THE UNIVERSITY OF CHICAGO
HARRIS SCHOOL OF PUBLIC POLICY

PPHA 30537: Data and Programming for Public Policy I – Python Programming

Dr. Jeffrey A. Levy
levyjeff@uchicago.edu
Keller 3101

Head TA: Jonas Heim
TAs: Nicholas D'azevedo, Polina Rozhkova, Brendon Krall, Emma Steyaert, José I. Villalobos González

Spring Quarter 2023
Section 1: T, Th 2:00 PM – 3:20 PM Keller 1002
Section 2: T, Th 3:30 PM – 4:50 PM Keller 1002

Office Hours for the professor can be reserved at this Calendly link, while remote lab sessions with the TAs are listed on the Canvas Zoom page.

Course Description

In this course, aspiring researchers will study rigorous data and programming using Python. As one of the most utilized (3rd) and most desired (1st) programming languages in the world, Python is an excellent choice for a new researcher to focus on. Python emphasizes a clear syntax, making code easy to learn and easy to read, while remaining both powerful and flexible. This makes it an ideal platform in which to learn the basics of data analysis in a way that applies to any programming language. While proprietary platforms such as Stata and SAS continue to play an important role in public policy research, newer open-source languages like Python and R have grown rapidly in usage. A good researcher in these fields must be able to adapt by changing tools (languages) as called for by the project.

Generations of researchers and practitioners have grown up in a computing environment dominated by this small number of proprietary computing platforms while relying on ad hoc coding skills acquired through trial and error. This imposes real costs, including the inability to collaborate with researchers using other platforms, difficulty picking up new skills, the inability to find needed functions that only exist in a different language, and worst of all, mistakes that taint results while hiding in sloppy code and bad practices.

This programming and data course is geared toward public policy students who have either no past programming experience, or minimal experience in other platforms. While the course covers basic programming, the focus wherever possible will be on applications to real-world data and research. It is designed to continue seamlessly into *PPHA 30538: Data and Programming for Public Policy II – Python Programming* in the autumn, which will culminate in a final research project covering topics from both classes.

Modes of Engagement

Instruction for this class will have four primary elements:

- New content will be introduced in **asynchronous lectures** posted to Canvas around noon on the day before scheduled class time. I aim to keep these below 30 minutes in length.
- The scheduled lecture times will be used as **live labs**, in which we delve deeper into the content introduced in the lectures, and work through examples in groups.
- Weekly optional **office hours** for the professor and TAs, where individuals can get one-on-one help with questions.
- An optional **discussion board** for questions and discussions outside of office hours and class.

Learning Objectives

Technical goals:

- Learn to write basic Python and understand its syntax.
- Learn the tools of data analysis in Python.
- Gain a deeper understanding of how Python works "under the hood".

Non-technical goals:

- Practice good programming and data principles that are relevant to working in other languages, such as R, Stata, or SAS, and how to make informed choices between them and Python.
- Understand how good programming practices relate to collaboration and reproducible research.
- Become comfortable looking up new programming skills and information using online resources.
- Develop skills that apply directly to summer internships working with data.

Assessment and Grading

Your progress in the learning objectives will be assessed in two ways:

In-class quizzes (15%) – Each class will have a brief (5 minute, 2-3 question) quiz on Canvas that will cover a core skill or concept from lecture. These will provide important instant feedback on the material.

Take-home assignments (85%) – Weeks 2-8 will have coding assignments that ask students to use class concepts to solve research programming questions. Assignments will test your ability to work on a question with a starting place and a broad goal, mimicking real-world research tasks wherever possible.

This class requires a 60% or above to pass and is not curved. All passing grades will use the following intervals:

A	[95% - 100%]	B+	[85% - 90%)	B-	[60% - 80%)
A-	[90% - 95%)	B	[80% - 85%)		

Class Policies

No **attendance** is taken, but quizzes can only be completed in the classroom. Your lowest two quiz scores are dropped automatically to account for unexpected absences or illness.

Assignments **must be turned in** using GitHub and Gradescope, a process we will cover in week one. General feedback according to an assignment-specific rubric will be provided through Gradescope approximately one week after the due date.

Regrade requests must be submitted on Gradescope with a (polite) explanation, which will then be re-evaluated by the original grader. Continued disagreement may be escalated to the head TA first, and finally to the professor. All regrade requests may result in a full regrade and potentially a higher or lower score. See the Gradescope Guidelines document on Canvas for additional important details.

Every student has **four 12-hour late tokens** available to them during the quarter. Those extensions will be automatically applied to any late take-home assignments and require no excuse to be given. These extensions are used in complete blocks of time – e.g. turning in an assignment 12 hours and 30 minutes late will use two tokens. Once your late tokens are used up for the quarter, all assignments will be penalized at a rate of 5% per 12-hour block. These tokens are intended to cover ordinary illness, family events, and so on – only issues of sufficient magnitude that academic affairs is involved in the discussion can qualify for exceptions. Once solutions have been posted to the class (generally Wednesday), no further assignments may be turned in.

See the **academic integrity policy** and a general **grading rubric** on the Canvas course page.

Support

Your mental and physical health is important. As graduate students, I recognize that you are all under immense pressure to achieve academic excellence alongside maintaining personal and often professional lives. Please take care of yourselves and each other, and speak to me if, for any reason, you are having difficulty keeping up with the course. Many other sources of support are available, including:

Find the Harris Student Affairs office here.

Learn more about accommodations for students with disabilities here.

See the Harris academic support programs, including tutoring and code labs, here.

Software and Resources

There are no required textbooks, as 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. The text Python for Data Analysis 3rd Edition by Wes McKinney may be helpful as both a quick reference and when read comprehensively as a guide, but will not be referenced directly in class.

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

- The Anaconda Python distribution
- The GitHub Desktop application

Course Outline

Week 1: Introduction — No homework

- March 21st: Introduction, software review, and setup
- March 23rd: Setup, GitHub basics

Week 2: Python basics — Homework 1 (Mar 27th – Apr 2nd)

- March 28th: Data types
- March 30th: Logic control statements and loops

Week 3: Python functions and classes — Homework 2 (Apr 3rd – Apr 9th)

- April 4th: Functions and lambdas
- April 6th: Classes and methods

Week 4: Python functions and classes — Homework 3 (Apr 10th – Apr 16th)

- April 11th: More functions and classes
- April 13th: More functions and classes: creating a game

Week 5: The Pandas DataFrame — Homework 4 (Apr 17th – Apr 23rd)

- April 18th: Pandas I
- April 20th: Pandas II

Week 6: More Pandas — Homework 5 (Apr 24th – Apr 30th)

- April 25th: Pandas III
- April 27th: Pandas IV

Week 7: Data visualization — Homework 6 (May 1st – May 7th)

- May 2nd: Matplotlib I
- May 4th: Matplotlib II

Week 8: Web scraping — Homework 7 (May 8th – May 14th)

- May 9th: Using data APIs, introduction to HTML
- May 11th: Requests and BeautifulSoup

Week 9: Advanced Topics — No homework

- May 16th: NumPy and Statsmodels
- May 18th: Data transformations and missing values