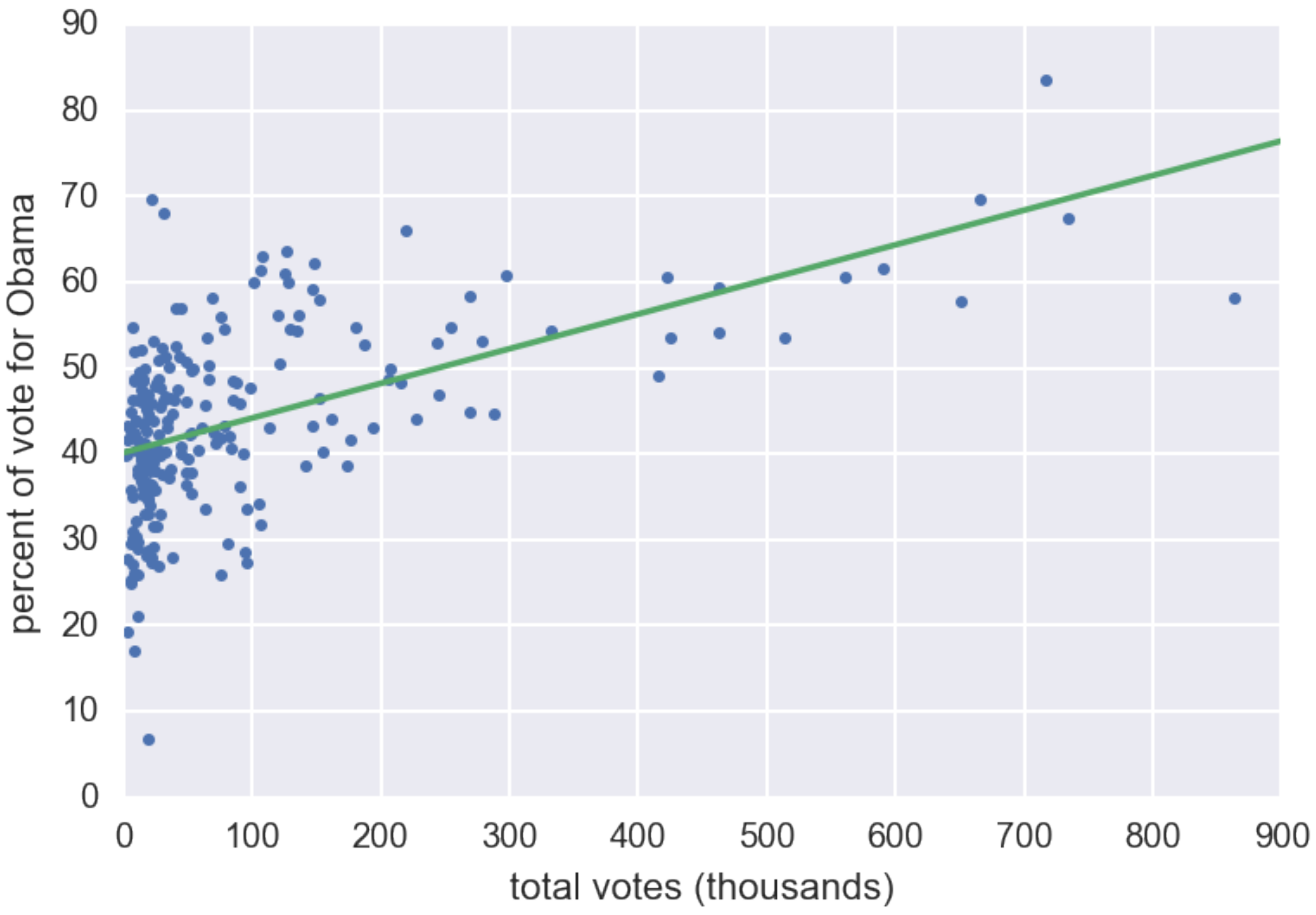
Formulating and simulating hypotheses

2008 US swing state election results



assume linear model

then, estimate parameters that are defined by that model

how reasonable it is our observed data are actually described by the model ??

this is the realm of hypothesis testing

Data retrieved from Data.gov (https://www.data.gov/)

OH and PA are similar states

Hypothesize that county level voting in these two states have identical probability distributions.

voting data to help test if this hypothesis.

--> access how reasonable the observed data are assuming the hypothesis is true

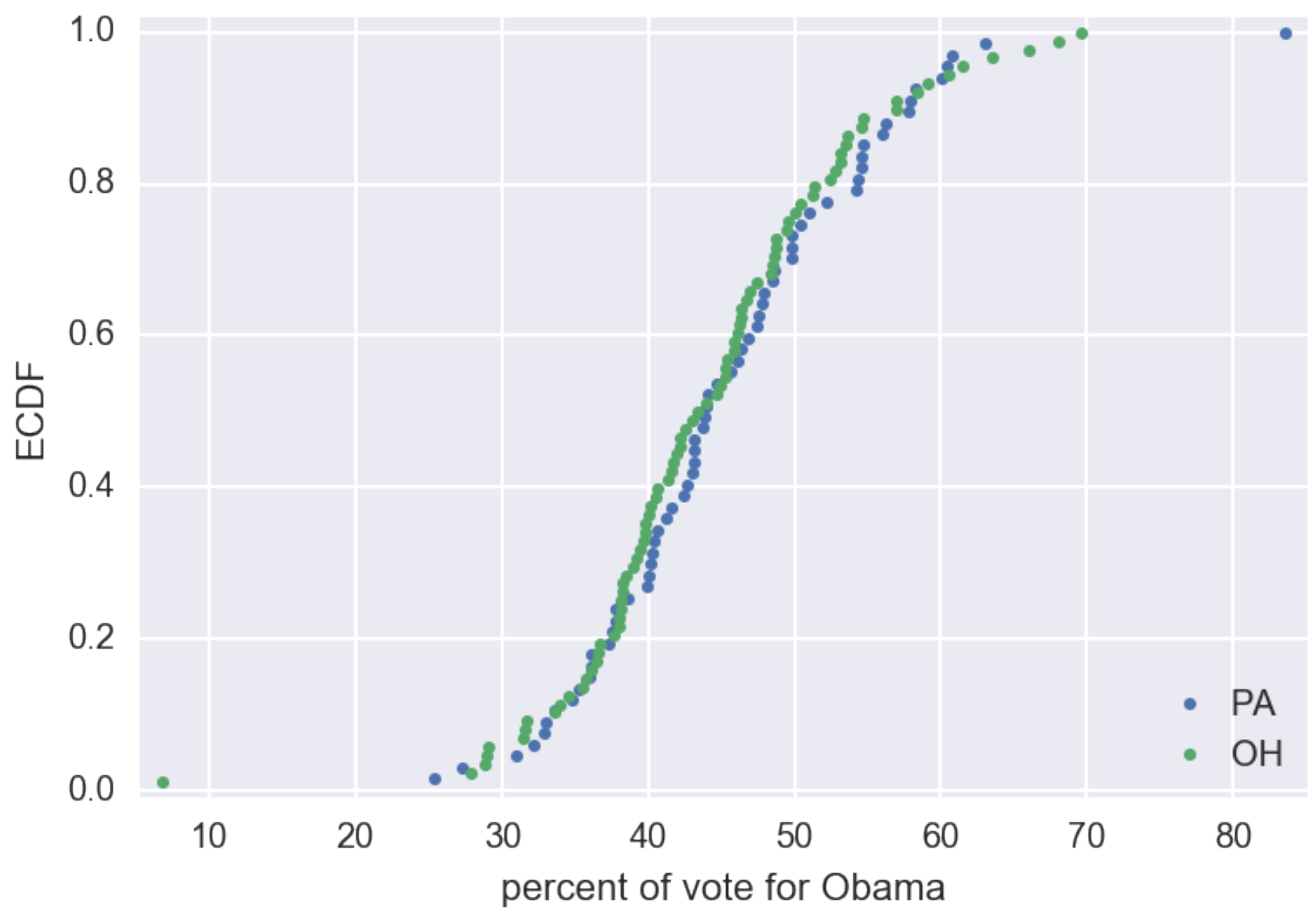
Hypothesis testing

- Assessment of how reasonable the observed data are assuming a hypothesis is true

Null hypothesis

- Another name for the hypothesis you are testing

ECDFs of swing state election results



PA seems to be slightly more toward Obama in the middle part of the ECDFs, but not much.

can not really draw a conclusion here

Data retrieved from Data.gov (https://www.data.gov/)

Percent vote for Obama

compare some summary statistics

	PA	OH	PA — OH difference
mean	45.5%	44.3%	1.2%
median	44.0%	43.7%	0.4%
standard deviation	9.8%	9.9%	—0.1%

---> tough case

eyeballing the data is not enough

Data retrieved from Data.gov (https://www.data.gov/)

Simulating the hypothesis

to resolve this issue, simulate what the data would like if the county level voting trends in the 2 states were identically distributed.

60.08,	40.64,	36.07,	41.21,	31.04,	43.78,	44.08,	46.85,
44.71,	46.15,	63.10,	52.20,	43.18,	40.24,	39.92,	47.87,
37.77,	40.11,	49.85,	48.61,	38.62,	54.25,	34.84,	47.75,
43.82,	55.97,	58.23,	42.97,	42.38,	36.11,	37.53,	42.65,
50.96,	47.43,	56.24,	45.60,	46.39,	35.22,	48.56,	32.97,
57.88,	36.05,	37.72,	50.36,	32.12,	41.55,	54.66,	57.81,
54.58,	32.88,	54.37,	40.45,	47.61,	60.49,	43.11,	27.32,
44.03,	33.56,	37.26,	54.64,	43.12,	25.34,	49.79,	83.56,
40.09,	60.81,	49.81,	56.94,	50.46,	65.99,	45.88,	42.23,
45.26,	57.01,	53.61,	59.10,	61.48,	43.43,	44.69,	54.59,
48.36,	45.89,	48.62,	43.92,	38.23,	28.79,	63.57,	38.07,
40.18,	43.05,	41.56,	42.49,	36.06,	52.76,	46.07,	39.43,
39.26,	47.47,	27.92,	38.01,	45.45,	29.07,	28.94,	51.28,
50.10,	39.84,	36.43,	35.71,	31.47,	47.01,	40.10,	48.76,
31.56,	39.86,	45.31,	35.47,	51.38,	46.33,	48.73,	41.77,
41.32,	48.46,	53.14,	34.01,	54.74,	40.67,	38.96,	46.29,
38.25,	6.80,	31.75,	46.33,	44.90,	33.57,	38.10,	39.67,
40.47,	49.44,	37.62,	36.71,	46.73,	42.20,	53.16,	52.40,
58.36,	68.02,	38.53,	34.58,	69.64,	60.50,	53.53,	36.54,
49.58,	41.97,	38.11					

Pennsylvania

now, putting the vote for all PA (67 counties) and OH (88 counties together.

ignore what state they belong to.

Ohio

Data retrieved from Data.gov (https://www.data.gov/)

Simulating the hypothesis

```
60.08,  40.64,  36.07,  41.21,  31.04,  43.78,  44.08,  46.85,
44.71,  46.15,  63.10,  52.20,  43.18,  40.24,  39.92,  47.87,
37.77,  40.11,  49.85,  48.61,  38.62,  54.25,  34.84,  47.75,
43.82,  55.97,  58.23,  42.97,  42.38,  36.11,  37.53,  42.65,
50.96,  47.43,  56.24,  45.60,  46.39,  35.22,  48.56,  32.97,
57.88,  36.05,  37.72,  50.36,  32.12,  41.55,  54.66,  57.81,
54.58,  32.88,  54.37,  40.45,  47.61,  60.49,  43.11,  27.32,
44.03,  33.56,  37.26,  54.64,  43.12,  25.34,  49.79,  83.56,
40.09,  60.81,  49.81,  56.94,  50.46,  65.99,  45.88,  42.23,
45.26,  57.01,  53.61,  59.10,  61.48,  43.43,  44.69,  54.59,
48.36,  45.89,  48.62,  43.92,  38.23,  28.79,  63.57,  38.07,
40.18,  43.05,  41.56,  42.49,  36.06,  52.76,  46.07,  39.43,
39.26,  47.47,  27.92,  38.01,  45.45,  29.07,  28.94,  51.28,
50.10,  39.84,  36.43,  35.71,  31.47,  47.01,  40.10,  48.76,
31.56,  39.86,  45.31,  35.47,  51.38,  46.33,  48.73,  41.77,
41.32,  48.46,  53.14,  34.01,  54.74,  40.67,  38.96,  46.29,
38.25,   6.80,  31.75,  46.33,  44.90,  33.57,  38.10,  39.67,
40.47,  49.44,  37.62,  36.71,  46.73,  42.20,  53.16,  52.40,
58.36,  68.02,  38.53,  34.58,  69.64,  60.50,  53.53,  36.54,
49.58,  41.97,  38.11
```

randomly scrambel the ordering of the counties

Data retrieved from Data.gov (https://www.data.gov/)

Simulating the hypothesis

```
59.10,  38.62,  51.38,  60.49,   6.80,  41.97,  48.56,  37.77,
48.36,  54.59,  40.11,  57.81,  45.89,  83.56,  40.64,  46.07,
28.79,  55.97,  33.57,  42.23,  48.61,  44.69,  39.67,  57.88,
48.62,  54.66,  54.74,  48.46,  36.07,  43.92,  49.85,  53.53,
48.76,  41.77,  36.54,  47.01,  52.76,  49.44,  34.58,  40.24,
44.08,  46.29,  49.81,  69.64,  60.50,  27.32,  45.60,  63.10,
35.71,  39.86,  40.67,  65.99,  50.46,  37.72,  50.96,  42.49,
31.56,  38.23,  37.26,  41.21,  37.53,  46.85,  44.03,  41.32,
45.88,  40.45,  32.12,  35.22,  49.79,  43.12,  43.18,  45.45,
25.34,  46.73,  44.90,  56.94,  58.23,  39.84,  36.05,  43.05,
38.25,  40.47,  31.04,  54.25,  46.15,  57.01,  52.20,  47.75,
36.06,  47.61,  51.28,  43.43,  42.97,  38.01,  54.64,  45.26,
47.47,  34.84,  49.58,  48.73,  29.07,  54.58,  27.92,  34.01,
38.07,  31.47,  36.11,  39.26,  41.56,  52.40,  40.18,  47.87,
46.33,  46.39,  43.11,  38.53,  33.56,  42.65,  68.02,  35.47,
40.09,  36.43,  36.71,  60.08,  50.36,  39.43,  28.94,  58.36,
42.20,  47.43,  44.71,  43.78,  39.92,  37.62,  63.57,  53.61,
40.10,  46.33,  53.16,  32.88,  38.96,  41.55,  56.24,  38.11,
42.38,  38.10,  43.82,  45.31,  60.81,  54.37,  53.14,  32.97,
61.48,  50.10,  31.75
```

Simulating the hypothesis

re-label the first 67 to be PA and remaining ones to be OH.

redid the election as if there was no difference btw PA and OH

```
59.10,  38.62,  51.38,  60.49,   6.80,  41.97,  48.56,  37.77,
48.36,  54.59,  40.11,  57.81,  45.89,  83.56,  40.64,  46.07,
28.79,  55.97,  33.57,  42.23,  48.61,  44.69,  39.67,  57.88,
48.62,  54.66,  54.74,  48.46,  36.07,  43.92,  49.85,  53.53,
48.76,  41.77,  36.54,  47.01,  52.76,  49.44,  34.58,  40.24,
44.08,  46.29,  49.81,  69.64,  60.50,  27.32,  45.60,  63.10,
35.71,  39.86,  40.67,  65.99,  50.46,  37.72,  50.96,  42.49,
31.56,  38.23,  37.26,  41.21,  37.53,  46.85,  44.03,  41.32,
45.88,  40.45,  32.12,  35.22,  49.79,  43.12,  43.18,  45.45,
25.34,  46.73,  44.90,  56.94,  58.23,  39.84,  36.05,  43.05,
38.25,  40.47,  31.04,  54.25,  46.15,  57.01,  52.20,  47.75,
36.06,  47.61,  51.28,  43.43,  42.97,  38.01,  54.64,  45.26,
47.47,  34.84,  49.58,  48.73,  29.07,  54.58,  27.92,  34.01,
38.07,  31.47,  36.11,  39.26,  41.56,  52.40,  40.18,  47.87,
46.33,  46.39,  43.11,  38.53,  33.56,  42.65,  68.02,  35.47,
40.09,  36.43,  36.71,  60.08,  50.36,  39.43,  28.94,  58.36,
42.20,  47.43,  44.71,  43.78,  39.92,  37.62,  63.57,  53.61,
40.10,  46.33,  53.16,  32.88,  38.96,  41.55,  56.24,  38.11,
42.38,  38.10,  43.82,  45.31,  60.81,  54.37,  53.14,  32.97,
61.48,  50.10,  31.75
```

"Pennsylvania"

"Ohio"

Data retrieved from Data.gov (https://www.data.gov/)

Permutation

the technique of scrambling the oder of an array ---> permutation

it is at the heart of simulating a null hypothesis were we assume 2 quantities are identically distributed.

- Random reordering of entries in an array

Generating a permutation sample

```
In [1]: import numpy as np

In [2]: dem_share_both = np.concatenate(
   ...:                    (dem_share_PA, dem_share_OH))

In [3]: dem_share_perm = np.random.permutation(dem_share_both)

In [4]: perm_sample_PA = dem_share_perm[:len(dem_share_PA)]

In [5]: perm_sample_OH = dem_share_perm[len(dem_share_PA):]
```

make a single array with all of the counties

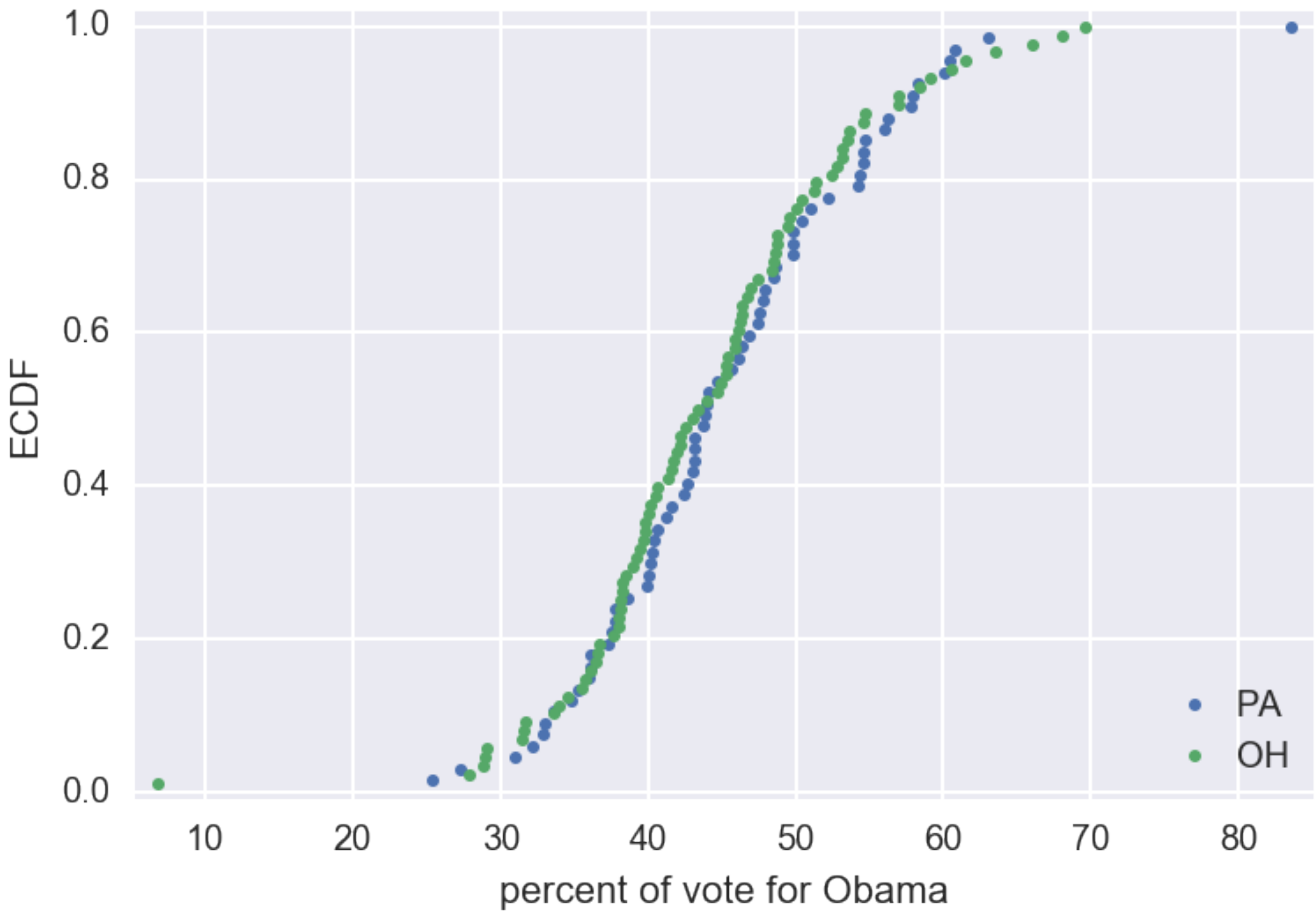
conveniently permute the entries of the array

permutation samples

Let's practice!

Test statistics and p-values

Are OH and PA different?



Null hypothesis that the county-level voting is identically distributed between the two states.

Data retrieved from Data.gov (https://www.data.gov/)

Hypothesis testing

- Assessment of how reasonable the observed data are assuming a hypothesis is true

what about the data do we assess and how do we quantify the assessment?

--> concept of a test statistic

Test statistic

- A single number that can be computed from observed data and from data you simulate under the null hypothesis
- It serves as a basis of comparison between the two

what the hypothesis predicts vs what we actually observed

Permutation replicate

is the value of a test statistic computed from a permutation sample

```
In [1]: np.mean(perm_sample_PA) - np.mean(perm_sample_OH)
Out[1]: 1.122220149253728

In [2]: np.mean(dem_share_PA) - np.mean(dem_share_OH) # orig. data
Out[2]: 1.1582360922659518
```

did not quite get as big of a diff in means than what was observed in the orginal data

if they are identical, they should have the same mean vote share.

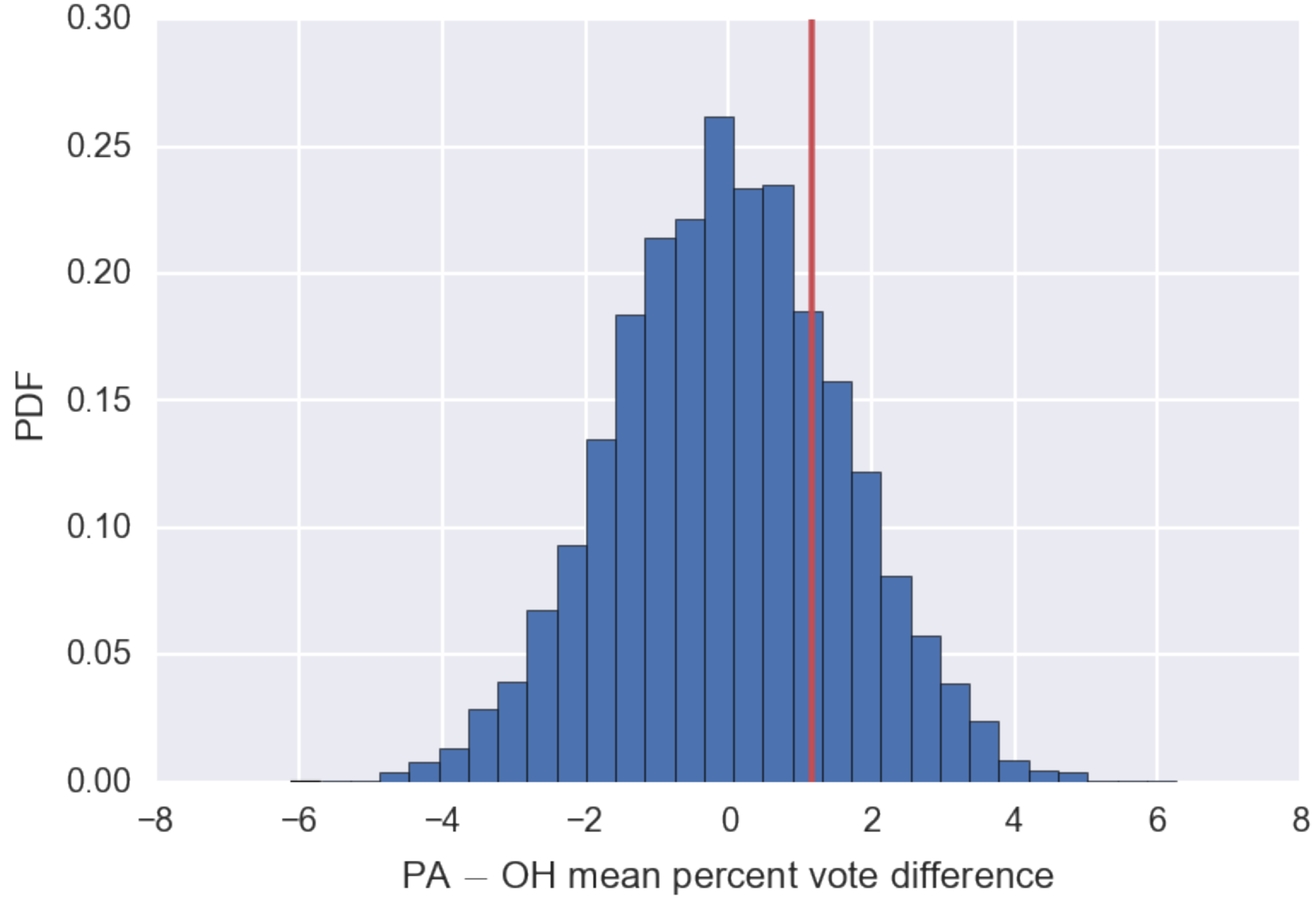
so, the difference in mean vote share should be zero.

-> choose the difference in means as our test statistic.

Mean vote difference under null hypothesis

1.16%

actual mean



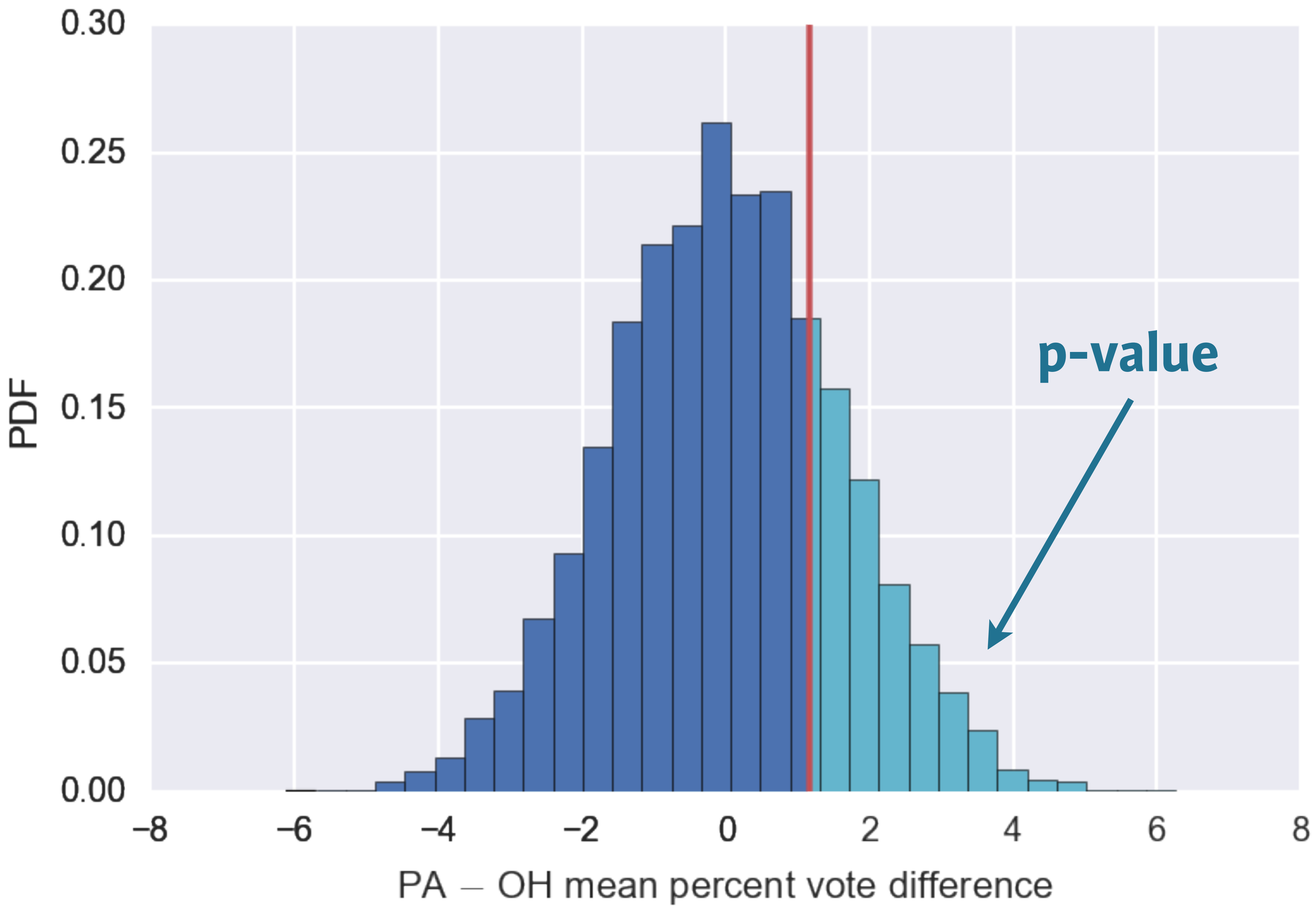
"redo" the election 10000 times under null hypo by generating many permutation replicates.

plot histogram of all the perm replicates.

the diff of means from elections simulated under the null hypo lies somewhere btw -4 and 4%.

Data retrieved from Data.gov (https://www.data.gov/)

Mean vote difference under null hypothesis



if tally up the area of the histogram that is to the right of the red line, we get that about 23% of the simulated elections had at least a 1.16% diff or greater.
This value (0.23) is p-value.
It is the probability of getting at least 1.16% difference in the mean vote share assuming the states have identically distributed voting.

it happened 23% of the time under the null hypo

Data retrieved from Data.gov (https://www.data.gov/)

p-value

p-value is only meaningful if null hypo is clearly stated, along with the test statistic used to evaluate it.

- The probability of obtaining a value of your test statistic that is at least as extreme as what was observed, under the assumption the null hypothesis is true
- **NOT** the probability that the null hypothesis is true

when the p-value is small, it is often said that data are statistically significantly different than what we would observe under the null hypo.

Statistical significance

when the p-value is small, it is often said that data are statistically significantly different than what we would observe under the null hypo.

- Determined by the smallness of a p-value

Null hypothesis significance testing (NHST)

- Another name for what we are doing in this chapter statistical significance ≠ practical significance

(that is low p-values)

whether or not the difference of the data from the null hypo

matters for practical considerations.

Let’s practice!

HW: -->

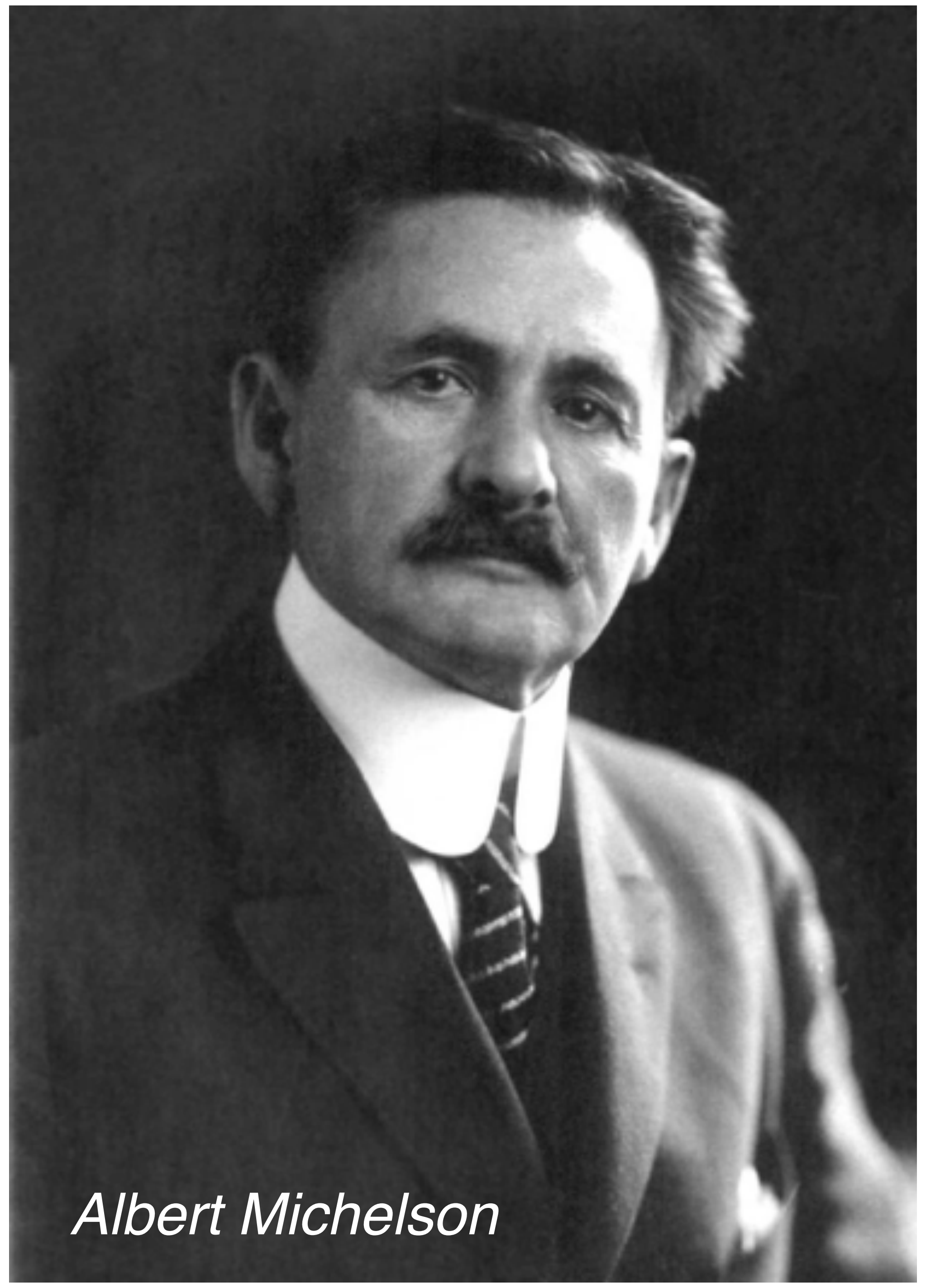
The p-value tells you that there is about a 0.6% chance that you would get the difference of means observed in the experiment if frogs were exactly the same. A p-value below 0.01 is typically said to be "statistically significant,", but: warning! warning! warning! You have computed a p-value; it is a number. I encourage you not to distill it to a yes-or-no phrase. p = 0.006 and p = 0.000000006 are both said to be "statistically significant," but they are definitely not the same!

Bootstrap hypothesis tests

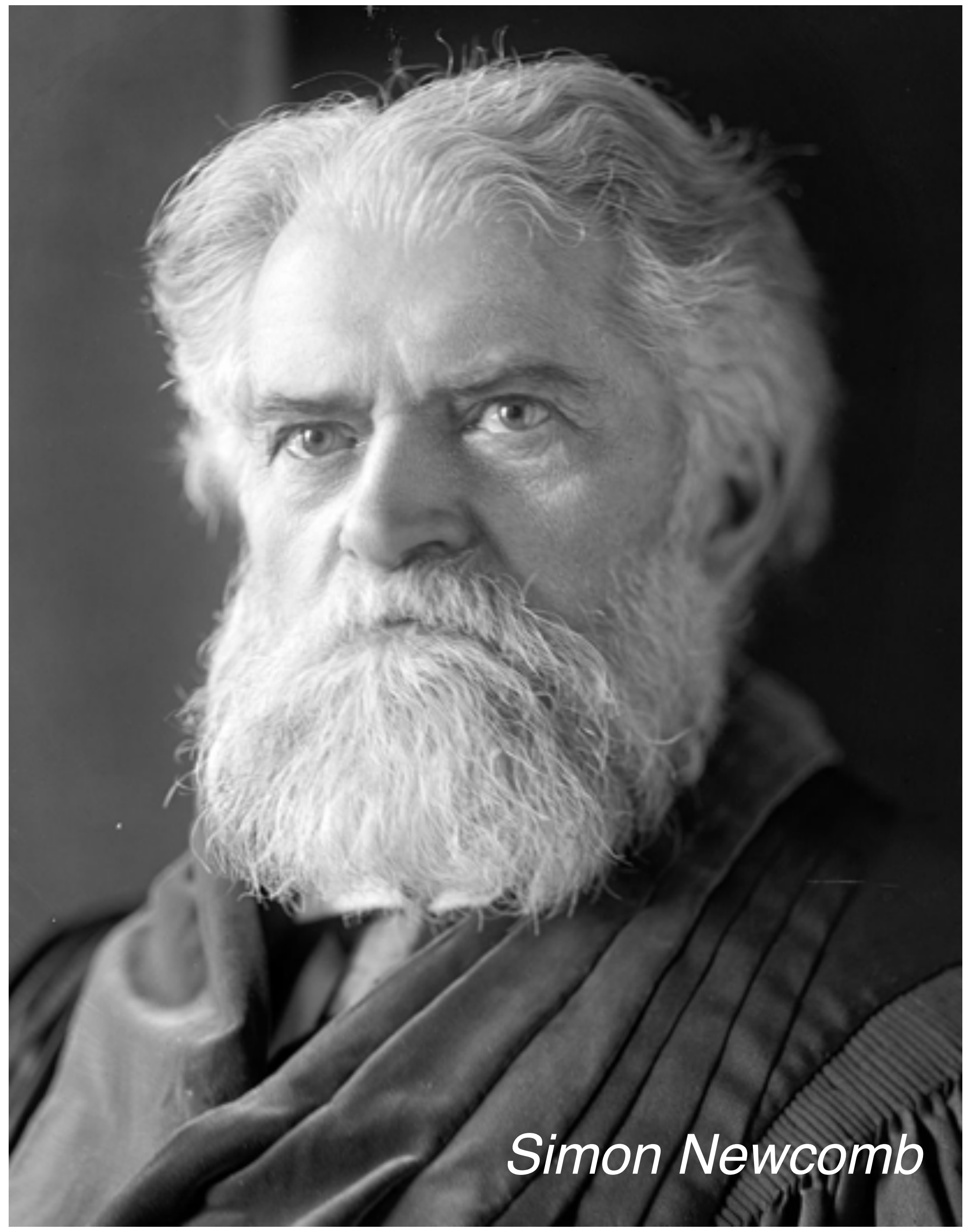
Pipeline for hypothesis testing

- Clearly state the null hypothesis
- Define your test statistic
- Generate many sets of simulated data assuming the null hypothesis is true
- Compute the test statistic for each simulated data set
- The p-value is the fraction of your simulated data sets for which the test statistic is at least as extreme as for the real data

Michelson and Newcomb: speed of light pioneers



299,852 km/s



299,860 km/s

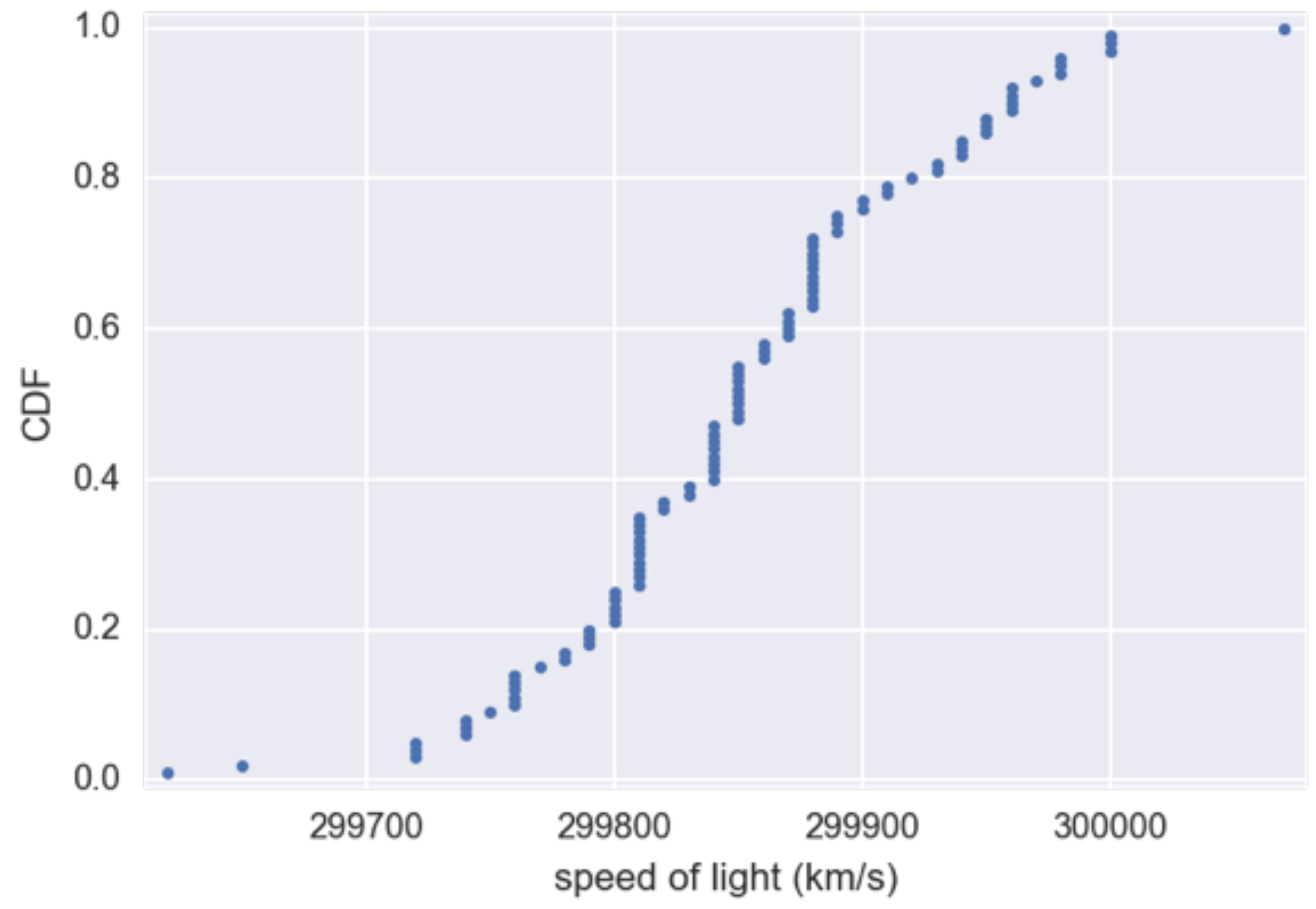
the thing is: we only have Newcombe's mean and none of his data points

Michelson image: public domain, Smithsonian

Newcomb image: US Library of Congress

The data we have

Michelson:



Newcomb:

mean = 299,860 km/s

the question is: could Michelson have gotten the data set he did if the true mean speed of light in his exp was equal to Newcombe's?

Data: Michelson, 1880

Null hypothesis

- The true mean speed of light in Michelson’s experiments was actually Newcomb's reported value

*** say mean speed of light in Michelson's exp, think the mean Michelson would have gotten had done his exp lots and lots of times

comparing a data set with a value, a permutation test is not applicable.
--> simulate the situation in which the true mean speed of light in Michelson's exp is Newcomb's value

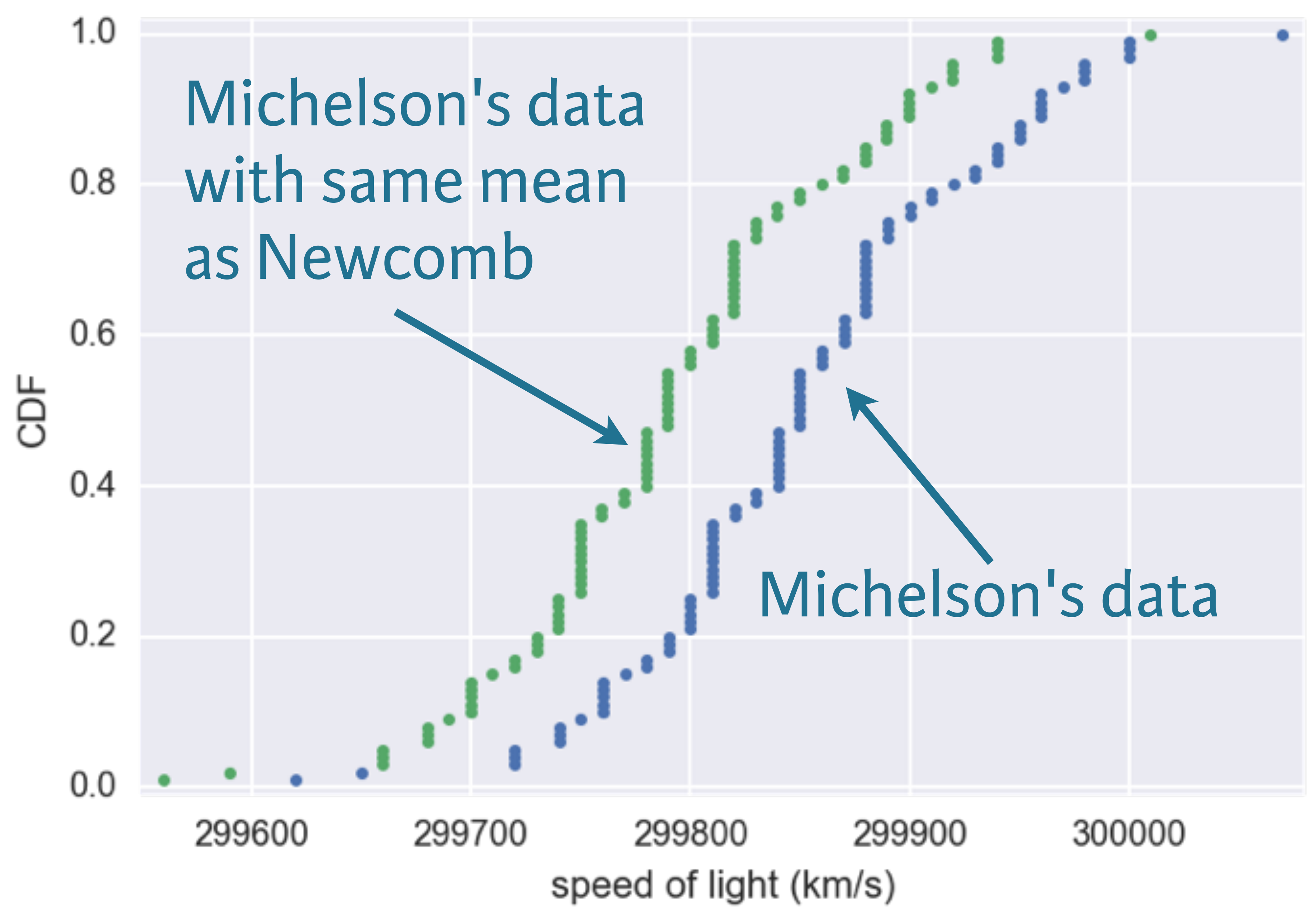
Shifting the Michelson data

```
In [1]: newcomb_value = 299860  # km/s

In [2]: michelson_shifted = michelson_speed_of_light \
   ...:         - np.mean(michelson_speed_of_light) + newcomb_value
```

shift Michelson's data such that its mean now matches Newcomb's value

Shifting the Michelson data

```
In [1]: newcomb_value = 299860  # km/s

In [2]: michelson_shifted = michelson_speed_of_light \
   ...:         - np.mean(michelson_speed_of_light) + newcomb_value
```



use bootstrapping on this shifted data to simulate data acquisition under the null hypo.

Calculating the test statistic

```
In [1]: def diff_from_newcomb(data, newcomb_value=299860):
   ...:     return np.mean(data) - newcomb_value
   ...:

In [2]: diff_obs = diff_from_newcomb(michelson_speed_of_light)

In [3]: diff_obs
Out[3]: -7.5999999999767169
```

test statistic is the mean of bootstrap sample minus Newcomb's value

Computing the p-value

```
In [1]: bs_replicates = draw_bs_reps(michelson_shifted,
   ...:                              diff_from_newcomb, 10000)

In [2]: p_value = np.sum(bs_replicates <= diff_observed) / 10000

In [3]: p_value
Out[3]: 0.16039999999999999
```

bootstrap replicates

use less than because the mean from Michelson's exp was less than Newcomb's value

p value suggests that it is quite possible the Newcomb and Michelson did not really have fundamental differences in their measurements.
This is an example of a one-sample test

One sample test

- Compare one set of data to a single number

Two sample test

- Compare two sets of data

You performed a one-sample bootstrap hypothesis test, which is impossible to do with permutation. Testing the hypothesis that two samples have the same distribution may be done with a bootstrap test, but a permutation test is preferred because it is more accurate (exact, in fact). But therein lies the limit of a permutation test; it is not very versatile. We now want to test the hypothesis that Frog A and Frog B have the same mean impact force, but not necessarily the same distribution. This, too, is impossible with a permutation test.

Let’s practice!

To do the two-sample bootstrap test, we shift both arrays to have the same mean, since we are simulating the hypothesis that their means are, in fact, equal. We then draw bootstrap samples out of the shifted arrays and compute the difference in means. This constitutes a bootstrap replicate, and we generate many of them. The p-value is the fraction of replicates with a difference in means greater than or equal to what was observed.