

CCJS 418E: Coding for Criminology (Fall 2026)

Location: [SQH 1117](#)

Time: Tuesday (Online) / Thursdays (SQH) 3.30 to 4.45 p.m.

Instructor: Zubin Jelveh (zjelveh@umd.edu)

Office: 2220F LeFrak

Office Hours: Fridays 1 to 2pm, by appointment through [ELMS](#)

Course Description

This course teaches criminology students to answer questions about crime and criminal justice using data. Criminal justice increasingly relies on data for major decisions—from setting bail to deploying patrols—and the questions that matter are usually easy to ask and hard to answer. Did arrests actually fall after this policy changed? Are some neighborhoods policed differently than others? By the end of the semester you'll be able to take a question like that, work through it with real data, and arrive at an answer you can explain and defend.

We begin with computational thinking, which is essentially a structured way to approach complex problems. A large part of it is decomposition: breaking a question into steps small enough that a computer can carry them out. "Did crime go up?" can't be answered until you've decided what counts as crime, over what stretch of time, and measured how. "High-crime neighborhood" can't be either, until you've decided what counts as high.

Being precise isn't the same as being right. Once a question is exact, you still have to decide what to compare against—last year, the rest of the city, or the same months before a policy took effect. Different comparisons can point in different directions, and this is a judgment the code will not make for you. We start on it in the first weeks of the semester and keep coming back to it.

Once the steps are clear, you need something to carry them out on more data than you could work through by hand, and something that lets you redo the whole thing when the question changes. That's where Python comes in. Python has become the standard language for data analysis in many fields. You'll learn the fundamentals through real examples from criminal justice—instead of printing "Hello World," you'll be calculating crime rates. Instead of abstract math problems, you'll analyze actual arrest data. Most of the course focuses on pandas, Python's data manipulation library, covering data cleaning, filtering, aggregation, and data visualization. Each new piece of pandas arrives when we need it to answer something we couldn't answer the week before.

We embrace AI tools like ChatGPT and Claude from day one because they're becoming standard in programming work. They can help at every stage—from sharpening a question and suggesting comparisons to writing and explaining code. But a fluent answer is not evidence that the analysis is sound. You remain responsible for deciding what you want to know, what should be compared, what the data actually measures, and what the result allows you to say. The assessments in this class are designed to develop that judgment.

You'll complete three problem sets throughout the semester plus a final project and presentation. You're encouraged to use AI tools to help complete these assignments. However, grading happens through one-on-one code reviews where you'll walk me through your solution and explain what your code does. During these conversations, you should be prepared to answer questions about how your code works, what would happen if we changed certain parts, and interpret your results. These collaborative discussions focus on making sure you understand the tools you are using.

Course Objectives

After completing this course, students will be able to:

1. Turn a broad question about crime or criminal justice into one precise enough to answer with data
2. Write and modify Python code to load, clean, and analyze criminal justice datasets, using pandas to filter, group, aggregate, and merge
3. Create meaningful visualizations that communicate findings from criminological data
4. Work effectively with AI tools while understanding what the generated code actually does
5. Explain their code and analytical choices in plain English during code review discussions, and adapt that code when the question changes

Prerequisites

CCJS 100, CCJS 105, CCJS 200, and CCJS 300. Permission of the department is also required.

No prior Python experience is needed. CCJS 300 introduces R, and that experience helps, but this course starts Python from the beginning and assumes no coding background beyond it.

CCJS 300 is the natural companion to this course. It covers research you design and data you collect — surveys, sampling, interviews, field observation. This course works with data that already exists: police records, 311 complaints, court data. You do not control how any of it was recorded, which changes what you can ask of it and what you can conclude.

Technical Requirements

Students will need:

- A laptop
- Access to an AI assistant (free and university-provided options will be discussed in class)
- No software purchases required - we'll use free, open-source tools

AI Tools and Resources

UMD provides and supports several AI tools for students. See [AI Resources at UMD: Available Services](#) for current options, access information, and data privacy guidance.

Assessment

► **Problem Sets (60% total - 20% each)**

► **Project Proposal (10%)**

► **Final Project (30%)**

Weekly Schedule

Links to Lecture Videos (coming soon)

Week 1: Questions, Comparisons, and Code (9/3)

- Slides: [Introduction](#)

Resources:

- [BBC Bitesize: Computational Thinking](#)
 - [Video: Making Cocount Stew](#)
 - [Canaries in the Coal Mine? Six Facts about the Recent Employment Effects of Artificial Intelligence](#)
-

Week 2: Getting Started - Python and Colab (9/8, 9/10)

- Notebook: [Intro to Colab and Python](#)
- Slides: [Intro to Jupyter and Python](#)

Required Reading: Nick Huntington-Klein, *The Effect: An Introduction to Research Design and Causality*

- [Chapter 1: Designing Research](#)
- [Chapter 2: Research Questions](#)

Resources:

- [Resources for Learning Python](#)
-

Week 3: Doing It for Every Case - Loops and Functions (9/15, 9/17)

- Notebook: [Loops](#)
- Notebook: [Functions](#)

Resources:

- 30-Days-of-Python: [Loops](#)
 - 30-Days-of-Python: [Functions](#)
 - GeeksforGeeks: [Functions](#)
-

Week 4: From Question to Steps (9/22, 9/24)

- Final Project: [Example Datasets and Stories](#)
- Final Project: [Crime Data Analysis Ideas](#)
- Final Project: [Proposal Guidelines](#)

Problem Set 1 Due: 9/27 11:59am (covers Weeks 2-4)

Code Review: 9/29 (Tuesday session)

Week 5: Meeting Your Data (9/29, 10/1)

- Slides: [Intro to Pandas](#)

- Notebook: [Intro to Pandas](#)

Resources:

- [Resources for Learning Pandas](#)
-

Week 6: Building the Two Groups - Filtering (10/6, 10/8)

- Notebook: [Filtering](#)

Resources:

- Kaggle: [Filtering and Sorting](#)
 - Real Python: [Pandas DataFrames](#)
-

Week 7: Comparing Many Groups at Once - Groupby (NO CLASS 10/13 - Fall Break, 10/15)

- Notebook: [Groupby](#)

Resources:

- Kaggle: [Grouping and Sorting](#)
 - Real Python: [Pandas GroupBy](#)
-

Week 8: Showing a Comparison (10/20, 10/22)

- Notebook: [Visualization](#)

Resources:

- Seaborn Tutorial: [Introduction to Seaborn](#)
 - Real Python: [Python Plotting with Seaborn](#)
 - Storytelling with Data: [Chart Chooser](#)
 - Kaggle: [Data Visualization](#)
-

Week 9: Compared to What? Choosing a Baseline (10/27, 10/29)

Problem Set 2 Due: 10/25 11:59am

Code Review: 10/27

Week 10: Comparing to Yourself - Dates and Time (11/3, 11/5)

- Notebook: [Dates and Time](#)

Project Proposal Due: 11/1 11:59am ([Proposal Guidelines](#))

Week 11: Bringing Two Tables Together - Merging (11/10, 11/12)

- Notebook: [Merging](#)
-

Week 12: From Comparison to Claim (11/17, 11/19)

- PowerPoint: [Yankees Noise Analysis Example](#)
- Notebook: [Complete Analysis Demo](#)

Problem Set 3 Due: 11/22 11:59am

Code Review: 11/24

Week 13: Code Review (11/24 - NO CLASS 11/26, Thanksgiving Recess)

Week 14: When the Comparison Breaks (12/1, 12/3)

- Slides: [Anatomy of a Comparison](#)
 - Slides: [What If? From Code to Causality to LLMs](#)
 - Notebook: [Causal Inference](#)
-

Week 15: FINAL PRESENTATIONS (12/8, 12/10)

Final Project Presentations Due: 12/7 11:59pm

Final Project Due: 12/16 11:59pm

Course Policies

Contacting the Instructor

Students should contact me via email at zjelveh@umd.edu. Please begin your subject line with **"CCJS418E"** or **"Coding for Criminology"** to ensure your email receives prompt attention. I typically respond to emails within 24-48 hours during weekdays.

Course-Related Policies

Policies relevant to Undergraduate Courses are found [here](#). Topics that are addressed in these various policies include academic integrity, student and instructor conduct, accessibility and accommodations, attendance and excused absences, grades and appeals, copyright and intellectual property.

Academic Integrity

The University of Maryland, College Park has a nationally recognized Code of Academic Integrity, administered by the Student Honor Council. This Code sets standards for academic integrity at Maryland for all undergraduate and graduate students. As a student you are responsible for upholding these standards for this course. It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit <http://www.studentconduct.umd.edu>.

Accessibility and Disability Support

The University of Maryland is committed to creating and maintaining a welcoming and inclusive educational, working, and living environment for people of all abilities. The University of Maryland is also committed to the principle that no qualified individual with a disability shall, on the basis of disability, be excluded from participation in or be denied the benefits of the services, programs, or activities of the University, or be subjected to discrimination.

Students with disabilities who require accommodations for this course should contact the Accessibility and Disability Service (ADS) at 301-314-7682 or adsfrontdesk@umd.edu. Please inform me of any accommodations you need as soon as possible, preferably within the first two weeks of the semester.

More information about ADS and accommodations can be found at <https://www.counseling.umd.edu/ads/>

