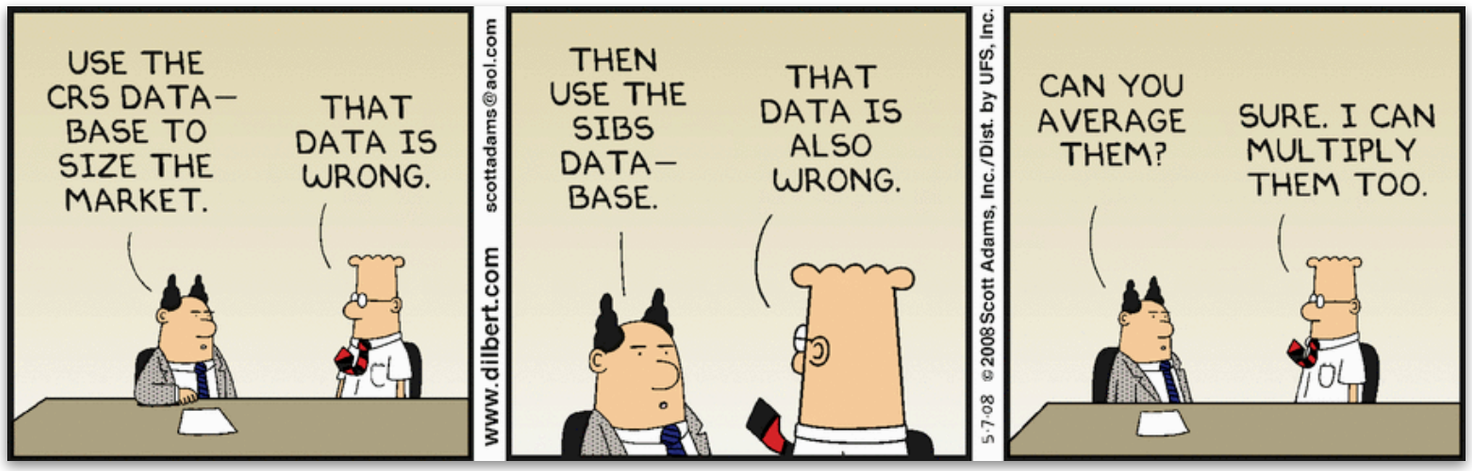
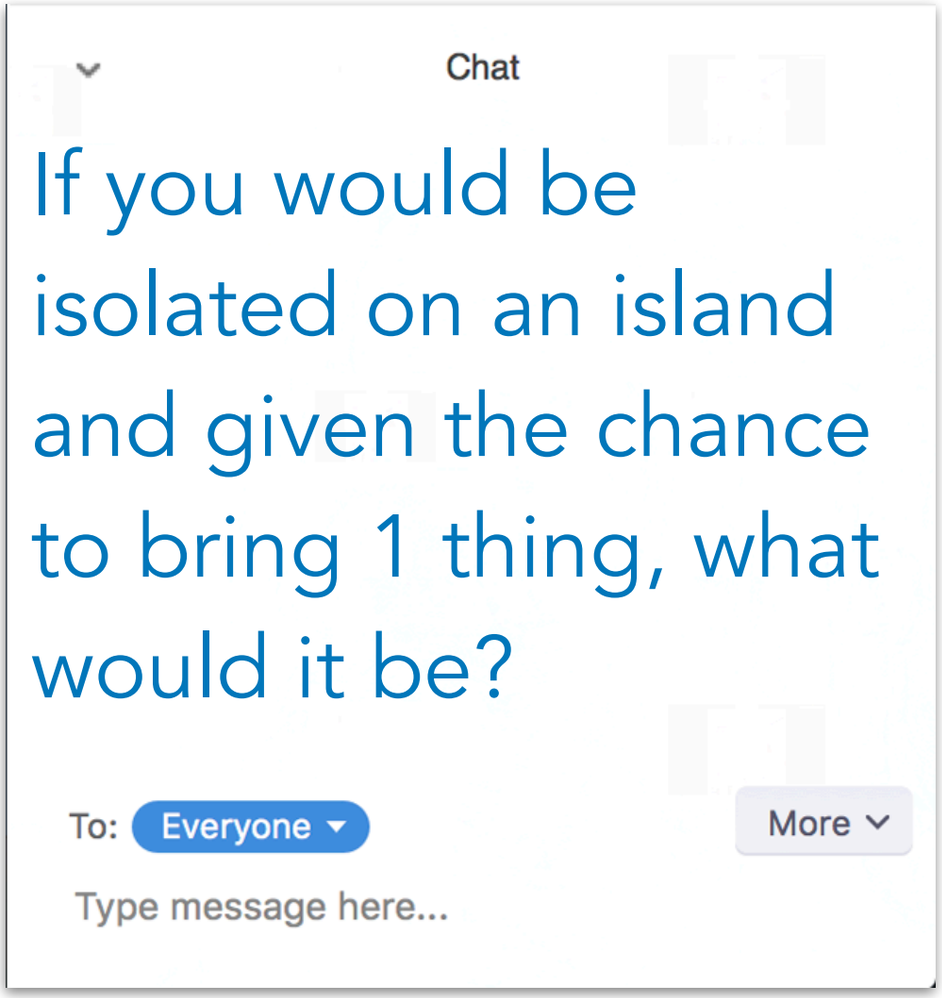
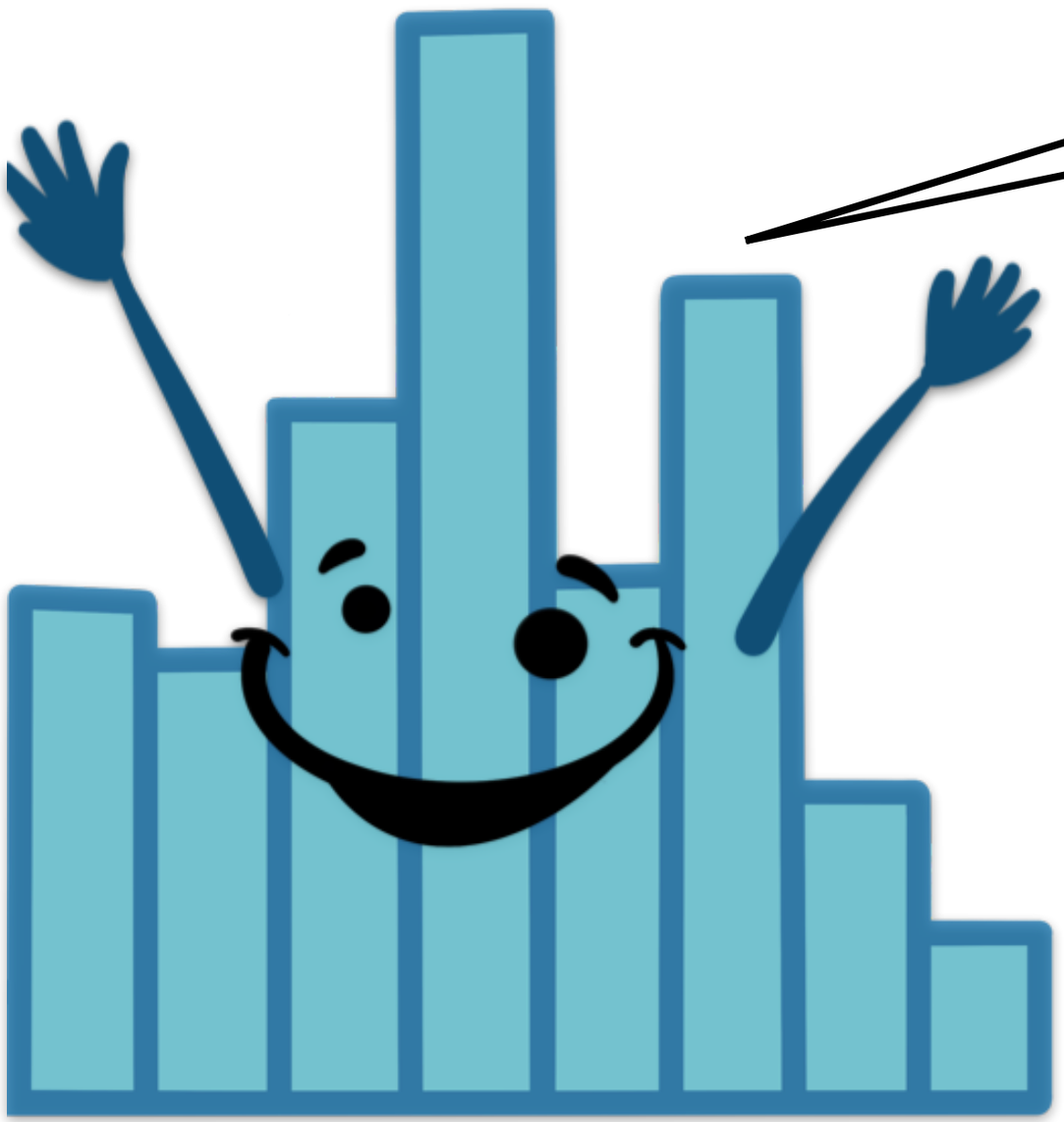
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Simulation 2





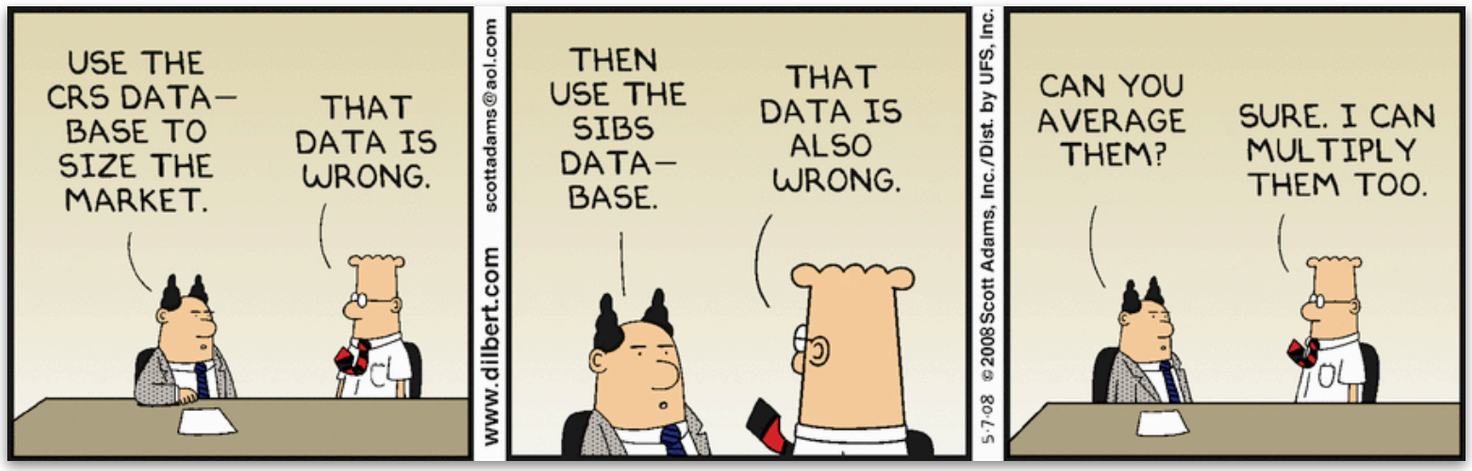
01/29/2021

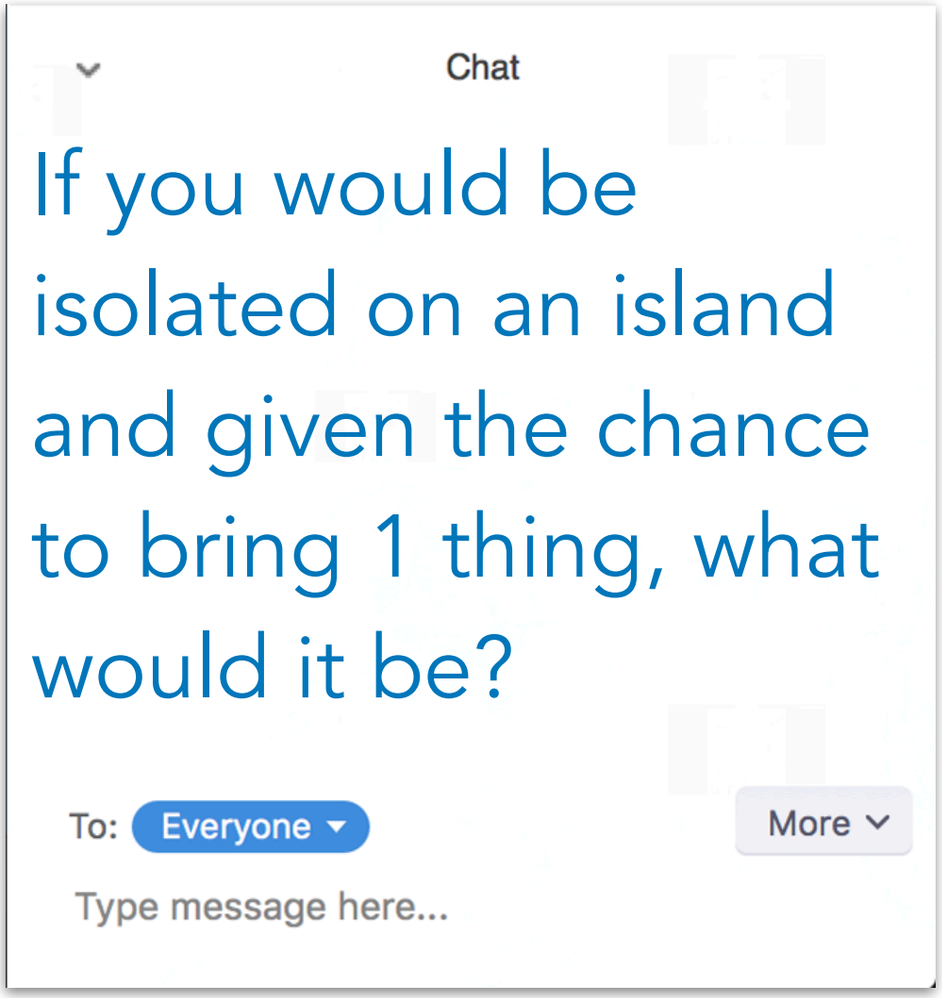


Remember to record the lecture!

PSYCH 252: Statistical Methods for Behavioral and Social Sciences

Simulation 2



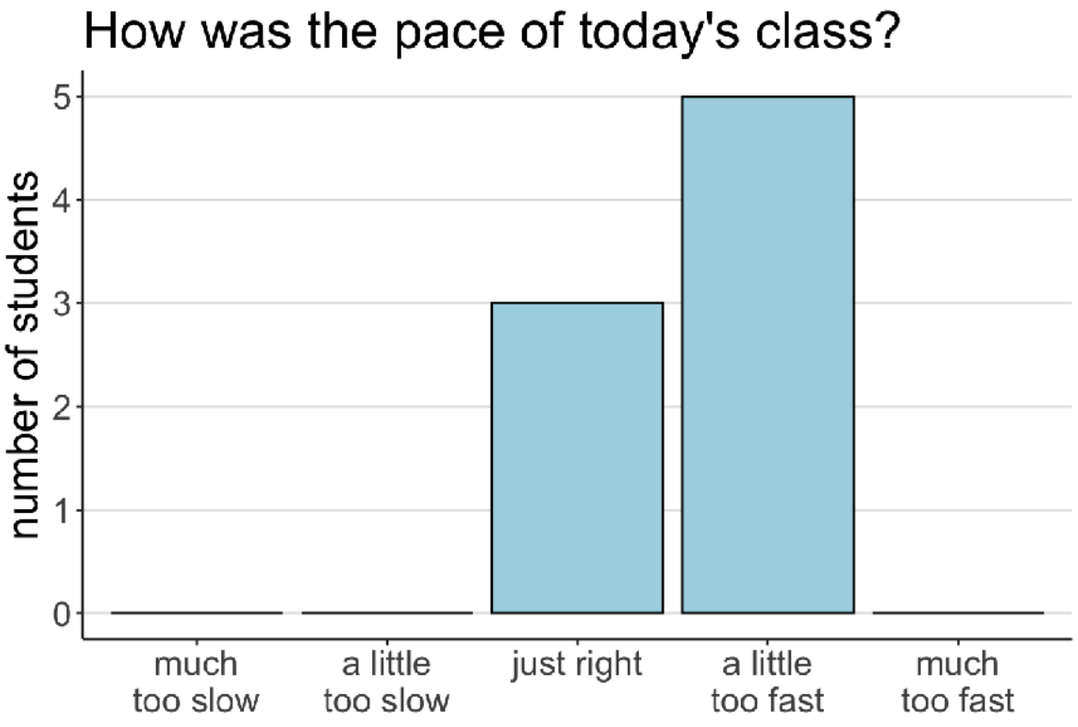


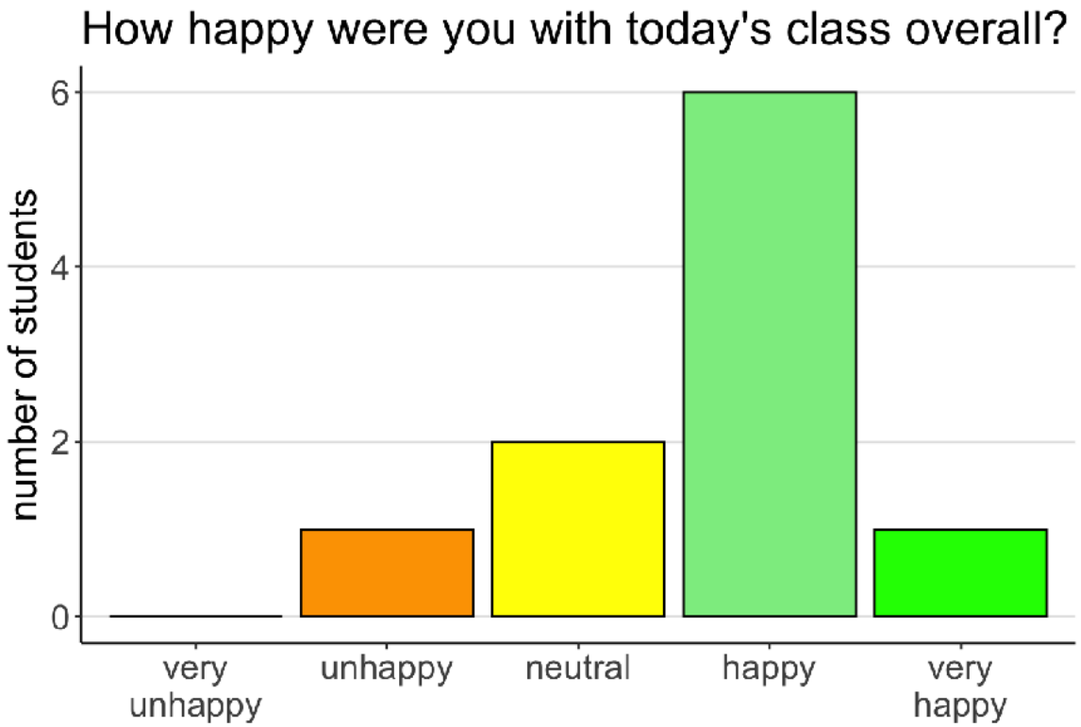
01/28/2021

Logistics

Your feedback

Your feedback





i liked working on the penguin questions in a breakout room, but i felt a little frustrated that we didn't have enough time to get through much of it. ...

for me ... (also, help me slow down by asking questions)

Homework

Show case

Figure 1: Weight of chicks at each time point for 21 days after birth. Each point represents an individual chick's weight at each time point. Ribbons represent bootstrapped 95% confidence intervals.

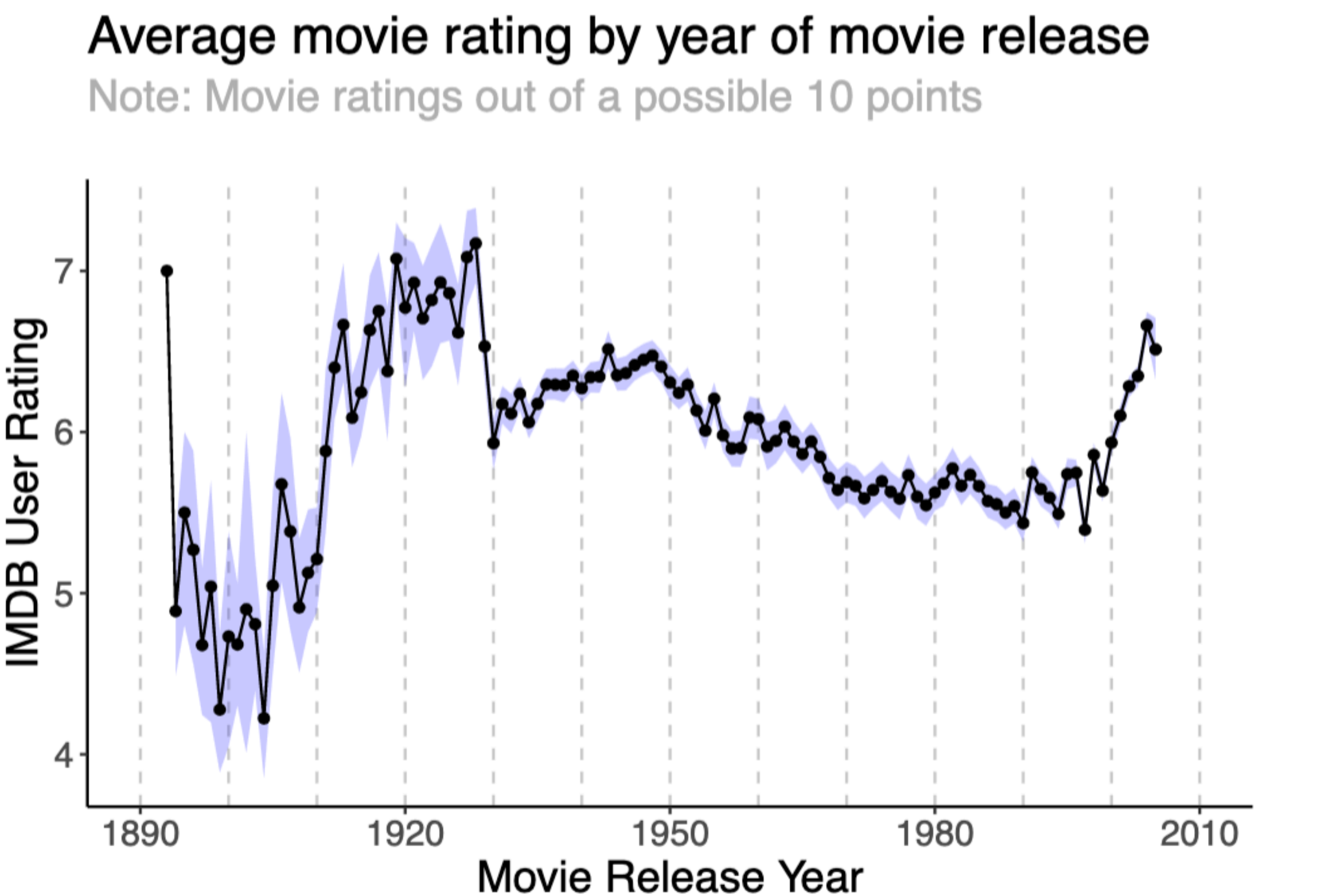
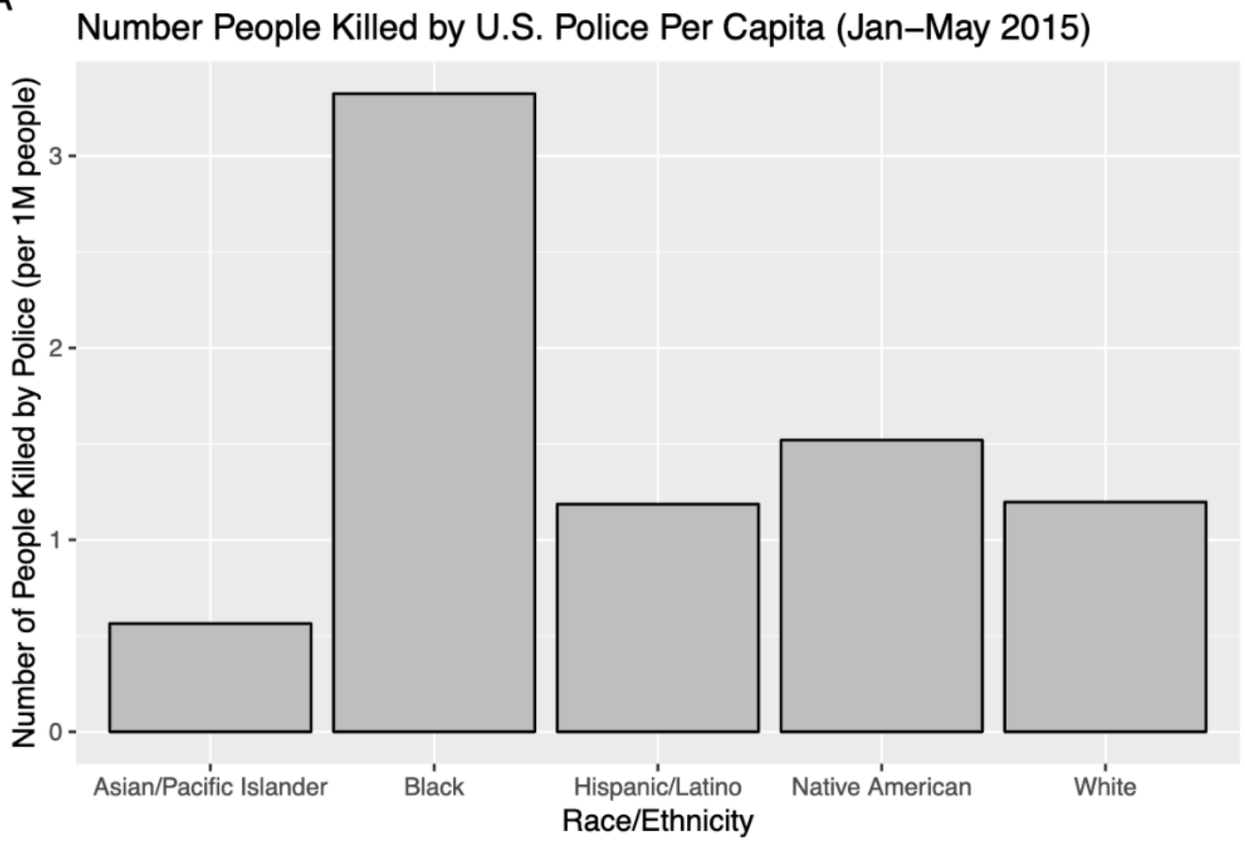


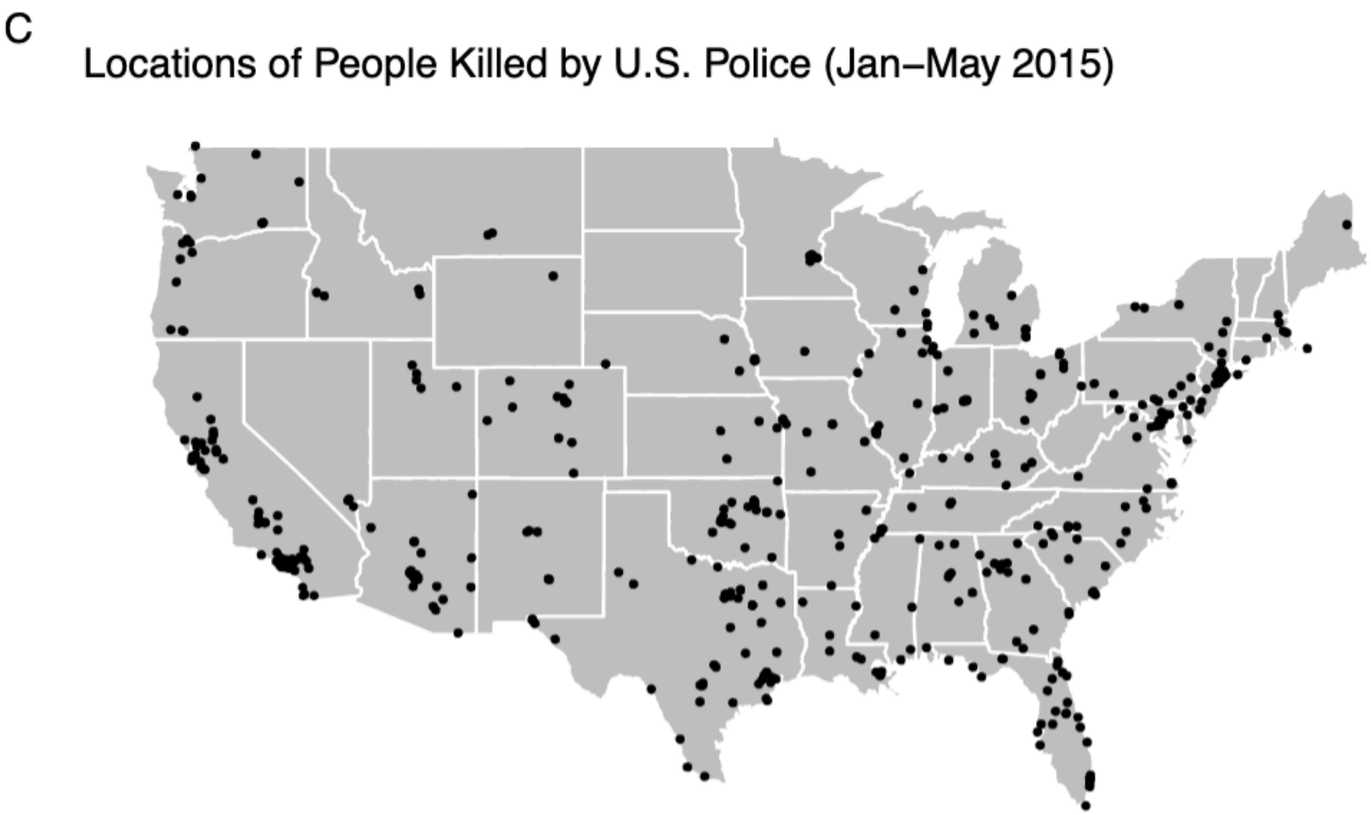
Figure 1: Average IMDB user rating of movies by year of movie release. The haded blue area represents the 95% confidence interval for each year

A



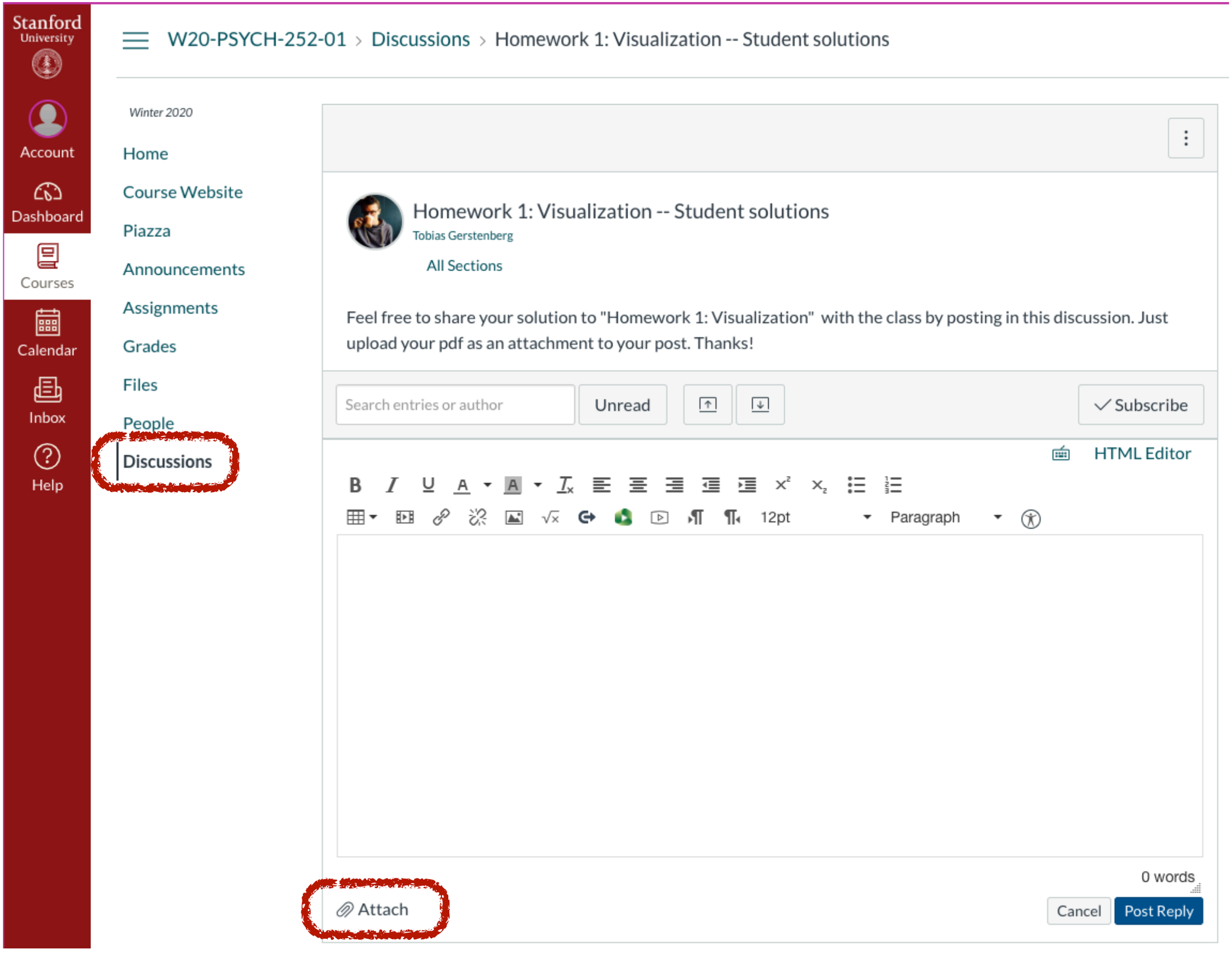
Caption A: The racial disparity in police violence.

C

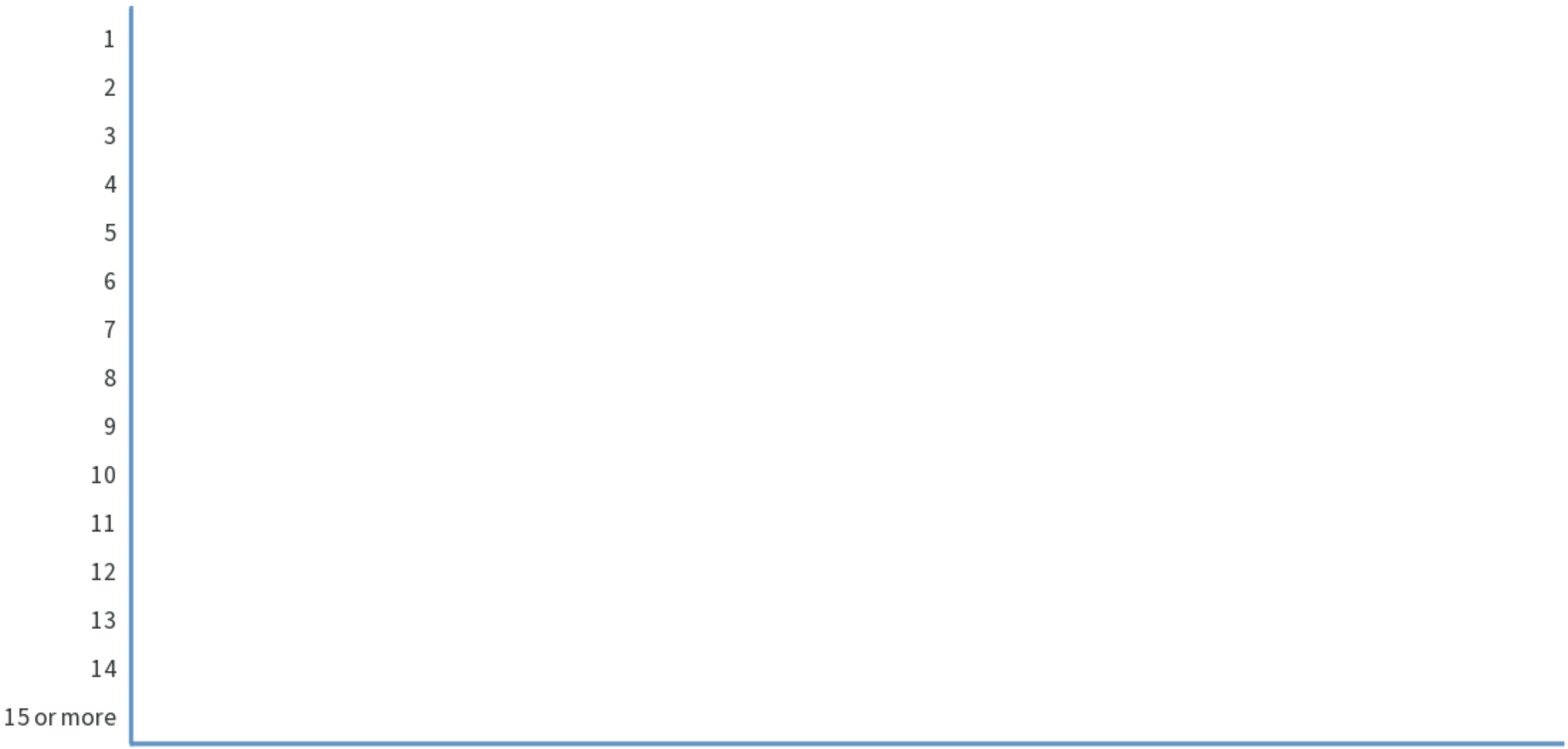


(C) The locations of killings by U.S. police.

Upload your solution to HW1 (if you like)



How many hours did it take you to complete Homework 2?



Homework 3

1 Distributions (1 point)

When we have empirical data, we can compute cumulative probabilities and create probability density functions using `quantile()` and `density()`, respectively. Take a look at the help files for both of these functions to better understand what they're doing.

Consider the following data set:

```
df.p1 = tibble(observation = 1:20,
               rating = c(0.3775909, 0.5908214, 0.07285336, 0.06989763, 0.2180343,
                          1.447484, 0.614781, 0.2698414, 0.4782837, 0.073523,
                          0.6953676, 0.3810149, 0.6188018, 2.211967, 0.5272716,
                          0.517622, 0.9380176, 0.3273733, 0.1684667, 0.2942399))
```

1.1 Quantile (0.5 points)

What's the 60% percentile of the `rating` vari
60% of the values are lower than that value?)

```
### YOUR CODE HERE ###

######################
```

1.2 Density (0.5 points)

Plot the density of the `rating` variable. Use

```
### YOUR CODE HERE ###

######################
```


2 Sampling distribution (7 points)

The sample standard deviation $s = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{n-1}}$ is an unbiased estimator of the population standard deviation.In this exercise, we will run a simulation to compare s with another estimator $s' = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{n}}$ and show that s' is biased. Note that the only difference between s and s' is in the denominator.

2.1 Generate the sample distributions (2 poi

Let's simulate drawing 10,000 samples from a population distril
sample should be 40. To start off, create a new data frame `df.`
3 columns:

- **sim**: index for the simulation (from 1 to 10,000)
- **sample**: index for the sample in each simulation (from 1
- **x**: sample values drawn from the population distribution

For the population distribution assume a normal distribution

3 Permutation test (7 points)

Imagine that you collected data about people's heights from three different places and you are interested whether there are any differences in people's height between the three places.

By visualizing the data, we can see that the variances between the three groups differ considerably, which is troublesome for parametric tests (e.g. a t-test). However, we can perform a permutation test, which is non-parametric. In this case, we are interested in whether the maximal difference between each of the pairs of group means, is greater than we would expect to see by chance.

```
set.seed(1)

df.heights = read_csv("data/df_heights.csv")

df.heights %>%
  ggplot(data = .,
         mapping = aes(x = group,
                       y = height)) +
  geom_point(position = position_jitter(height = 0,
                                        width = 0.1),
             alpha = 0.5) +
  stat_summary(fun.data = "mean_cl_boot",
               shape = 21,
               fill = "lightblue",
               size = 1)
```

Homework 3

remember to set `eval = T` when knitting the file

```{r p2.1, eval = F}
set.seed(1)

n_simulations = 10000 # number of simulations
n_samples = 40 # number of samples in each simulation
population_mean = 0 # ground truth mean
population_sd = 1 # ground truth standard deviation

### YOUR CODE HERE ###
df.samples =
##########################

df.samples %>%
  head(5)

df.samples %>%
  summary()
```

Outline **Goal: Revisit and understand key statistical concepts**

- Inference in frequentist statistics
- Sampling distributions
- What is a p-value?
  - Permutation test
  - t-test
- Confidence intervals
- Bootstrapping

Inference in frequentist statistics

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?

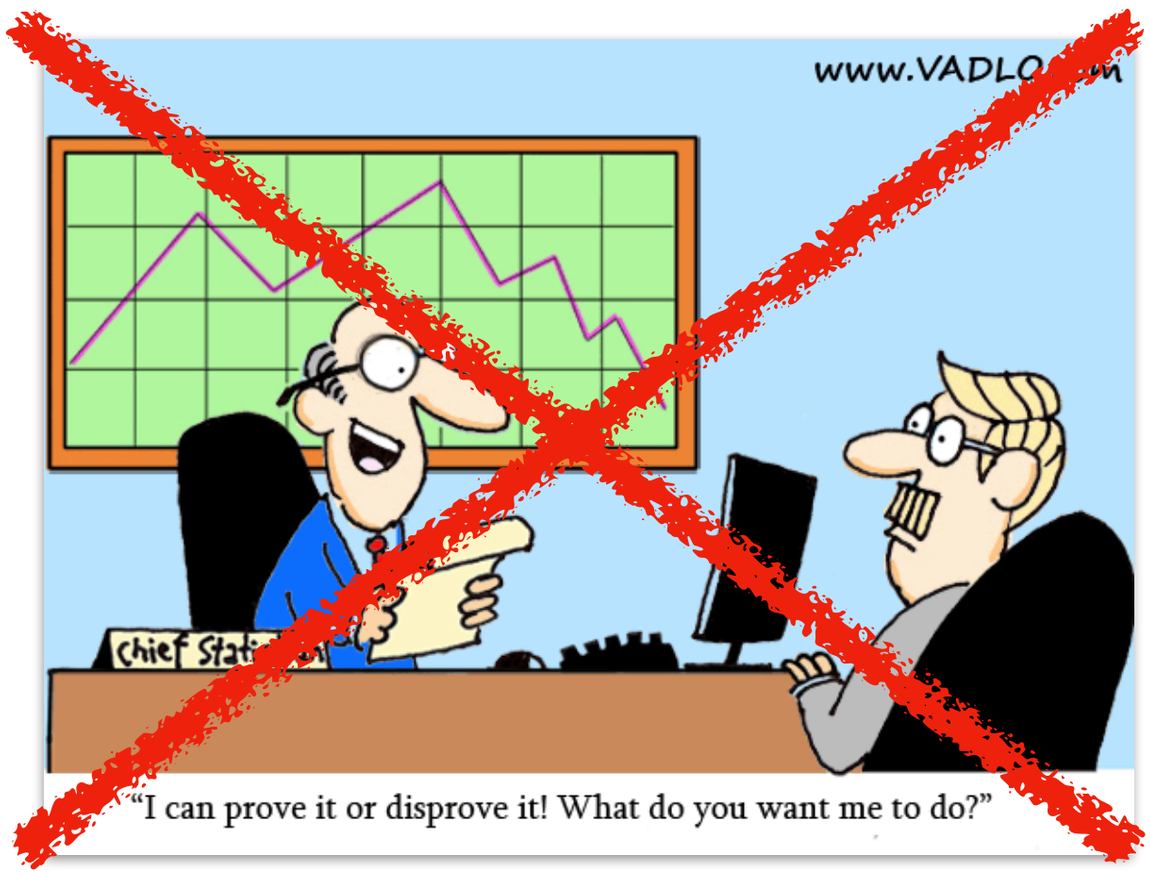
Statistical inference

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!



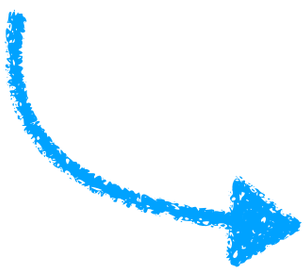
Underlying principle of statistical testing

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. Evaluate the observed test statistic on the sampling distribution; make a decision (either reject or don't reject H_0) based on pre-specified significance level α

The magical component

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

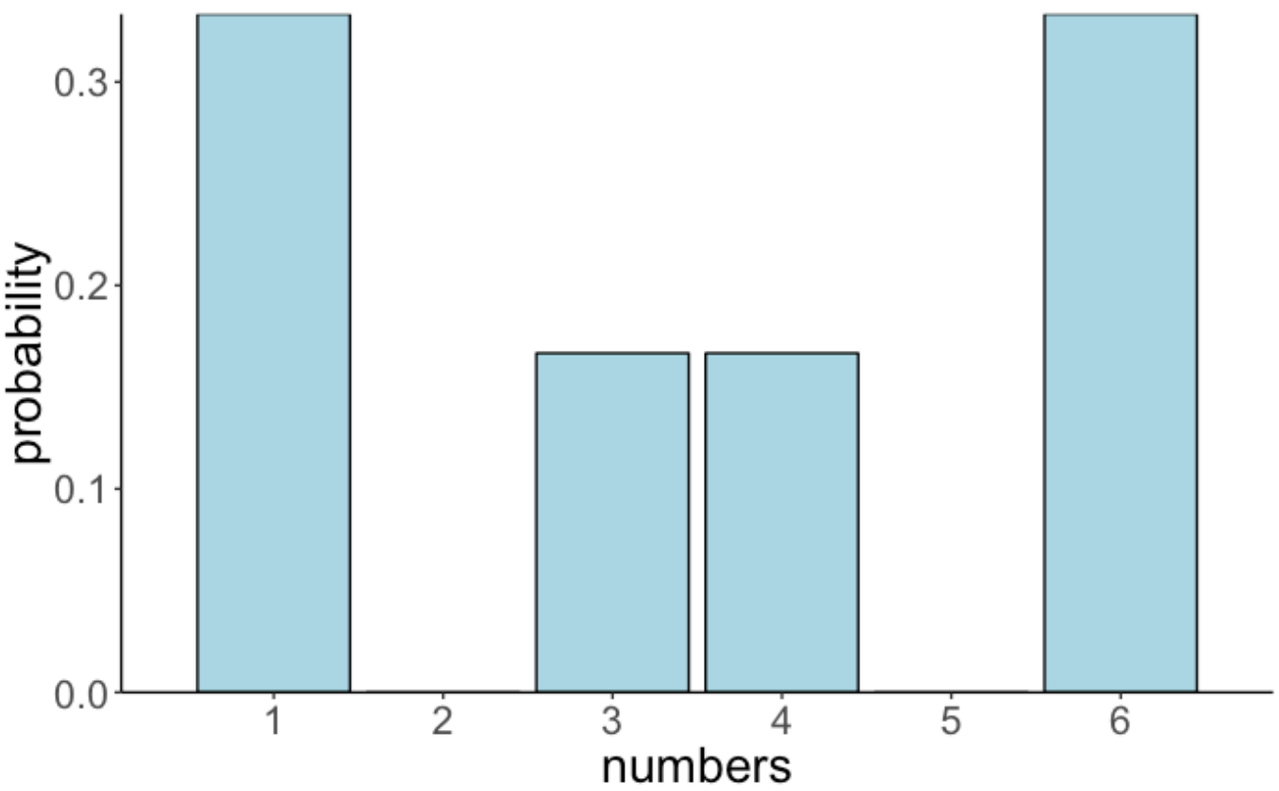
This dates back to pre-computer era where statisticians derived mathematically the distribution of statistical measures for an infinite amount of samples! That's are tricky thing to do and these approximations are typically tied to assumptions such as normality, homoscedasticity, independent observations,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



instead: simulation-based approach

Sampling distributions

heavy metal distribution

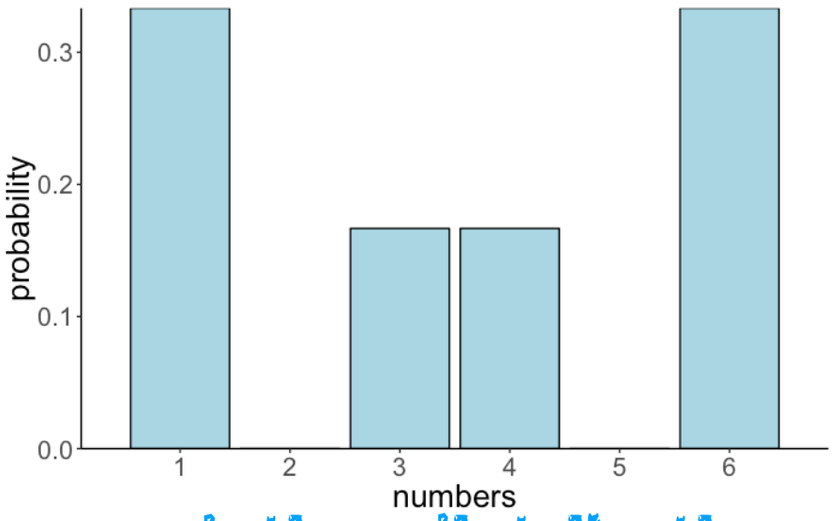


population distribution

Statistical inference

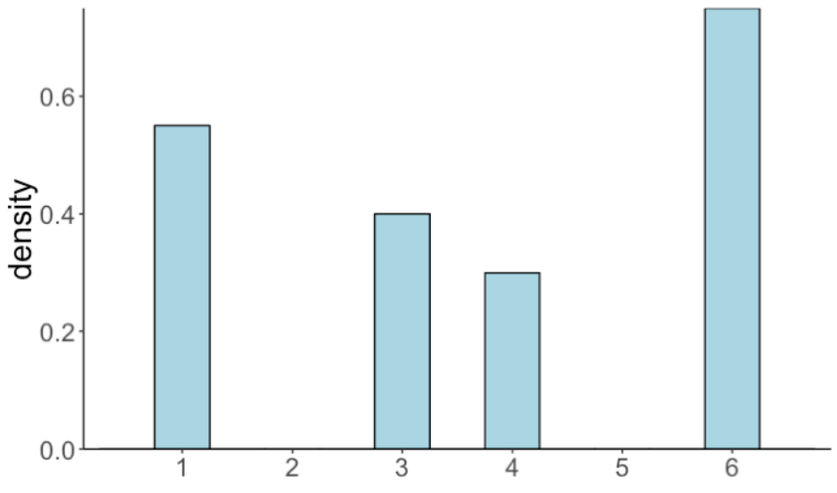
The process of making claims about a population based on information from a sample.

heavy metal distribution



sampling

inference



population distribution

our sample

Statistical inference

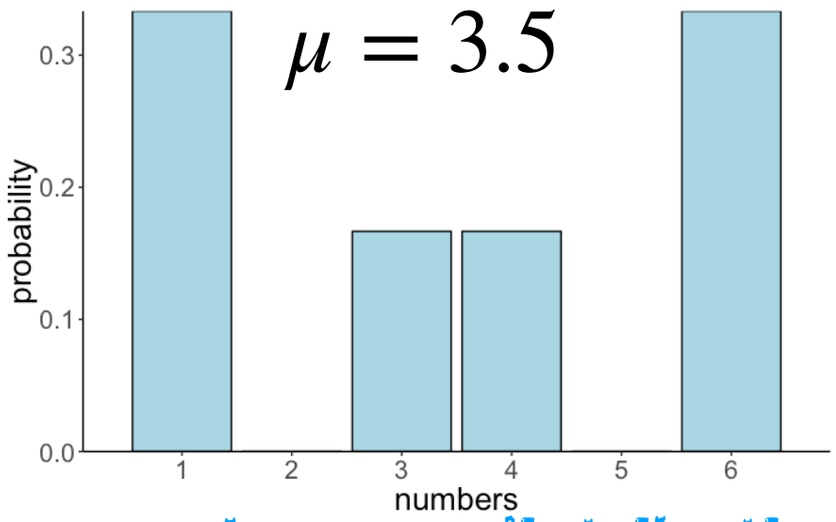
what's the
population mean?

sample mean = 3.725

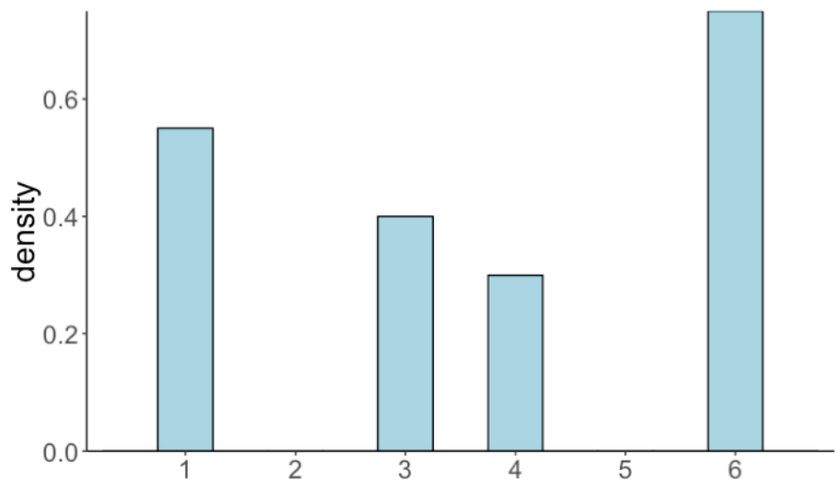
standard deviation = 2.05

n = 40

heavy metal distribution



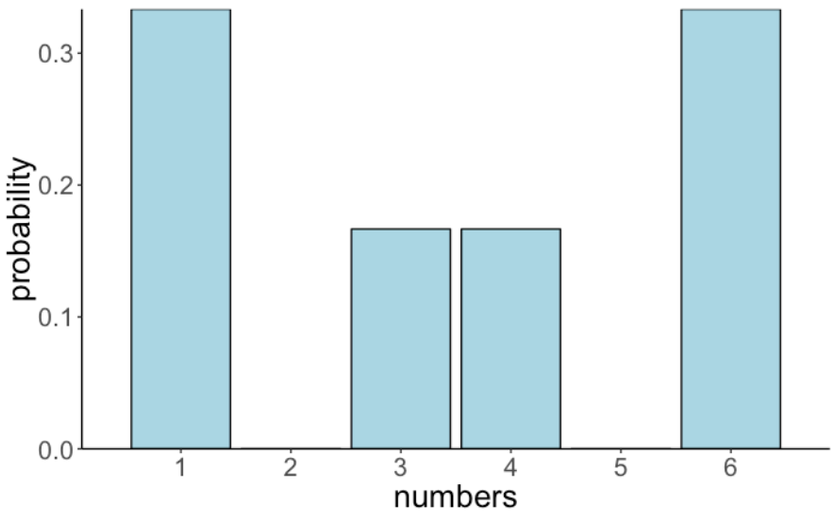
true unknown distribution



our sample

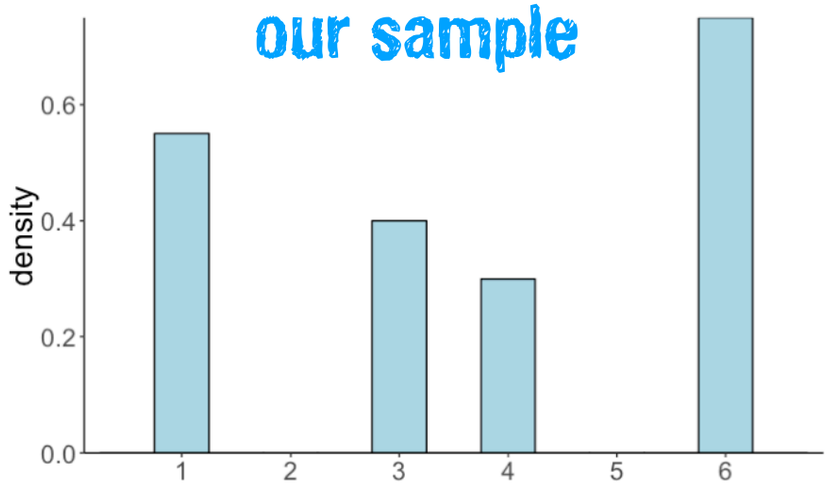
Sampling variation

heavy metal distribution

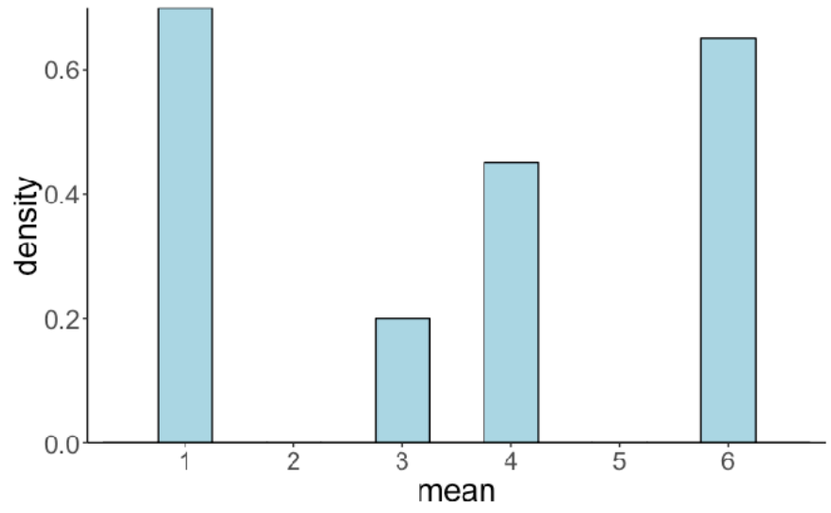


population distribution

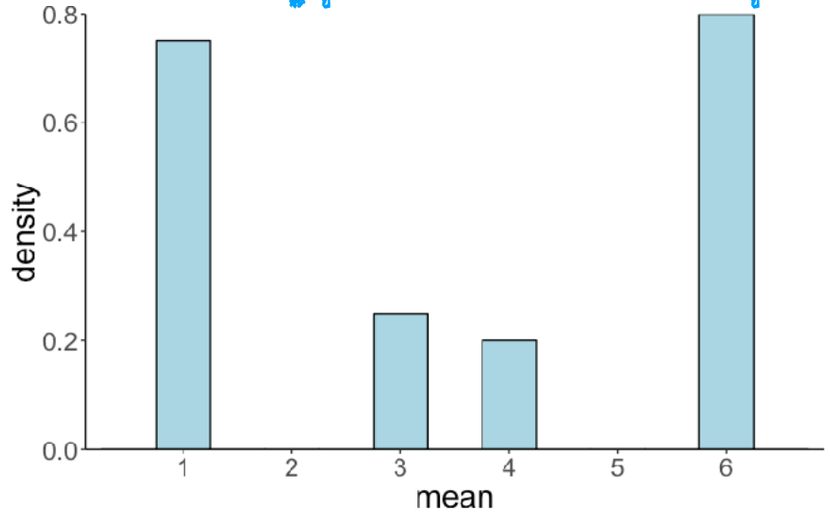
our sample



hypothetical sample



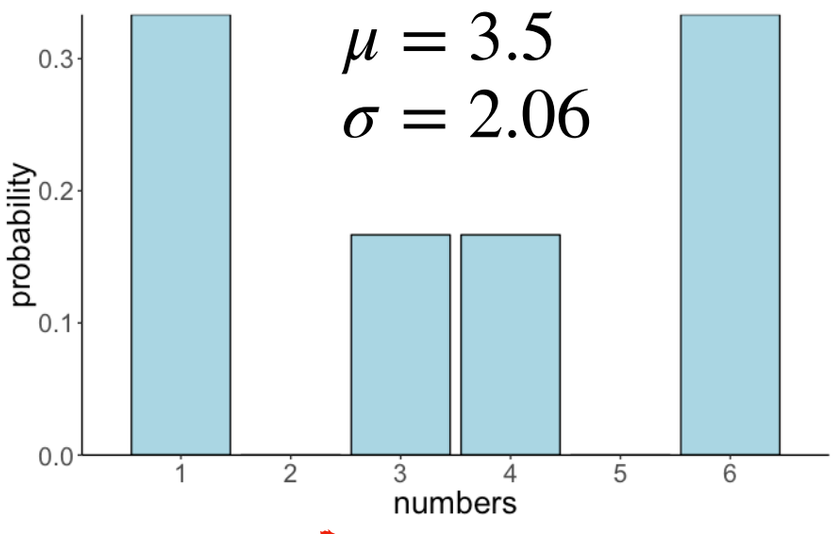
another hypothetical sample



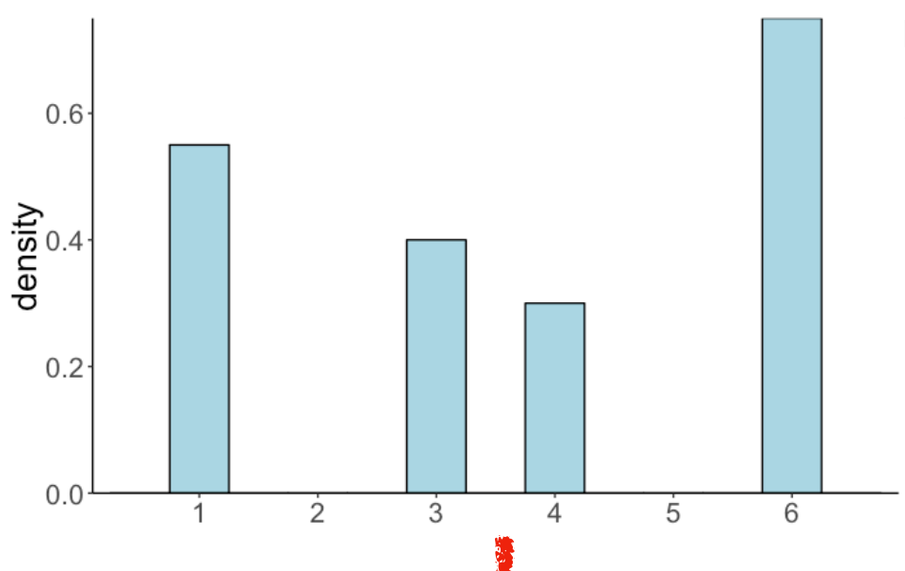
Sampling distribution

population distribution

heavy metal distribution

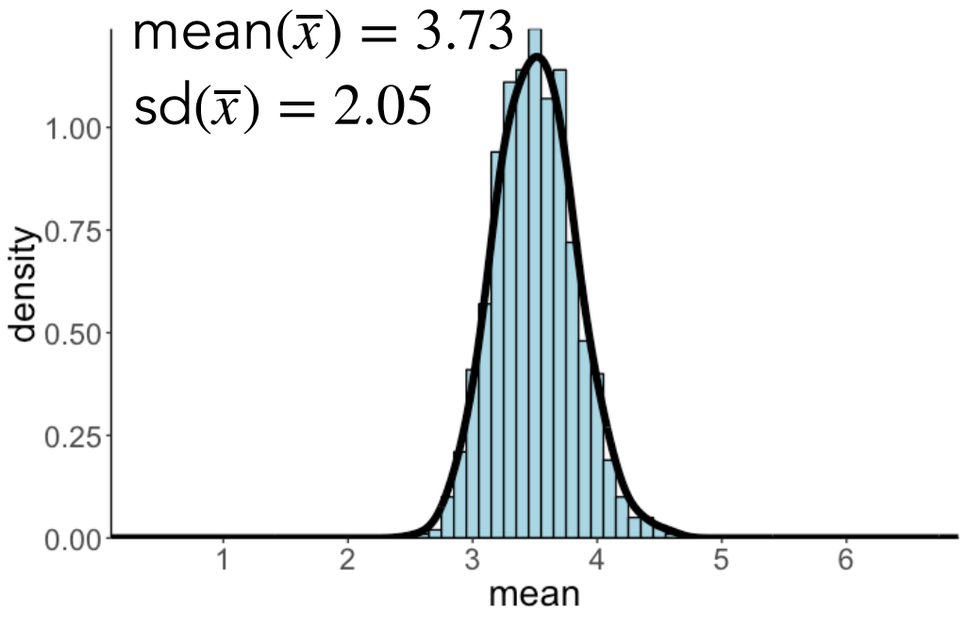


our sample



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

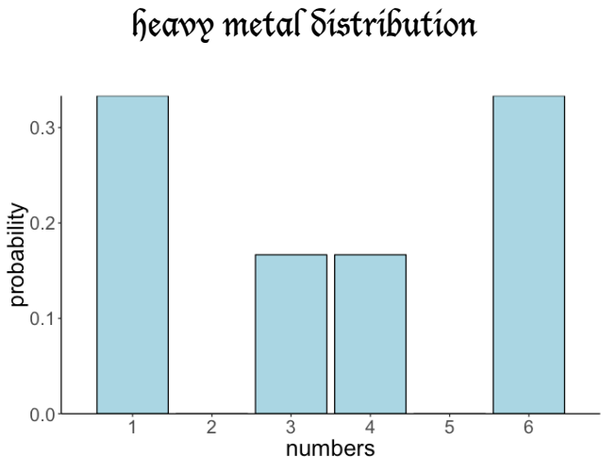
sampling distribution of the mean



p-values

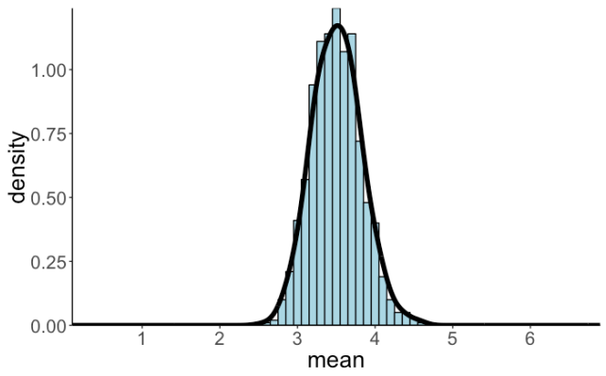
confidence intervals

3 distributions in statistical inference



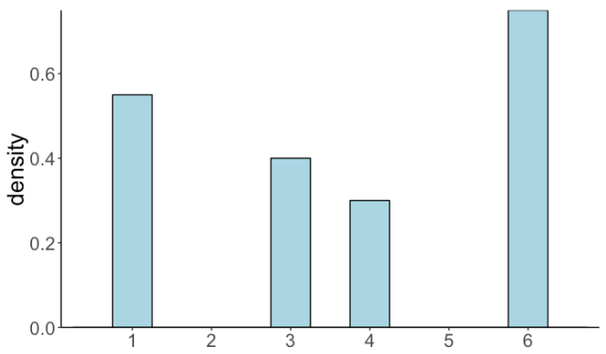
population distribution

- unknown
- our target for inference
- e.g. we might be interested in the mean of the population distribution



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, ...)
- make an inference about the population via the sampling distribution

What is a p-value?

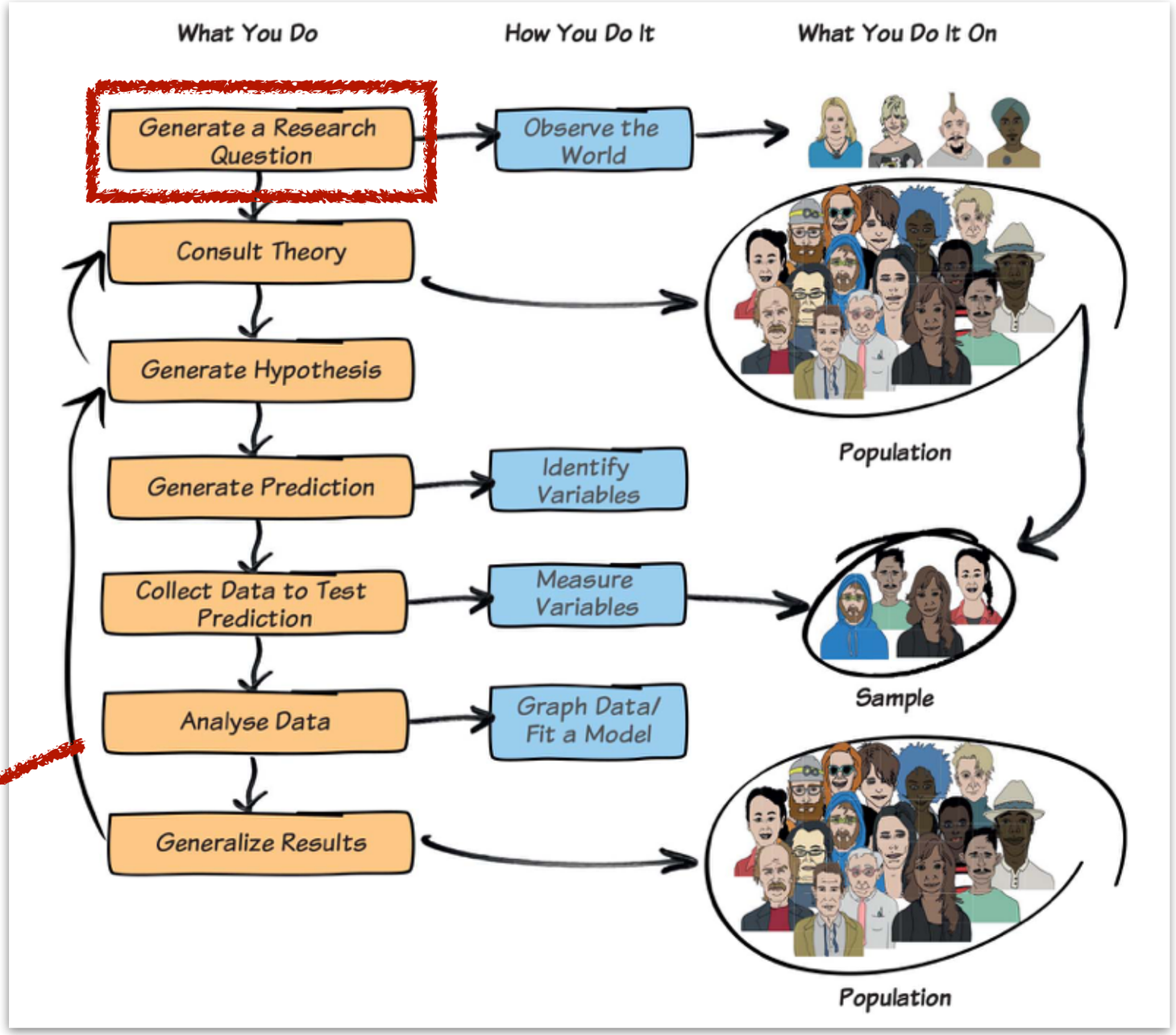
Statistical inference

null hypothesis

$$H_0 : \mu_1 = \mu_2$$

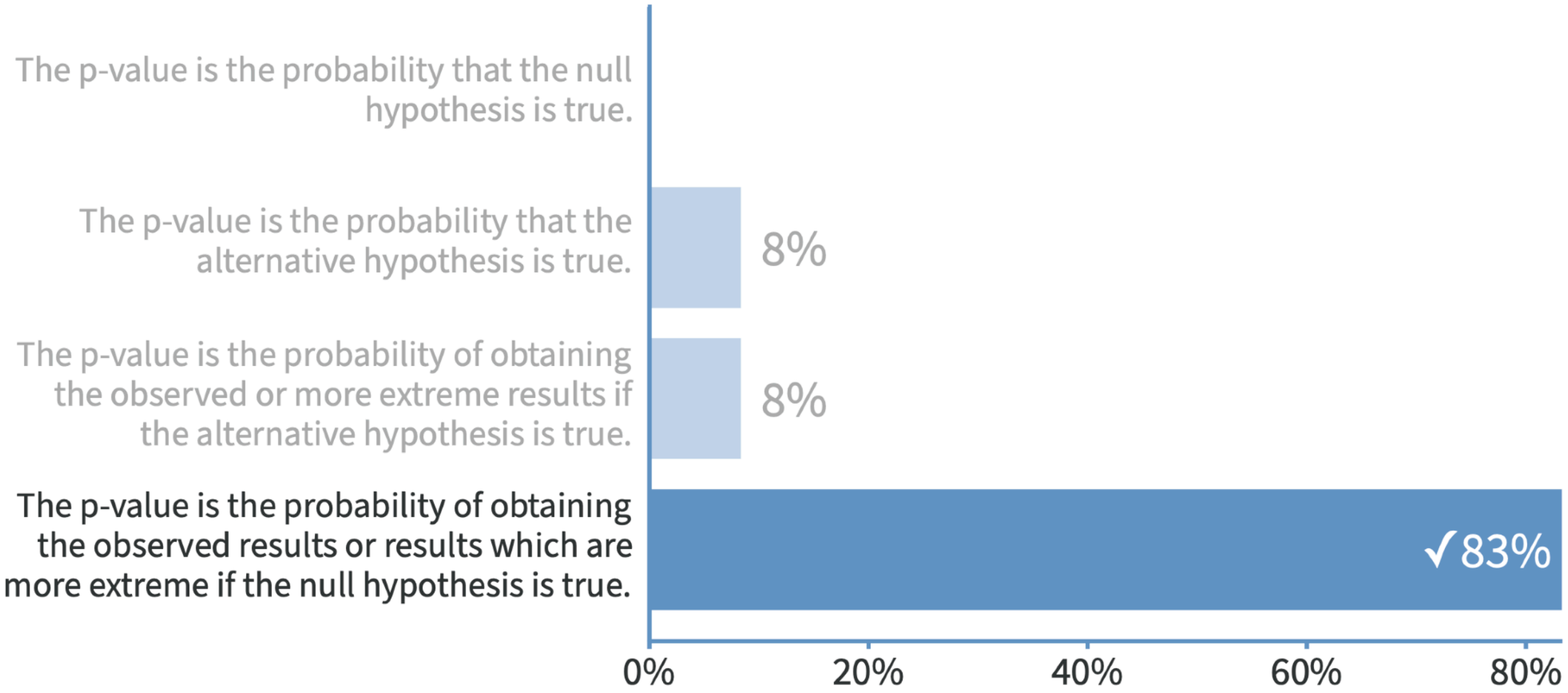
alternative hypothesis

$$H_1 : \mu_1 < \mu_2$$

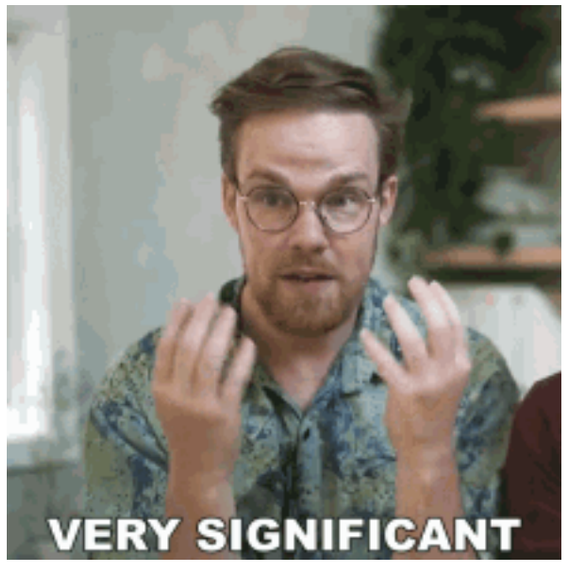


a p-value, yay!

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



What is a p-value? Your answers



- something related to the probability of type I or II error. normally the smaller the better and < .05 is required.
- Probability that the results obtained are true assuming that the null hypothesis is in place
- an arbitrary threshold where there is (usually, with a p value of 0.05) a 95% chance the the results you are seeing didn't occur by chance
- Percent likelihood that result was by chance
- the degree to which one is statistically certain an event has occurred outside the realm of statistical probability.
- it's the probability that the value I got "is true"
- It is the probability that a null hypothesis is held or rejected.
- Nope
- probability of observing data or more extreme data under the null hypothesis

What is a p-value?

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

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

what we're actually interested in!

$$\rightarrow p(H_1 = \text{true} \,|\, \text{test statistic} \geq \text{observed value})$$

... we'll have to wait for Reverend Bayes

$$p(H \,|\, D) = \frac{p(D \,|\, H) \cdot p(H)}{p(D)}$$

H = Hypothesis

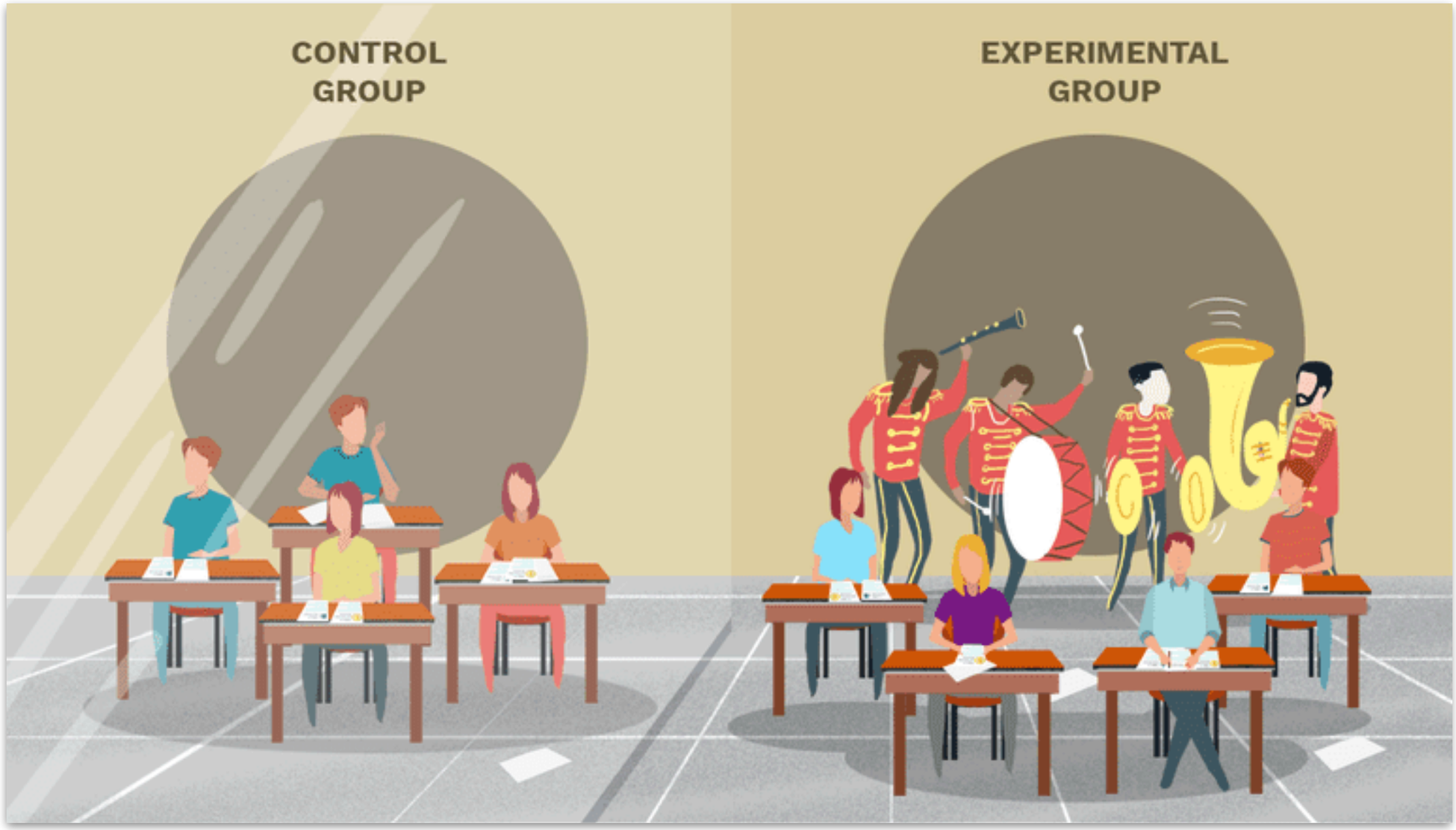
D = Data

Logic of inference

- calculate a **test statistic** based on the sample
  - for example, the difference between the means of two conditions
- build a **sampling distribution** of this statistic assuming that the null hypothesis is true
  - use math or resampling methods
- **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

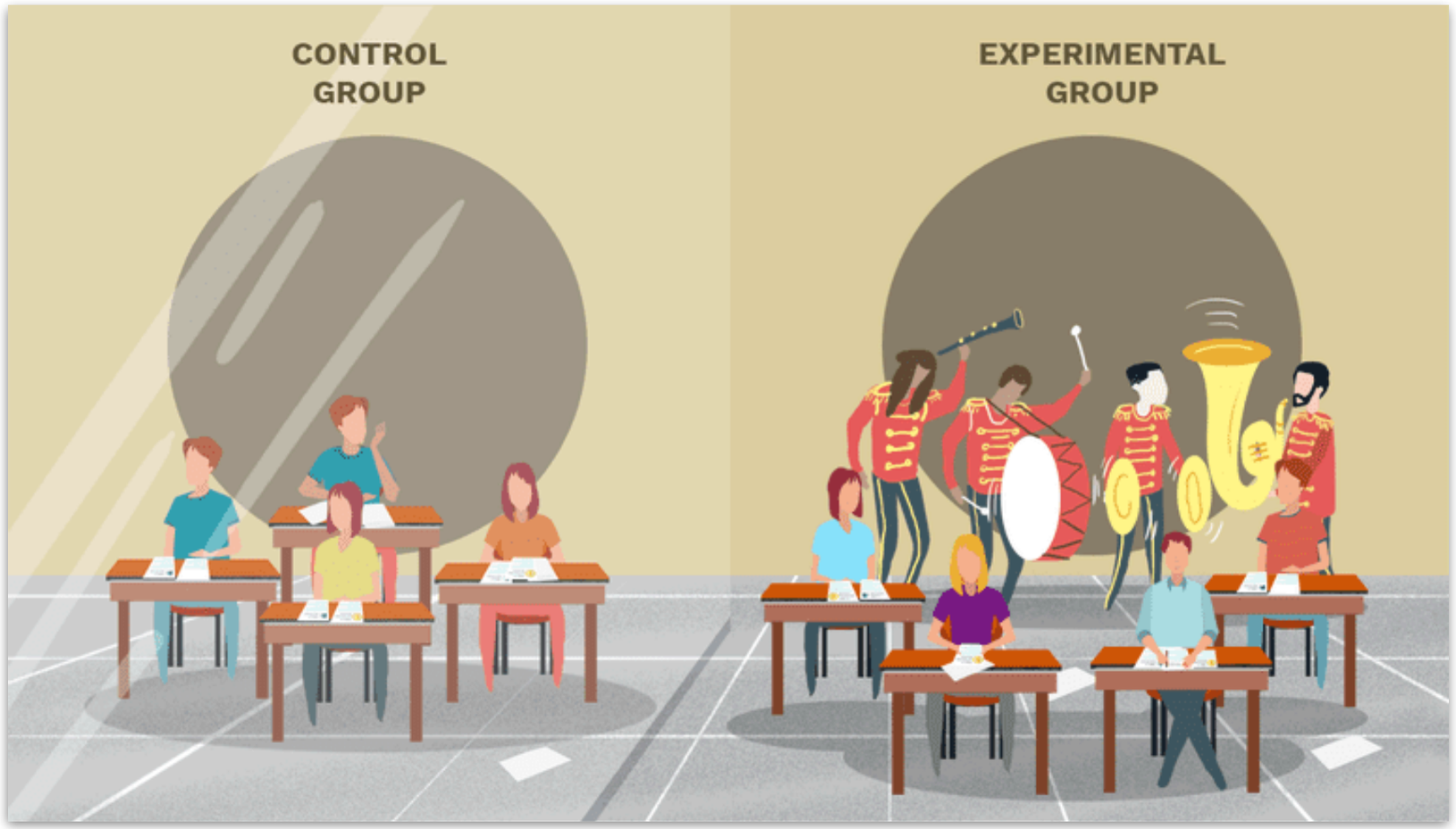
Permutation test



Research question:

Will student test scores be affected by distracting sounds (e.g. the Stanford marching band)?

Permutation test



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

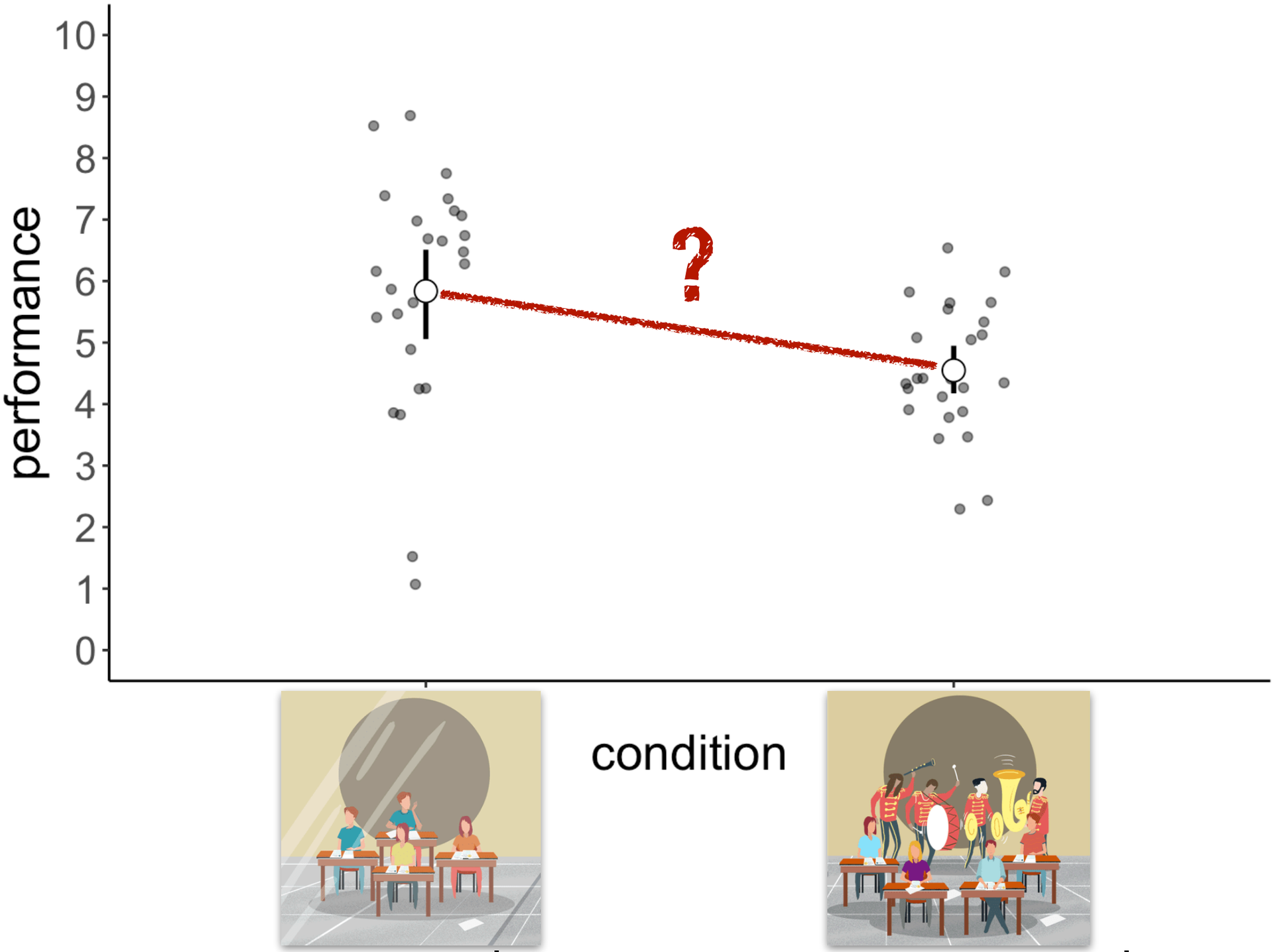
There is no difference between the control group and the experimental group

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

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

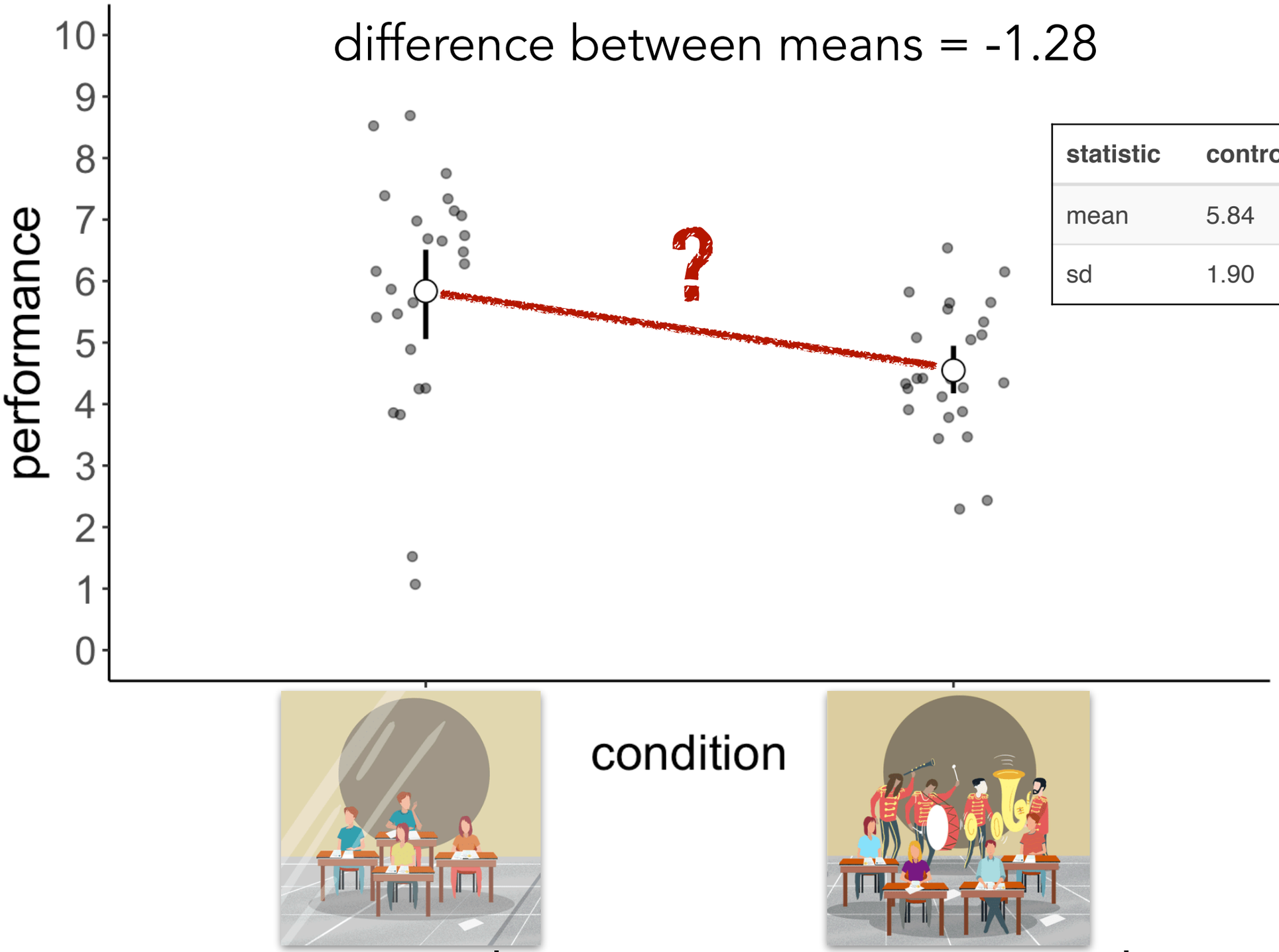
Permutation test

Is the difference in performance statistically significant?



Permutation test

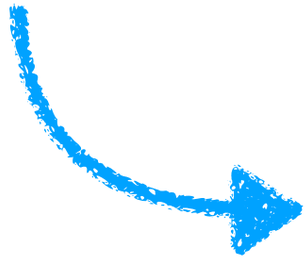
Is the difference in performance statistically significant?



statistic	control	experimental
mean	5.84	4.55
sd	1.90	1.06

Permutation test

- **logic**:
  - assuming our experimental manipulation made no difference, what would be 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

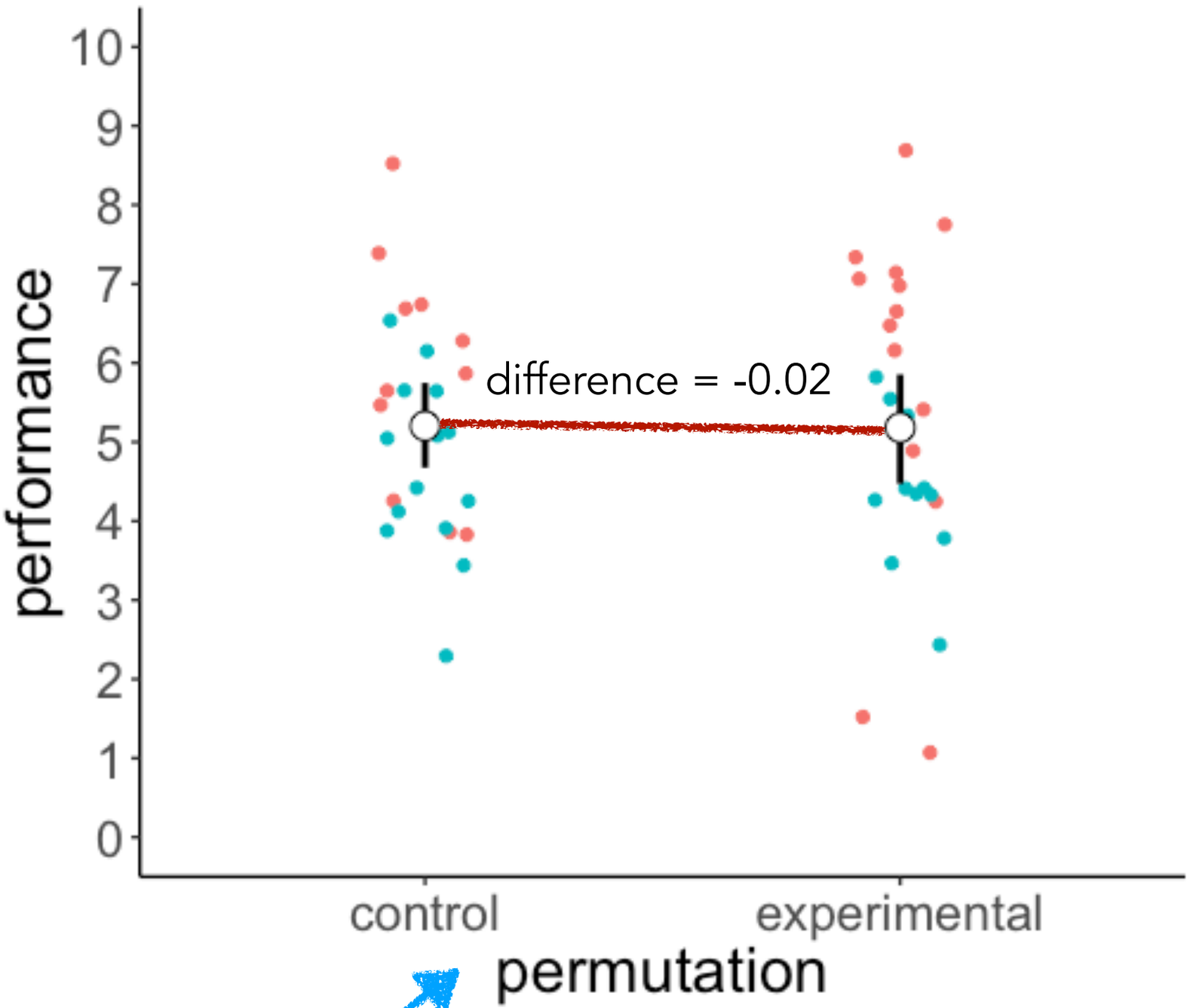
observed data

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	experimental	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

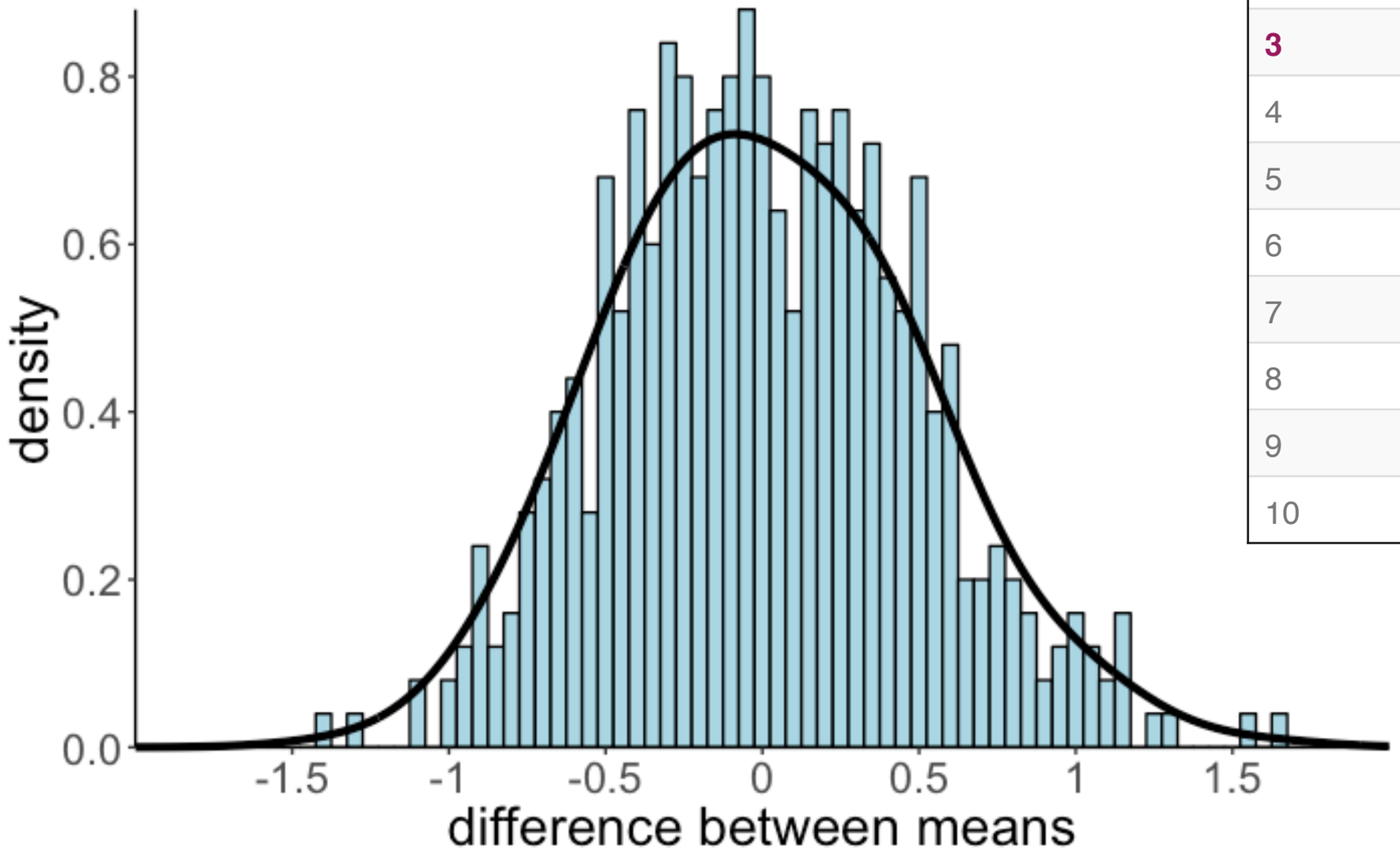
participant	condition	performance
1	experimental	4.25
2	control	5.87
3	experimental	3.83
4	experimental	8.69
5	experimental	6.16
26	control	4.42
27	experimental	4.27
28	control	2.29
29	control	3.78
30	experimental	5.13

3

⋮

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

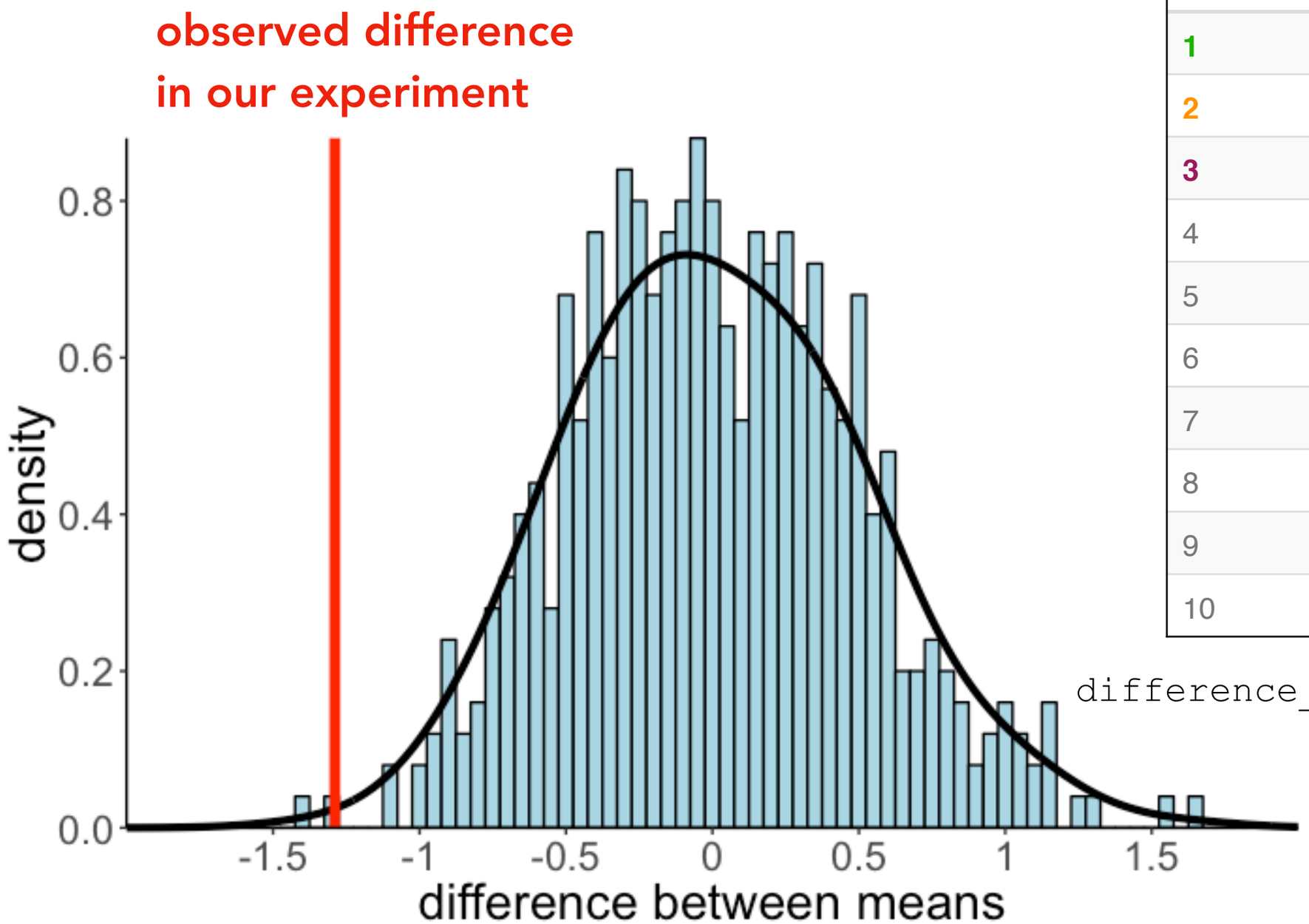


df.permutations

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 was true)

Permutation test



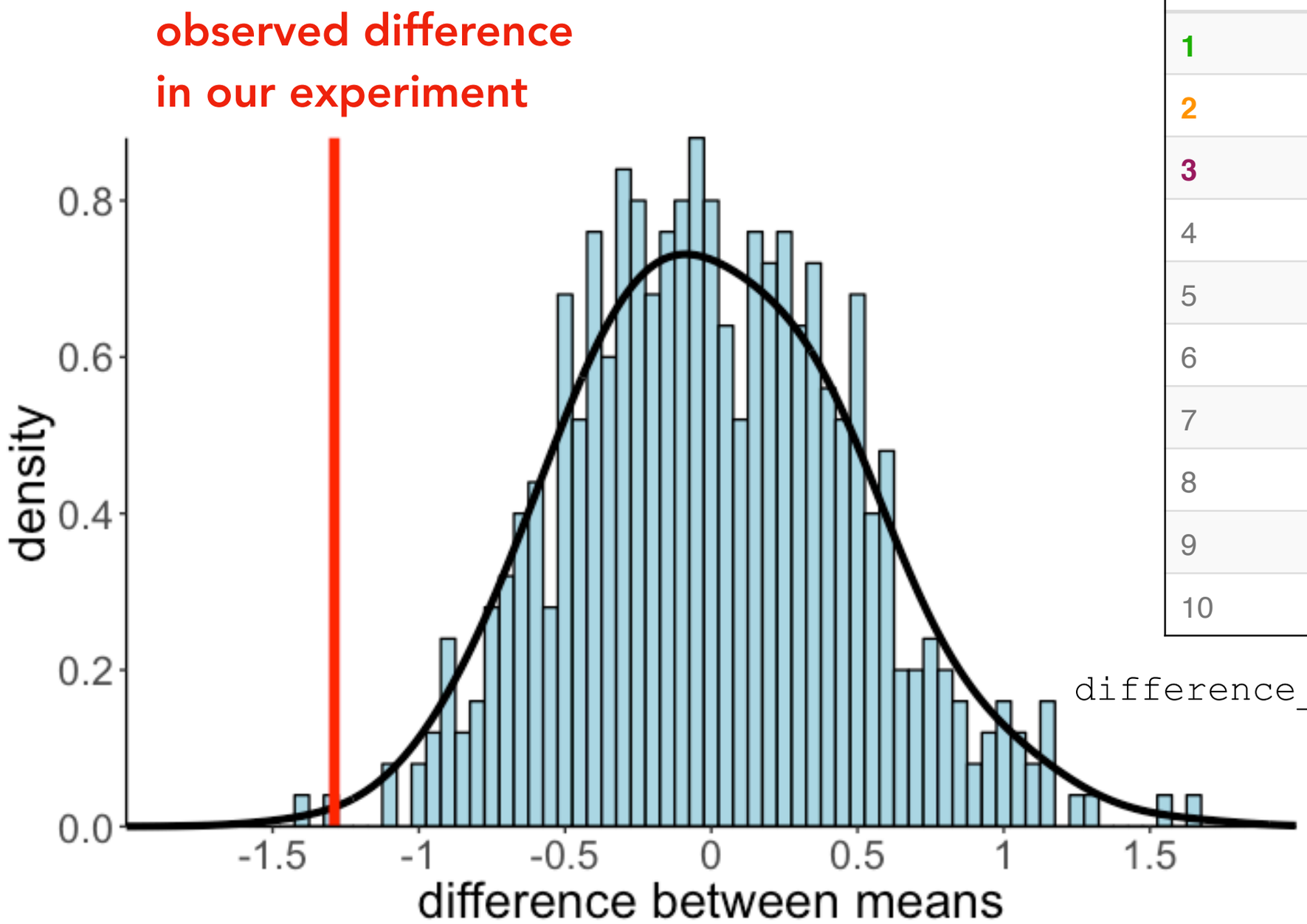
df.permutations

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

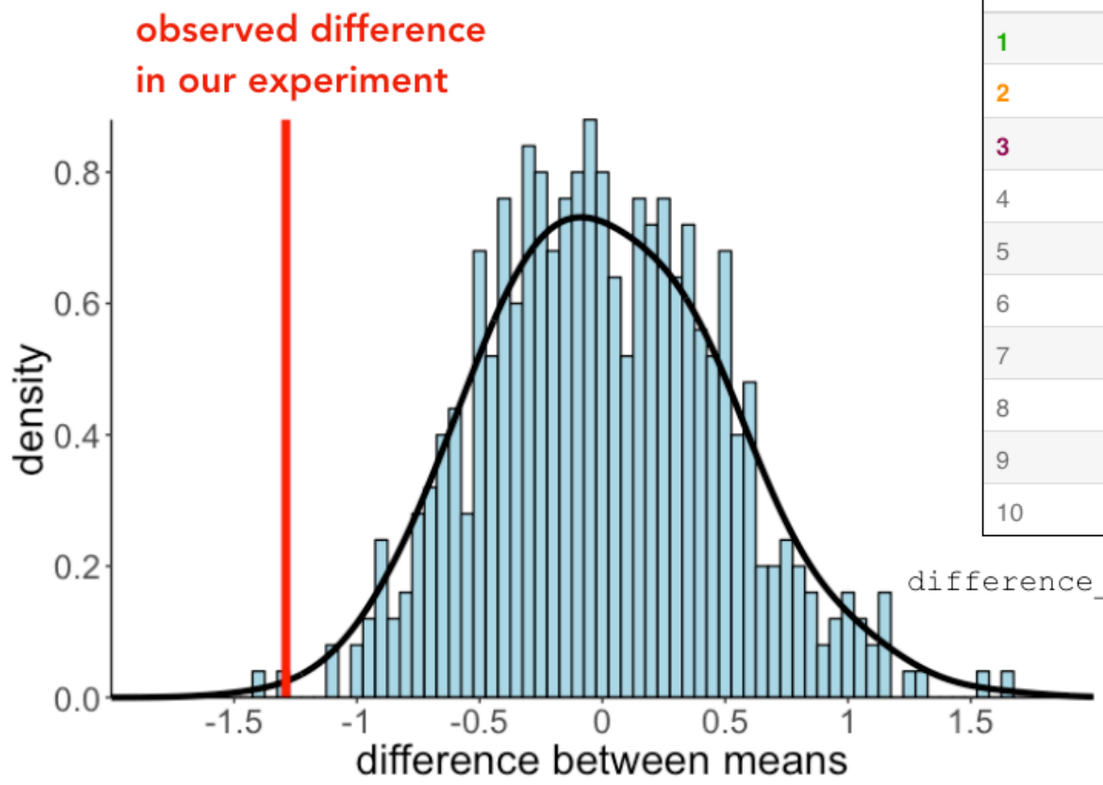
```
1 #calculate p-value of our observed result
2 df.permutations %>%
3   summarize(p_value = sum(mean_difference <= difference_actual)/n())
```

p-value = .002

What is a p-value?

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

57

Permutation test

```
 1 n_permutations = 500
 2
 3 # permutation function
 4 func_permutations = function(df){
 5   df %>%
 6     mutate(condition = sample(condition)) %>%
 7     group_by(condition) %>%
 8     summarize(mean = mean(performance)) %>%
 9     pull(mean) %>%
10     diff()
11 }
```

set the number of permutations

shuffle the condition labels

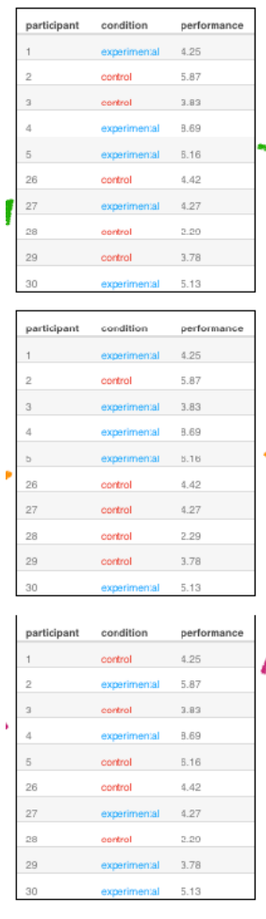
calculate difference between group means

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	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

```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 = tibble(
  permutation = 1:n_permutations,
  mean_difference = replicate(n = n_permutations, func_permutations(df.data))
)
```

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

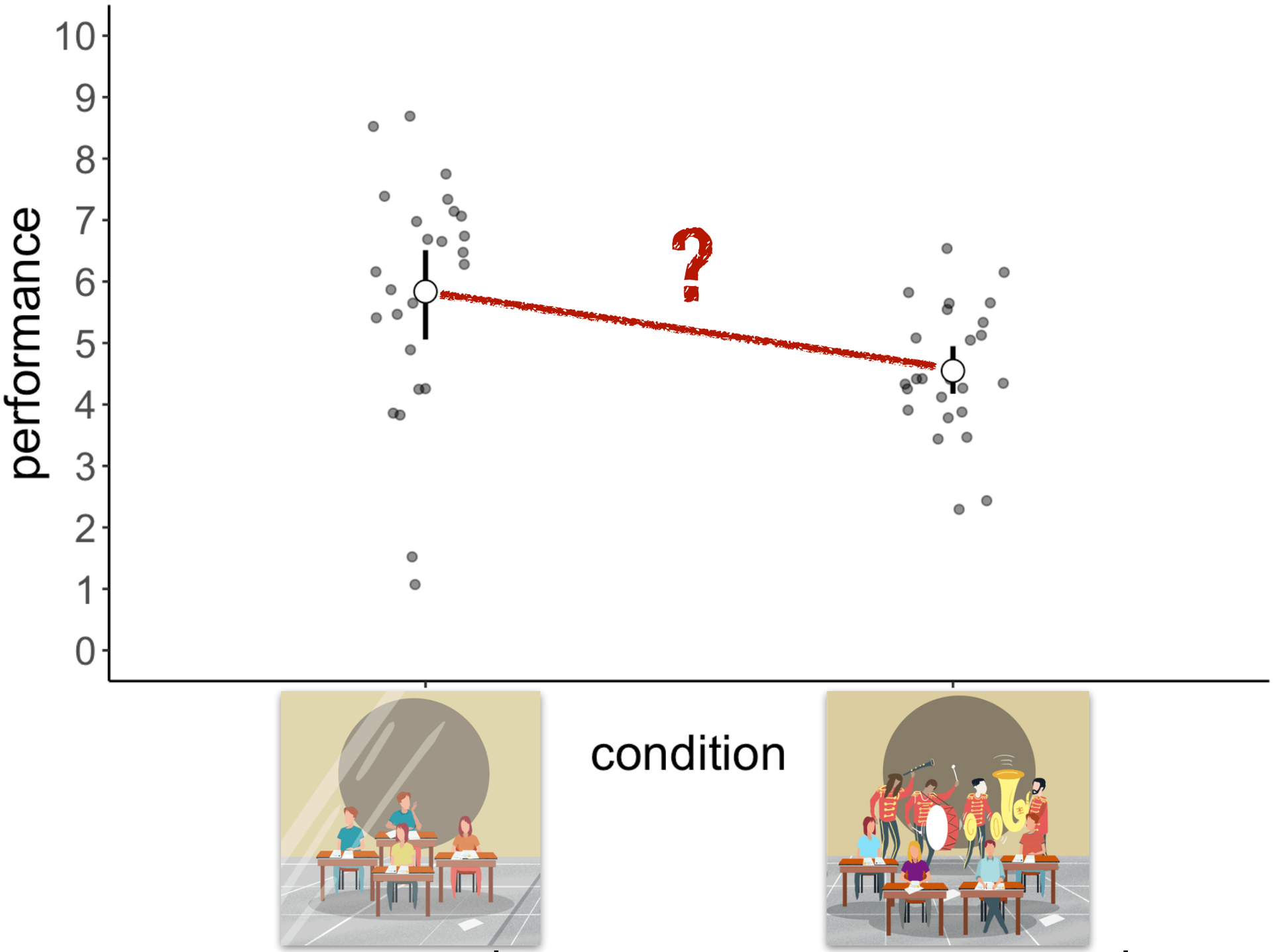
run the `func_permutations()` function many times

(instead of using a for loop)

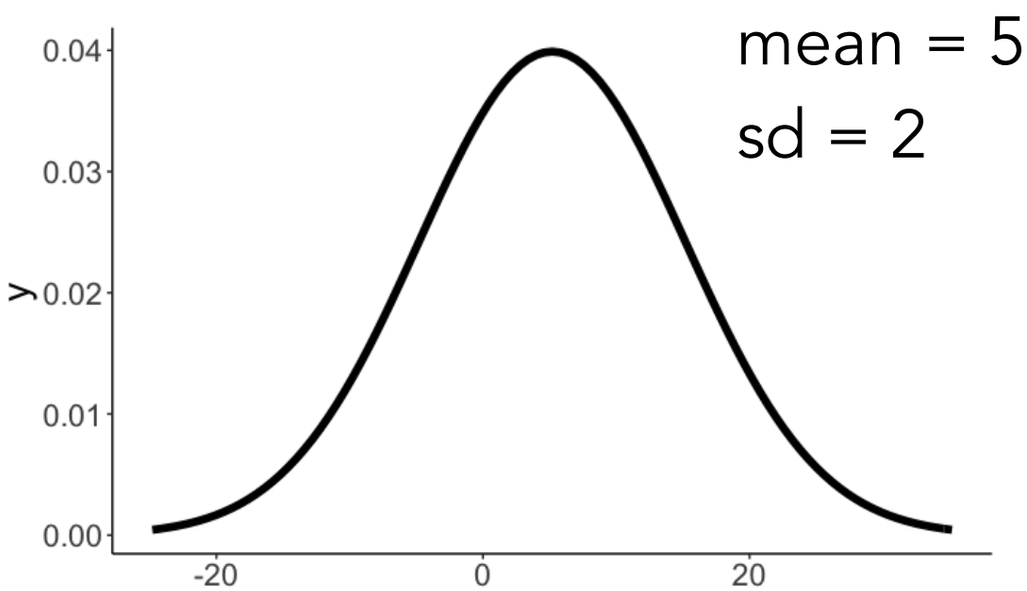
t-test

t-test

Is the difference in performance statistically significant?



two-sample t-test

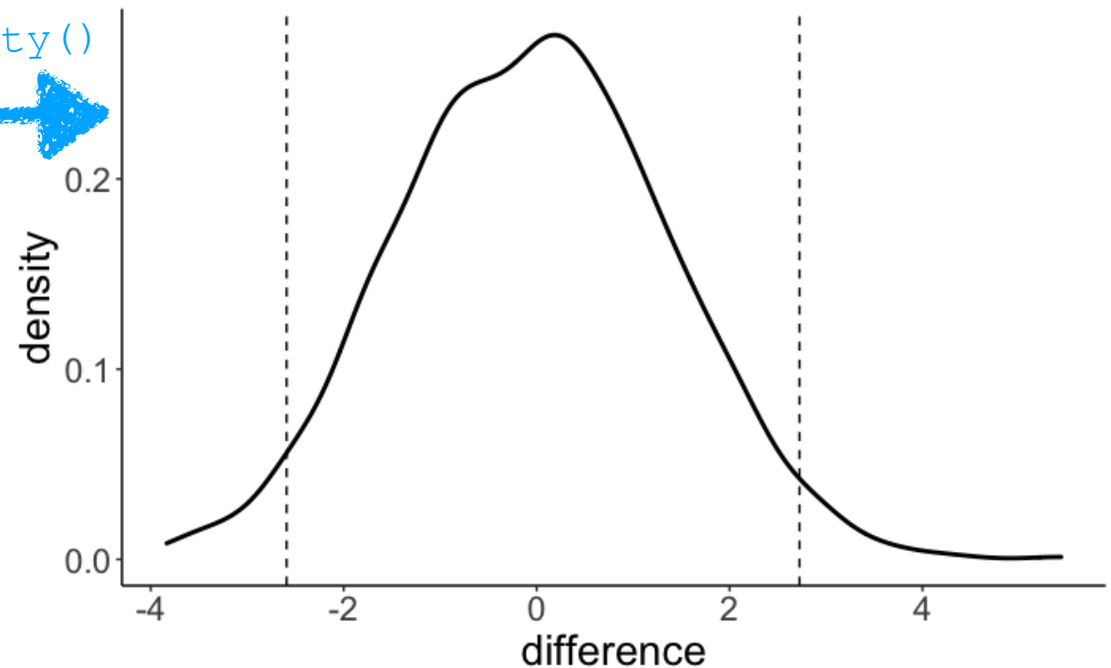



```r
set.seed(1)

n_simulations = 1000
sample_size = 100
mean = 5
sd = 2

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2)
```

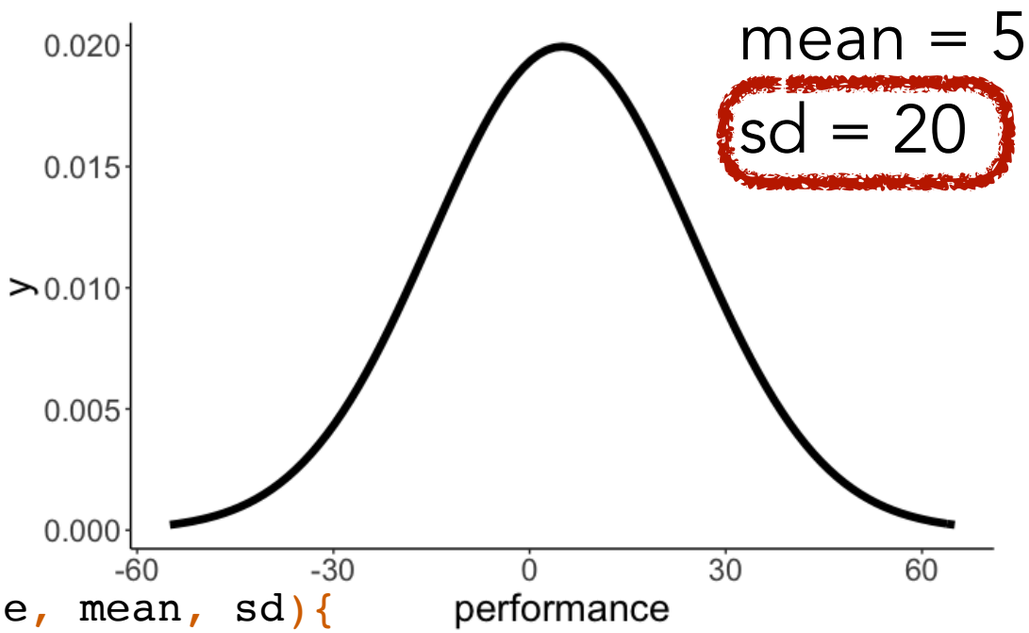
simulation	sample1	sample2	difference
1	6.28	5.16	1.13

geom_density()

What if sd = 20?



two-sample t-test



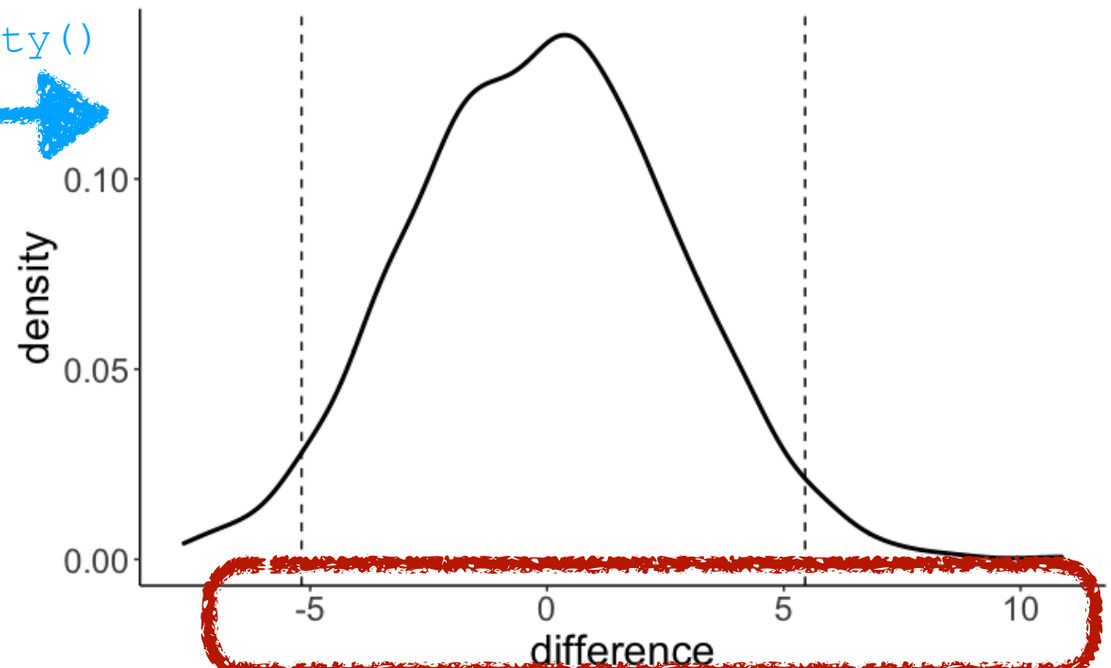

```r
set.seed(1)

n_simulations = 1000
sample_size = 100
mean = 5
sd = 20

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2)
```

simulation	sample1	sample2	difference
1	7.18	4.93	2.25
2	4.24	5.57	-1.32
3	5.59	8.99	-3.40
4	6.03	4.71	1.32
5	4.22	2.48	1.73

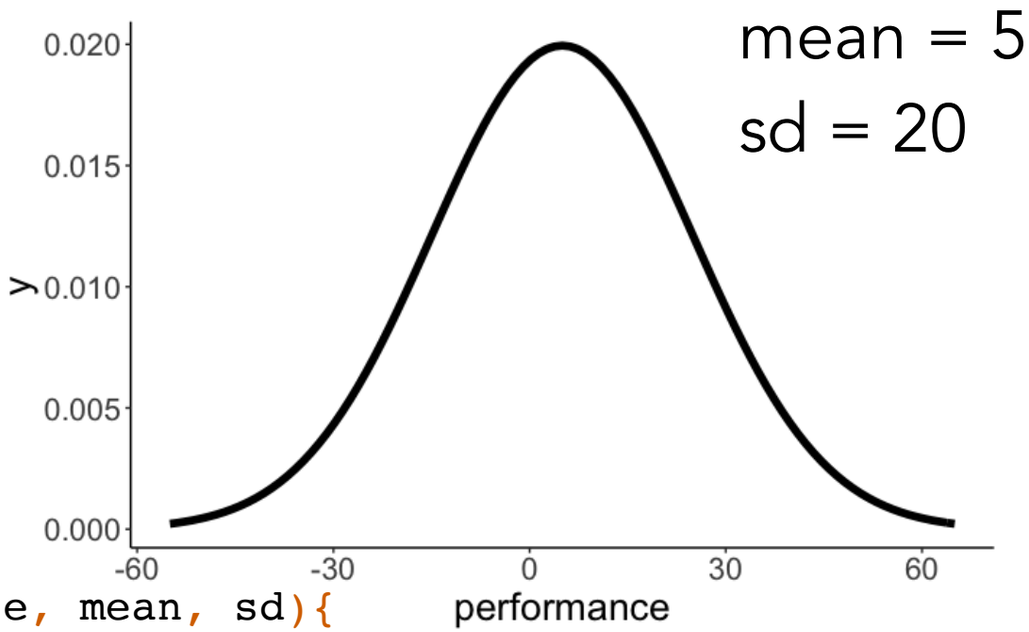
⋮

geom_density()

What if sample size N = 1000?



two-sample t-test



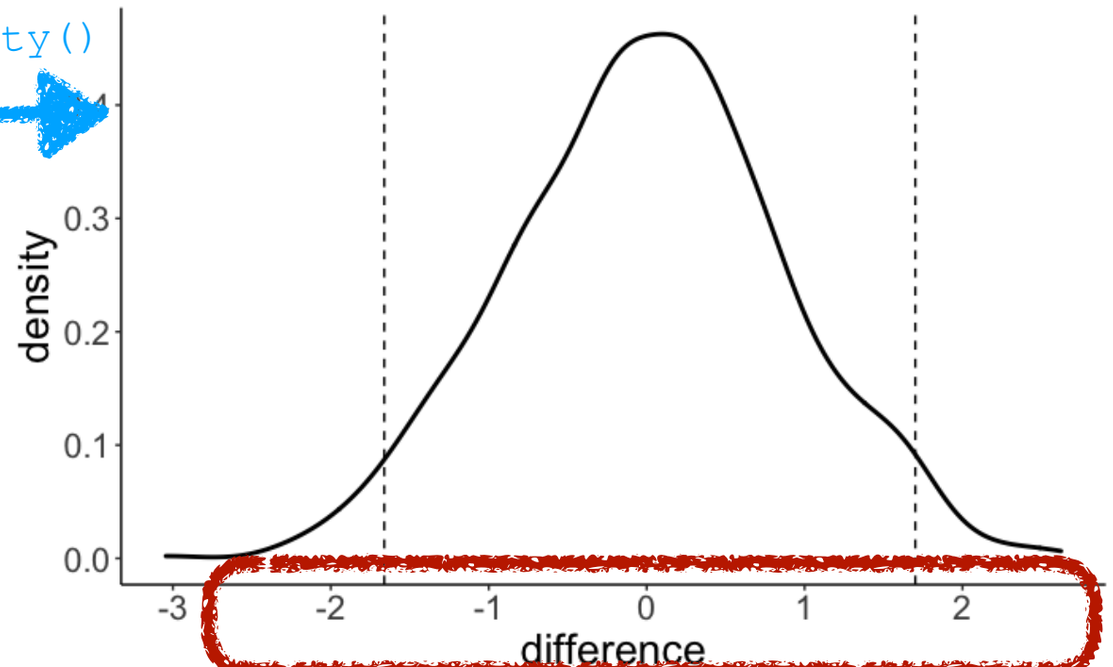

```r
set.seed(1)

n_simulations = 1000
sample_size = 1000
mean = 5
sd = 20

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2)
```

simulation	sample1	sample2	difference
1	4.77	4.77	0.00
2	4.67	5.10	-0.43
3	5.31	5.87	-0.56
4	5.33	6.28	-0.94
5	4.60	5.52	-0.92

⋮

geom_density()

What if sample size N = 1000?



two-sample t-test

difference in means (relative to the standard deviation and sample size)

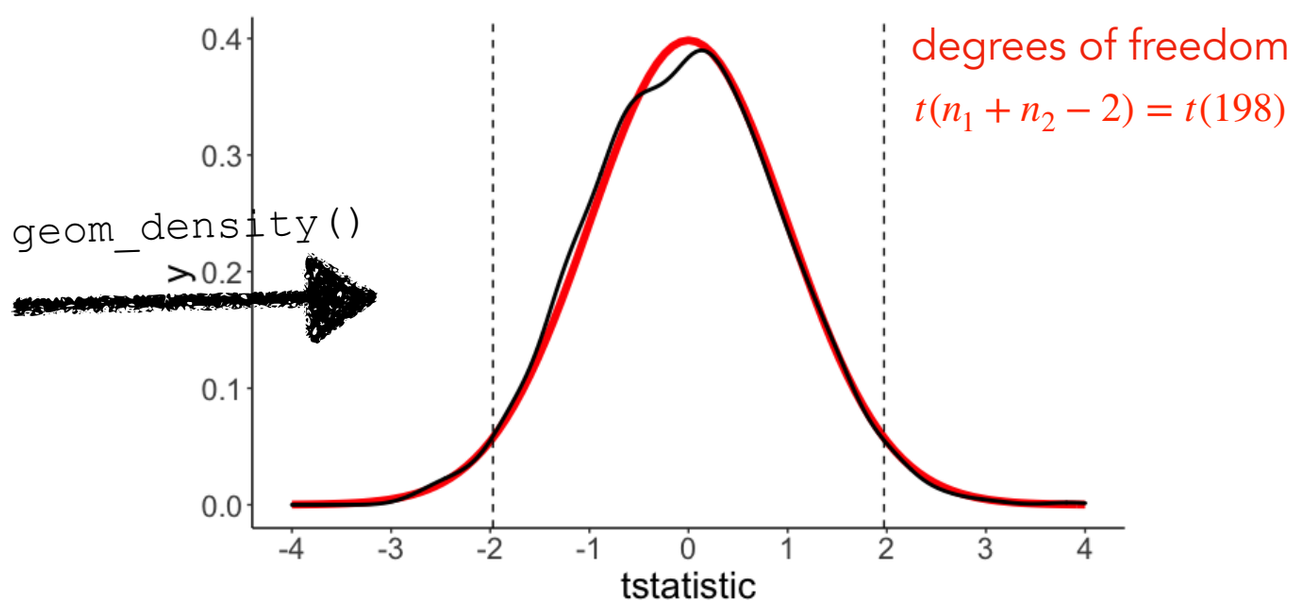
$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2(\frac{1}{n_1} + \frac{1}{n_2})}}$$

pooled sample variance

```
set.seed(1)

n_simulations = 1000
sample_size = 100
mean = 5
sd = 2

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2,
         # assuming the same standard deviation in each sample
         tstatistic = difference / sqrt(sd^2 * (1/sample_size + 1/sample_size)))
```

simulation	sample1	sample2	difference	tstatistic
1	5.22	4.99	0.23	0.80
2	4.92	5.06	-0.13	-0.47
3	5.06	5.40	-0.34	-1.20
4	5.10	4.97	0.13	0.47
5	4.92	4.75	0.17	0.61

geom_density()



degrees of freedom

$t(n_1 + n_2 - 2) = t(198)$

two-sample t-test

difference in means (relative to the standard deviation and sample size)

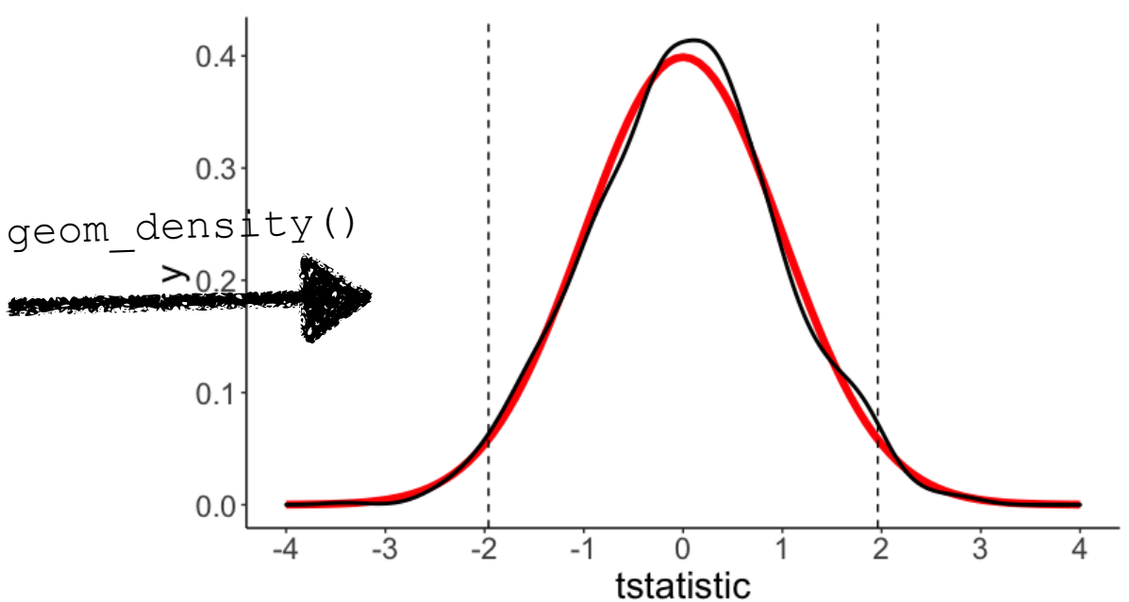
$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s^2(\frac{1}{n_1} + \frac{1}{n_2})}}$$

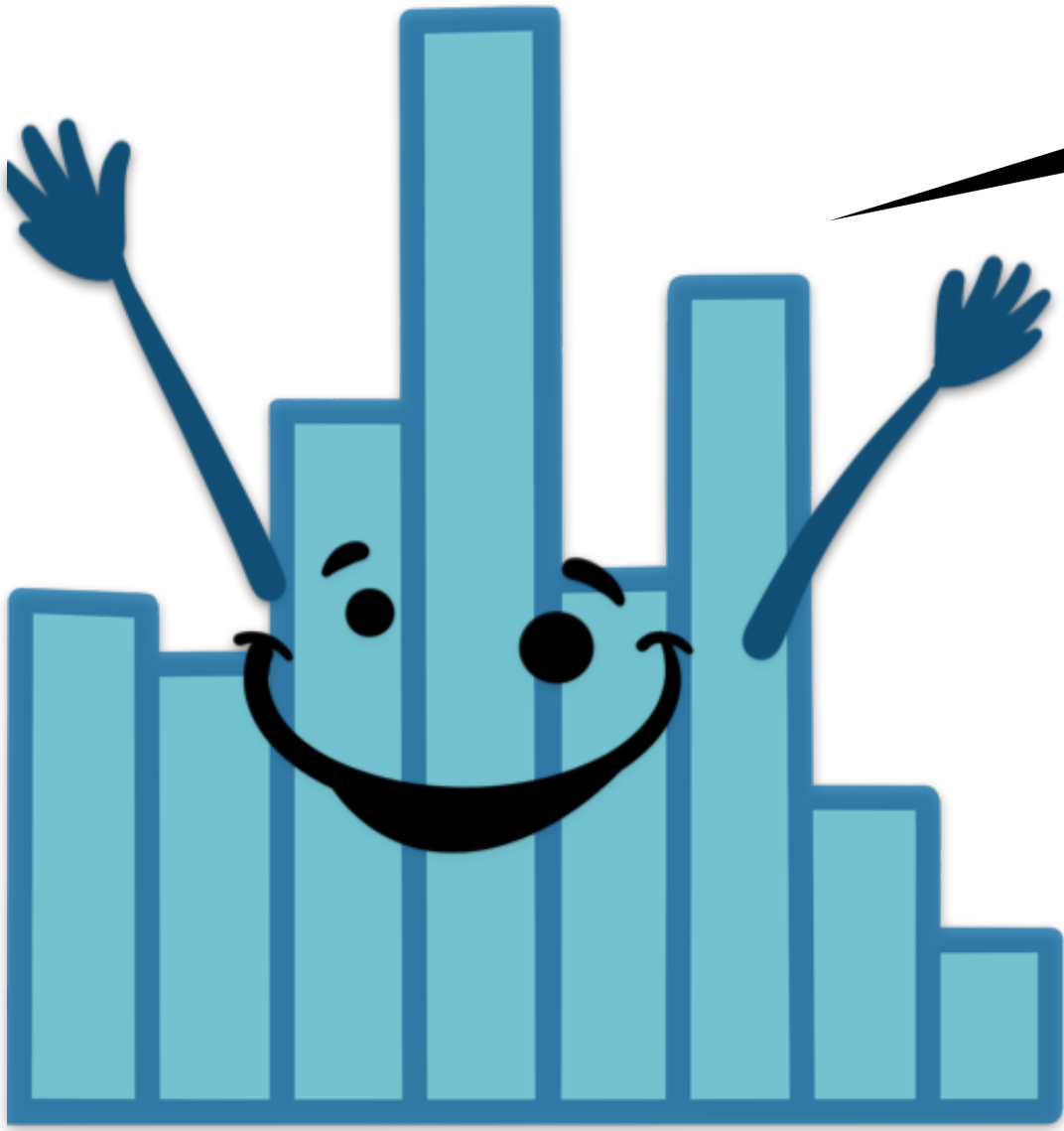
pooled sample variance

```r
set.seed(1)

n_simulations = 1000
sample_size = 1000
mean = 5
sd = 20

fun.normal_sample_mean = function(sample_size, mean, sd){
  rnorm(n = sample_size, mean = mean, sd = sd) %>%
    mean()
}

df.ttest = tibble(simulation = 1:n_simulations) %>%
  mutate(sample1 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd)),
         sample2 = replicate(n = n_simulations,
                             expr = fun.normal_sample_mean(sample_size, mean, sd))) %>%
  mutate(difference = sample1 - sample2,
         # assuming the same standard deviation in each sample
         tstatistic = difference / sqrt(sd^2 * (1/sample_size + 1/sample_size)))
```

simulation	sample1	sample2	difference	tstatistic
1	7.18	4.93	2.25	0.80
2	4.24	5.57	-1.32	-0.47
3	5.59	8.99	-3.40	-1.20
4	6.03	4.71	1.32	0.47
5	4.22	2.48	1.73	0.61

geom_density()



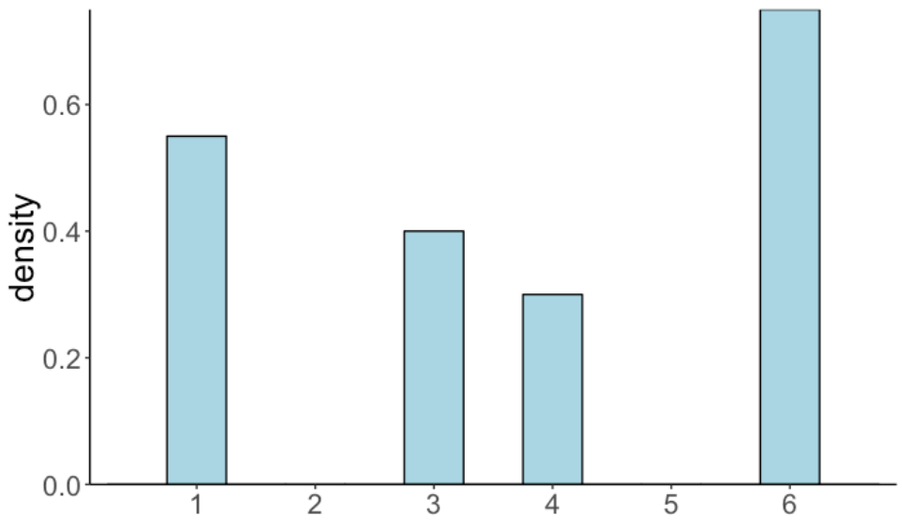


01:00

stretch break!

Confidence intervals

our sample

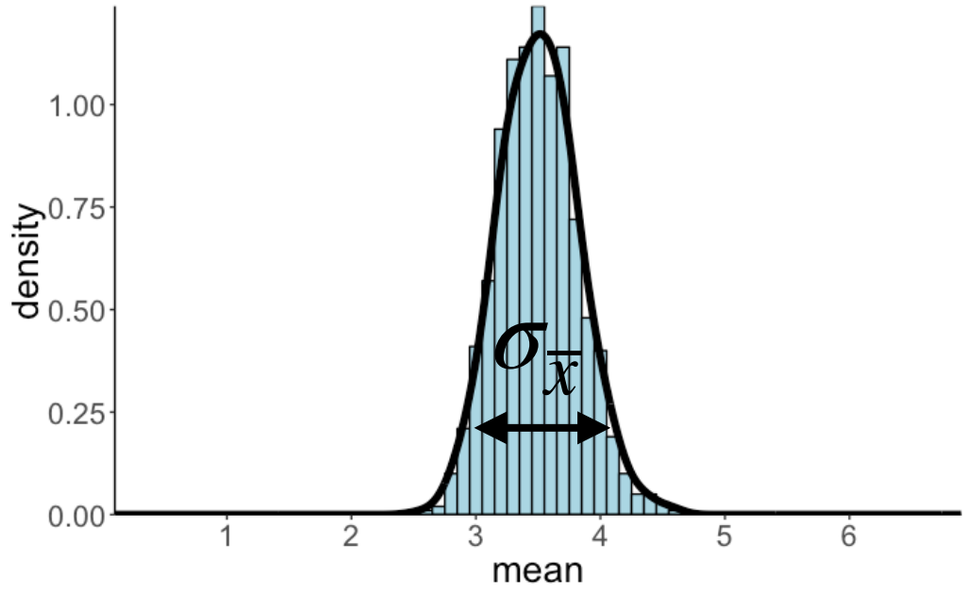


standard deviation

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

gives a sense for how well the mean captures the data

sampling distribution



standard error of the mean

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

$\sigma_{\bar{x}}$ 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

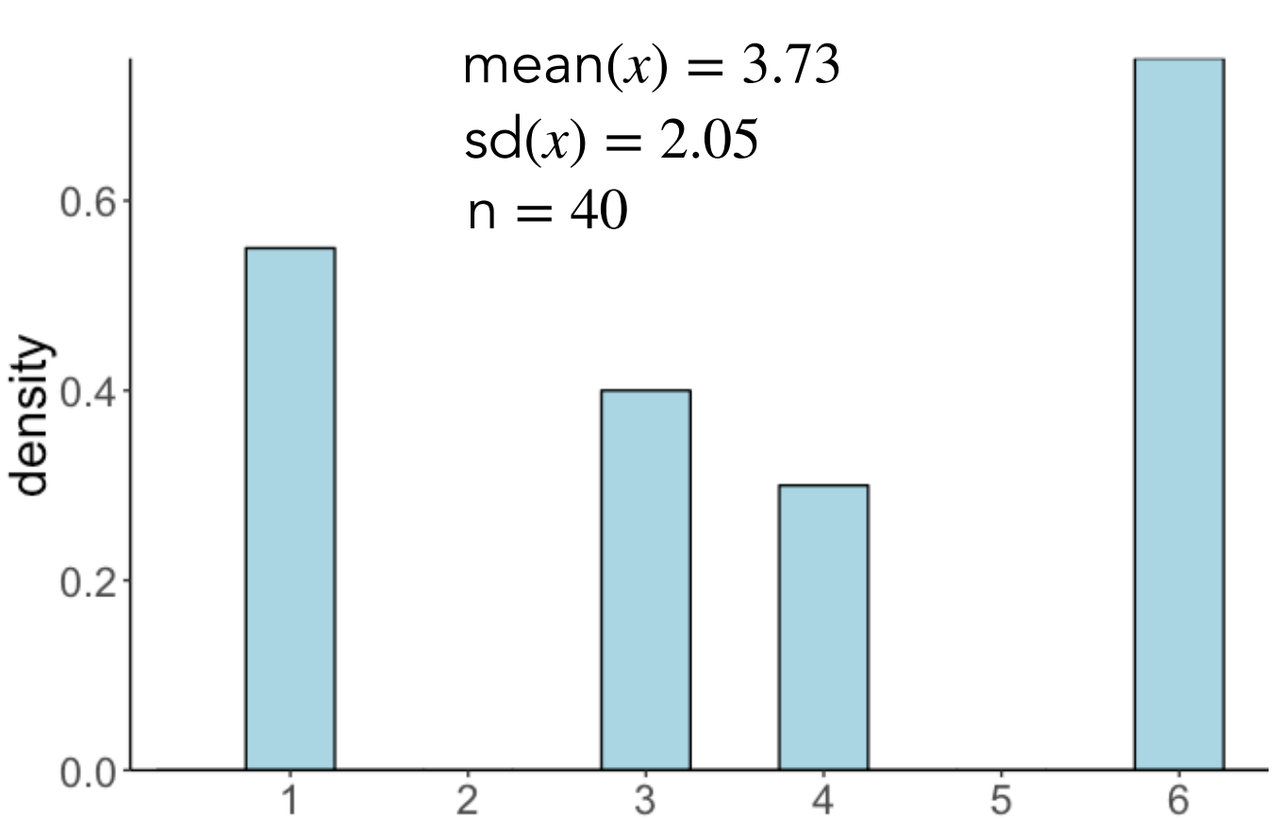
Goal: Estimate the mean of the population distribution μ

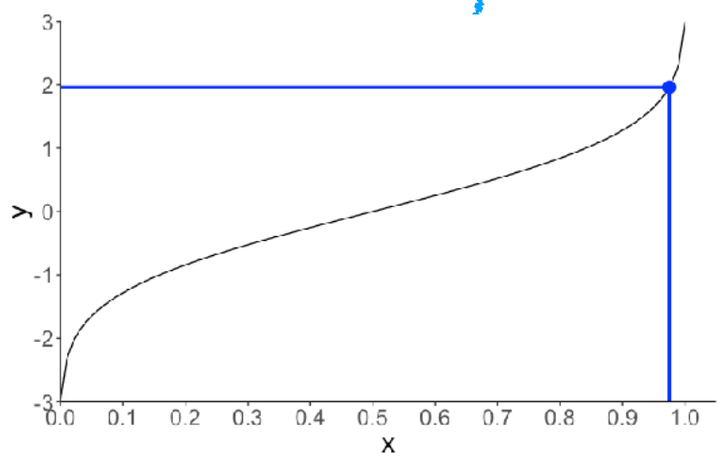
Confidence interval = point estimate ± critical value

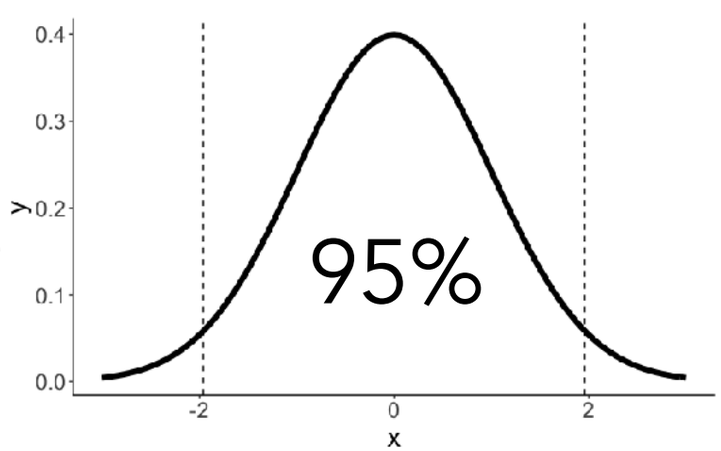
`mean(x)`

`qnorm(.975)×SE`

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







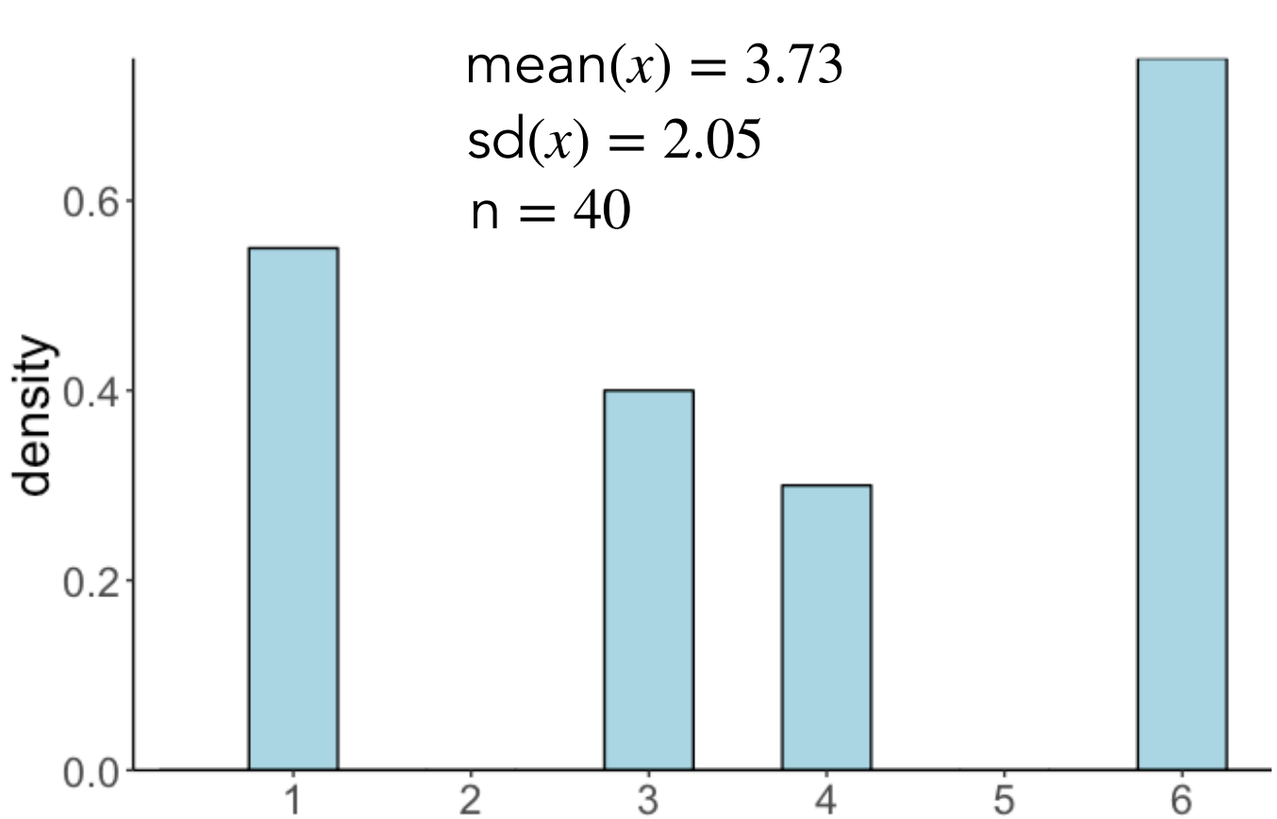
Confidence interval

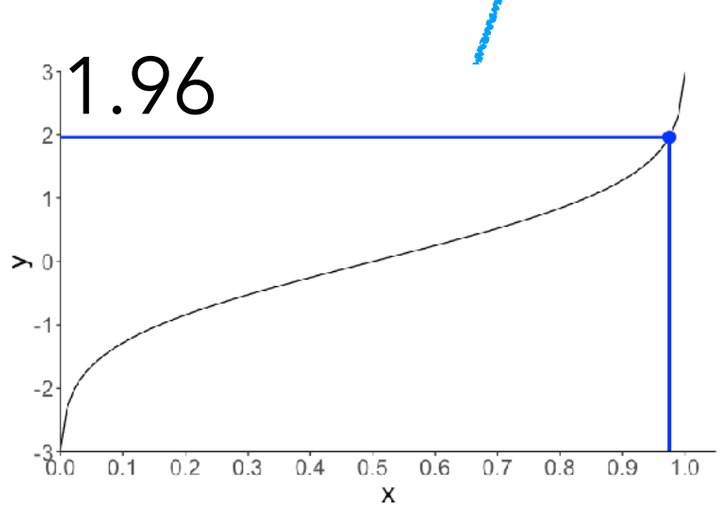
Goal: Estimate the mean of the population distribution μ

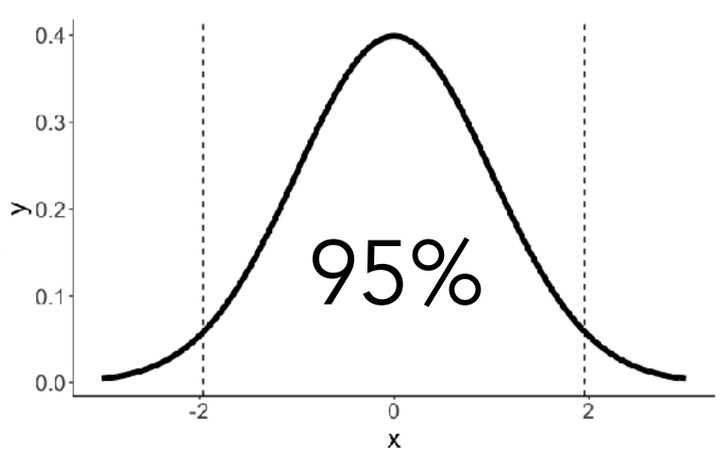
Confidence interval $= 3.73 \pm 0.63$

Confidence interval = point estimate ± critical value

`mean(x)` `qnorm(.975)×SE`

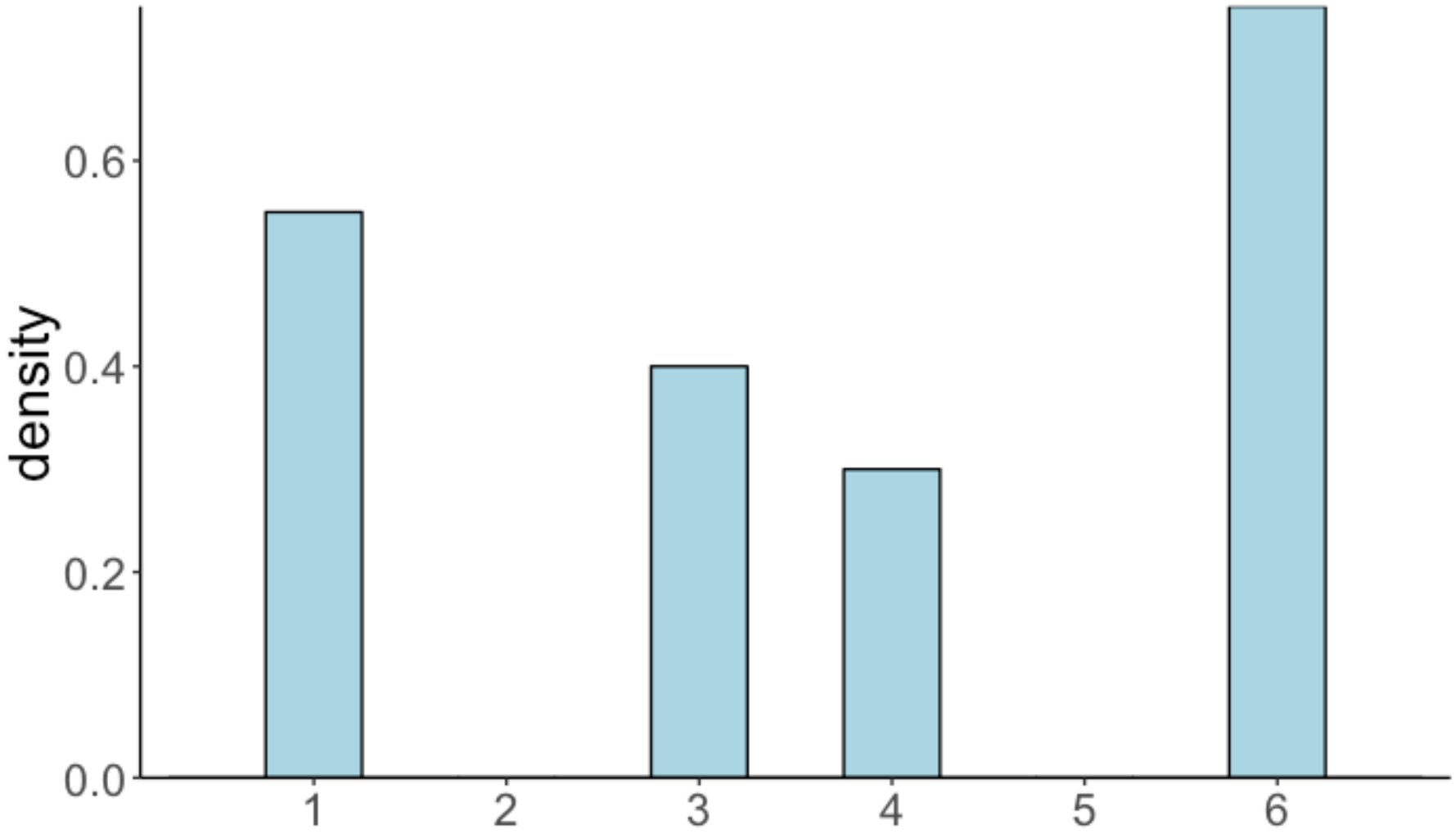






$$\sigma_{\bar{x}} = \frac{2.05}{\sqrt{40}}$$

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.

30%

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

✓10%

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

40%

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

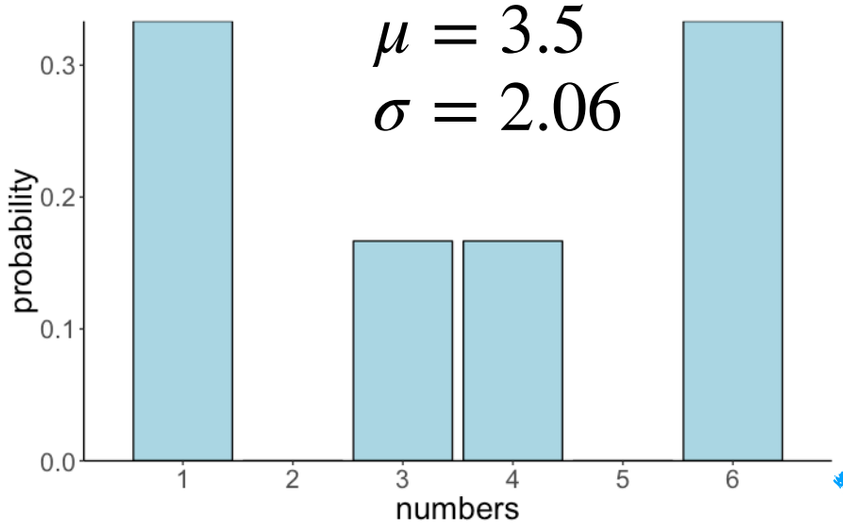
20%

0% 10% 20% 30% 40%

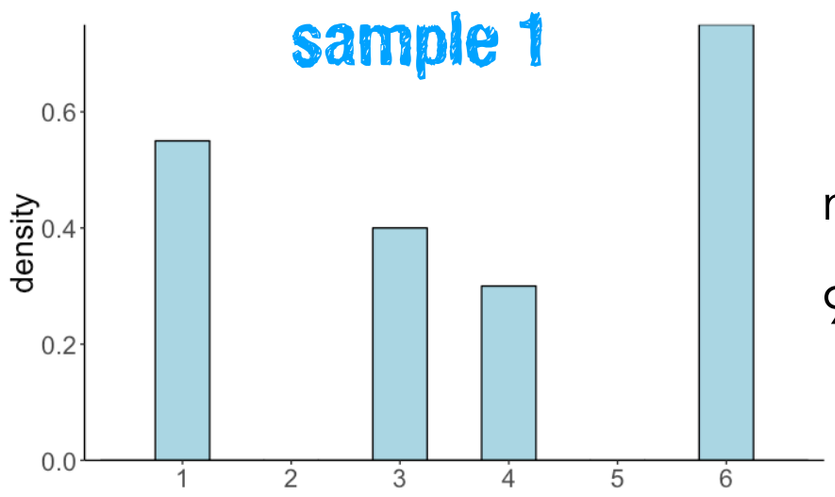
What is a confidence interval? Your answers

- A range that represents the plausible values for the parameter from the data.
- it's a range that provides certainty of my results
- The probability that a particular range of values contains the true mean of the population.
- Your confidence to a certain degree (typically it's 95%) that you can replicate the findings if you were to do the experiment again
- If you sampled many many times from the same underlying distribution (which you don't actually have access to), X% of the time the sample mean would be in the given range.
- ...

Confidence interval

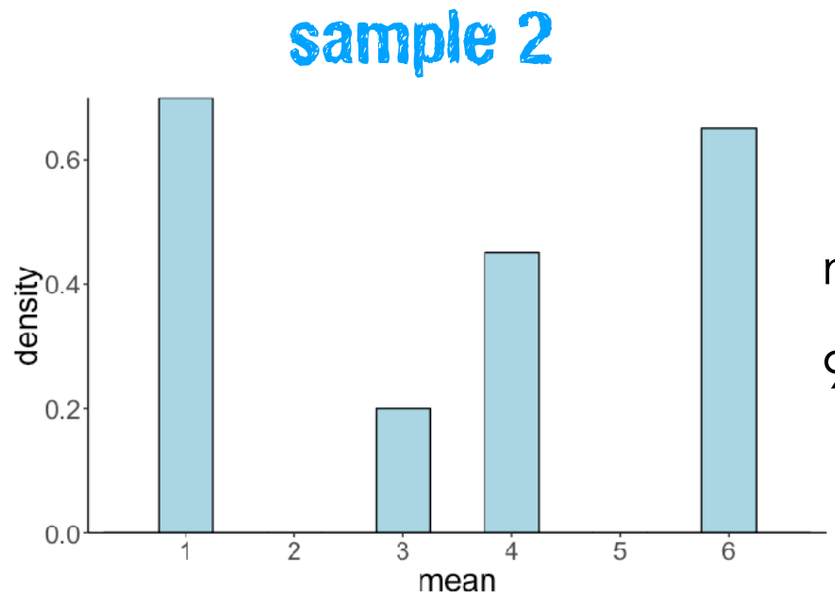


population distribution



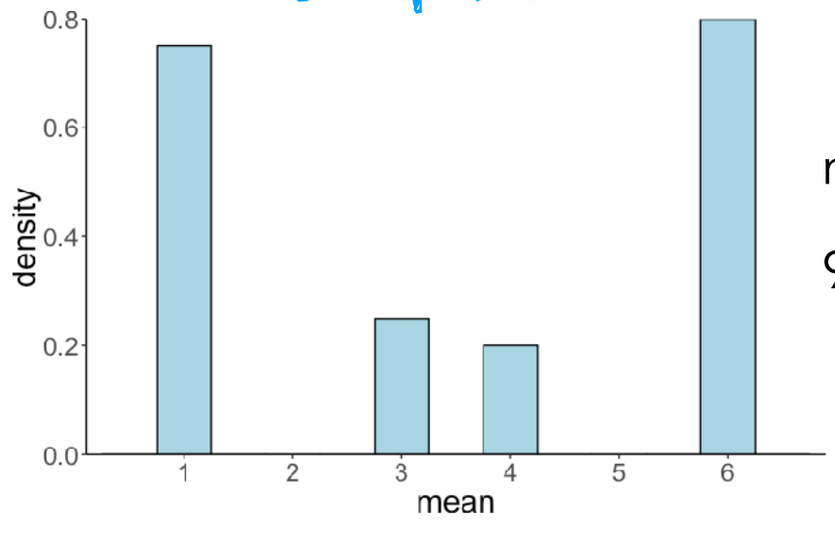
mean = 4.04

95% CI = [3.24, 4.83]



mean = 3.52

95% CI = [2.66, 4.37]



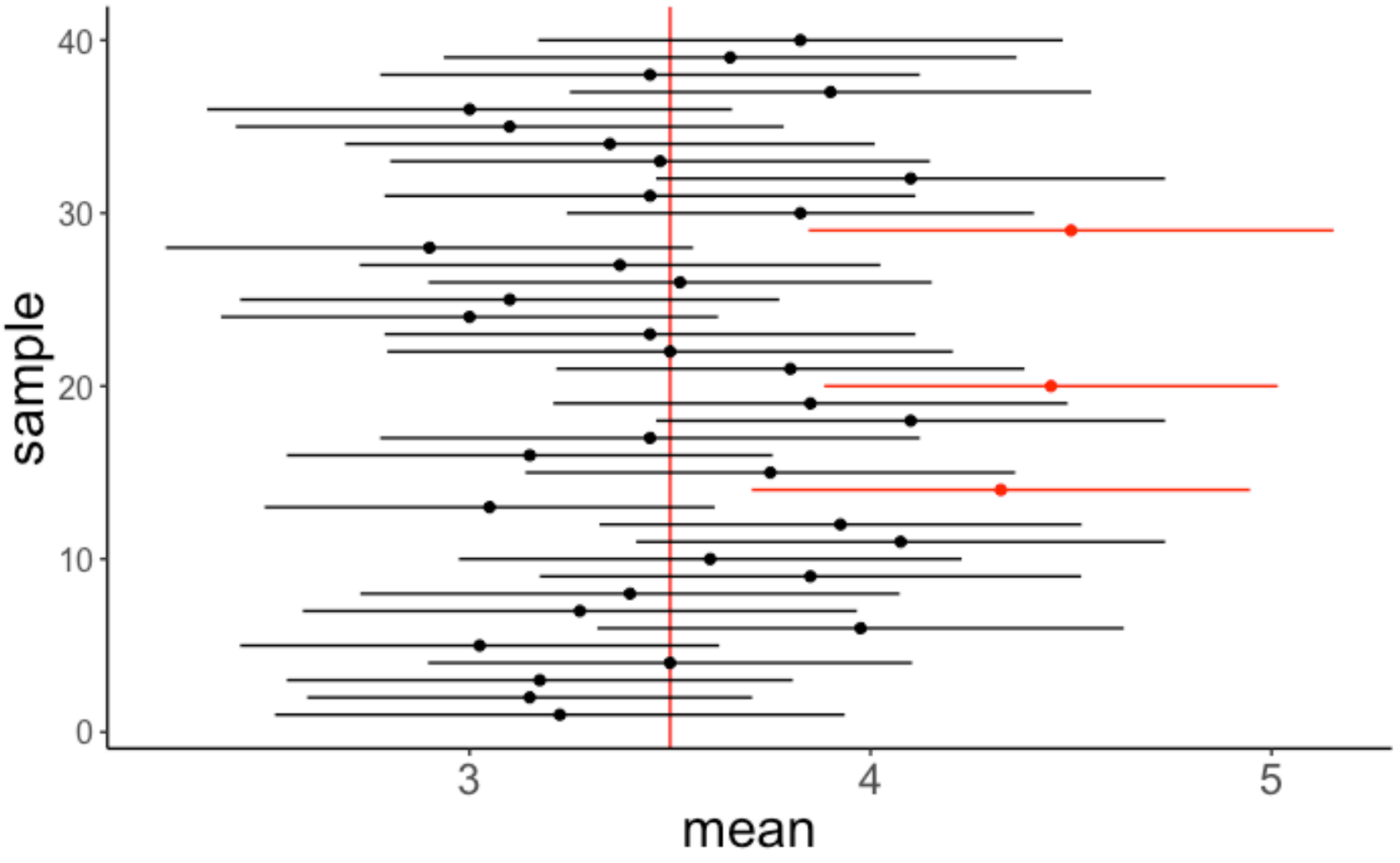
mean = 3.44

95% CI = [2.64, 4.24]

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.

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.

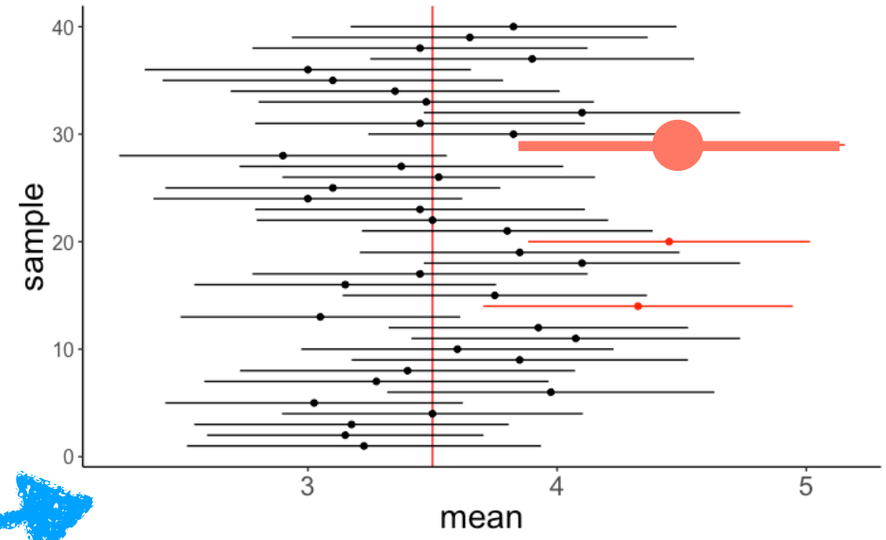
The sample means of 95% of the random samples of size 40 will be in this interval.

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

It either is in this interval or isn't.

correct

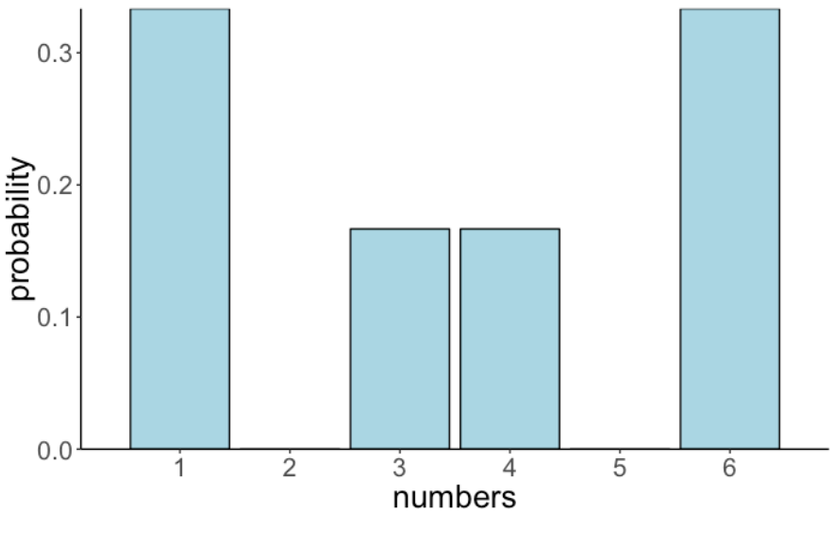
incorrect



We know what the sample mean is.

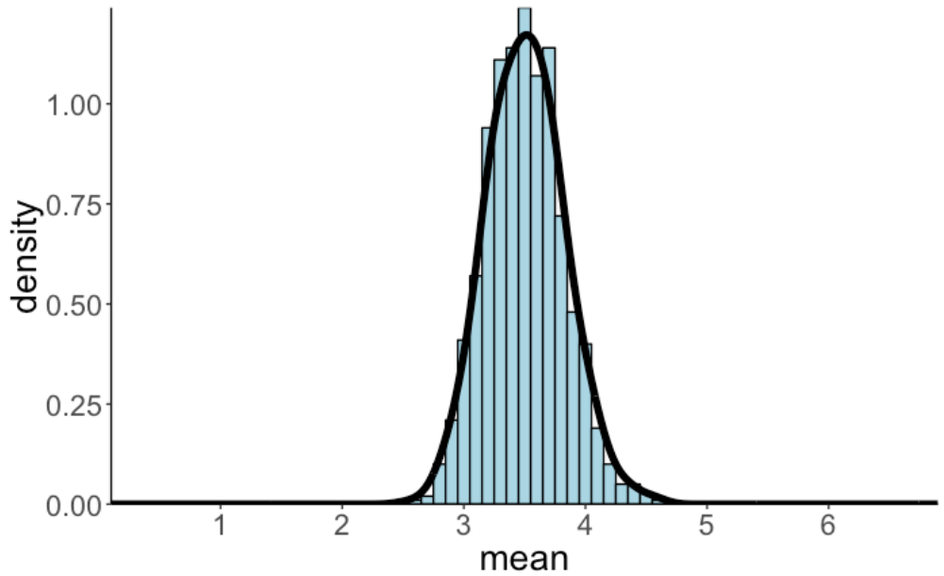
Bootstrapping

Bootstrap



population distribution

repeated sampling



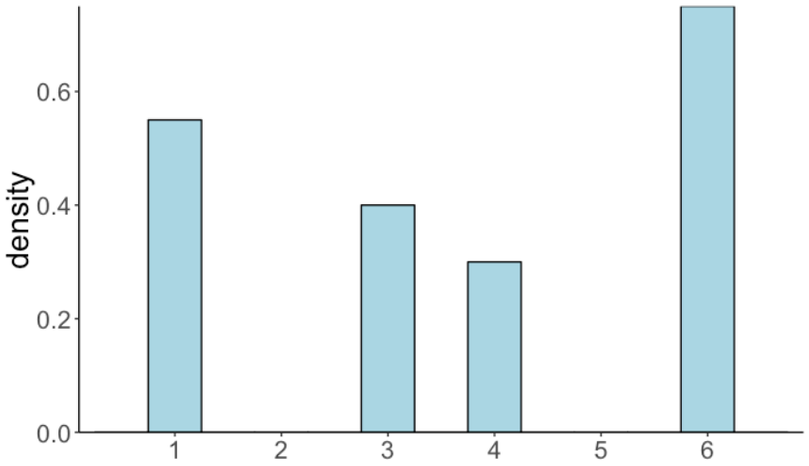
sampling distribution

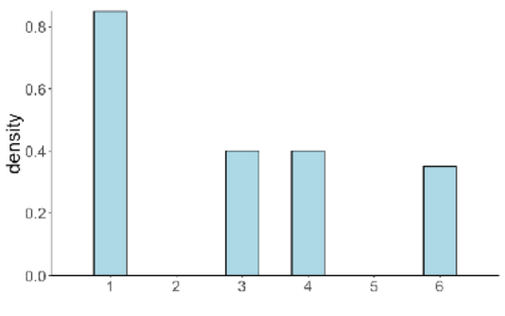
but we don't know the population distribution!

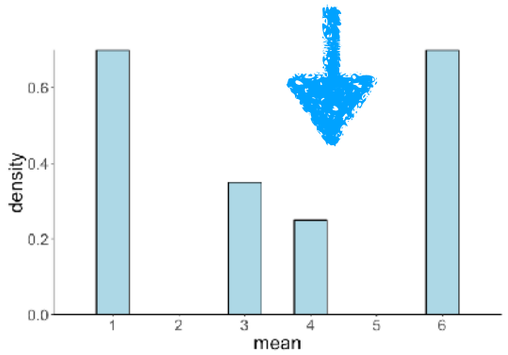
Bootstrap

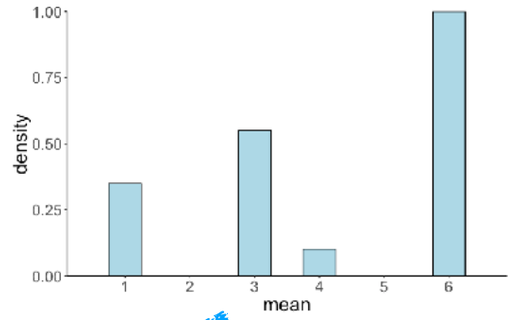
all we have is our sample

repeated sampling with replacement





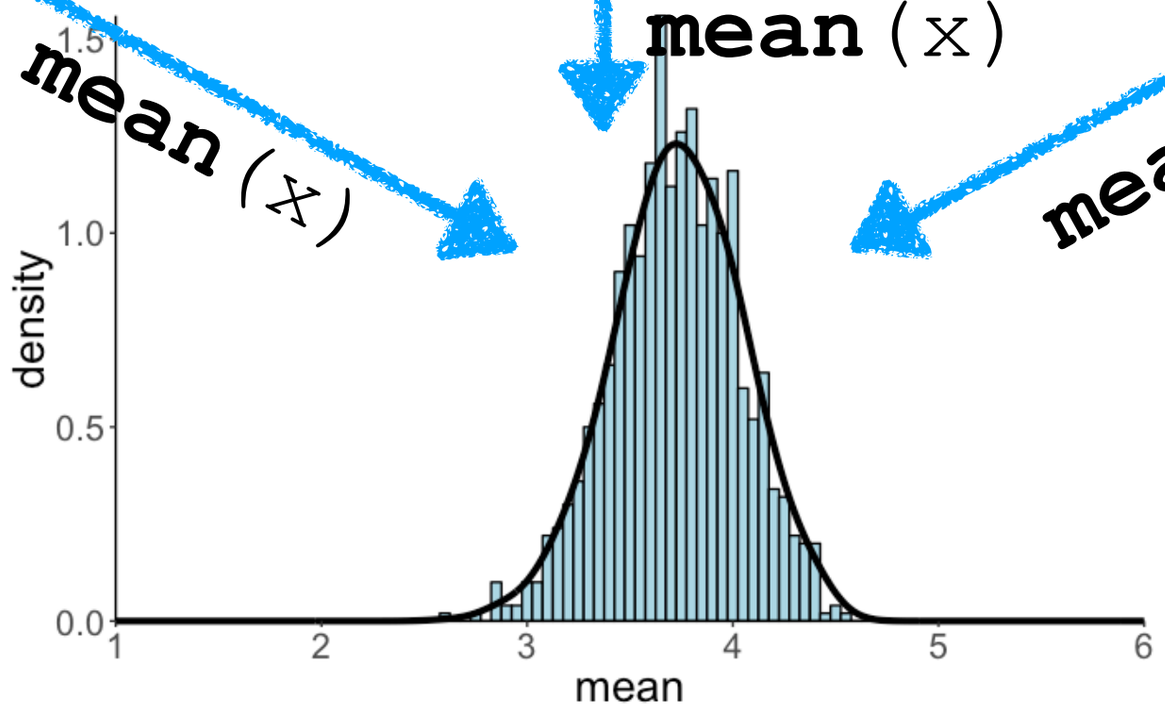




mean(x)

mean(x)

mean(x)



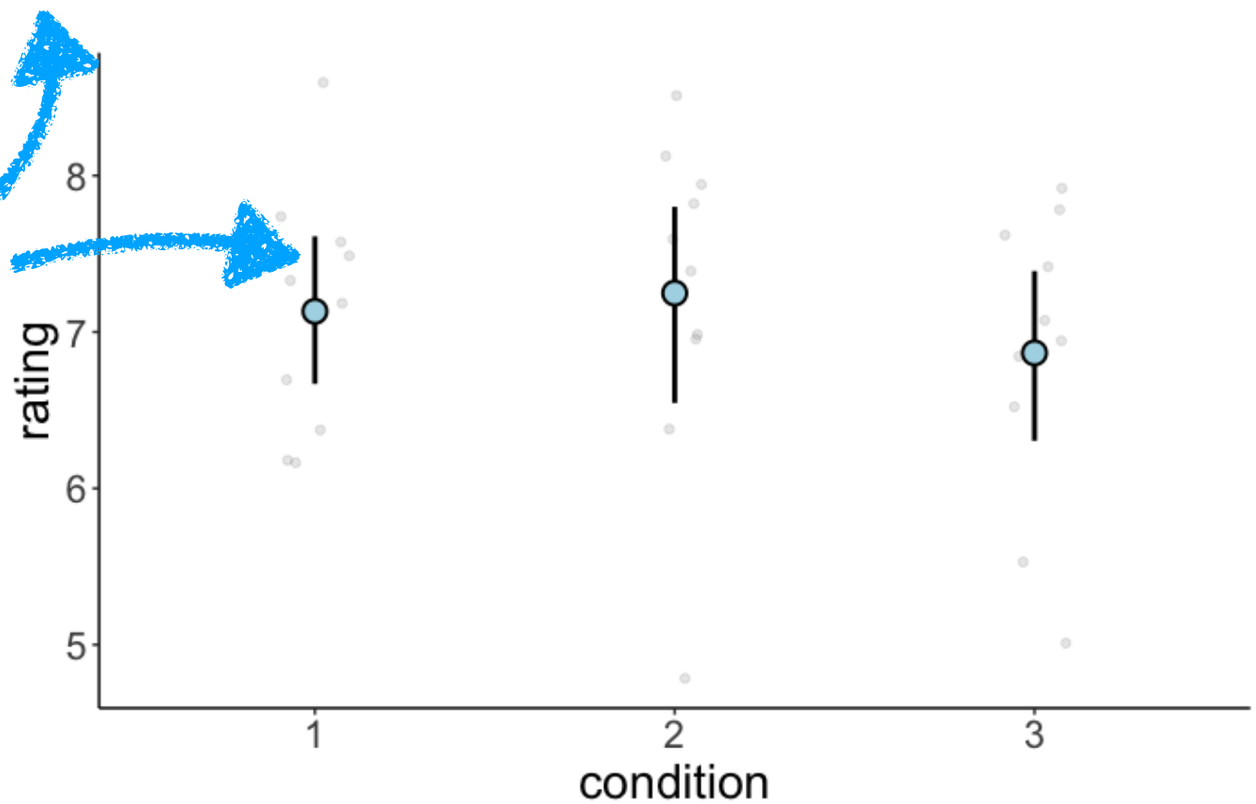
sampling distribution

mean_cl_boot() explained

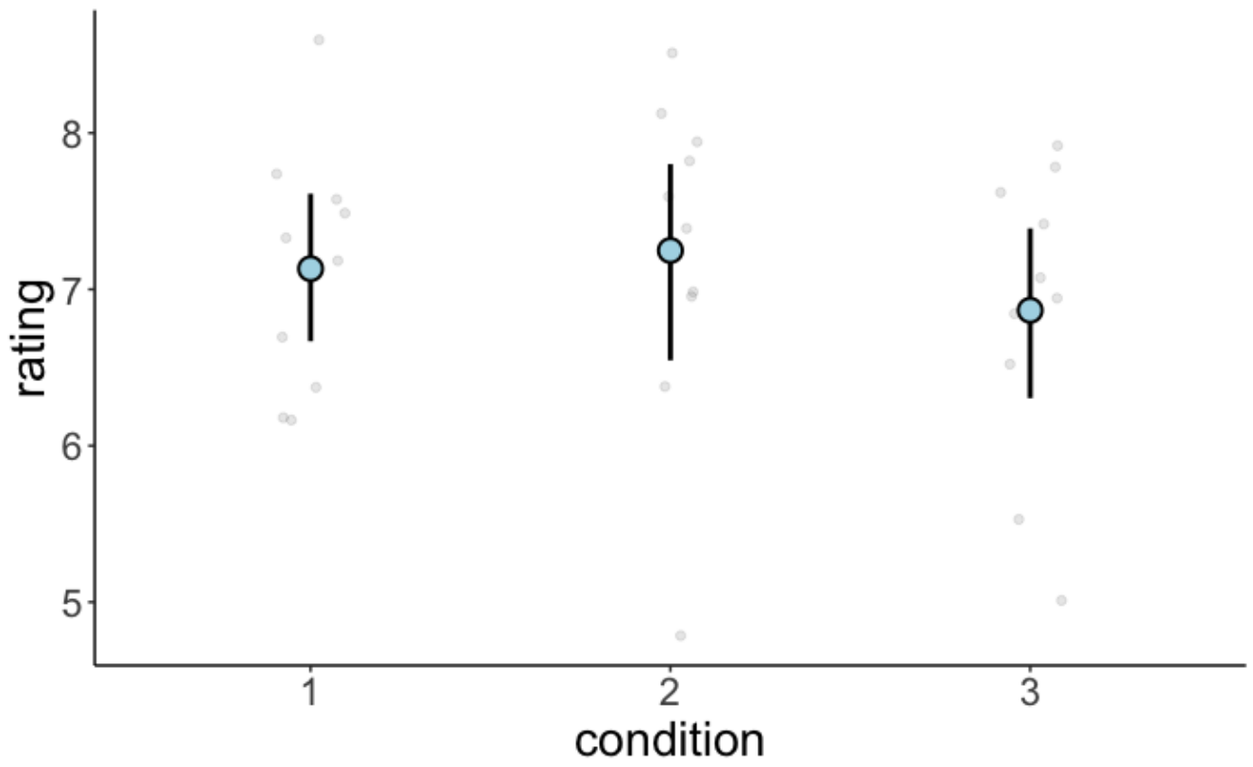
```r
set.seed(1)

n = 10 # sample size per group
k = 3 # number of groups

df.data = tibble(participant = 1:(n*k),
                 condition = as.factor(rep(1:k, each = n)),
                 rating = rnorm(n*k, mean = 7, sd = 1))

ggplot(data = df.data,
       mapping = aes(x = condition,
                     y = rating)) +
  geom_point(alpha = 0.1,
             position = position_jitter(width = 0.1, height = 0)) +
  stat_summary(fun.data = "mean_cl_boot",
               shape = 21,
               size = 1,
               fill = "lightblue")
```

participant	condition	rating
1	1	6.37
2	1	7.18
3	1	6.16
4	1	8.60
5	1	7.33

what is this magic?



`mean_cl_boot()` explained

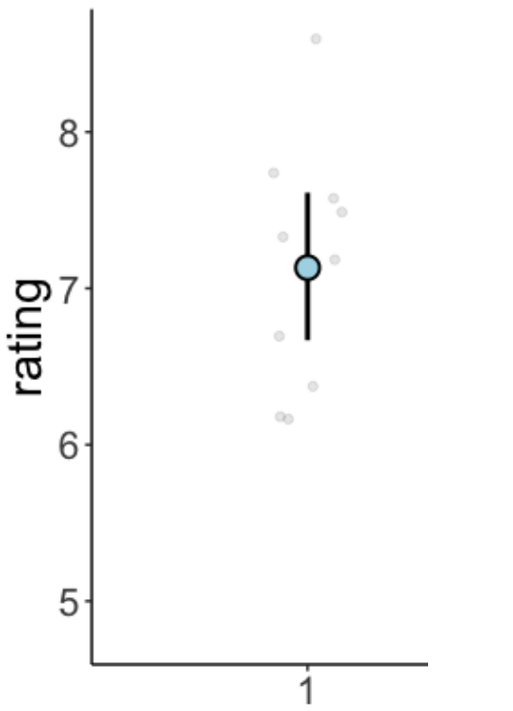



```
ggplot_build(p)
```

nice function for peeking behind the scenes of ggplots

x	group	y	ymin	ymax	PANEL	flipped_aes	colour	size	linetype	shape	fill	alpha	stroke
1	1	7.13	6.70	7.59	1	FALSE	black	1	1	21	lightblue	NA	1
2	2	7.25	6.54	7.83	1	FALSE	black	1	1	21	lightblue	NA	1
3	3	6.87	6.26	7.39	1	FALSE	black	1	1	21	lightblue	NA	1

`mean_cl_boot()` explained



x	group	y	ymin	ymax
1	1	7.13	6.70	7.59

```
1  set.seed(1)
2
3  df.condition1 = df.data %>%
4    filter(condition == 1)
5
6  fun.sample_with_replacement = function(df){
7    df %>%
8      slice_sample(n = nrow(df),
9                   replace = T) %>%
10     summarize(mean = mean(rating)) %>%
11     pull(mean)
12 }
```

this is what makes this a bootstrap

original

participant	condition	rating
1	1	6.37
2	1	7.18
3	1	6.16
4	1	8.60
5	1	7.33
6	1	6.18
7	1	7.49
8	1	7.74
9	1	7.58
10	1	6.69

bootstrap 1

participant	condition	rating
9	1	7.58
3	1	6.16
3	1	6.16
2	1	7.18
3	1	6.16
7	1	7.49
4	1	8.60
8	1	7.74
5	1	7.33

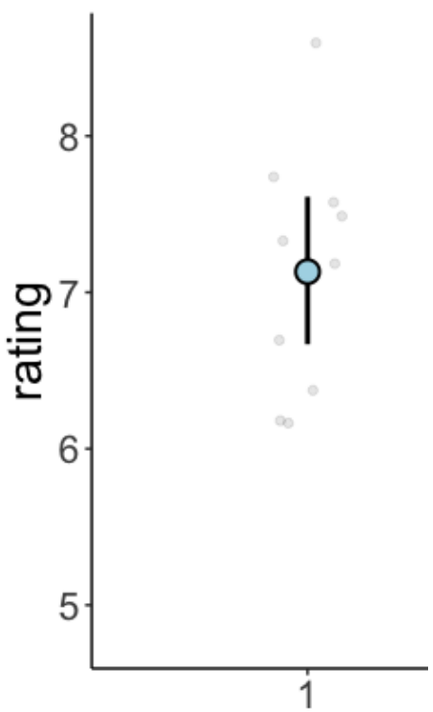
bootstrap 2

participant	condition	rating
1	1	6.37
1	1	6.37
6	1	6.18
1	1	6.37
4	1	8.60
1	1	6.37
9	1	7.58
4	1	8.60
3	1	6.16

note the repeated observations

...

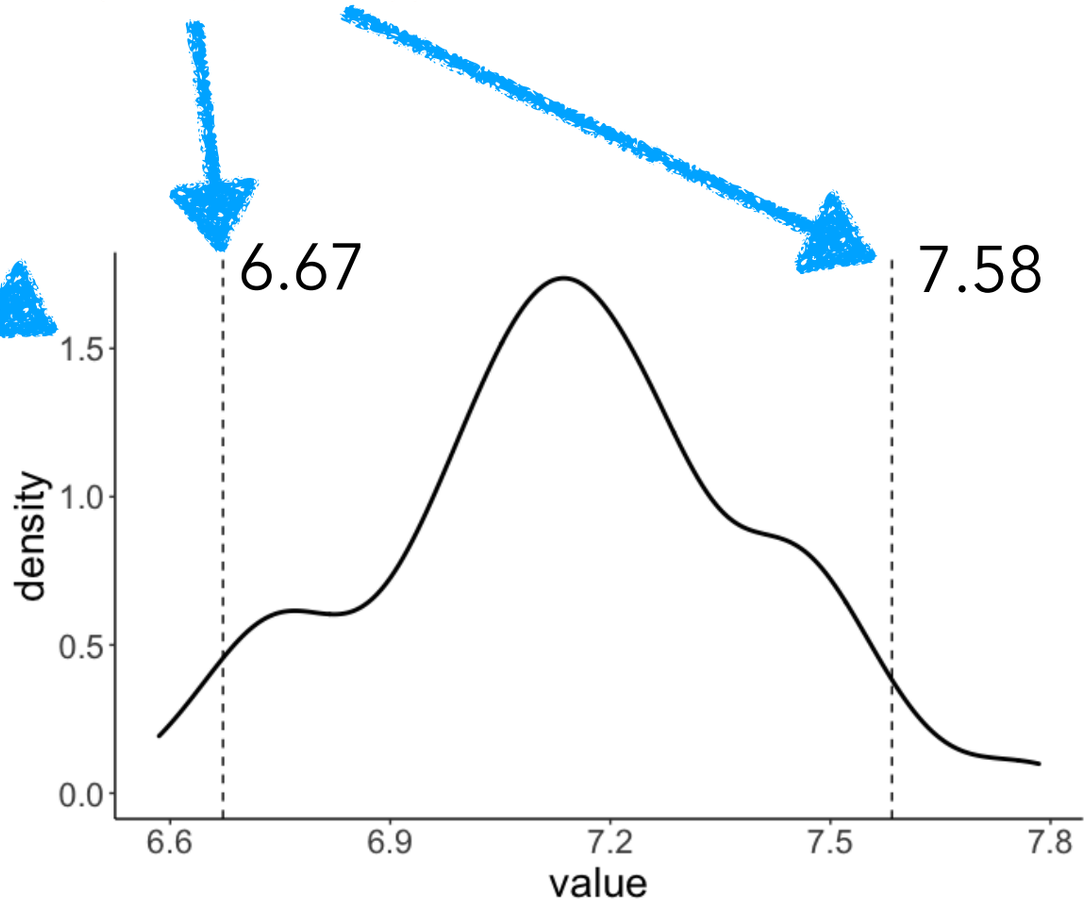
mean_cl_boot() explained




```r
set.seed(1)

df.condition1 = df.data %>%
  filter(condition == 1)

fun.sample_with_replacement = function(df){
  df %>%
    slice_sample(n = nrow(df),
                 replace = T) %>%
    summarize(mean = mean(rating)) %>%
    pull(mean)
}

bootstraps = replicate(n = 100, fun.sample_with_replacement(df.condition1))

quantile(bootstraps, prob = c(0.025, 0.975))
```

x	group	y	ymin	ymax
1	1	7.13	6.70	7.59

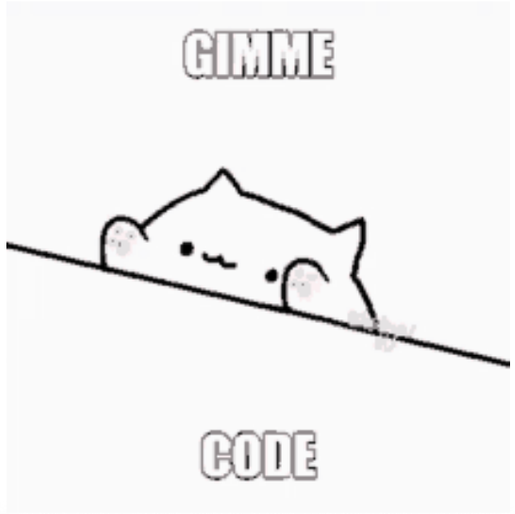
```r
ggplot(data = as_tibble(bootstraps),
       mapping = aes(x = value)) +
  geom_density(size = 1) +
  geom_vline(xintercept = quantile(bootstraps,
                                   probs = c(0.025, 0.975)),
             linetype = 2)
```

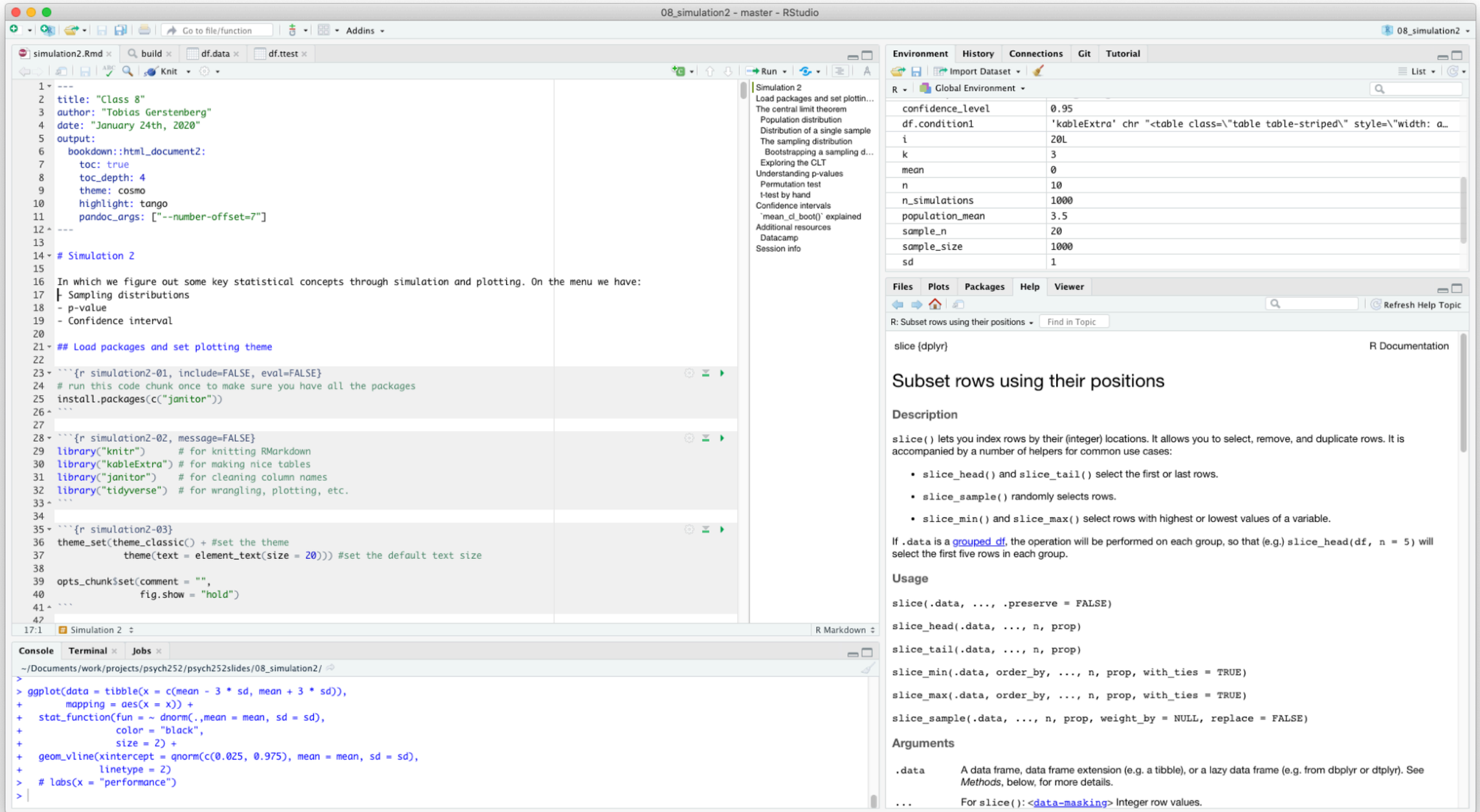


Summary **Revisit and understand key statistical concepts**

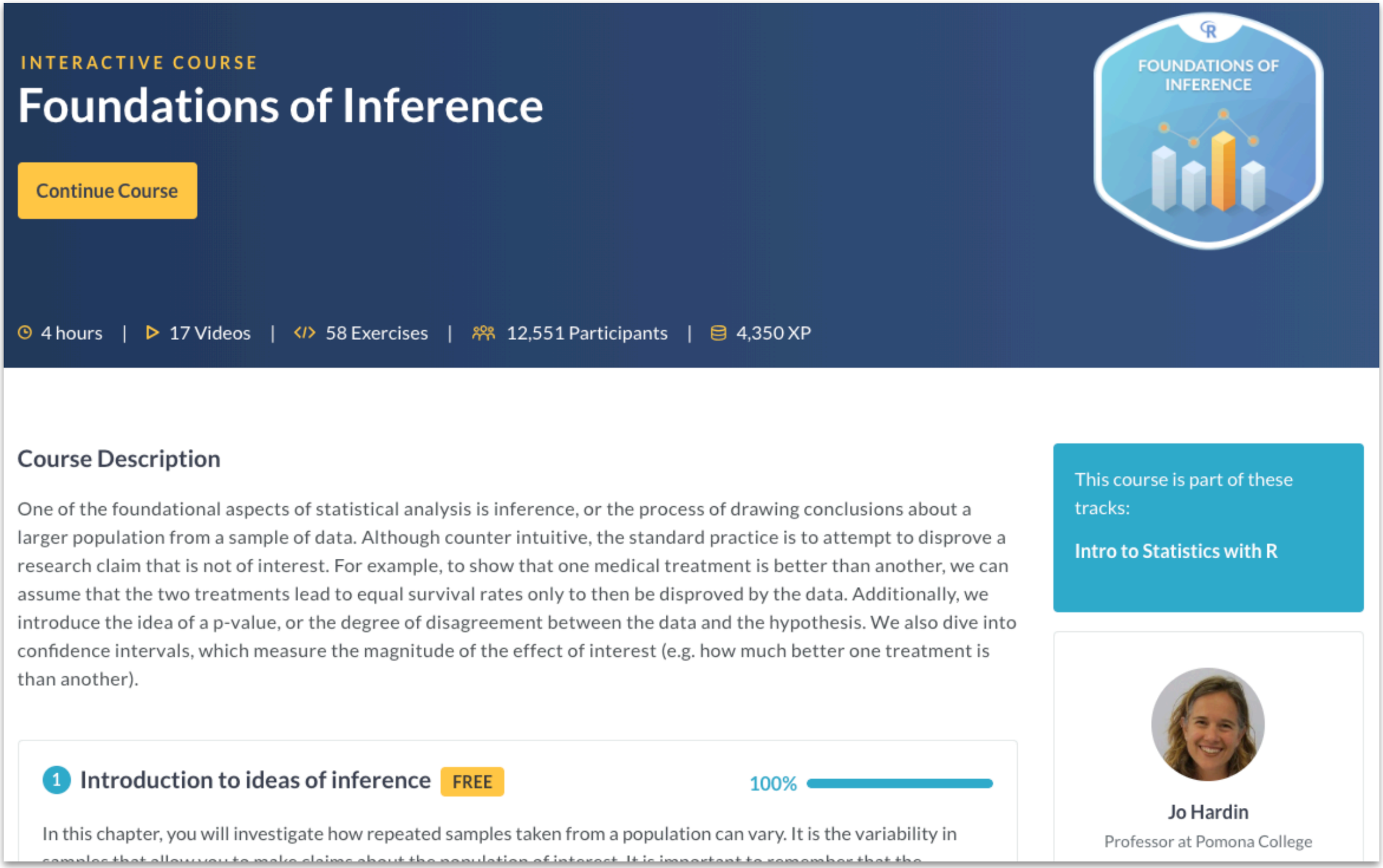
- **Inference in frequentist statistics**
  - goal is to make inference from sample to population
  - we do so via a complicated procedure that involves sampling distributions
- **Sampling distributions**
  - the link between sample and population in frequentist statistics
  - theoretical (or simulated) distribution of a test statistic under the assumption that the H_0 of no difference is true
- **What is a p-value?**
  - the probability of the observed test result (or a more extreme result) assuming that the H_0 is true
- **Confidence interval (of the mean)**
  - "If we were to repeat the experiment over and over, then 95% of the time the confidence intervals contain the true mean."
- **Bootstrapping**
  - a way to generate sampling distributions (by sampling with replacement) without making any assumptions about the underlying population distribution

How to better understand!





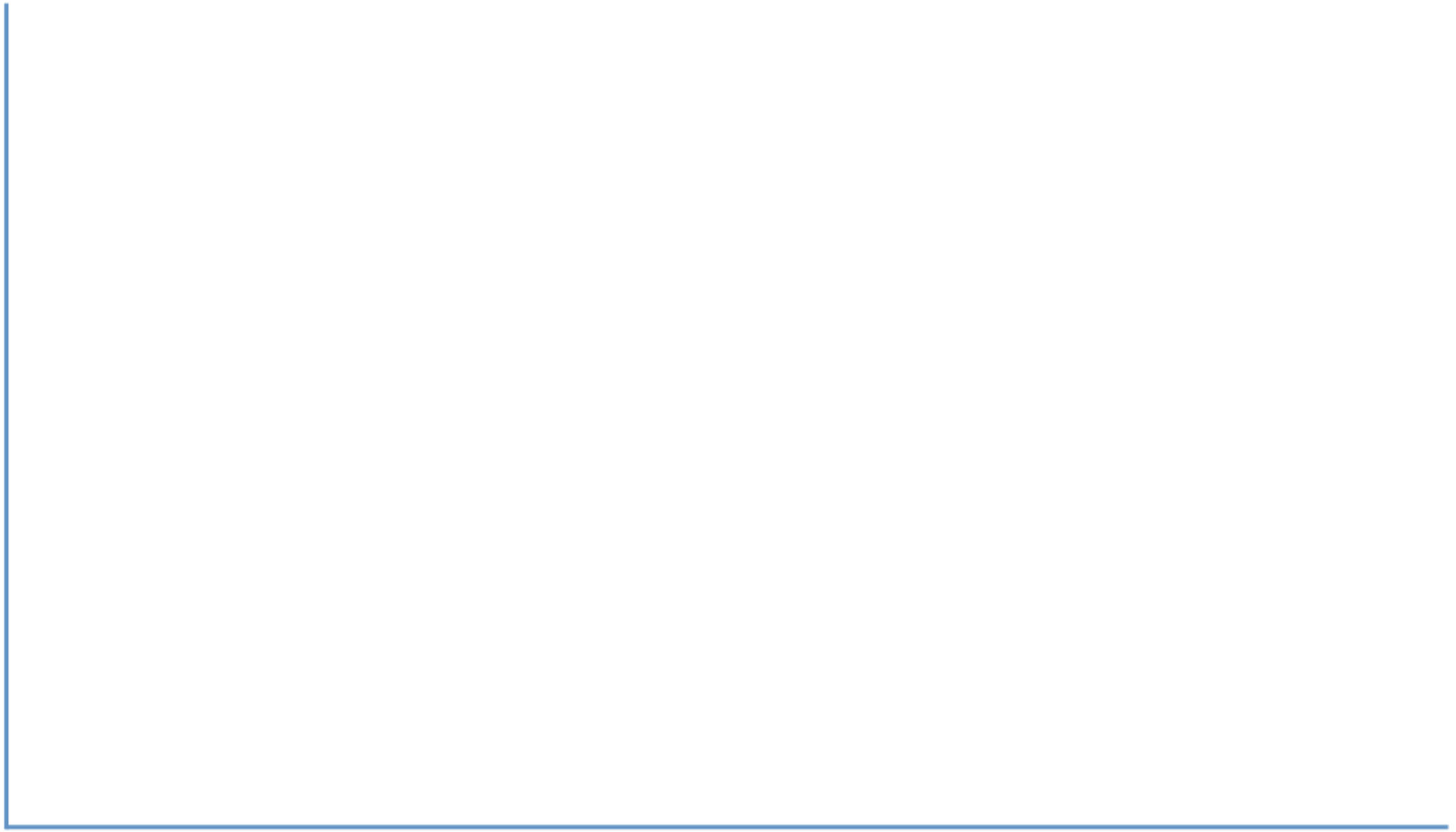
Datacamp



https://www.datacamp.com/courses/foundations-of-inference

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!