Syllabus: Data and Programming for Public Policy I - R

Harris School of Public Policy
University of Chicago
Course number

Nate Barker
barkern@uchicago.edu

Spring 2024: 30535

Information about this course

Background and Goals

This course offers an introduction to programming and data analysis for students with no prior coding experience.[1] This course is targeted towards MPP students; undergraduates are welcome to enroll as well.

The course will be taught in R. We will learn how to write and share code, and how to clean, wrangle, and visualize data. There are two main objectives associated with the class:

- For those of you who go on to work in policy or policy-adjacent jobs, we want you to feel comfortable using programming (in general) and R (specifically) as a way of answering policy questions.
- For those of you who discover after taking this course that you enjoy programming (or at the very least, find it valuable and tolerable), we want to provide a strong coding foundation for subsequent courses in programming and analysis at Harris, and in your future careers.

Prerequisites

Harris Statistics for Data Analysis I & II. If you are a non-Harris student and therefore have not taken these two courses and would like to enroll in this course, you may petition to join by sending an email to the course address with what languages you know and examples of code you have written.

Relationship to other programs

This course is the first of a three-quarter sequence in the Harris Data Analytics certificate.

How this class will work

[1] Or experience with a proprietary platform like Stata, SASS, SPSS, etc.

This class will consist of lectures twice per week and (optional) lab sessions.

The lectures will be held:

Section 1: Tuesday and Thursday, 9:30am-10:50am, Keller 0021
Section 2: Tuesday and Thursday, 11:00am-12:20pm, Keller 0021

The labs will be held:

Labs (to be assigned once class starts):
Lab Section A: Thursday, 3:30pm-4:50pm, Keller 0001
Lab Section B: Friday, 10:30am-11:50pm

In lectures, we will see new content related to coding and programming, answer questions from previous classes and problem sets, and will at times break into smaller groups to work on problems and ideas seen in class.

Lab will occasionally involve 10-15 minutes at the start reviewing specific material, but will otherwise function as de facto office hours, where TAs will work to answer any questions you may have from the material covered in class.

We will also use the Ed board as a way for you to ask questions (anonymously or publicly) about any questions from class, Problem Sets, etc.

Student assignments and Grading

Grading will be based on the following components:

Problem Sets: We will have nine problem sets, one per week. They will comprise 80% of your grade. Your lowest score will be dropped. These will be submitted via Canvas.

Problem sets will typically be due on Monday afternoon, at 5pm. The provisional due date timeline will be:

PS1: March 25
PS2: April 1
PS3: April 8
PS4: April 15
PS5: April 22
PS6: April 29
PS7: May 6
PS8: May 13
PS9: May 20

Participation: Participation in class will comprise 5% of your final grade, which is earned by (a) attending class regularly, and (b) answering and asking questions.

Quizzes: In-class quizzes, lasting about 10-15 minutes) will comprise 15% of your final grade. I will (provisionally) have five of them (and no more than five); the date of the quizzes will be announced at a minimum of one week in advance. If you are not in attendance for the date of a quiz, we will use the average from those you do attend (the quizzes will enter the final grade in a curved manner).

Extra credit: We will additionally award up to 5% extra credit for answering other students' questions on Ed.

Late coins: You will begin the quarter with six late coins. A late coin enables you to turn a problem set in one day late. The maximum number of late coins for one assignment is two. You are only allowed to submit integers of coins. If you want to submit your assignment 47 minutes late, the cost is not 0.0327 late coins—it is one coin!

Passing: grade of 60% or more

Curve: Among students who pass, the curve is one-third A, one-fourth A minus, one-fourth B plus, one-twelfth B and one-twelfth lower grades.

Textbook and topics

Textbook: We will closely follow Hadley Wickham and Garret Grolemund's R for Data Science, Second Edition (R4DS 2e). The online textbook is free.[2]

The topics we will cover in this class will (provisionally) include:

- Intro to R and Course Expectations
- Data Visualization
- Data Transformation
- Exploratory Data Analysis
- Wrangling Data
- Tidying Data
- Joins
- Coding and ChatGPT
- Factor Data
- Working with Dates and Times
- Pipes and Functions
- Iteration

[2] You are welcome to buy the book if you would like. I find the online version substantially easier to use though. For example, it has sample code you can paste directly.

- Intro to Maps

Assistance and Teaching Assistants

Provide plans for office hours, contact information (email, phone, Slack, Piazza, etc.), and preferred contact methods and protocols.

There are three ways to reach out with questions:

1) Talk to me at the end of class, or the TAs during the lab sessions
2) Use the Ed message board to ask a question

*Note that for the Ed board, the last day to ask questions related to the problem set is the day before the problem set is due**. That is, the TAs will not respond to any requests the day the problem set is due!***

3) Email me. **When emailing, please begin your email with the subject line "DAP:**". For example, "DAP: Clarification about Syllabus."

Teaching assistants will lead lab sessions, and will also be available on the Ed Board to answer questions.

Teaching and learning

My expectation is that you are coming to class every lecture (except when unable to, e.g., due to illness or extenuating circumstances). As mentioned above, attendance and participation will constitute 5% of your total grade for the class.

Is this class right for me?

I recognize that some of you may have only touched R briefly in the context of other classes, others may have more substantial experience. I don't have the ability to assess whether this class is right for each individual person. A few thoughts though:

1. Relative to some other classes (e.g., Statistics for Data Analysis I&II, Program Evaluation), this class is much more focused on the nuts and bolts of coding, and developing a foundation, rather than specifically coding in service of analysis. From seeing problem sets for these analysis-forward courses, my assessment has been that this course is complementary to what is taught in other courses.
2. That being said, if you have a very strong foundation in R, this is probably not right for you—it does not assume a prior background.
3. One way to evaluate the fit might be to look at a few chapters of the aforementioned R4DS 2e book, and to gauge whether the ideas and exercises embedded in them are commands/operations you already know. Some examples of chapters to examine (roughly corresponding to 25%, 60% and 85% of the way through the quarter) include:

- Groups in Data Transformation
- Joins

- Functions

If ideas in these slides are largely familiar concepts, this class might not be right for you; if much of it looks novel, then this class is more likely to be valuable (though ultimately, it is up to you!).

Integrity

All submitted code should be your own

What you CAN'T do:
• Share your code
• Ask for someone else's code
• Copy and paste online solutions for book exercises
• Copy and paste (or read and rewrite) code from online sources (stackoverflow, github, medium, etc)

What you CAN do:
• Via Zoom/Screen sharing or in person:
– Clarify questions
– Discuss conceptual aspects (pseudo code)
– Show output and error messages
– Always list all collaborators on top of your pset
• Canvas:
– Ask questions
– Share error messages
– Share generic or pseudo code
• Online resources: always cite the source by leaving the link commented on your code

ChatGPT:

ChatGPT and Large Language Models (LLMs) are currently capable of quickly producing useful code. My assumption is this will only grow over time. However, LLMs also have the potential to "hallucinate," and to confidently give wrong answers to questions. Thus, in order to be a successful programmer, it is necessary to have a strong foundation, both to be able to code directly, and to be able to appraise and test whether any code obtained from LLMs is accurate. There are things to note in this domain:

1. **Learning how to use LLMs well**: I do want you to get a sense of when and where LLMs can be useful, and where they are (currently) less likely to be useful. We will have a problem set midway through the term, called "ChatGPT and Me," where we will try different problems in ChatGPT (or your LLM) of choice, see how well it performs at various tasks, and see how to most precisely define our requests to yield useful information.

2. **What I will NOT ALLOW as an instructor:** You are not allowed to simply copy and paste questions into an LLM, or to take a question and paraphrase slightly before submitting.
3. **What I recommend as an instructor:** Ultimately, my goal as an instructor of this course to help you become successful coders and analysts! My advice therefore is that whenever you are stuck conceptually, to try to work through the slides, the online textbook, and your notes before turning to an LLM. Often the subjective experience of struggling is when learning happens! You will undoubtedly be a stronger coder if you work through things first, rather than run to ChatGPT every time you have a question. (In much the same way that the best way to learn a language is not to enter every sentence you see into Google Translate whenever you are confused about its meaning.)

General Resources Available to Students

- Harris Academic Support Programs and Handbook
- Student Wellness
- University Learning Resources

Harris School and University of Chicago Policies

.

- Harris School Policies
- University General Policies
- University Academic Polices