

COGS 108: Data Science in Practice

Winter 2024

Table of Contents

- Overview
- Staff
- Objectives
- Grading
- Lecture
- Discussion Section
- Assignments
- Project
- Schedule
- Other Good Stuff

COURSE OVERVIEW

This hands-on, practical course is intended to get you experience working on data science projects. You may have theoretical knowledge from other courses, but here we are going to implement. Doing something is rarely simple. You will likely attempt to do something, do it wrong, learn from your mistakes, and with a bit of luck and skill, eventually succeed. That's just part of the scientific process, and data science is no exception. This course is all about the *practice* of data science.

In focusing on the practice, there is theory that won't be discussed and mathematical proofs that won't be done. That is by design. In particular:

1. There are entire courses dedicated to each of the topics we'll cover. To have time to do anything, we can't teach all the details in a single course.
2. Experts in each of these domains are out there and excited to teach you the nitty gritty about each topic. Take those opportunities... the theory we are skipping is important and useful!
3. We're promoting data literacy. We believe that everyone who is data literate is at an advantage as they go out into the modern world. Data literacy is not limited to those who are computational gurus or math prodigies. You do not have to be either of those to excel at this course.

In this course, you will try many methods. You'll even be asked to implement a technique that has not been explicitly taught. Again, this is by design. As a data scientist, you'll regularly be asked to step outside of your comfort zone and into something new. Our goal is to get you as comfortable as possible in that space now. We want to provide you with a technical and a data science mindset that will allow you to ask the right questions for the problem at hand and set off alarm bells when something in your dataset or analysis is "off."

COURSE STAFF

Role	Name	Section	Office Hours
Instructor	Shannon Ellis	–	Fri 10AM-12PM
TA	Yueying Dong		TBD
TA	Matthew Feigelis		TBD
TA	Yueyan Tang		TBD
TA	James Michaelov		TBD
TA	Kunal Rustagi		TBD
TA	Samyak Mehta		TBD
IA	Siddhant (Sid) Joshi		–
IA	Devin Chen		–
IA	Shenova Davis		–
IA	Zhicheng Wang		–
IA	Jietao Xie		–
IA	Nate del Rosario		–
IA	Eric Song		–

Course GitHub: <https://github.com/COGS108>

Course Piazza*: <https://piazza.com/class/lqzo7aetmf06m8/> (can be accessed directly from Canvas)

Course Canvas: <https://canvas.ucsd.edu/courses/52091>

Assignment Submission: <https://datahub.ucsd.edu>

Anonymous Course Feedback: <https://goo.gl/forms/2nXnDNbgYuS1OsGF2>

*You will be able to post anonymously on Piazza; however, you will only be anonymous to your classmates. Your Instructor and TAs will be able to see who you are.

COURSE OBJECTIVES

- Formulate a plan for and complete a data science project from start (question) to finish (communication)
- Explain and carry out descriptive, exploratory, inferential, and predictive analyses in Python
- Communicate results concisely and effectively in reports and presentations
- Identify and explain how to approach an unfamiliar data science task

COURSE MATERIALS

- All course materials will be provided on GitHub and datahub
- There is no textbook
- Python (≥ 3.9) and a recent version of Jupyter Notebooks. You may fulfill this by any of the following
 - Datahub (Datahub use is required for assignments; other options are OK for project)
 - A local copy on your own computer (we recommend the Anaconda distribution)

- Cloud services such as Google Colab, Microsoft Azure, etc. These all have a free tier or free hours before billing.
- git and a GitHub login
- iclicker access (*optional*, but used in lecture for extra credit)

GRADING

	% of Grade	# to submit
Pre-course Survey (#finalid)	1%	1
Discussion Labs	16%	8 (1/wk)
Lecture Quizzes	8%	8 out of 9 (1/wk)
Assignments	32%	4
Project Review*	5%	1
Project Proposal*	9%	1
Project Checkpoints*	10%	2
Final Report*	15%	1
Final Video*	3%	1
Team Evaluation Survey	1%	1

* Indicates Group Submission

Final exam: This course has no final exam. You do not have to show up anywhere on the date/time of the final exam. Instead, this course has a final project./

Grades: All grades will be released on Canvas. It is *your responsibility to check that your assignment was submitted, that your grade is accurate, and to get in touch if any are missing and/or you think there is a problem.*

Due Dates: For keeping things simple for you all, all due dates each week will be **Sundays at 11:59 PM**. Please DO NOT wait until Sunday at 11:50 PM each week to start everything that is due.

Extra credit worth up to 2% of the final grade will be awarded for: - Attending at least 75% of lectures (starting lecture 2; excluding guest lectures) (0.5% of grade, see Lecture section below) - Answering the mid- & post-course surveys will give 0.5% extra credit (0.25% each) - Attending guest lecture(s) in-person is 0.5% extra credit (0.25% each)

- Filling out the weekly project progress surveys is 0.5% of grade (0.07% each, see Project section below)

Our grading scale is:

A+	>= 97%
A	< 97% to 94%
A-	< 94% to 90%
B+	< 90 % to 87%
B	< 87 % to 84%
B-	< 84 % to 80%
C+	< 80 % to 77%
C	< 77 % to 74%
C-	< 74 % to 70%
D	< 70 % to 60%
F	< 60 %

Know that a third of the class typically feels overwhelmed at the start of the quarter. That said, the average is quite high in this course (typically A-). So, while we anticipate you all doing well in this course, if you are feeling lost or overwhelmed, that's ok! Should that occur, we recommend: (1) asking questions in class, (2) attending office hours and/or (3) asking for help on Piazza.

LECTURE

This quarter we are running two lecture sections to keep up with demand.

Lecture will be podcast. You'll be able to view both/either lecture.

which	when	where
LE A00	MWF 9:00a-9:50a	Catalyst - CTL 0125
LE B00	MWF 2:00p-2:50p	Peterson Hall - PETER 108

Lecture Attendance

You are free to attend either lecture section at any point. Note that the same material will be covered in each section; however, I'm not a robot, so there could be slight variations between the lectures.

We believe that attending in-person is the best way to learn for most people. Our goal is to make the lecture and discussion section worth your while to attend through interactive content and making it a good place to get questions answered. We also use active learning paradigms (live coding, think-pair-share, etc.) where you learn most through active participation (which requires attendance).

To incentivize lecture attendance, students who attend 75% of lectures (starting lecture 2 and excluding guest lectures) will earn 0.5% extra credit to the final grade. Students will receive credit for attendance if they respond to $\leq 75\%$ of a given day's iclicker questions.

Note that attendance is not required because you know what fits your constraints and needs best. If you can't come attend or this is not the best use of your time, that's OK. Your grade will not be negatively impacted. Also, if you're sick, do stay home.

Lecture Quizzes

At the end of each Friday, a quiz with ~10 questions will be released, covering the material from lecture that week. The quiz will be due that Sunday at 11:59 PM. For example, the first quiz will be released Friday of week 1, covering week 1 material, and will be due two days later at 11:59 PM. Each question is worth ~0.1 points and you will have a single, timed (15 min) attempt to complete. *There are no late extensions on quizzes*, but your lowest quiz score will be dropped.

DISCUSSION SECTION

Discussion sections are a place where you will be able to ask questions, get help on technical topics, work on your group projects, and complete the weekly lab exercises. If you are struggling with course material, come to discussion section and/or office hours!

The first discussion section will start Wed of week 2. There will be no discussion sections run until that date.

Students are encouraged to attend the discussion section to which they've enrolled; however, if you cannot attend that section, please feel free to attend a section that works for you. You do not need to let anyone know that you're attending a different section. That said, if any one section becomes too crowded, we'll revisit this policy. (Historically, it has not been an issue.)

Please note that there are two Monday holidays this quarter. If you are in a Monday section, please note that in weeks 2 and 7, if you want to attend section, you'll have to plan to attend a different than usual section.

Section	Day	Time	Location	Room
A03	M	8:00a-8:50a	PETER	103
A02	M	1:00p-1:50p	PCYNH	120
A01	W	11:00a-11:50a	PCYNH	120
B03	W	12:00p-12:50p	PCYNH	121
B01	Th	5:00p-5:50p	PCYNH	120
B02	Th	6:00p-6:50p	PCYNH	120
A04	F	8:00a-8:50a	PETER	103
B04	F	3:00p-3:50p	WLH	2111
B05	F	5:00p-5:50p	WLH	2207
B06	F	6:00p-6:50p	WLH	2207
A05	F	5:00p-5:50p	DIB	121
A06	F	6:00p-6:50p	DIB	121

Discussion section lab exercises

Each week there will be a short lab exercise to review material from lecture and give hands-on programming experience. You can work on these exercises in your lab sections, and your section leaders will be happy to help guide you towards the correct answers. Or, if you don't need help, you can complete them on your own and never come to lab. If you wish to work collaboratively on lab exercises you may do so, but please realize that this is your time to practice and learn new skills with low stakes. If someone else does it for you, it's hard to practice and learn!

Lab exercises will be due on Sundays 11:59PM and released on the prior Friday (more than one week before they are due). Each lab exercise is worth 2 points. There is no late deadline for labs as answers are posted after labs are due.

ASSIGNMENTS

If lab exercises are low stakes and you can get lots of help, then assignments are high stakes and you can only get minimal help. They will be completed **individually** in Jupyter Notebooks and both released and submitted on datahub.

The practice of data science involves writing code to answer questions and accomplish tasks. Thus, to get practice, your assignments will require you to use Python to do just that. Not everything will be explicitly mapped out step-by-step for you. This is intentional. Figuring things out when it's not entirely clear what to do next is part of the practice here. You'll attempt things that won't work and become comfortable with this. You'll get stuck and work to get unstuck. Not quite knowing exactly what's going on at all times is part of the process. And, to be honest, part of the job of being a data scientist.

That said, the first two assignments will be the simplest assignments and aim to get you up to speed in Python and familiar with **pandas**. If the first two assignments are particularly difficult for you, that's ok. But, it's then up to you to determine if you want to put in the work to make it through the rest of the quarter. **Assignments will take more time and be more difficult starting with the third assignment.**

As assignments become more difficult, we don't want you to get or feel totally lost. If you've thought long and hard, gone down a long rabbit hole using ChatGPT, and can't even get a sense of what the next step may be, take a step away. Take a break. Then, come back and see if you can't solve it with a refreshed mind. If you're still totally stuck, ask on Piazza, talk to a classmate, and/or attend office hours for help.

With regards to asking questions of instructional staff, we're here to help you, but there are way more students than there are instructors. So, help each other. In fact, this is how the best data science gets done. Diverse minds solving a problem invariably improves the solution. Also, teaching something to someone else is the best way to determine if you really know something. So, it's win-win. The person who's stuck gets unstuck and the person who helped is more sure in their knowledge. Help one another! Section and office hours are meant to be collaborative.

But how should you help one another on assignments? You should: ask a question that leads the student to figure it out for themselves, point out the correct principle/theory that applies in this case, provide a link to an explanation, or a chunk of pseudo-code. You should not: provide the full answer or code they can just copy/paste.

Turning In Assignments

Assignments will be submitted individually on datahub. We'll talk about the details for submission in class. Assignments will always be released at least a week before the assignment due date. On weeks with assignment deadlines, they will always be due Sunday at 11:59 PM of the week specified (see Course Schedule below).

Check to ensure that your file shows up under "Submitted assignments" on datahub after you click submit!!! If the file is the incorrect file, corrupted, or otherwise unreadable, we cannot grade it and we will mark your assignment as late.

Late Assignments

Late assignments will be accepted at 75% credit for 72 hours (3 days) after the assignment's due date. Once the late deadline passes, assignments will be graded, feedback will be made available on datahub, and assignments will no longer be able to be submitted for credit.

Assignment Regrades

We will work hard to grade everyone fairly and return assignments quickly. But, we know you also work hard and want you to receive the grade you've earned. Occasionally, grading mistakes do happen, and it's important to us to correct them.

If you think there is a mistake in your grade, request a regrade within 72 hours of your receipt of the grade on Piazza via a *private* note to instructors and use the "regrades" tag. This message should include evidence of why you think your answer was correct (i.e. a specific reference to something said in lecture) and should point to the specific part of the assignment in question.

Note that points will *not* be rewarded if you fail to follow instructions. For example, if the instructions say to name the variable **orange** and you name it **ornage** (misspelled), you will not be rewarded credit upon regrade. This is because (1) following instructions and being detail-oriented is important and (2) there are hundreds of students taking the course this quarter. It would be an unfair burden to place on TAs if we didn't have this policy.

COURSE PROJECT

Your course project will be completed in an assigned group of 4-5 people. The reality of data science is that you will have to work with others on a team you didn't get to pick. You'll need to work together to communicate effectively, manage time, organize your projects, and accomplish a goal. People will have different knowledge and skills sets. It is your job as a group to work together to figure out how to maximize each group member's skills to make sure that your differences are helpful to accomplishing your goal, rather than a hindrance. For example, some of you will find the programming aspects of the class assignments very easy, while others will struggle. Alternatively, some of you may find experimental research and hypothesis testing intuitive, while others find it confusing and frustrating.

Probably the most important way to start a team off on the right foot is to discuss expectations of how you will work together. How will you divide up the work? How often and where will you meet? How will you communicate with each other? What's the maximum amount of time it should take for someone to respond to a message? How will you double check that things that need to happen did happen? If there is a problem meeting a deadline how should the person responsible let others know, and then how should the rest of the team react? There are many possible answers to these questions, and there are many more questions about expectations that you might want to ask yourselves. Your team should decide what's right for you, *and then write down these expectations in a place that the team will often see it*. This will help things go more smoothly.

Project Components

These project components are completed and submitted *as a group* and are described in the Project documentation: <https://github.com/COGS108/Projects/tree/wi24>. This includes: 1) Previous Project Review 2) Project Proposal 3) Project Checkpoint #1: Data, 4) Project Checkpoint #2: EDA, 5) Final Project Report, and 6) Final Project Video.

At the end of the course there is a *required* team evaluation survey...this is your opportunity to provide feedback and let us know if a teammate did not contribute. Each student will submit a survey.

Finally, starting week 3, there will be an *optional* weekly survey to be completed individually describing your project progress. Students will earn a small amount of extra credit for each survey completed such that students who complete all 7 weeks' surveys will earn 0.5% extra credit to their final grade.

Project Grading

Each project group will be graded by the same staff member throughout the quarter. You will know who your grader is, and they'll be very familiar with your project, so they'll be a great resource as you complete your project. It's best to think of them as a manager; you won't go to them first (that's what your group mates are for), but they're there for guidance. Given different groups will have different graders, grading will use a standardized rubric and adjustments will be made if one grader is harsher than another to ensure consistency across graders.

In the vast majority of groups, all individuals will receive the same score for each project component; however, if there is evidence of a group member failing to participate fully, individuals can receive a lower grade relative to their group. The same is true in the other direction. If there is evidence of a group member going above and beyond, their individual score can be increased relative to the group.

COURSE SCHEDULE

Date	Week	Day	Topic	Lab due	Lecture quiz due	Assignment due
Jan-8	1	M	Welcome!			
Jan-10	1	W	Python Review			
Jan-12	1	F	Version Control I			
Jan-14	1	Su			Q1	
Jan-15	2	M	No Class - MLK			
Jan-17	2	W	Version Control II			
Jan-19	2	F	Data & Intuition			
Jan-21	2	Su		D1	Q2	A1, Pre-course survey
Jan-22	3	M	Data Wrangling (pandas)			
Jan-24	3	W	Ethics			
Jan-26	3	F	Guest Lecture - Brendan Tomoschuk			
Jan-28	3	Su		D2	Q3	
Jan-29	4	M	Data Science questions			
Jan-31	4	W	Dataviz			
Feb-2	4	F	Intro to Analysis			
Feb-4	4	Su		D3	Q4	A2, Project Review*
Feb-5	5	M	EDA I			
Feb-7	5	W	EDA II			
Feb-09	5	F	Inference I			
Feb-11	5	Su		D4	Q5	Project Proposal*, mid-course survey
Feb-12	6	M	Inference II			
Feb-14	6	W	Inference III			
Feb-16	6	F	Guest Lecture (tentative)			
Feb-18	6	Su		D5	Q6	A3
Feb-19	7	M	No Class - President's Day			
Feb-21	7	W	Text Analysis I			
Feb-23	7	F	Text Analysis II			

Date	Week	Day	Topic	Lab due	Lecture quiz due	Assignment due
Feb-25	7	Su		D6	Q7	Checkpoint #1: Data*
Feb-26	8	M	Machine Learning I			
Feb-28	8	W	Machine Learning II			
Mar-1	8	F	Text + ML			
Mar-3	8	Su		D7	Q8	A4
Mar-4	9	M	Nonparametric Analysis			
Mar-6	9	W	Geospatial I			
Mar-8	9	F	Geospatial II			
Mar-10	9	Su		D8	Q9	Checkpoint #2: EDA*
Mar-11	10	M	How to Be Wrong			
Mar-13	10	W	Communication			
Mar-15	10	F	Data Science Jobs			

* indicates group submission. All other assignments/quizzes/surveys are completed & submitted individually.

OTHER GOOD STUFF

Class Conduct

In all interactions in this class, you are expected to be respectful. This includes following the UC San Diego principles of community .

This class will be a welcoming, inclusive, and harassment-free experience for everyone, regardless of gender, gender identity and expression, age, sexual orientation, disability, physical appearance, body size, race, ethnicity, religion (or lack thereof), political beliefs/leanings, or technology choices.

At all times, you should be considerate and respectful. Always refrain from demeaning, discriminatory, or harassing behavior and speech. Last of all, take care of each other.

If you have a concern, please speak with anyone on the instruction team (professor, TAs, or IAs). If you are uncomfortable doing so, that's ok! The OPHD (Office for the Prevention of Sexual Harassment and Discrimination) and CARE (confidential advocacy and education office for sexual violence and gender-based violence) are wonderful resources on campus.

Academic integrity

Don't cheat. Here are some flyers on important aspects of academic integrity every student should know. You can also obtain deeper training from Academic Integrity, click around their website and explore!

You will work together on projects. You should help one another learn in general. Assignments and discussion labs should be completed individually, although you may seek help from your fellow students. However, you may not give answers to each other at any time. You should understand, reproduce on your own, and be able to explain any work you submit. In discussion labs it's ok to be much more leading and provide more of the answer. In assignments you should be more careful about how much detail you are providing in your help to others.

Examples of good collaboration on assignments: - Student posts non-working code to get help, others send a link to a good reference page in **sklearn** documentation, or point out the generic kind of mistake being made (e.g., you've messed up the order of operations). Nobody just writes the correct code for the student. - Student posts a question about a theory or concept. If its not directly related to an assignment question you can choose to answer in full. However, it's generally more helpful for learning if you use the Socratic method: ask the student questions that lead them to find the answer themselves. Also doing this helps you cement your own knowledge of the subject!

- Student posts a question about an assignment problem. Others point out important principles that we have learned in class that can be used to solve it. They describe the important points, or mention important pitfalls to avoid. References to book pages, lecture slides, or lecture video times are helpful. Nobody posts the correct answer for the student.

For group projects, you will work together but every person in the group is expected to understand every aspect of the project. Projects may include ideas and code from other sources—but these other sources must be documented with clear attribution.

Cheating and plagiarism have been and will be strongly penalized.

Policy on using AI programming assistance

I believe that using large language models (LLMs) or other kinds of AIs can help a good programmer work more efficiently. I also believe that using AI assistance will probably slow down the development of a beginning programmer into a good programmer.

My advice: if you struggle to conceptualize how you want to write a program you should probably NOT use an LLM. The beginning or intermediate programmer needs to practice their craft... just like you will never get to be great at a video game by just watching other people's speed runs. I think its fine to use AI assistance if you can immediately imagine how to solve the problem, but you just want help with implementation details, to see alternative algorithms you could use, or help writing it faster.

You can use AI to help you program as long as you:

- make a code comment that cites the AI used, and provides an estimate of how much code in a given block is machine generated. For instance you might write this `# The (code/design) of this function is (completely/mostly/partially) generated by Github Copilot from the prompt "write a python function to bubble sort a list"` Feel free to include a description of any specific changes you made from the machine generated code... it was edited to reduce execution time, to deal with edge cases, to deal with an empty data file, etc..
- don't assume LLM code is working and just hand in without checking. You are always responsible for functionality and understanding how something works.
- understand that programming with LLMs still requires you to do programming. But instead of creating code from scratch (the part many people enjoy) you will need to do debugging and unit testing (the part many people don't like)
- you understand that you may be asked to explain your code at any time. If you can't explain how your code works and why the design is that way, you may lose points

Policy on using AI writing assistance

My beliefs here mirror the section on programming, but with an important extra problem.

I believe using LLMs can help a good writer work more efficiently. I also believe that using AI assistance will probably slow down the development of a poor writer. Even worse, current LLMs tend to produce B+ writing and C+ thinking and arguments.

IMO the only way to use LLMs for writing and thinking is to use them the same way you would a teammate with unreliable writing skills. Bounce ideas off the AI. Let it produce writing and then read it with a critical eye. Carefully edit the output to remove poor logic, false citations that don't exist, trite expressions, and poor rhetoric. Given that it's an AI, you can ask it to produce the output multiple times and start from the best of the lot... then edit mercilessly.

You can use AI to help you write as long as you:

- make a footnote that cites the AI used, and provides an estimate of how much text in a given block is machine generated. For instance you might write `This (text/ideas) in this chapter were (completely/mostly/partially) generated by GPT-4 from the prompt "describe the ethical issues that need to be considered when using data scraped from public posts on social media for an analysis of an upcoming election"` Feel free to include a description of any specific changes you made from the machine generated text... the text was edited to remove wordiness, wrong citations were removed and replaced with actual cites, some aspects of the ideas and structure of the machine text served as inspiration for a full rewrite, etc..
- don't assume the LLM text is good without examining it closely for issues we list above.

Disability Access

Students requesting accommodations due to a disability must provide a current Authorization for Accommodation (AFA) letter. These letters are issued by the Office for Students with Disabilities (OSD), which is located in University Center 202 behind Center Hall. Please make arrangements to contact Prof Ellis privately to arrange accommodations. If you are struggling to get a meeting with OSD, you can let Prof Ellis know and she's likely able to help accommodate while you work to get official documentation. Contacting the OSD can help you further: 858.534.4382 (phone) osd@ucsd.edu (email) <http://disabilities.ucsd.edu>

Difficult Life Situations

Sometimes life outside of academia can be difficult. Please email me or come to office hours if stuff outside the classroom prevents you from doing well inside it. I can often refer you on to the help you need.

If you don't have the most essential resources required to thrive as a student, please contact UCSD Basic Needs who can help you access nutritious food and stable housing, and help you seek the means to reach financial wellness.

If you need emergency food, finances, and/or academic and social support you can also contact UCSD Mutual Aid. They provide mentoring and aid that comes from volunteers among your peers. If you don't need that kind of support, consider joining them in helping your fellow classmates who do.

If you need counseling or if you are in a mental crisis you can contact CAPS. They provide psychiatric services, workshops, and counseling; they also operate a 24/7 crisis hotline at 858.534.3755

Optional Readings:

There are no required readings for this course; however, if you're interested in learning more and reading about data science topics, we recommend the following texts as supplementary to the main elements of the course:

- Grus J (2019, 2nd ed) Data Science from Scratch. This book takes you into HOWs and WHYs, rather than just learning to use a library you don't really understand. This is the harder book, but you will grow tremendously working through it. Can be accessed for free through your UCSD login
- Vanderplas, J (2023, 2nd ed) Python Data Science Handbook. Short and to the point. Both the text and the code are freely available on Github. Learn to use standard libraries to get things done.

We also maintain a list of readings that can provide insight into various data science topics.

What should you call me?

Most students call me Professor/Prof Ellis, and that's great! This is how I typically sign emails to students. I'm also totally OK with you addressing me as Shannon or Dr. Ellis. I would prefer you *not* address me as Ms./Miss/Mrs. Ellis.

What I should call you

I should call you by your preferred name, with the correct pronunciation and any honorific or pronouns you choose. Please correct me (either in the moment or via email/Piazza after the fact...however you're most comfortable) if I ever make a mistake.

How to Get Your Question(s) Answered and/or Provide Feedback

It's great that we have so many ways to communicate, but it can get tricky to figure out who to contact or where your question belongs or when to expect a response. These guidelines are to help you get your question answered as quickly as possible and to ensure that we're able to get to everyone's questions.

That said, to ensure that we're respecting their time, TAs and IAs have been instructed they're only obligated to answer questions between normal working hours (M-F 9am-5pm). However, I know that's not when you may be doing your work. So, please feel free to post whenever is best for you while knowing that if you post late at night or on a weekend, you may not get a response until the next day. As such, do your best not to wait until the last minute to ask a question.

If you have...

- **Questions about course content:** these are awesome! We want everyone to see them and have their questions answered too...so post these to Piazza!
- **A technical assignment question:** Come to office hours (or post to Piazza). Answering technical questions is often best accomplished in person where we can discuss the question and talk through ideas. However, if that is not possible, post your question to Piazza. Be as specific as you can in the question you ask. And, for those answering, help your classmates as much as you can without just giving the answer. Help guide them, point them in a direction, provide pseudo code, but do not provide code that answers assignment questions.
- **Been stuck on something for a while (>30min) and aren't even really sure where to start:** Programming can be frustrating and it may not always be obvious what is going wrong or why something isn't working. That's ok - we've all been there! IF you are stuck, you can and should reach out for help, even if you aren't exactly sure what your specific question is. To determine when to reach out, consider the 2-hour rule. This rule states that if you are stuck, work on that problem for an hour. Then, take a 30 minute break and do something else. When you come back after your break, try for another 30 minutes or so to solve your problem. If you are still completely stuck, stop and contact us (office hours, post on Piazza). If you don't have a specific question, include the information you have (what you're stuck on, the code you've been trying that hasn't been happening, and/or the error messages you've been getting).

- **Questions about course logistics:** First, check the syllabus. If the answer is not there, ask a classmate. If you still are unsure, post on Piazza.
- **Questions about a grade:** Post a **private** note to instructors on Piazza and select the ‘regrades’ tag. Include specifics as to why you feel you mistakenly/unfairly lost points in that post.
- **A specific section-related question:** send a direct message on Piazza to your TA/IA
- **Something super cool to share related to class:** feel free to email Prof Ellis or come to office hours. Be sure to include COGS 108 in the email subject line and your full name in your message.
- **Something you want to talk about in-depth:** meet in person during office hours or schedule a time to meet by email. Be sure to include COGS 108 in the email subject line.
- **Some feedback about the course you want to share anonymously:** If you’ve been offended by an example in class, really liked or disliked a lesson, or wish there were something covered in class that wasn’t but would rather not share this publicly, etc., please fill out the anonymous Google Form