DATA 8

Week 10 – Bootstrapping, More Confidence Intervals

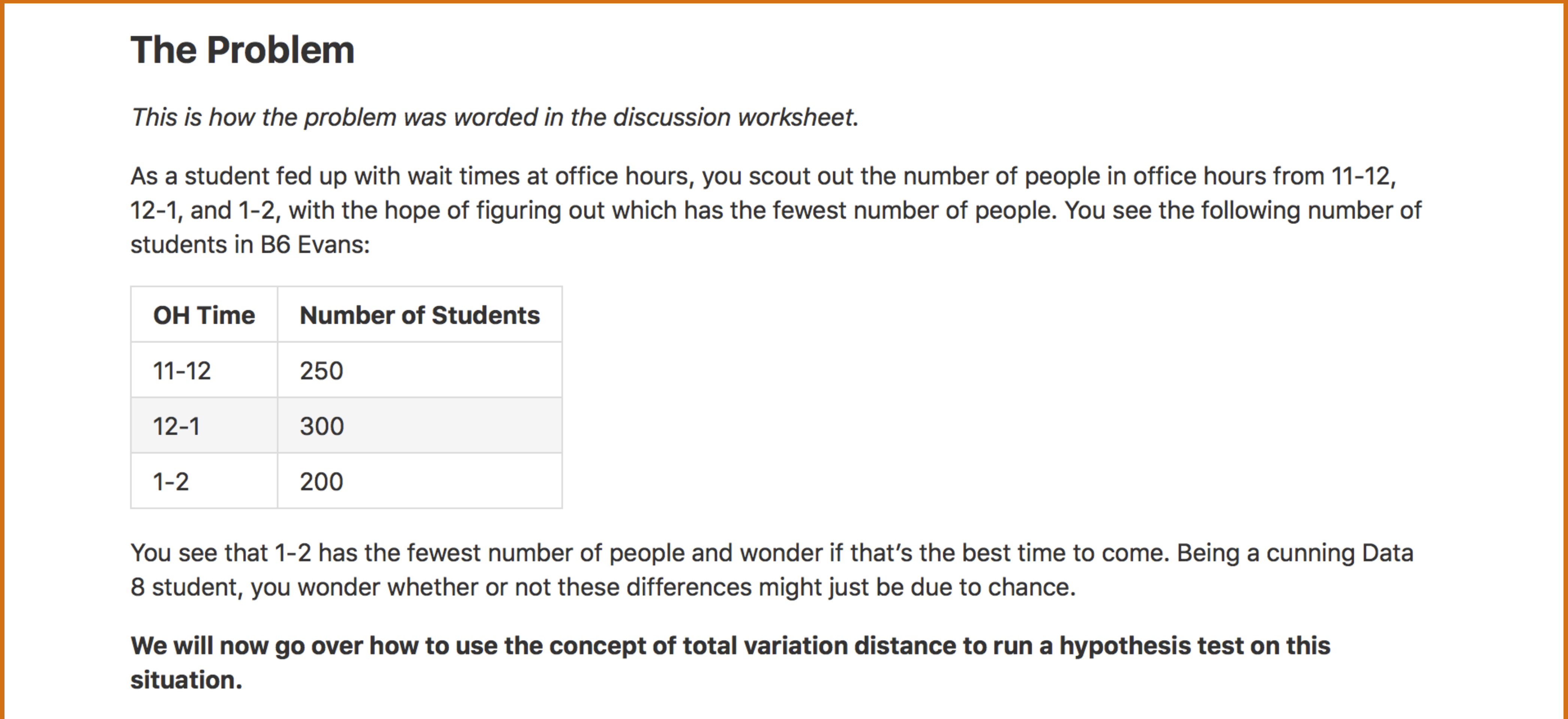
Slides by Suraj Rampure

Fall 2017

Administrative Notes

Project 2 is due tonight. Please don't work on it during lab until you're finished the actual lab assignment; you're only doing yourself a disservice.

My explanation of TVD last week was meh. I wrote http://surajrampure.com/resources/data8/tvd-walkthrough.html to help. You should definitely check this out and ask me if you have any questions.



The Problem

This is how the problem was worded in the discussion worksheet.

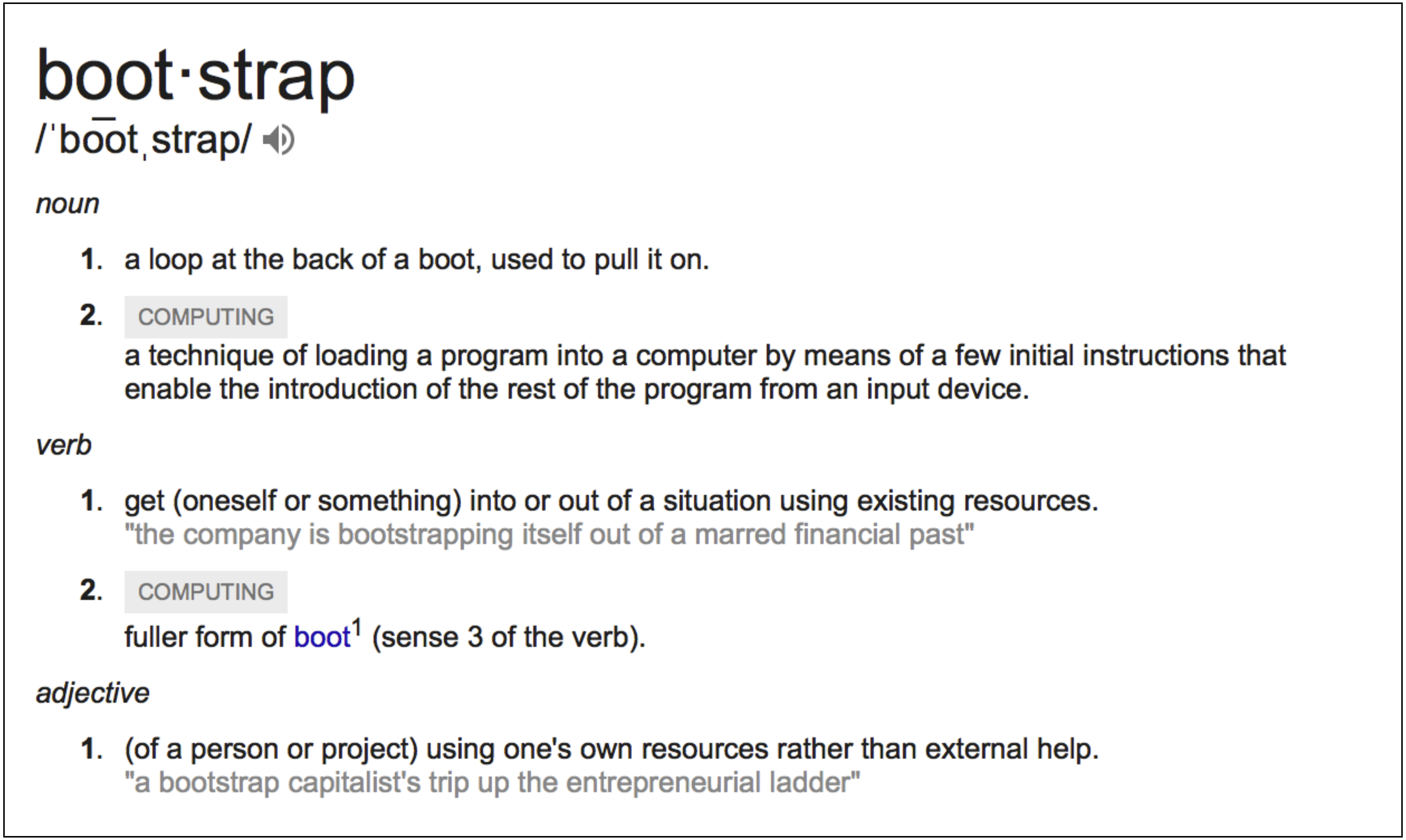
As a student fed up with wait times at office hours, you scout out the number of people in office hours from 11-12, 12-1, and 1-2, with the hope of figuring out which has the fewest number of people. You see the following number of students in B6 Evans:

OH Time	Number of Students
11-12	250
12-1	300
1-2	200

You see that 1-2 has the fewest number of people and wonder if that's the best time to come. Being a cunning Data 8 student, you wonder whether or not these differences might just be due to chance.

We will now go over how to use the concept of total variation distance to run a hypothesis test on this situation.

The Bootstrap



boot·strap

/ˈbo͞otˌstrap/

noun

1. a loop at the back of a boot, used to pull it on.
2. COMPUTING
   a technique of loading a program into a computer by means of a few initial instructions that enable the introduction of the rest of the program from an input device.

verb

1. get (oneself or something) into or out of a situation using existing resources.
   "the company is bootstrapping itself out of a marred financial past"
2. COMPUTING
   fuller form of boot^1 (sense 3 of the verb).

adjective

1. (of a person or project) using one's own resources rather than external help.
   "a bootstrap capitalist's trip up the entrepreneurial ladder"

The Bootstrap

What?

Bootstrapping is a sampling technique that generates new samples by **resampling from the original sample.**

Why?

If you're looking to estimate a population parameter, but are **unable to sample** more.

When?

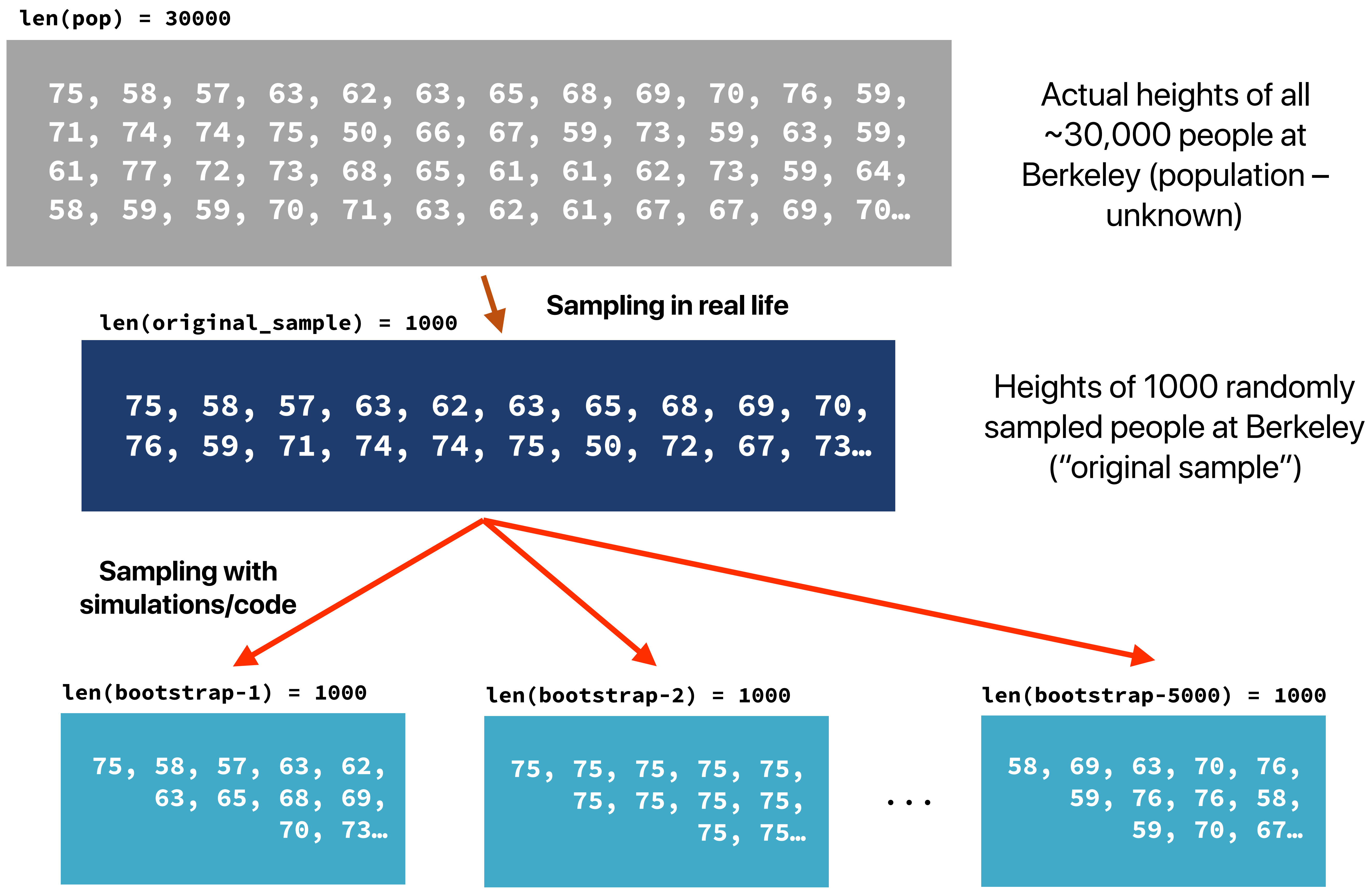
To bootstrap, your sample must be drawn **randomly** and **large** enough that it represents the population it comes from.

How?

Treat the **original sample** as if it were the **population**. Draw from the "population" at random **with replacement**, the same number of times as the original sample size.

Why do we have to **sample with replacement** when bootstrapping?

If we were to sample without replacement, we’d get back the original sample each time. By sampling with replacement, we’ll get different combinations of elements from the original sample.



`len(pop) = 30000`

```
75, 58, 57, 63, 62, 63, 65, 68, 69, 70, 76, 59,
71, 74, 74, 75, 50, 66, 67, 59, 73, 59, 63, 59,
61, 77, 72, 73, 68, 65, 61, 61, 62, 73, 59, 64,
58, 59, 59, 70, 71, 63, 62, 61, 67, 67, 69, 70…
```

Actual heights of all ~30,000 people at Berkeley (population – unknown)

Sampling in real life

`len(original_sample) = 1000`

```
75, 58, 57, 63, 62, 63, 65, 68, 69, 70,
76, 59, 71, 74, 74, 75, 50, 72, 67, 73…
```

Heights of 1000 randomly sampled people at Berkeley ("original sample")

Sampling with simulations/code

`len(bootstrap-1) = 1000`

```
75, 58, 57, 63, 62,
    63, 65, 68, 69,
        70, 73…
```

`len(bootstrap-2) = 1000`

```
75, 75, 75, 75, 75,
    75, 75, 75, 75,
        75, 75…
```

. . .

`len(bootstrap-5000) = 1000`

```
58, 69, 63, 70, 76,
    59, 76, 76, 58,
        59, 70, 67…
```

This probably wouldn't happen, just as you probably wouldn't draw the Queen of Hearts 10 times in a row when drawing from a shuffled deck of cards. The point of including this is to show a possibility.

Simulating the Bootstrap

```
sample = […] # The data that you’ve collected, say, heights
num_trials = 10000
means = make_array()

for i in np.arange(num_trials):
 # Note: np.random.choice samples with replacement, as we want
 resampled = np.random.choice(sample, 1000)
 means = np.append(means, np.mean(resampled))
```

Note that this looks exactly like a regular hypothesis test simulation. That’s because that’s exactly what this is – the only difference is, we’re sampling from a sample!

Bootstrapping – The End Goal

Remember, at the end of all of this, we're trying to **estimate a population parameter**. We don't know the actual parameter, and we have no way of finding it out – if we did, all of this would be useless.

Suppose we're trying to estimate the mean height of everyone at Berkeley. After bootstrapping, we'll have a large number (say, 10,000) of sample statistics. How can we analyze these to estimate where the parameter lies?

Confidence intervals.

Bootstrapping – Constructing a CI

We construct a **k% confidence interval** by looking at the **middle k% of values** from our re-sampled statistics. Continuing with our example, we'd look at the middle k% of our re-sampled heights to get an estimate of where our parameter lies.

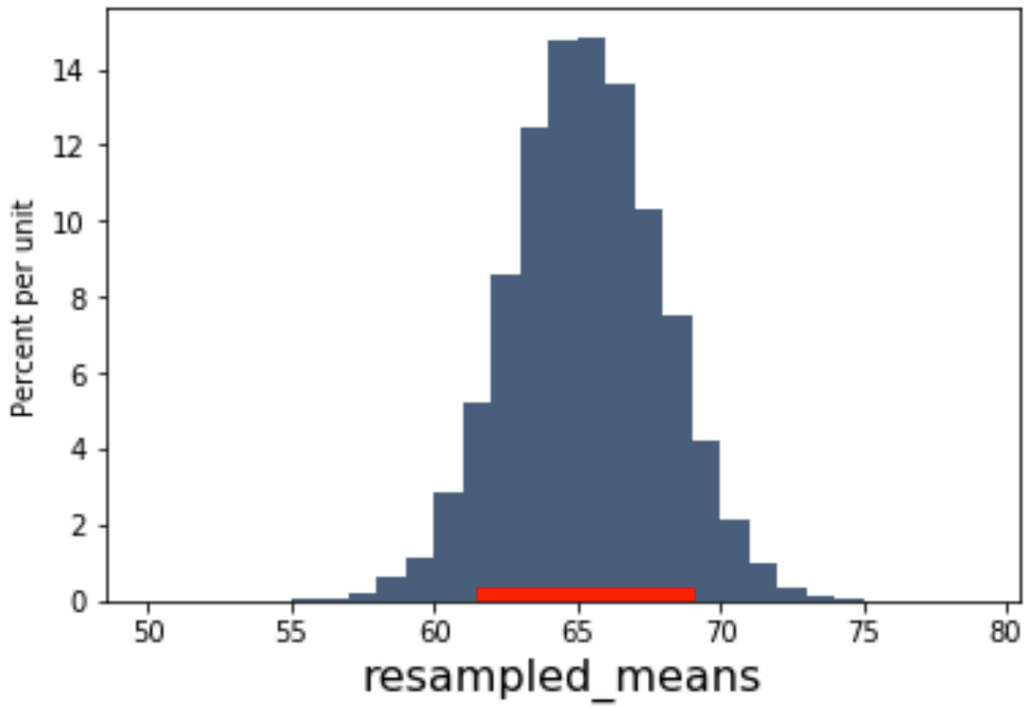
Typically we use a confidence level of 90% or 95%.

Remember: 95% confidence **doesn't mean that there is a 95% chance** that the parameter is in the interval. It means that using this sampling technique, **about 95% of the time we will have created a good interval** (one that contains the parameter).

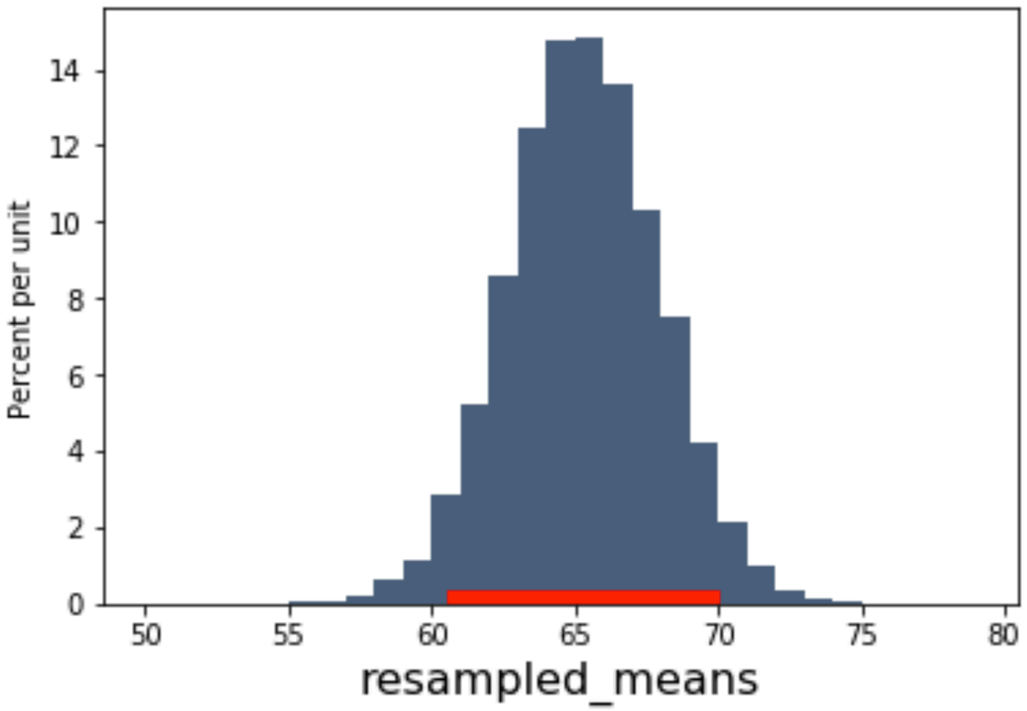
This is a subtle difference, but an important one. We never know where the true parameter lies; this is an important concept to keep in mind.

Confidence vs. Interval Width

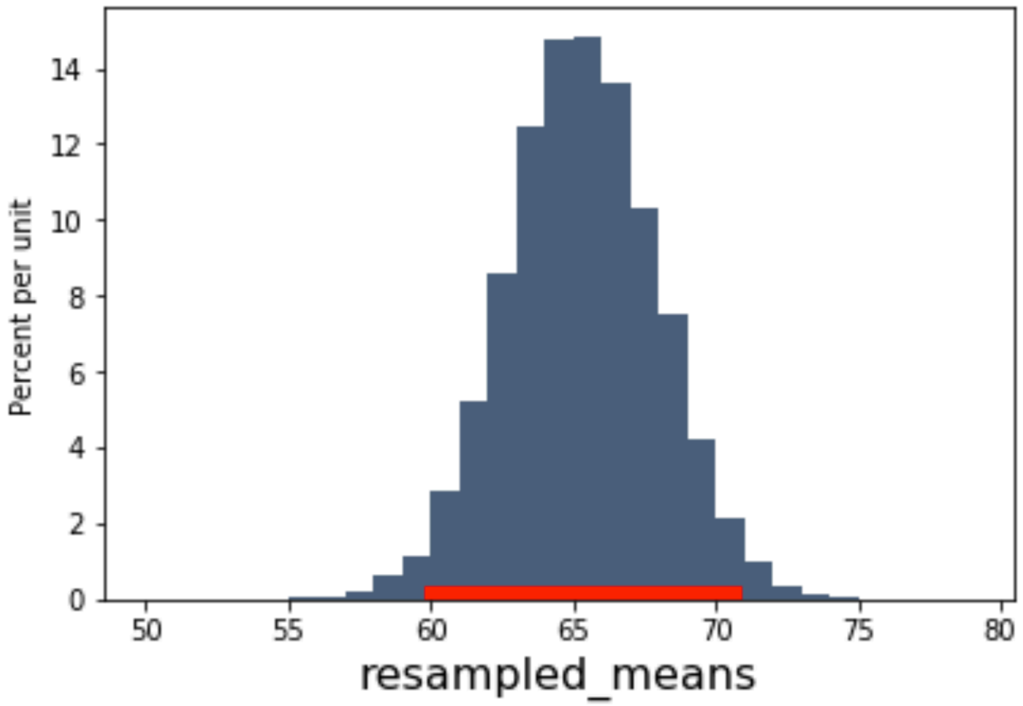
80% confidence



90% confidence

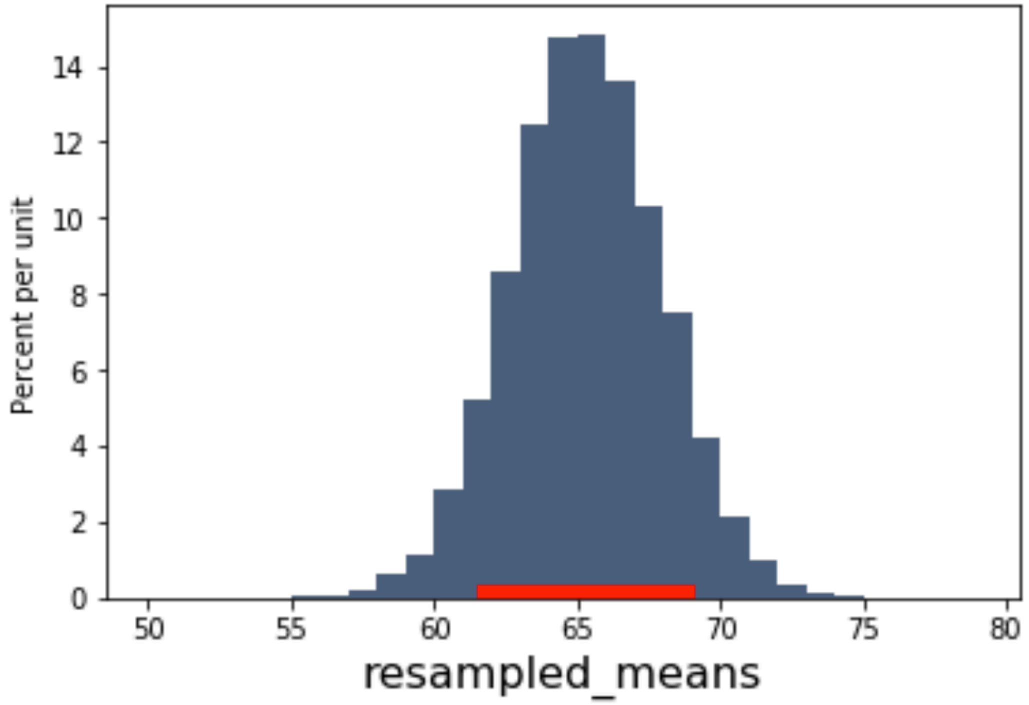


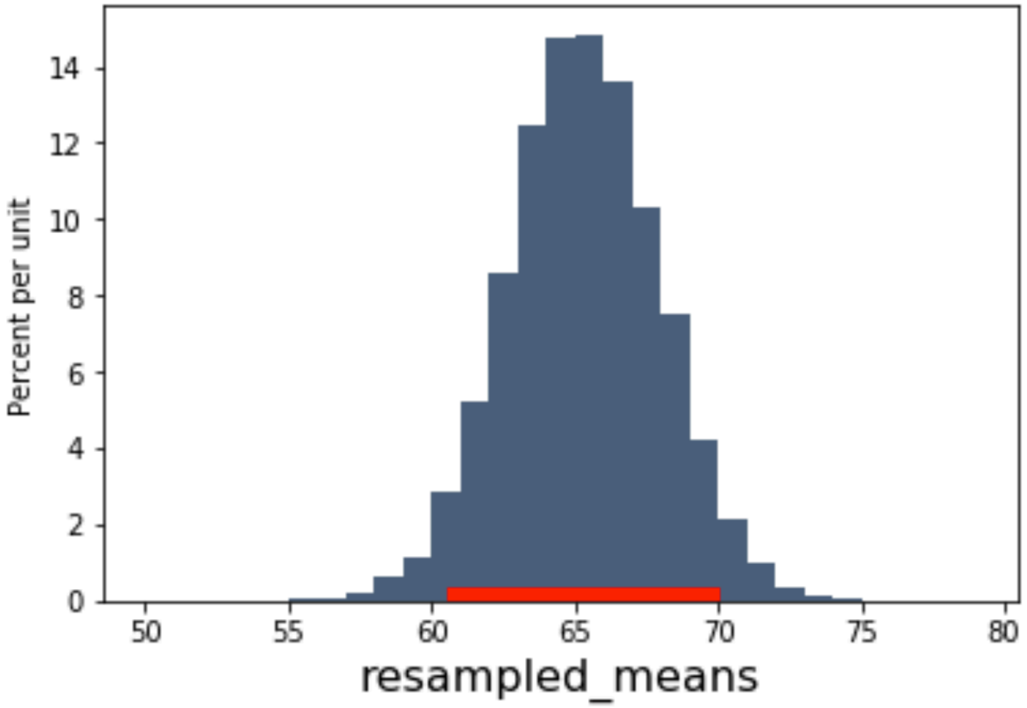
95% confidence

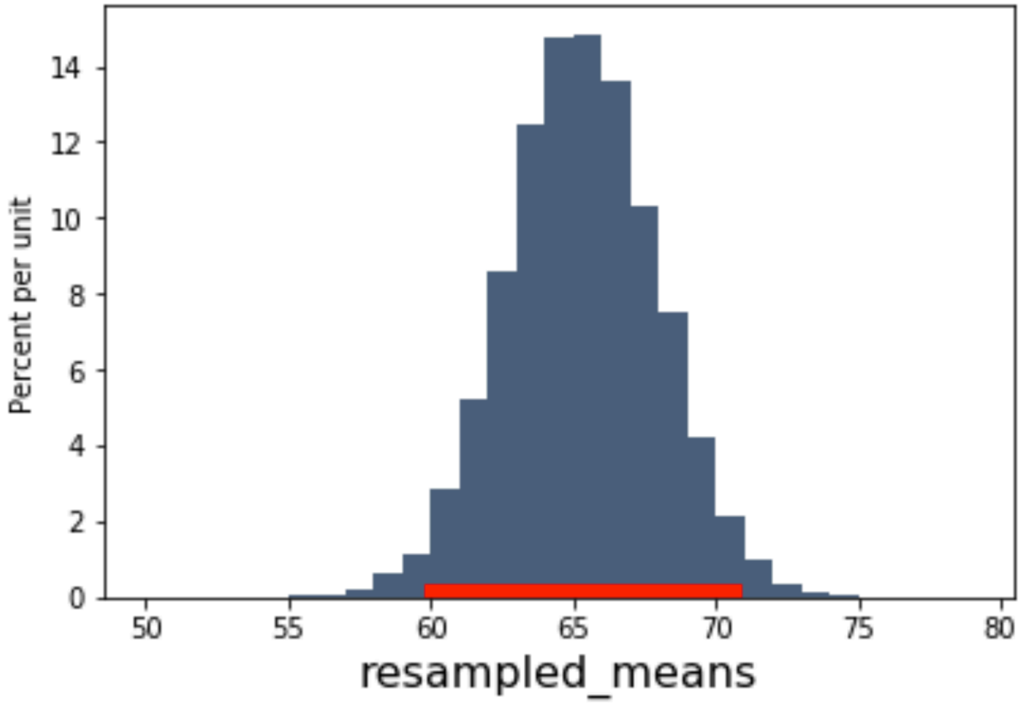


The greater the confidence level we wish to have, the greater the width of our confidence interval will be. This makes sense, because to be more confident that an interval will contain our parameter, we need to look at more values.

Confidence vs. Interval Width


80% confidence


90% confidence


95% confidence

```
# How to find a k% confidence interval
lower_bound = percentile((100-k)/2, means)
upper_bound = percentile(100-(100-k)/2, means)
```