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
Office hours: Calendly

Head TA: Brian Curran
bcurran@uchicago.edu

Dr. Christopher Clapp
cclapp@uchicago.edu
Keller 3039
Office hours:

Spring Quarter 2025
March 24th – May 30th

Section 1	M, W	1:30 PM – 2:50 PM	Levy	Keller 1002
Section 2	M, W	3:00 PM – 4:20 PM	Levy	Keller 1002
Section 3	T, Th	9:30 AM – 10:50 AM	Clapp	Keller 1002
Section 4	T, Th	11:00 AM – 12:20 PM	Clapp	Keller 1002
Section 5	T, Th	2:00 PM – 3:20 PM	Levy	Keller 0023
Section 6	T, Th	3:30 PM – 4:50 PM	Levy	Keller 0023

Course Description

In this course, aspiring researchers and data analysts 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 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.

Should I Take This Class?

We believe that the only way to learn to use good coding skills for data analysis is to get lots of practice writing actual code. You should take this class if you are invested in quantitative public policy and want to be far more prepared than the average graduate for professional roles in quantitative work. This is an intensive class that will require a heavier weekly workload than many other 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 Sunday and Tuesday. We aim to keep the total length to around 30 minutes per class.
- The scheduled lecture times will be used as **live labs**, in which we work through skills problem sets that set students up to work on the applied problem sets.
- Weekly optional **office hours** for the professors 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".
- Practice exploring and analyzing real-world data using Python.

Non-technical goals:

- Practice good programming and data principles that are relevant to working in other languages, such as R, Stata, or SAS.
- 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.

Assessment and Grading

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

Attendance (5%) – Per Harris policy, attendance to scheduled class time is mandatory and graded. Attendance starts in week two, and everyone gets two automatically excused absences. These are intended to cover ordinary things, like minor illness. Additional excused absences will require approval from the dean of student's office to be considered.

Skills Problem Sets (25%) – Most classes will have a problem set focusing on programming skills. These assignments are collaborative, will be worked on in class, and will directly set students up for success in applied problem sets.

Applied Problem Sets (50%) – There will be seven larger, individual problem sets focusing on applying class material to understand and solve policy-relevant questions. The lowest score will be automatically dropped.

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

This class is not curved. All grades will use the following intervals:

A	[95% - 100%]	B+	[85% - 90%)	B-	[60% - 80%)
A-	[90% - 95%)	B	[80% - 85%)	F	[0% - 60%)

Class Policies

Assignments **must be turned in** using 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. See the Gradescope Guidelines document on Canvas for details.

Every student has **four 12-hour late tokens** available to them during the quarter. Those extensions will be automatically applied to any late problem sets 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. No assignments will be accepted beyond three days from the due date.

Any data resulting from in-class or out-of-class work may be used for research purposes. All such use will always be anonymous.

See the general **grading rubric** on the Canvas course page.

Academic Integrity

We take the Harris Academic Honesty and Plagiarism Policies seriously. Students suspected of academic dishonesty will be reported to the Harris Dean of Students for investigation and adjudication. The disciplinary process can result in sanctions up to and including suspension or expulsion from the University.

The overarching principle is that all code you turn in must be your own.

In **applied problem sets**, you may not share or look at each other's code. For **skills problem sets**, you may look at each other's code and collaborate directly, though you must still turn in only your own work. You may not use solutions from students who previously took the class.

If you violate the integrity policy you will receive a failing grade in the class.

How you can collaborate and get help

- With classmates or on Ed Discussion
  - Share output (e.g. plots or error messages)
  - Discuss concepts, pseudo-code, and theory (e.g. using a whiteboard)
  - On Ed, you may post a reproducible example of a bug
- Search for help online (e.g. StackOverflow, ChatGPT)
  - **You may not copy verbatim -** find inspiration and then write your own version

There may be some assignments where you will be notified you can work in groups. Groups will be assigned or need to be declared when the assignment is given. In this case, you may collaborate freely, share code, and submit only one assignment. The attribution rules still apply.

Attribution for help

- Work with your classmates
  - **Cite the individuals you collaborate directly with** by including their names in the comments at the top of your assignment.
- Online sources
  - **Cite all code you use with a URL**, even a one-line snippet.
  - **For AI tools** provide the initial query string you used and an explanation of how you used the AI tool's response in writing your own response.

If you are unsure of whether a specific action is consistent with this policy, ask. By taking this class you are agreeing to these policies.

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

While not required, we will be primarily following Python for Data Analysis 3rd Edition by Wes McKinney in this class, with some material also drawn from R for Data Science 2nd Edition by Hadley Wickham, Mine Cetinkaya-Rundel & Garrett Grolemund, and Introduction to Python for Geographic Data Analysis by Henrikki Tenkanen, Vuokko Heiknheimo & David Whipp. All three are available online for free, and the first two can be ordered in print if desired.

All software used is open-source and freely available online regardless of what type of computer you use. Details and instructions will be provided in the first week of class.

Course Outline

Date*	Day*	Topic	Reading§	Applied Due‡
Week 1: Introduction				
Mar 24, Mar 25	M, Tu	Intro, syllabus		
Mar 26, Mar 27	W, Th	Python setup		
Week 2: Python Basics				
Mar 31, Apr 1	M, Tu	Syntax, dtypes, conditionals	WM2, 3	
Apr 2, Apr 3	W, Th	For loops, comprehensions	WM3	
Week 3: Functions and Classes 1				
Apr 7, Apr 8	M, Tu	Functions, PEP8, comments	WM3	Pset 1A Sun Apr 13th
Apr 9, Apr 10	W, Th	Classes		
Week 4: Functions and Classes 2				
Apr 14, Apr 15	M, Tu	Inheritance, lambdas, unpacking	WM3	Pset 1B Sun Apr 20th
Apr 16, Apr 17	W, Th	Error handling, simulations	WM3	
Week 5: Data Analysis 1				
Apr 21, Apr 22	M, Tu	Pandas, indexing, dates	WM5, 6, 11	Pset 2A Sun Apr 27th
Apr 23, Apr 24	W, Th	Visualizations with Matplotlib	WM9	
Week 6: Data Analysis 2				
Apr 28, Apr 29	M, Tu	“Tidy” data, pivot, melt	HW5, WM8	Pset 2B Sun May 4th
Apr 30, May 1	W, Th	More reshaping	WM8	
Week 7: Data Analysis 3				
May 5, May 6	M, Tu	Merging, concatenating	WM8	Pset 3A Sun May 11th
May 7, May 8	W, Th	Groupby	WM10	
Week 8: Data Analysis 4				
May 12, May 13	M, Tu	String methods + regex	WM7	Pset 3B Sun May 18th
May 14, May 15	W, Th	NaNs, transformations, models	WM7	
Week 9: Spatial Data				
May 19, May 20	M, Tu	Shapefiles + geopandas	THW5, 6	Pset 4 Sun May 25th
May 21, May 22	W, Th	Choropleths	THW 8	

Final Exam: Date TBD

Section 1	Keller TBD
Section 2	Keller TBD
Section 3	Keller TBD
Section 4	Keller TBD
Section 5	Keller TBD
Section 6	Keller TBD

** Class for sections 1 and 5 are on the first date/day, class for sections 2, 3, 4, and 6 are on the second.*

§ WM = Python for Data Analysis
HW = R for Data Science
THW = Intro to Python for Geographic Datal Analysis

‡ Homework is always due at 11:59pm on the day listed