CE888: Data Science and Decision Making

Lab 2: Resampling Statistics

Ana Matran-Fernandez

21 January 2019

Institute for Analytics and Data Science
University of Essex

Table of contents

Setting up

Overleaf

Overleaf is an online latex typesetting system. You will need it to create the project for this module.

- ☐ Go to `https://www.overleaf.com` and create an account and a new document.

CE888 repository

- Send me an email NOW with subject “CE888 github” and your username. (e.g., “CE888 github amatra”).
- You need to do this so that we can check your lab practices for this course.
- No need to write anything in the body of the email.
- You don’t need to email me when you finish the practice.
- If you have changed anything in your repository since the last time you were in this computer, make sure you do: `git pull` from the repository folder.
- This will download all the changes you did into your local folder.

Downloading the lab 2 materials

- Go to the Moodle page for this week:
- `https://moodle.essex.ac.uk/course/view.php?id=6683&section=8`
- Download the slides and code for today's practice into your local Github directory (e.g., `/labs/lab2`).
- Unzip the code, commit and push it before you make any changes
  - Go to the folder
  - `git add -A -v`
  - `git commit -m "Message"` (Write your own message!!)
  - `git push origin master`

Data visualisation

Exercise: Plotting

A business is looking at changing their current vehicle fleet and replacing their vehicles with ones used by their competitors. They are have captured the MPG of some of the cars in both fleets.

- ☐ Start Pycharm and create a project in your lab2 folder.
- ☐ Create a file called `vehicles.py`. In this file:
  - ☐ Read the data for the vehicles from `vehicles.csv`
  - ☐ Create and save histograms and scatterplots for the current fleet and the proposed fleet
  - ☐ You can look at the file `salaries.py` for an example of how to do this
- ☐ Once you are done, save your changes in github:
  - ☐ Go inside your lab directory
  - ☐ `git add -A -v`
  - ☐ `git commit -m "Plotting"` (The message should describe what you did!)
  - ☐ `git push origin master`

The bootstrap!

Bootstrap algorithm

`def bootstrap(sample, sample_size, iterations):`

- ☐ Create an array of samples of shape (`iterations`, `sample_size`)
- ☐ Calculate and save the mean of the array (we return it at the end)
- ☐ In each iteration
  - ☐ Get the data corresponding to that iteration (`new_samples[iteration, :]`)
  - ☐ Calculate the mean of the iteration data and store it in an array
- ☐ (At this point you should have an array of `iterations` values)
- ☐ Calculate the lower and upper bounds for a 95% CI (Hint: check the `percentile` function on Numpy)
- ☐ `return data_mean, lower, upper`

Exercise: The Bootstrap (1)

- ☐ In Pycharm, open the `bootstrap.py` file.
- ☐ Implement the bootstrap function (that is currently empty) following the bootstrap algorithm (check previous slide).
- ☐ Save your changes in Github in the usual way.
- ☐ Extra: if you want, add a parameter to the function that allows you to pass the desired CI (and edit the calculation of the upper and lower bounds accordingly).

Exercise: The Bootstrap (2)

The business analysts come up a comparison algorithm that requires the upper and lower bounds for the mean in order to say which fleet is better.

- ☐ Calculate the mean of both samples.
- ☐ Using the bootstrap function that you created:
  - ☐ Find the upper and lower bound of the mean of the current fleet.
  - ☐ Do the same with the new fleet.
  - ☐ Are they comparable? (i.e. is one better than the other?)
- ☐ Save your changes in Github in the usual way.

Exercise: Analysis of results

- ☐ Write a very small text on what you did in your README.md for this lab. Include the plots you generated at the beginning as: `![logo](./scaterplot.png?raw=true)`
- ☐ Write a very small text description of the analysis in Overleaf, download the pdf and put in in Github alongside the rest of your lab2.
- ☐ Once you are done, save your changes in Github in the usual way

Power Analysis

Power Analysis

Calculate the power for a given sample size and alpha

```
def power(sample1, sample2, reps, size, alpha):
```

- ☐ 1: Repeat reps times:
  - ☐ 1.a: generate a new sample from the first sample
  - ☐ 1.b: Generate a new sample form the second sample
  - ☐ 1.c: Compare the two samples and calculate the p-value
- ☐ 2: Return the percentage of times that the p-value was < 1-alpha

Exercise: Power Analysis

- ☐ Code the algorithm from the previous slide on `power.py`.
- ☐ As usual, once you are done, save your changes in Github.
- ☐ Call me or the teaching assistant to show us your work!
- ☐ But don't worry if we don't get to see it.
- ☐ If you didn't email your GitHub username before, make sure you do it before leaving today!