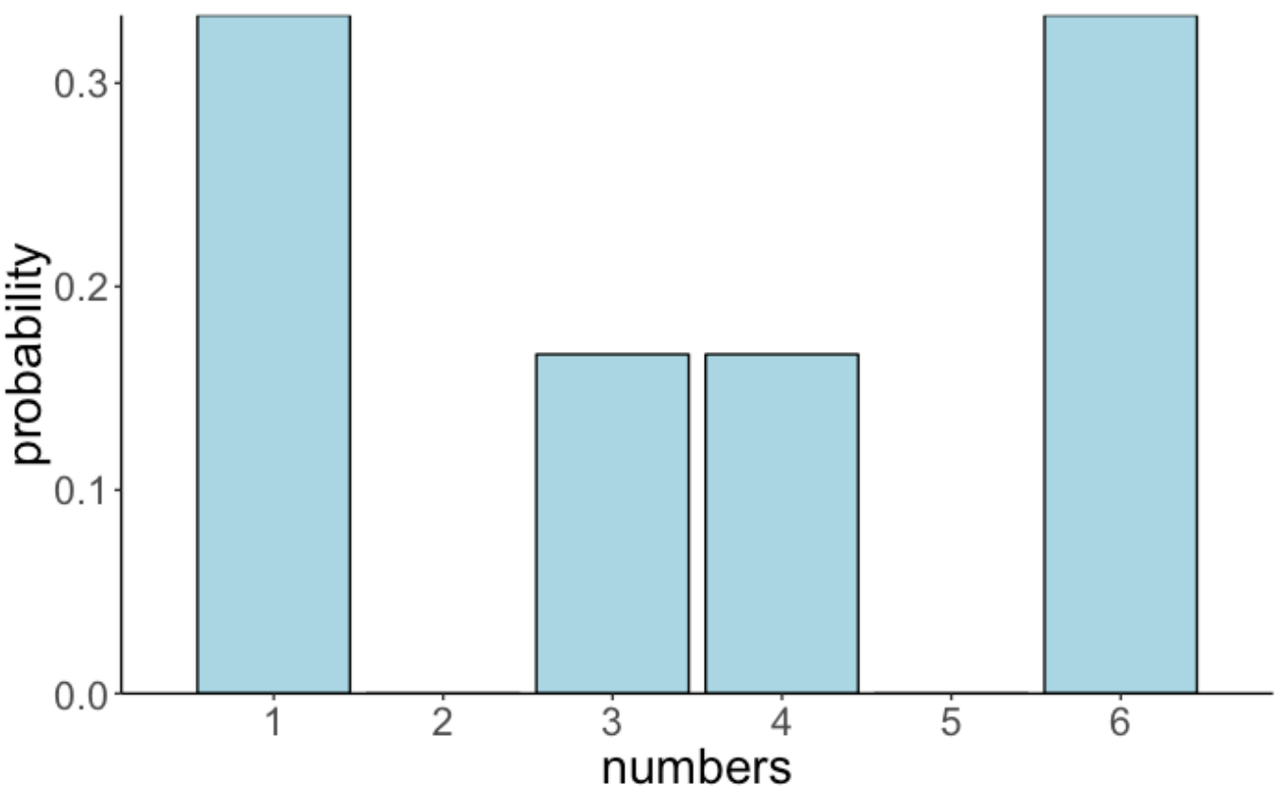
- is not trivial
- mathematical, statistical, philosophical (Bayesian vs. frequentist) machinery involved
- **important**: we can never make deterministic statements!
- we can only make probabilistic claims



Central limit theorem

Central limit theorem

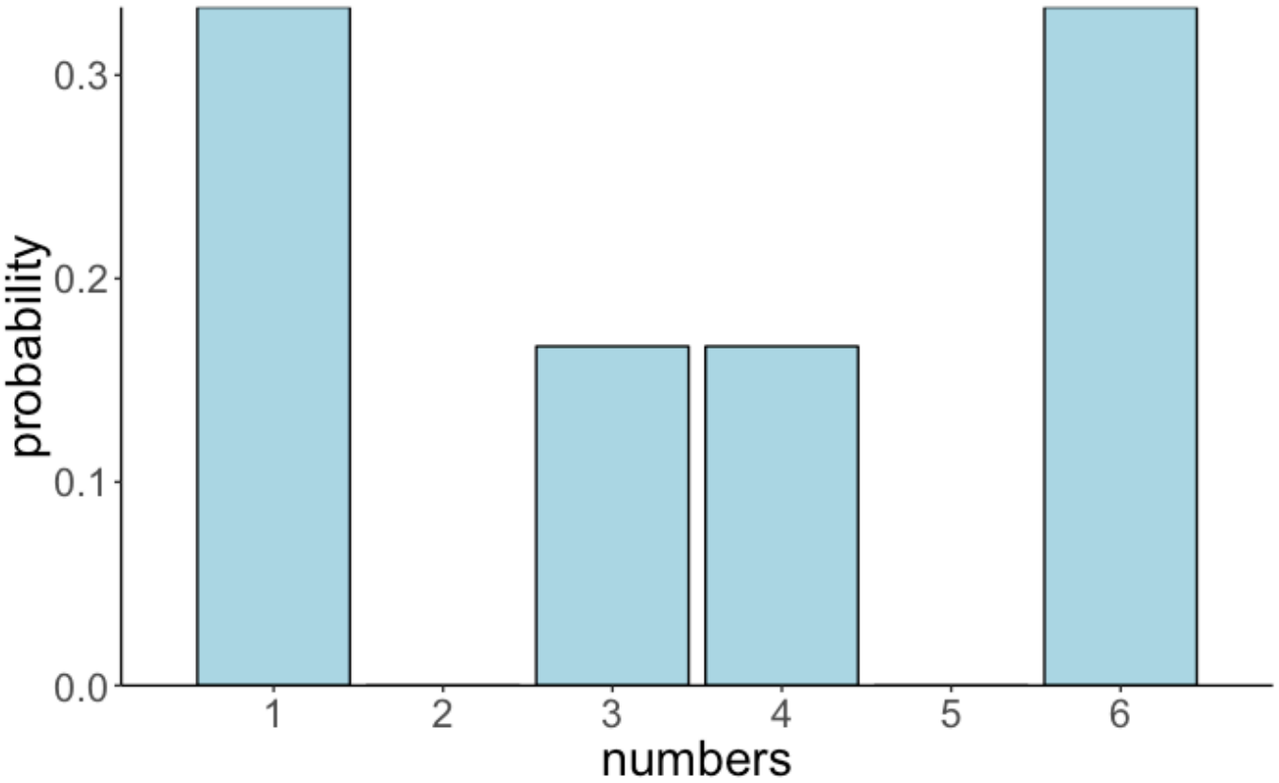
heavy metal distribution



population distribution

Central limit theorem

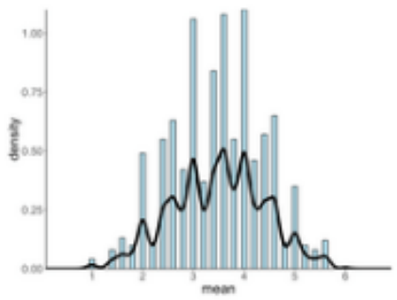
heavy metal distribution

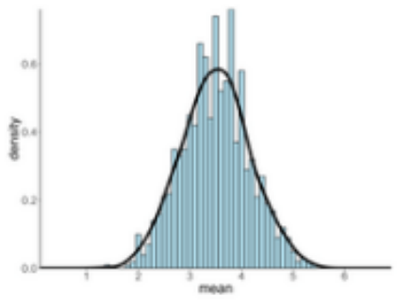


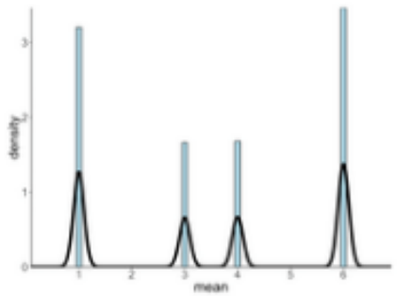
population distribution

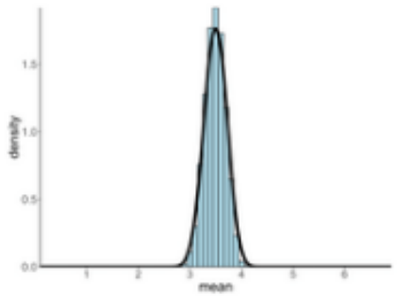
1. draw **1000 samples** of **size 1**
2. plot a histogram of the samples

What would the distribution look like? (N = 1)

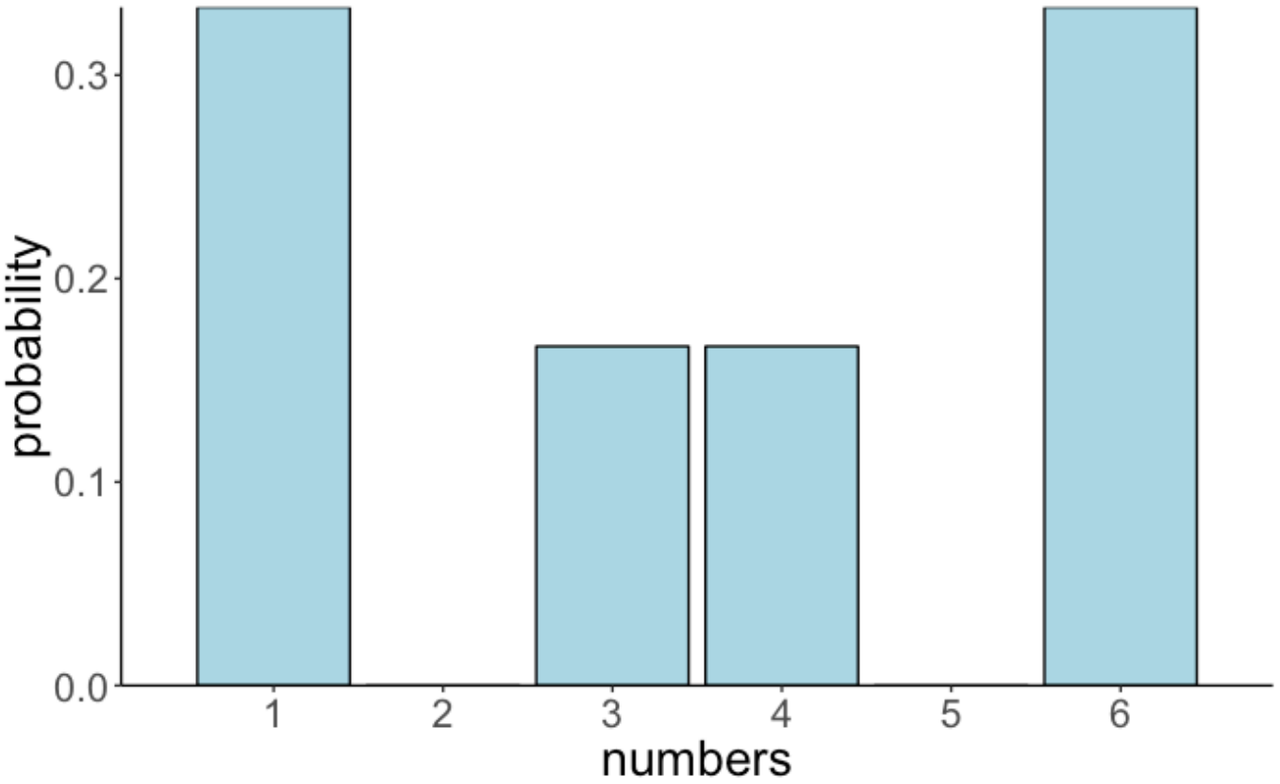






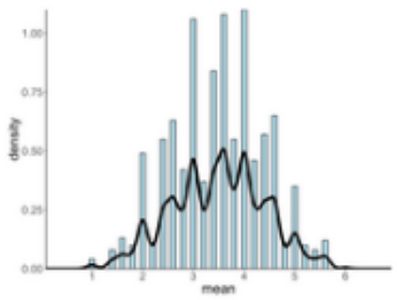


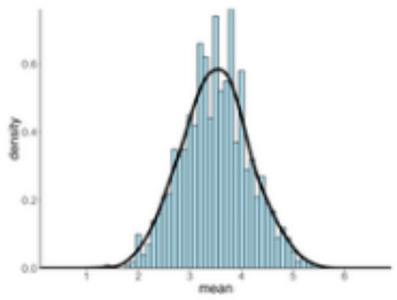
Central limit theorem

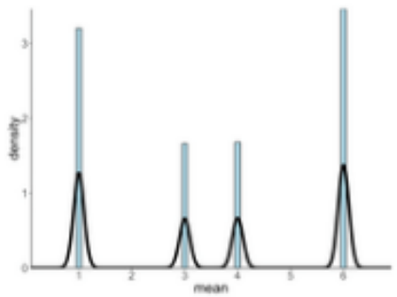


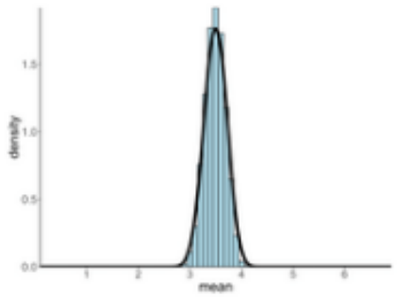
1. draw **1000 samples** of **size 100**
2. calculate the mean of each sample
3. plot a histogram of the sample means

What would the distribution look like? (N = 100)

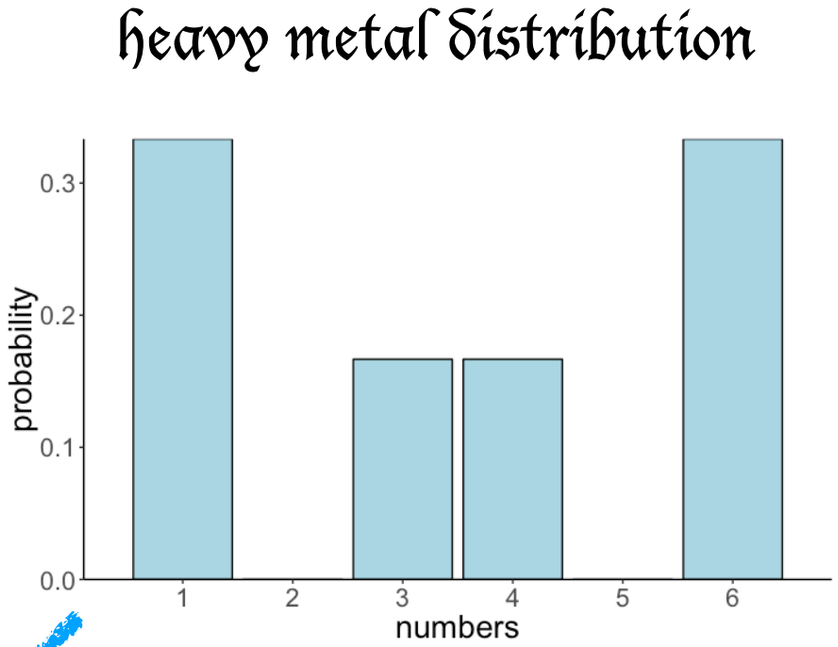






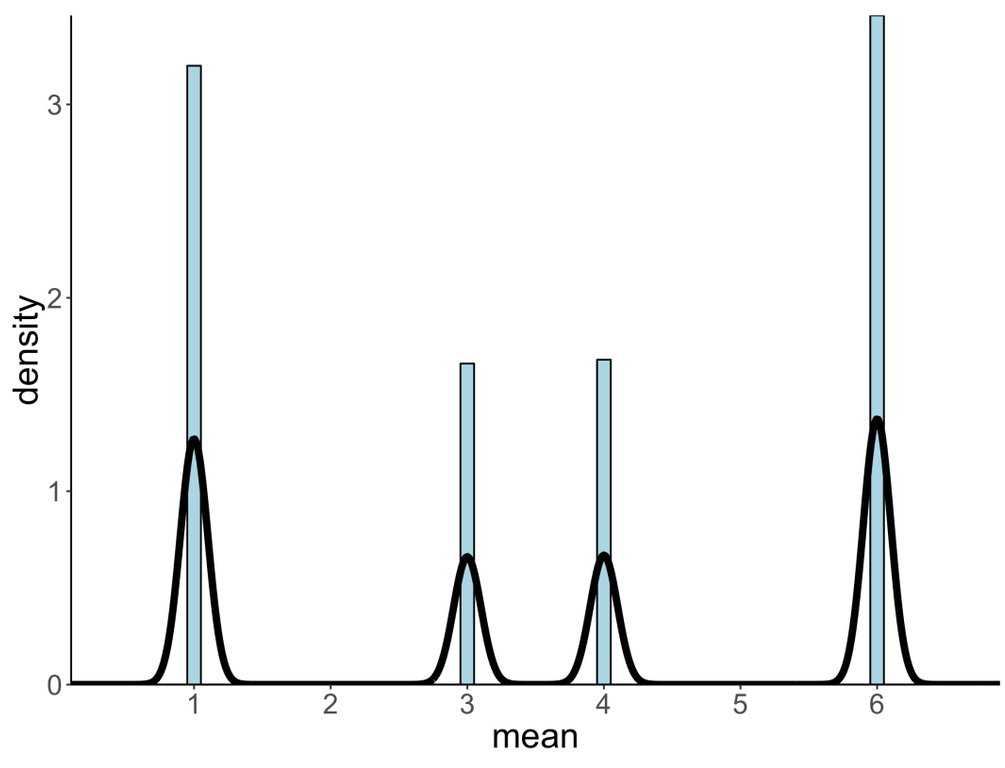


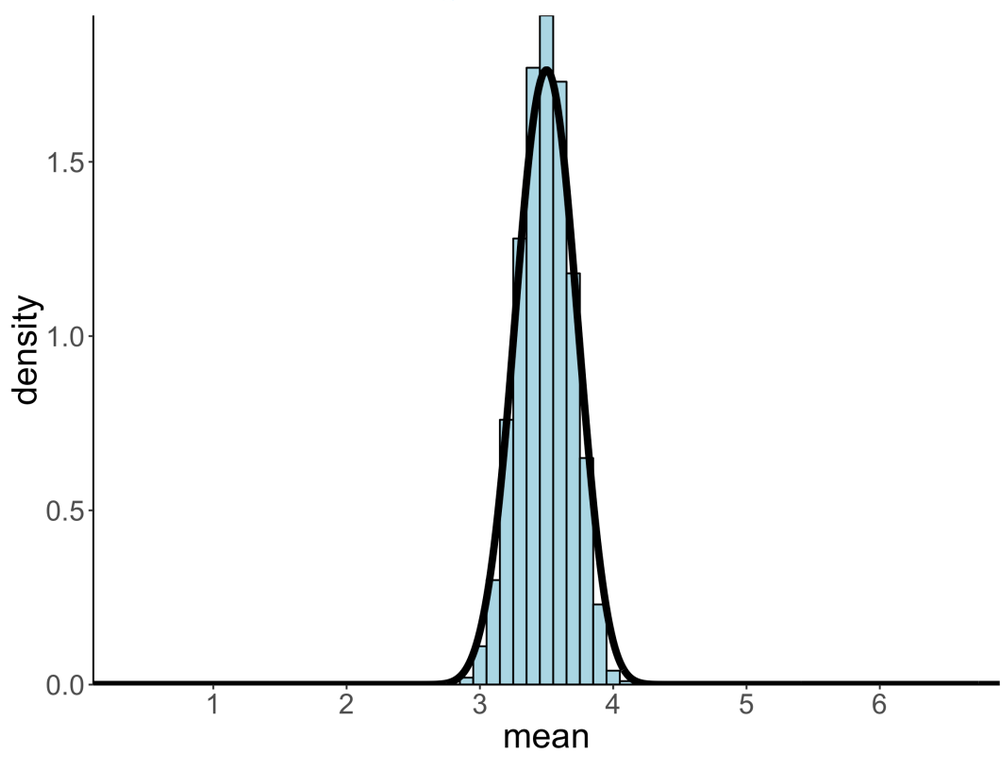
Central limit theorem



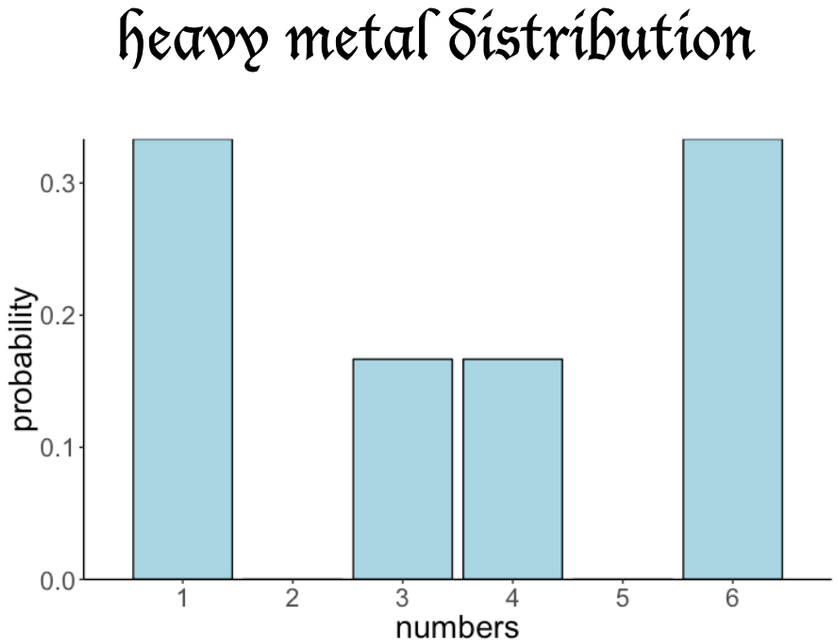
sample size = 1

sample size = 100

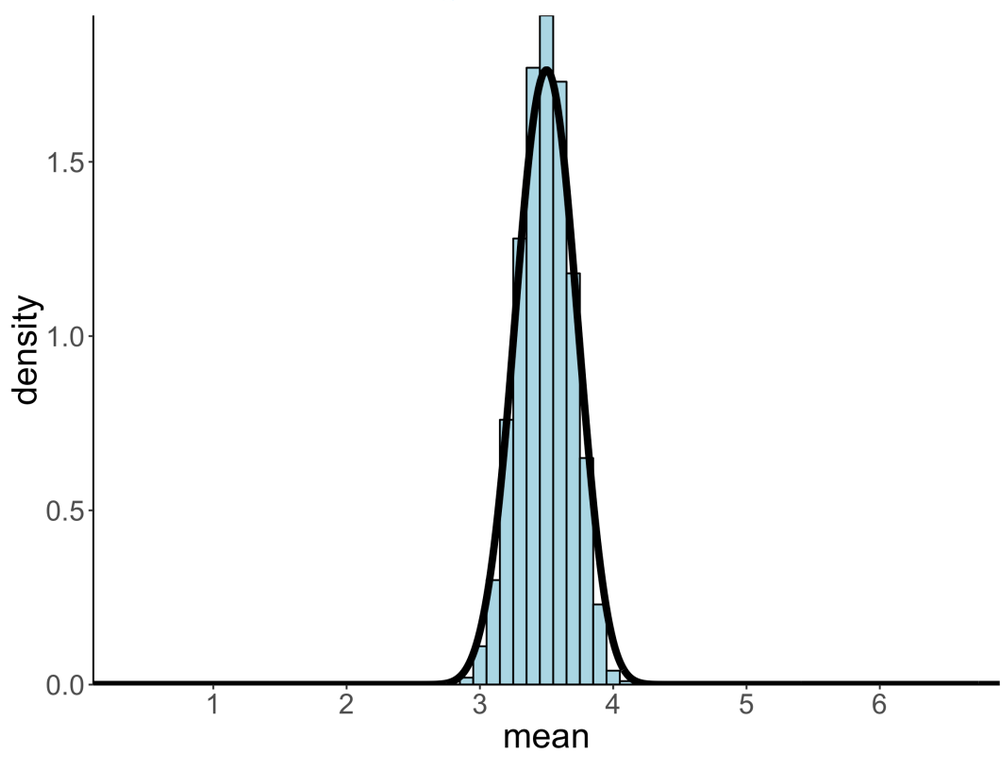




Central limit theorem



sample size = 100



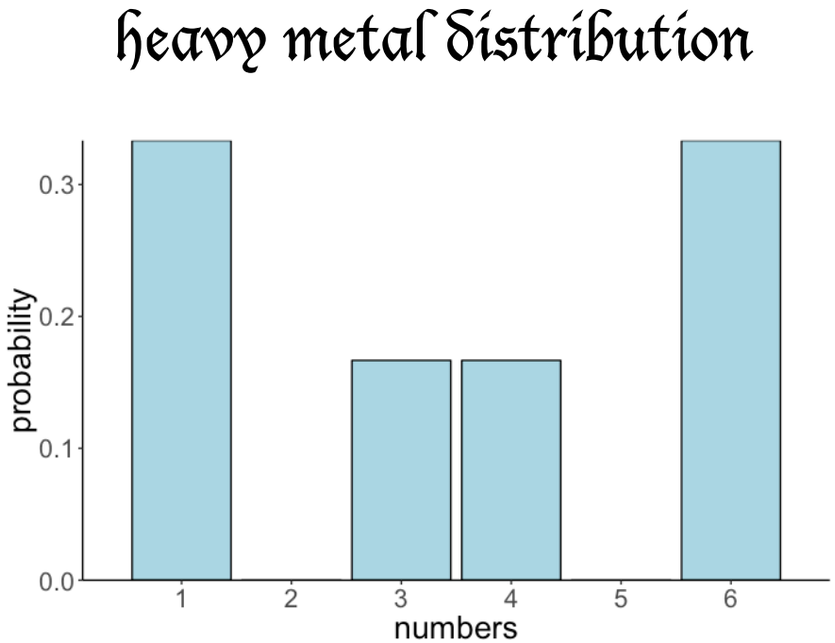
The Central Limit Theorem (CLT) states that the sample mean of a **sufficiently large number of independent and identically distributed (i.i.d.) random variables is approximately normally distributed**. The larger the sample, the better the approximation.

The theorem is a key ("central") concept in probability theory because it implies that **statistical methods that work for normal distributions can be applicable to many problems involving other types of distributions**.

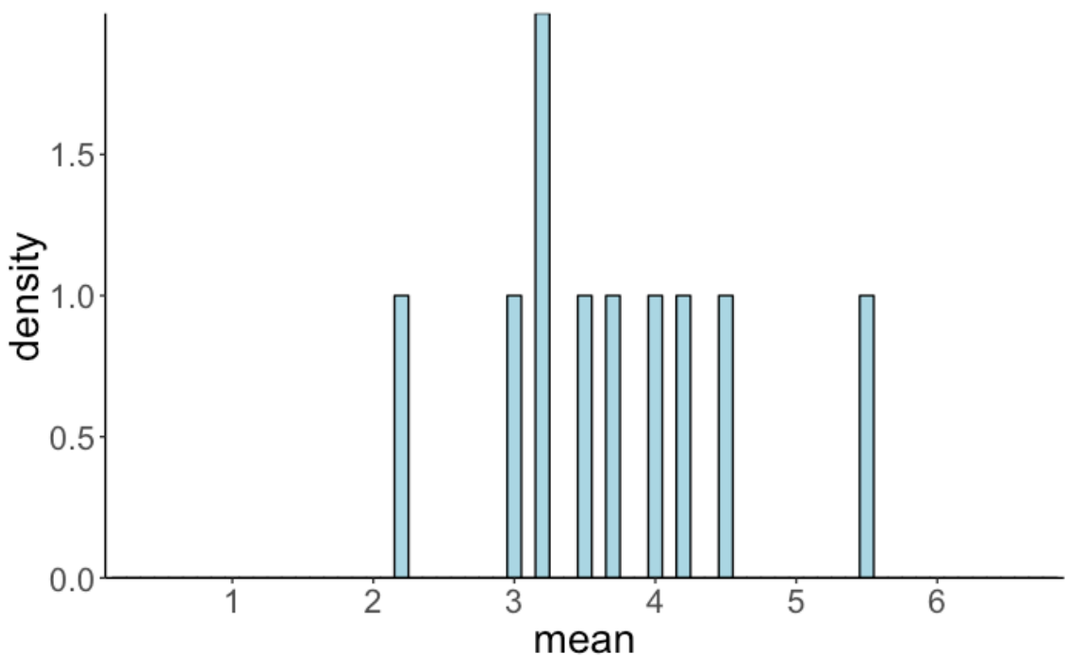
Central limit theorem

sample size = 4

number of samples = 10



sample	draw_1	draw_2	draw_3	draw_4
1	1	6	6	4

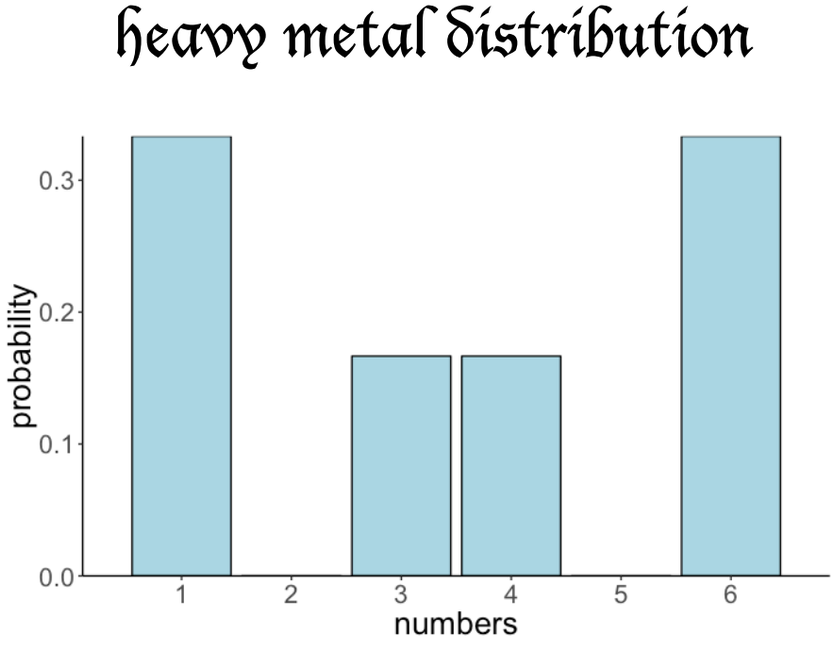


Central limit theorem

sample size = 100

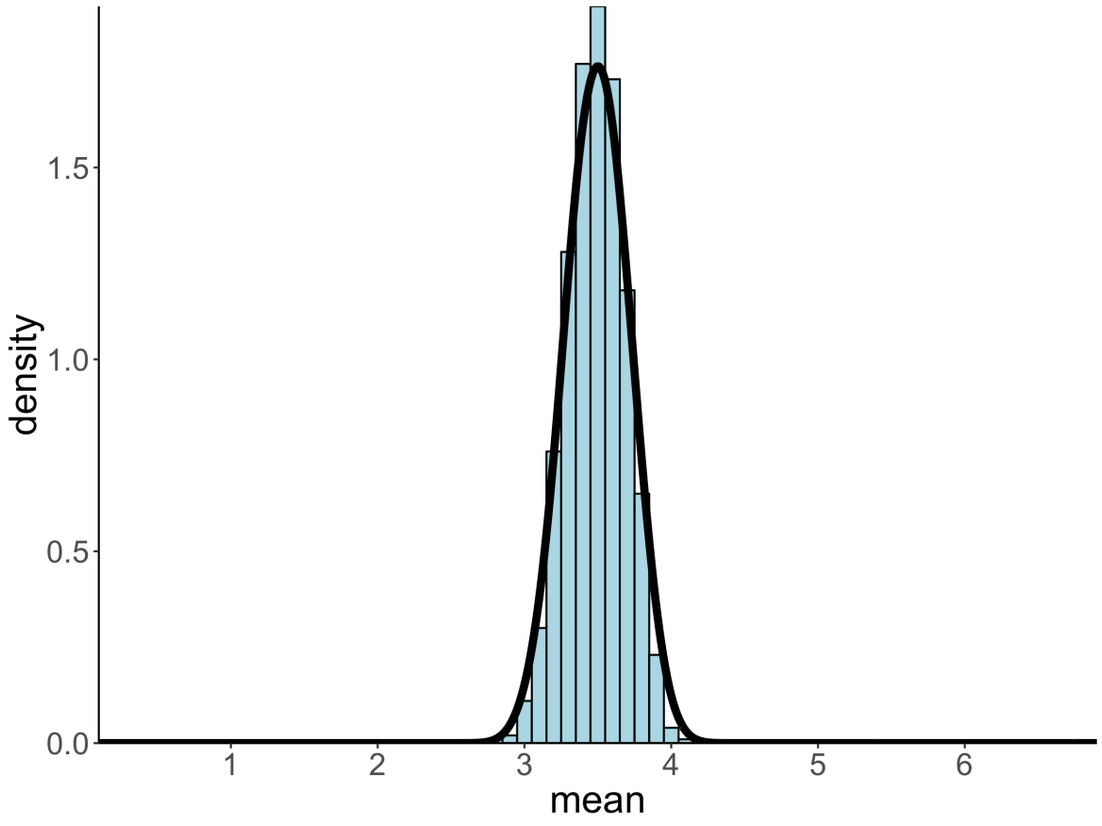
number of samples = 1000

...



sample	draw_1	draw_2	draw_3	draw_4	sample_mean
1	1	6	6	4	4.25
2	1	4	4	6	3.75
3	6	1	1	1	2.25
4	3	6	3	6	4.50
5	3	4	6	3	4.00
6	4	1	6	1	3.00
7	1	6	1	6	3.50
8	4	6	6	6	5.50
9	6	1	3	3	3.25
10	3	1	3	6	3.25

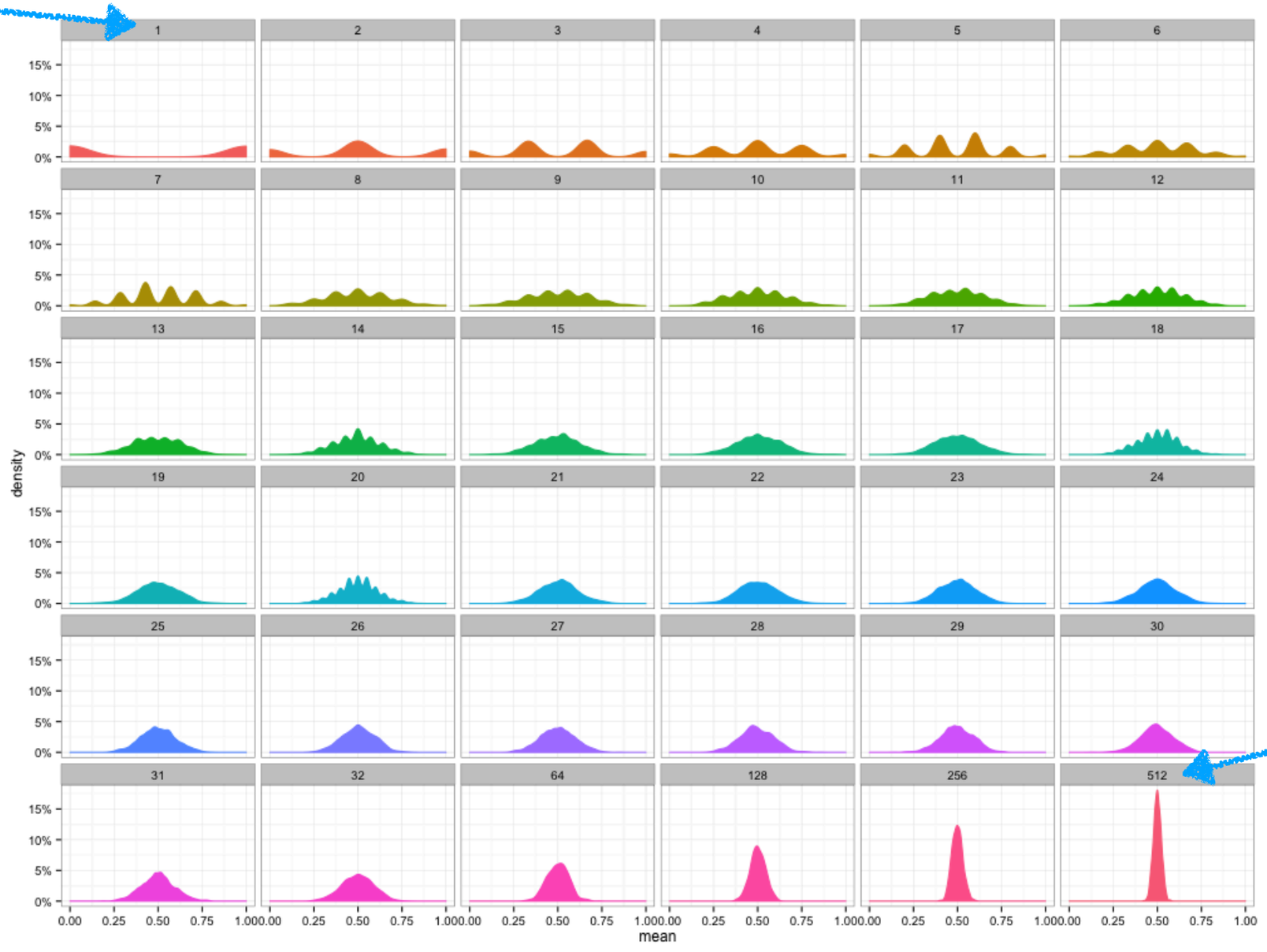
⋮



Central limit theorem

Binomial distribution: generate 0s and 1s and calculate their mean

sample size



more normal & narrower

The larger the sample, the better the approximation.

Central limit theorem

Central limit theorem

seeing theory demo

Central limit theorem

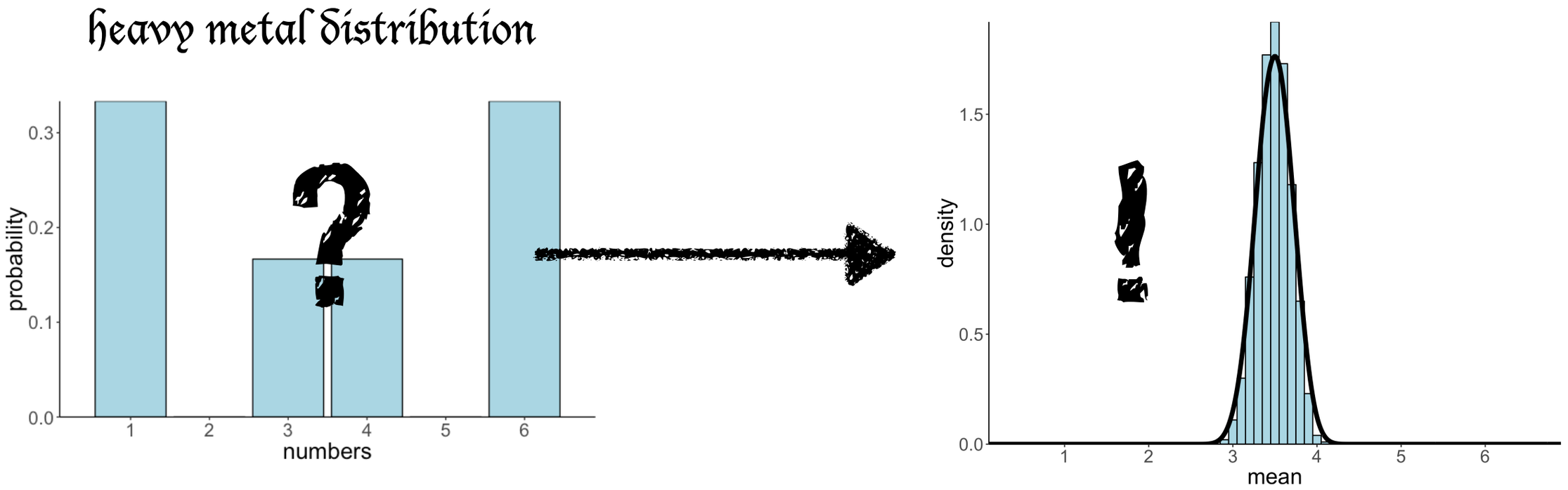
Why do we care?

- normal distributions **are everywhere**
  - outcomes that are affected by many factors that combine additively will tend to be normally distributed

Central limit theorem

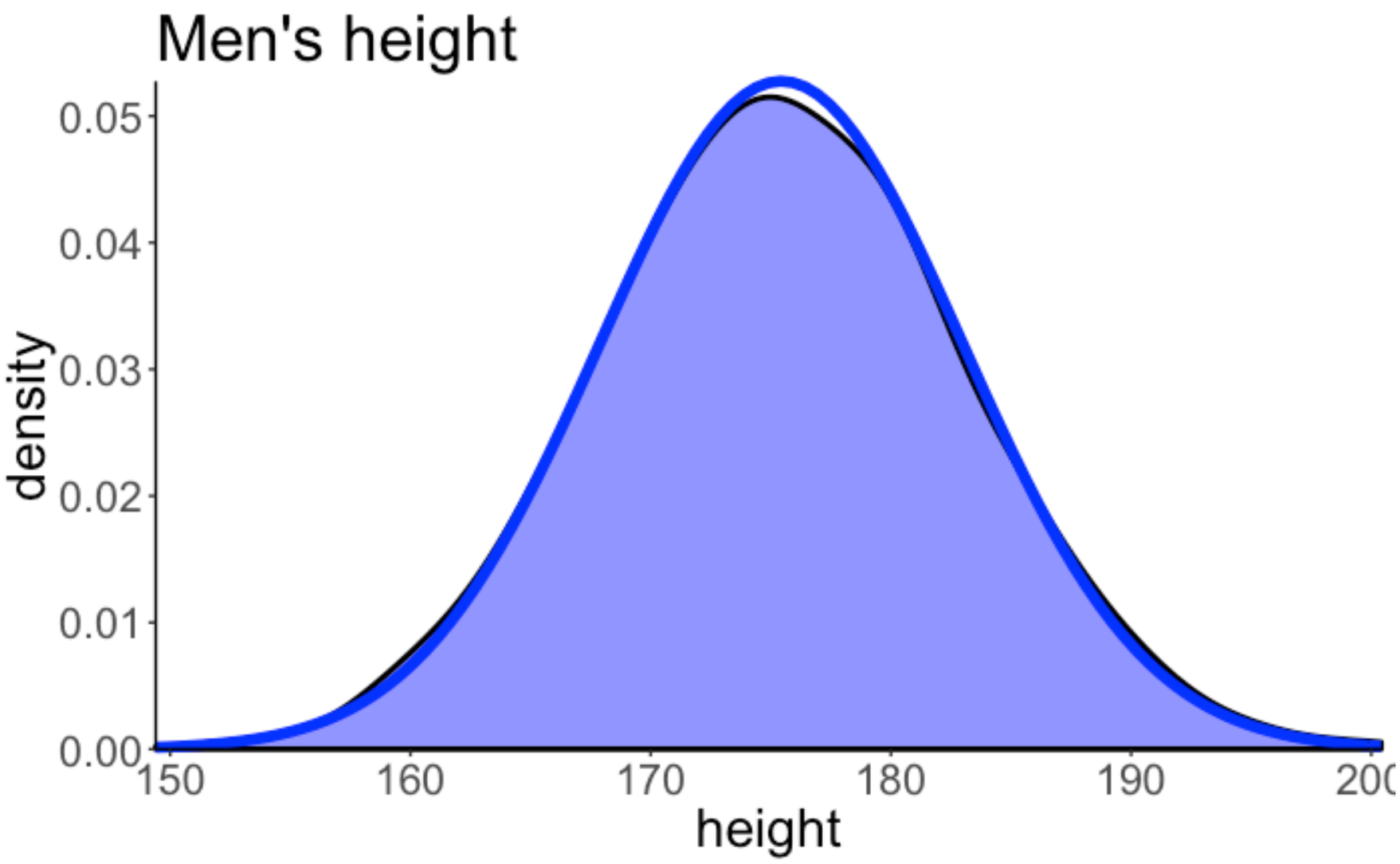
Why do we care?

- even if we don't know the true underlying probability distribution (and we cannot accurately predict any individual observation), we can still make inferences about the mean of the distribution (even when that distribution is far from normal)



Central limit theorem

Where it breaks down



men's height is normally distributed

```
library("NHANES")
```

Data from the US National Health and Nutrition Examination Study

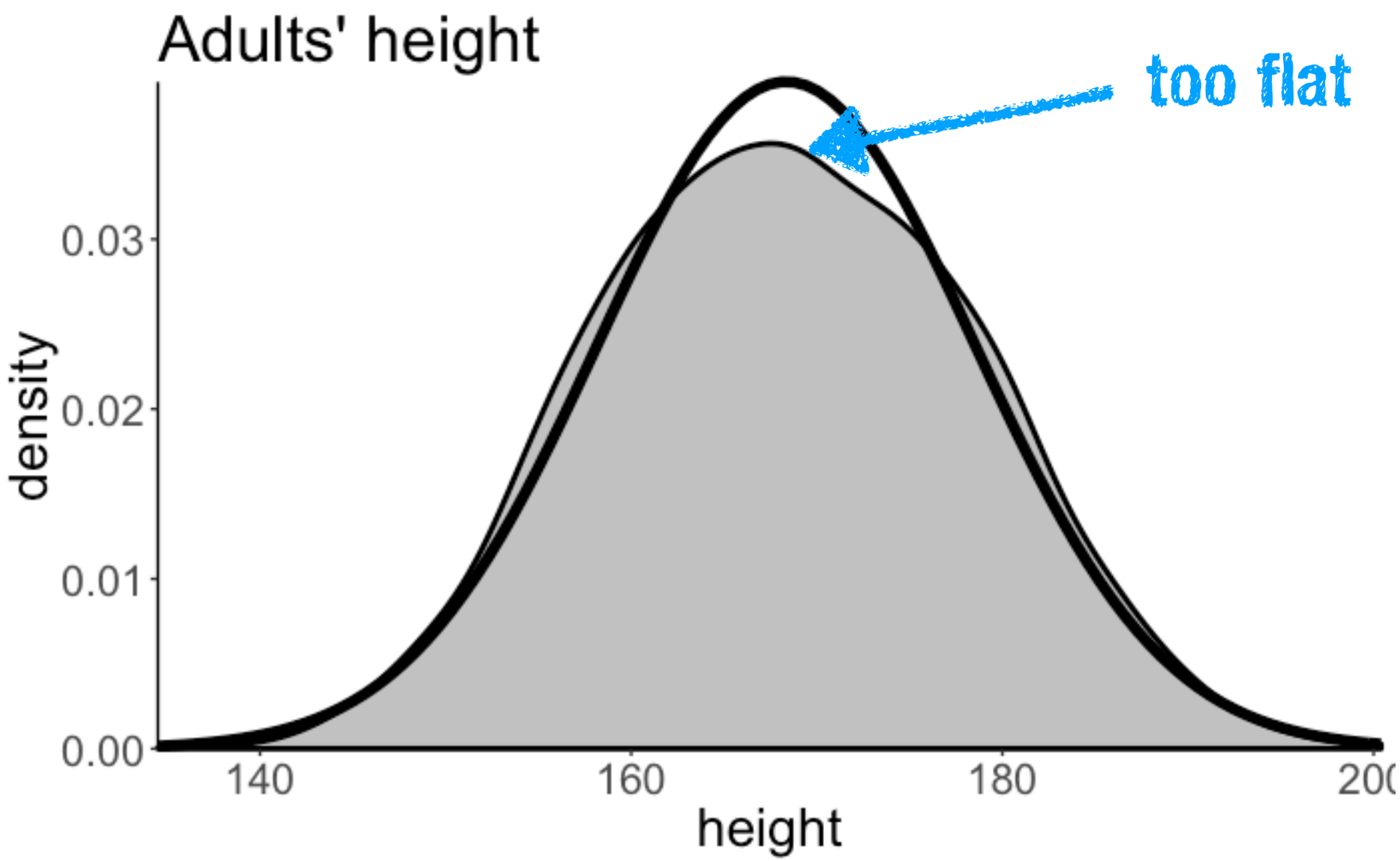
Central limit theorem

Where it breaks down

women's height is normally distributed

Central limit theorem

Where it breaks down

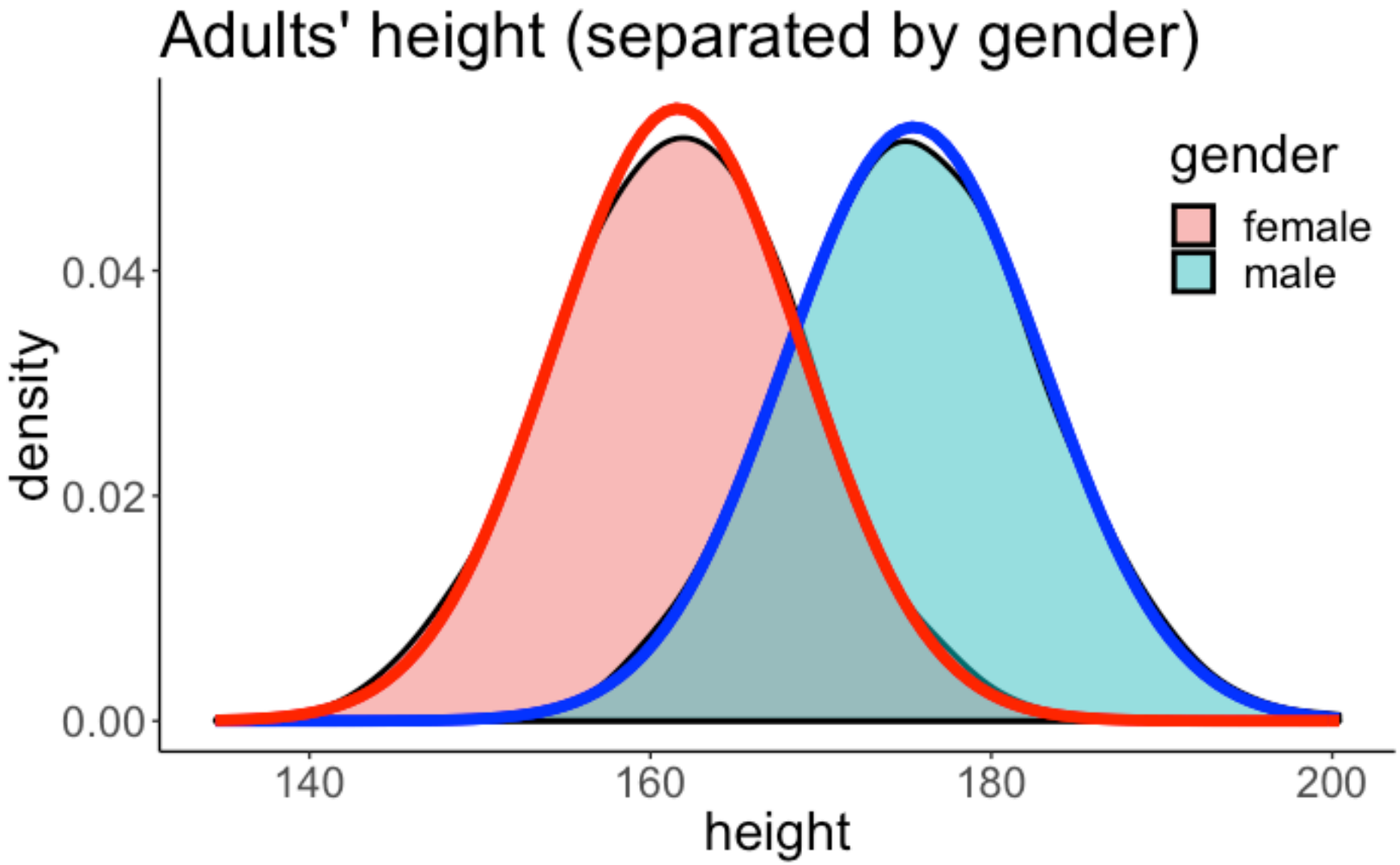


adults' height is **not** normally distributed

one factor (gender) accounts for much of the variance

Central limit theorem

Where it breaks down



adults' height is a mixture of two normal distributions

Central limit theorem

Where it works

- sufficiently large number of i.i.d. factors that combine additively

Where it breaks down

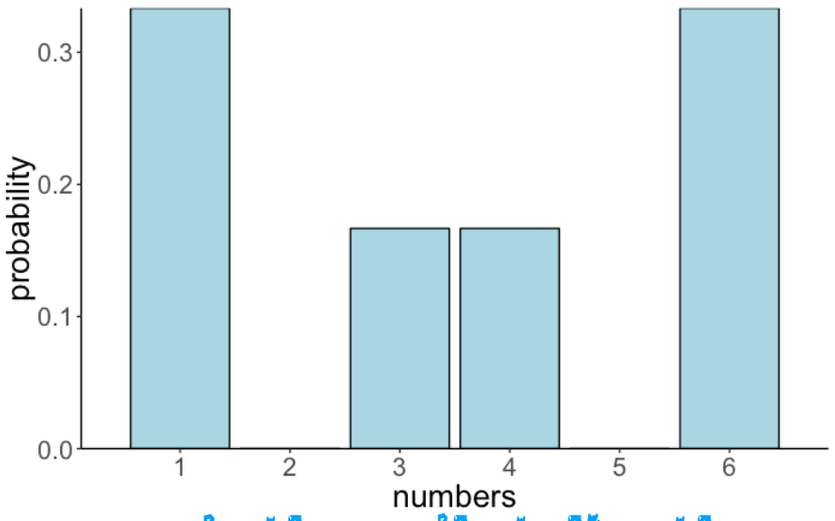
1. when one factor affects the outcome more strongly than others
2. when processes involve strong dependence
   - e.g. rich get richer dynamics (distribution of wealth)
3. when factors combine multiplicatively
   - many diseases (e.g. how cancer cells divide and grow)
   - (such phenomena often follow a log-normal distribution)

Sampling distributions

Statistical inference

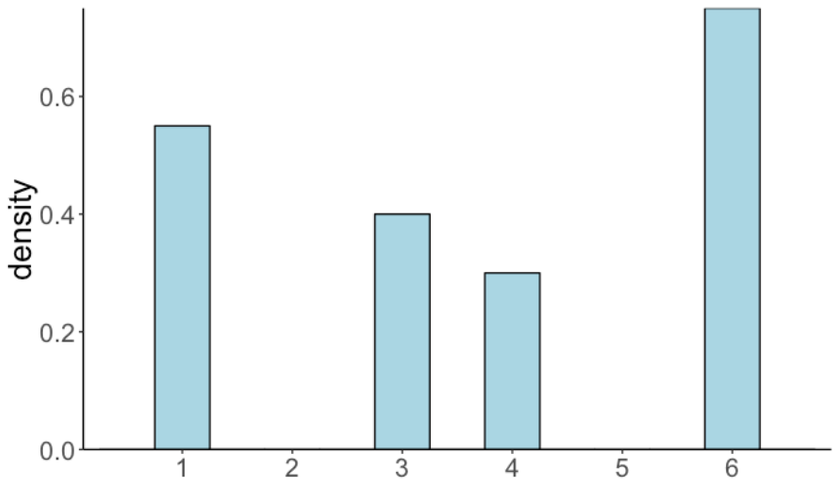
The process of making claims about a population based on information from a sample.

heavy metal distribution



sampling

inference



population distribution

our sample

Statistical inference

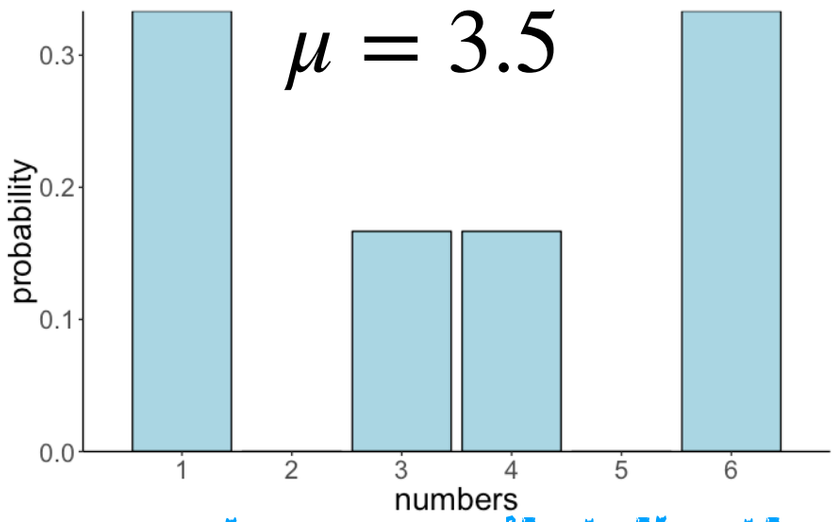
what's the
population mean?

sample mean = 3.725

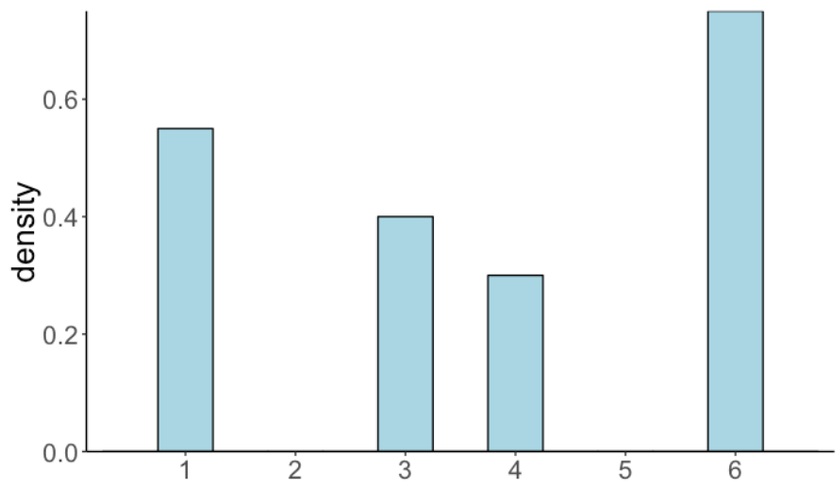
standard deviation = 2.05

n = 40

heavy metal distribution



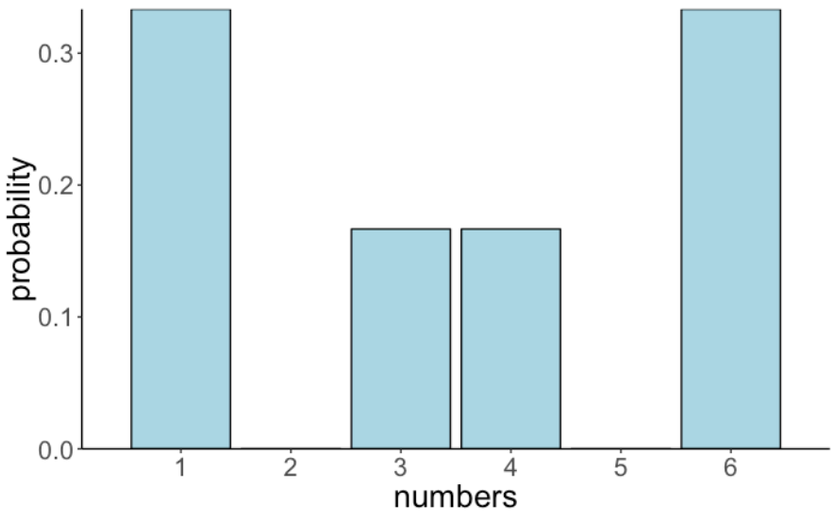
true unknown distribution



our sample

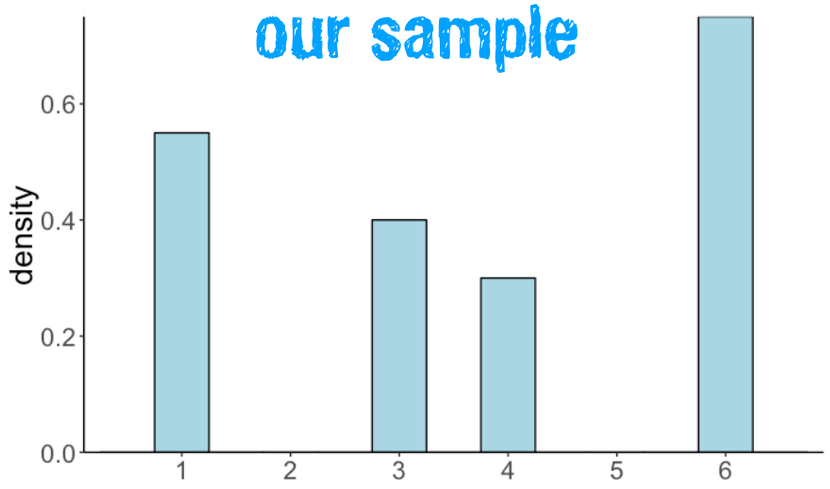
Sampling variation

heavy metal distribution

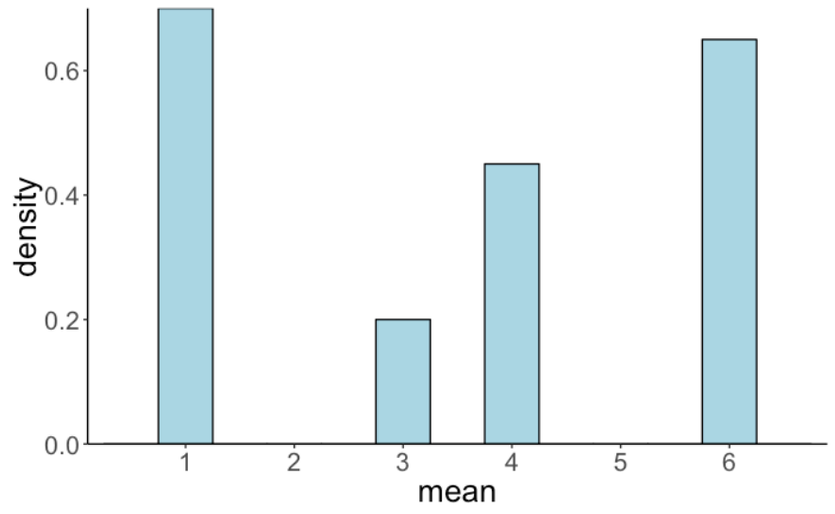


population distribution

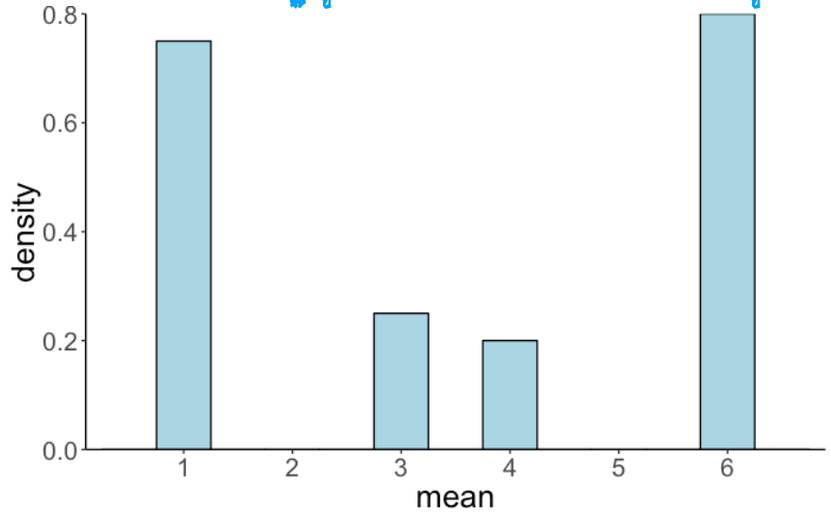
our sample



hypothetical sample



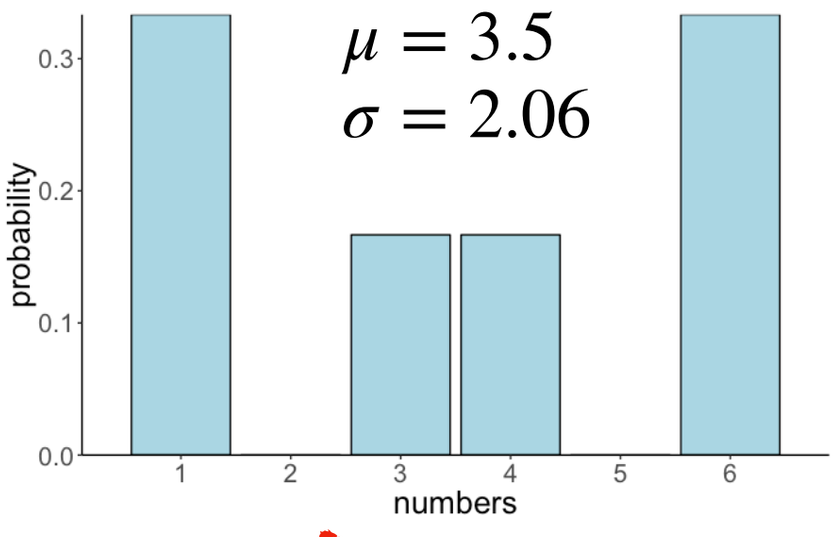
another hypothetical sample



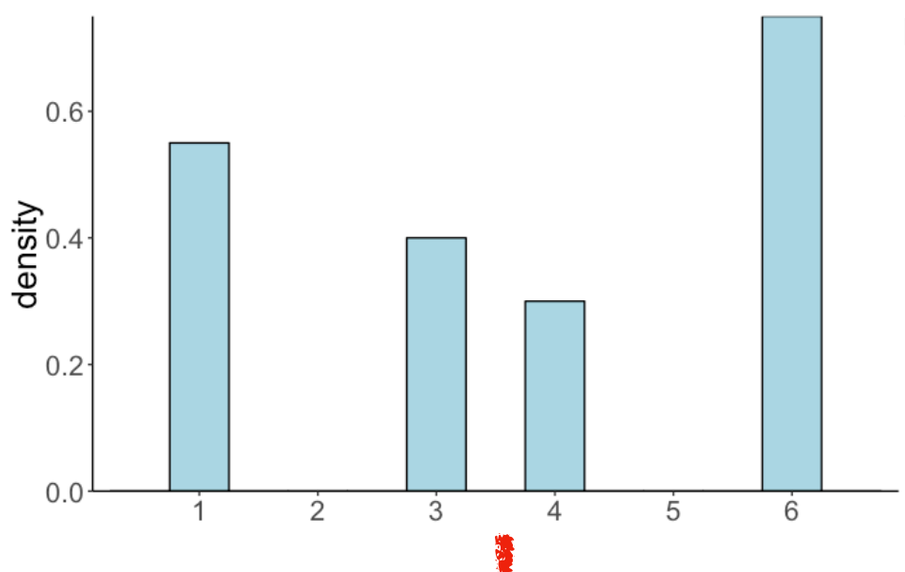
Sampling distribution

population distribution

heavy metal distribution

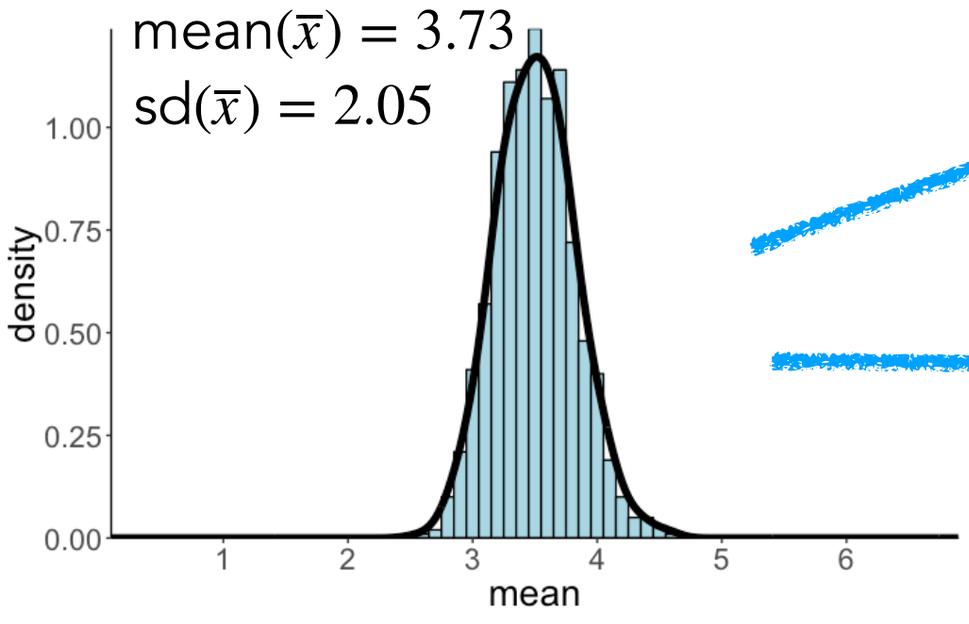


our sample



$\text{mean}(x) = 3.73$
$\text{sd}(x) = 2.05$
$\text{n} = 40$

sampling distribution



p-values

confidence intervals

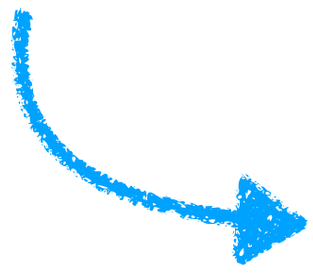
Underlying principle of statistical testing

1. Define population, state hypotheses
2. Draw one (ideally large) random sample
3. Compute measure of interest (e.g. mean, correlation coefficient, difference between condition means), and then the test statistic
4. Apply statistical distribution theory to get the **sampling distribution** of a test statistic
5. Make a decision (either reject or don't reject H_0) based on pre-specified significance level α

The magic component

"Apply statistical distribution theory to get the **sampling distribution** of a test statistic"

This dates back to pre-computer era where statisticians derived mathematically the distribution of statistical measures for an infinite amount of samples! That's are tricky thing to do and these approximations are typically tied to assumptions such as normality, homoscedasticity, independent observations, and: the sample needs to be "large" (cf. Central Limit Theorem CLT).



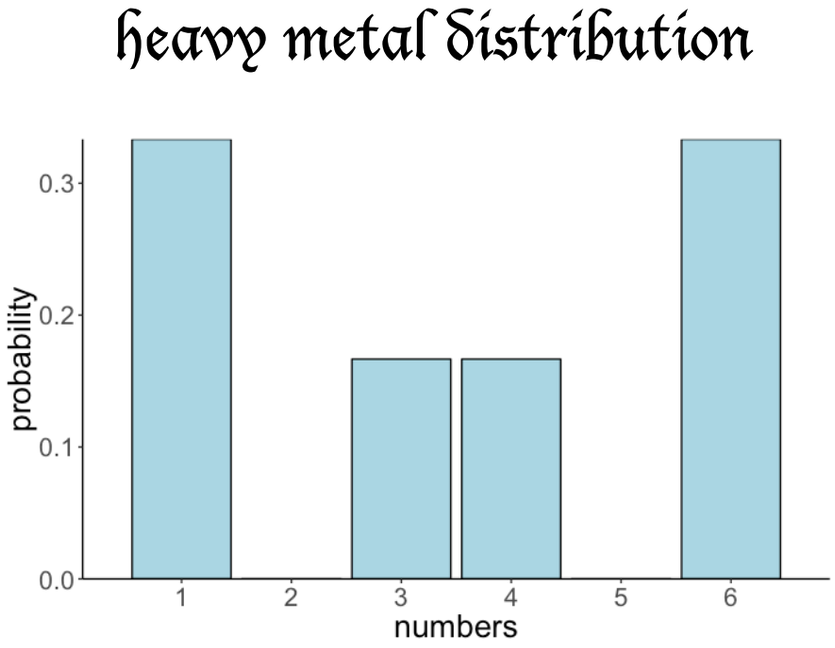
instead: simulation-based approach

Central limit theorem

sample size = 100

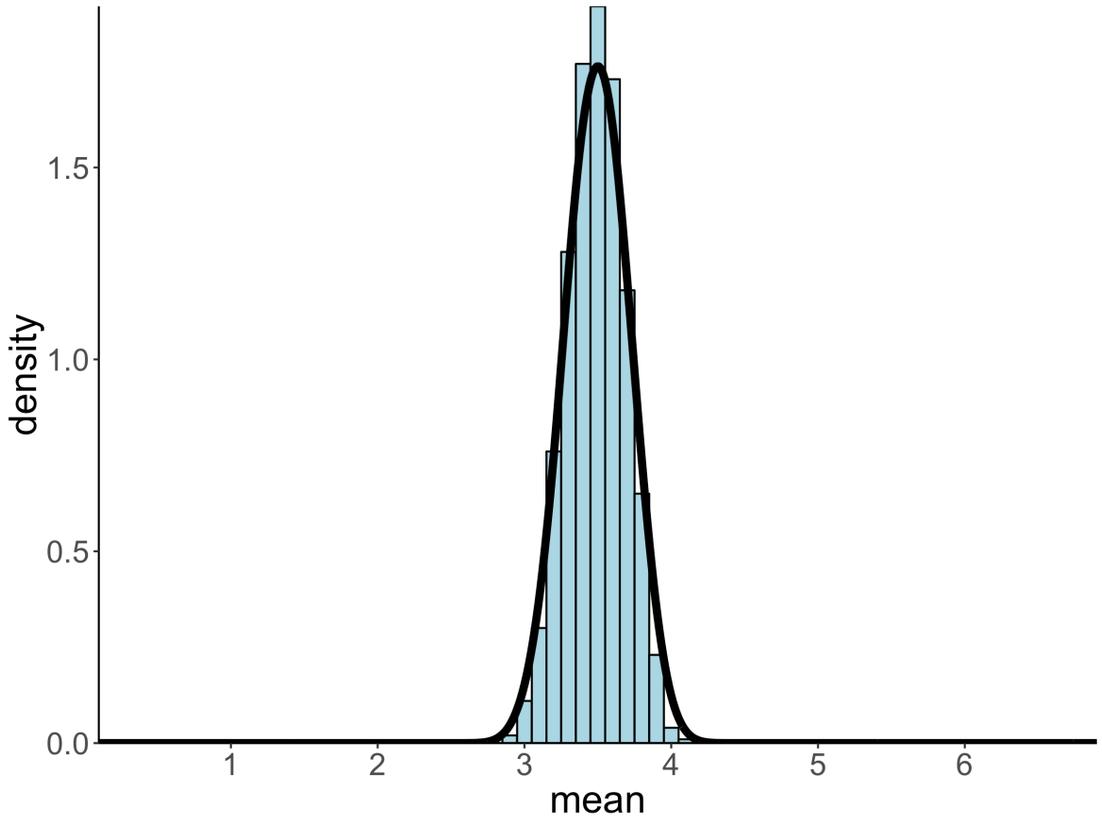
number of samples = 1000

...

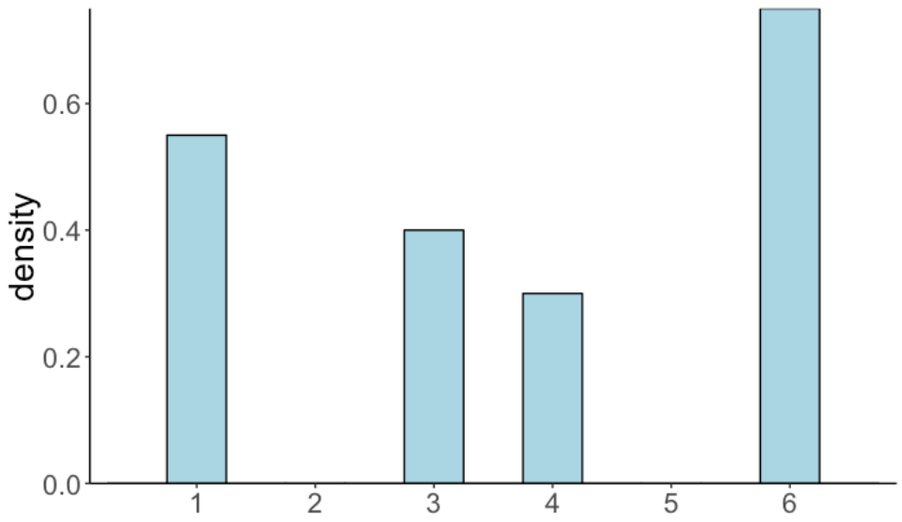


sample	draw_1	draw_2	draw_3	draw_4	sample_mean
1	1	6	6	4	4.25
2	1	4	4	6	3.75
3	6	1	1	1	2.25
4	3	6	3	6	4.50
5	3	4	6	3	4.00
6	4	1	6	1	3.00
7	1	6	1	6	3.50
8	4	6	6	6	5.50
9	6	1	3	3	3.25
10	3	1	3	6	3.25

⋮



our sample

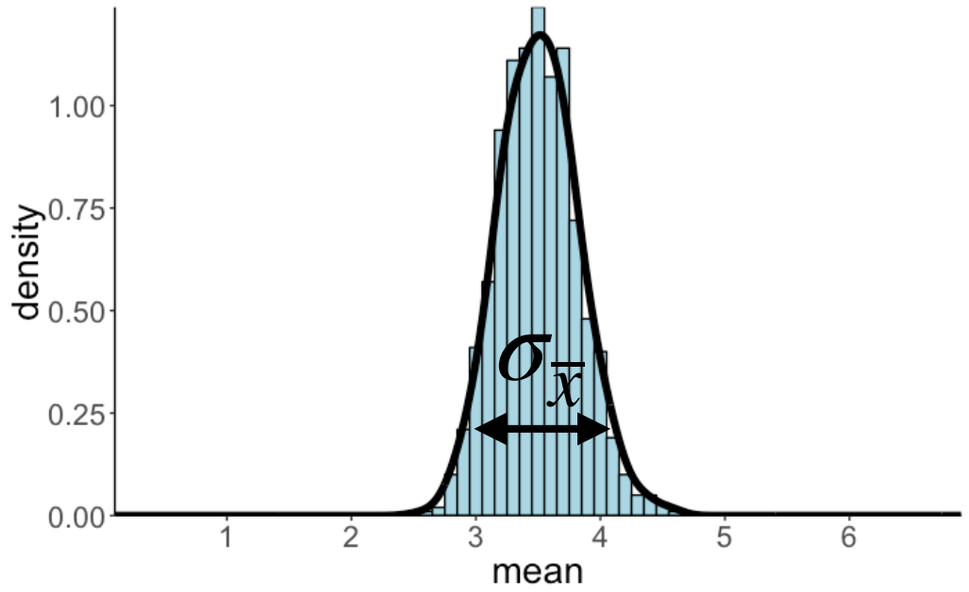


standard deviation

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N - 1}}$$

gives a sense for how well the mean captures the data

sampling distribution



standard error

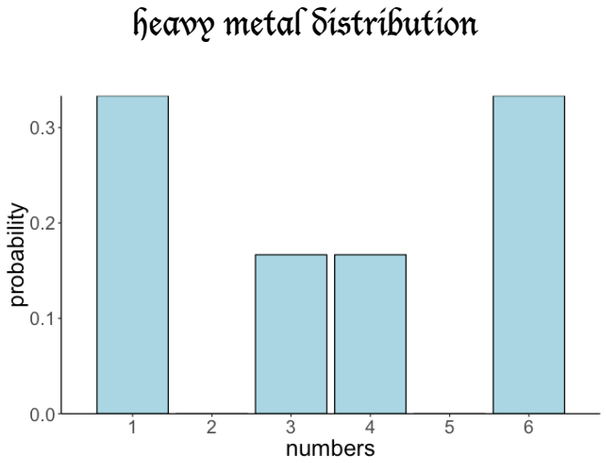
$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$$

the standard deviation of the sampling distribution

how much variation do we expect between the means of samples

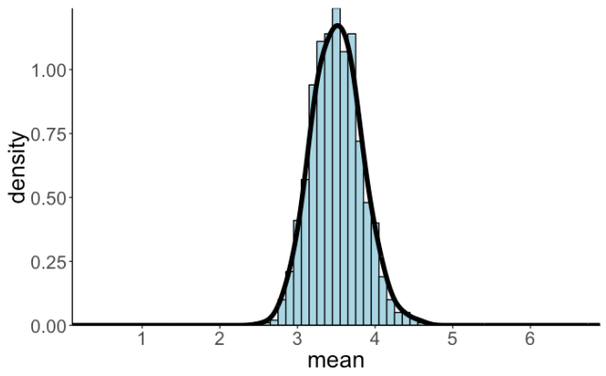
how likely is it that our sample is representative of the population

3 distributions in statistical inference



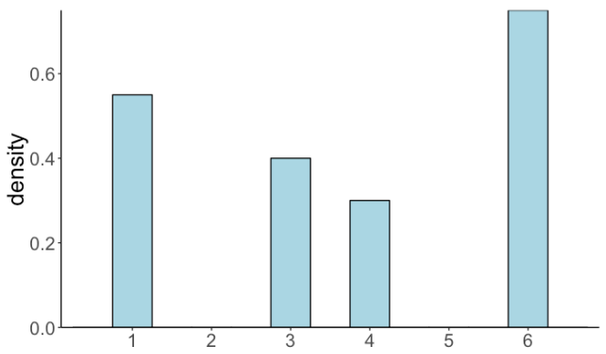
population distribution

- unknown
- our target for inference



sampling distribution

- bridge between sample and population
- derived mathematically / computationally
- asymptotic distribution theory or resampling approaches
- shows how test statistic varies between samples



sample distribution

- our observed sample
- we compute statistics of interest (mean, variance, correlation, ...)

p-values

Statistical inference

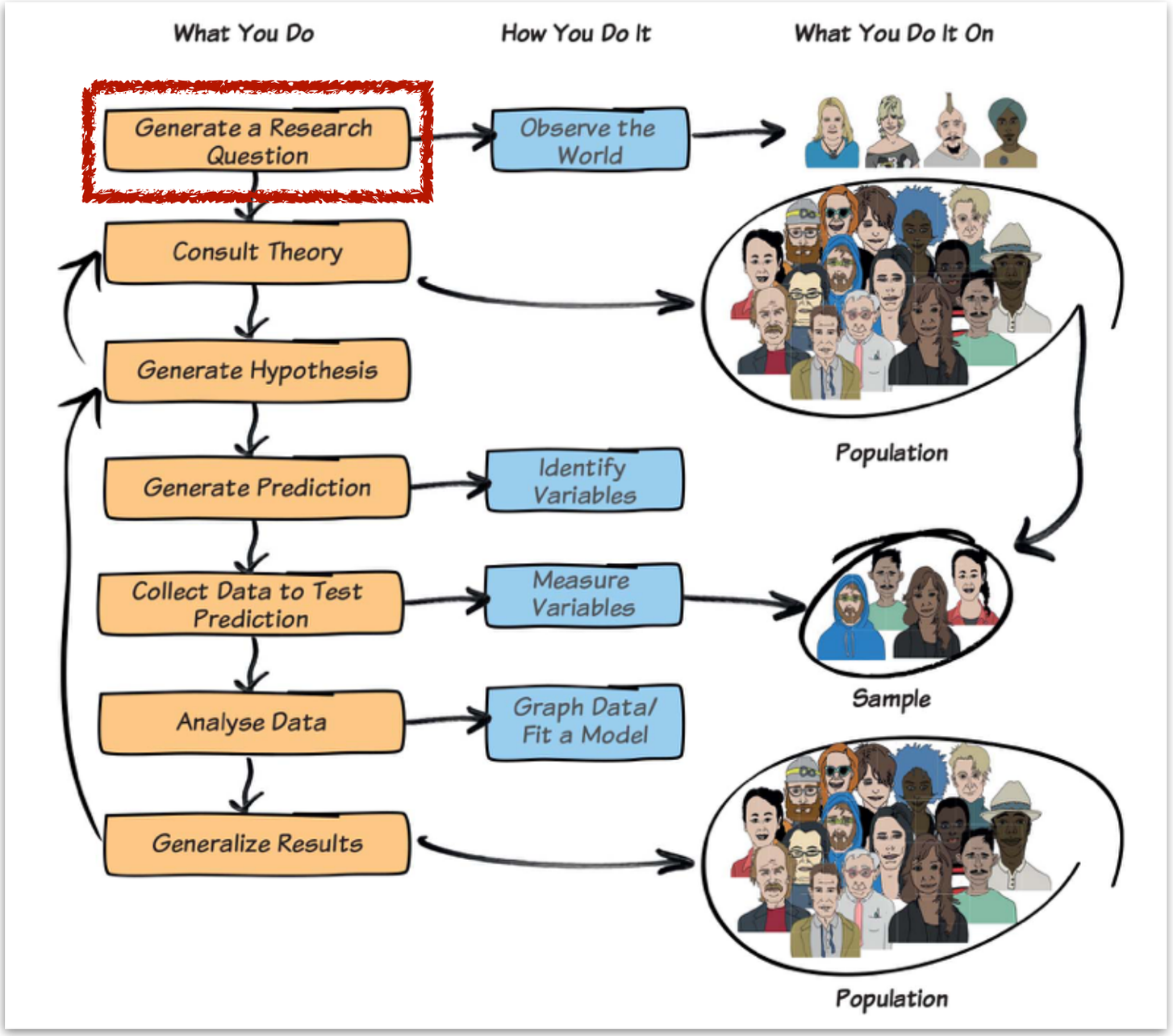
null hypothesis

$$H_0 : \mu_1 = \mu_2$$

alternative hypothesis

$$H_1 : \mu_1 < \mu_2$$

Greek letters
population parameters!



Which of the following statements about the p-value do you believe to be true?

The p-value is the probability that the null hypothesis is true.

The p-value is the probability that the alternative hypothesis is true.

The p-value is the probability of obtaining the observed or more extreme results if the alternative hypothesis is true.

The p-value is the probability of obtaining the observed results or results which are more extreme if the null hypothesis is true.

The **p-value** is the probability of finding the observed, or more extreme, results when the null hypothesis (H_0) is true.

$$p(\text{test statistic} \geq \text{observed value} \,|\, H_0 = \text{true})$$

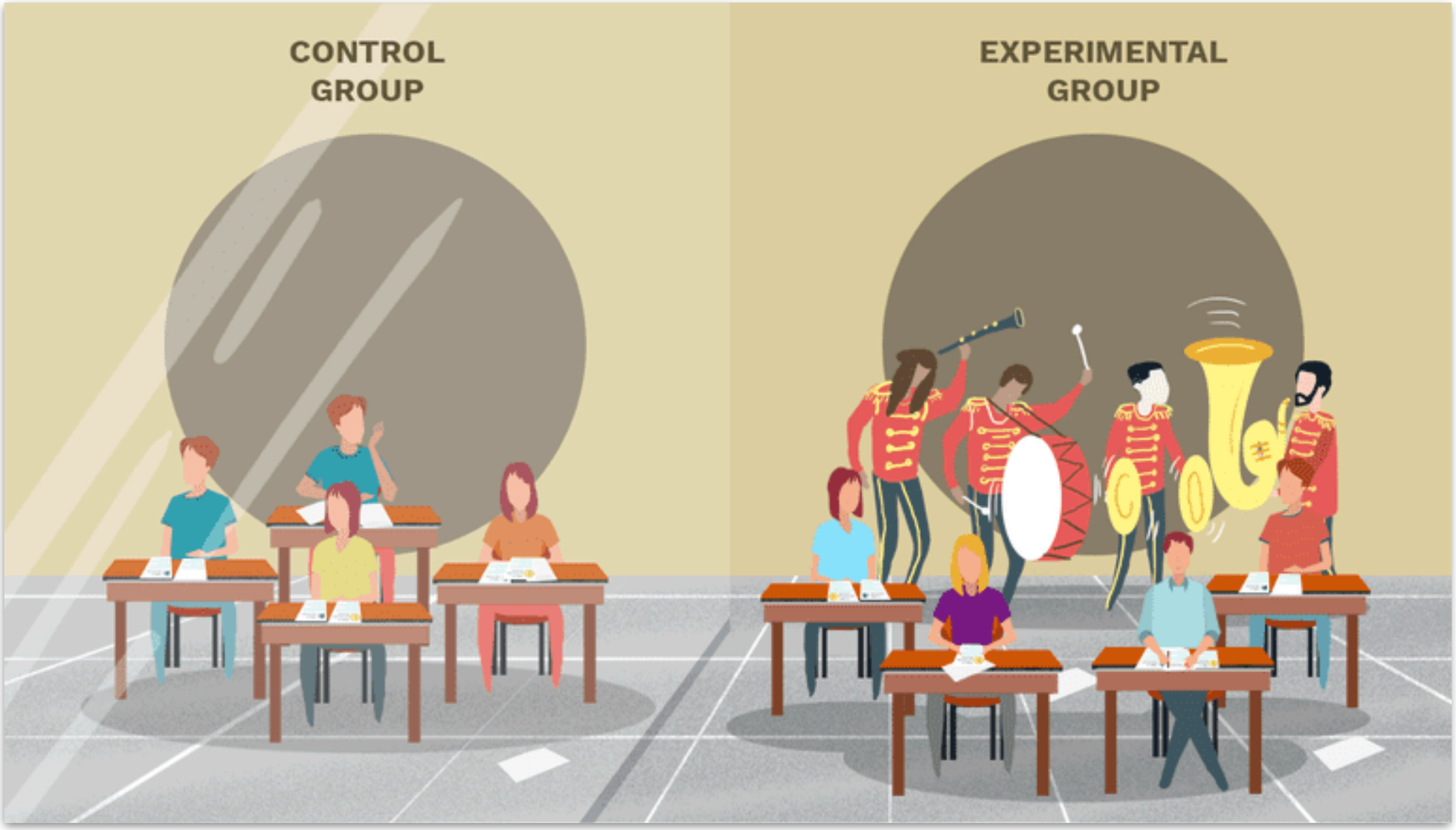
$$p(H_0 = \text{true} \,|\, \text{test statistic} \geq \text{observed value})$$

... we have to wait for Reverend Bayes

Logic of inference

- calculate a **test statistic** based on the sample
  - for example, the difference between the means in two conditions
- build a **sampling distribution** of this statistic assuming that the null hypothesis is true
  - use math or sampling
- **calculate the probability** of the observed statistic on the sampling distribution
- reject the null hypothesis if the probability of the observed statistic is less than the pre-specified α level

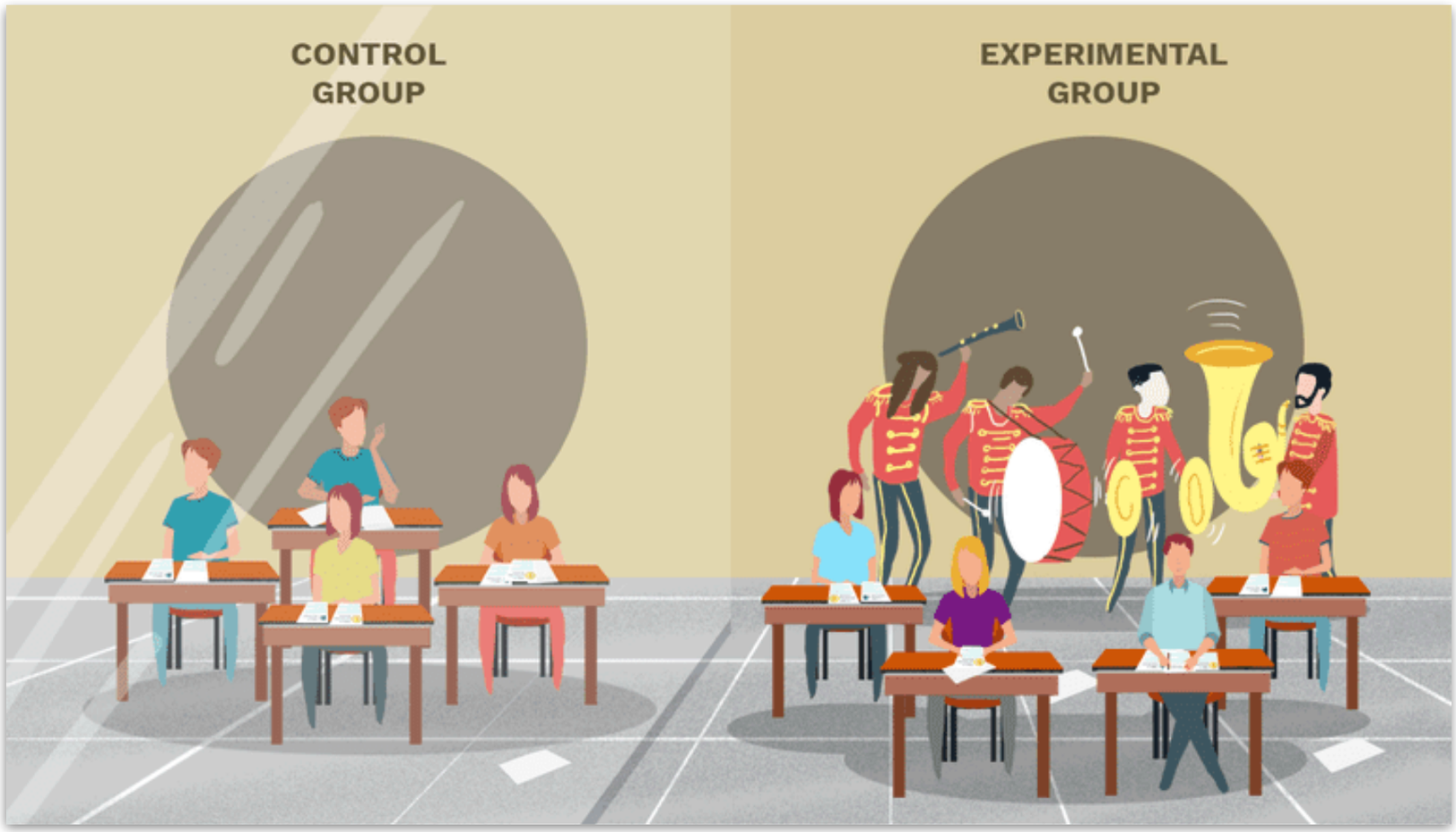
Permutation test



Research question:

Will student test scores be affected by distracting sounds (aka the Stanford band)?

Permutation test



$H_0 : \mu_{\text{control}} = \mu_{\text{experimental}}$

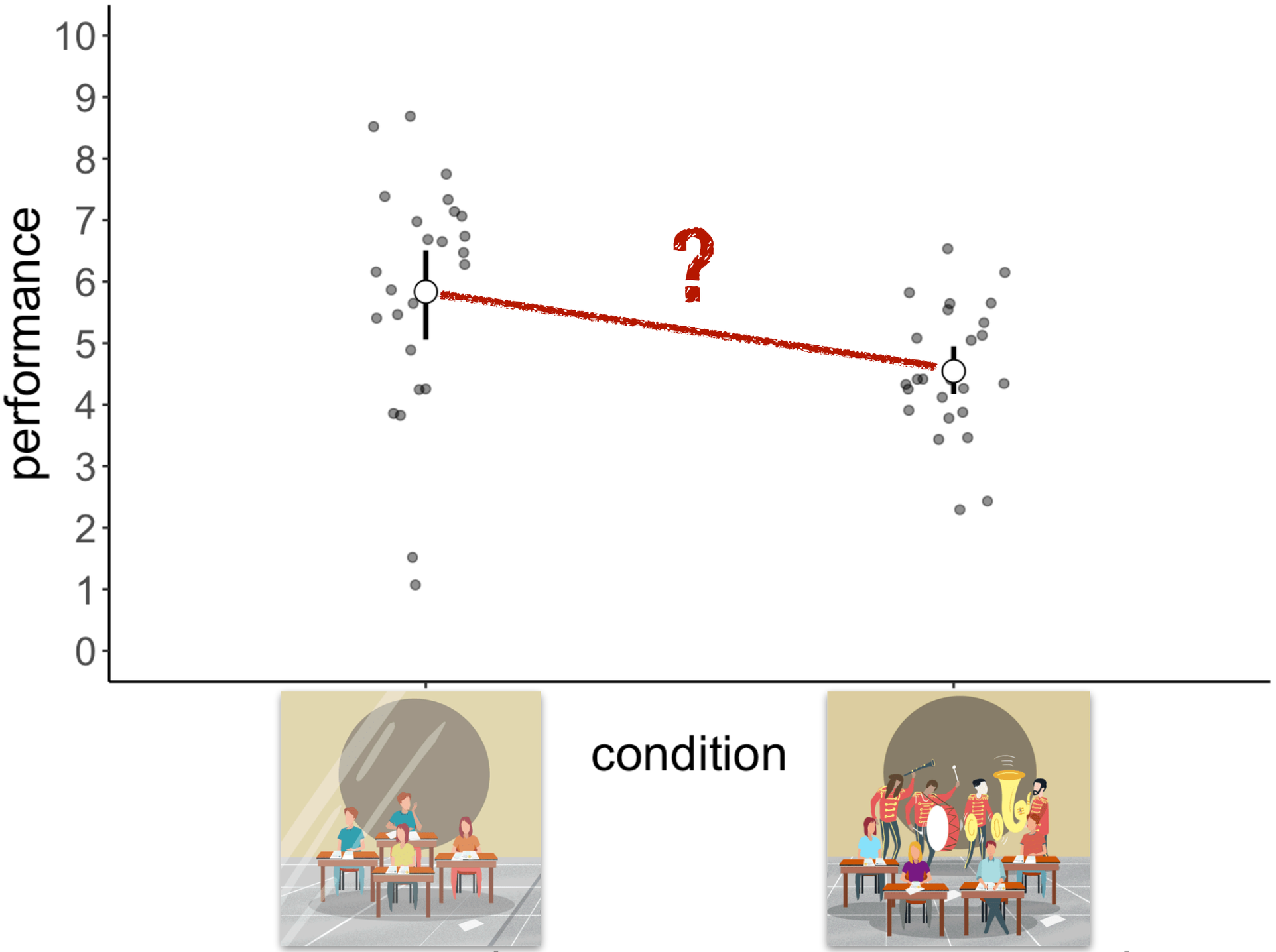
There is no difference between the control group and the experimental group

$H_1 : \mu_{\text{control}} > \mu_{\text{experimental}}$

Performance in the control group is better than in the experimental group

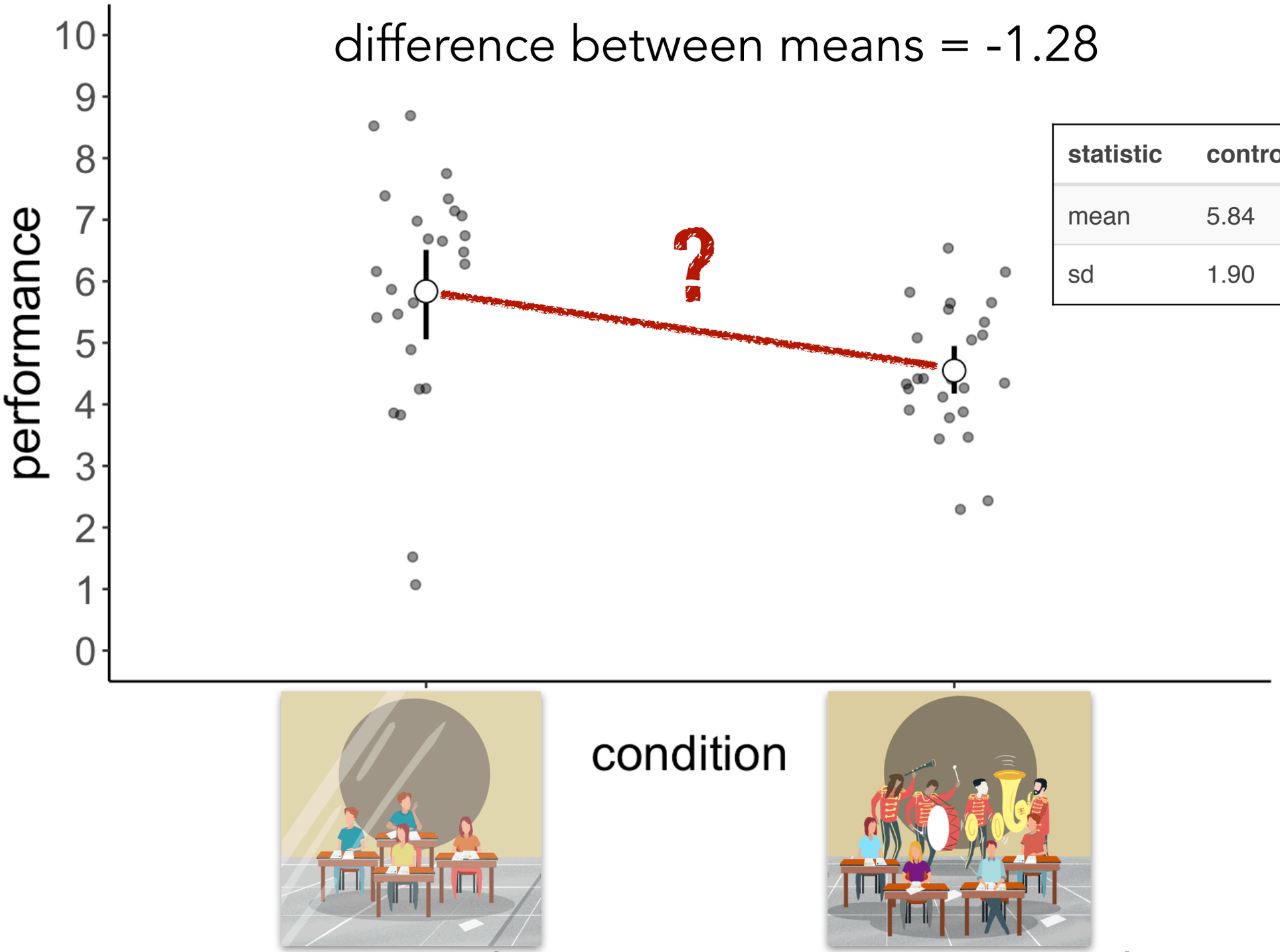
Permutation test

Is the difference in performance statistically significant?



Permutation test

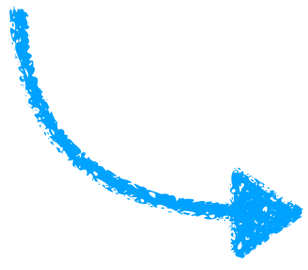
Is the difference in performance statistically significant?



statistic	control	experimental
mean	5.84	4.55
sd	1.90	1.06

Permutation test

- **logic**:
  - assuming our experimental manipulation makes no difference, what's the probability of observing the data the we did?
  - if, assuming that the null hypothesis is true, the probability of observing the data (or data that is more extreme) is less than 5%, we reject the null hypothesis



we need a sampling distribution of our test statistic (difference between means)

Permutation test

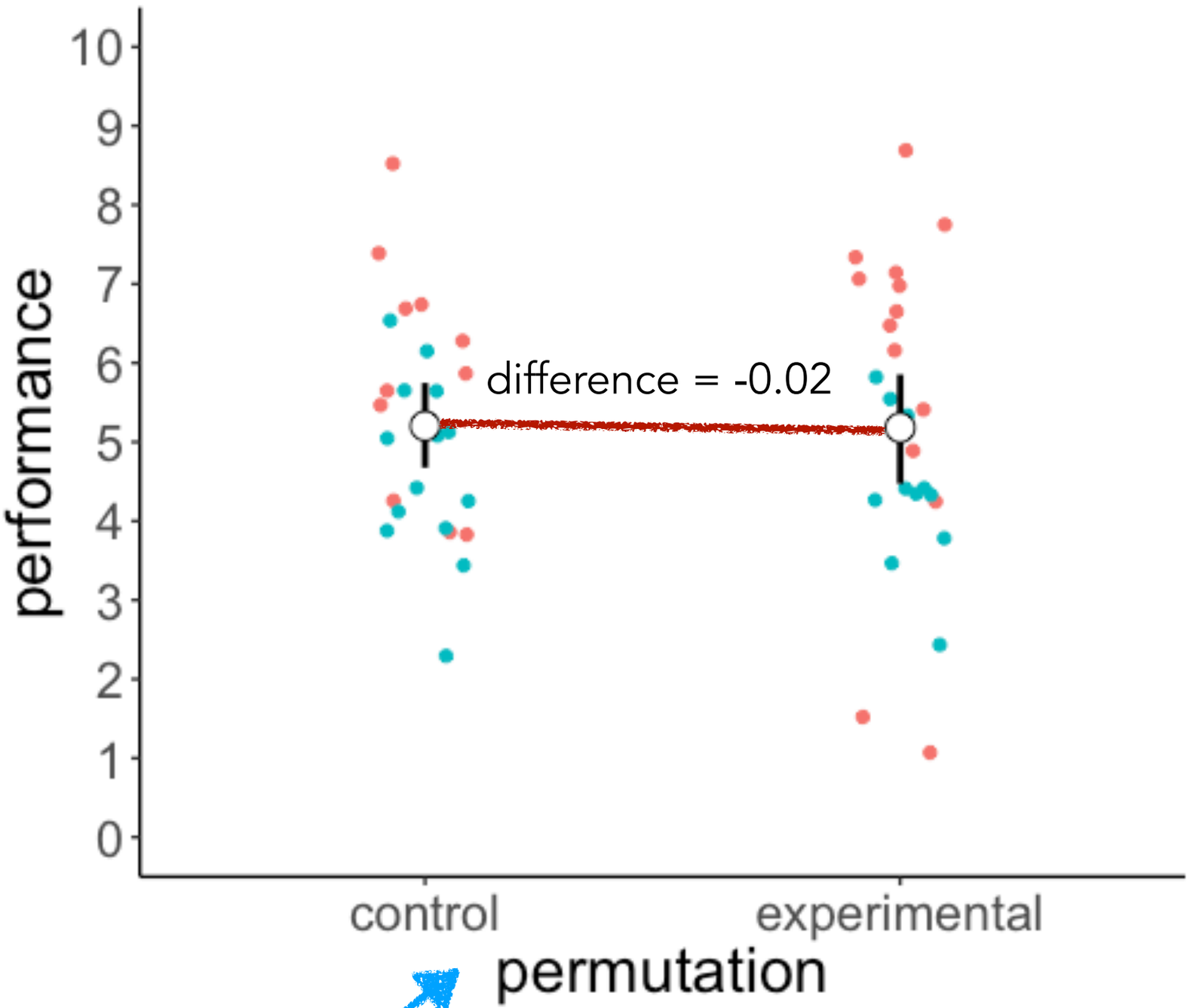
observed data

participant	condition	performance
1	control	4.25
2	control	5.87
3	control	3.83
4	control	8.69
5	control	6.16
26	experimental	4.42
27	experimental	4.27
28	experimental	2.29
29	experimental	3.78
30	experimental	5.13

random permutation

participant	condition	performance
1	control	4.25
2	experimental	5.87
3	control	3.83
4	experimental	8.69
5	control	6.16
26	control	4.42
27	experimental	4.27
28	control	2.29
29	experimental	3.78
30	experimental	5.13

Permutation test



in the actual experiment

condition
- control
- experimental

in the permutation

Permutation test

observed data

participant	condition	performance
1	control	4.25
2	control	5.87
3	control	3.83
4	control	8.69
5	control	6.16
26	experimental	4.42
27	experimental	4.27
28	experimental	2.29
29	experimental	3.78
30	experimental	5.13

1

participant	condition	performance
1	experimental	4.25
2	control	5.87
3	control	3.83
4	experimental	8.69
5	experimental	6.16
26	control	4.42
27	experimental	4.27
28	control	2.29
29	control	3.78
30	experimental	5.13

2

participant	condition	performance
1	experimental	4.25
2	control	5.87
3	experimental	3.83
4	experimental	8.69
5	experimental	6.16
26	control	4.42
27	control	4.27
28	control	2.29
29	control	3.78
30	experimental	5.13

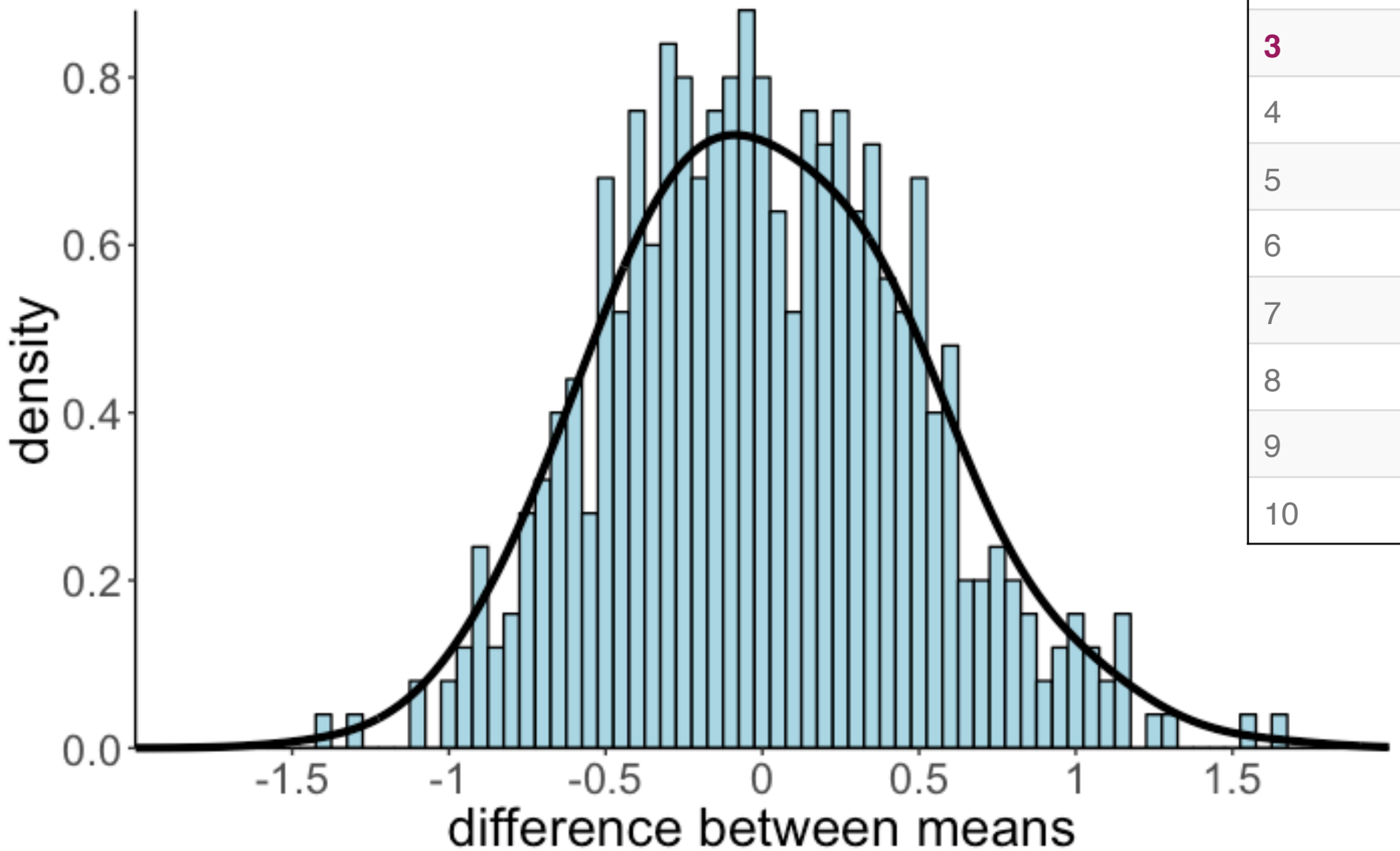
3

participant	condition	performance
1	control	4.25
2	experimental	5.87
3	control	3.83
4	experimental	8.69
5	control	6.16
26	control	4.42
27	experimental	4.27
28	control	2.29
29	experimental	3.78
30	experimental	5.13

⋮

permutation	mean_difference
1	-0.88
2	-0.26
3	-0.94
4	0.47
5	-0.28
6	1.15
7	0.98
8	0.38
9	-0.08
10	0.31

Permutation test

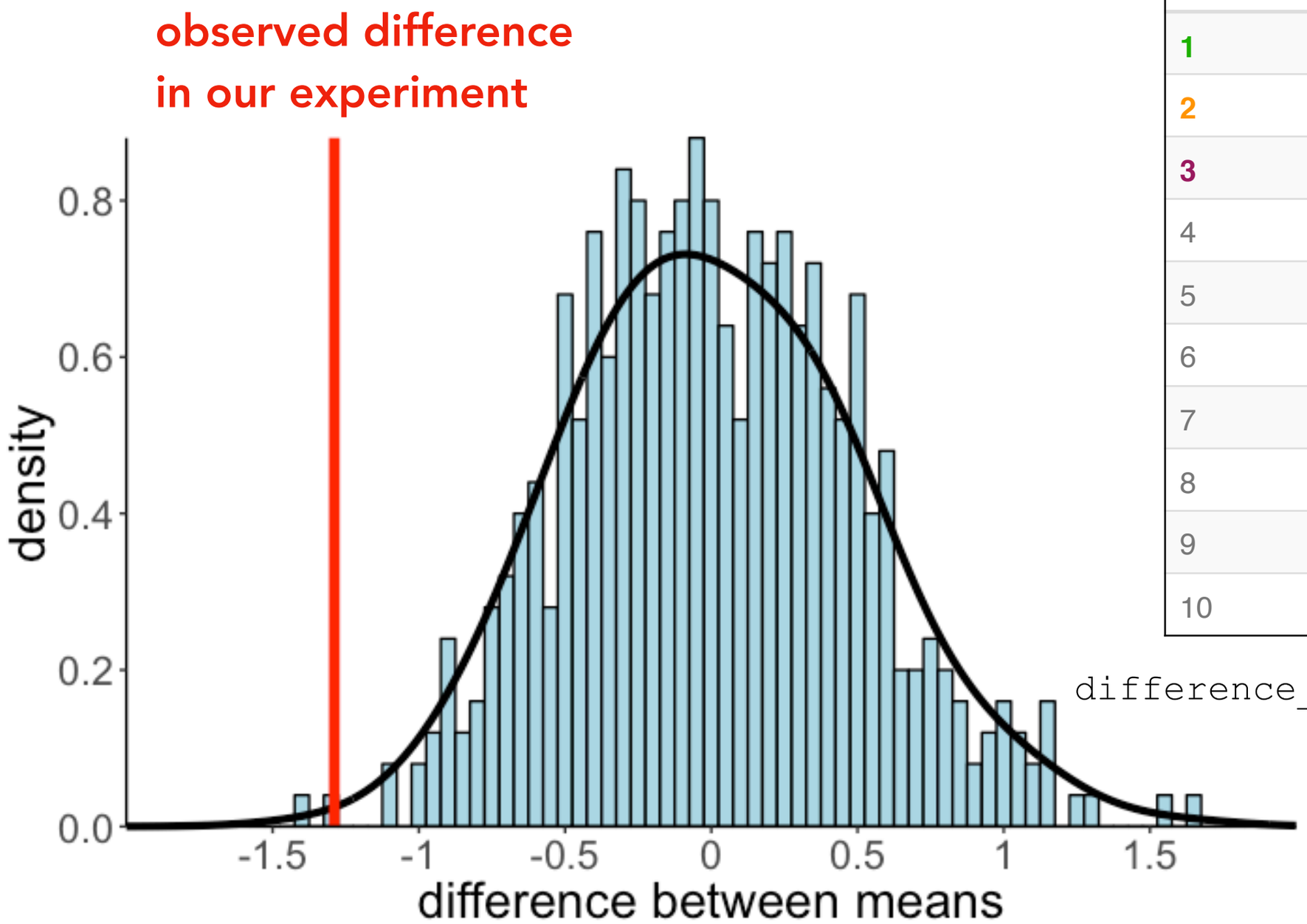


`df.permutations`

permutation	mean_difference
1	-0.88
2	-0.26
3	-0.94
4	0.47
5	-0.28
6	1.15
7	0.98
8	0.38
9	-0.08
10	0.31

Sampling distribution of differences
(expected differences if the null hypothesis is true)

Permutation test



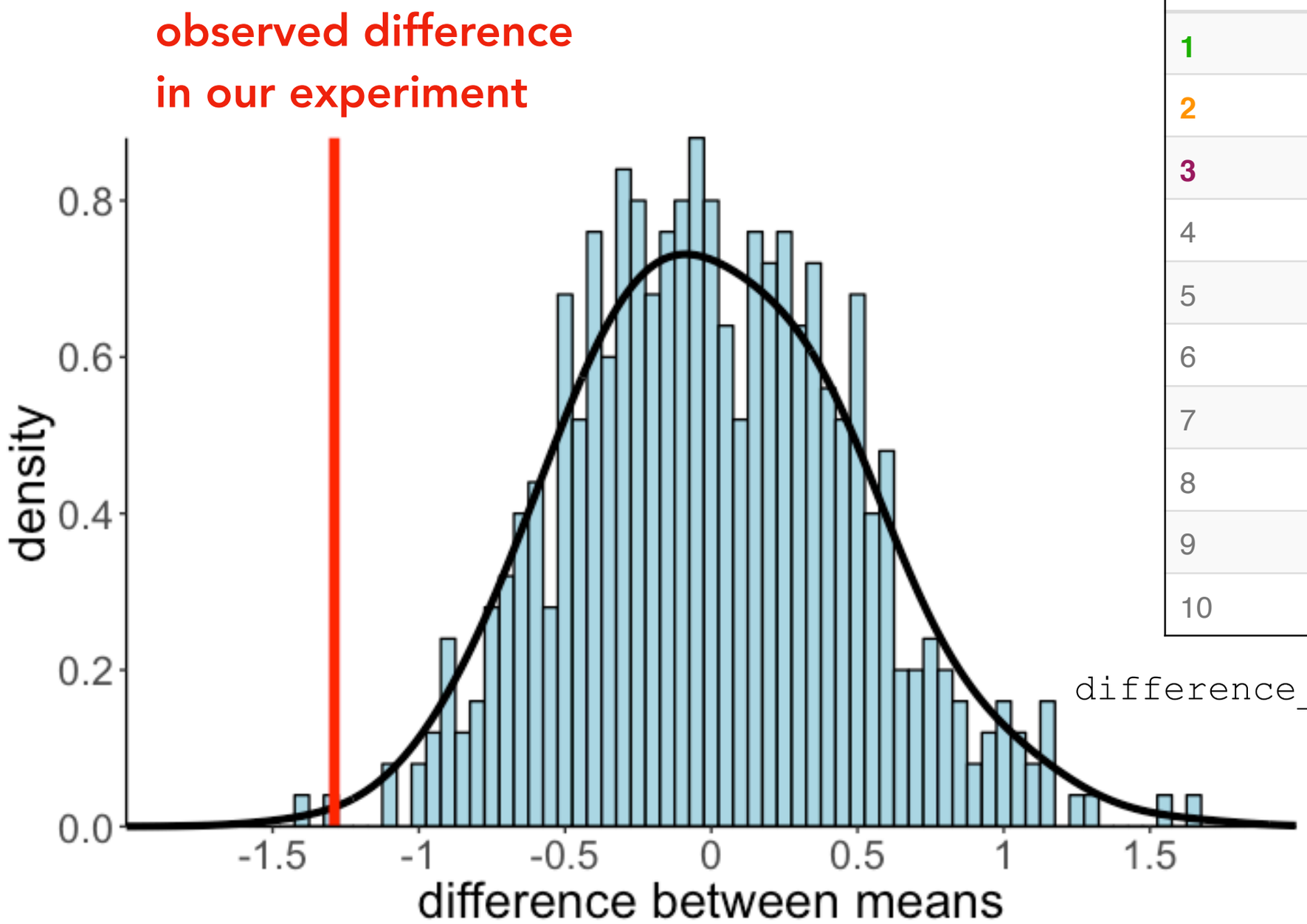
df.permutations

permutation	mean_difference
1	-0.88
2	-0.26
3	-0.94
4	0.47
5	-0.28
6	1.15
7	0.98
8	0.38
9	-0.08
10	0.31

`difference_actual = -1.28`

Sampling distribution of differences
(expected differences if the null hypothesis is true)

Permutation test



`df.permutations`

permutation	mean_difference
1	-0.88
2	-0.26
3	-0.94
4	0.47
5	-0.28
6	1.15
7	0.98
8	0.38
9	-0.08
10	0.31

`difference_actual = -1.28`

```
1 #calculate p-value of our observed result
2 df.permutations %>%
3   summarize(p_value = sum(mean_difference <= difference_actual)/n())
```

p-value = .002

Permutation test

```r
n_permutations = 500

# permutation function
func_permutations = function(df){
  df %>%
    mutate(condition = sample(condition)) %>%
    group_by(condition) %>%
    summarize(mean = mean(performance)) %>%
    pull(mean) %>%
    diff()
}

# data frame with permutation results
df.permutations = data_frame(
  permutation = 1:n_permutations,
  mean_difference = replicate(n = n_permutations, func_permutations(df.data))
)
```

set the number of permutations

shuffle the condition labels

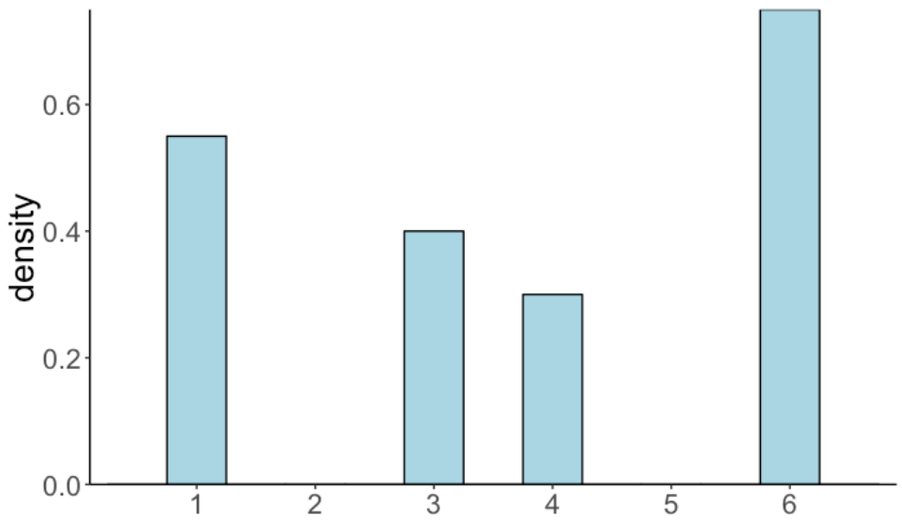
calculate difference between group means

run the `func_permutations()` function many times

(instead of using a for loop)

Confidence intervals

our sample

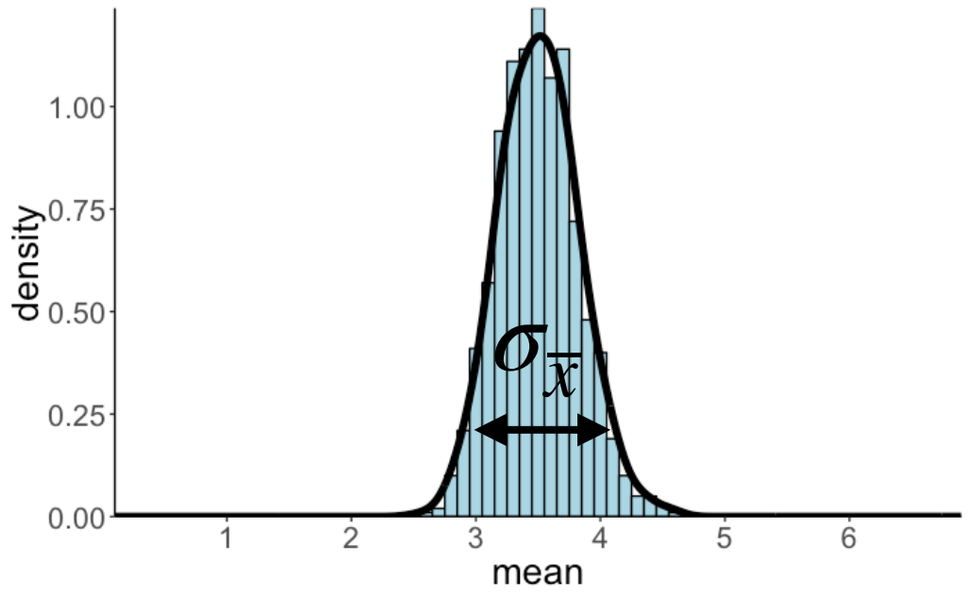


standard deviation

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N - 1}}$$

gives a sense for how well the mean captures the data

sampling distribution



standard error

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$$

$\sigma_{\bar{x}}$ decreases as:

σ decreases

N increases

$$\hat{\sigma} = s$$

for large enough samples (> 30)

we are more confident in our inference with larger samples, and less variance

Confidence interval

Goal: Estimate the mean of the population distribution μ

what we need:

- sample mean
- standard deviation
- sample size
- desired level of confidence (often 95%)

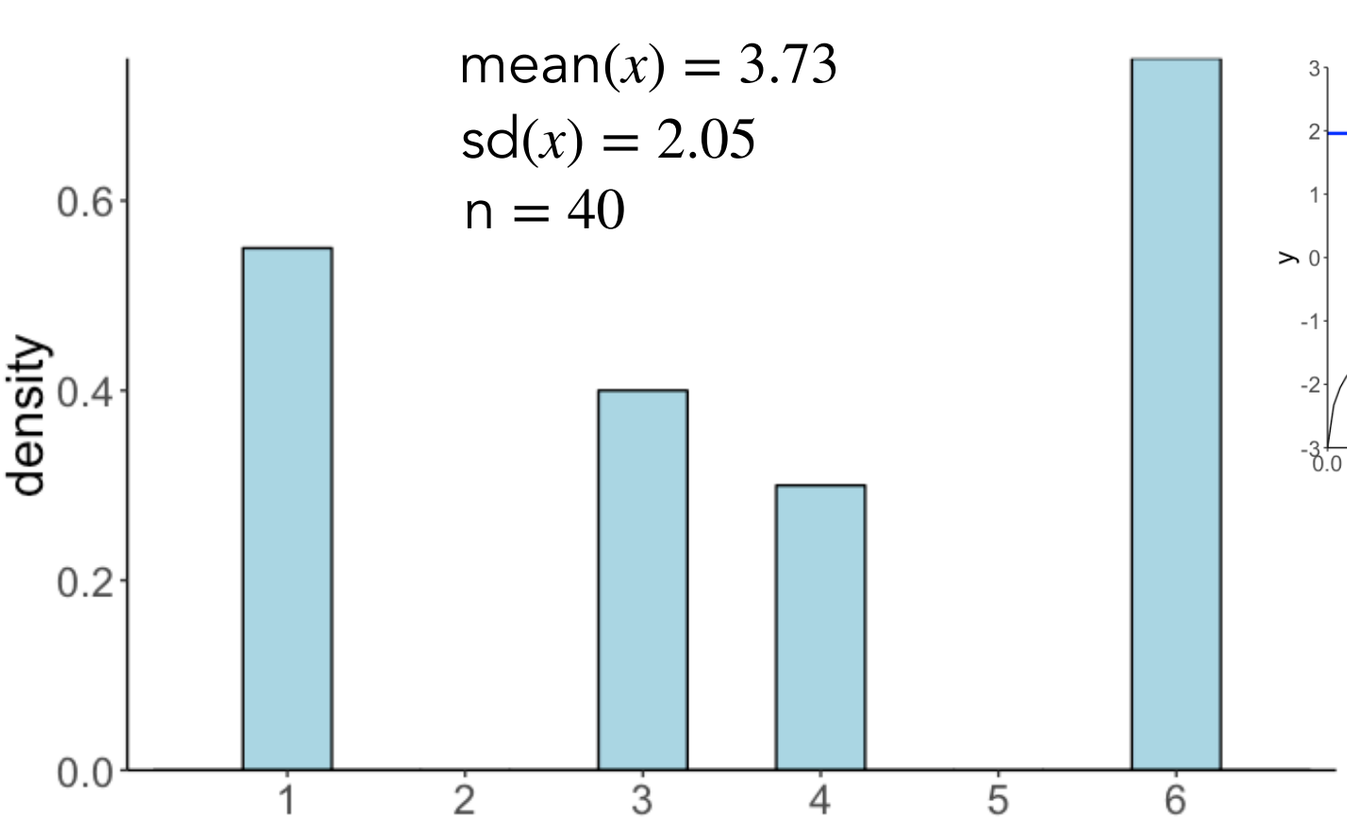
Confidence interval

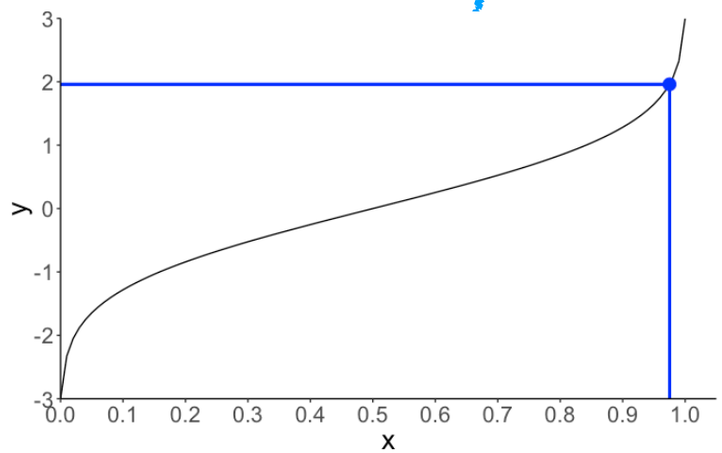
Goal: Estimate the mean of the population distribution μ

Confidence interval = point estimate ± critical value

`mean(x)`

`qnorm(.975)`×`SE`





$$\sigma_{\bar{x}} = \frac{s}{\sqrt{N}}$$

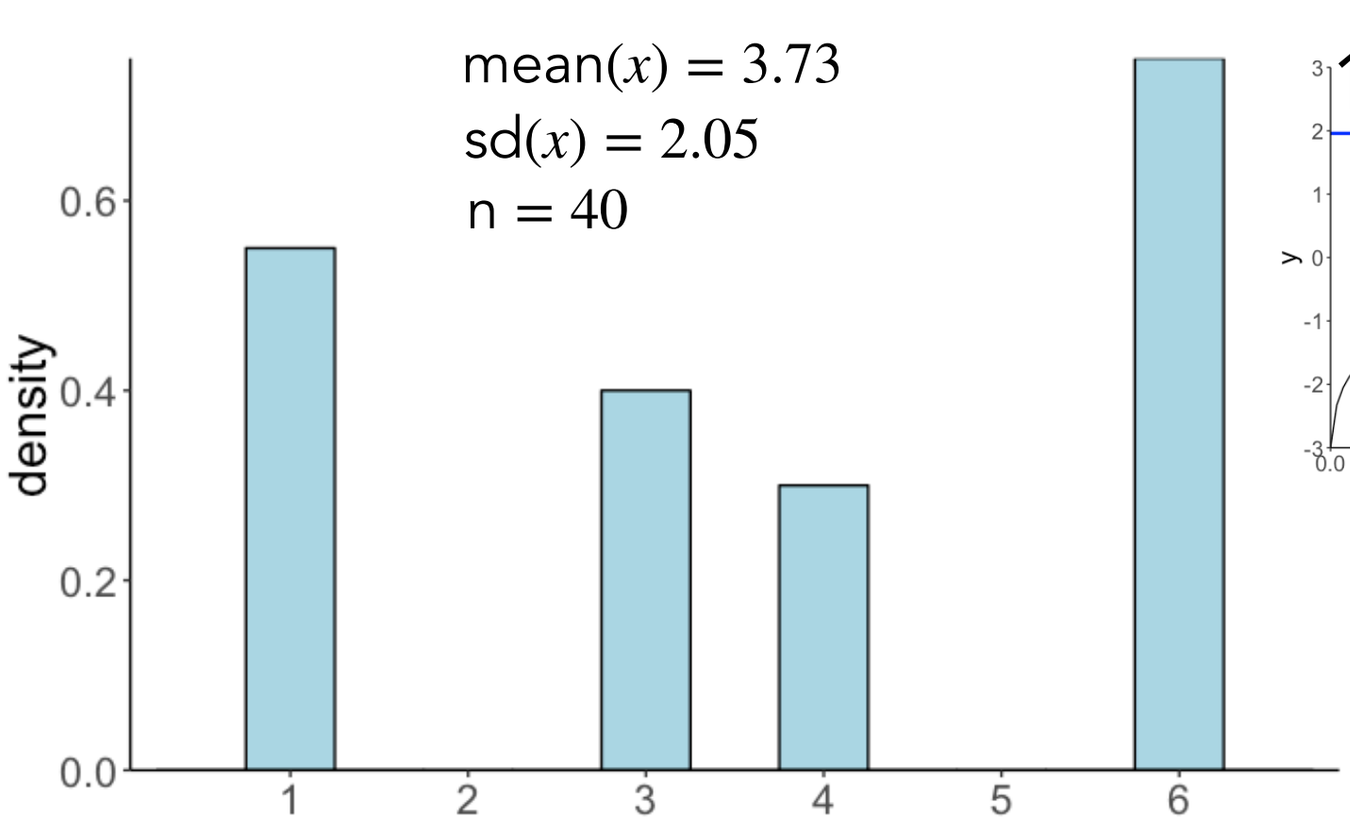
Confidence interval

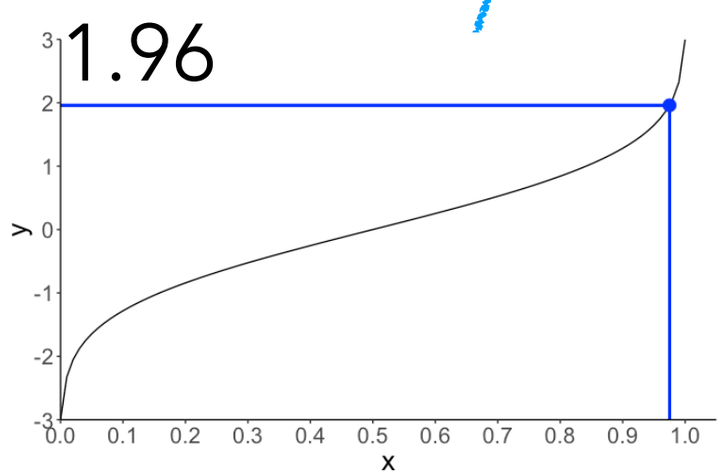
Goal: Estimate the mean of the population distribution μ

Confidence interval $= 3.73 \pm 0.63$

Confidence interval = point estimate ± critical value

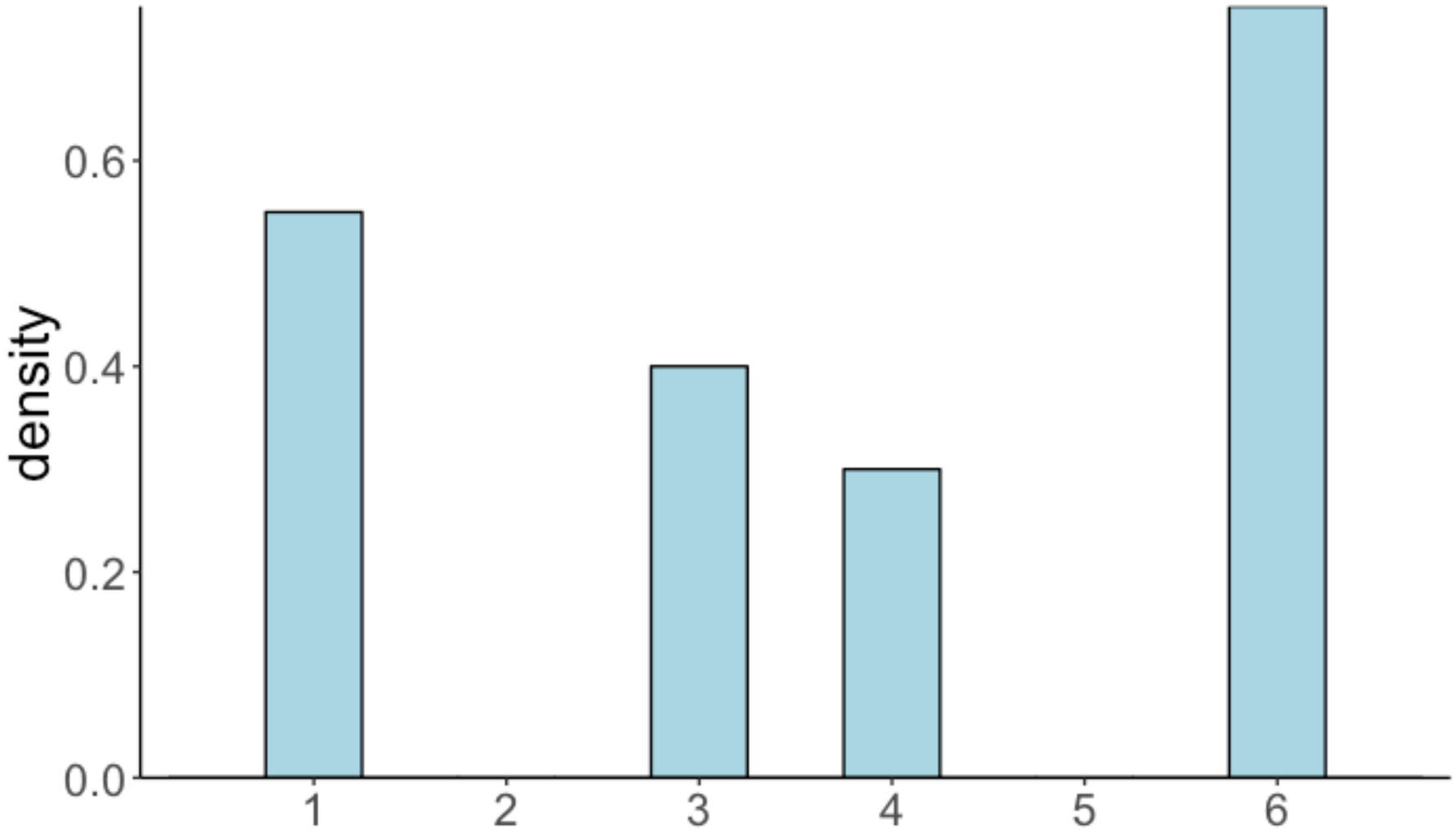
`mean(x)` `qnorm(.975)×SE`





$$\sigma_{\bar{x}} = \frac{2.05}{\sqrt{40}}$$

What does the confidence interval mean?



Mean = 3.73 ± 0.63 (95% CI)

What can we say based on the result of our sample (N = 40): Mean = 3.73 ± 0.63 (95% CI)?

95% of the time, the true population mean will be in this interval.

95% of random samples of size 40 will yield confidence intervals that contain the population mean.

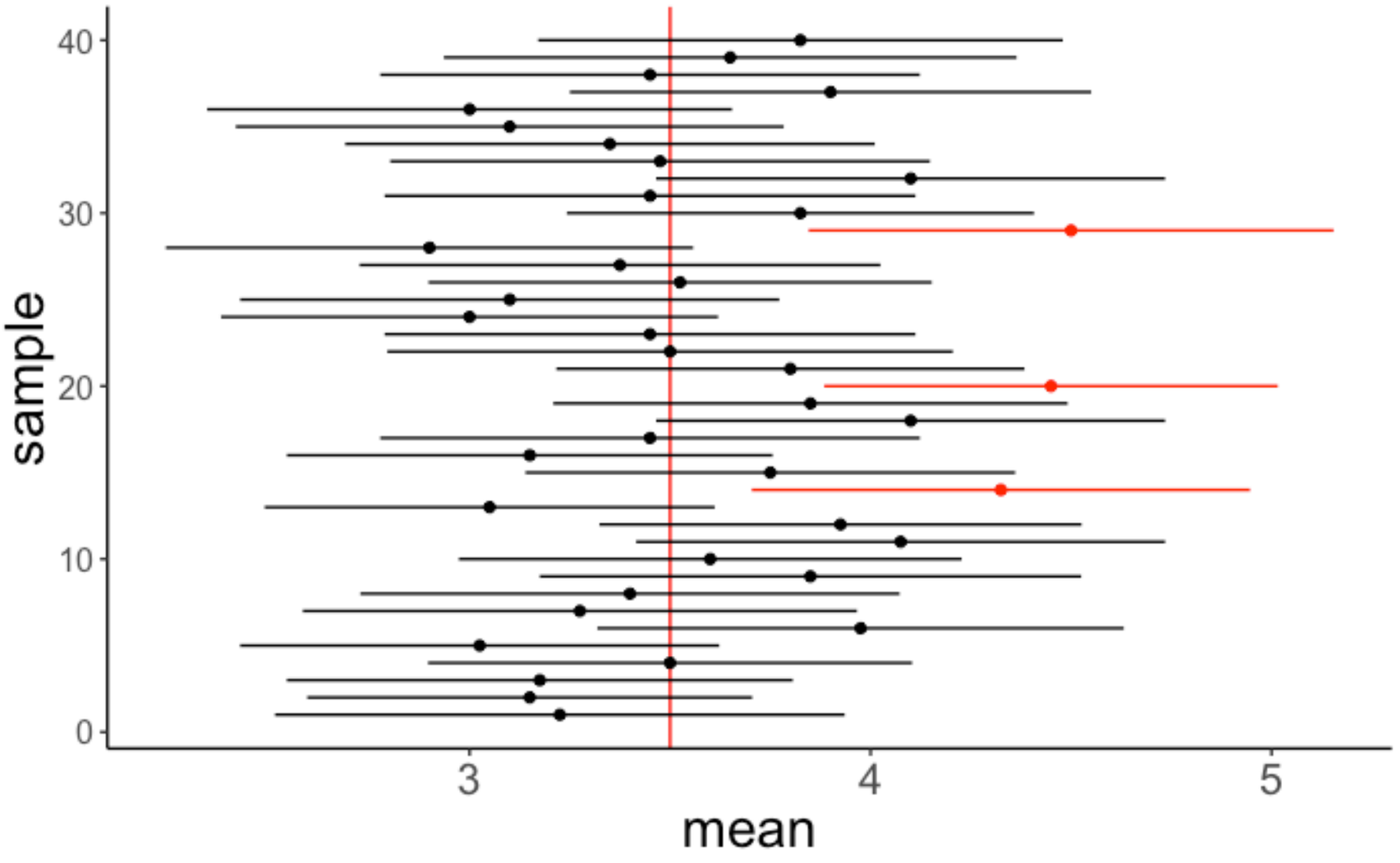
The sample means of 95% of the random samples of size 40 will be in this interval.

We can be 95% confident that the sample mean is in this interval.

Confidence interval

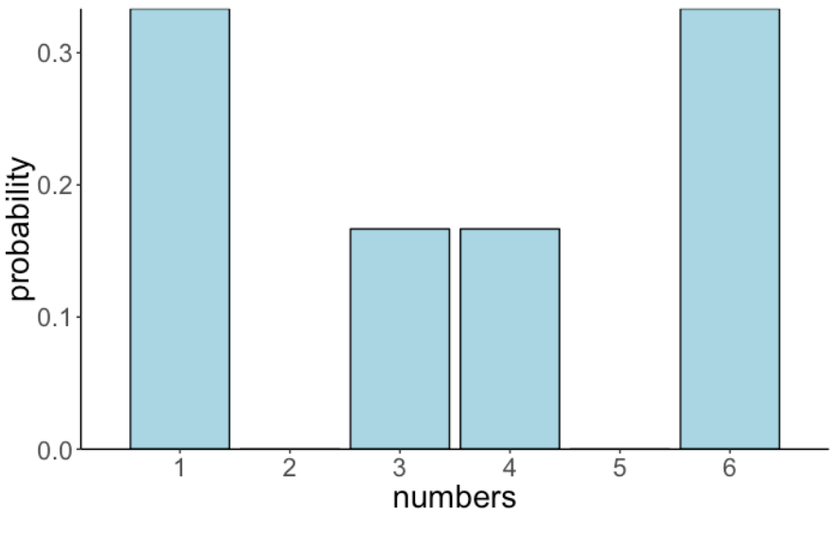
Definition

"If we were to repeat the experiment over and over, then 95 % of the time the confidence intervals contain the true mean."



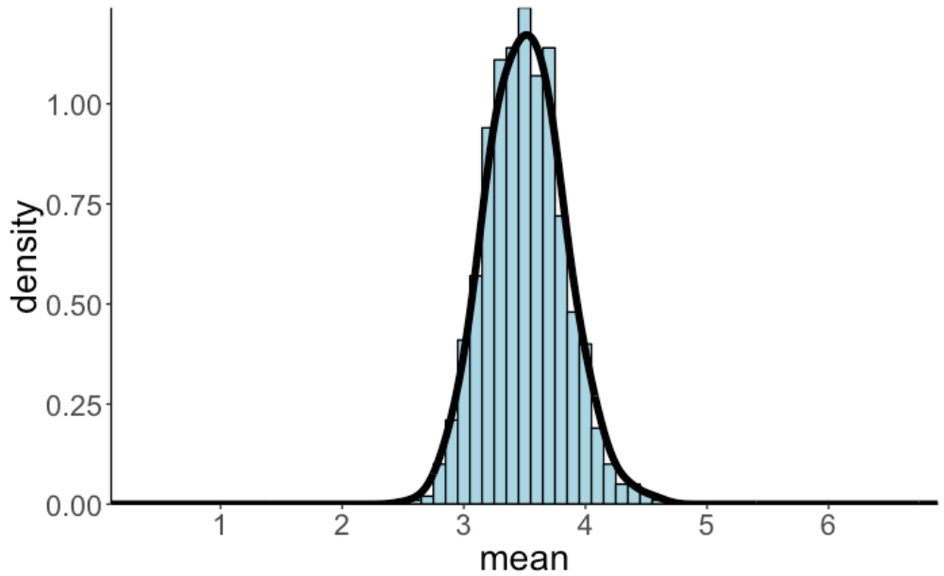
Hoekstra, R., Morey, R. D., Rouder, J. N., & Wagenmakers, E.-J. (2014). Robust Misinterpretation of Confidence Intervals. *Psychonomic Bulletin & Review*, 21(5), 1157-1164.

Bootstrap



population distribution

repeated sampling



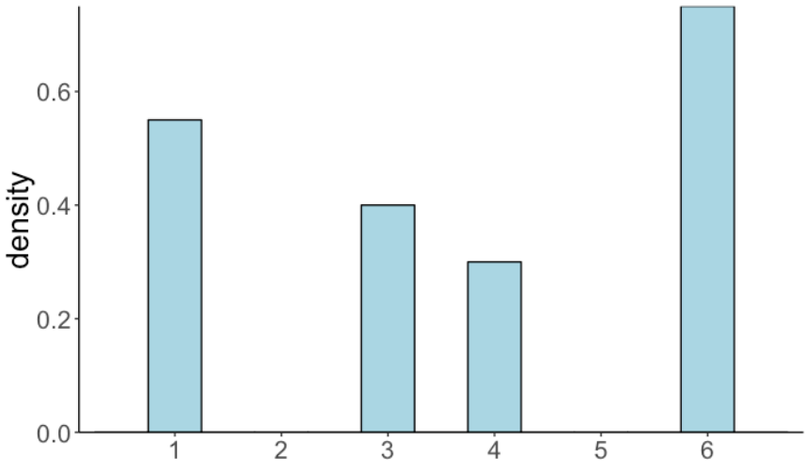
sampling distribution

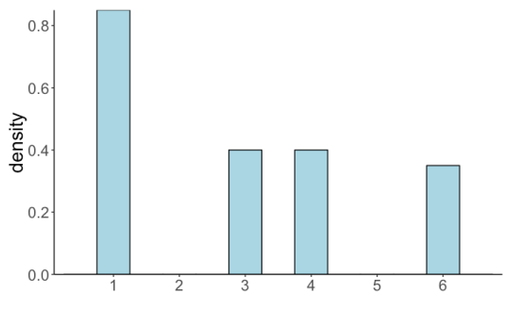
but we don't know the population distribution!

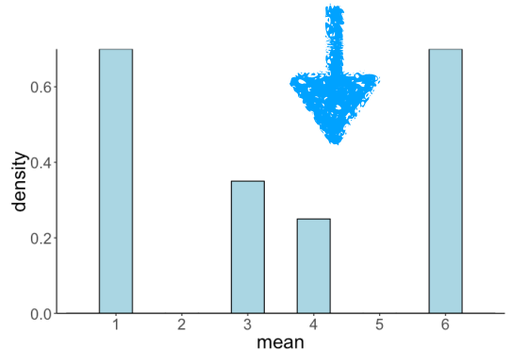
Bootstrap

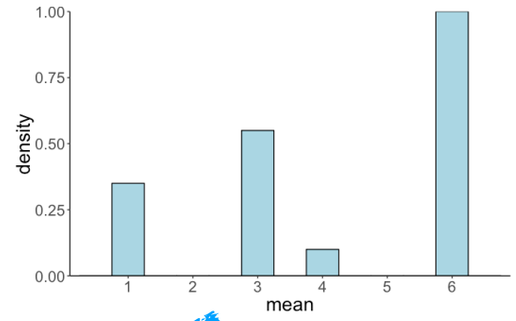
all we have is our sample

repeated sampling with replacement

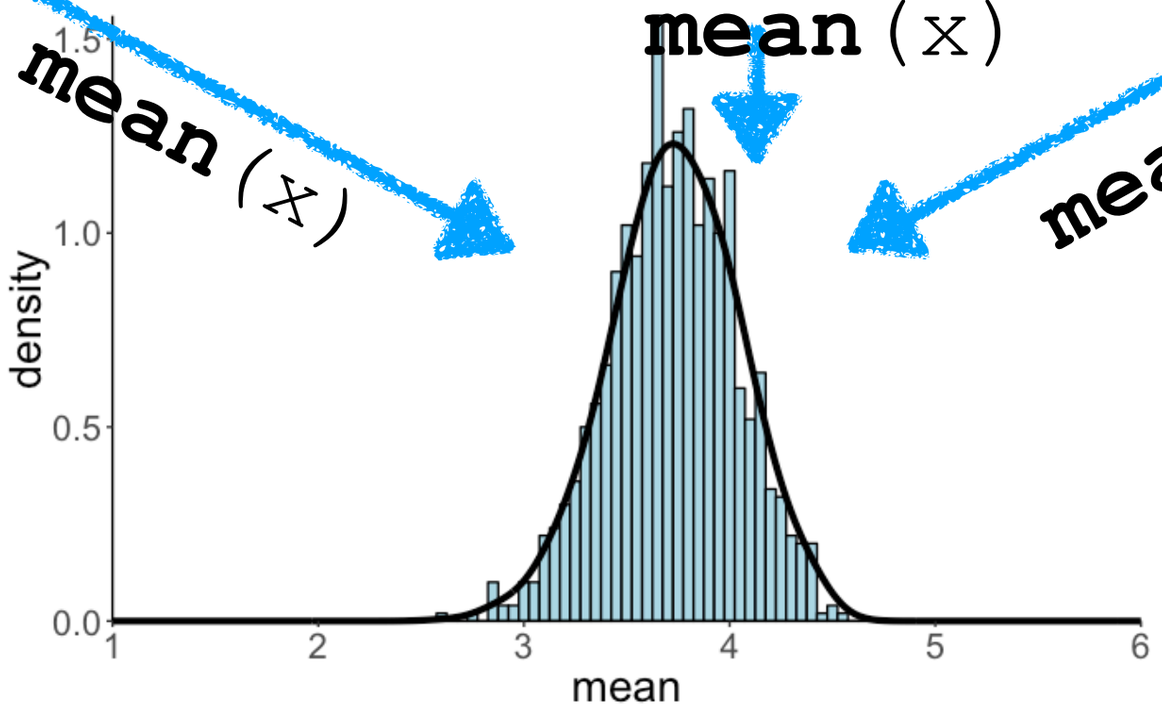








mean(x) **mean(x)** **mean(x)**



sampling distribution

Outline

- **Statistical inference**
  - drawing inferences about the population from our sample
- **Central limit theorem**
  - normal distribution of the mean (under certain conditions)
- **Sampling distributions**
  - the bridge between sample and population
  - theoretical distribution
  - simulate via permutation (or bootstrap)
- **p-values**
  - used for hypothesis testing
- **Confidence intervals**
  - parameter estimation

Datacamp

INTERACTIVE COURSE

Foundations of Inference

Continue Course

4 hours | 17 Videos | 58 Exercises | 12,551 Participants | 4,350 XP

Course Description

One of the foundational aspects of statistical analysis is inference, or the process of drawing conclusions about a larger population from a sample of data. Although counter intuitive, the standard practice is to attempt to disprove a research claim that is not of interest. For example, to show that one medical treatment is better than another, we can assume that the two treatments lead to equal survival rates only to then be disproved by the data. Additionally, we introduce the idea of a p-value, or the degree of disagreement between the data and the hypothesis. We also dive into confidence intervals, which measure the magnitude of the effect of interest (e.g. how much better one treatment is than another).

This course is part of these tracks:

Intro to Statistics with R

Jo Hardin

Professor at Pomona College

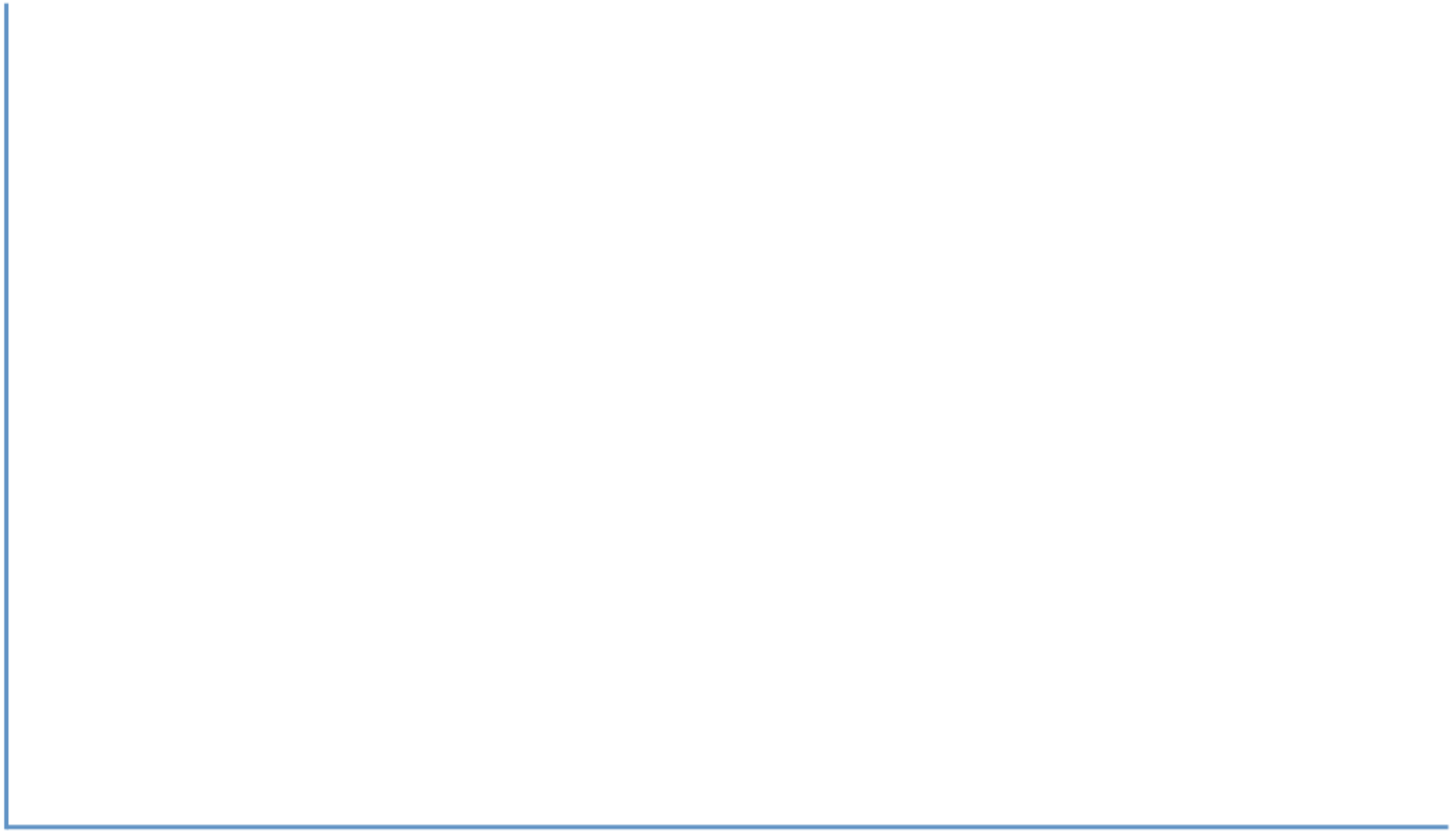
1 Introduction to ideas of inference FREE

100%

In this chapter, you will investigate how repeated samples taken from a population can vary. It is the variability in samples that allow you to make claims about the population of interest. It is important to remember that the

Feedback

How was the pace of today's class?



How happy were you with today's class overall?

What did you like about today's class? What could be improved next time?

Thank you!