Intro to PPHA 30538

Peter Ganong and Maggie Shi

September 30, 2024

Table of contents I

Course overview

AI tools and integrity policy

Course mechanics

Course overview

Following along

The native lecture slides and the PDF are posted on the course's student repo

Search "student repo" on the canvas home page.

Introduction: Who A mI?

- Maggie Shi
- Assistant Professor at Harris since 2023
- PhD in Economics from Columbia, 2022
- Health economist thinking about how to reduce waste in healthcare
- Avid foodie and proud dog mom of Truffles

Roadmap: what you will learn

- Data extraction, visualization and communication, plus some tech skills
  1. Curriculum
  2. Apply this knowledge to the public sector
- Course quirks
- Is this course for me?

Step 1: Curriculum

- Visualization Curriculum
- Introduction to Python for Geographic Data Analysis
- Version control
- Dashboards
- Webscraping
- Natural language processing

Step 2: public sector: datasets

- Real-world data are both messy and informative
  - Assess data quality
  - Make thoughtful data cleaning choices

Data Type	**Description**
Hospital closures	CMS
Healthcare fraud	HHS OIG
Parking tickets	Chicago DOT
Alerts and jams	Waze

Course Quirks

- There are a number of things that make this class different from other classes that you have taken at Harris
- Everything good—and everything frustrating—about this course stems from these quirks
- We will tell you about the four quirks now, so you can decide if this course is for you.

Course Quirks

1. teaching method – intensive, hands-on support, active participation
2. open source curriculum
3. self sufficiency
4. contemporary & realistic over perfect & clean

quirk 1: teaching method – intensive, hands-on support

How we try to support you

1. Afternoon labs. 10 min quiz, 10 min mini lesson (most weeks), 60 min pair programming or group office hours (time to give and get help).
2. Awesome teaching assistants
3. Practice, practice, practice

Teaching Team

- Ozzy Houck (Head TA)
- 3rd Year PhD student

- Fatima Irfan
- Second-year MSCAPP student

Teaching Team

- Andre Oviedo
- Second-year MPP student

- Akbar Saputra
- Second-year MSCAPP student

Teaching Team

- Dema Therese
- Second-year MSCAPP student

- Vitor Silveria
- MPP alum

Teaching Team

- Joaquin Pinto
- Second-year MSCAPP student

quirk 1 – active participation required I

- We're looking forward to getting to know you!
- We ask that you please bring your name tent in to each class
- Active, broad-based participation is key to learning in this class.
  - In prior years, every student has asked or answered at least one question in lecture. We are striving for that goal again this year.
- Questions need not be sophisticated. We welcome "repeat that please" or "I don't understand that yet, can you try again?"

quirk 1 – active participation required II

- Attendance
  - sign-in sheet at lecture
  - Canvas quiz at lab
  - TA audits of both lecture and lab
- Up to 2 absences with no questions asked
- Attending via Zoom/recording request counts as an absence
- For longer-term absences, we can work out a plan with your academic advisor

quirk 1 – active participation required III

- **M**ost **U**nclear **D**iscussion cards
  - At the end of each class, please fill out a MUD card with the material in class that was most unclear to you or any other item you were wondering about
  - Name is optional
  - When a topic comes up on multiple MUD cards, we will do a mini lesson about it at lab and/or I will discuss it at the start of the next class
  - This feedback loop requires ~48 hours to execute, so much better when possible to just say that you have questions in real-time!

quirk 2: open source

- We are teaching entirely based on open-source, free material.
- You could potentially teach all of this to yourself on your own.
- Because we are using general-purpose open source materials, the examples from the textbook will not involve public policy. But the examples on the problem sets will.

quirk 3: self-sufficiency

DEAR VARIOUS PARENTS, GRANDPARENTS, CO-WORKERS, AND OTHER "NOT COMPUTER PEOPLE."

WE DON'T MAGICALLY KNOW HOW TO DO EVERYTHING IN EVERY PROGRAM. WHEN WE HELP YOU, WE'RE USUALLY JUST DOING THIS:

PLEASE PRINT THIS FLOWCHART OUT AND TAPE IT NEAR YOUR SCREEN. CONGRATULATIONS; YOU'RE NOW THE LOCAL COMPUTER EXPERT!

quirk 3: self-sufficiency

Lecture will **not** cover everything you need to know for problem sets! This is a design choice for the course rather than an accident.

Online

- Google: "my problem + Python"
  - functions
  - error messages
  - StackOverflow
- AI tools

At Harris (continues on next slide)

quirk 3: developing self-sufficiency at Harris

- Lecture **and** lab
- Your classmates
- Ed – message board for pset questions
- Harris tutoring

quirk 3: Time commitment: 10-20 hours a week

- Learning a foreign spoken language is incredibly time-consuming.
- Learning a computer language is similarly incredibly time-consuming.
- You will sometimes spend three hours trying to figure something out and then realize that you were doing it all wrong. It will be frustrating.
- Remark: if you are spending more than 20 hours a week, reach out to your instructor or a TA for additional help on being more efficient.

quirk 3: What to do if this class is going too fast?

- Try different ways to learn the same material
  - Read textbook
  - Run code in lecture notes
- Post questions on Ed
- Ask questions of your peers at lab
- Ask questions of TAs
- Set up a time to talk with us (PG: at lab, MS: at office hours)

quirk 4: contemporary & realistic over perfect & clean

- **Code**: Textbooks are very new, sometimes not even finished yet!
- **Datasets**: Each year we update the datasets and the problem sets. Data are messy on problem sets, just as is the case in the real world.
- **Instruction**: This is Peter and Maggie's first time teaching DAP II in Python. We might over- or under-estimate how long it will take to cover some material. And we don't (yet) know where our slides are unclear, so we will really benefit from your questions!

quirk 4: contemporary & realistic over perfect & clean

- In real-life, your manager or your end user doesn't tell you what query to run. Instead, they ask a plain English question which you need to translate into a query. The course problem sets will similarly, in some cases, ask plain English questions. They sometimes will not have a single "correct" answer.
- Our slides and problem sets will have typos. We have processes in place to try to prevent typos in problem sets, but they don't catch everything. If you think there's a typo, there probably is. Just ask.

Course Overview: Summary

Quirks

1. teaching method – intensive, hands-on support, active participation
2. open source curriculum
3. self sufficiency
4. contemporary & realistic over perfect & clean

- This course is for you if you have the time, grit, and interest.
- Any questions?

AI tools and integrity policy

AI tools and integrity policy: roadmap

We will say ChatGPT going forward, but this applies equally to all AI tools

- What ChatGPT can and cannot do
- How to use it in this class
- Integrity policy

Background on ChatGPT

- At its core, it is a text compression algorithm.
- You don't learn an instrument by listening to lots of songs on Spotify.
- You don't learn to program by watching the machine program.
- It's probably a better programmer than you, but it makes dumb mistakes because it doesn't really have an intent or goal. You need both to become a programmer but also to guide it towards real ends you have (once you learn the basics).

Acknowledgement: thanks to AI expert and MIT professor John Horton for helping us develop this slide

Use of ChatGPT I

- Appropriate and beneficial uses
  - Understanding a concept
  - Debugging/cleaning up syntax
  - Explaining error messages
  - Give it your solution and ask if it can make code more elegant or if it has a better solution

Use of ChatGPT II

- Discouraged uses
  - Having it solve the whole problem (*Why? If it gets the answer right, it will often be by accident. And you won't be able to check its work.*)
  - Prose that interprets your plot or statistical output (*Why? Because we, and your supervisors in the future, want to know your interpretation of the data. Not ChatGPT's interpretation of the data.*)
  - Code completion (*Why? (a) these tools are most useful once you have gained the ability to write some code yourself (b) you will need to write some code on the final*)
- Procedure: Attribution and explanation. Why? We will use a similarity checker, and we may use an AI checker as well. If you don't attribute, your work will be considered as plagiarism.

Settings where you may not use ChatGPT

To make sure ChatGPT doesn't serve as a substitute for learning, you *may not* using ChatGPT for these components:

- In-class solo exercises + do-pair-shares
- Weekly quizzes at start of lab
- Final exam (similar in style to quizzes)
- Pair programming at lab

By having *some* settings without ChatGPT, we are striving to create an environment where ChatGPT **enhances** your knowledge rather than **prevents** you from true learning.

In addition, we note that ChatGPT is not always available! When your boss asks you to critically evaluate something, it might not be a good idea to say "wait, I need to ask ChatGPT before I share my own opinion" :-)

Integrity policy (I)

- The overarching principle is that all code you turn in must be your own.
- You may not share or look at each other's code. You may not use solutions from students who previously took the class.
- If you violate the integrity policy, you will receive an F.

Integrity Policy (II)

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 `reprex` of a bug
- Search for help online (e.g. StackOverflow, ChatGPT)
  - You may not copy verbatim - find inspiration and then rewrite it

There may be some assignments where you will 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.

Integrity Policy (III)

Attribution for help

- Work with your classmates
  - Cite the individuals you collaborate with (via google form)
- Online sources
  - Cite in a comment all code you use with a URL, even a one-line snippet.
  - For AI tools provide in a comment the initial query string you used and an explanation of how you used the AI tool's response in writing your own response.

Dealing with ambiguity

If you are unsure of whether a specific action is consistent with this policy, ask.

AI tools and integrity policy

- ChatGPT is very helpful but can be obstacle to true learning
- Memorize the integrity policy. You will be asked about it on the quiz.

Course mechanics

Course mechanics: platforms

- We will be developing our code using **VSCode**
  - Code editor
  - Manages Python environments
  - Terminal/command line
  - Connects to many extensions
- VSCode will replace the Spyder code editor you may have used in DAP 1
  - All tutorials + screenshots will be in VSCode
  - If you're using another software, but something related to the environment is not working, we will first tell them to switch to VSCode before trying other things
- We will use two VSCode extensions: **Quarto** and **Jupyter**

Course mechanics: platforms

- **Quarto**: publishing system for articles, slides, etc.
  - Source files: `.qmd` format
  - Output files: `.html` or `.pdf`, depending on the lecture
  - At the top of each class, you will pull from the student repo, which will download a folder that contain every file you need to compile your `.qmd`
- **Jupyter**: turns VSCode into a "notebook"
  - This allows you to run code in-line
  - Like a Jupyter notebook, but inside of VSCode
- This week's mini-lesson at lab will cover using VSCode and these extensions

Course mechanics: use Ed to ask questions.

1. **Public Questions**
   - Use the problem set-specific categories. Try to avoid repeating already-posted questions.
   - General: logistical details
2. **Private Message**
   - Recording/zoom requests

Email discouraged. Will in most cases get a response: "send a PM in Ed". However, if you have a confidential matter which cannot be shared with the teaching staff, then please email `ganong@uchicago.edu` or `m.shi@uchicago.edu` w "30538 confidential".

Course mechanics: grades

- 6 problem sets (paired and solo) – 40%. We will drop your lowest problem set grade such that it maximizes your grade.
- Final group project – 20%
- Quizzes – 5%. These occur at the start of each lab. We will drop the two lowest quiz grades.
- In-person final exam – 30%
- Attendance – 5%
- Extra credit for collaboration – up to 3%

Course mechanics: collaboration

- We really want this class to be a collaborative environment. At the same time, you have to learn this on your own, and no one else can learn it for you.
- Extra credit for answering peers' content questions on Ed - up to additional 2% on final grade
- Extra credit for being acknowledged by many peers as someone who helped them with a problem set - up to additional 1% on final grade
  - Each problem set will require you to submit a Google Form indicating who you collaborated with

Course mechanics: problem sets

- 6 problem sets
- Combination of paired and solo
- Due at 5pm on Saturdays, solutions posted Sunday night
- Debrief on Wednesday/Thursday of subsequent week
- A late coin extends the deadline by 24 hours. You have four late coins.
  - Max one late coin per assignment because solutions posted Sunday night
  - After 24 hours you will receive a 0

Rhythms during lecture

- You are welcome to just listen or also run the `.qmd` code in real-time.
- Even if you plan to listen, you still need to bring your laptop and pull the `.qmd` lecture slides. This is because we will have in-class exercises during lecture.
- We ask that phones remain in your backpack. If you need to use your phone, please first step out of the classroom.

Rhythms during lecture – in-class IT setup

1. If you already cloned the course repo which Professor Ganong sent out on Sunday night, delete this clone (he was trying to be helpful, but failed…)

2. On the Github website, create a **fork** of the course repo.

   - Make sure that you uncheck the box "copy the `main` branch only" because you will use there are two branches : `main` and `after_lecture`

     - `main` excludes the answers to in-class exercises and do-pair-shares

     - `after_lecture` is the complete notes

   - In response to question of why are you creating the fork, choose "for my own purposes"

3. Go to your new fork and clone that repo to your computer

Detailed instructions in `README.md` file on how to clone, switch branches, and pull