



THE UNIVERSITY OF CHICAGO
HARRIS SCHOOL
OF PUBLIC POLICY

PPHA 30537:
**Data and Programming for Public
Policy**

Dr. Jeffrey A. Levy
levyjeff@uchicago.edu
Keller 3101
Office hours: [Calendly](#)

Head TA: TBD
tbd@uchicago.edu

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

Spring Quarter 2026

March 23rd – May 29th

Section 1	M, W	1:30 PM – 2:50 PM	Clapp	Keller 1002
Section 2	M, W	3:00 PM – 4:20 PM	Clapp	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
Lab	F	9:00 AM – 10:20 AM	Keller 0001, 0021, 0023, 2112	

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 **labs**, in which we work through skills problem sets that set students up to work on the applied problem sets.
- Friday **lab** and **live coding** with TAs to get guidance on 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 (10%) – Per Harris [policy](#), regular in-person class attendance is required of all Harris students. It is also necessary (but not sufficient) to do well in the class. We will take attendance in each class beyond the first week, and only on-time, in-person attendance for the full class period counts towards your attendance grade. Please be sure to display your name tent in every class to facilitate the taking of attendance. Every absence beyond the first two will reduce your course grade by one percentage point, up to a maximum of

ten percentage points.¹ Per the Harris policy, we will notify the Dean of Students about students who are chronically absent.

Skills Problem Sets (10%) – 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. They are graded on completion.

Applied Problem Sets (40%) – 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.

Quizzes (40%) – There will be three written quizzes given in Friday labs during the quarter. These quizzes will assess your understanding of the applied problem sets in a closed-book setting. The lowest score will be automatically dropped.

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

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

Students who wish to take the course [Pass/Fail \(P/F\)](#) must use the Harris [request form](#) and meet the appropriate deadline. To earn a Pass grade, students taking the course P/F must: turn in **six of the seven applied assignments**, take at least **two of the three quizzes**, and **earn at least a B-** on the above scale. Note that P/F grades or those below a B- do not meet the requirements for the [Specialization in Data Analytics](#).

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 use on applied problem sets.

- Tokens require no excuse to be given
- Extension tokens 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.
- No assignments will be accepted beyond three days from the due date.

¹ Please note that you do not need to notify us of short-term absences, and we cannot adjust the attendance policy on a case-by-case basis by excusing absences. We understand that students sometimes have legitimate reasons for being unable to attend class (e.g., illness), so you can miss class twice before it affects your grade. Only long-term issues of sufficient magnitude that warrant involving the Academic and Student Affairs team in the discussion can qualify for an exception to this policy.

- Tokens may not be used on the skills problem sets, as these are worked on in labs and are graded on completion.
- Regardless of late token use, you will still be responsible for the content of the assignment on the three Friday quizzes (see course schedule).

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

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 may 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
 - In class, Friday labs, or office hours
- Search for help online (e.g. StackOverflow, ChatGPT)
 - **You may not copy verbatim** - find inspiration and then write your own version

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

* Class for sections 1 and 2 are on the first date/day in a row, class for sections 3-6 are on the second.

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

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