

# Syllabus: Data Bootcamp

## ECON-UB.0232 | Spring 2018

Revised: January 16, 2018

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](mailto:mwaugh@stern.nyu.edu)

Office: KMC 7-74

Phone: 212-998-0288

Office hours: TBA

### Where and When

Meeting times: Tuesday and Thursday 11am-12:15pm

Meeting place: KMC 3-70

### Important Dates

- Mini-Midterm #1: February 22nd, 2018. Quick info: Python fundamentals 1-2, 30 minutes, in class, open book, open internet if the wireless is up, bring one page of notes.
- Mini-Midterm #2: March 22nd, 2018. Quick info: Intro to Pandas and Matplotlib, 30 minutes, in class, open book, open internet if the wireless is up, bring one page of notes.

- Three Project Ideas: March 29, 2018. Jupyter Notebook with three project ideas, briefly flushed out and potential data sources. Hard copy due at 5pm.
- Final Project Proposal + Data Report: April 20th, 2018. Jupyter Notebook with final proposal. More details to be provided. Hard copy due at 5pm.
- Final Project Due Date: End of Day May 14, 2018

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

### Teaching Fellows

Vineetha Kutty  
Email: [vkk242@nyu.edu](mailto:vkk242@nyu.edu)  
Office hours: TBA

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

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 and discussion groups

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

[https://nyu.data-bootcamp.com/undergrad\\_outline/](https://nyu.data-bootcamp.com/undergrad_outline/)

**Note we will NOT use NYU Classes.**

The **book** has its own site,

<https://www.gitbook.com/book/nyudatabootcamp/data-bootcamp/details>

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 mini-midterm #1 covers this material.
- 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 mini-midterm #2 covers 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.

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 three. 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.*

- **Exams.** There will be two "mini" exams. These exams are mini in the sense that they are only 30 minutes long. It will involve coding. **Bring your computer.** It will be open book, open internet if the wireless is up. We suggest you also **bring one page of notes**. If the internet is working you will be able to look things up on there. If not, your notes 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 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

Participation / Professionalism	5%
Code Practice (best three of four)	15%
Exam #1	15%
Exam #2	15%
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.

## 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, links don't work and 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 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.