

## Data Bootcamp: Syllabus

Revised: January 16, 2016

### Overview

Data Bootcamp is about nuts and bolts data analysis. You will learn about economic, financial, and business data, and enough about computer programming to work with it effectively. Applications include some or all of: leading economic indicators; emerging market country indicators; bond and equity returns; stock options; income by zip code; long tail sales data; innovation diffusion curves; and many others. We will use Python, a popular high-level computer language that's widely used in finance, consulting, technology, and other parts of the business world. "High-level" means it's less painful than most (the hard work is done by the language), but it's a serious language with extensive capabilities. "Data analysis" means primarily graphical 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 more things more efficiently than colleagues who rely on Excel.

If you're not convinced — or even if you are — we have a more elaborate sales pitch on the [course website](#).

There are two sections of this course, one for undergrads (ECON-UB.0232) and one for MBA students (ECON-GB.2313). The content is similar, but the schedules and teaching teams are different.

### Requirements

There are no prerequisites. **We welcome students with no prior programming experience** and have designed the course with them in mind. What you will need is the **courage** and **patience** to fix computer programs that don't work. That's a regular occurrence, even for experts. We will show you how to work your way through it.

Our one requirement is that **you must bring a laptop computer to class**. It should be your own computer, or at least one you can install new programs on. We will use it constantly in class, writing and correcting (mostly short) programs.

### Getting help

This course has a strong support system to help you when you run into problems — and anyone who codes runs into problems. Each section has two teachers and a teaching assistant, all of them happy to answer your questions. You can also post questions on our discussion group (see below).

The bottom line here is: **If you're stuck, ask for help**. Really. Don't be a hero, ask for help.

## Course website and discussion groups

Everything you need for the course, including this document, is posted on the **course website**:

<http://databootcamp.nyucon.com/>

The book has its own site:

<https://www.gitbook.com/book/davebackus/test/details>

There's also a link to it on the course site. We will not use NYU Classes.

We have set up **announcement and discussion groups** (Google groups) to make announcements and post questions. If you have coding questions, you can post them there. If you know the answer to someone else's question, you can post that, too. The links are

UG: [https://groups.google.com/forum/#!forum/nyu\\_data\\_bootcamp\\_ug](https://groups.google.com/forum/#!forum/nyu_data_bootcamp_ug)

MBA: [https://groups.google.com/forum/#!forum/nyu\\_data\\_bootcamp\\_mba](https://groups.google.com/forum/#!forum/nyu_data_bootcamp_mba)

You can access the same links from the course website. Follow the instructions to join under whatever email address you intend to use.

## Deliverables and grades

The 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. It ends with an exam. The second half 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 halves 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**. We don't believe the latter will work.

Graded work includes:

- **Code Practice.** There are three such assignments in the first month. We encourage you to do all of them — they're good practice — but your grade will be based on the best two. We will also distribute Optional Code Practice "assignments," but they will not be collected or graded. They are, however, a good way to develop your coding skills.
- **Examination.** The exam will cover the Python material from the first half of the course. It takes 75 minutes and will be held in class. It will involve coding. **Bring your computer.** We suggest you **bring one page of notes**. If the internet is working you will be able to look things up on there. If not, this will save you. It's also a good study tool: When you decide what to include, you'll be organizing your thoughts about what you've learned.

- **Project.** We work our way up to a project one step at a time, starting with idea generation and ending with a professional piece of data collection and analysis. The structure of the project is laid out in a separate document.

**Due dates** are posted on the course website. Assignments, whether code practice or parts of the project, are due at the start of class on the specified dates. **Dates are not negotiable. Anything handed in late will get a grade of zero.**

**We expect all of your work to be clean and professional.** Your grade depends on it.

**Final grades** will be computed from

Code Practice (best two of three)	25%
Exam	25%
Project	50%

Final grades are not subject to any fixed distribution. 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 the book and attending lectures. You need to write programs — the more the better — to learn 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.

**Read/Write/Review.** After each topic, we recommend you:

- **Read:** Read the relevant chapter in the book and work through all the exercises.
- **Write:** With the book closed, write down everything you can remember. Note things you don't recall for later reference.
- **Review:** Later on, go through what you wrote down and use the book to fill in any gaps. Ask for help with anything you still don't understand.

**Practice.** 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'd 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.

## Pacing

The course is designed to be cover material at whatever pace the class is capable of. The topics should take roughly a week each, but we can scale that up or down as needed. If you're an expert, don't worry, we'll cover a lot of material either way.

## Other questions

You can find answers to common questions on the course website. You can also post questions on the discussion group; see the link above and on the course website. For others, email Dave Backus, ([db3@nyu.edu](mailto:db3@nyu.edu)), the co-teacher, or the teaching fellow; their contact information is on the course website.

## Viewing and printing pdf files

This document and most others in the course are pdf's that we hand out in class and post on GitHub. If you view them online, the links won't work; that's an unfortunate feature of GitHub's pdf viewer. You should instead download them to your own computer and open them there. To download them, click on the Raw button above the document. To view them, use Adobe Acrobat Reader or the equivalent (FoxIt, Sumatra, etc). The links should then work.

For Mac OS users, Preview has this problem and others; for example, some fonts don't display or print. The solution is to install Adobe Acrobat Reader or the equivalent and set it up as your default for pdf files.

## 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 me 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.