PPHA 30535: Data and Programming for Public Policy I (Python)

Jeff Levy
levyjeff@uchicago.edu
Keller 3101

Spring Quarter, 2021

Course Information

March 29th - May 26th, 2021

Asynchronous lecture posted every Monday and Wednesday AM. The official scheduled class times are optional labs, therefore you may take this class even if the scheduled time conflicts with another course, or if you are in a timezone that makes it impractical.

Labs (on Zoom, all times CST)			
Tuesday	9:30	AM	Jeff Levy
Thursday	9:30	AM	Jeff Levy
TBD	TBD		TBD

Office Hours

Please email me to schedule meetings over Zoom. I am available most days.

Teaching Assistants

TBD

Language Selection

This class will use the Python programming language. According to the 2019 Stack Overflow Developer's Survey, Python is the 4th most popular programming language in the world, and the most popular choice of data professionals. It was also voted the 2nd “most loved” programming language in the world. Python is easy and enjoyable to work with, while offering the power and flexibility of a general-purpose programming language. I am confident that a student completing this course will learn both a highly useful, modern language that will apply to their future work with policy data, as well as the underlying programming principles that can be applied to any other language the student encounters.

In the spring this course is also offered in the R language, taught by professor Sobrino, which together with Python makes up the two most important open-source languages of research programming and data science. Both languages have their respective strengths, and I would recommend either to an aspiring researcher. While the topics vary slightly based on the strengths of the language, the R and Python sections are carefully synchronized by expected work load, grading, and broad topics covered.

If you will be following the Certificate in Data Analytics path of Data and Programming Skills II (PPHA 30536) in the fall, and Machine Learning in the spring (PPHA 30545), note that there will be Python and R sections of both of those classes as well. You should ideally plan to take the same language for all three classes, particularly for Data and Programming I and II, as the classes are designed to build on one another. You can read my syllabus for that course in Python here, and in R here.

Course Objectives

This programming and data course is geared toward public policy students who have either no past programming experience, or minimal experience primarily in proprietary platforms like Stata and SAS. The goal is to prepare students to reliably write their own research code, and for entry-level positions working in policy research, such as (but not limited to) research assistants or interns at government agencies, policy institutions, or think tanks. The content is drawn heavily from the professor's years of experience working at the Brookings Institution and the Urban Institute in Washington, DC. Over the quarter, students will learn:

- The basics of research programming: the languages used, how to set up needed software, coding best practices, and how to solve new tasks.
- The Python programming language:
  - Beginning Python, including logic control, loops, functions, classes, methods, and input-output
  - Data wrangling and visualization
  - Data APIs and web scraping

This course will take a narrow view of computer programming, focusing on learning skills necessary to work as a policy researcher, while passing over some more advanced skills that are common in computer science. This excludes, for example, an understanding of multiprocessing, unit testing, or the creation graphical user interfaces (GUIs). However, the skills learned in this class will provide a foundation for the student to pursue these topics in the future should they desire.

Software and Resources

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.

For the data sections, however, I 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 comphrehensively 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.

There are two pieces of software that are required for this class, both of which are free:

- *Please have ready on day one* the Anaconda Python distribution. Please select version 3.8, though most versions of 3.x will work if you, for example, had them installed from earlier. No version of Python 2.x is acceptable.
- *Please have ready on day one* the GitHub Desktop application. You may also use the Git command line interface if you know how to use it, though we will not be teaching the command line in this class.

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. **For those of you who have used RStudio before, my suggestion is that you choose Spyder**, which comes with Anaconda Python, since it looks very similar. We will discuss software the first week of class.

Attendance

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

Attendance to **at least one lab** per week is mandatory, and graded. Lab times are listed at the top of this syllabus.

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 or projects in this class. Assignments are due on GitHub Classroom just before midnight. Below is the list of assignments, with the date given (Mondays) and the date due (Sundays).

- Homework 1: *Python basics, for loops, conditionals* - Apr 5th-Apr 11th
- Homework 2: *Functions and classes I* - Apr 12th-Apr 18th
- Homework 3: *Functions and classes II* - Apr 19th-Apr 25th
- Homework 4: *Pandas I* - Apr 26th-May 2nd
- Homework 5: *Pandas II* - May 3rd-May 9th
- Homework 6: *Data visualization* - May 10th-May 16th
- Homework 7: *Web scraping* - May 17th-23rd

Your grade will be calculated as 90% weekly assignments, 10% weekly lab attendance. You must achieve a 60% or greater in the class in order to pass. All grades that meet this level will use the following curve: 1/3 A, 1/4 A-, 1/4 B+, 1/12 B, 1/12 B-.

Late Assignments

Every student has **four** 12-hour extensions available to them during the quarter. Those extensions can be used on any homework assignment at any time, and require no advance notice nor excuse to be given. Turning in an assignment at any time after the deadline will result in as many of your 12-hour extensions as necessary, up to the maximum, being used automatically. These extensions are used in complete blocks of time - e.g. turning an assignment in 12 hours and 30 minutes late will use two of your three extensions.

Once your extensions are used up for the quarter, all assignments will be penalized at a rate of 5% per 12-hour block. While you may use your extensions just because you need more time to work on an assignment, please note that this penalty will apply even if future illnesses, technical problems, or family issues arise.

Only issues of sufficient magnitude that academic affairs is involved in the discussion can qualify for exceptions.

Course Outline

This outline may be subject to change. Given date is when the pre-recorded lecture will be posted.

Week 1: Introduction - No homework

1. March 29th: Introduction, software review and setup
2. March 31st: Setup, GitHub basics

Week 2: Python basics - Homework 1

1. April 5th: Data types
2. April 7th: Logic control statements and loops

Week 3: Python functions and classes - Homework 2

1. April 12th: Functions and lambdas
2. April 14th: Classes and methods

Week 4: Python functions and classes - Homework 3

1. April 19th: More functions and classes
2. April 21st: More functions and classes: creating a game

Week 5: The Pandas dataframe - Homework 4

1. April 26th: Pandas I
2. April 28th: Pandas II

Week 6: More Pandas - Homework 5

1. May 3rd: Pandas III
2. May 5th: Pandas IV

Week 7: Data visualization - Homework 6

1. May 10th: Matplotlib I
2. May 12th: Matplotlib II

Week 8: Introduction to web scraping - Homework 7

1. May 17th: Using data APIs; introduction to html
2. May 19th: Requests and BeautifulSoup

Week 9: Advanced topics - No homework

1. May 24th: NumPy and Statsmodels
2. May 26th: Intro to big data