Big Data and Economics

Bootstrapping, Functions, and Parallel Programming

Kyle Coombs
Bates College | ECON/DCS 368

Table of contents

Prologue

Prologue

- By the end of class you will:
  - Appreciate statistical uncertainty and the role of statistical inference
  - Understand the intuition of bootstrapping
  - Be able to bootstrap in R

Hack-a-thon update

Sign-up for the hack-a-thon by April 2nd!

- Sign ups posted in GitHub discussions
- It will run from April 9 to April 11
- You can sign up as a team or as an individual

Questions

Attribution

I pull most of this lecture from the textbook Data Science in R by James Scott

Review of statistical inference

- A key, but often overlooked part of causal inference is "inference"
- In most analysis, the data make up a small (though it can get pretty big) sample of the population
  - We implement the perfect causal identification strategy perfectly: $y_{it} = \beta_0 + \beta_1 x_{it} + \epsilon_{it}$
  - We get a point estimate of $\hat{\beta}_1$ from our sample that is unbiased and consistent
  - Are we certain that the population's true $\beta_1 = \hat{\beta}_1$?

Review of statistical inference

- A key, but often overlooked part of causal inference is "inference"
- In most analysis, the data make up a small (though it can get pretty big) sample of the population
  - We implement the perfect causal identification strategy perfectly: $y_{it} = \beta_0 + \beta_1 x_{it} + \epsilon_{it}$
  - We get a point estimate of $\hat{\beta}_1$ from our sample that is unbiased and consistent
  - Are we certain that the population's true $\beta_1 = \hat{\beta}_1$?
- No, we need to make a statistical inference
- How do we make these inferences? (Hint: Think 95%)

Confidence Intervals

- We typically make some standard assumptions about the sampling distribution of our estimates that allow us to leverage the central limit theorem (CLT)
    - CLT means that as the sample size (N) approaches infinity, the sampling distribution of a mean (ie. OLS coefficients) approaches a normal distribution
    - Typically the necessary N is not that large, but it is not always clear how large it needs to be
- What is a confidence interval?

Confidence Intervals

- We typically make some standard assumptions about the sampling distribution of our estimates that allow us to leverage the central limit theorem (CLT)
    - CLT means that as the sample size (N) approaches infinity, the sampling distribution of a mean (ie. OLS coefficients) approaches a normal distribution
    - Typically the necessary N is not that large, but it is not always clear how large it needs to be
- What is a confidence interval?
- A 95% confidence interval for $\hat{\beta}_1$ is an interval that contains the true value of β_1 in 95% of repeated samples. (Neyman 1937)
- It is **NOT** the probability that β_1 is in the interval
    - People get this wrong all the time
    - ChatGPT will get this wrong all the time -- cause it is trained on people's wrong answers
    - Bayesian "credible intervals" are the probability that β_1 is in the interval
- I know that sounds pedantic but it is important to understand the difference especially when you are interpreting your results

Example

Imagine a simple example:

1. There are 1,000,000 people in the population of interest
2. 50 percent randomly receive $10K loan forgiveness
3. You're curious how many people saved money after the forgiveness (as opposed to spending it)
4. You randomly sample 1,000 people, stratified by forgiveness receipt, and run the following regression

$$\text{Savings}_i = \beta_0 + \beta_1 \text{Forgiveness}_i + \epsilon_i$$

```
savings ← feols(savings ~ forgiveness, data = samp)
etable(list('Truth'=true_regression, 'Sample'=savings),se.below=TRUE,fitstat=c('n')) %>% kable() # Mak
```

	Truth	Sample
Dependent Var.:	savings	savings
Constant	-0.5116	-33.99
	(2.447)	(79.11)
forgiveness	996.4*	1,115.3*
	(3.462)	(111.9)
S.E. type	IID	IID

New information

You take two new samples and get:

New information

You take two new samples and get:

```
savings_new2 ← feols(savings ~ forgiveness, data = samp2)
savings_new3 ← feols(savings ~ forgiveness, data = samp3)
etable(list('Truth'=true_regression, 'Sample 1'=savings, 'Sample 2'=savings_new2, 'Sample 3'=savings_ne
```

	Truth	Sample 1	Sample 2	Sample 3
Dependent Var.:	savings	savings	savings	savings
Constant	-0.5116	-33.99	11.00	12.99
	(2.447)	(79.11)	(43.88)	(44.27)
forgiveness	996.4*	1,115.3*	-12.39	5,023.1*
	(3.462)	(111.9)	(62.06)	(62.61)
S.E. type	IID	IID	IID	IID
Observations	1,000,000	1,000	1,000	1,000

What the heck?

- You learn that 80% of the people in your sample had no loans and 20% had over 10K in loans
  - Might that change your estimate?
- You get a grant to survey for loan history of the people in your sample
- You discover the following breakdown:

sample	share_with_loan	(Intercept)	forgiveness
1	0	-33.99083	1115.28902
2	0	11.00434	-12.39388
3	0	12.99163	5023.06071

The share treated changed in each sample and the point estimate of β_1 changed accordingly

- And it looks like \$5K of the \$10K forgiveness was saved, the rest was spent

Lesson: Sampling uncertainty

- The sample you get is just one random sample from the population of interest
- If there's something special about your sample, then your estimates may be a bit odd
- I explicitly guaranteed the last two samples were not representative of the population by design
    - I also presented a world in which two binary variables were the only things that mattered beyond random noise
- In general, we assume (and take precautions) to have a representative sample of our population, but sampling uncertainty remains^[1]
- Today is about **bootstrapping**: a way to understand the variability of your estimates across samples assuming it is representative

[1] Sometimes that means we change the scope of our research question to be about a specific population for which the sample is representative, e.g. focus on low-income populations if your sample is biased towards low-income people

Bootstrapping

Bootstrapping: Motivating example

- Imagine you gain powers to view every parallel, distinct universe[1]
- With these powers, you **obviously** decide to replicate critical results in economics
  - You collect equivalent sample sizes
  - You run the same regressions to estimate the same parameters
- Do you think the results will be the same in each parallel universe?

Bootstrapping: Motivating example

- Imagine you gain powers to view every parallel, distinct universe[1]
- With these powers, you **obviously** decide to replicate critical results in economics
    - You collect equivalent sample sizes
    - You run the same regressions to estimate the same parameters
- Do you think the results will be the same in each parallel universe?
- No! They'll differ a lot or a little, depending on how variable the data generating process is

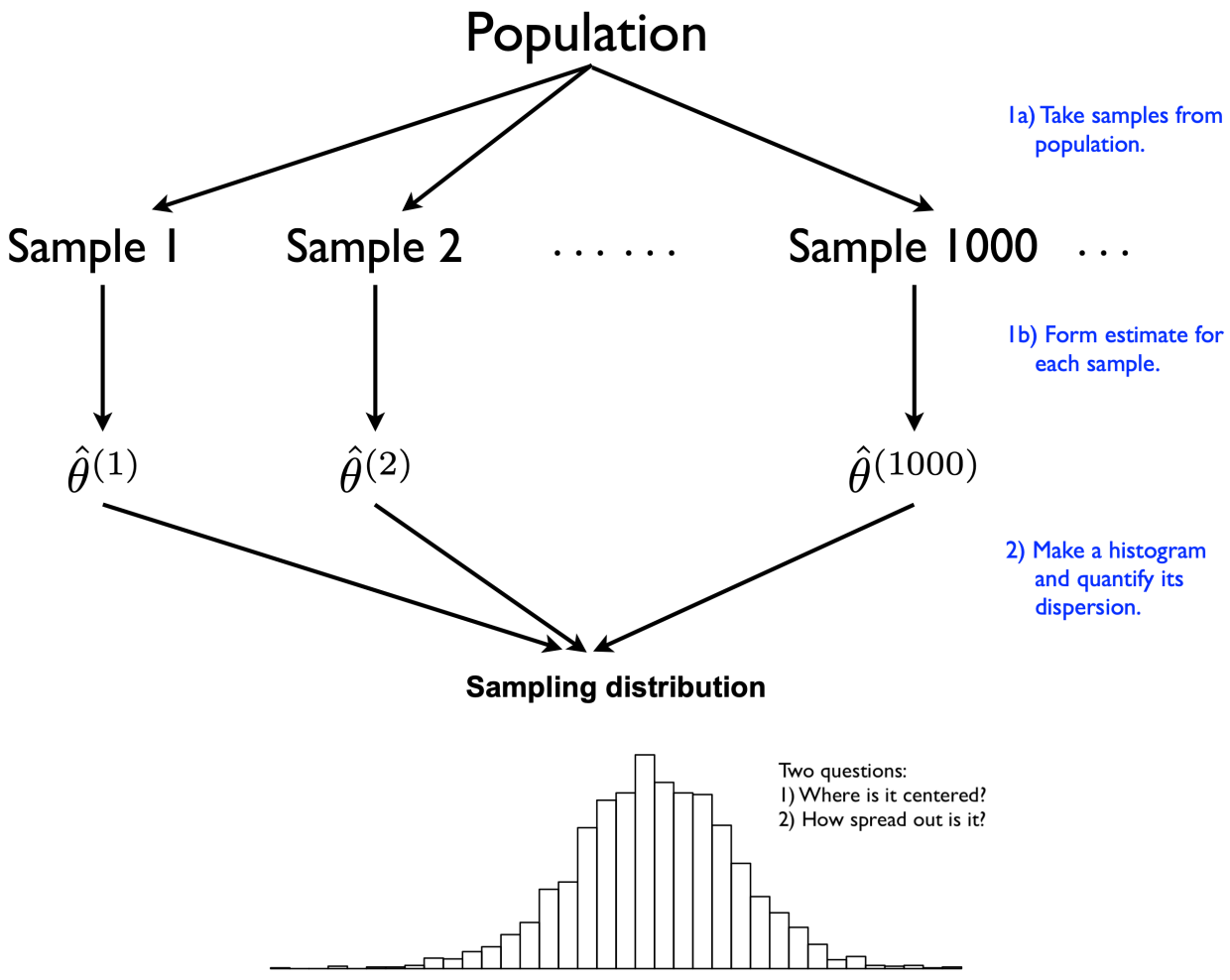
[1] Think *Everything, Everywhere, All At Once*, *Into The Spiderverse*, etc.

Return to earth

- We don't have powers to view parallel universes
- But we can view different random samples of a population of interest
- And each sample will provide a distinct estimate of the true parameters of interest
- We have two ways to use these samples to get close to our parallel universe powers:

1. **Mathematical approximations**: Make simple assumptions that randomness obeys mathematical regularities for large samples
    - e.g. *Central Limit Theorem* allows us to use the normal distribution to approximate the sampling distribution of the mean
2. **Resampling**: Use the same sample to estimate the variability of our estimates
    - e.g. *bootstrapping* which we will cover today

Visualizing samples



What is bootstrapping?

- Bootstrapping is named for "pulling yourself up by your bootstraps," a joke[2] because the method seems preposterous and impossible
- Bootstrapping has two repeated steps:

1. Draw a random sample **with replacement** of size *N* from your sample.
2. Perform the same analysis on the new sample.

- Repeat steps 1 and 2 a bunch of times saving each, the 2.5th and 97.th percentiles show the 95% confidence interval

Then plot the distribution of the estimates from each sample

[2] Not a great one.

What is a bunch of times?

How many bootstraps is enough?

What is a bunch of times?

How many bootstraps is enough?

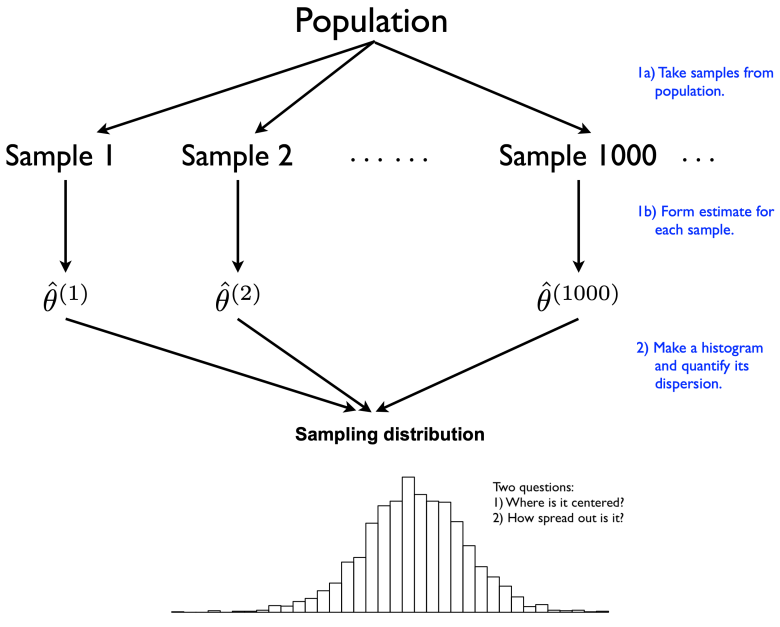
It depends. Intuively:

- The more bootstraps, the better
- But the more bootstraps, the longer it takes to run
- Many econometricians and statisticians have purported to solve for "optimal" bootstrapping, but it is still an open question

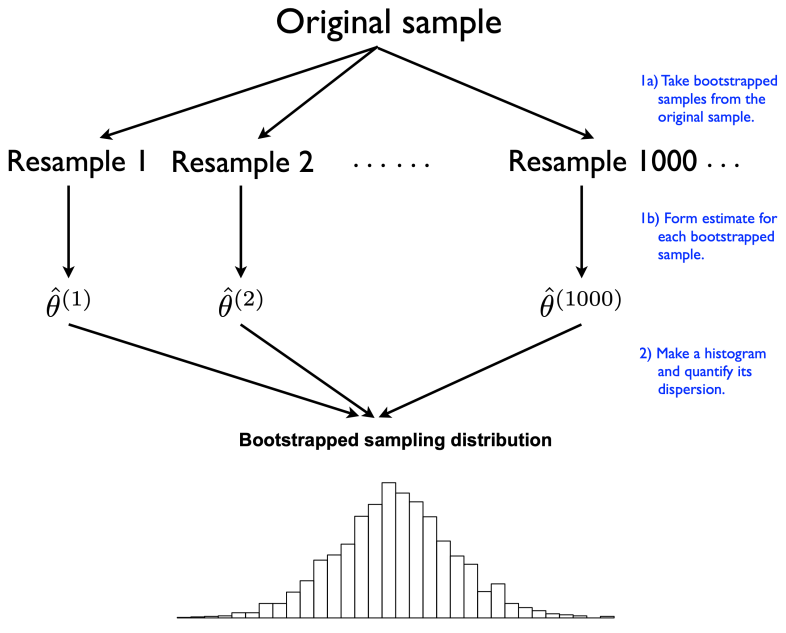
- Arguably, you should do 1000s, if not more times!
    - In this lecture, I did not do that because it would take too long to generate my slides

- See Parallel Programming for speed ups!

Visualizing Bootstrapping vs. population



Population samples



Bootstrap analog

Schematics taken from Data Science in R: A Gentle Introduction by James Scott

What does bootstrapping show?

- Bootstrapping shows how much your estimates vary across samples
- It shows the **sampling distribution** of your estimates
- The 95% confidence interval is the 2.5th and 97.5th percentile of the sampling distribution
    - Technically, there are a variety of ways to calculate the confidence interval, this is the most intuitive
    - The "Basic bootstrap" or "Reverse percentile" method is the most common: $2\hat{\theta} - \theta^*_{1-\alpha/2}, 2\hat{\theta} - \theta^*_{\alpha/2}$, but as you get started, use professionally-developed packages

What does bootstrapping show?

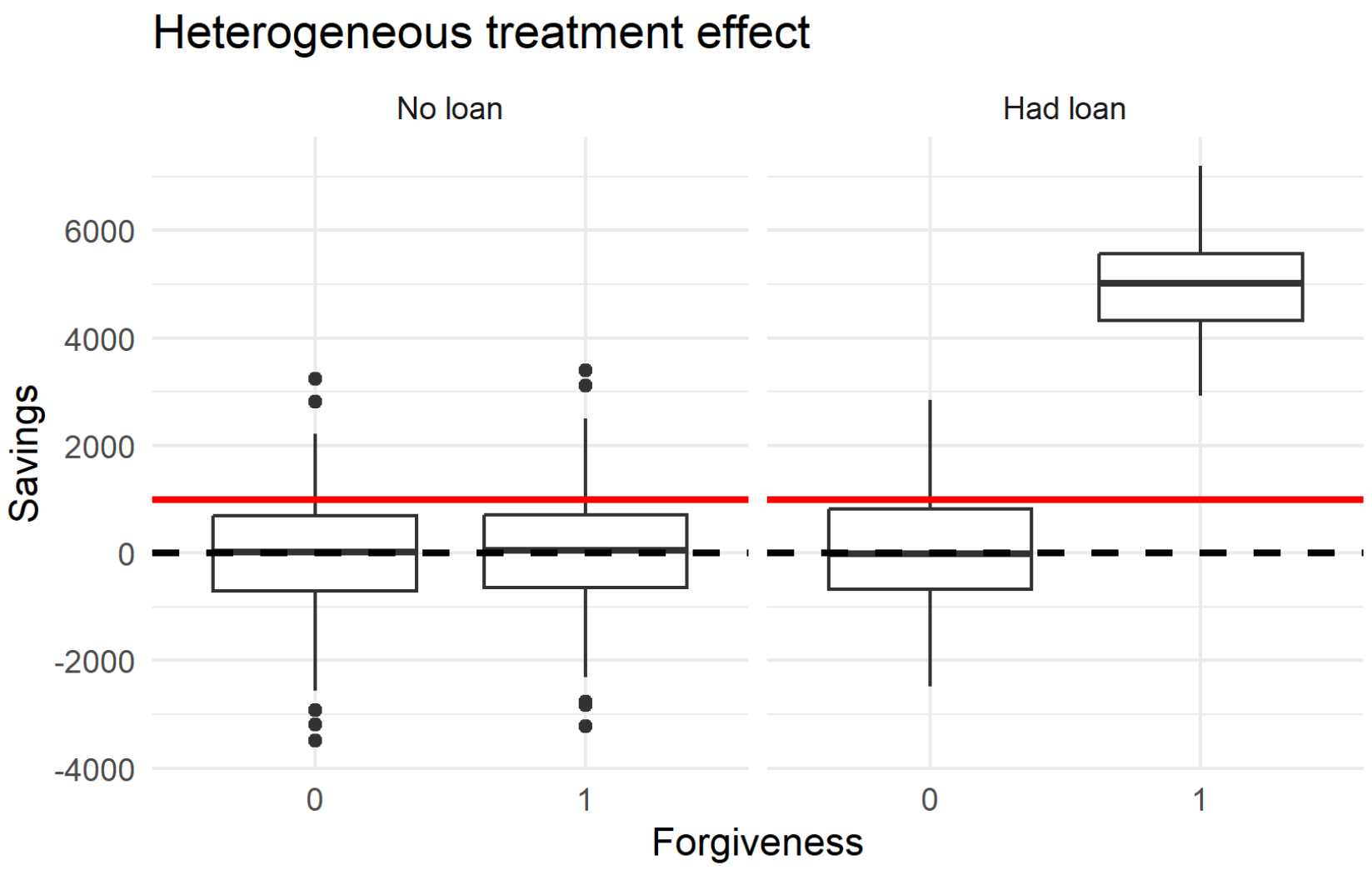
- Bootstrapping shows how much your estimates vary across samples
- It shows the **sampling distribution** of your estimates
- The 95% confidence interval is the 2.5th and 97.5th percentile of the sampling distribution
    - Technically, there are a variety of ways to calculate the confidence interval, this is the most intuitive
    - The "Basic bootstrap" or "Reverse percentile" method is the most common: $2\hat{\theta} - \theta^*_{1-\alpha/2}, 2\hat{\theta} - \theta^*_{\alpha/2}$, but as you get started, use professionally-developed packages
- **Intuition**: Bootstrapping simulates the process of collecting new samples
    - If your sample is truly representative, then any shuffled sample should be representative too!
    - Your own sample is itself a random sample generated from some other random sample

Back to loan forgiveness

- Let's go back to that loan forgiveness example take use our initial sample of 1000 people, but without knowledge of who has loans
- We'll use bootstrapping to estimate the variability of our estimates

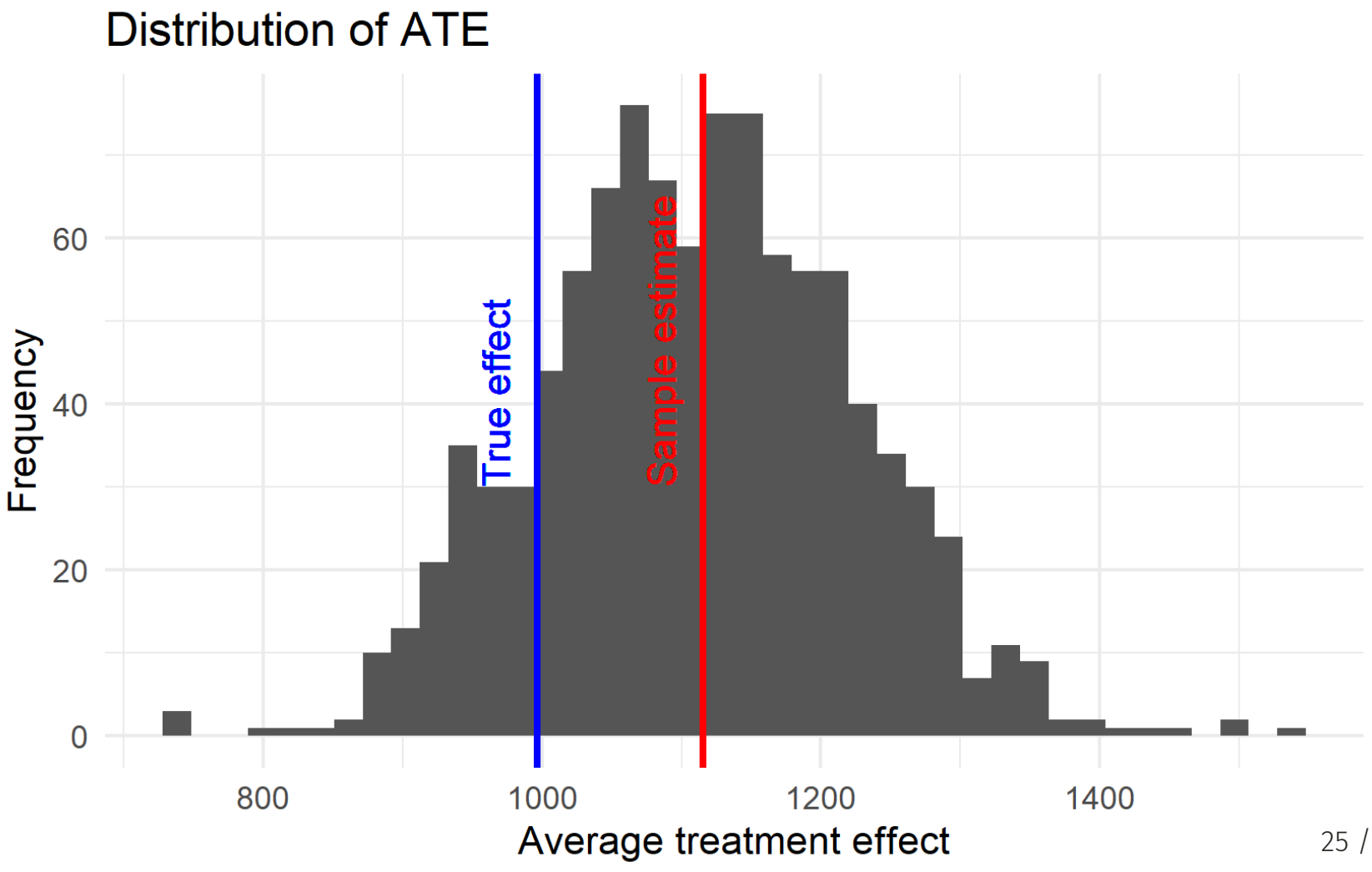
On average the treatment effect is 1000, but that varies a lot as the subsample shifts

Visualizing simple example

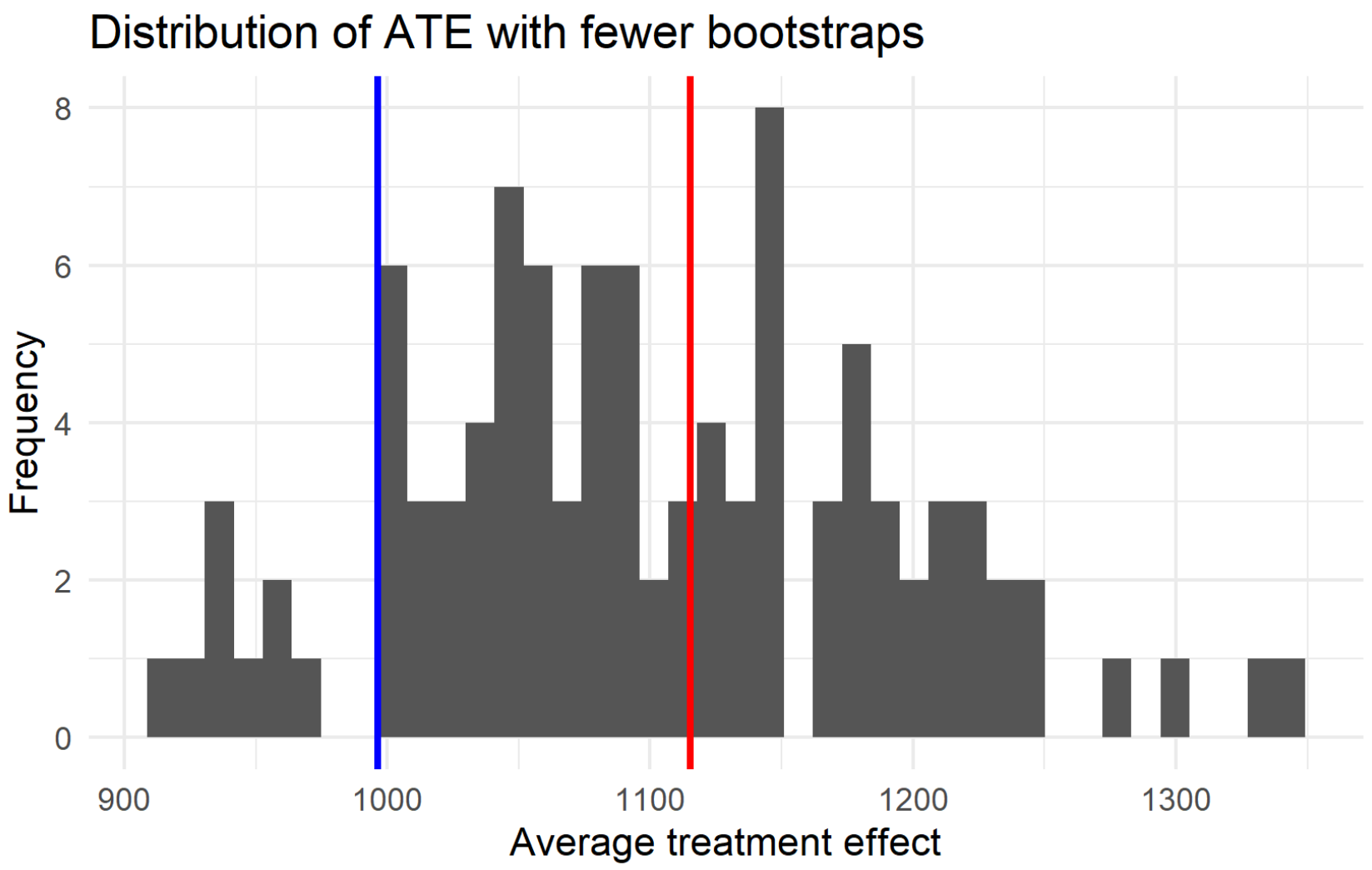


Bootstrap to get random samples

- Let's take a bunch of random samples and see how the average treatment effect varies

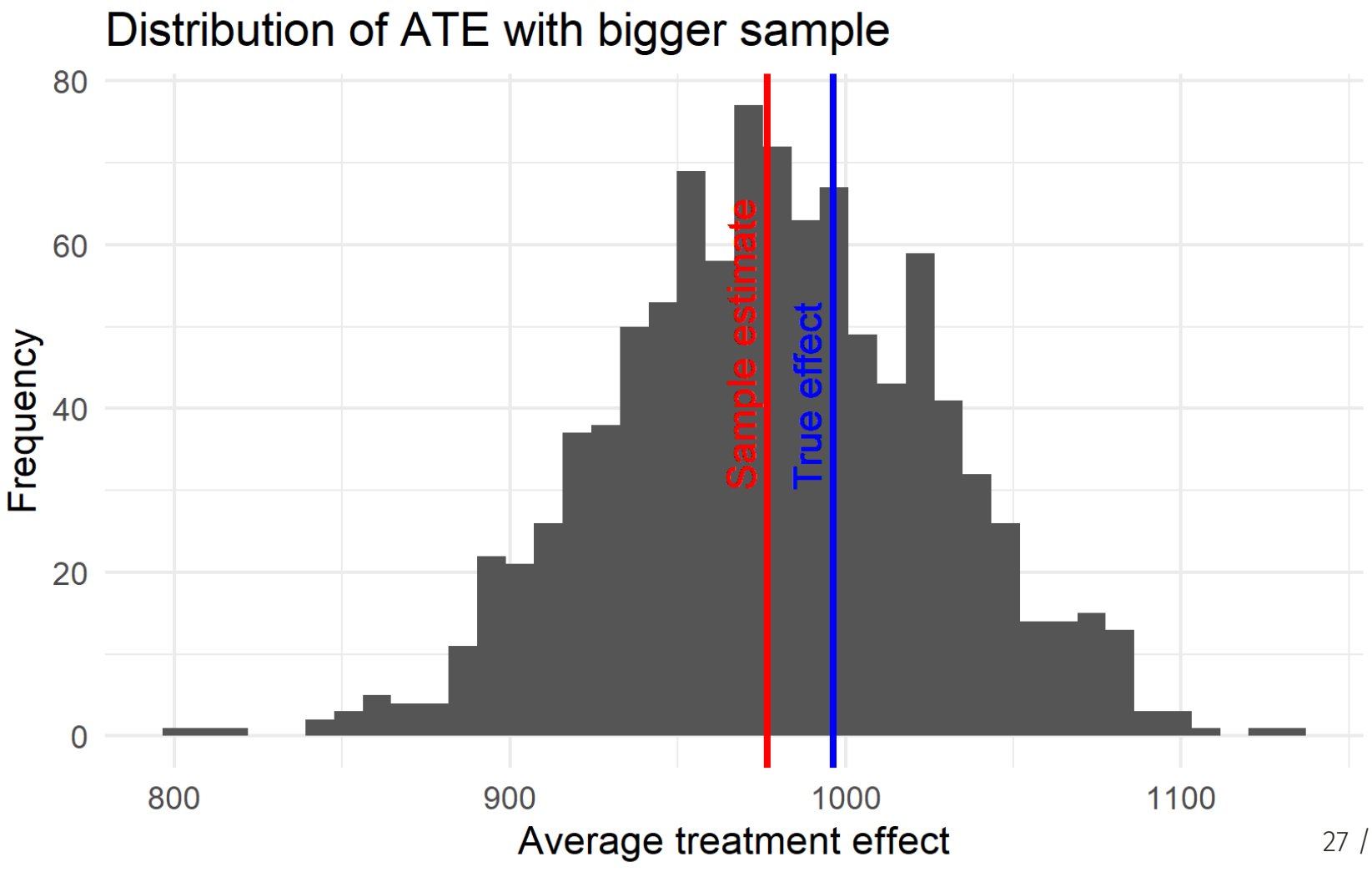


Fewer bootstraps



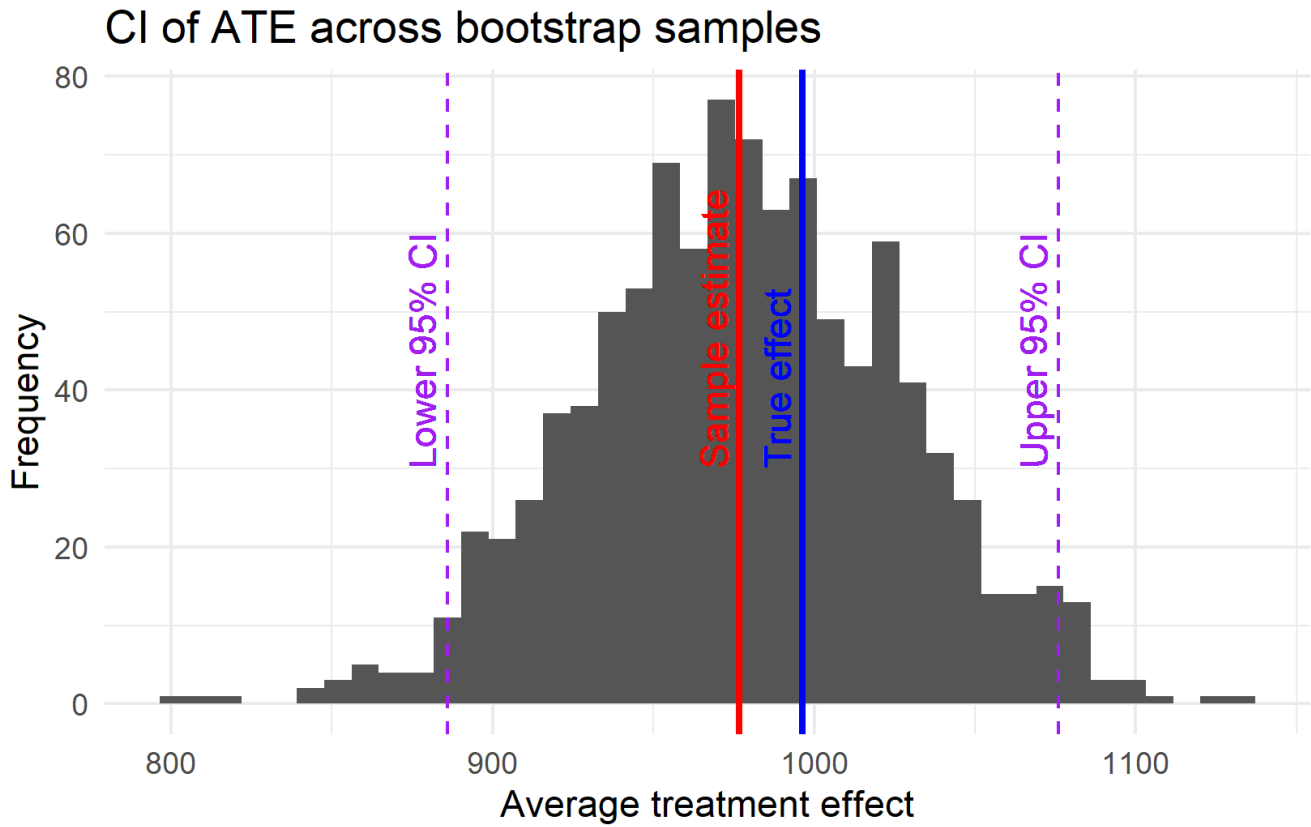
Now with a bigger sample

- With a larger baseline sample ($N=2500$), the distribution of the average treatment effect is tighter



Get standard errors from results

- Now that we have a bunch of estimates, take the 2.5th and 97.5th percentiles to get a type of 95% Confidence Interval[3]



[3] There are a variety of ways to calculate the confidence interval, this is the most intuitive, but it is not the most econometrically sound. See the "Basic bootstrap" or "Reverse percentile" method.

Bootstrapping assumptions

- Your sample is just one random sample from the population of interest
- Bootstrapping assumes that randomness in data is driven by sampling
- Bootstrapping assumes a distribution that is not "highly" skewed
- (Basic) bootstrapping assumes independent and identically distributed
    - But you can do clustering and other forms of correlation, etc.
- Other technical assumptions!

When should I do it?

- The bootstrap simulates the sample distribution of your estimates
- Use it to:

1. Calculate the standard error of your estimates
    - Especially when you can't use analytical formulas (e.g. the median)
2. Look for bias in your estimate: is the average of your bootstraps far off from your actual estimate?
3. Do power simulations on pilot data
4. Generate "training data sets" for machine learning models
5. Explore variation in model predictions as the sample changes
6. Other robustness checks and more

How do I bootstrap?

There are two main requirements:

1. Always use the same sample size as your original sample
    - This ensures the "same" data-generating process and approximates the same randomness
2. Always sample **with replacement**
    - That means you may sample the same observation twice

Variations on sampling

I'm showing you index bootstrapping, where you just grab random observations from your sample with replcaement

There are two main variations on bootstrapping:

1. **Frequency bootstrap**: If your data is a frequency table, you can just randomly assign new frequencies

    - If an observation has frequency 7, that means it occurred 7 times
    - You randomly assign it new frequencies, representing a new sample where frequency still sums to the same amount

2. **Weight bootstrapping**: You can assign weights to each observation and sample with replacement

    - In your original sample, each observation got a weight of 1
    - You assign new weights, so a weight of 1.5 means the observation is 1.5 times more likely to be sampled, .5 means .5 times as likely to be sampled, etc.
    - The non-intger weights just needs to sum to N the observations in your data

Both are powerful ways to do bootstrapping when your data are in a format that makes index bootstrapping hard

Limitations of bootstrapping

- Bootstrapping cannot save you if your sample is biased
- Bootstrapping cannot save you if your sample is too small
- Bootstrapping cannot save you if your sample is not representative

Packages

- The **boot** package in one of many dedicated to bootstrapping
- It handles many cases, but it can get a little slow for big data
- It has built-in Parallel Programming, but it may not work on different systems
- Best to know how to do it yourself as well cause it is pretty easy once you get the hang of it!

What next?

- Go try how to bootstrap in R!
- Better yet, learn to do it in parallel
- Navigate to the lecture activity 13a-bootstrapping-functions-practice

Next lecture: Functions