PPHA 30536: Data and Programming for Public Policy II

Jeff Levy
levyjeff@uchicago.edu
Keller 3101

Fall Quarter, 2020

Course Information

September 29th - December 2nd, 2020

Regardless of which section you enroll in, the schedule will be:

Lecture 1:
Tuesday 11:20 AM - 12:40 PM, Keller TBD or Zoom

Lecture 2:
Wednesday 10:50 AM - 12:10 PM, Zoom only

All lectures will be available asynchronously on Canvas.

Labs (Zoom only, all times CST):
Monday 2pm
Wednesday 2pm & 8pm
Thursday 2:40pm
Friday 10am & 8pm
Sunday 8pm

Office Hours

Please email me to schedule meetings over Zoom.

Teaching Assistants

Sarah Gill - gills@uchicago.edu
Stephanie Gomez - stephanieramos@uchicago.edu

Emily Zhang - zhange@uchicago.edu
Merritt Smith - merrittsmith@uchicago.edu

Prerequisites

The course PPHA 30535, Data and Programming for Public Policy I, is required to take this course. If you did not take PPHA 30535, or took PPHA 30535 sections 1 or 2 (taught in R), you must email me for approval, and will be required to demonstrate proficiency with Python and Pandas.

Note that Data and Programming for Public Policy II in the R language will be offered Winter quarter 2020.

Course Objectives

This course will build directly on the material covered in PPHA 30535. We will assume a grasp of the Python skills from the previous class at the start, so that we can focus on practical applications to research. Whereas the goal of the first class was introduce Python as a tool for data analysis, and to prepare students for internship-level policy research positions, the goals of this course will be to:

1. Go from simply applying Python to solve research questions, to applying Python professionally, in a way that supports code maintenance, collaboration, efficiency, and readability
2. Deepen existing skills; for example, we will go from learning to create plots to discussing the principles of creating good plots
3. Broaden into new skills that require a higher level of Python proficiency
4. Prepare for the post-graduation job market

Additionally, we will have guest speakers throughout the quarter depending on their schedules. Confirmed so far:

- **Urban Institute** - Jessica Kelly & Michelle Menezes - Research Programming
- **UChicago Crime Lab** - Emma Nechamkin - Data Scientist
- **UChicago Biological Sciences Division** - Mark Nardone - Chief Information Security Officer

Speakers are subject to change, and will require a flexible course schedule on a week-by-week basis.

Software and Resources

The software and resources for this class are identical to the Python sections of PPHA 30535 in the spring.

There are no required text books for this class. Python is extremely well supported online. I expect students will primarily be using the official Python documentation and StackOverflow, which will be discussed in class.

I do suggest purchasing the text Python for Data Analysis 2nd Edition by Wes Mckinney, which is available online. Not only is it very useful both as a quick reference and when read comprehensively as a guide, it is also written by the author of Pandas, the package used for data analysis in Python. The package is free and open-source, so this is also a good way of giving back to the creator. If you purchase this it is very important you get the 2nd edition, as the original is outdated. Note that this text is not required and will not be referenced directly during class.

There are two pieces of software that are required for this class, and two that I suggest, all of which are free. Since installation and setup is covered in Data and Programming for Public Policy I, we will not go over it again. Required:

- *Please have ready on day one* the Anaconda Python Distribution installed already. Please select version 3.7, though 3.6 will also work if you installed it previously. No version of Python 2.x is acceptable.

- *Please have ready on day one* the GitHub Desktop installed. You may also use the Git command line interface if you have it installed from before and know how to use it, though it also comes as part of the Desktop download.

Suggested:

- The Atom open-source text editor. I suggest Atom, but perfectly viable alternatives include Sublime, Vim, and many others. You may also chose to use one of the IDEs (integrated development environments) that comes with Anaconda, such as Spyder.

- The Notepad++ text editor. This is strictly optional, but I find this lightweight text editor useful for viewing raw data alongside my code when necessary. You can choose it (or others like it) as your text editor for coding, as well.

Note that overall, the software environment you choose for developing code is entirely personal preference. You simply must have some distribution of the programming language you will work in, some place to write your code, and some place to run your code.

Attendance

You do not have to be on campus to take this class. All in-person content is optional, and will be streamed over Zoom, with recordings available on Canvas.

Attendance to a minimum of one lab per week is mandatory, and graded. The labs are run by the TAs, and are listed at the top of the syllabus. Anyone from either section can attend any of the labs. You may also, of course, attend more than one in a given week if you choose.

There is no attendance policy for lecture, and you don't need to give me excuses not to attend. However, you will be fully responsible for the material covered in class, and while code from class may be posted, it will rarely if ever come with the explanations provided in class.

If you experience issues with attending class or completing work due to child care, please speak with me directly so we can find an accommodation.

Academic Integrity

All code you turn in must be your own. Do not share your code with your classmates, or ask others for theirs. That said, the practice of writing code is very often a collaborative one. To avoid academic dishonesty, and a potential failing grade, please follow these guidelines:

1. You MAY **search for help online** (e.g. StackOverflow)
   - You must always cite the source by leaving a link to it in the comments of your code
   - You MAY NOT copy verbatim - find inspiration and then rewrite it
   - You MAY NOT take solutions to problem sets from online
2. You MAY **work with your classmates**
   - You must always cite the individuals you collaborate directly with by including their names in the comments at the top of your program
   - You MAY NOT share or look at each other's code
   - You MAY share output (e.g. plots or error messages)
   - You MAY discuss concepts and theory (e.g. using a whiteboard)
3. You MAY participate in **discussions on Piazza**
   - You MAY share generic or pseudo code, and ideas
   - You MAY NOT share specific code from your own work
4. When explicitly allowed, you MAY **work in groups**
   - If groups are optional, you must declare your group the day the assignment is given
   - You will collaborate, share code, and submit only one assignment

It is very important that you use proper citations. If you turn in an assignment that the grader deems to be too unoriginal (e.g. your solutions too closely follow a solution found online, or another classmates), but you have citied all the sources, then you may be allowed a chance to redo your work. If the same thing happens but you have not cited the sources, you will receive a failing grade and possibly be subject to other sanctions under the university's Academic Integrity guide.

The above rules apply to interactions with your classmates and the internet. You may present your code and questions to the professor or the TAs at any time.

Homework, Exams, and Grading

There are no exams for this class. Your grade will consist of weekly assignments, lab attendance, and a project. Assignments will generally be due within a week of assignment, over GitHub Classrooms. Late work is not accepted without prior approval or with a documented emergency.

Your grade will be calculated as 60% weekly assignments, 30% final project, and 10% weekly lab attendance. A minimum of 60% is required to pass this course. Among those who pass, final grades will use the following curve: 1/3 A, 1/4 A-, 1/4 B+, 1/12 B, 1/12 B-.

Course Outline

*This exact outline is **tentative**, and will almost certainly change to accommodate the schedules of guest speakers.*

Week 1

- September 29th - Introduction, project discussion
- September 30th - Code generalization and organization, functions, PEP8

Week 2

- October 6th - Irregularly shaped data
- October 7th - Parsing data from PDF documents

Week 3

- October 13th - Natural Language Processing 1
- October 14th - Natural Language Processing 2

Week 4

- October 20th - Guest speaker 1
- October 21st - Data visualization 1

Week 5

- October 27th - Data visualization 2
- October 28th - Urban Institute guest speakers

Week 6

- November 3rd - Spatial data 1
- November 4th - Spatial data 2

Week 7

- November 10th - Group project presentations
- November 11th - Group project presentations

Week 8

- November 17th - Intro to Machine Learning in Python 1
- November 18th - Intro to Machine Learning in Python 2

Week 9

- November 24th - Other common data languages 1
- November 25th - UChicago Crime Lab guest speaker

Week 10

- December 1st - Other common data languages 2
- December 2nd - Creating good code samples, wrap-up, additional topics