Studio 10 Bootstrap confidence intervals
18.05, Spring 2022

Overview of the studio

This studio simulates bootstrap confidence intervals type I error rates. We can use the simulations to estimate the true confidence (1− type I CI error rate) of bootstrip intervals with a given nominal confidence. We will also compare the performance of the percentile and basic methods.

Here is a little more detail on what we are hoping to do.

1. We know that a 95% confidence intervals has the following meaning: If 1000 labs each ran an experiment and used their data to make a 95% CI for the mean (or any parameter) then we would expect about 95% of the intervals to contain the true value, i.e. there should be about a 5% type I CI error rate

2. In real life you can never actually check the error rate because you don't know the true value

3. The bootstrap CI is this weird construction. Calling it a 95% CI is plausible, but we can check if it truly is a 95% CI by simulation, where, unlike in real life, we do know the true value of the mean (or other quantity)

We will do these simulations for two distributions.

1. Standard normal distribution. This is symmetric and well behaved.

2. Log-normal distribution. This is highly skewed and certain statistics will give the bootstrap some trouble.

R introduced in this studio

The matrixStats package has functions colMeans2, colMedians, colSds, colQuantiles that are optimized for speed on matrices. This means you can generate all your bootstrap samples at once and put them in a matrix. This will be much faster than doing them one at a time in a loop.

New functions: `rlnorm()`

The R needed is introduced in mit18_05_s22_studio10-samplecode.r.

Download the zip file

- You should have downloaded the file `mit18_05_s22_studio10.zip` from our MITx site.
- Unzip it in your 18.05 studio folder.
- You should see the following R files
```
mit18_05_s22_studio10.r
mit18_05_s22_studio10-samplecode.r
mit18_05_s22_studio10-test.r
```

and the following other files

`mit18_05_s22_studio10-test-answers.html`

Prepping R Studio

- In R studio, open `mit18_05_s22_studio10-samplecode.r` and `mit18_05_s22_studio10.r`
- Using the Session menu, set the working directory to source file location. (This is a good habit to develop!)
- Answer the questions in the detailed instructions just below. Your answers should be put in `mit18_05_s22_studio10.r`
- Solution code will be posted tomorrow at 10 pm

Detailed instructions for the studio

• Go through **mit18__05__s22__studio10-samplecode.r** as a tutorial.

Summary of questions

1. For normal data: compute the simulated 1 CI error rate for empirical bootstrap confidence intervals of the mean, median and standard deviation. Do this for both percentile and basic intervals.

2a. Look up the log-normal distribution on Wikipedia

2b. Repeat problem 1 for log-normal data

Problem 1

Problem 1. Here you will finish the code for the function
`studio10_problem_1(true_mean, true_sd, n_data, n_boot, n_trials, confidence)`

The arguments to this function are:
`true_mean` = the mean of the normal distribution used to generate the data
`true_sd` = the standard deviation of the normal distribution used to generate the data
`n_data` = number of values in the original and each bootstrap sample (original generated from a normal distribution)
`n_boot` = number of bootstrap samples to use in each trial
`n_trials` = number of trials in the simulation
`confidence` = the bootstrap confidence level, e.g. 0.95, 0.9 etc

Your code will simulate finding empirical bootstrap type 1 CI error rates for the mean, median and standard deviation.

Do this by running `n_trials` of the following simulation

1. Generate a normal sample of size `n_data`. Do this using `rnorm` and the given `true_mean` and `true_sd`.

2. Compute the statistics in question (mean, median and standard deviation) of the sample.

3. From the original sample, generate `n_boot` empirical bootstrap samples.

4. Using the bootstrap samples compute the empirical percentile and basic bootstrap confidence intervals with the given `confidence`.

5. Using the (known) true value of the statistics in question, check for a type I CI error for each statistic.

From all the trials, report the type I CI error rate.

Problem 2

In this problem we will explore empirical bootstrap confidence intervals for the log-normal distribution. Go to

https://en.wikipedia.org/w/index.php?title=Log-normal_distribution&oldid=1089459082

Look at the graphs of the pdf and notice how asymmetric they are. For some orientation, read the section 'Occurrence and applications'

Problem 2a. Here you will finish the code for the function

```
studio10_problem_2a(meanlog, sdlog)
```

The arguments to this function are:
`meanlog` = value of `meanlog` parameter in rlnorm
`sdlog` = value of `sdlog` parameter in rlnorm

Go to the Wikipedia page and find the formulas for the mean, median and standard deviation of the log-normal distribution in terms of the parameters to the R function `rlnorm`. Your solution should implement these formulas and print out the values for of the mean, median and standard deviation for the given `meanlog` and `sdlog`

Problem 2b. Here you will finish the code for the function

```
studio10_problem_2b(meanlog, sdlog, n_data, n_boot, n_trials, confidence)
```

The arguments to this function are:

```
    meanlog = value of meanlog parameter in rlnorm
    sdlog = value of sdlog parameter in rlnorm
    n_data = number of values in each sample (Original sample generated using
a log-normal distribution.)
    n_boot = number of bootstrap samples to use in each trial
    n_trials = number of trials to run in simulation
    confidence = the bootstrap confidence level
```

Repeat problem 1 using the log-normal instead of the normal distribution (R: `rlnorm()`). Use the given `meanlog` and `sdlog` values as the parameters to `rlnorm`.

Testing your code

For each problem, we ran the problem function with certain parameters. You can see the function call and the output in `mit18_05_s22_studio10-test-answers.html`. If you call the same function with the same parameters, you should get the same results as in `mit18_05_s22_studio10-test-answers.html` – if there is randomness involved the answers should be close but not identical.

For your convenience, the file `mit18_05_s22_studio10-test.r` contains all the function calls used to make `mit18_05_s22_studio10-test-answers.html`.

Before uploading your code

1. Make sure all your code is in `mit18_05_s22_studio10.r`. Also make sure it is all inside the functions for the problems.
2. Clean the environment and plots window.
3. Source the file.
4. Call each of the problem functions with the same parameters as the test file `mit18_05_s22_studio10-test-answers.html`.
5. Make sure it runs without error and outputs just the answers asked for in the questions.
6. Compare the output to the answers given in `mit18_05_s22_studio10-test-answers.html`.

Upload your code

Use the upload link on our MITx site to upload your code for grading.

Leave the file name as `mit18_05_s22_studio10.r`. (The upload script will automatically add your name and a timestamp to the file.)

You can upload more than once. We will grade the last file you upload.

Due date

Due date: The goal is to upload your work by the end of class.

If you need extra time, you can upload your work any time before 10 PM ET the day after the studio.

Solutions uploaded: Solution code will be posted on MITx at 10 PM the day after the studio.