

Syllabus: Data Bootcamp

ECON-UB.0232 | Spring 2019

Revised: April 16, 2019

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; education and income; 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 friends 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](#).

About the Instructor

[Professor Michael Waugh](#)

Email: mwaugh@stern.nyu.edu

Office: KMC 7-74

Phone: 212-998-0288

Office hours: TBA

Where and When

Meeting times: Tuesday and Thursday, 2:00 - 3:15

Meeting place: TISCH UC24

Important Dates

- Take Home Midterm: March 13. Quick info: Project covering python fundamentals 1-2, intro to pandas, basic plotting.
- Guided Project: Month of April. Quick info: We will work on a project together in steps using an industry grade data set.

- Three Project Ideas: April 11. Three project ideas, briefly flushed out and potential data sources. .
- Final Project Proposal + Data Report: May 2nd. Jupyter Notebook with final proposal.
- Final Project Due Date: Monday May 20th

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.

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

We have a myself and a teaching fellow to help. Below is the contact info for our teaching fellow this semester.

Teaching Fellows

Aditya Vashistha

Email:

Office hours: TBA

My email policy. During day time on weekdays, I generally answer email (mwaugh@stern.nyu.edu) quickly. During evenings and weekends, you should expect that my response will be less prompt. My official policy is that I reserve the right to get back to you within 24 hours. If I don't, feel free to email me again.

Sometimes you want to know more. Below are two resources that I find quite useful:

- [Python for Data Analysis \(2nd Edition\)](#), by Wes McKinney. Definitive guide to Python for data science and scientific computation, by the creator of Pandas.

- [Python Data Science Handbook](#), by Jake VanderPlas. Very similar focus as above, extensive chapter on Machine Learning. Great GitHub repository with lots of nice, worked out notebooks.

Course website

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

https://nyu.data-bootcamp.com/undergrad_outline.html

Note we will NOT use NYU Classes.

The **book** has its own site,

<https://nyudatabootcamp.gitbook.io/thebook/>

There's a link to it on the course site.

Deliverables and grades

The course divides naturally into three parts.

- The first part is an introduction to basic aspects of the Python programming language.
- The second part covers data analysis aspects of Python. The key tools we will learn about are Pandas and Matplotlib. We cover this material with as many applications to real data as we can think of. The take home midterm will cover this material.
- The third covers advanced topics (mostly advanced functionality of Pandas and visualization) and ends with a project of your own. The ultimate goal is for you to have a piece of work you can show potential employers to illustrate your quantitative skill set. The “guided project” will apply the tools we learn in a controlled, but real-live like setting.

All 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**. We don't believe the latter will work.

Graded work includes:

- **Code Practice.** There are four such assignments in the first month and a half. We encourage you to do all of them—they're good practice—but your grade will be based on the best two. We find that people who finish these assignments tend to keep up with the material better and these are easy points to get in terms of grades.

These assignments have questions that range from easy to moderately difficult, the latter marked challenging. They're all good practice, but if you're new to programming you might want to skip the challenging ones. If you get 70-80 percent of them, you're on the right track.

- **Exam.** There will be one take home exam. Details to be provided later.
- **Guided Project.** This will be a directed project using advanced Pandas and visualization skills on an industry grade data set. This will be concurrent with the later half of the semester and help set you up for executing your own independent project. This will be divided into three steps that build on each other. We encourage you do to all of them—they're good practice—but your grade will be based on the best two.
- **Independent 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.

You may work in groups of up to size two. Group projects will be evaluated on a different basis relative to individual projects.

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

Participation / Professionalism	5%
Code Practice (best two of three)	15%
Midterm	15%
Guided Project (best two of three)	15%
Independent Project	50%

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.

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—gaps, we call them.

- *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.

Pacing

The course is designed to be cover material at whatever pace the class is capable of. The topics should take roughly one 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.

Professional Behavior

In the interest of having a high-quality experience for all, your classmates and I ask that you:

- Attend. They say "eighty percent of success is just showing up." I see it often, those who show up perform systematically better.
- Arrive to class on time and stay to the end of the class period. Chronically arriving late or leaving class early is unprofessional and disruptive to the entire class.
- Eating or drinking in class in any way that interferes with class decorum is inappropriate.
- Be respectful to myself, the teaching fellow, and your classmates.
- Join your classmates. Avoid the last row. Be Engaged!

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 me or the teaching fellow.

Policies

- **General Behavior.** The School expects that students will conduct themselves with respect and professionalism toward faculty, students, and others present in class and will follow the rules laid down by the instructor for classroom behavior. Students who fail to do so may be asked to leave the classroom.
- **Collaboration on Graded Assignments.** 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. Violation of this policy will result in a failing grade for the course.
- **Academic Integrity.**
 - Integrity is critical to the learning process and to all that we do here at NYU Stern. As members of our community, all students agree to abide by the NYU Stern Student Code of Conduct, which includes a commitment to:
 - Exercise integrity in all aspects of one's academic work including, but not limited to, the preparation and completion of exams, papers and all other course requirements by not engaging in any method or means that provides an unfair advantage.
 - Clearly acknowledge the work and efforts of others when submitting written work as one's own. Ideas, data, direct quotations (which should be designated with quotation marks), paraphrasing, creative expression, or any other incorporation of the work of others should be fully referenced.
 - Refrain from behaving in ways that knowingly support, assist, or in any way attempt to enable another person to engage in any violation of the Code of Conduct. Our support also includes reporting any observed violations of this Code of Conduct or other School and University policies that are deemed to adversely affect the NYU Stern community.

The entire Stern Student [Code of Conduct](#) applies to all students enrolled in Stern courses. Discussion of this code of conduct and its application to the undergraduate college is here: [Undergraduate College Conduct](#). **Any violation of the policies pertaining to Academic Integrity will result in a failing grade for the course.**

Students with disabilities

If you have a qualified disability that requires academic accommodation during this course, 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 guaranteed accommodation.