



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](mailto:levyjeff@uchicago.edu)  
Keller 3101

Head TA: [Nadir Shahzad Khan](#)  
TA office hours: See Canvas  
Professor office hours: [Calendly](#)

### Spring Quarter 2024

March 18<sup>th</sup> – May 24<sup>th</sup>

**Section 3:** T, Th 9:30 AM – 10:50 PM Keller 1002

**Section 4** T, Th 11:00 AM – 12:20 PM Keller 1002

Sections 1 and 2 are taught by professor Navin Kumar

### Course Description

In this course, aspiring researchers and data analysts will study rigorous data and programming using Python. As one of the [most utilized](#) (3<sup>rd</sup>) and [most desired](#) (1<sup>st</sup>) 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 work with 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 work through practice assignments in groups.
- Weekly optional **office hours** for the professor and TAs.
- An optional **discussion board** for questions and discussions.

## 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.
- Build a foundation that will improve your ability to quickly look up and properly utilize programming information from online sources.
- Develop skills that apply directly to summer internships working with data, particularly the ability to skillfully handle imprecise, broad, or open-ended data tasks.

## Assessment and Grading

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

**Daily quizzes (10%)** – Each class day will have a brief (5 minute, 2-3 question) quiz on Canvas that will cover a core skill or concept from lecture, due by 8pm.

**Lab work (10%)** – There will be one grade per week (9 total) for deliverables from projects given during scheduled lab time. This work will generally be done in groups, and will receive full marks for completion.

**Take-home assignments (60%)** – There will be four 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.

**Final exam (20%)** – This class will have a written final exam during finals week covering all material from the quarter.

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 labs and quizzes must be completed on time regardless of attendance. Requests for excused labs and quizzes must be sent to the professor and head TA.

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. Assignments turned in any other way cannot be accepted.

**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, but all regrade requests may result in a full regrade and potentially a higher or lower score at the grader's discretion. 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 the 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:

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](#), [StackOverflow](#), and LLMs, which will be discussed in class. The text [Python for Data Analysis 3<sup>rd</sup> 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 (or similar)
- The [GitHub Desktop](#) application

## Course Outline

### Week 1: Mar 19, Mar 21

- Introduction, software review
- Setup, GitHub basics

**Homework 1:** Given  
Monday Mar 25, due by  
11:59pm Sunday Apr 7

### Week 2: Mar 26, Mar 28

- Data types, logic control and loops

### Week 3: Apr 2, Apr 4

- Functions, classes and methods

**Homework 2:** Given  
Monday Apr 8, due by  
11:59pm Sunday Apr 21

### Week 4: Apr 9, Apr 11

- Pandas basics

### Week 5: Apr 16, Apr 18

- Pandas merging, reshaping, groupby

**Homework 3:** Given  
Monday Apr 22, due by  
11:59pm Sunday May 5

### Week 6: Apr 23, Apr 25

- Figures with Matplotlib

### Week 7: Apr 30, May 2

- Applied data analysis

**Homework 4:** Given  
Monday May 6, due by  
11:59pm Sunday May 12

### Week 8: May 7, May 9

- APIs and web scraping

### Week 9: May 14, May 16

- Other advanced topics

### Final Exam: Date TBD