Studio 11 Bootstrap confidence intervals
18.05, Spring 2025

Overview of the studio

This studio simulates bootstrap confidence intervals type I error rates. We can use the simulations to estimate the true confidence (type I CI error rate) of bootstrap 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 the log-normal distribution. This is highly skewed and certain statistics will give the bootstrap some trouble. Importantly, we don't necessarily know the sampling distributions for all statistics, so constructing vanilla confidence interval could be hard.

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 studio11-samplecode.r.

Prepping R Studio

- In R studio, open studio11-samplecode.r and studio11.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 studio11.r
- Solution code will be posted on Saturday at 4 am

Detailed instructions for the studio

• Go through **studio11-samplecode.r** as a tutorial.

Summary of questions
1a. Look up the log-normal distribution on Wikipedia.
1b. For log-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.
1c. (optional) Plot histograms for the means, medians and standard deviations of the bootstrap samples.

Problem 1

In this problem we will explore empirical bootstrap confidence intervals for the log-normal distribution. Go to
https://en.wikipedia.org/wiki/Log-normal_distribution

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

Problem 1a. Here you will finish the code for the function
`studio11_problem_1a(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 1b. Here you will finish the code for the function
`studio11_problem_1b(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

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 log-normal sample of size `n_data`. Do this using `rlnorm` and the given `meanlog` and `sdlog`.

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 both the empirical percentile **and** basic bootstrap confidence intervals with the given confidence. You will need to produce different confidence interval for the mean, median and standard deviation. So altogether 6 different confidence intervals, 2 different types for each of the 3 statistics.

5. Repeat your calculations from Problem 1a and use the 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 1c.(**OPTIONAL**) Here you will finish the code for the function
`studio11_problem_1c(meanlog, sdlog, n_data, n_boot)`

In this problem you will plot histograms for the means, medians and standard deviations of the bootstrap samples.

Start by drawing a sample of size `n_data` from a log-normal distribution and with the given parameters `meanlog` and `sdlog`. Then, using this sample, generate `n_boot` different bootstrap samples. For each sample compute the mean, median, and standard deviation. Finally, plot 3 histograms for the statistics you computed; a histogram for all the means, a histogram for all the medians, and a histogram for all the standard deviations.

Run your function with `n_data` large, and `n_boot` larger. Do any of the histogram look familiar? What about the histogram for the median, how would you design a confidence interval for that sampling distribution?

Testing your code

For each problem, we ran the problem function with certain parameters. You can see the function call and the output in studio11-test-answers.html. If you call the same function with the same parameters, you should get the same results as in studio11-test-answers.html – if there is randomness involved the answers should be close but not identical.

For your convenience, the file studio11-test.r contains all the function calls used to make studio11-test-answers.html.

Before uploading your code

1. Make sure all your code is in studio11.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 studio11-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 studio11-test-answers.html.

Upload your code

Upload your code to Gradescope.

Leave the file name as studio11.r.

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