



SYLLABUS: Data Bootcamp

Fall 2020

Data Bootcamp is about nuts and bolts data analysis. You will learn how to analyze economic and business data and enough about computer programming to work with it efficiently.

We will use **Python**, a popular high-level computer language that is widely used in finance, consulting, technology, and other parts of the business world. “High-level” means it is less painful than most (the hard work is being done by the language), but it is a serious language with extensive capabilities.

“Data analysis” means descriptions that summarize data in ways that are helpful and informative. “Bootcamp” is a reminder that expertise takes work. Don’t worry, it’s worth it. **You will be more valuable to current and future employers.** And you will be able to do things more effectively than friends who rely on Excel.

Instructor

Benjamin Zweig

Email: bzweig@stern.nyu.edu

Office Hours: By appointment

Requirements

There are no prerequisites. **We welcome students with no prior programming experience** and have designed the course with them in mind. What you need is the courage to take on a challenge and the patience to fix computer programs that don’t work. That’s a regular occurrence, even for experts.

Getting help

This course has a strong **support system** to help you when you run into a problem – and anyone who codes runs into programs runs into problems. We have myself and a **teaching fellow**.

You can also ask your classmates for help! You can post questions on our discussion group (on NYU Classes). The teaching fellow and I will monitor the discussion group.

The bottom line: **If you’re stuck, ask for help.**

Course website

Everything you need, including this syllabus, is posted on the course website:

<https://NYU.Data-Bootcamp.com>

Deliverables and grades

This course divides naturally into two parts. The first part is an introduction to those aspects of the Python programming language useful for data analysis. We cover this material with as many applications to real data as we can think of. The second part covers advanced topics and ends with a project of your own. The goal is for you to have a piece of work you can show potential employers to illustrate your quantitative skill set. Both parts include a number of graded deliverables. The idea is to do some work all the time rather than lots of work once in a while.

Graded work includes:

- **Code Practice.** There are eight assignments over the course of the semester. They are a great way to develop your skills and come to the following class with questions about what you may not yet understand.
- **Midterm Project.** Midway through the semester, you will be asked to derive unique insight from a dataset that will be delivered to you and visualize that.
- **Final Project.** We work our way up to the project one step at a time, starting with idea generation and ending with a professional piece of data collection and analysis that you can share with potential employers. We have found that the quality of final project had a surprisingly low correlation to previous programming experience. A little thought and effort go a long way in creating an interesting project.

Due dates are posted on the course website.

Dates are not negotiable. Anything handed in late will get a grade of zero.

All your work should be clean and professional. Your grade depends on it.

Final grades will be computed from

Code practice	45%
---------------	-----

Midterm project	20%
Final Project	35%

Final grades are not subject to any fixed distribution or curve. The number of A grades, for example, will depend only on your performance in the course. If you make a good-faith effort, we expect it to be hard to get less than a B. We are the sole judges of what constitutes good-faith effort.

Recommended work habits

Python is not something you can learn from reading a book and attending lectures. You need to write programs – the more the better – to understand how they work. Think about how you'd learn to play basketball or soccer; reading and listening to lectures aren't enough, you need to do it. We'll do a lot of programming in class, but it's essential that you follow up outside of class.

Here's how.

Write & Review. After each topic, we recommend you:

- *Write:* Shortly after class, write down everything you remember without looking at your notes or the book. Note things you don't understand.
- *Review:* Read the relevant section of the book. Fill in the gaps. Ask for help with anything you still don't understand.

Practice. For the first half of the term, each topic has an assignment that covers the same material. We suggest you do them, even the ones that aren't graded.

We also recommend you **practice coding whenever you have the chance**. Start small. Write short programs to do anything that crosses your mind. Use Python to do things you would ordinarily do in Excel. Try doing assignments from other courses in Python. At first this will be more work than doing it by hand or in Excel, but once you have some experience it will typically be easier in Python. Even if that's not the case, the practice will expand your skill set.

Policies

Ethics, disabilities, and many other things are governed by NYU and Stern policies. If you have questions about them, please ask.

On graded work: You may discuss assignments with anyone (in fact, we encourage it), but anything you submit, including your code, should be your own. Exams should be entirely your own work.

On disabilities: If you have a qualified disability that requires academic accommodation, please contact the Moses Center for Students with Disabilities (CSD, 212-998-4980) and ask them to send us a letter verifying your registration and outlining the accommodation they recommend. If you need to take an exam at the CSD, you must



submit a completed Exam Accommodations Form to them at least one week prior to the scheduled exam time to be assured accommodation.