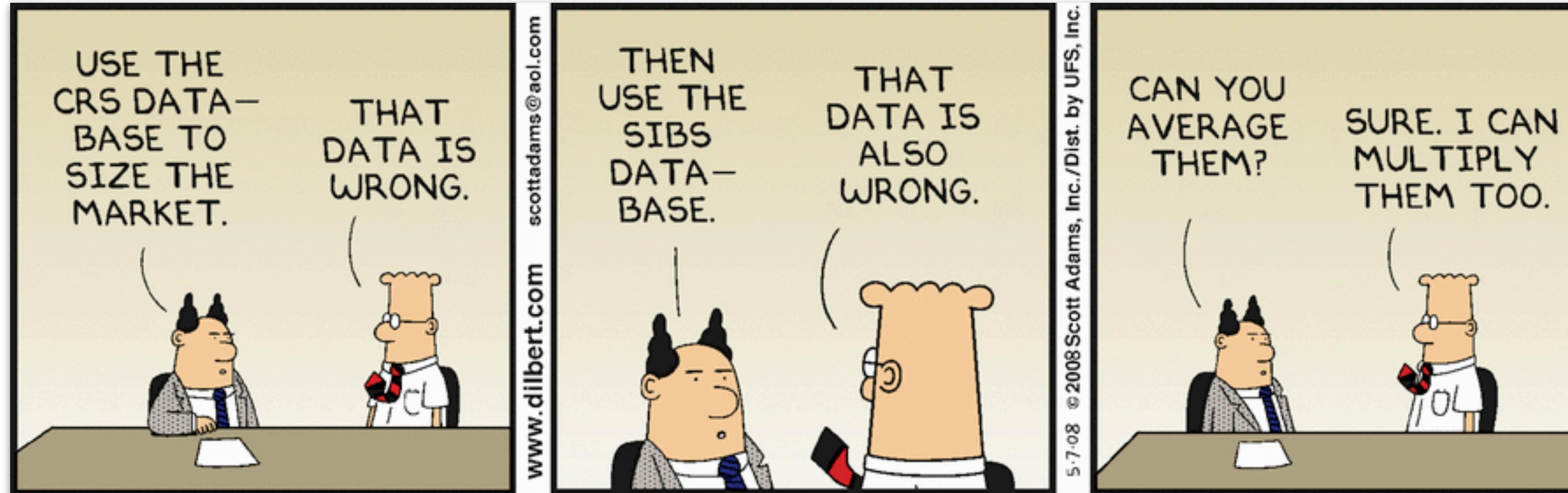


# Simulation 2



01/24/2025

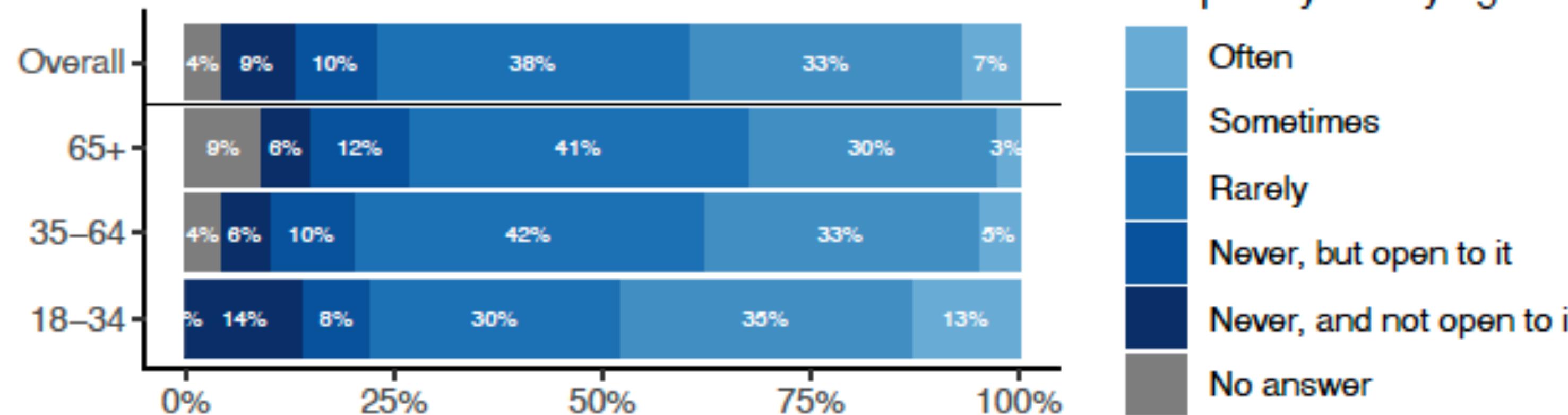
# **Homework 1**

## **Show case**

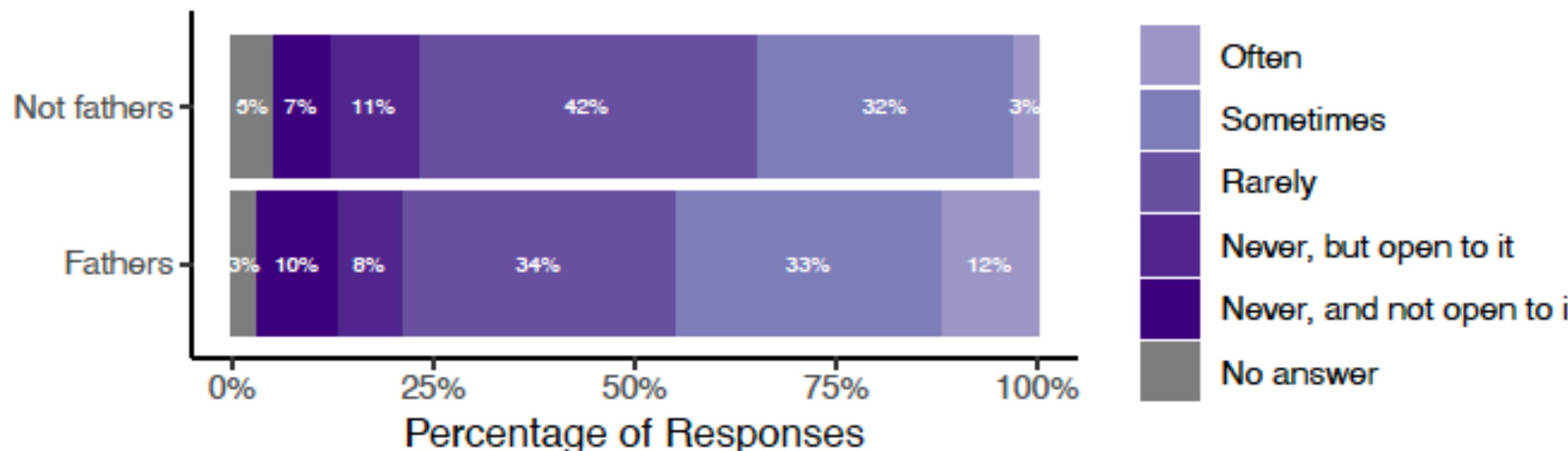
# Micaela Bonilla

## Do Boys Still Not Cry? Yes and No

### By Age Group



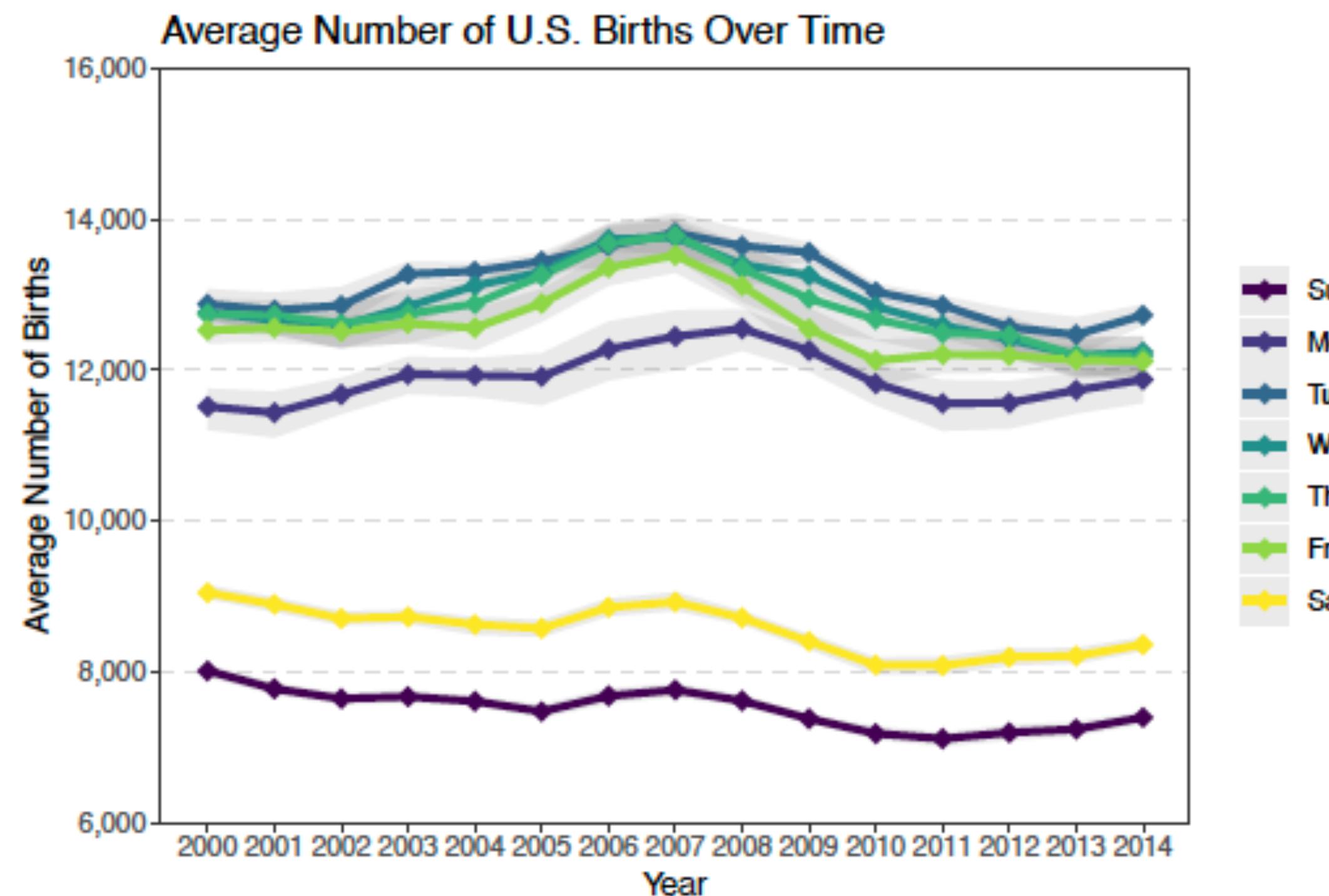
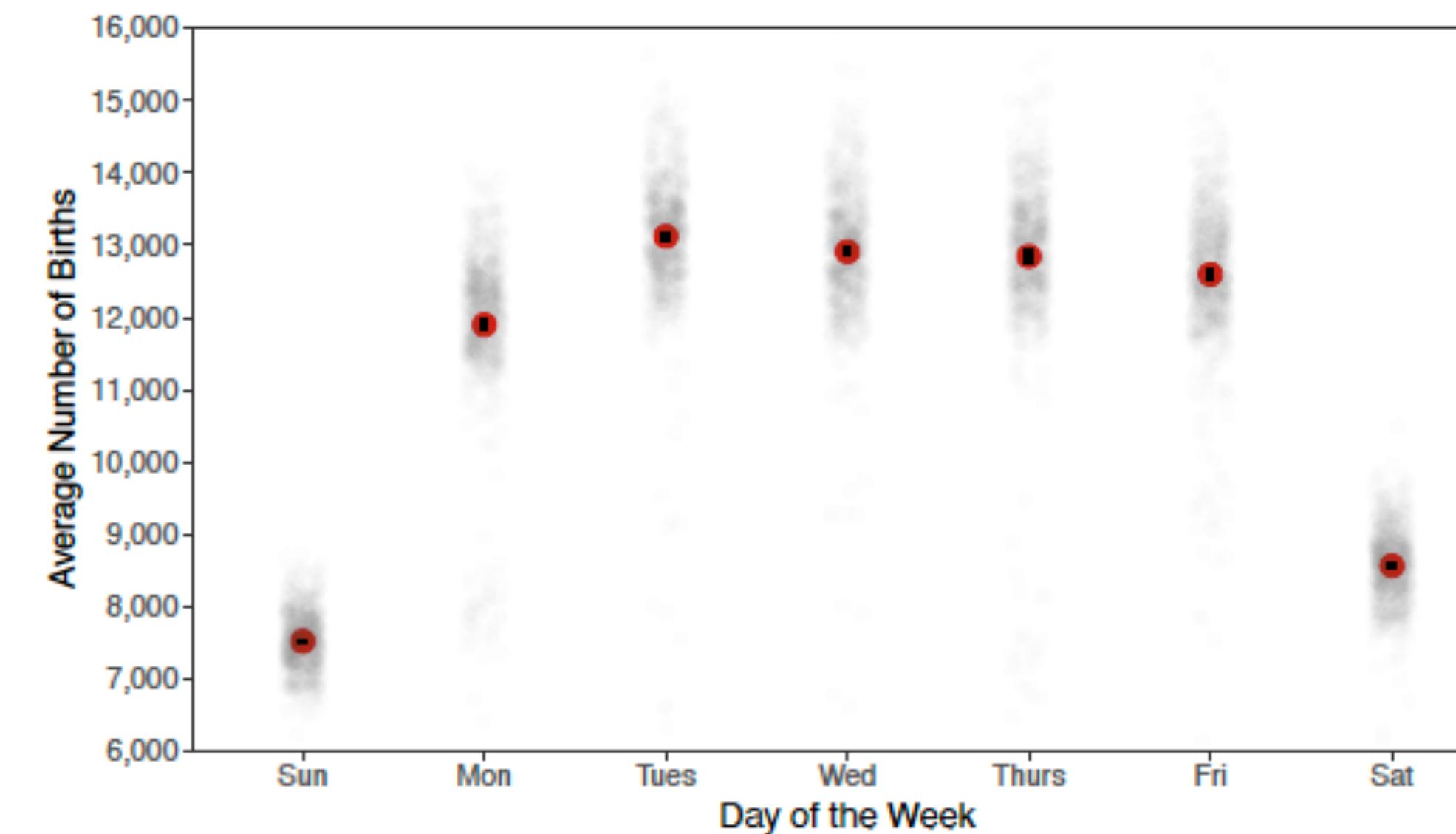
### By Parenthood



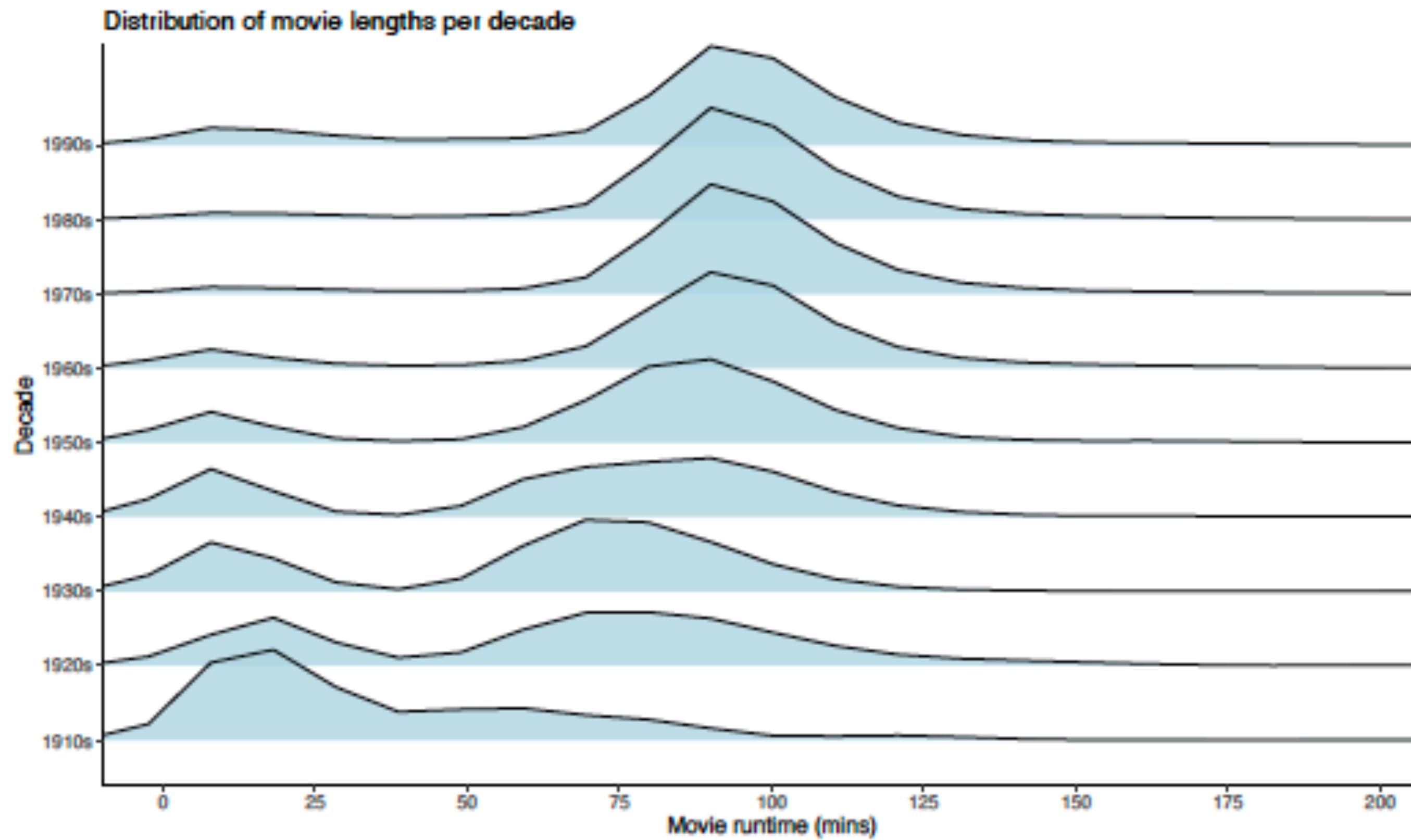
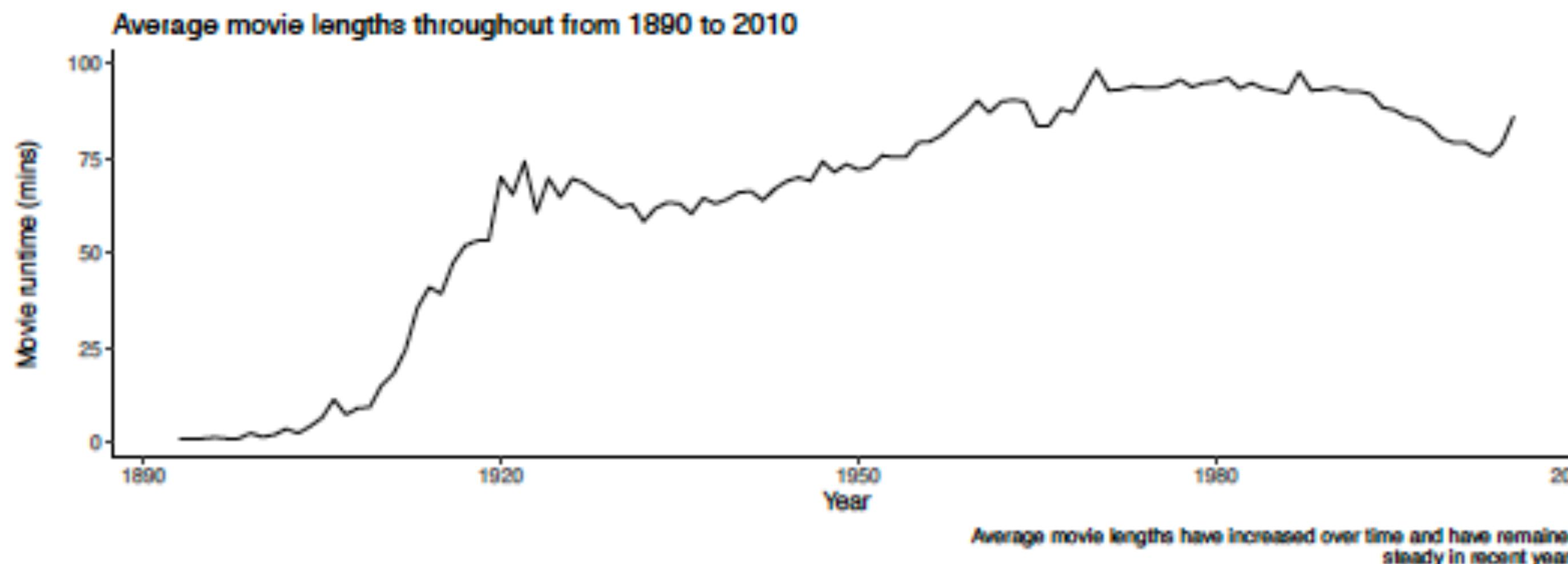
# Kevin Kennedy

Average Number of U.S. Births by Day of the Week

Note: Data from 2000–2014



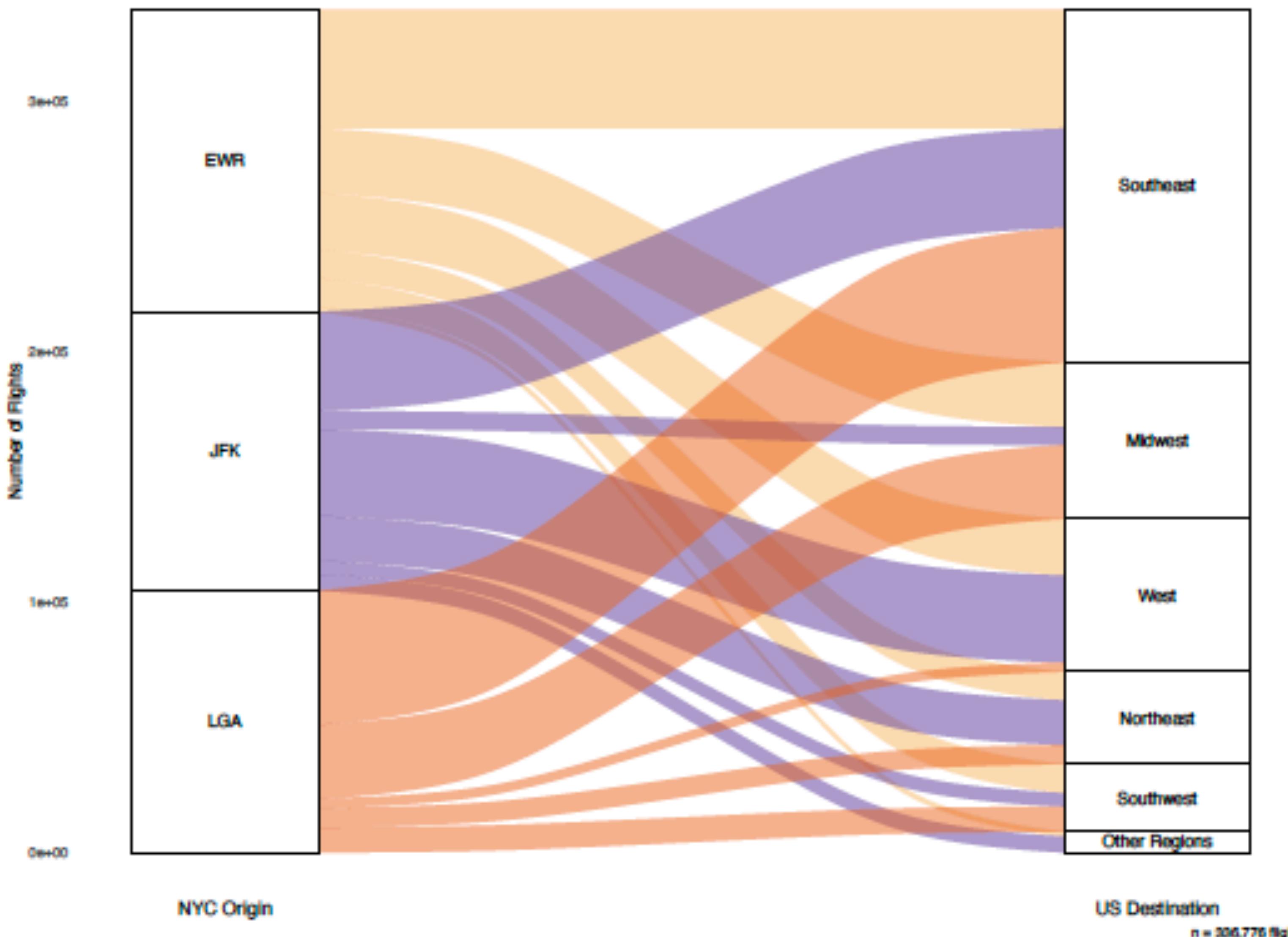
# Bruno Lam



Movie lengths used to be bi-modal, before converging into longer movies in recent history, where as short movies were much more popular from 1910–40

# Marcos Rojas

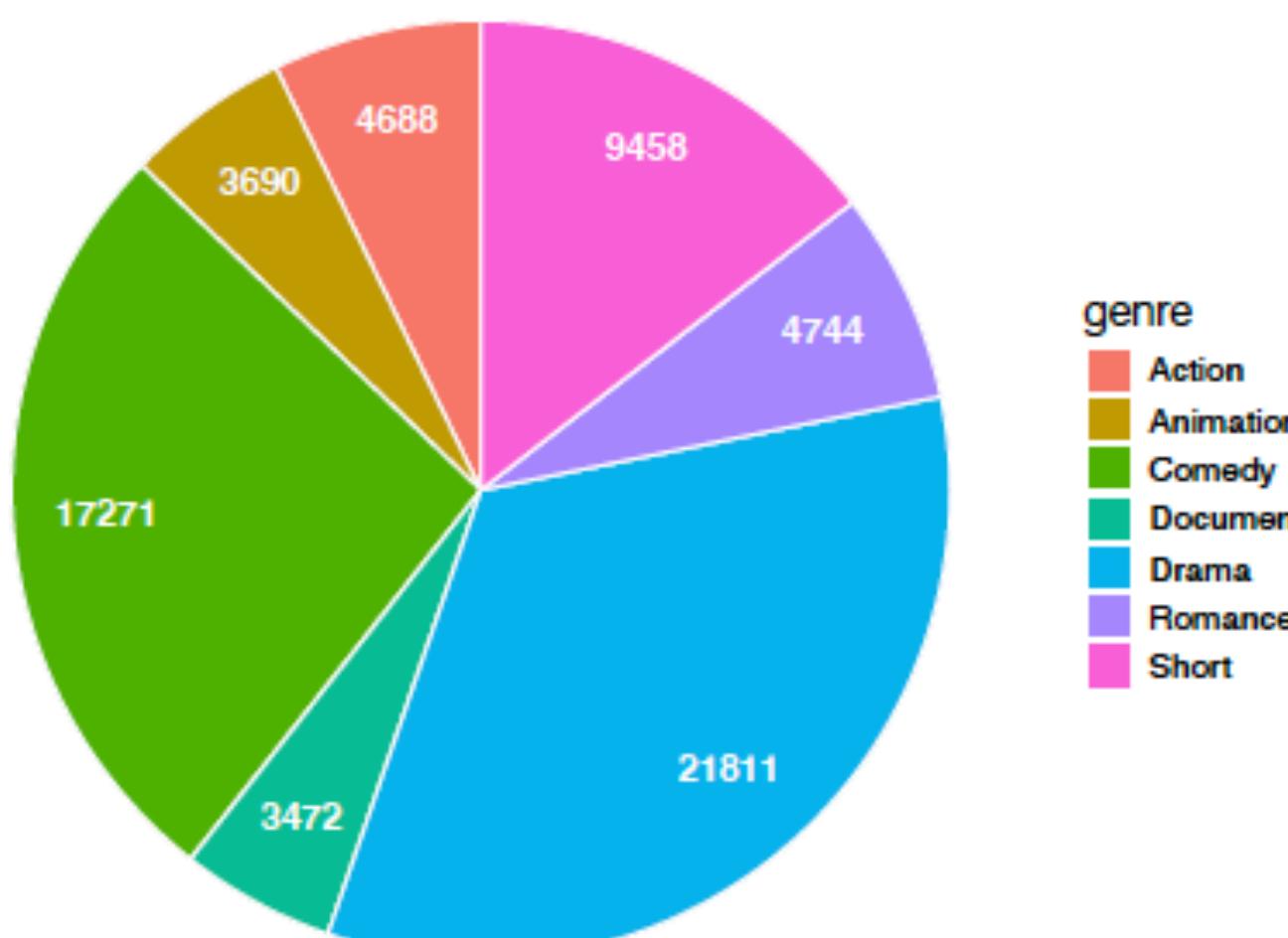
Flight Routes from NYC Airports to U.S. Regions  
Data from 2013



# Yiheng Yao

A

Total number of films per genre



B

Average number of votes per film

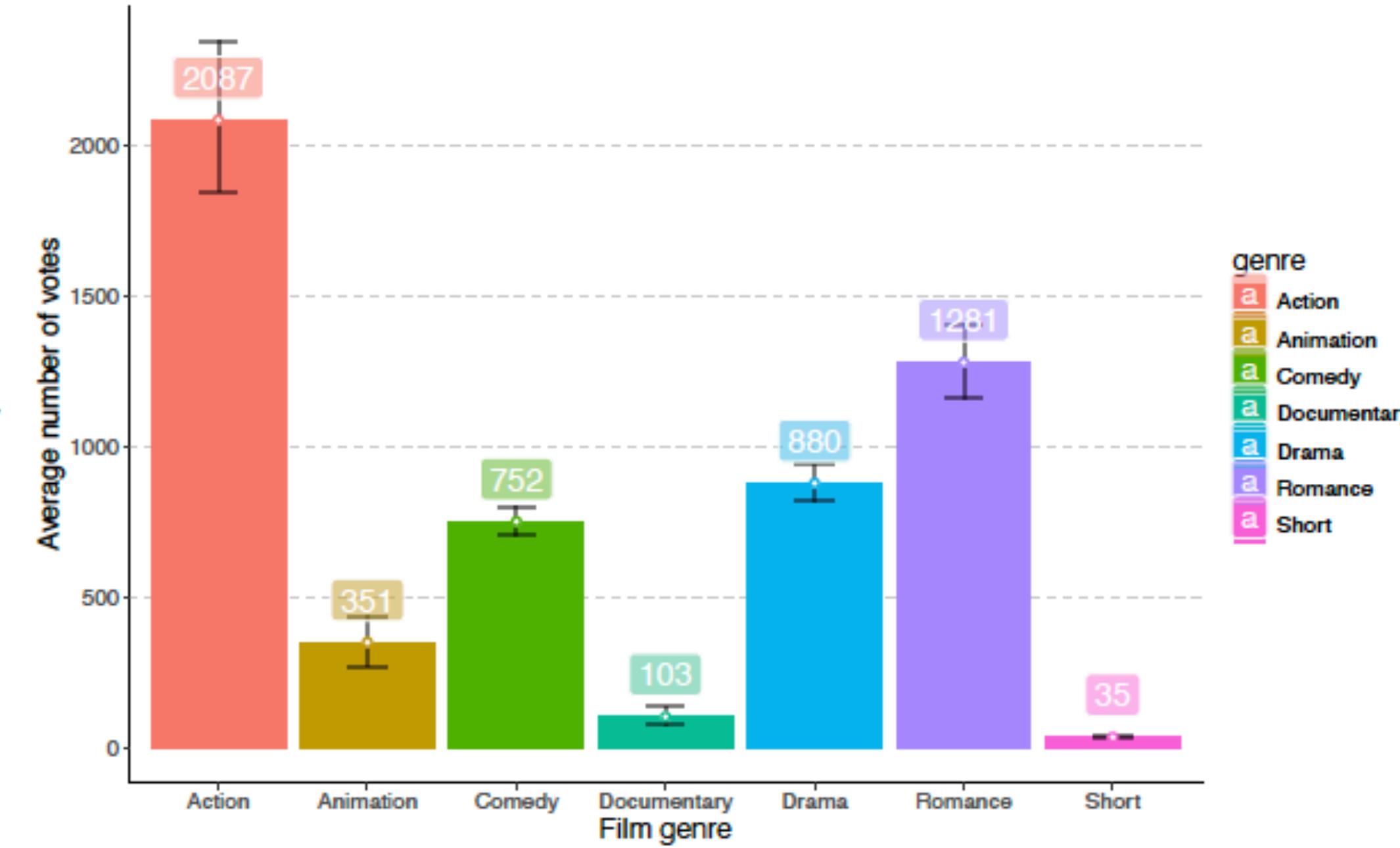
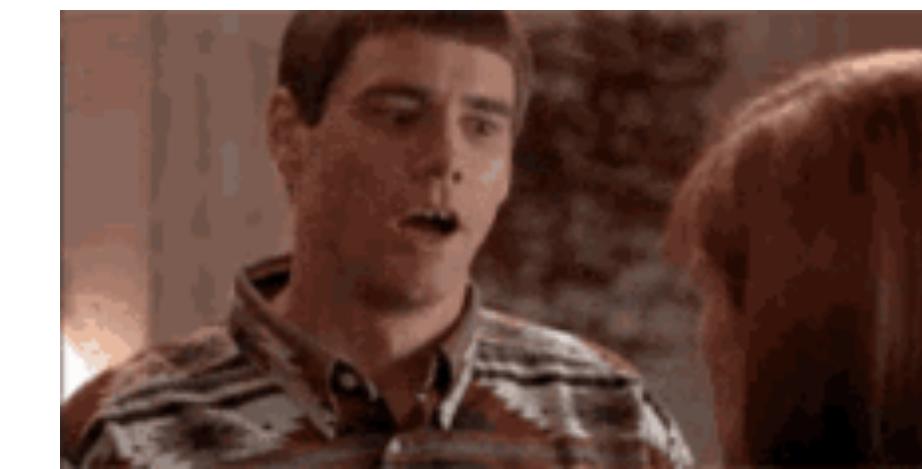


Figure 1A: Total number of films with a positive label for each genre category. Some films fall into multiple genres and will contribute to the count for each. This makes the total count greater than the total number of unique films, but a comparison still provides us an estimate of the proportion of films created in each genre.

Figure 1B: The mean number of votes per film in each genre is a proxy for the popularity of the genre. The total number of votes is normalized by the number of films in the genre category. Action films have the largest number of votes per film (most popular) whereas Short films have the smallest (least popular).

# **Homework**

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

1

## 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 =
#####

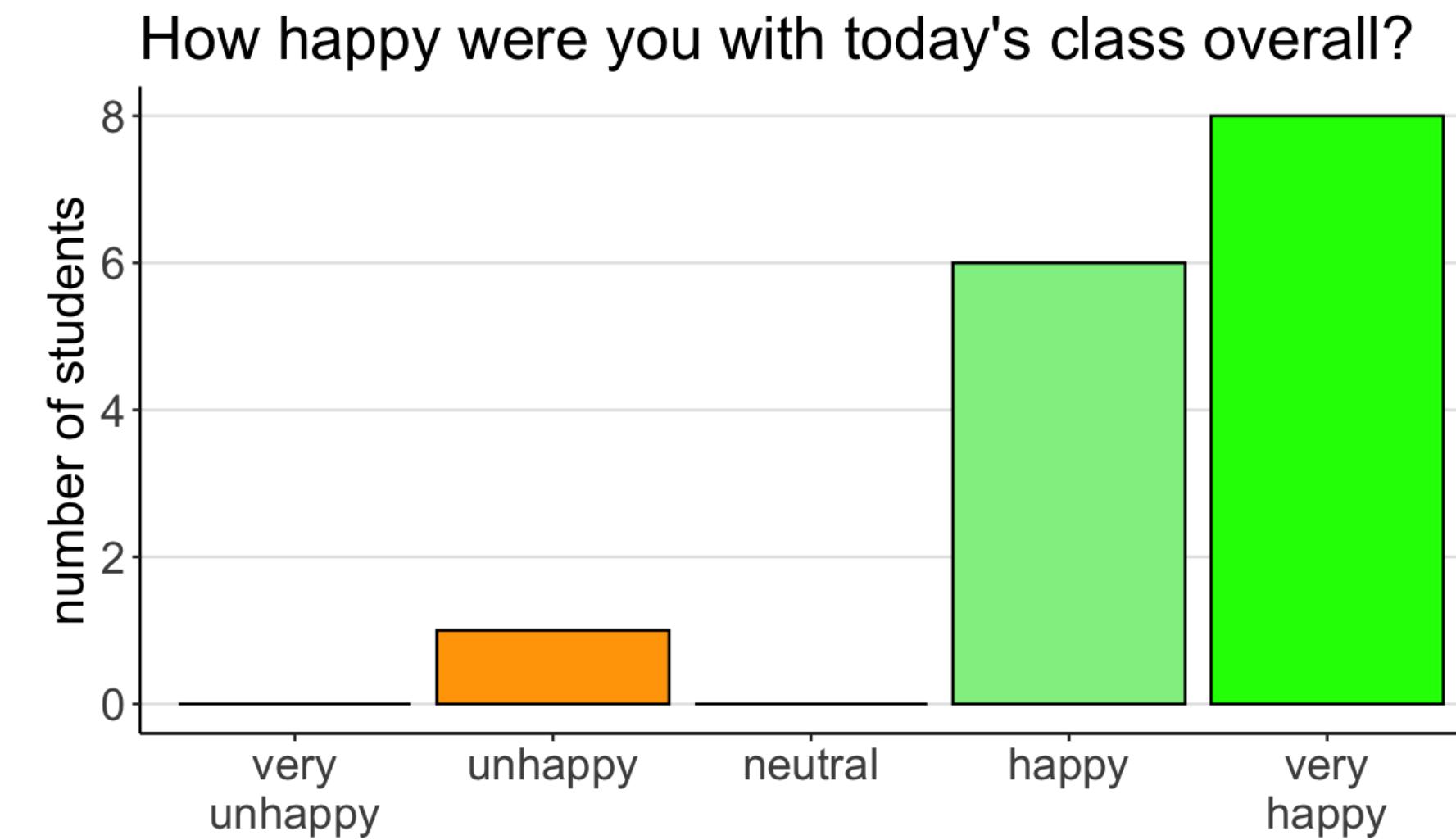
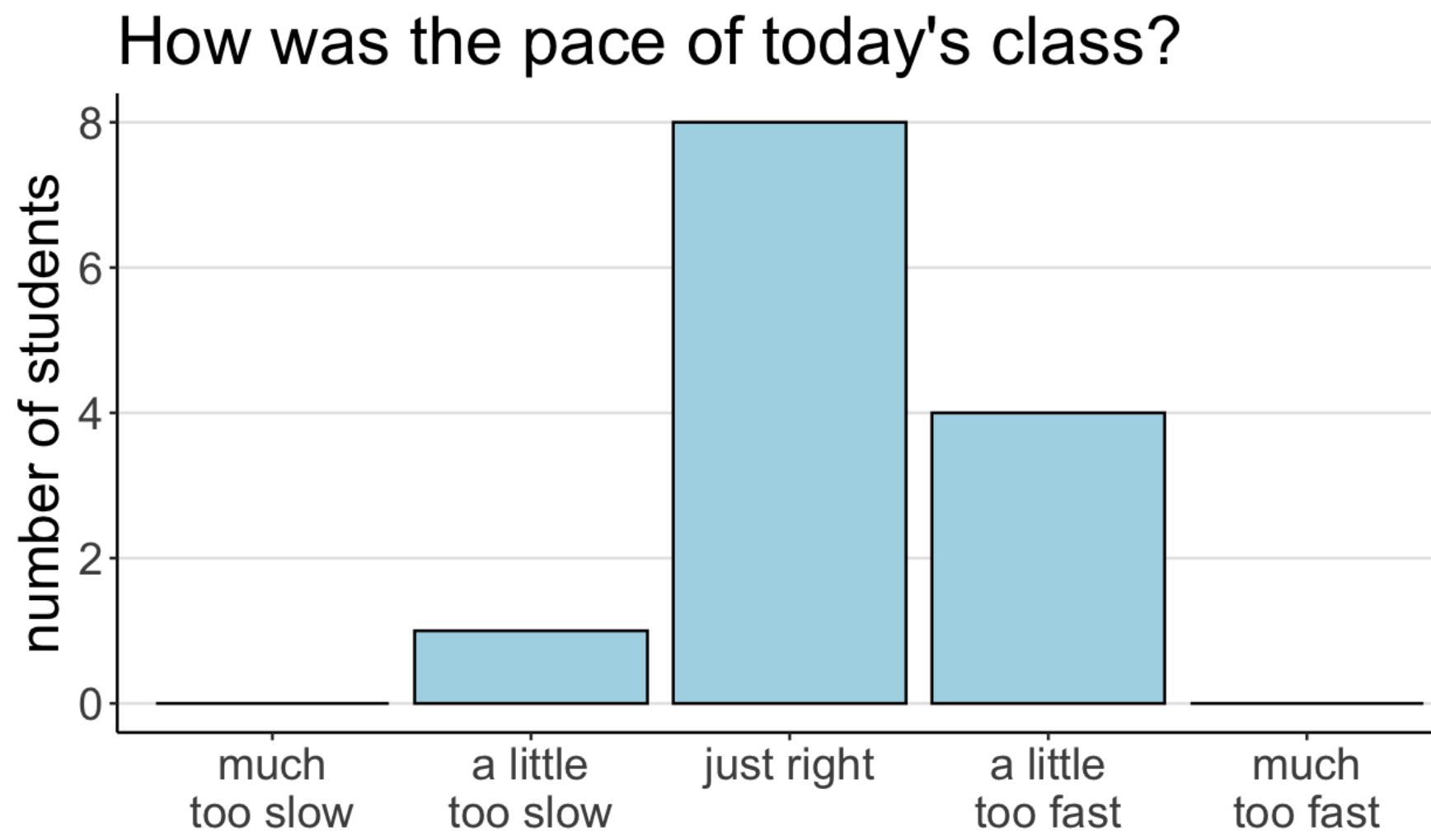
df.samples %>%
  head(5)

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

# **Logistics**

# Your feedback

# Your feedback



When possible, can you write on the whiteboard directly under the screen?

Great examples to illustrate statistical concepts! Thanks :)

Hard to keep up with slides and rmd file simultaneously - live coding sometimes easier to follow

I'm concerned without in-class exercises whether I am developing a sufficient understanding to apply these concepts. I would like to do more hands on examples

Penguin example was great!

Also Nilam is so funny!!! Need humor to get through all these stats

# Outline

## Goal: Revisit and understand key statistical concepts

- Quick recap
- Doing Bayesian Analysis
- Inference in frequentist statistics
- Sampling distributions
- What is a p-value?
  - Permutation test

# Quick recap

# Quick recap from Simulation 1

Simulating data: How?

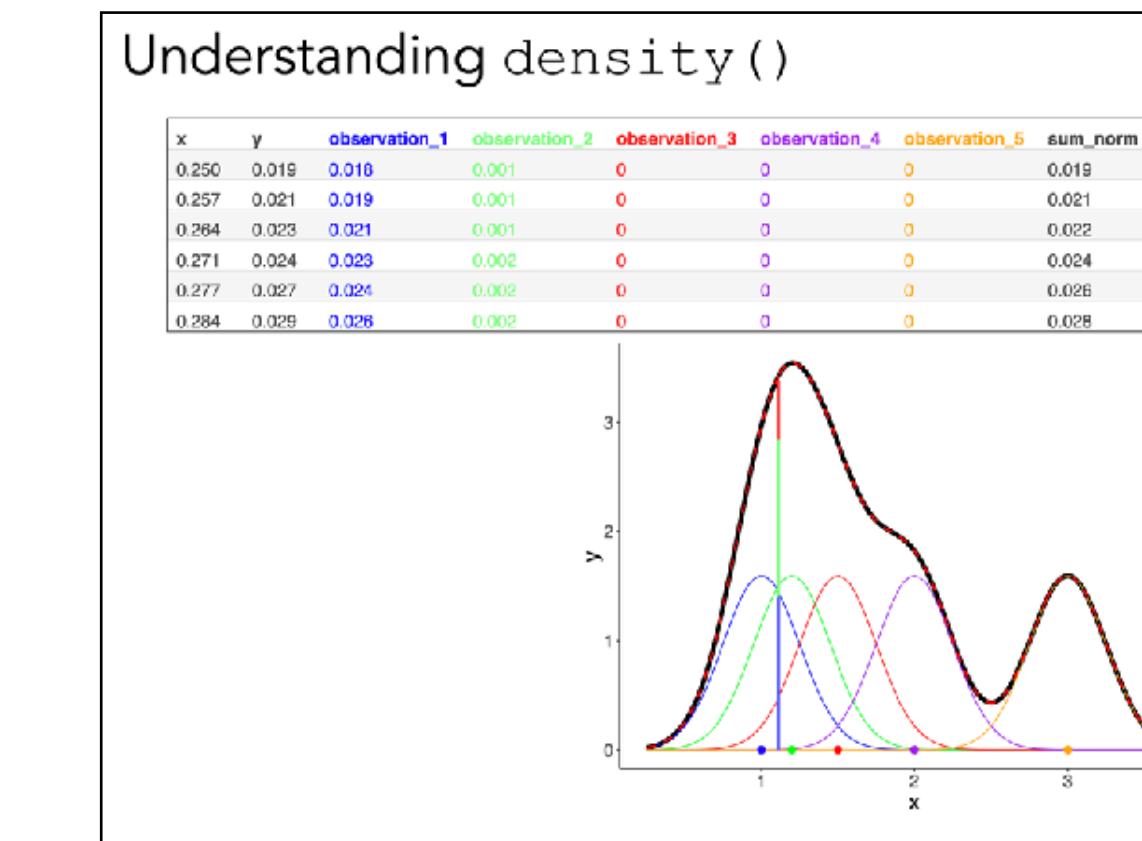
line numbers

```
1 numbers = 1:3
2
3 numbers %>%
4   sample(size = 10,
5     replace = T)
with replacement please
```

[1] 3 3 1 2 2 3 2 3 1 2

thank you

11



sampling values from a vector

understanding density()

Simulating data: How?

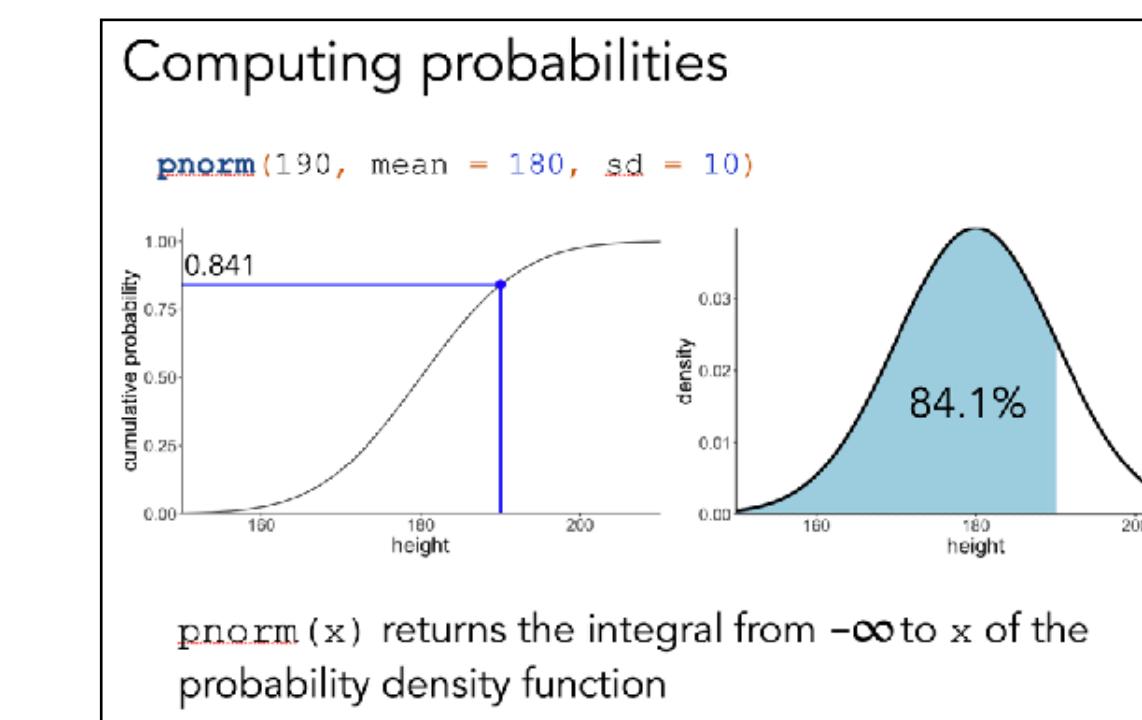
Sampling rows from a data frame

```
1 set.seed(1)
2 n = 10
3 df.data = tibble(trial = 1:n,
4   stimulus = sample(c("flower", "pet"), size = n, replace = T),
5   rating = sample(1:10, size = n, replace = T))

sample 6 rows with replacement
1 df.data %>%
2   slice_sample(n = 6,
3     replace = T)

sample 50% of the rows
1 df.data %>%
2   slice_sample(prop = 0.5)
```

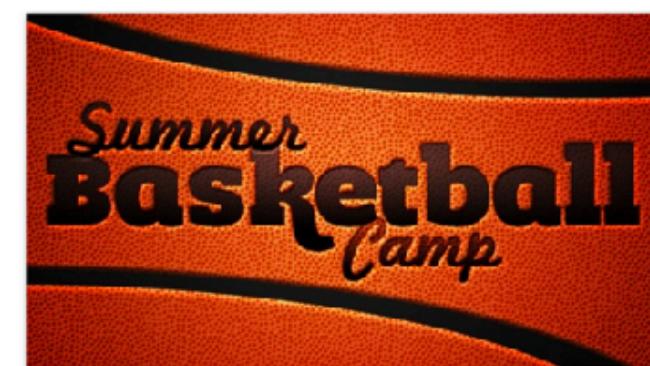
sampling rows from a data frame



answering questions with probability distributions

or via drawing samples  
rnorm() + wrangling

# Quick recap: Bayesian inference



twice as many kids go to the basketball camp

$$X \sim \text{Normal}(\mu = 170, \sigma = 8)$$



$$X \sim \text{Normal}(\mu = 180, \sigma = 10)$$



Can you feel the Bayes?

$$H = \{\text{basketball, chess}\}$$

$$D = 175 \text{ cm}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{probability of the data!}}$$

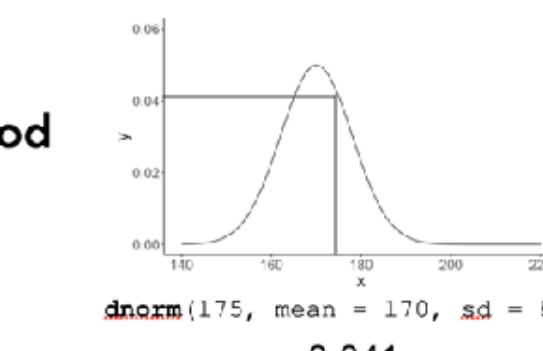
$$p(H|D) = \frac{p(D|H) \cdot p(H)}{p(D)}$$

*H = Hypothesis  
D = Data*

Summer camp

$$\text{prior } p(\text{chess}) = \frac{1}{3}$$

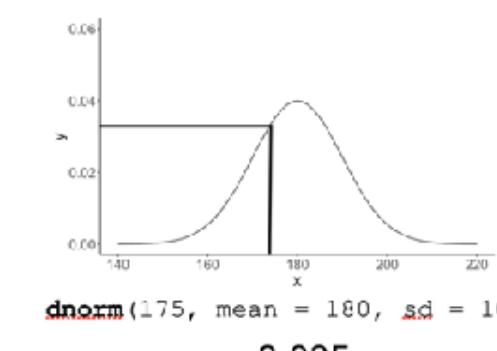
likelihood



$$= 0.041$$

Analytic solution

$$p(\text{basketball}) = \frac{2}{3}$$



$$= 0.035$$

posterior

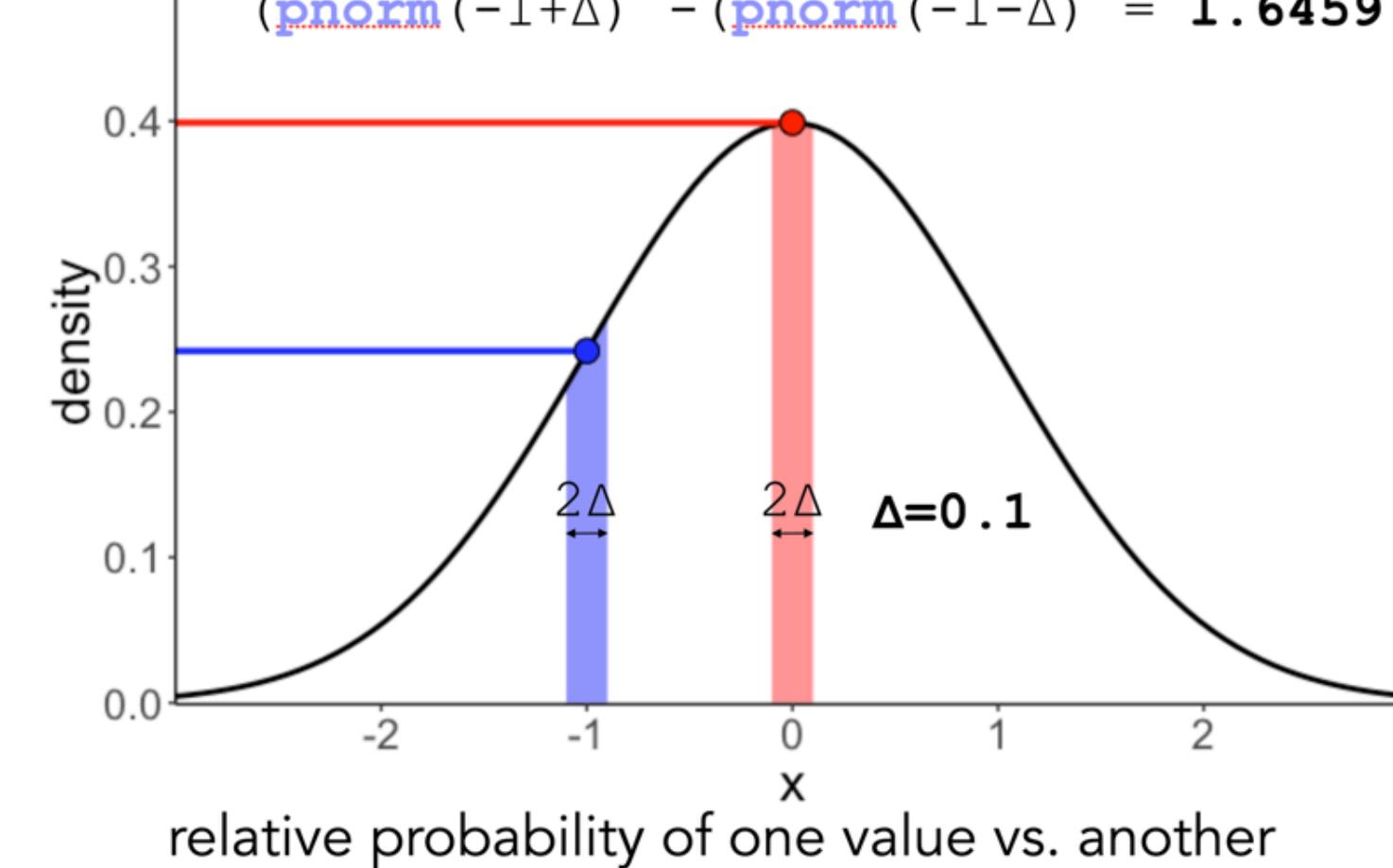
$$p(\text{sport} = \text{basketball} | \text{height} = 175) = \frac{p(175 | \text{basketball}) \cdot p(\text{basketball})}{p(175) \text{ data}}$$

$$p(\text{basketball} | 175) = \frac{p(175 | \text{basketball}) \cdot p(\text{basketball})}{p(175 | \text{basketball}) \cdot p(\text{basketball}) + p(175 | \text{chess}) \cdot p(\text{chess})}$$

Probability vs. likelihood

$$\text{dnorm}(0) / \text{dnorm}(-1) = 1.6487$$

$$\frac{(\text{pnorm}(0+\Delta) - \text{pnorm}(0-\Delta))}{(\text{pnorm}(-1+\Delta) - (\text{pnorm}(-1-\Delta)))} = 1.6459$$



Summer camp: Via sampling

**Sampling solution**

```
1 df.camp = tibble(
2   kid = 1:100000,
3   sport = sample(c("chess", "basketball"),
4                 size = 100000,
5                 replace = T,
6                 prob = c(1/3, 2/3))) %>%
7   rowwise() %>%
8   mutate(height = ifelse(test == sport == "chess",
9                         yes = rnorm(., mean = 170, sd = 8),
10                        no = rnorm(., mean = 180, sd = 10))) %>%
11  ungroup()
```

| kid | sport      | height |
|-----|------------|--------|
| 1   | basketball | 164.84 |
| 2   | basketball | 163.22 |
| 3   | basketball | 191.18 |
| 4   | chess      | 160.16 |
| 5   | basketball | 182.99 |
| 6   | chess      | 163.54 |
| 7   | chess      | 168.56 |
| 8   | basketball | 192.99 |
| 9   | basketball | 171.91 |
| 10  | basketball | 177.12 |

this works!

```
1 df.camp %>%
2   filter(between(height,
3                   left = 174,
4                   right = 176)) %>%
5   count(sport)
```

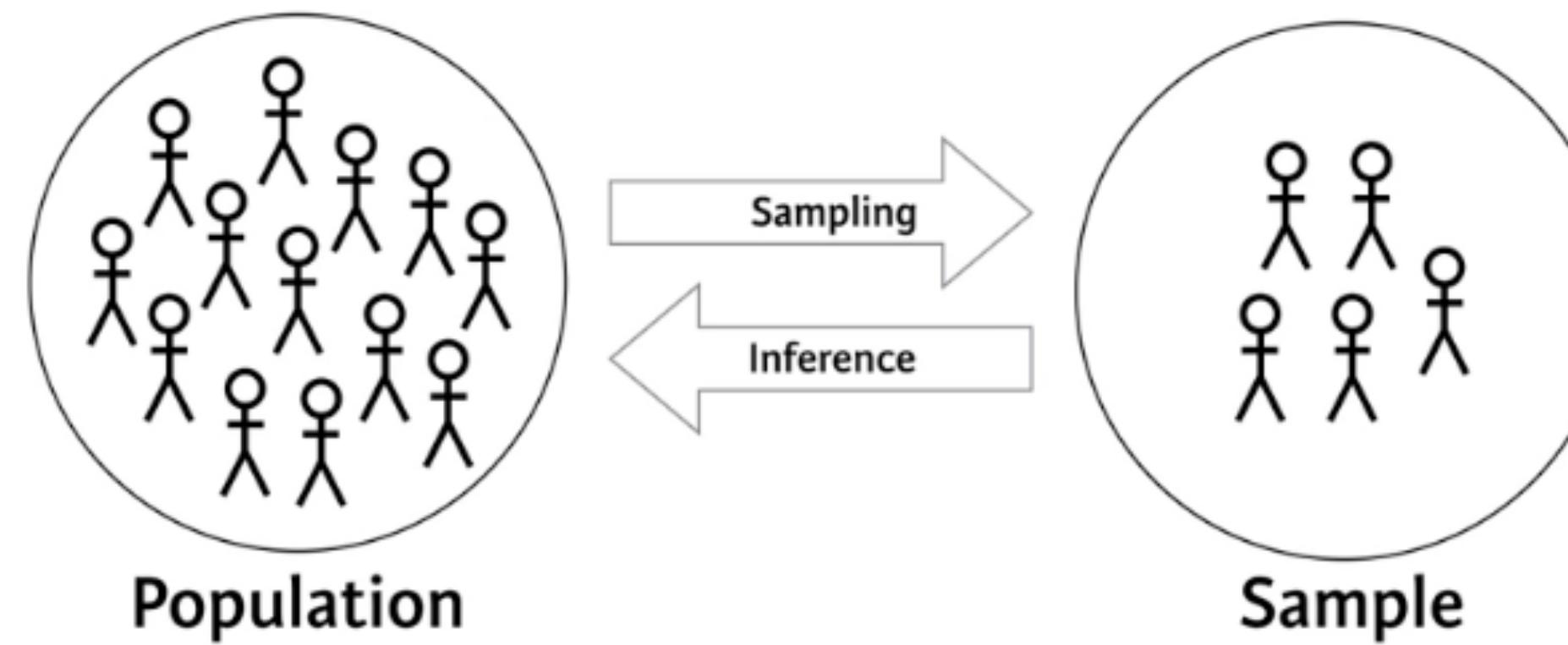
| sport      | n   |
|------------|-----|
| basketball | 469 |
| chess      | 273 |

$$\frac{\text{basketball}}{\text{basketball} + \text{chess}} \approx 0.63$$

# **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 be deterministic



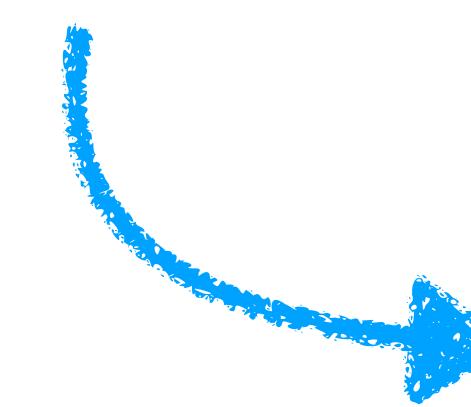
# Underlying principle of statistical testing

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  $\alpha$

# The magical component

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

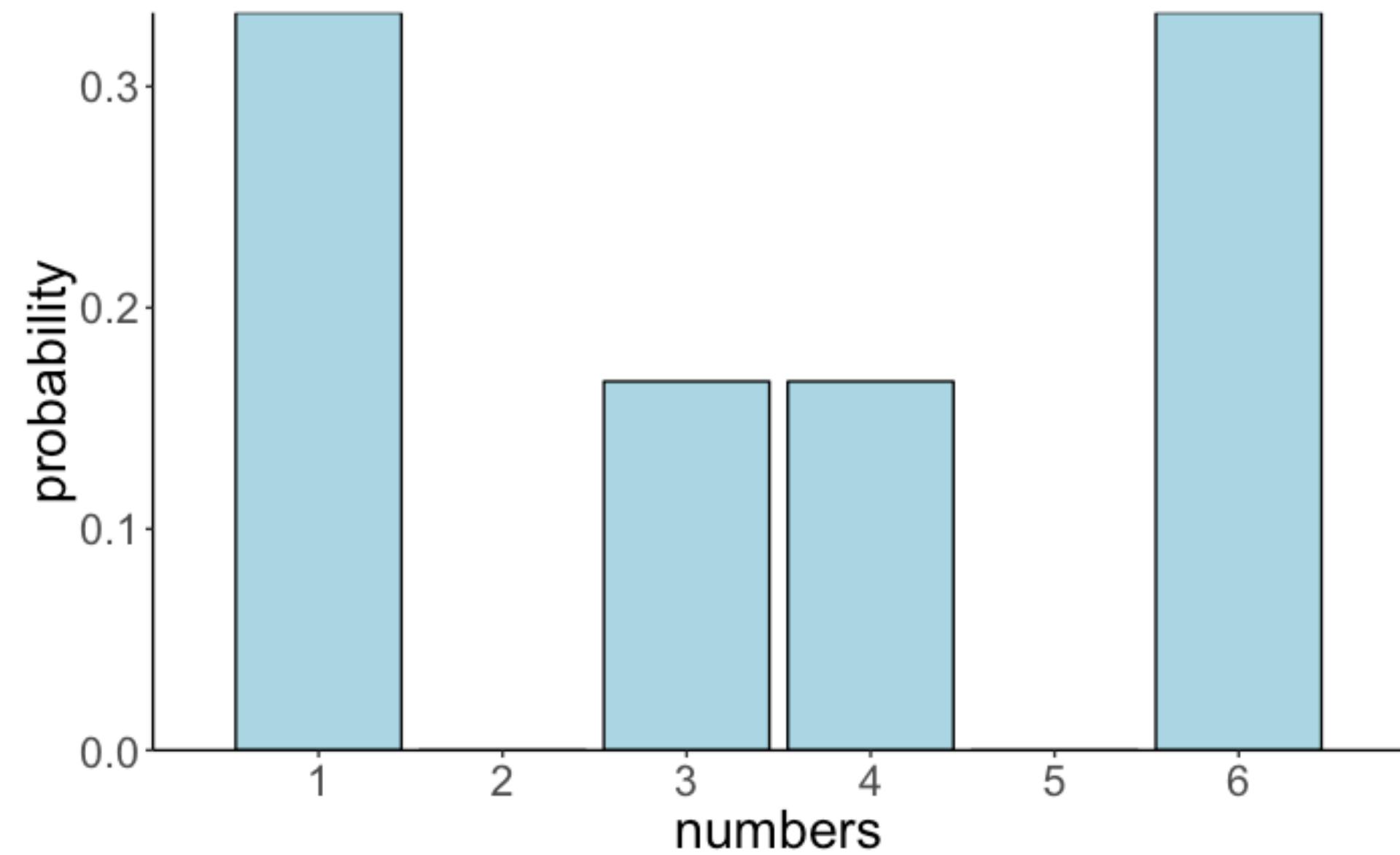
This dates back to pre-computer era where statisticians derived mathematically the distribution of statistical measures for an infinite amount of samples! That's a tricky thing to do and these approximations are typically tied to assumptions such as normality, homoscedasticity, independent observations,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



**instead: simulation-based approach**

# **Sampling distributions**

# heavy metal distribution

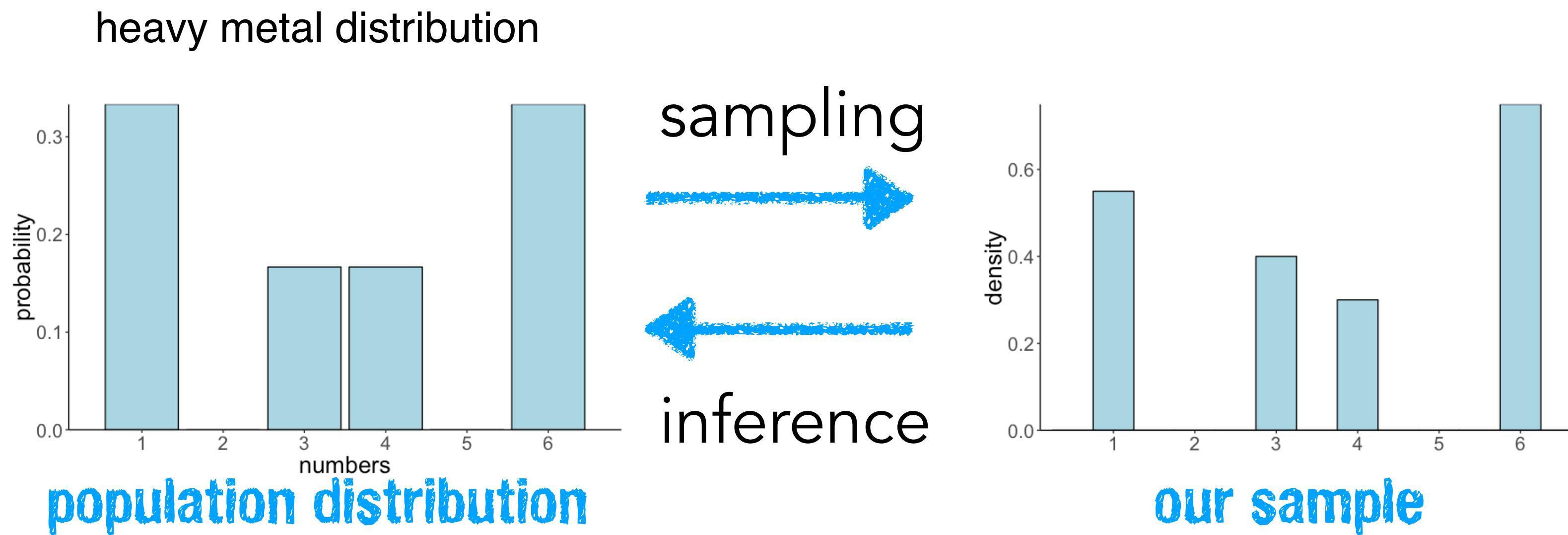
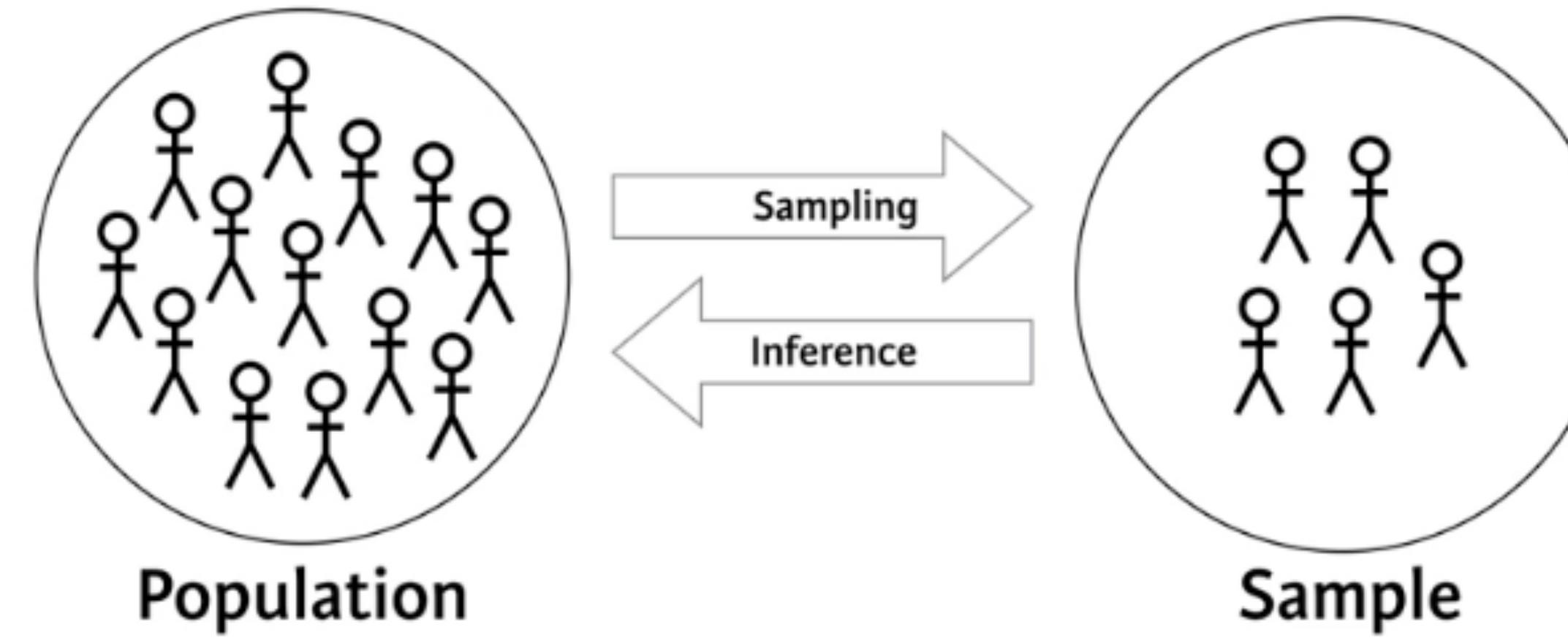


population distribution

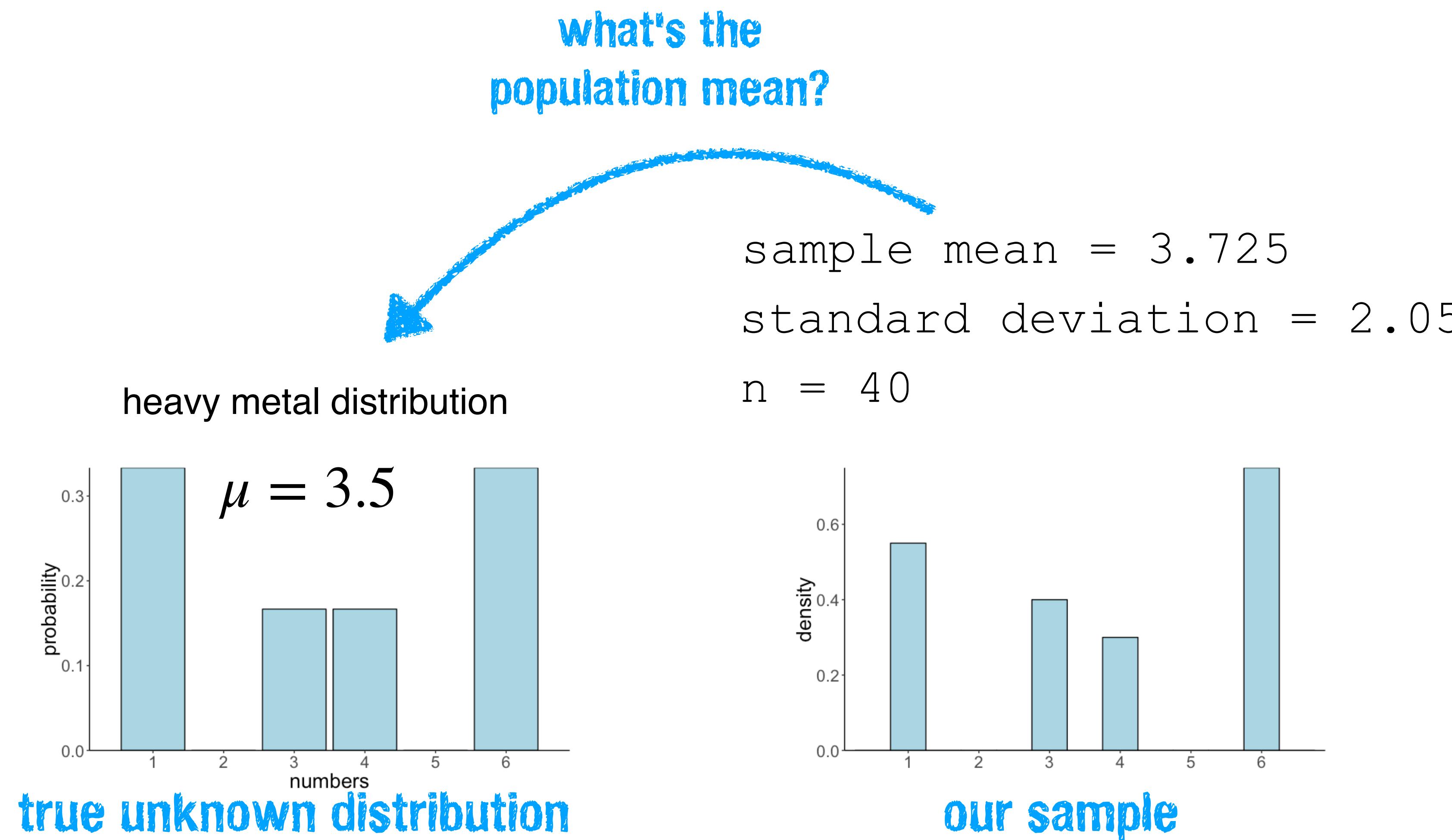


# Statistical inference

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

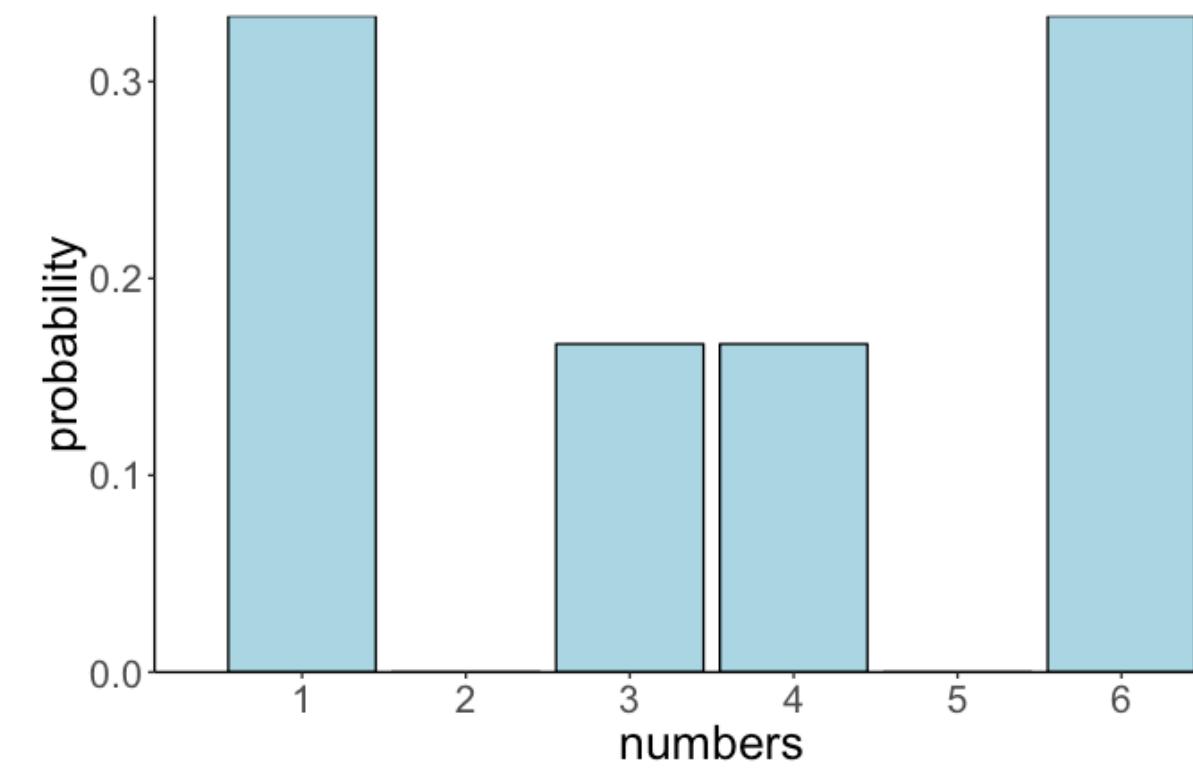


# Statistical inference

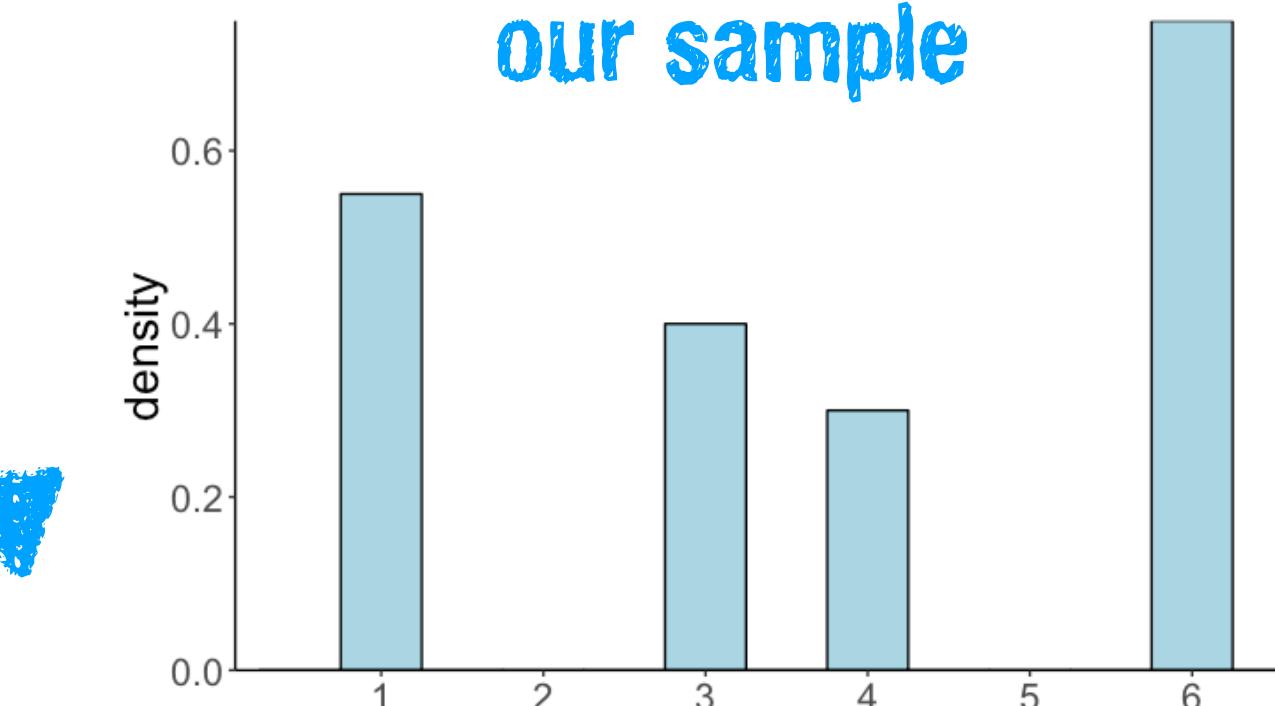


# Sampling variation

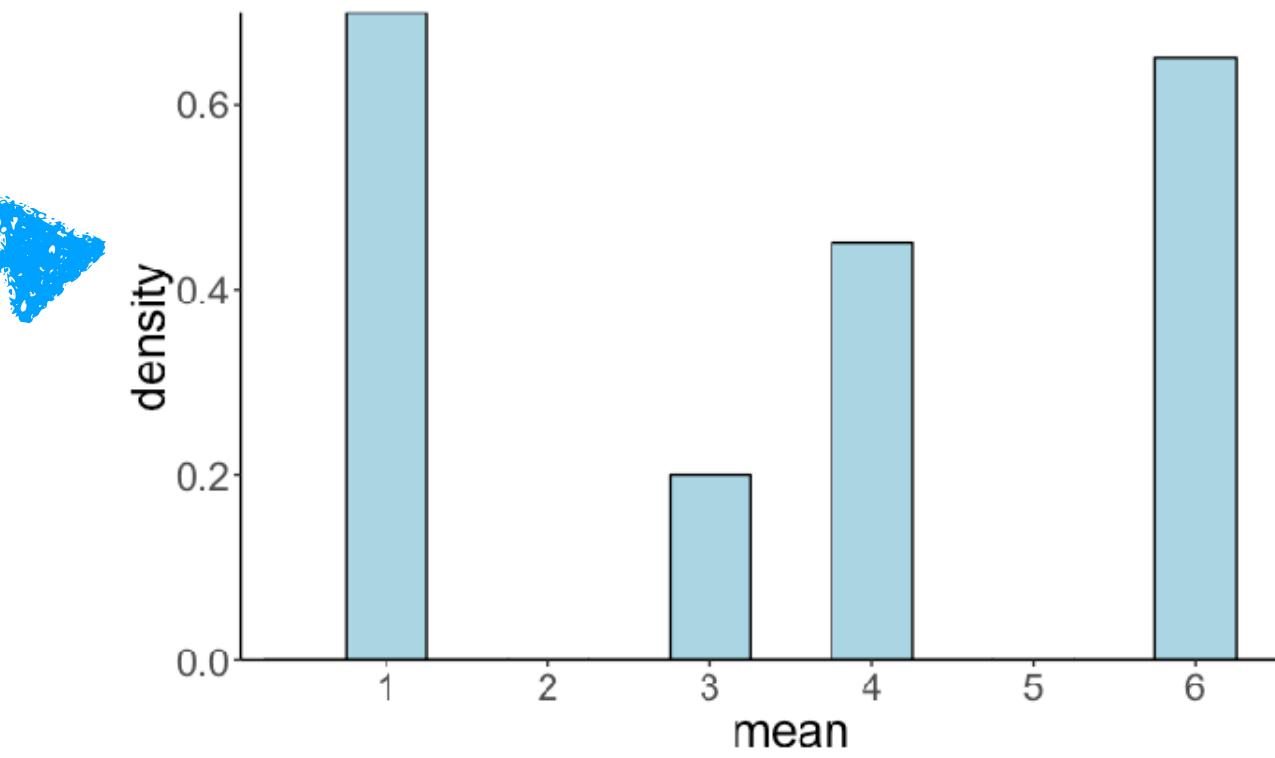
heavy metal distribution



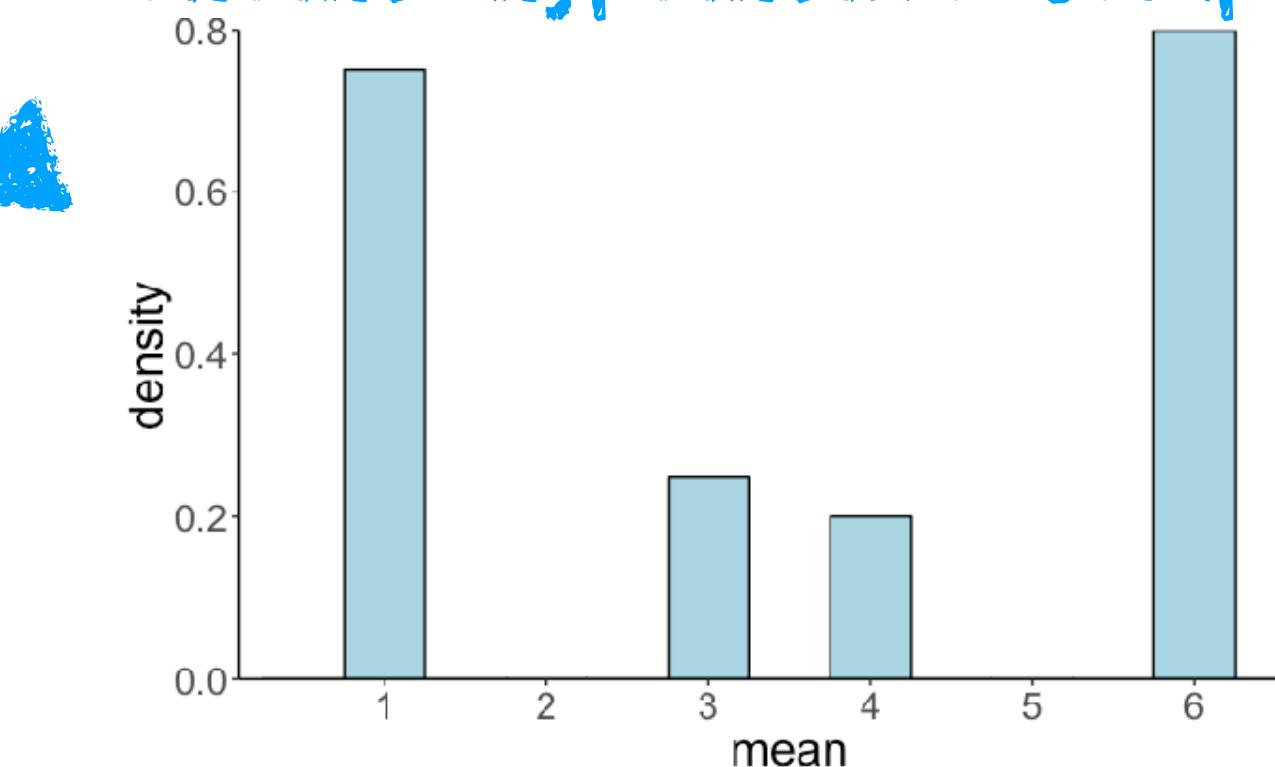
population distribution



hypothetical sample



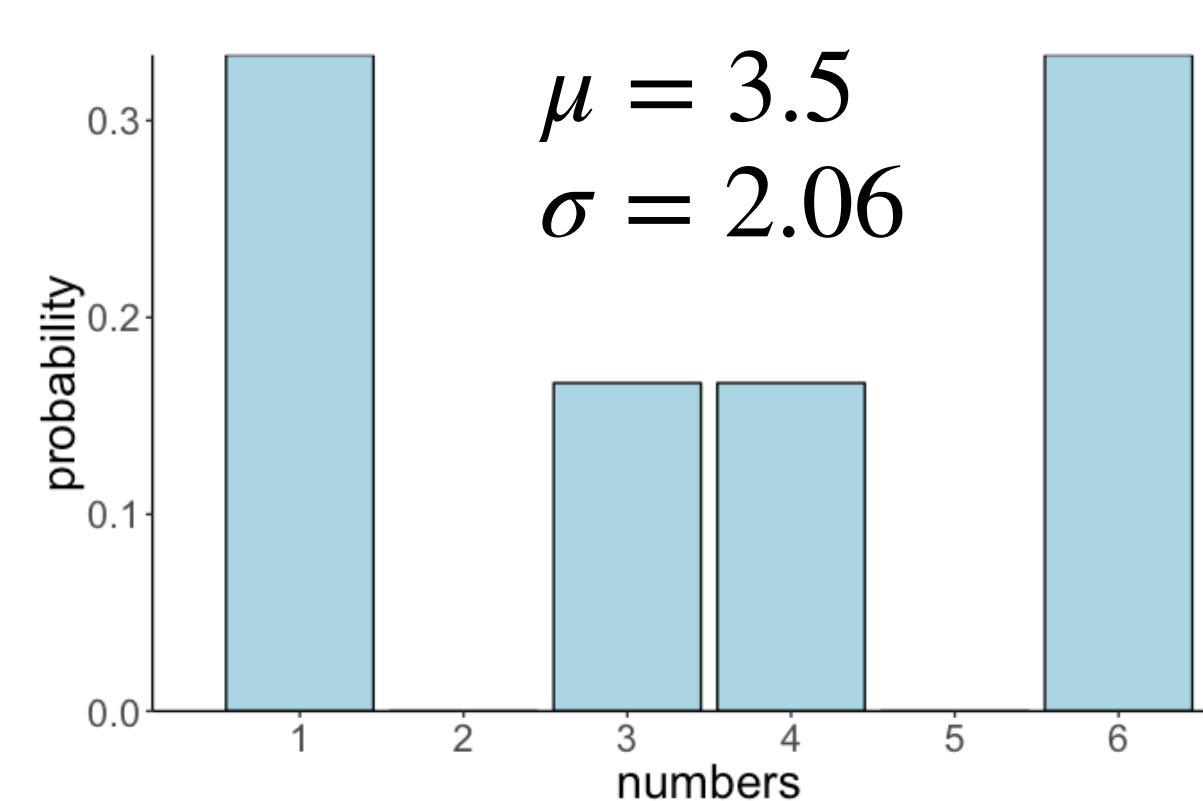
another hypothetical sample



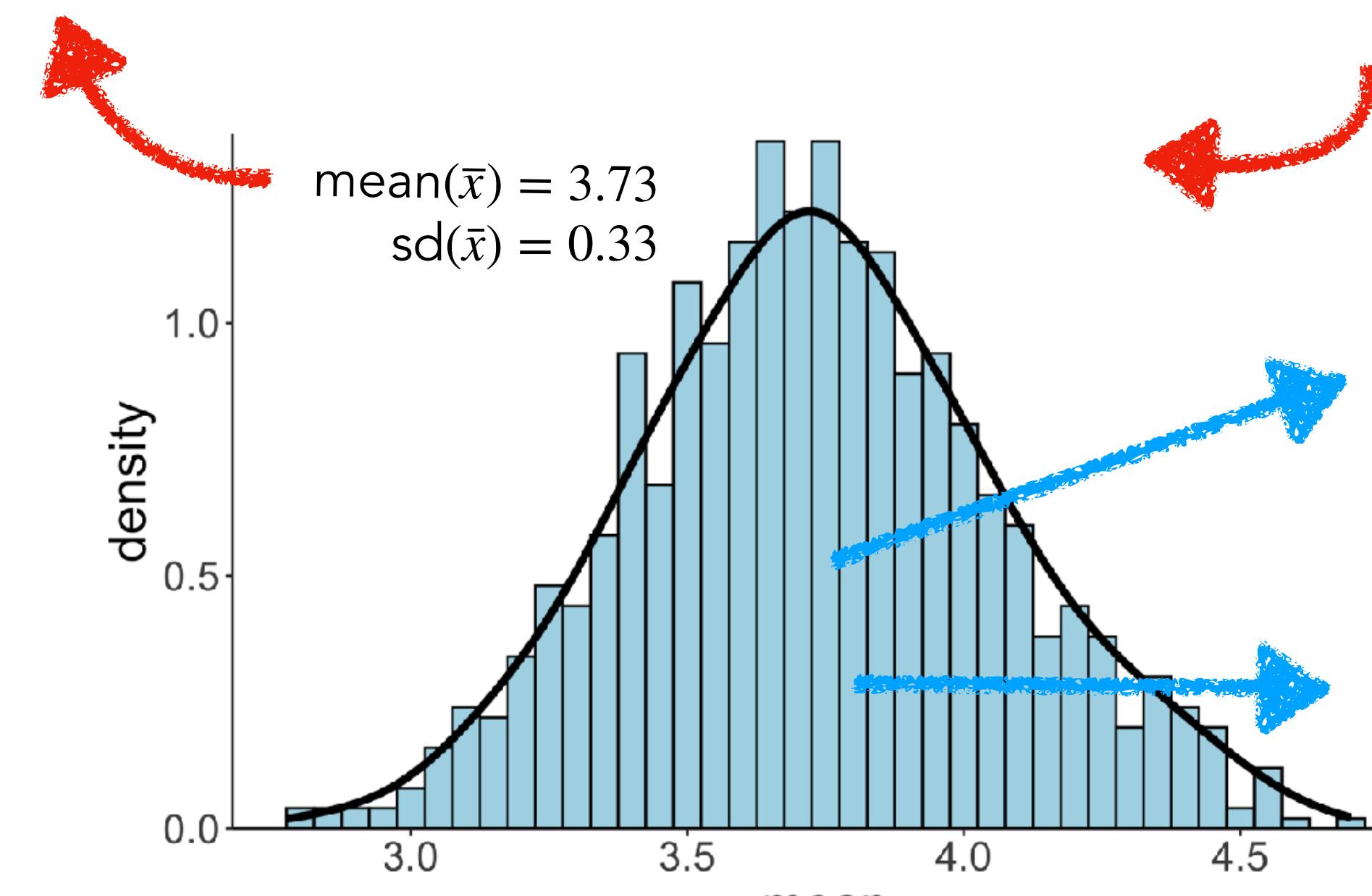
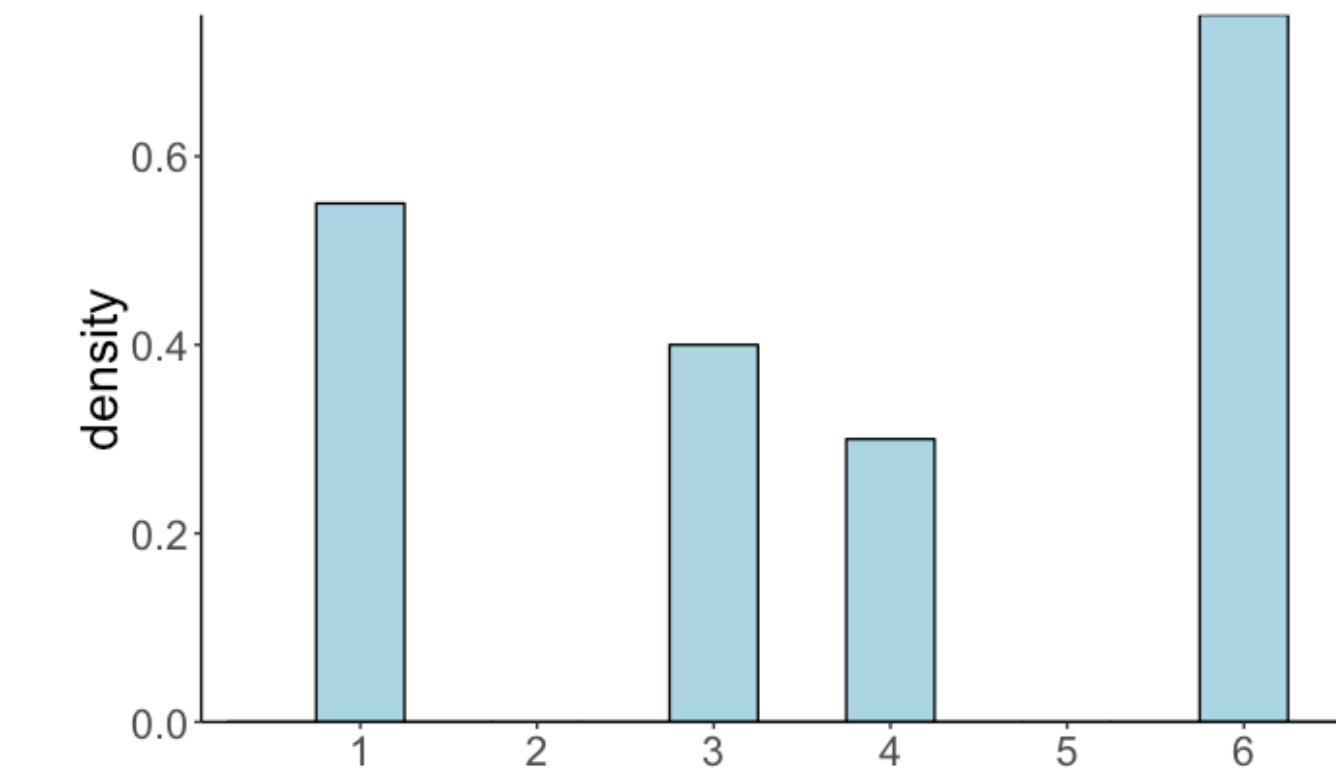
# Sampling distribution

population distribution

heavy metal distribution



our sample

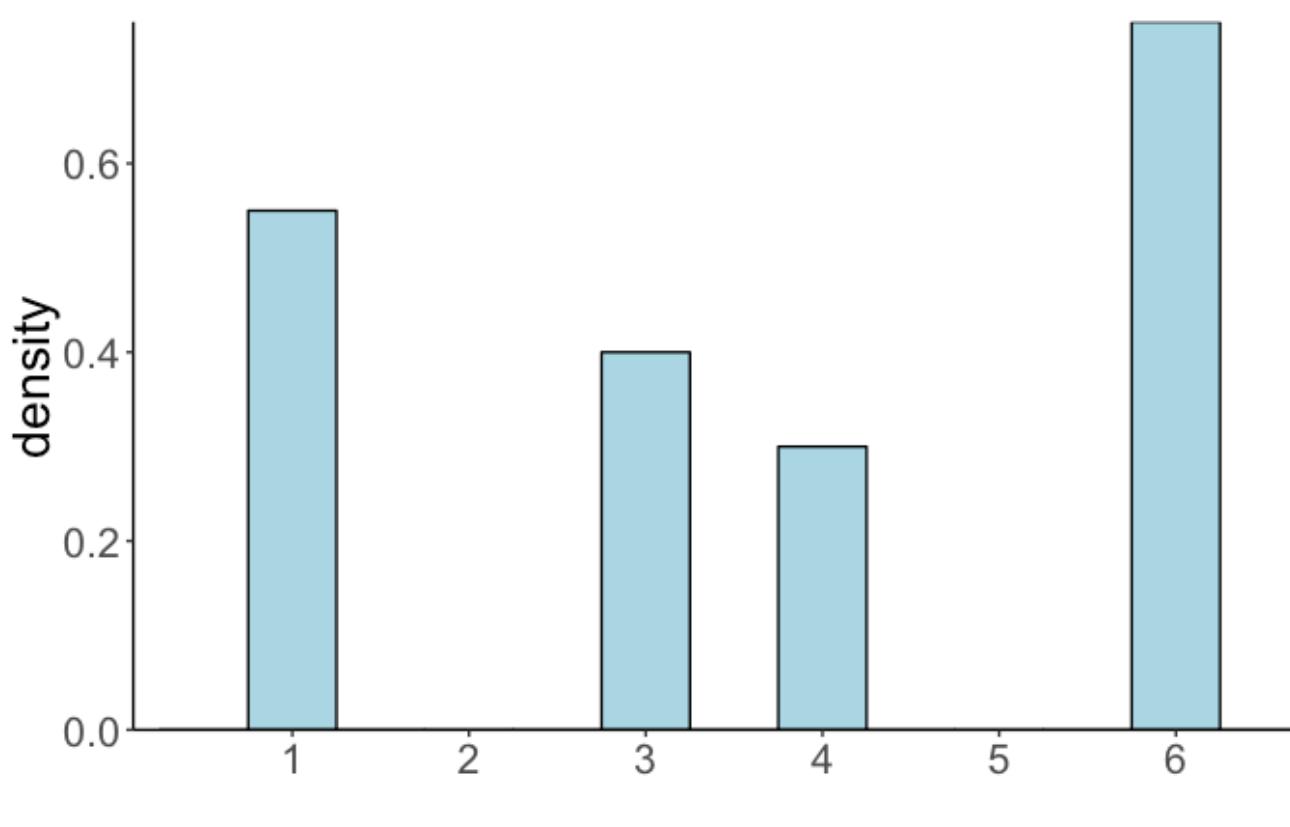


sampling distribution of the mean

p-values

confidence intervals

## our sample

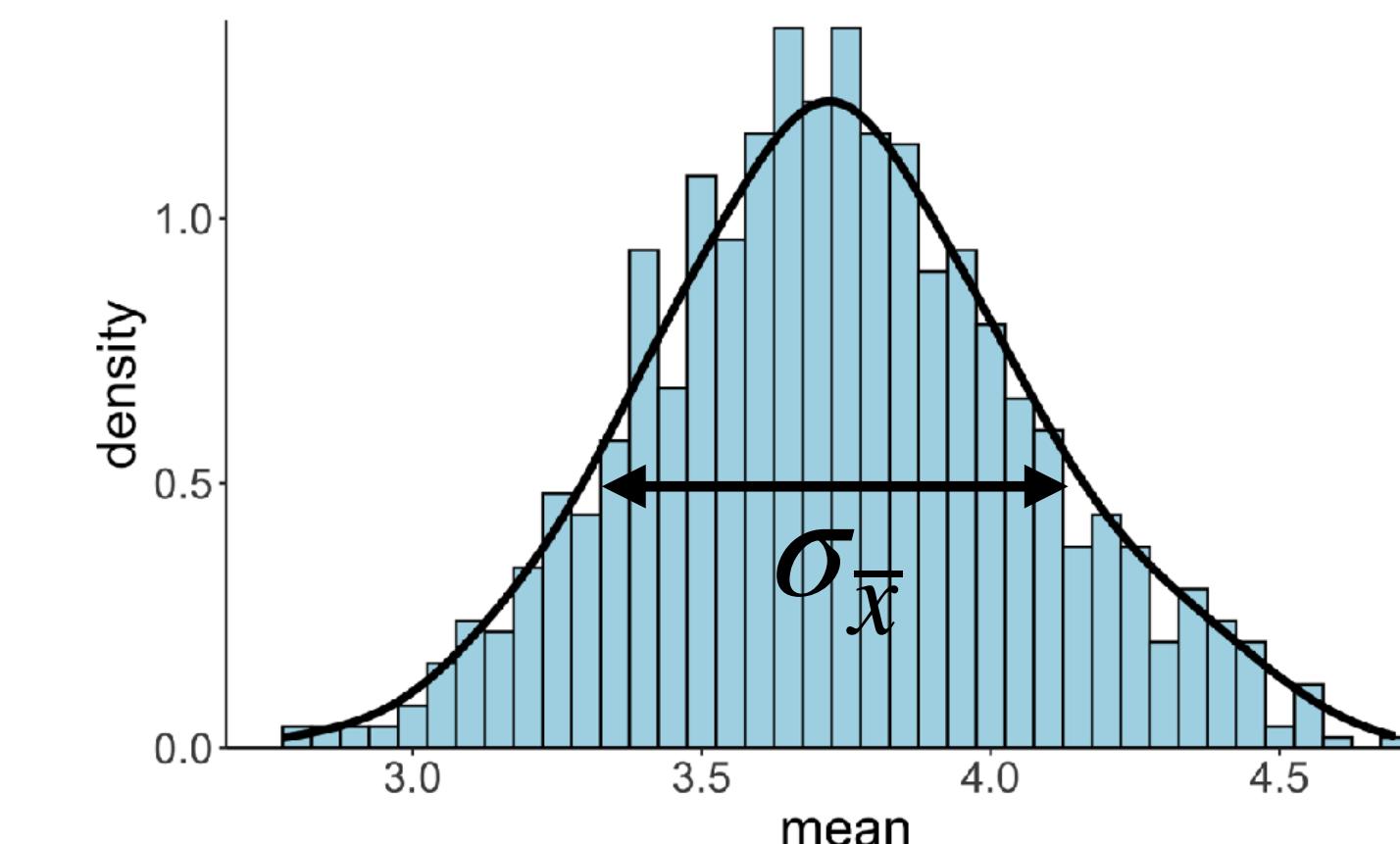


## standard deviation

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

gives a sense for how well the mean summarizes the data

## sampling distribution



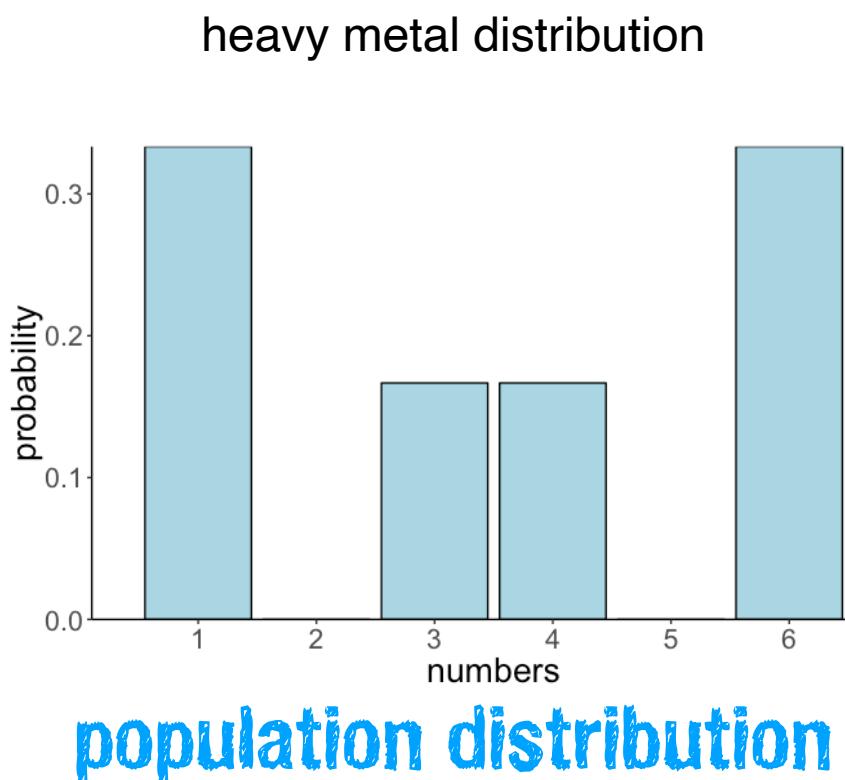
## standard error

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

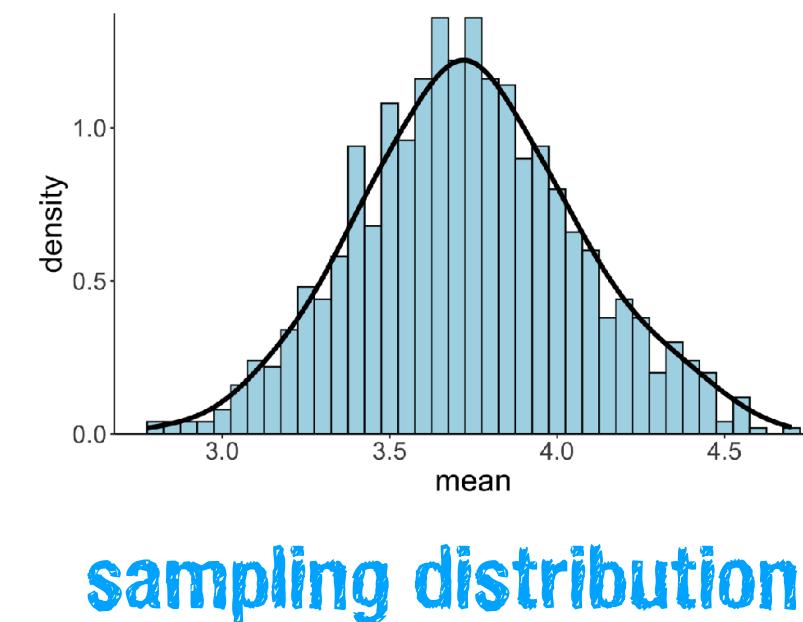
the standard deviation of the sampling distribution  
how much variation would we expect between the means of different samples

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

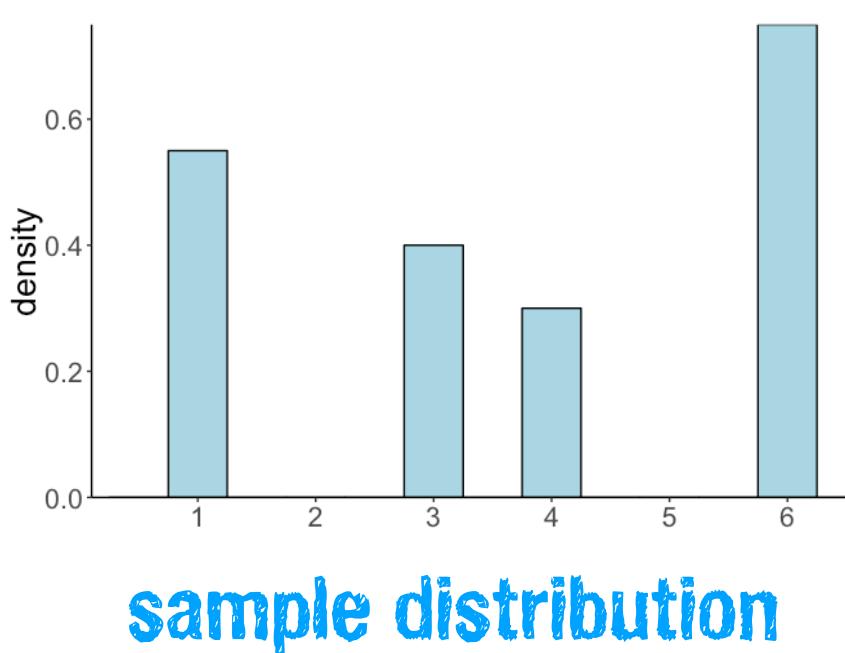
# 3 distributions in statistical inference



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



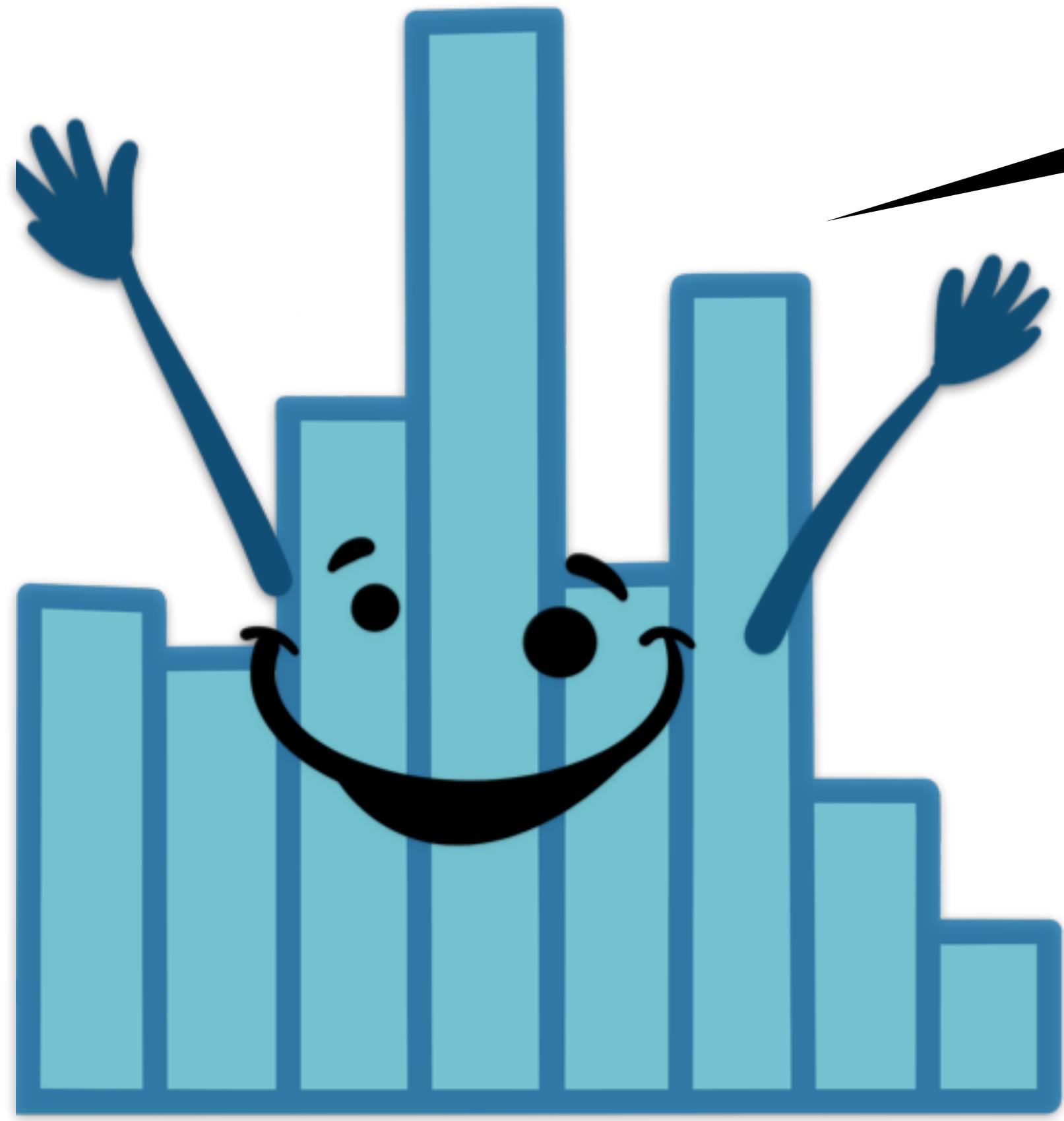
- bridge between sample and population
- derived mathematically / computationally
- asymptotic distribution theory or resampling approaches
- shows how test statistic varies between samples



- our observed sample
- we compute statistics of interest (mean, variance, correlation, ...)
- make an inference about the population via the sampling distribution

02:00

stretch break!



# **What is a p-value?**

# Statistical inference

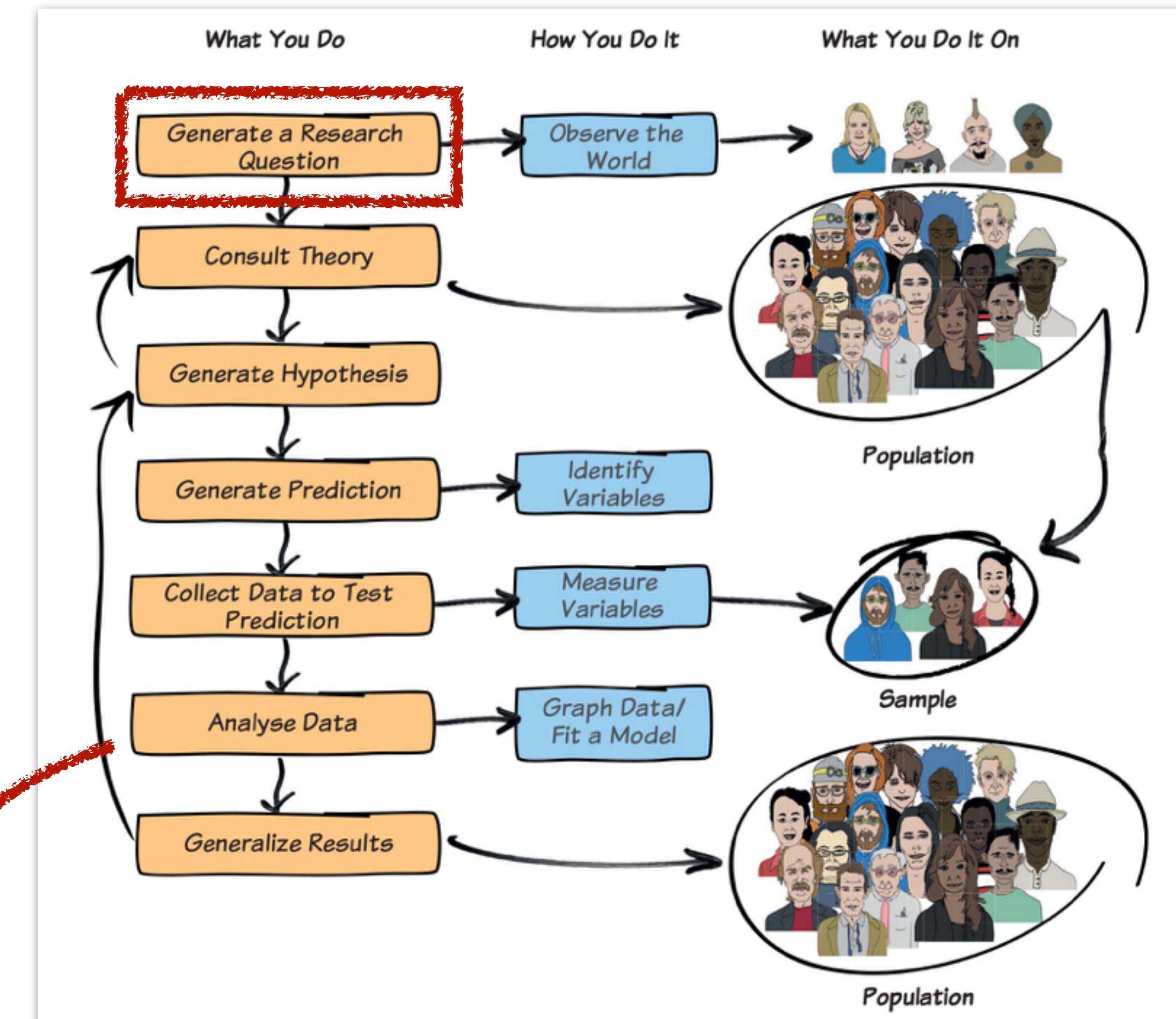
null hypothesis

$$H_0 : \mu_1 = \mu_2$$

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?

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

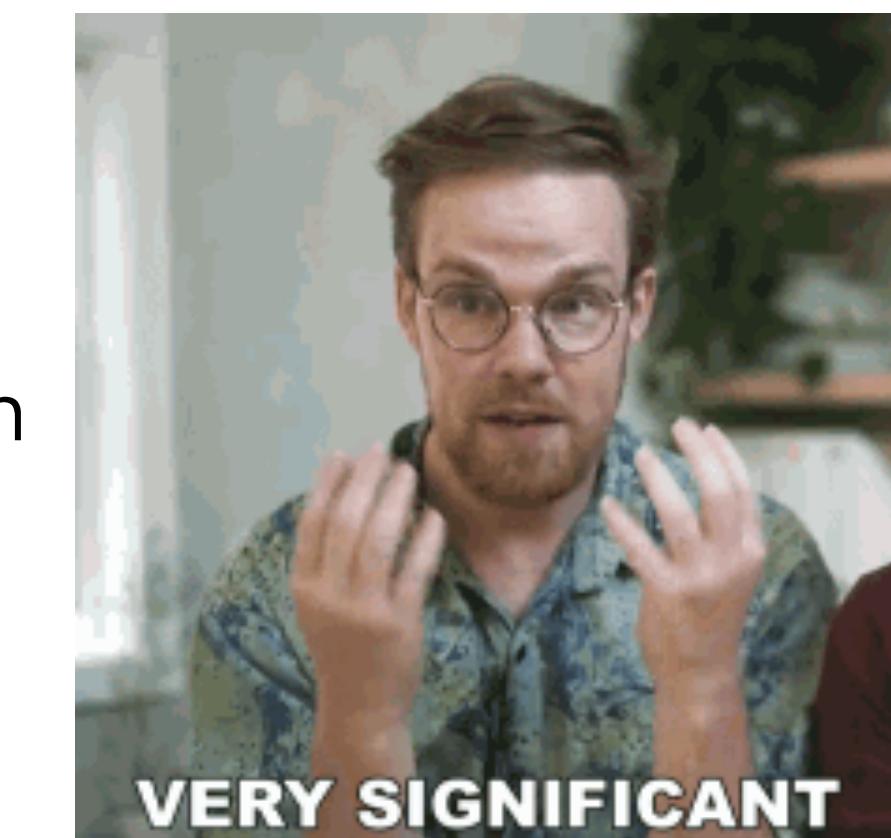
The p-value is the probability that the alternative hypothesis is true.

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.

# What is a p-value? **Your answers**

- Used in hypothesis significance testing. It's quantifies the strength of the evidence against the null hypothesis provided by the data.
- The probability that your data occurred by chance and that there is a statistical difference
- An indication of statistical power related to how likely the null hypothesis is true
- p-value is the likelihood of one data point fall into the sampling distribution
- Probability of getting results that are significant above random chance
- level of significance, usually set to  $p < .05$ ,  $p <.01$ , or  $p <.001$
- Statistical measurement used to validate a hypothesis.
- probability of a given observation to occur
- A value representing confidence in a statistical test's results. E.g., " $p < 0.05$ " means that there is less than a 5% chance that the results were due to random chance rather than a statistically significant pattern



# 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

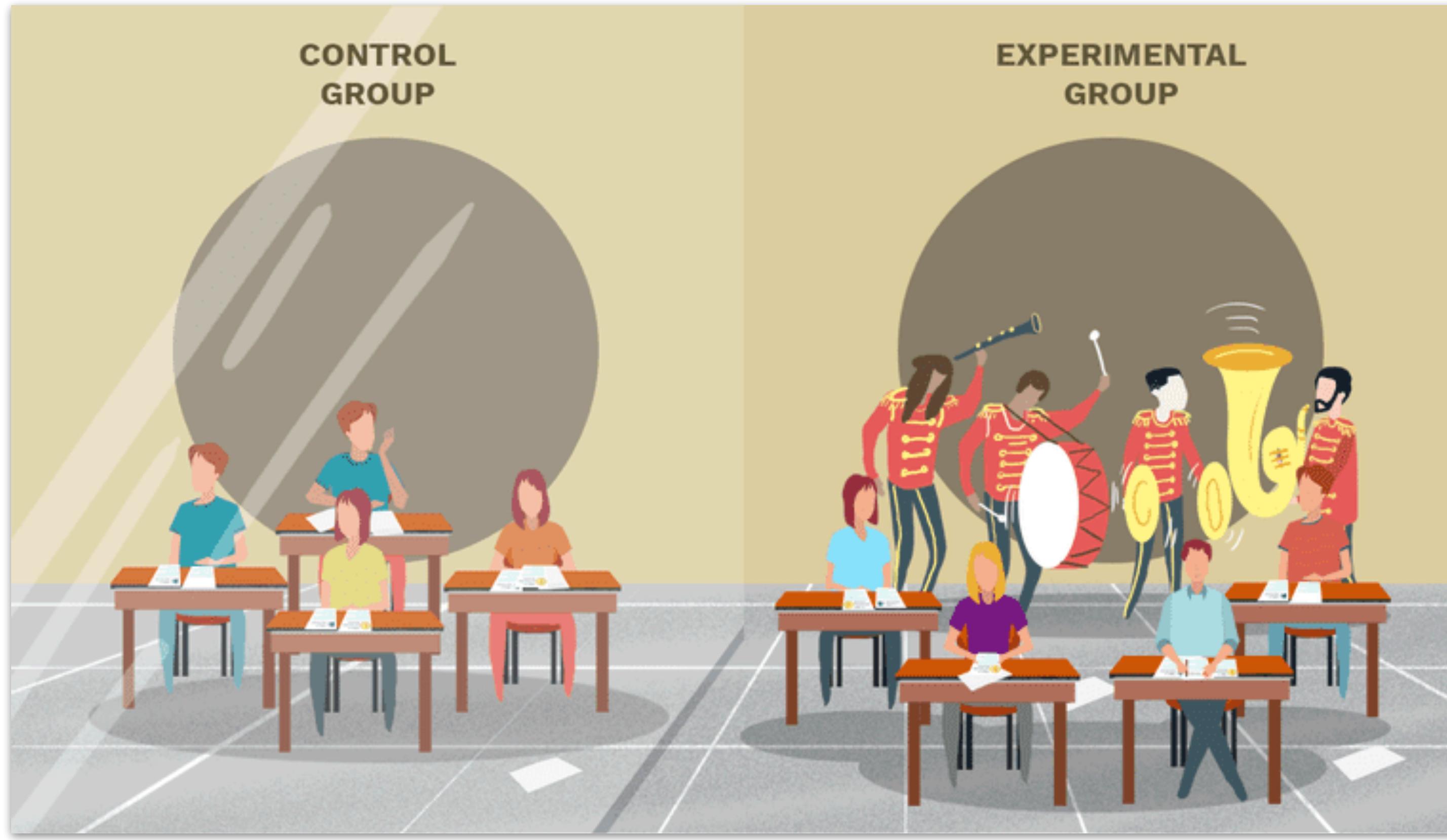
$$p(H | D) = \frac{p(D | H) \cdot p(H)}{p(D)} \quad \begin{aligned} H &= \text{Hypothesis} \\ D &= \text{Data} \end{aligned}$$

# 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  $\alpha$  level

# **Permutation test**

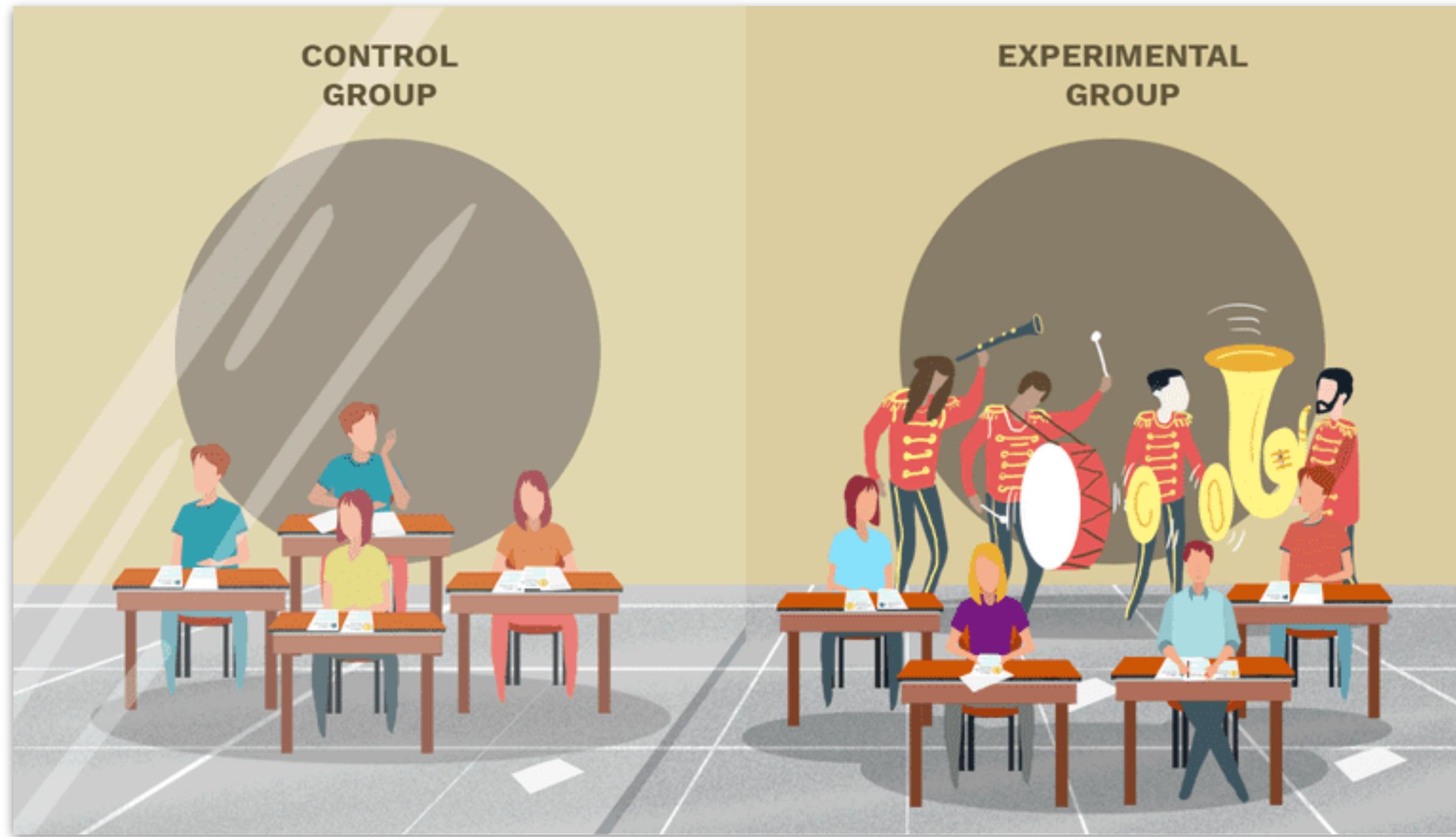
# Permutation test



## Research question:

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

# Permutation test



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

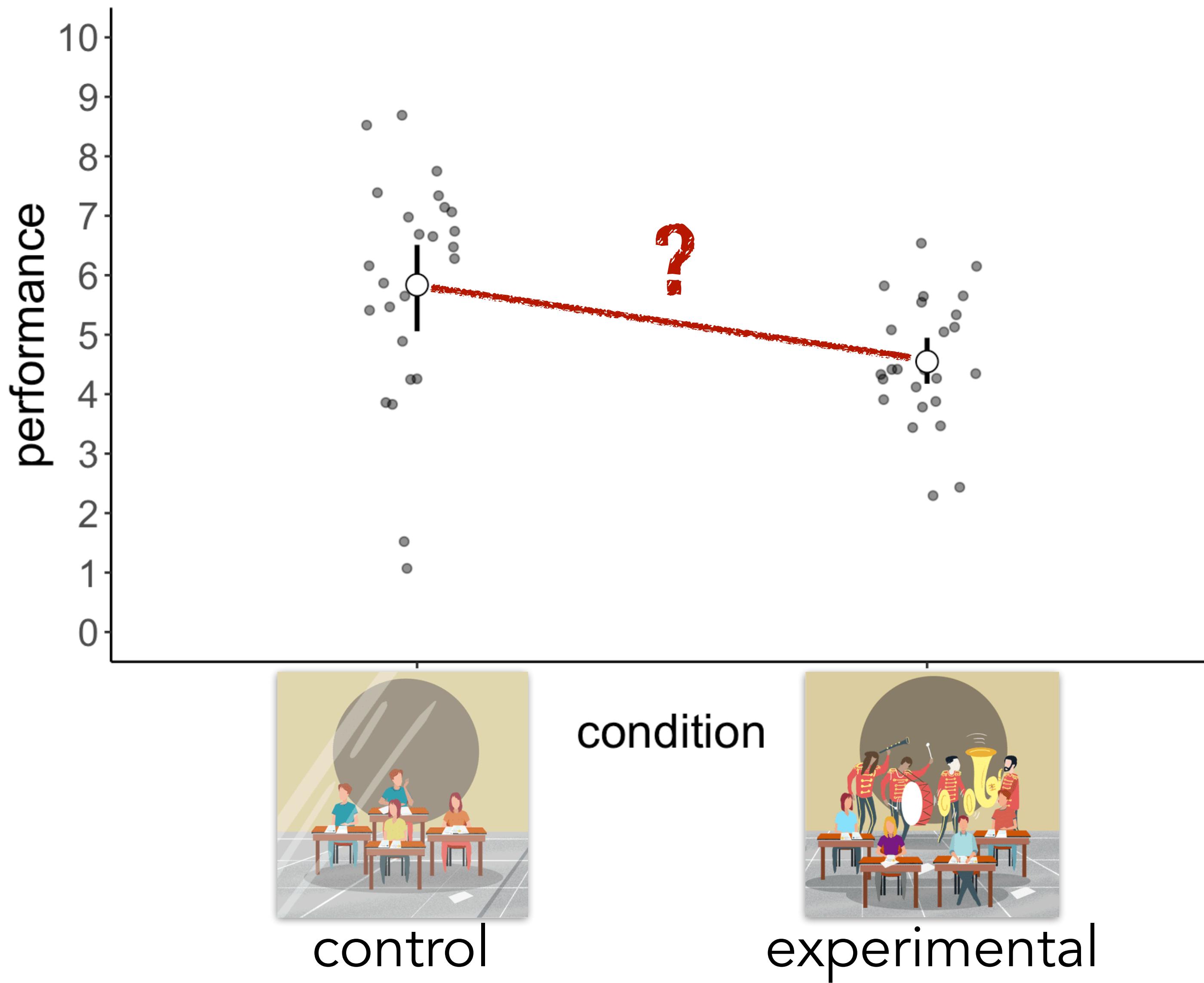
There is no difference between the control group and the experimental group

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

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

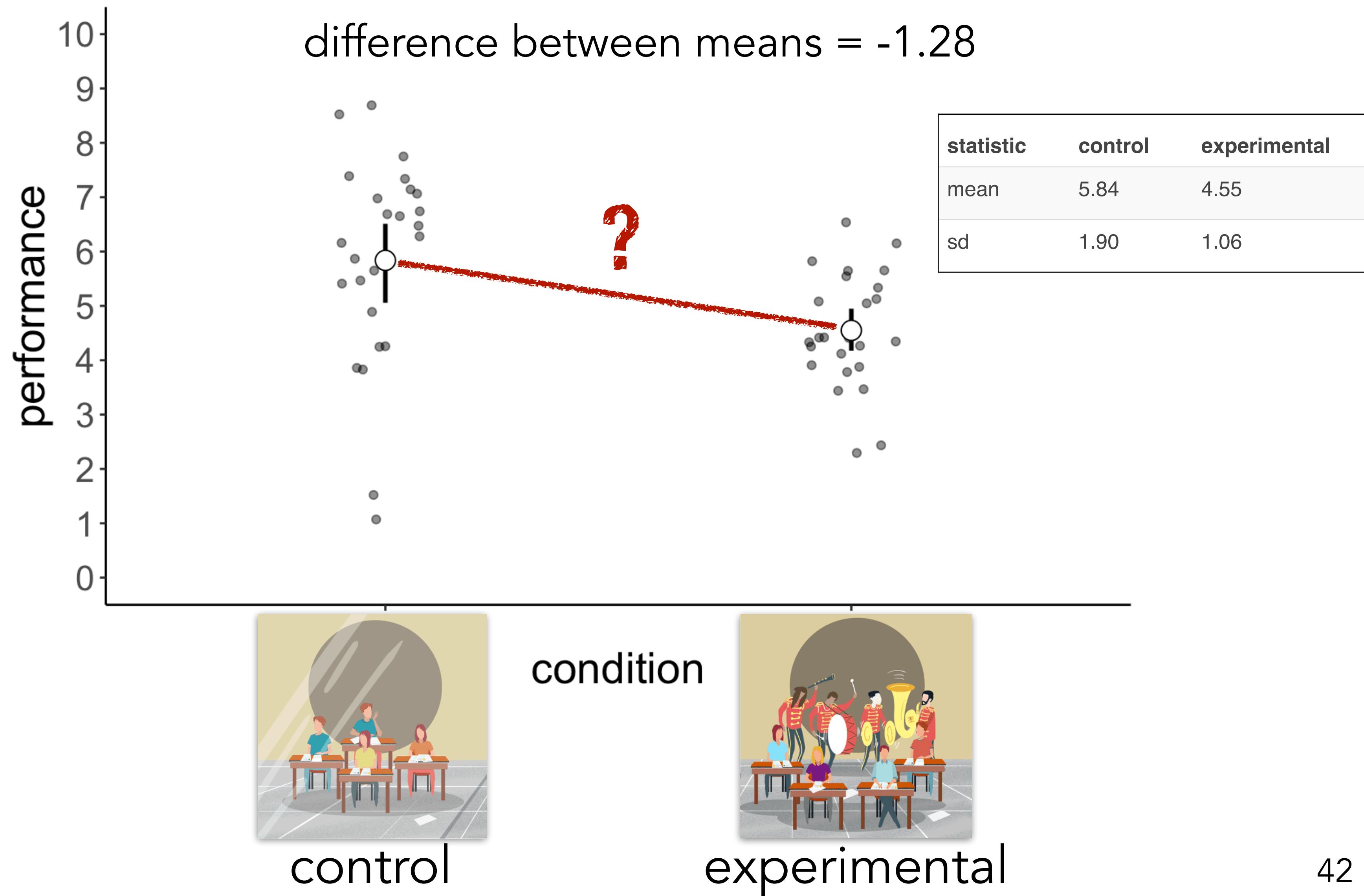
# Permutation test

Is the difference in performance statistically significant?



# Permutation test

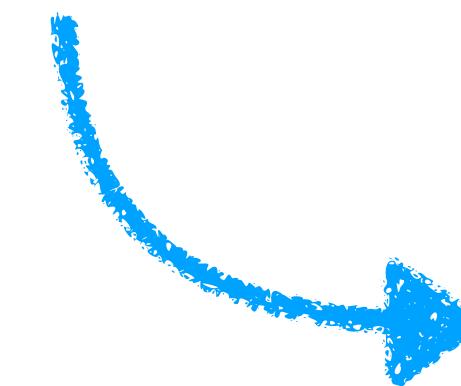
Is the difference in performance statistically significant?



# Permutation test

- **logic:**

- assuming our experimental manipulation made no difference, what would be the probability of observing the data that 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

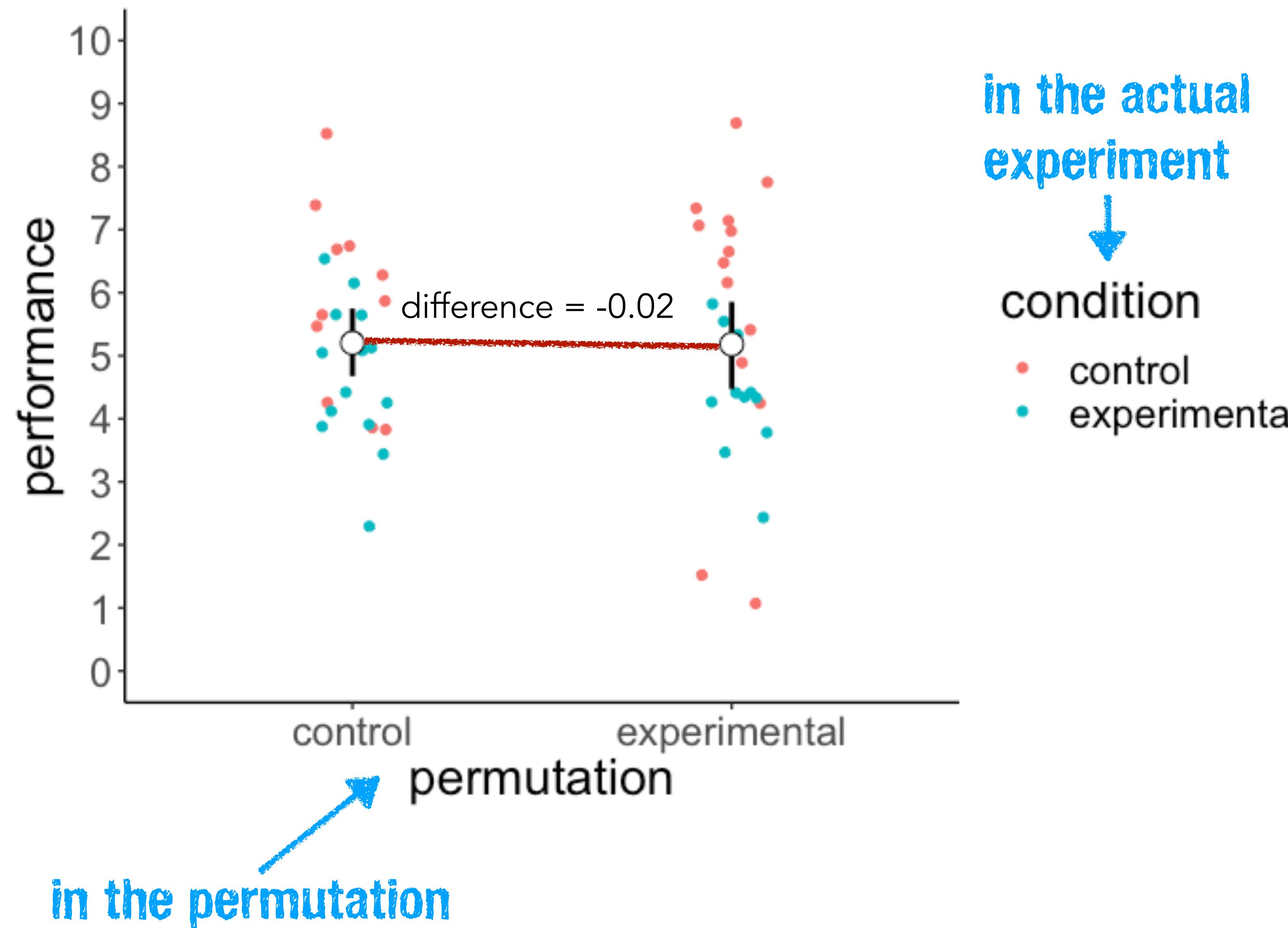
observed data

random permutation

| 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        |

| 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



# 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



2



3



| 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        |

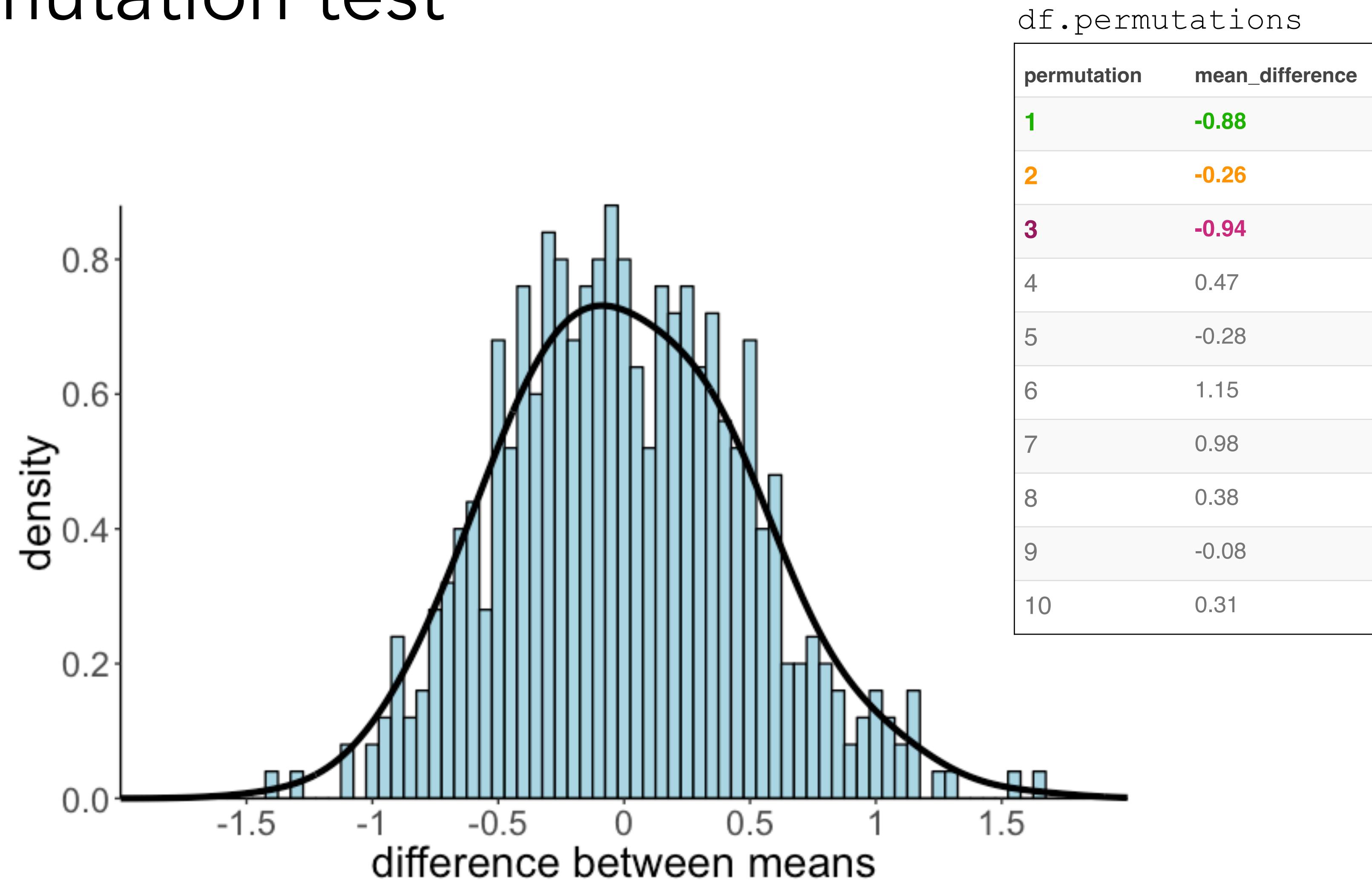
| participant | condition    | performance |
|-------------|--------------|-------------|
| 1           | experimental | 4.25        |
| 2           | control      | 5.87        |
| 3           | experimental | 3.83        |
| 4           | experimental | 8.69        |
| 5           | experimental |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

| 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            |

⋮

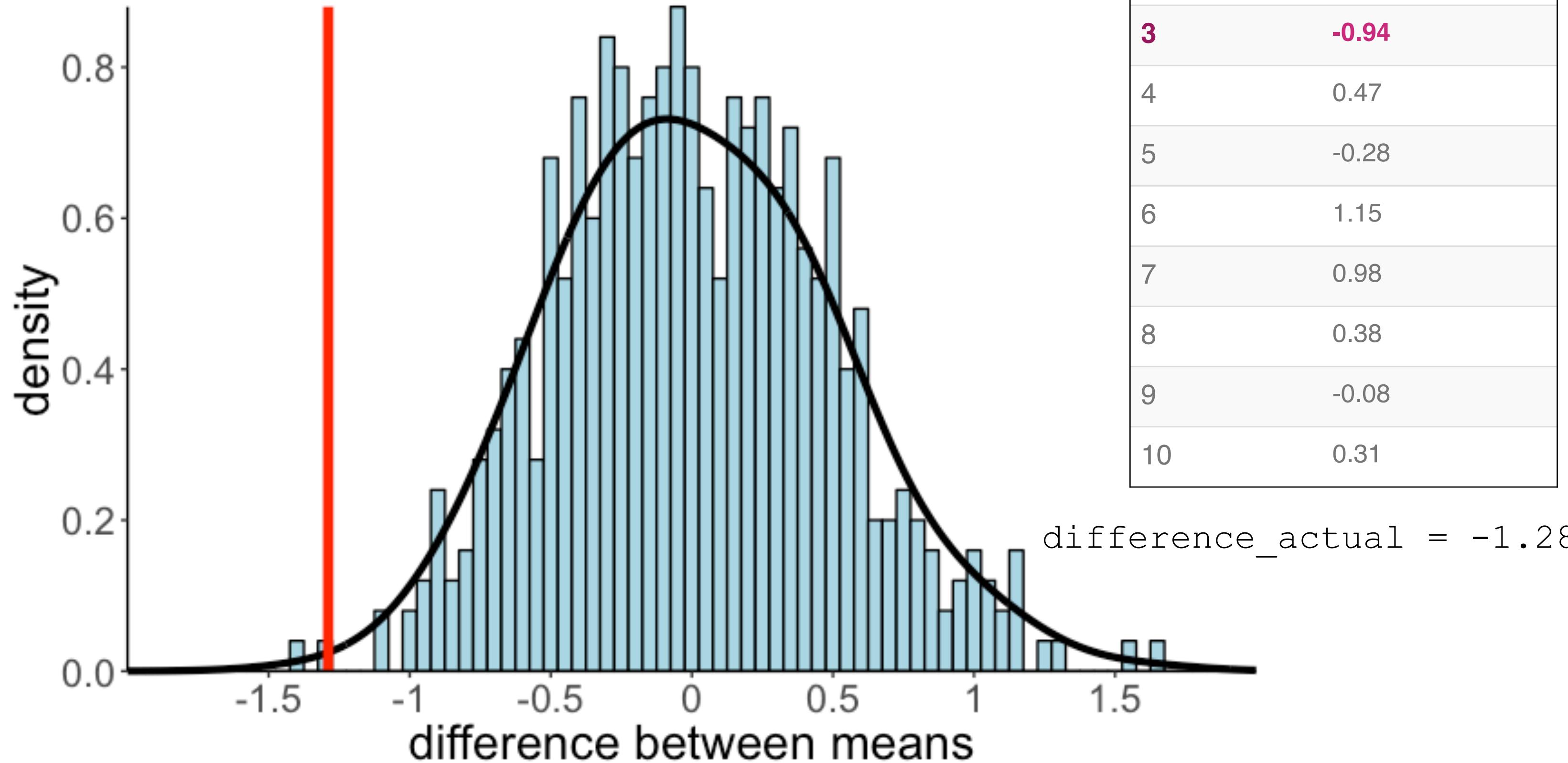
# Permutation test



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

# Permutation test

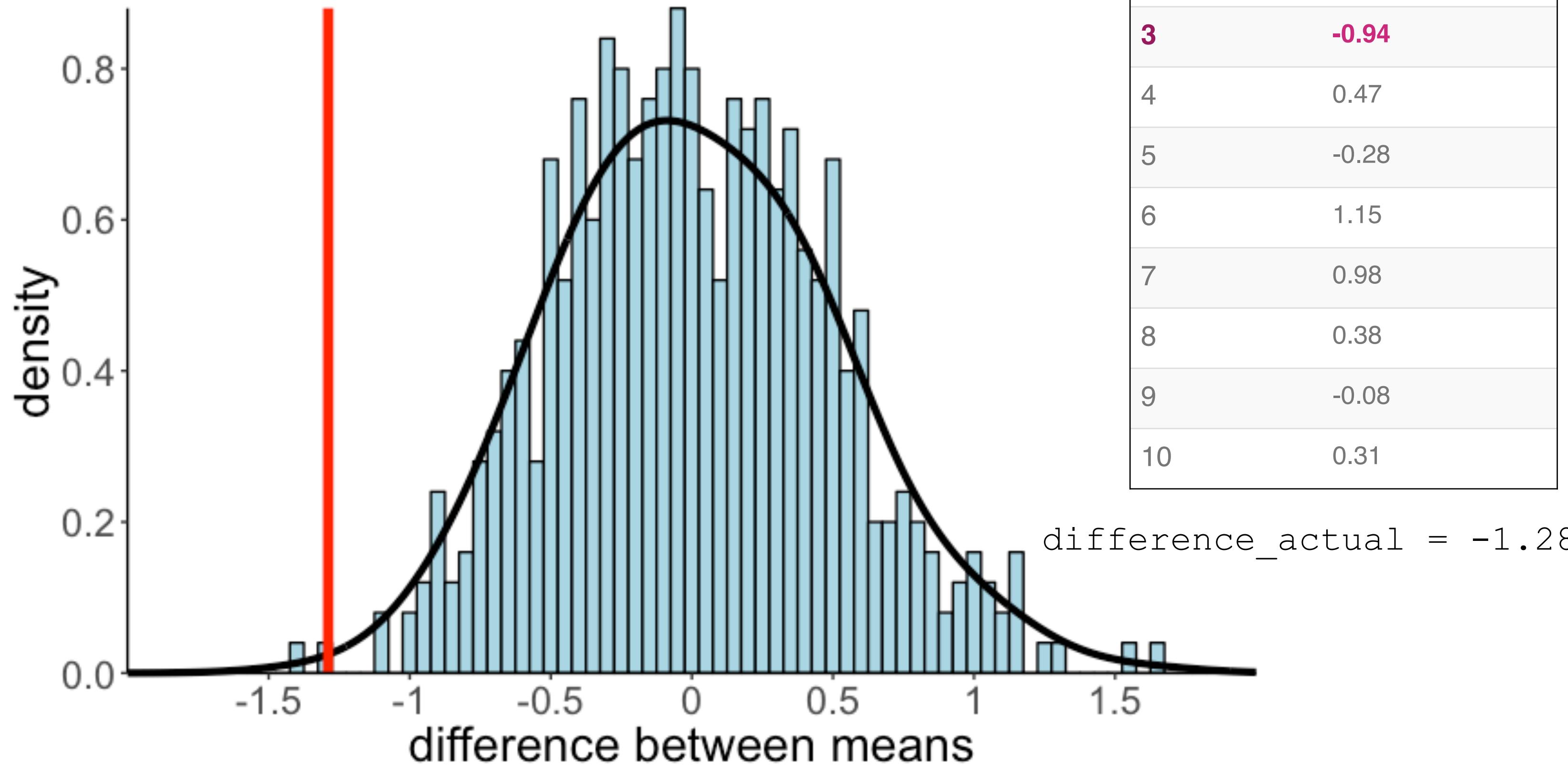
observed difference  
in our experiment



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

# Permutation test

observed difference  
in our experiment

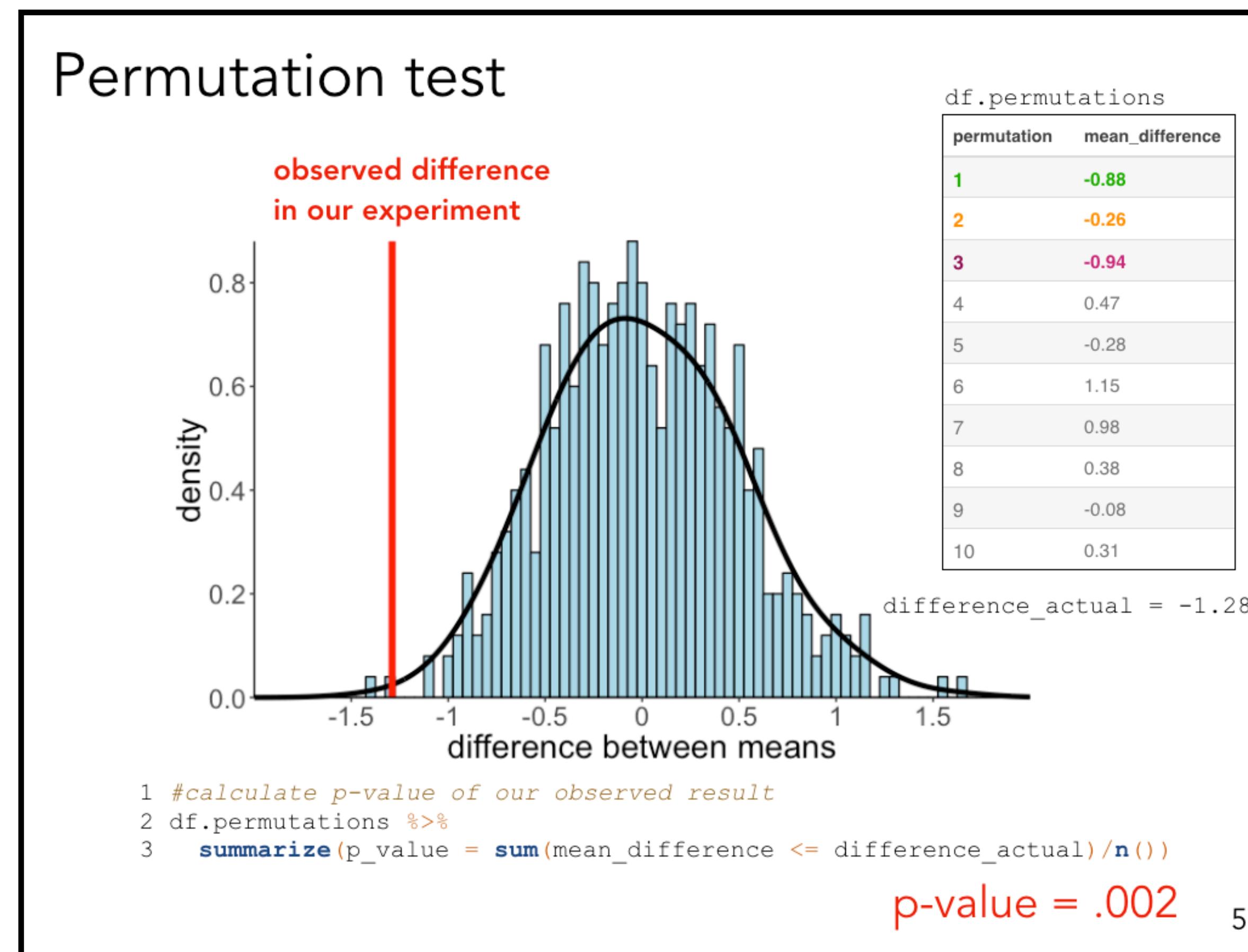


```
1 #calculate p-value of our observed result
2 df.permutations %>%
3   summarize(p_value = sum(mean_difference <= difference_actual) / n())
```

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

```
1 n_permutations = 500 ← set the number of permutations  
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 }
```

| participant | condition    | performance |
|-------------|--------------|-------------|
| 1           | experimental | 4.25        |
| 2           | control      | 5.87        |
| 3           | control      | 3.83        |
| 4           | experimental | 8.69        |
| 5           | experimental | 8.16        |
| 26          | control      | 4.42        |
| 27          | experimental | 4.27        |
| 28          | control      | 2.20        |
| 29          | control      | 3.78        |
| 30          | experimental | 5.13        |

| participant | condition    | performance |
|-------------|--------------|-------------|
| 1           | experimental | 4.25        |
| 2           | control      | 5.87        |
| 3           | experimental | 3.83        |
| 4           | experimental | 8.69        |
| 5           | experimental | 8.16        |
| 26          | control      | 4.42        |
| 27          | control      | 4.27        |
| 28          | control      | 2.29        |
| 29          | control      | 3.78        |
| 30          | experimental | 5.13        |

| participant | condition    | performance |
|-------------|--------------|-------------|
| 1           | control      | 4.25        |
| 2           | experimental | 5.87        |
| 3           | control      | 3.83        |
| 4           | experimental | 8.69        |
| 5           | control      | 8.16        |
| 26          | control      | 4.42        |
| 27          | experimental | 4.27        |
| 28          | control      | 2.20        |
| 29          | experimental | 3.78        |
| 30          | experimental | 5.13        |

calculate difference  
between group means

| 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 | 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            |

set the number of permutations

shuffle the condition labels

| participant | condition    | performance |
|-------------|--------------|-------------|
| 1           | control      | 4.25        |
| 2           | control      | 5.87        |
| 3           | control      | 3.83        |
| 4           | control      | 8.69        |
| 5           | control      | 8.16        |
| 26          | experimental | 4.42        |
| 27          | experimental | 4.27        |
| 28          | experimental | 2.29        |
| 29          | experimental | 3.78        |
| 30          | experimental | 5.13        |

| 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

```
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 }
12
13 # data frame with permutation results
14 df.permutations = tibble(
15   permutation = 1:n_permutations,
16   mean_difference = replicate(n = n_permutations, func_permutations(df.data))
17 )
```

df.permutations

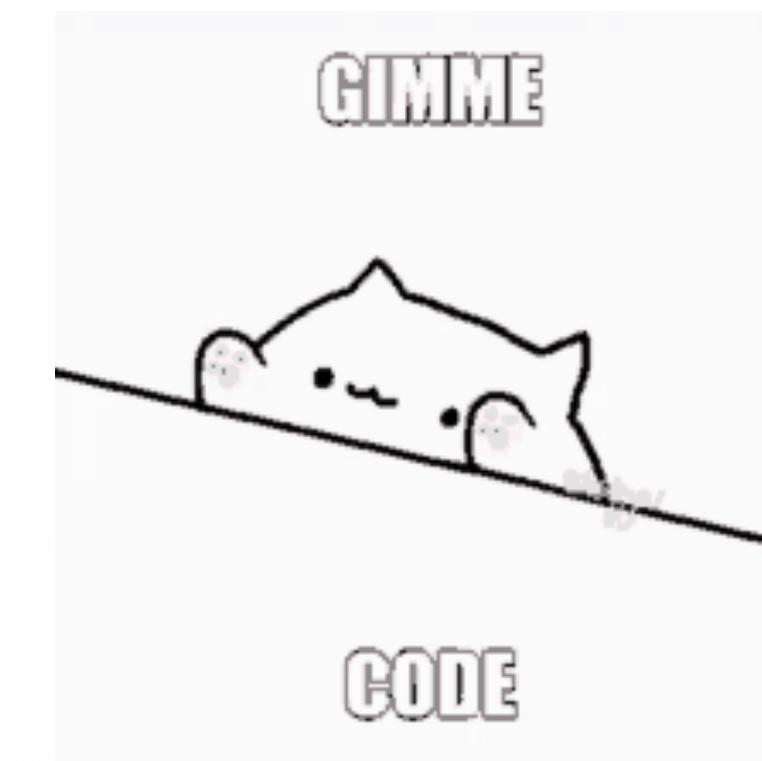
| permutation | mean_difference |
|-------------|-----------------|
| 1           | -0.88           |
| 2           | -0.26           |
| 3           | -0.94           |
| 4           | 0.47            |
| 5           | -0.28           |
| 6           | 1.15            |
| 7           | 0.98        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

run the `func_permutations()` function many times  
(instead of using a for loop)

# 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
- **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.”

# How to better understand!



08\_simulation2 - master - RStudio

simulation2.Rmd

```
1 ---  
2 title: "Class 8"  
3 author: "Tobias Gerstenberg"  
4 date: "January 24th, 2020"  
5 output:  
6   bookdown::html_document2:  
7   toc: true  
8   toc_depth: 4  
9   theme: cosmo  
10  highlight: tango  
11  pandoc_args: ["--number-offset=7"]  
12 ---  
13  
14 # Simulation 2  
15  
16 In which we figure out some key statistical concepts through simulation and plotting. On the menu we have:  
17 | Sampling distributions  
18 | - p-value  
19 | - Confidence interval  
20  
21 ## Load packages and set plotting theme  
22  
23 ``{r simulation2-01, include=FALSE}  
24 # run this code chunk once to make sure you have all the packages  
25 install.packages(c("janitor"))  
26 ``  
27  
28 ``{r simulation2-02, message=FALSE}  
29 library("knitr") # for knitting RMarkdown  
30 library("kableExtra") # for making nice tables  
31 library("janitor") # for cleaning column names  
32 library("tidyverse") # for wrangling, plotting, etc.  
33 ``  
34  
35 ``{r simulation2-03}  
36 theme_set(theme_classic() + #set the theme  
37   theme(text = element_text(size = 20))) #set the default text size  
38  
39 opts_chunk$set(comment = "",  
40   fig.show = "hold")  
41 ``  
42  
17:1 Simulation 2
```

Console

```
> ggplot(data = tibble(x = c(mean - 3 * sd, mean + 3 * sd),  
+   mapping = aes(x = x)) +  
+   stat_function(fun = ~ dnorm(., mean = mean, sd = sd),  
+     color = "black",  
+     size = 2) +  
+   geom_vline(xintercept = qnorm(c(0.025, 0.975), mean = mean, sd = sd),  
+     linetype = 2)  
> # labs(x = "performance")  
>
```

Environment

|                  |  |
|------------------|--|
| confidence_level | 0.95   |
| df.condition1    | 'kableExtra' chr "<table class='table table-striped' style='width: a..." |
| i                | 20L  |
| k                | 3  |
| mean             | 0  |
| n                | 10   |
| n_simulations    | 1000   |
| population_mean  | 3.5  |
| sample_n         | 20   |
| sample_size      | 1000   |
| sd               | 1  |

Plots

Packages

Help

Viewer

R: Subset rows using their positions

slice (dplyr)

R Documentation

### Subset rows using their positions

#### Description

slice() lets you index rows by their (integer) locations. It allows you to select, remove, and duplicate rows. It is accompanied by a number of helpers for common use cases:

- slice\_head() and slice\_tail() select the first or last rows.
- slice\_sample() randomly selects rows.
- slice\_min() and slice\_max() select rows with highest or lowest values of a variable.

If .data is a grouped\_df, the operation will be performed on each group, so that (e.g.) slice\_head(df, n = 5) will select the first five rows in each group.

#### Usage

```
slice(.data, ..., .preserve = FALSE)  
slice_head(.data, ..., n, prop)  
slice_tail(.data, ..., n, prop)  
slice_min(.data, order_by, ..., n, prop, with_ties = TRUE)  
slice_max(.data, order_by, ..., n, prop, with_ties = TRUE)  
slice_sample(.data, ..., n, prop, weight_by = NULL, replace = FALSE)
```

#### Arguments

.data A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dplyr). See Methods, below, for more details.

... For slice():<data-masking> Integer row values.

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).

**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 allows you to make claims about the population of interest. It is important to remember that the

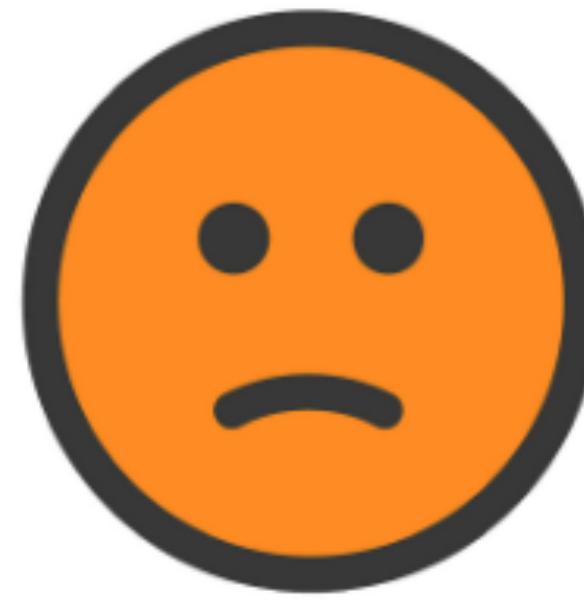
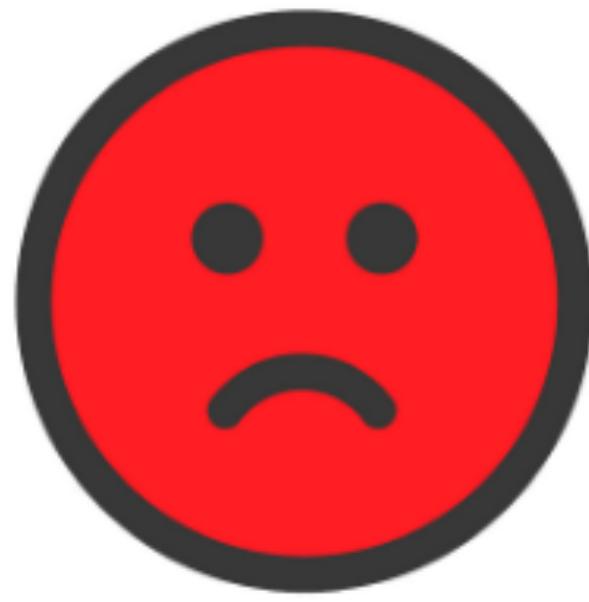
This course is part of these tracks:  
**Intro to Statistics with R**



**Jo Hardin**  
Professor at Pomona College

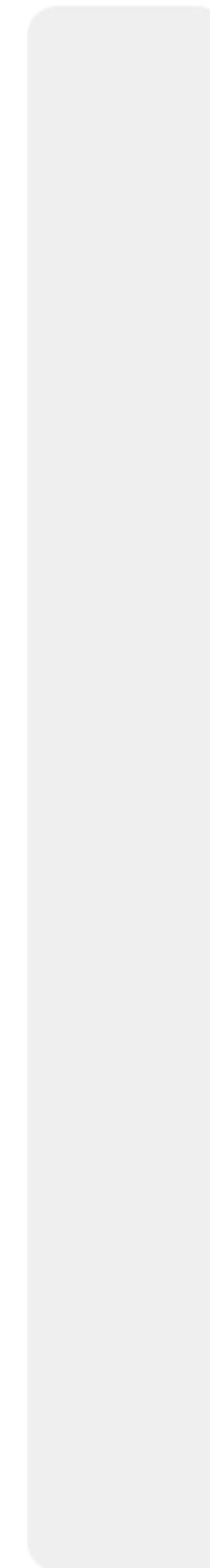
# **Feedback**

1

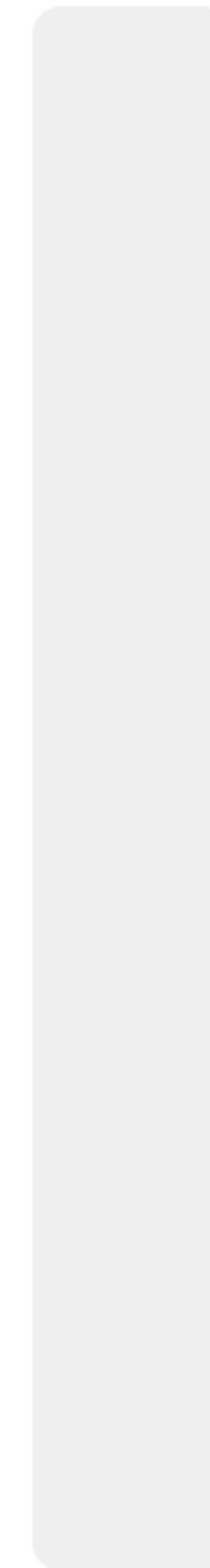


 0

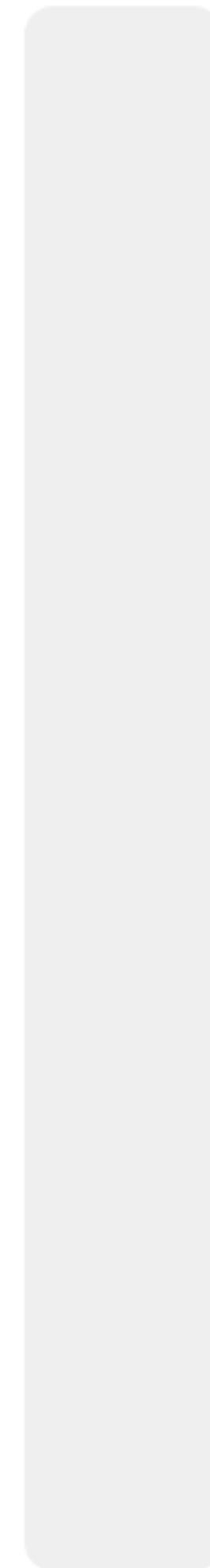
0%



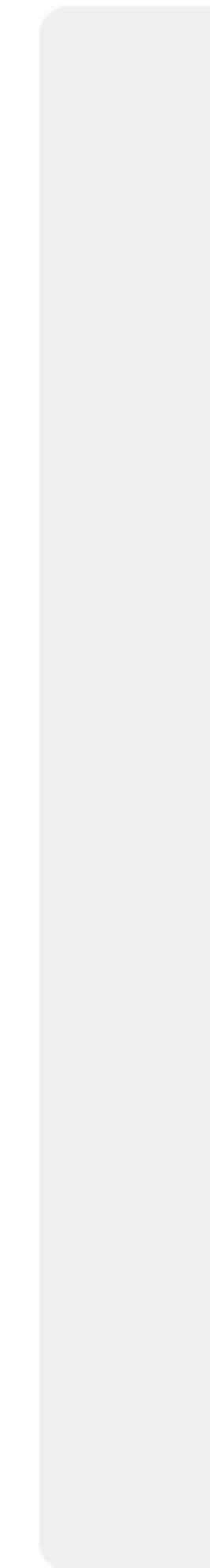
0%



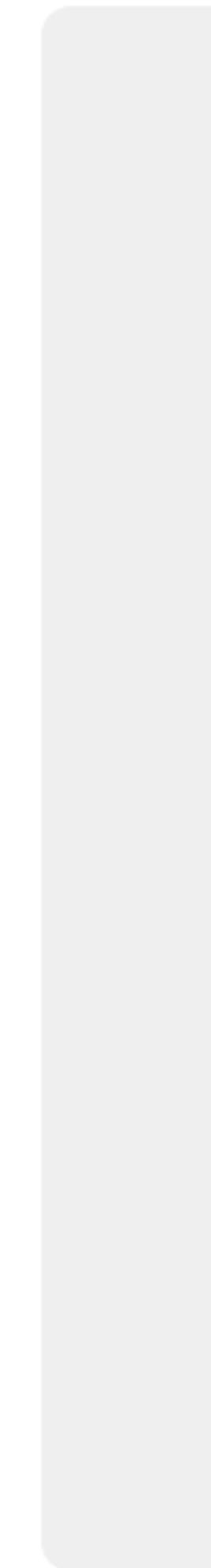
0%



0%



0%



much too slow

a little too slow

just right

a little too fast

much too fast



Nobody has responded yet.

Hang tight! Responses are coming in.

# Thank you!