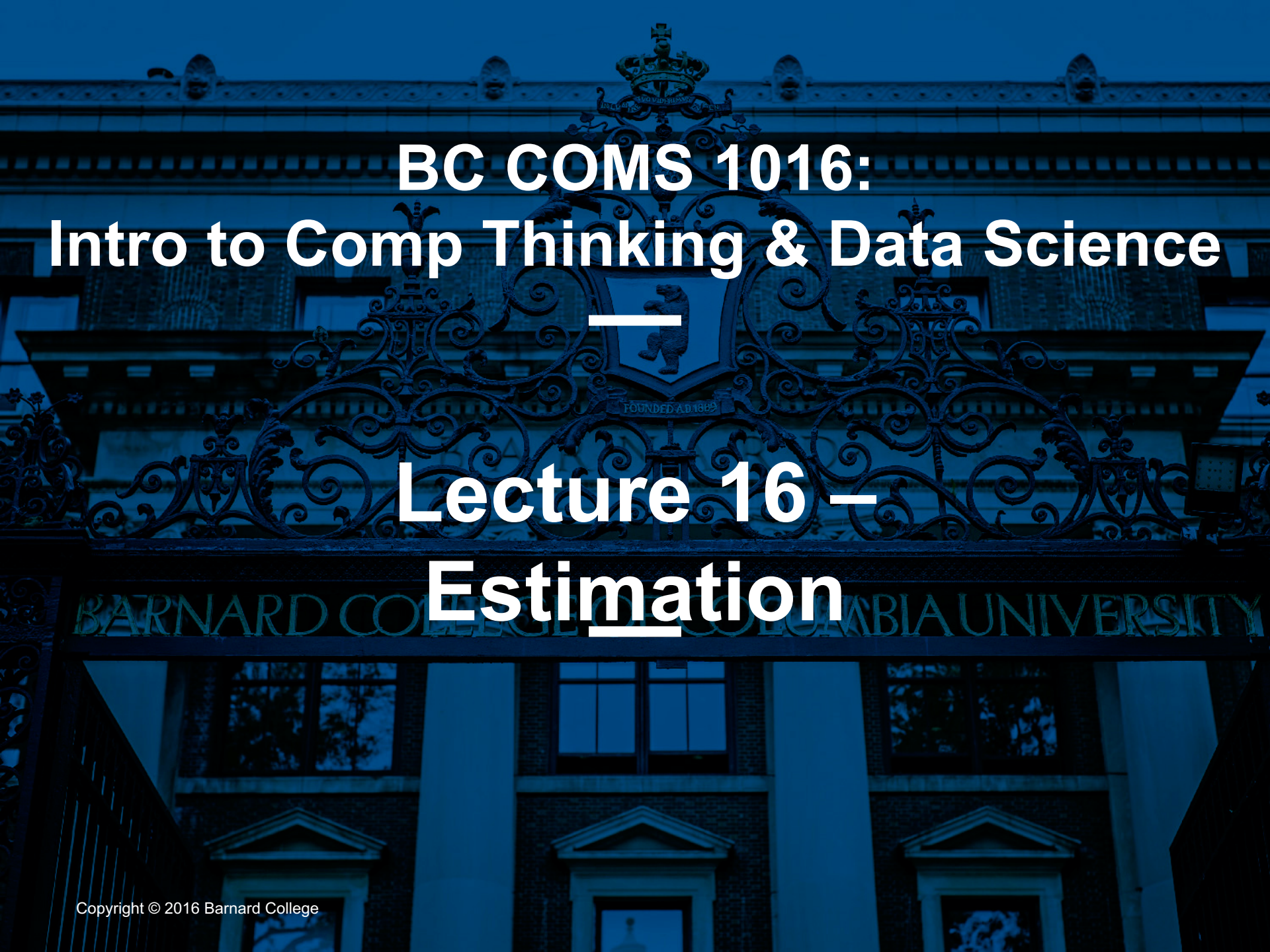


BC COMS 1016: Intro to Comp Thinking & Data Science

—

Lecture 16 – Estimation

Announcements - update

- HW06 - Testing Hypotheses
  - Due last night (Monday 11/22)
  - Allowed up to 3
  - If you want to use 4, email me
- Checkpoint/Project 2 (midterm):
  - Released due Monday 11/30
- Lab05 – moved to this week
  - Due Wednesday 11/23
- Homework 05
  - Question 3.4
- Feedback from is open for the rest of the semester
  - I'll check it daily

Thanksgiving Week Office Hours

- Fran and Lucie office hours cancelled
- Priya:
  - Wednesday office hours happening
  - Friday office hours cancelled
- Adam:
  - Wednesday: 9:00am – 12:00pm
  - Friday: 8:30am – 10:30am

Percentiles

Computing Percentiles

The Xth percentile is first value on the sorted list that is at least as large as X% of the elements

Example:

s = [1, 7, 3, 9, 5]

s_sorted = [1, 3, 5, 7, 9]

percentile(80, s) = ?

The 80th percentile is ordered element 4: (80/100) * 5

For a percentile that does not exactly correspond to an element, take the next greater element instead

The percentile Function

- The *p*th percentile is the **smallest value** in a set that is **at least as large as *p*%** of the elements in the set
- Function in the datascience module:

  percentile(p, values)

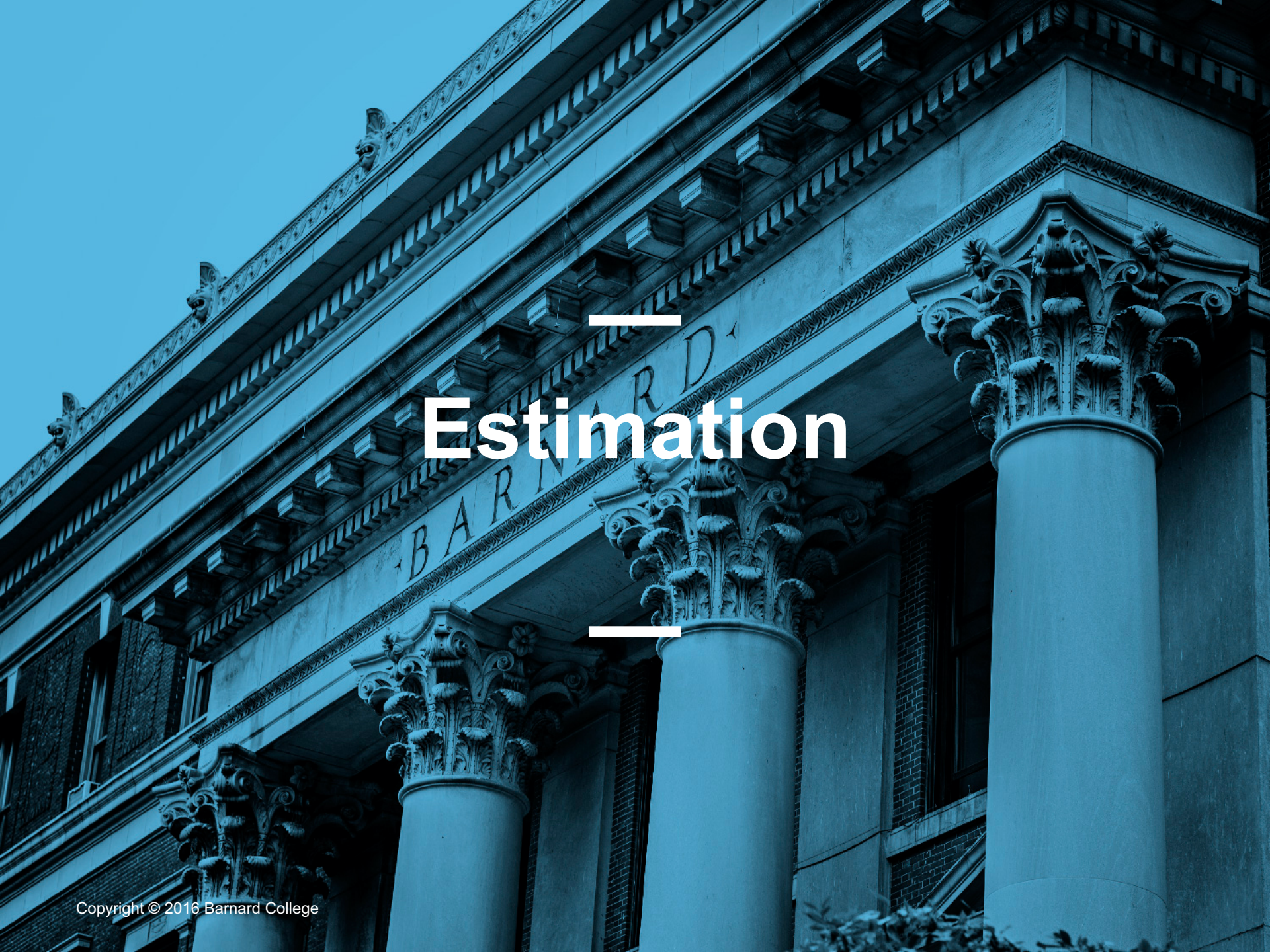
- p is between 0 and 100

- Returns the *p*th percentile of the array

Discussion Question?

Which are True, when s = [1, 7, 3, 9, 5]?

1. percentile(10, s) == 0
2. percentile(39, s) == percentile(40, s)
3. percentile(40, s) == percentile(41, s)
4. percentile(50, s) == 5



Estimation

Inference: Estimation

- How do we calculate the value of an unknown parameter?
- If you have a census (that is, the whole population):
  - Just calculate the parameter and you're done
- If you don't have a census:
  - Take a random sample from the population
  - Use a statistic as an **estimate** of the parameter

Estimation Variability

Variability of the Estimate

- One sample → One estimate
- But the random sample could have come out differently
- And so the estimate could have been different
- Big question:
  - How different would it be if we estimated again?

Quantifying Uncertainty

- The estimate is usually not exactly right.
- Variability of the estimate tells us something about how accurate the estimate is:

Estimate = Parameter + Error

- How accurate is the estimate, usually?
- How big is a typical error?
- When we have a census, we can do this by simulation

Where to Get Another Sample?

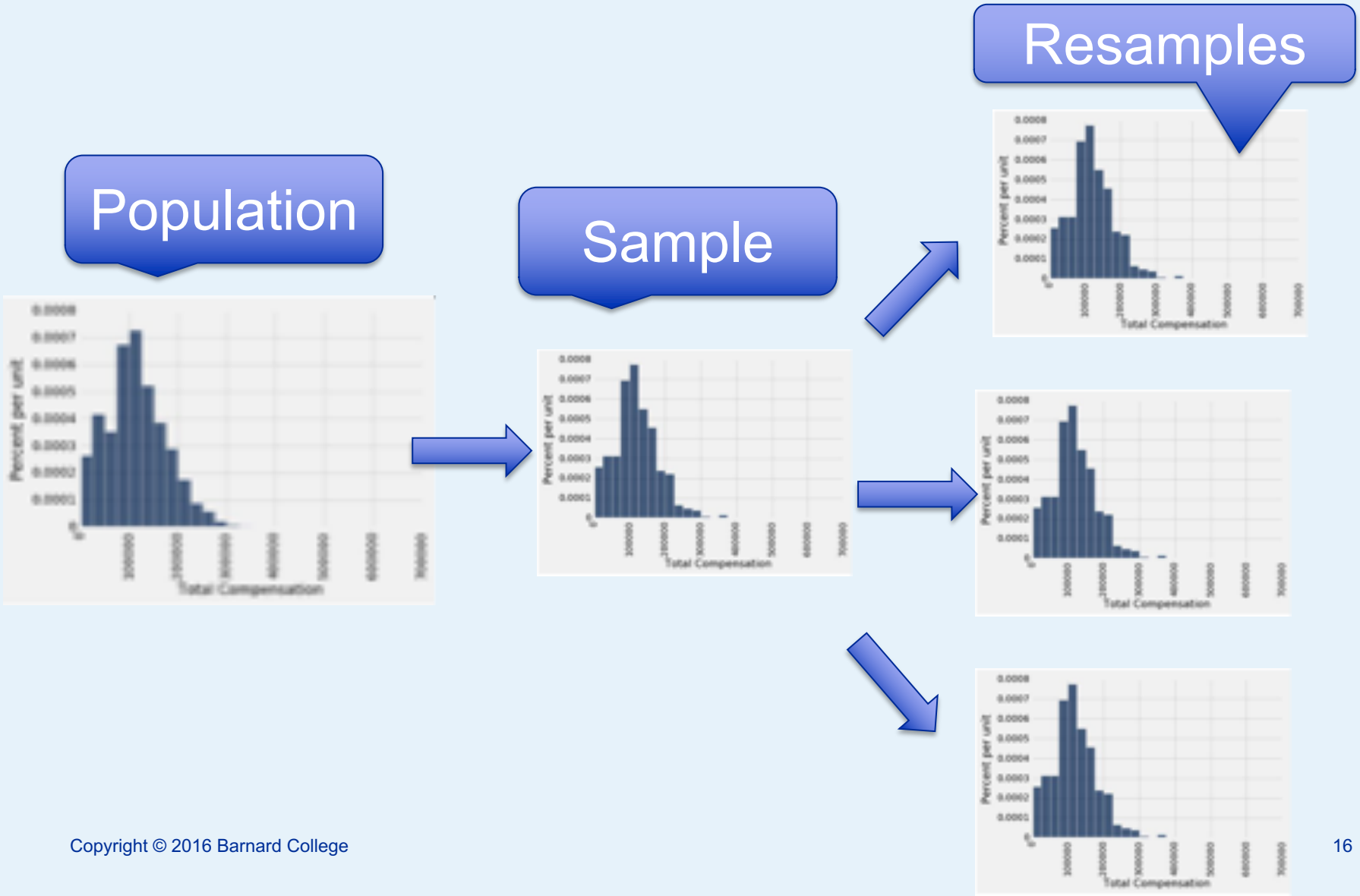
- We want to understand errors of our estimate
- Given the **population**, we could simulate
  - ...but we only have the **sample**!
- To get many values of the estimate, we needed many random samples
- Can't go back and sample again from the population:
  - No time, no money
- Stuck?

The Bootstrap

The Bootstrap

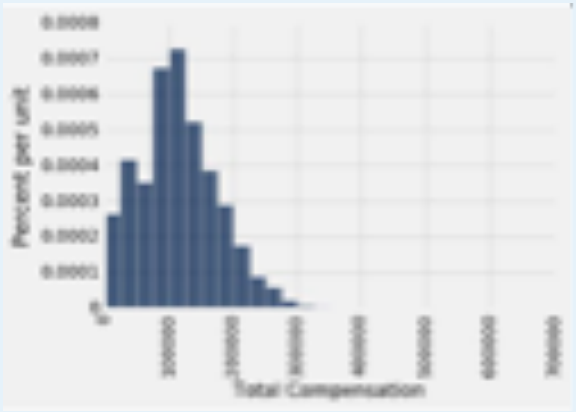
- A technique for simulating repeated random sampling
- All that we have is the original sample
  - ... which is large and random
  - Therefore, it probably resembles the population
- So we sample at random from the original sample!

How the Bootstrap works



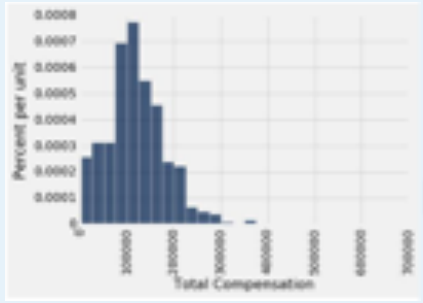
Why the Bootstrap works

Population



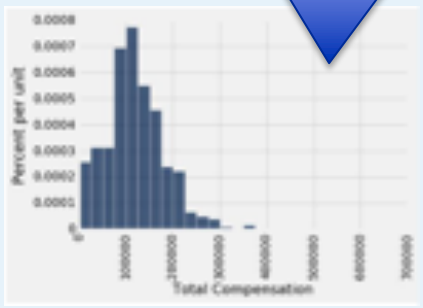
What we wish we could get

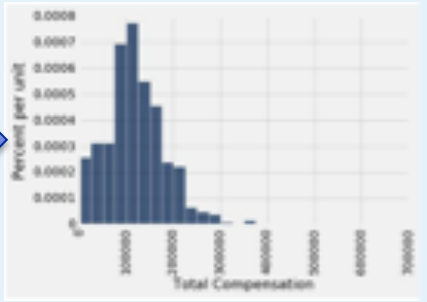
Sample

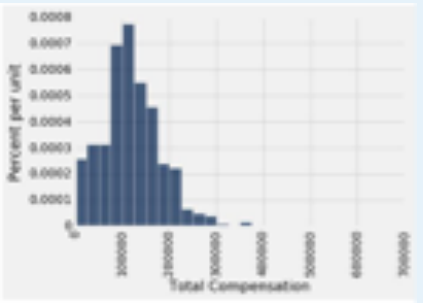


What we actually can get

Resamples







Real World vs Bootstrap World

Real World

- True probability distribution (population)
  - Random sample 1
    - Estimate 1
  - Random sample 2
    - Estimate 2
  - …
  - Random sample 1000
    - Estimate 1000

Bootstrap World

- Empirical distribution of original sample ("population")
  - Bootstrap sample 1
    - Estimate 1
  - Bootstrap sample 2
    - Estimate 2
  - …
  - Bootstrap sample 1000
    - Estimate 1000

Hope: these two scenarios are analogous

The Bootstrap Principle

- The bootstrap principle:
  - **Bootstrap-world** sampling ≈ **Real-world** sampling
- Not always true!
  - ... but reasonable if sample is large enough
- We hope that:
  a) Variability of bootstrap estimate
  b) Distribution of bootstrap errors

  ...are similar to what they are in the real world

Key to Resampling

- From the original sample,
  - draw at random
  - with replacement
  - as many values as the original sample contained

- The size of the new sample has to be the same as the original one, so that the two estimates are comparable

Confidence Intervals

95% Confidence Interval

- Interval of **estimates of a parameter**
- Based on random sampling
- 95% is called the confidence level
  - Could be any percent between 0 and 100
  - Higher level means wider intervals

- The **confidence is in the process** that gives the interval:
  - It generates a “good” interval about 95% of the time

Can You Use a CI Like This?

By our calculation, an approximate 95% confidence interval for the average age of the mothers in the population is (26.9, 27.6) years.

True or False:

- About 95% of the mothers in the population were between 26.9 years and 27.6 years old.

Answer:

- **False.** We're estimating that their **average age** is in this interval.

Is This What a CI Means?

An approximate 95% confidence interval for the average age of the mothers in the population is (26.9, 27.6) years.

True or False:

There is a 0.95 probability that the average age of mothers in the population is in the range 26.9 to 27.6 years.

Answer:

False. The average age of the mothers in the population is unknown but it's a constant. It's not random. No chances involved

When *NOT* to use the Bootstrap

- if you're trying to estimate very high or very low percentiles, or min and max
- If you're trying to estimate any parameter that's greatly affected by rare elements of the population
- If the probability distribution of your statistic is not roughly bell shaped (the shape of the empirical distribution will be a clue)
- If the original sample is very small

Using a CI for Testing

- Null hypothesis: **Population average = *x***
- Alternative hypothesis: **Population average ≠ *x***
- Cutoff for P-value: *p*%
- Method:
  - Construct a (100-*p*)% confidence interval for the population average
  - If *x* is not in the interval, reject the null
  - If *x* is in the interval, can't reject the null