Welcome to COGS 18:
Introduction to Python

COGS 18

Shannon E. Ellis
Associate Teaching Professor
Department of Cognitive Science, HDSI
✉sellis@ucsd.edu

hello

my name is

Shannon Ellis

Annapurna

Divya

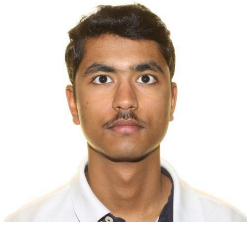
Hui

Liz

Rachel

Zoe

TAs



Akshay

Emanoel

Phoebe

Bella

Roxy

Jenny

Yiting

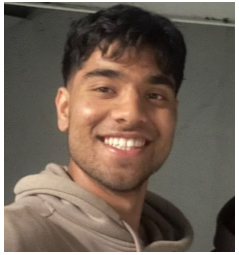
Jonathan

Princess

Zidane

Marissa

Zoe



Pranav

Coco

IAs

The (dreaded) waitlist

1. I do not handle the waitlist - our staff (cogsadvising@ucsd.edu) do
2. I do not have access to the waitlist nor the system that enrolls students from the waitlist.
3. Typically ~1-2 students from each section are enrolled (drop-dependent).
4. Enrollment cannot exceed no. of seats due to testing limitations.
5. The waitlist clears at the end of week 2.
6. You should complete the pre-course assessment if you hope to be enrolled.

If you email me about the waitlist or your specific circumstance/need to take this course this quarter, I will point you to cogsadvising@ucsd.edu.

Let's chat: Teaching & Learning Programming

Why Python?

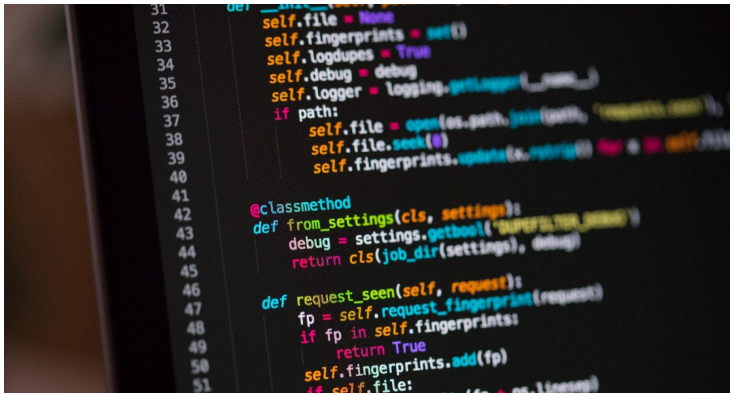
simple(r) syntax

widely-used

Jupyter Notebooks

"It's not the best language for anything, but it's the second best for everything"

Intro Programming courses are often thought of as difficult and are courses with the highest dropout rates



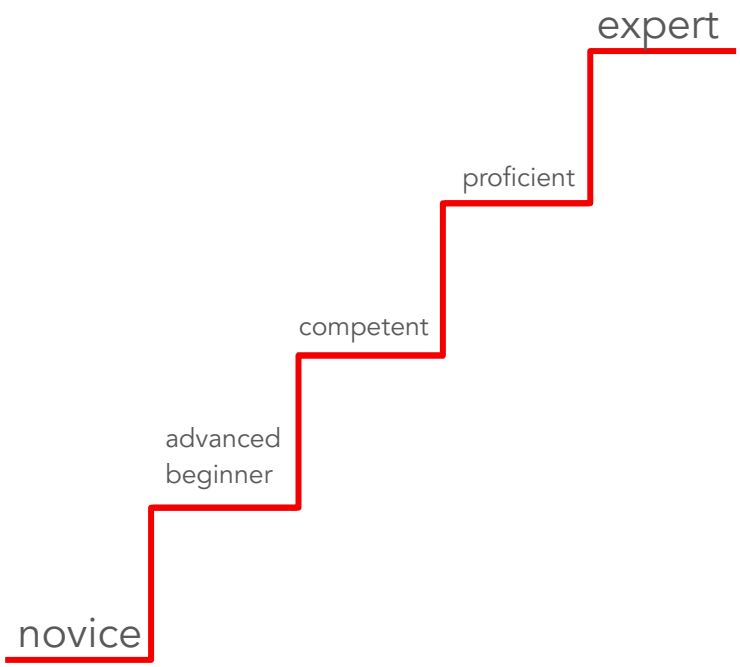
….yet, the only thing that is slightly predictive of success in an intro programming course is…*how successful the student thinks they will be*

Things that do NOT predict success:

- gender
- age
- personality
- math ability

If you think you're going to be unsuccessful…

- On the first video quiz there is a question about how you feel about your likely success in this course - answer honestly! (There is no right answer.)
- Have no prior experience? You're in the right place!
  - Are super fearful?
  - Failed a programming course previously?
- There will be a group focused on additional support.

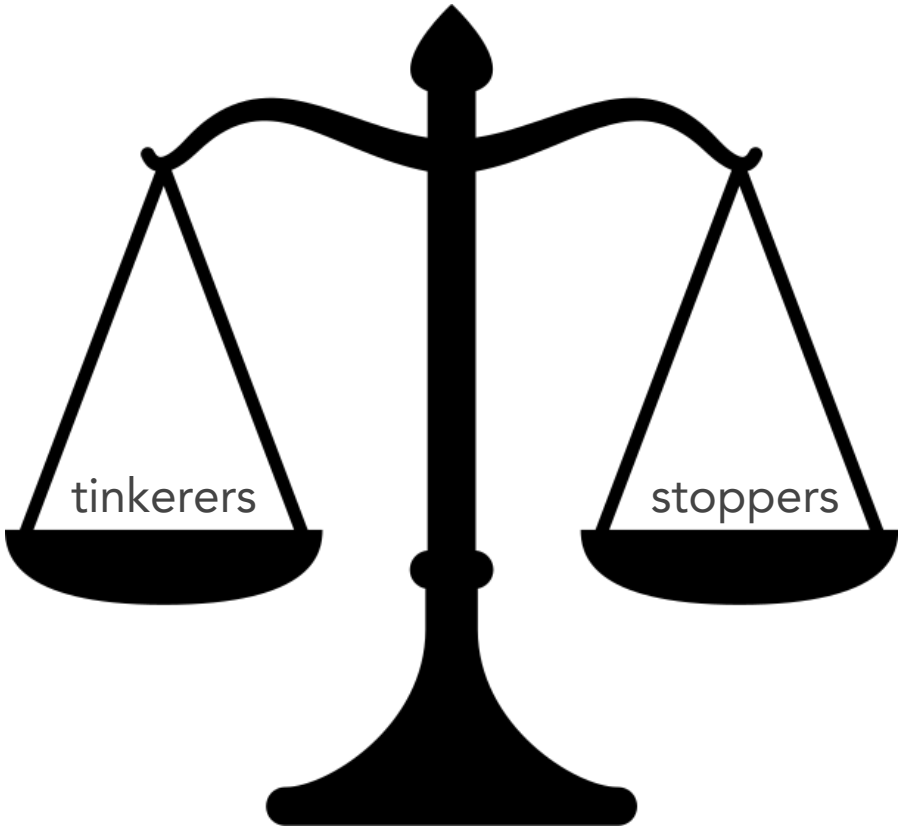


My goal is to have you all be able to program at an introductory level

It's generally accepted that it takes people 10 years to move from novice to expert programmer. But, there are lots of steps in between! We're working to move you further away from novice (& in the direction of expert) than you are right now.

Mixed Messages: We tell people learning to program will be tough and frustrating but that if you're not having fun, you're doing it wrong.

Building Blocks: Too often, we also tell people to "just try things out" without explaining basic concepts. Other courses aren't taught this way...



Be a mover: Make forward progress. Strike a balance between just stopping and tinkering forever.

If you're not moving forward, consider the 2-hour rule.

If you're trying to figure something out and struggling to move forward at all, consider the 2-hour rule. If you're stuck, work on the problem for an hour. If you're still stuck, walk away & take a 30 min break. Then, try again for another 30 minutes or so. If you're still completely stuck, stop and contact us (come to office hours, post on Piazza). If you're not even sure what your question is, include what information that you do have have - what you're stuck on, what you've tried, error messages you've received, etc.

COGS 18: How this course is going to work

To avoid the common pitfalls of intro programming courses, we're going to take the following approach:

Note: Google Forms are for *extra credit only*; every lecture attended is EC (up to 2%); can attend either lecture time on a given day

1. First 2/3 of course: basic concepts
2. Pre-lecture videos + quiz (comprehension)
3. In-class practice (no stakes)
   a. Google Forms for comprehension
   b. time to apply what was just explained
4. Coding Labs (low stakes)
   a. Staff/classmates there to help
   b. Multiple attempts for full credit
5. Assignments (mid stakes)
   a. Completed individually (*can* work together)
   b. Assignments Programmatically graded
6. Oral Exams + Exams + Final (higher stakes)
   a. Closed notes
   b. Exams: TTC-CBTF (Oral exams in lab)
   c. Completed totally individually

Your point of contact for COGS 18 will be the course website: https://cogs18.github.io

Course Website	https://cogs18.github.io	Syllabus, lecture notes, links to stuff below
Canvas	https://canvas.ucsd.edu/courses/64347	grades, OH times/locations
PrairieLearn	https://us.prairielearn.com/pl/course_instance/161488	Assessments (Video quizzes, coding labs, assignments, exams) and regrade requests
Ed	https://edstem.org/us/courses/77515/	questions, discussion
Anonymous Feedback	Submit via Google Form	if I ever offend you, use an example you hate, or to provide general feedback

Please don't send me a Canvas message. The UI is the worst and I miss messages and then feel bad.

Order I reply:

1. Ed
2. Email
3. ~~Canvas~~

In technical classes, Ed is a particularly helpful resource

There are rules:

1. No duplicates.
2. Include Assignment & Question in Summary line.
3. Posts must include your question, what you've tried so far, and resources used.
4. Public posts are best.
5. Helping one another is encouraged.
6. No assignment code in public posts.
7. We're not robots.

COGS 18: How You'll Be Evaluated

	% of Grade	Requirement
Pre- and Post-Assessment	4%	2 surveys (4 total)
Video Quizzes	12%	12 VQs
Coding Labs	16%	8 Coding Labs
Assignments	25%	5 assignments
Oral Exams	5%	2 oral exams
Midterms	24%	2 midterms
Final	14%	Complete final exam

The “I already know Python” option

Graded on:

- Pre- and Post (2% each)
- (2) Oral Exams (5% each)
- (2) Midterm Exams (25%)
- Final Exam (36%)

Opt in:

- Via Google Form (link also on syllabus)
- Once you opt in you cannot change your mind
- If you are unsure, proceed with full course for now; you can always opt-in later.

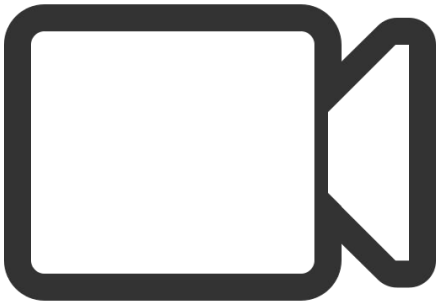
Pre- and Post-Assessment surveys (4%)

Students will have to complete two surveys at the beginning of the course: 1) Coding pre-assessment (~30 min); 2) Information about you (~15 min)

Notes:

1. Be sure to type your PID in correctly to the survey. This is being used to determine you've submitted both surveys.
2. This is an intro programming course. You are NOT expected to know the answers to the questions on the pre-assessment yet.
3. Completion is required for this course (#finaid); *however,* YOU get to decide if you opt into your responses being used for research. Also, correctness does not matter for credit, but real effort does.

Video Quizzes (12%)



The night before most lectures (Mon and Wed), you'll have ~3 short (~3 min each) videos to watch from our textbook and a quiz to complete. Each quiz is worth 1% of your grade.

Why? Student feedback/experience. We're taking the most basic concepts out of lecture to allow for harder examples and more hands-on activities in-class

CodingLabs: apply concepts discussed in lecture using **coding labs (16%)**. Practice makes progress.

Attempt for full credit (2% each)

- Have to make a concerted effort to complete labs
- Coding Labs will be submitted on datahub
- Answers will be sent out the following week
- Encouraged to work with others

You should attend the section to which you're assigned. You *can* attend a different section. However, if one section becomes too crowded each week, we'll revisit this policy.

Note: CodingLabs (and Office Hours) begin Week 2

(5) Assignments (25%) : Jupyter notebooks that are completed individually & graded programmatically.

Assignments due Sun @ 11:59 PM.

Assignment	Week	Median Time Spent (hours)
A1	wk3	2
A2	wk4	4
A3	wk5	4
A4	wk7	5
A5	wk9	5

Oral Exams (5%)

- 5 min exams focused on core concepts and ability to read/explain code
- Will sign up for one slot in weeks 3-6 and another in 7-10
- Will take place during/in lab
- 1-on-1 with a staff member

(2) Midterms (24%) + Final Exam (14%): will be completed individually.

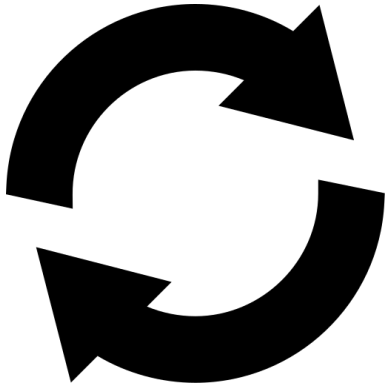
Two parts:

- In-person, closed notes
- Focused on: concepts, code reading and debugging

These completed at the TTC-CBTF.

Sign up for midterms asap! …but don't just pick the latest time slot! (note: you won't be able to sign up for final exams yet)

You *can* sign up for a practice exam slot week 1/2 (3/31-4/11) (encouraged for: nervous test takers/anxious about new environments; NOT required)

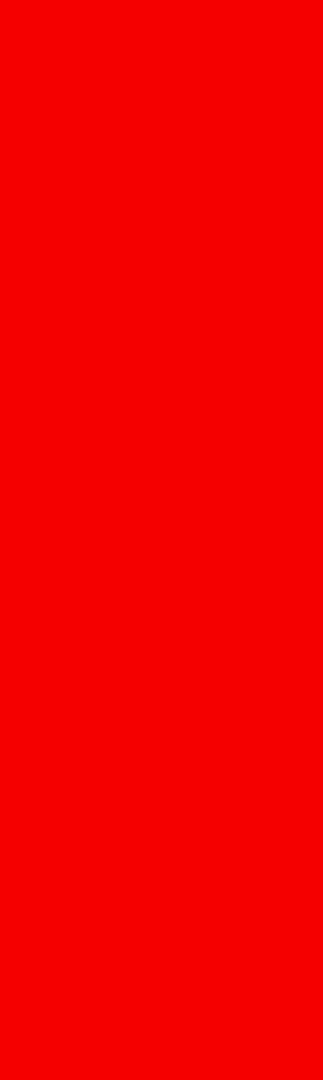


Exam re-take (weeks 9 + 10)

- Must have taken exam as originally scheduled
- Can take *either* E1 or E2
- Will take 75% of highest score and 25% of lowest score
- You don't *need* to sign up for this yet. (it's optional; you don't know if you'll need it…)

Any questions about course logistics?

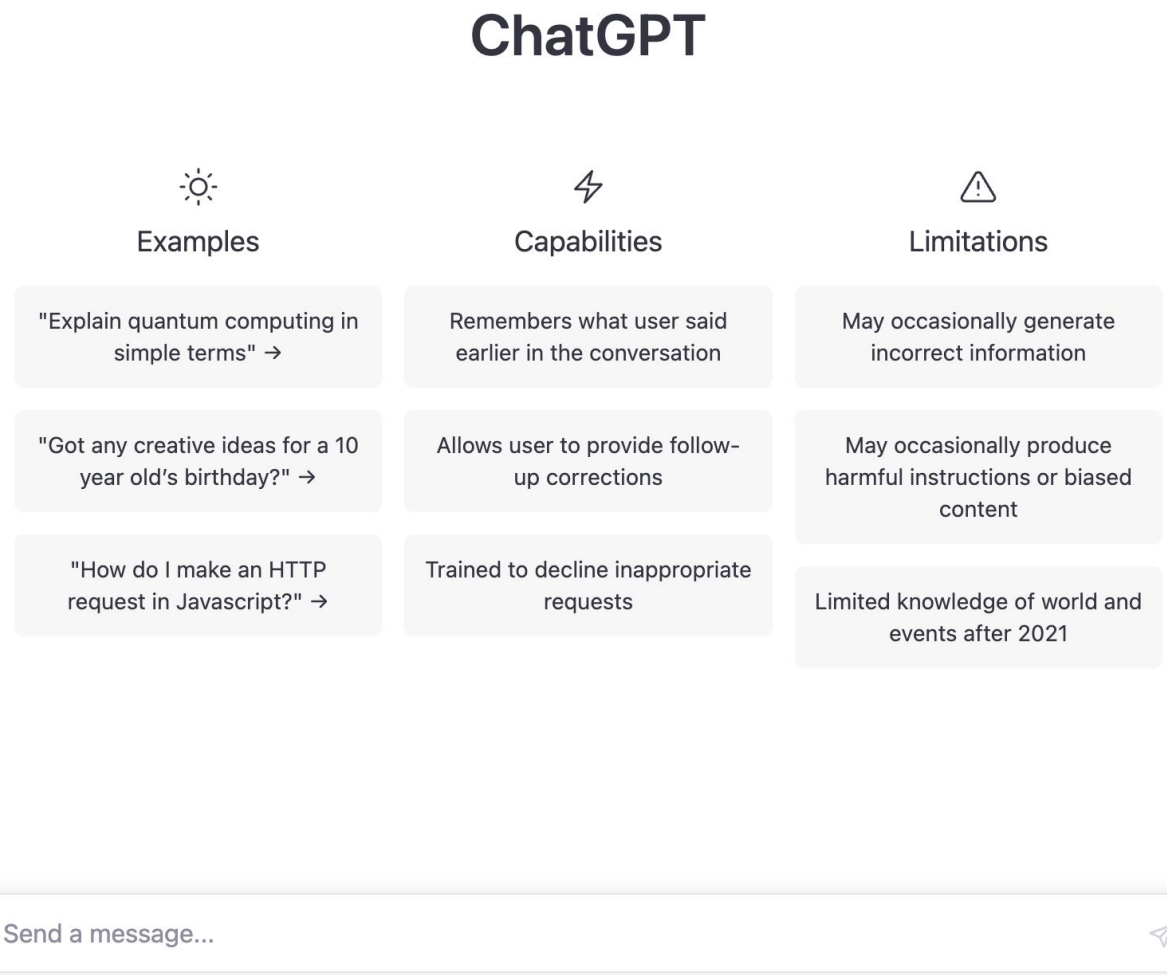
Where to turn for help and practice when learning to program?



Many Avenues for help

- Office Hours
- Classmates
- Ed
- AI/LLMs (ChatGPT, CoPilot, etc.)
- Google with “in python”
- StackOverflow
- GitHub

You ARE encouraged to help one another in COGS 18.



ChatGPT

A conversational LLM that will produce prose and code. It has pros (you can get your questions answered!) and cons (it can be confidently wrong).

When CAN'T you use LLMs in COGS 18? Exams

When CAN you use them? Everything else (but we'll be clear about what you *should* know without an LLM)

…but you need to be able to explain everything you turn in.
Be able to answer yes to BOTH:

- Can I explain each piece of code and each analysis carried out in what I'm submitting?
- Could I reproduce this code/analysis on my own?

Using AI/LLMs as a beginner

Process:

- Try on your own first
- Avoid copy + pasting whole questions
- Come up with your own questions
- Read *and think* about the answers
- Work to <u>understand</u>

Why learning without LLMs?

You don’t know what you don’t know

Core concepts necessary for prompting

Discuss: Google Maps crutch (looking things up forever; not actually saving time)

How we'll help

(and how to best help one another)

Dialogue-based | questions & back-and-forth

Student-driven | work to help you find the solution

Example-heavy | ask about other examples, refer back to previous work, etc.

Scaffold with prior knowledge | review material first to get prerequisite knowledge

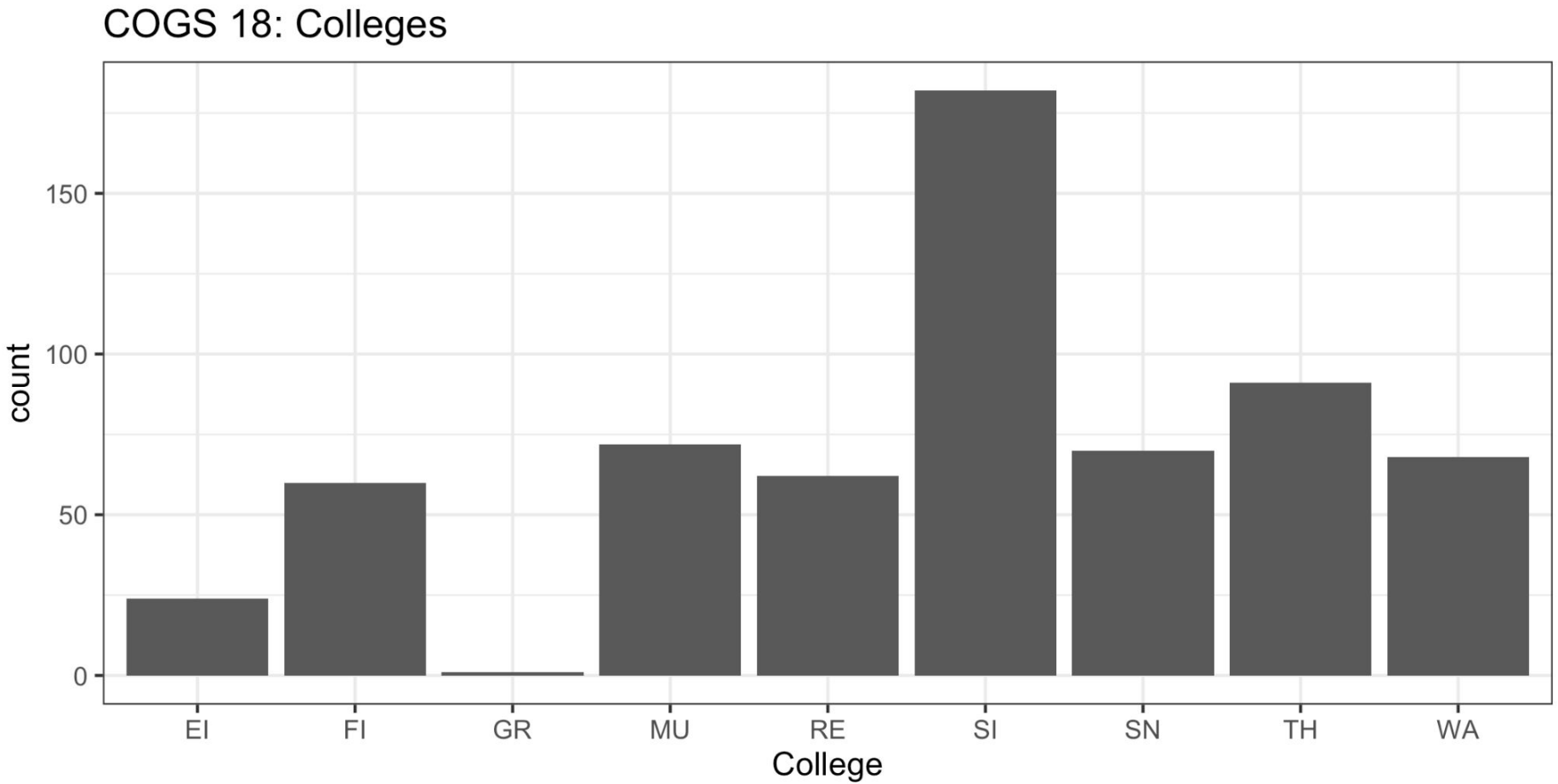
A message for first-gen students, transfer students, and those who don't have older siblings/friends who have attended UCSD/university

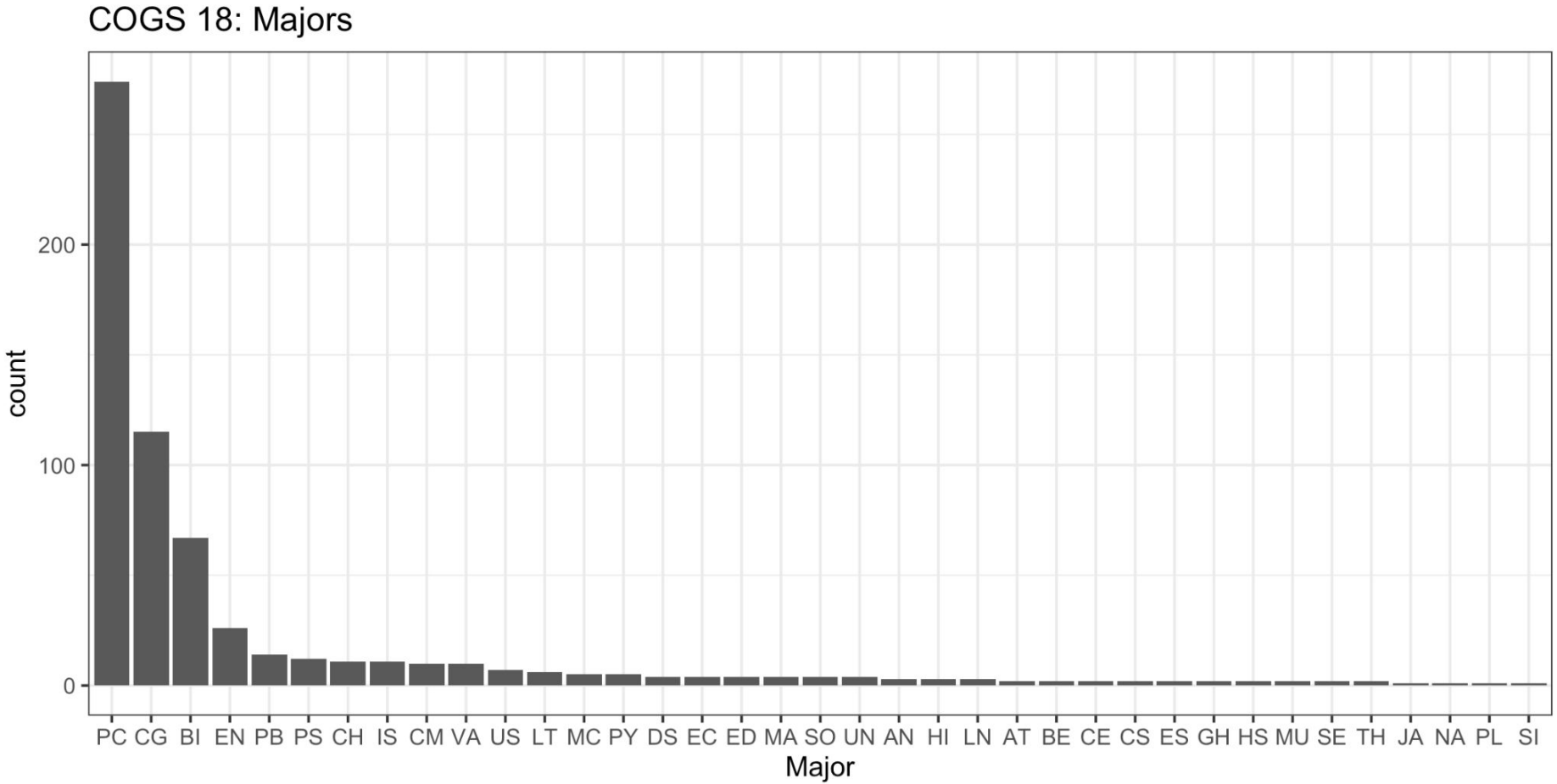
If you are struggling, come to office hours. Ask questions on Ed. Reach out to me to ask for better approaches. Your classmates ARE doing this. And, you're not alone.

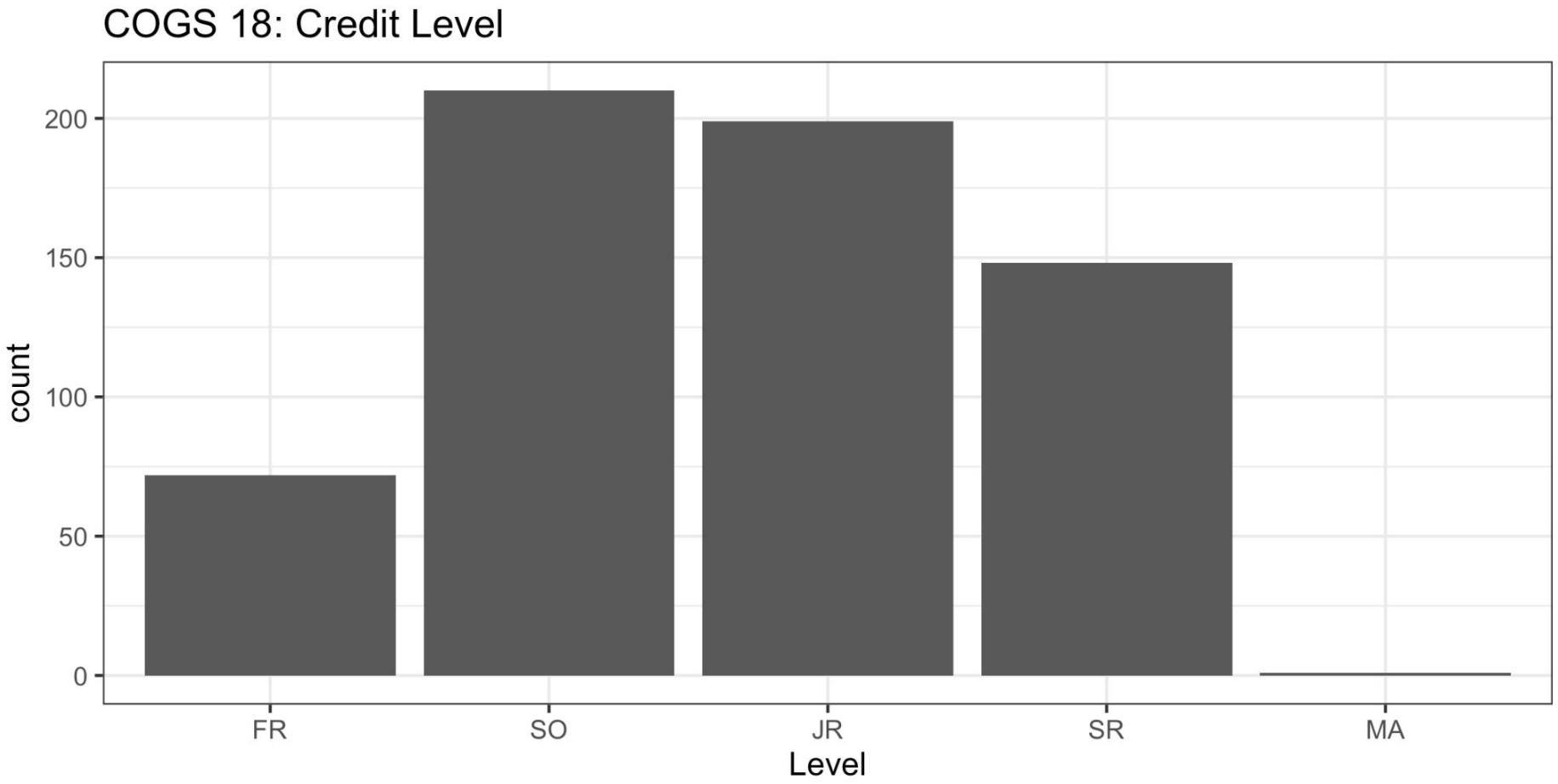
If you need a bit longer on something b/c you fell sick, a family thing came up, work called you in for an extra shift, etc., ask for an extension. Your classmates ARE doing this.

Today I used a PDF slideshow, but every other day of class, lecture notes will be presented in a Jupyter notebook


hello
my name is


COGS 18: Colleges
count
150
100
50
0
EI
FI
GR
MU
RE
SI
SN
TH
WA
College


COGS 18: Majors
count
200
100
0
PC CG BI EN PB PS CH IS CM VA US LT MC PY DS EC ED MA SO UN AN HI LN AT BE CE CS ES GH HS MU SE TH JA NA PL SI
Major


COGS 18: Credit Level
count
200
150
100
50
0
FR
SO
JR
SR
MA
Level

Please complete the two pre-assessment surveys (~20-40min) before Sunday after week 2.