

Welcome to COGS 18:

Introduction to Python

COGS 18

Reminder: This (and all lectures) in COGS 18 are being **podcast**.

Shannon E. Ellis, PhD

UC San Diego

Department of Cognitive Science

✉ sellis@ucsd.edu





TAs

Holly(Yueying) Dong
Dillan Cellier
Eena Kosik

IAs

Efrain Contreras
Evan Eguchi
Brian Fang
Rongyu Ma
Yvonne Luo
Hannah Contreras

Note: CodingLabs & Office Hours all begin **Week 1**

In-person, synchronous learning

1. I will be teaching (so long as I'm healthy) live and in person.
2. Lectures will be podcast.
3. Attendance will be neither required nor incentivized.
4. If you're not feeling well, please stay home. I will do the same.
5. Exams will be take-home.

I am committed to offering in-person learning this quarter. However, I am also aware of circumstances surrounding many students, making it difficult or impossible for them to learn in-person. The course will work for all.

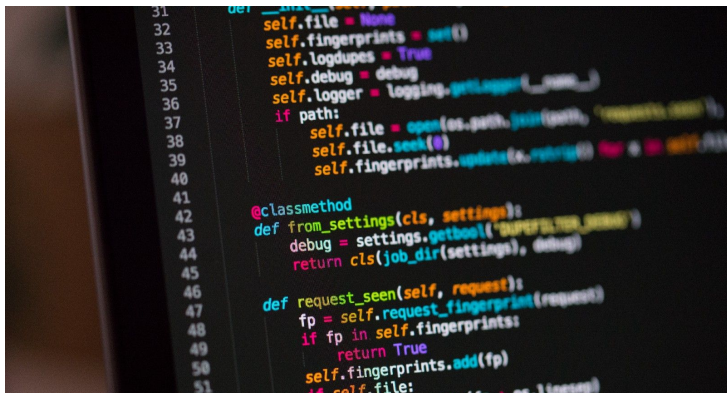
The (dreaded) waitlist

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 ~3-5 students from each section are enrolled by our staff
4. The waitlist clears at the end of week 2.

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

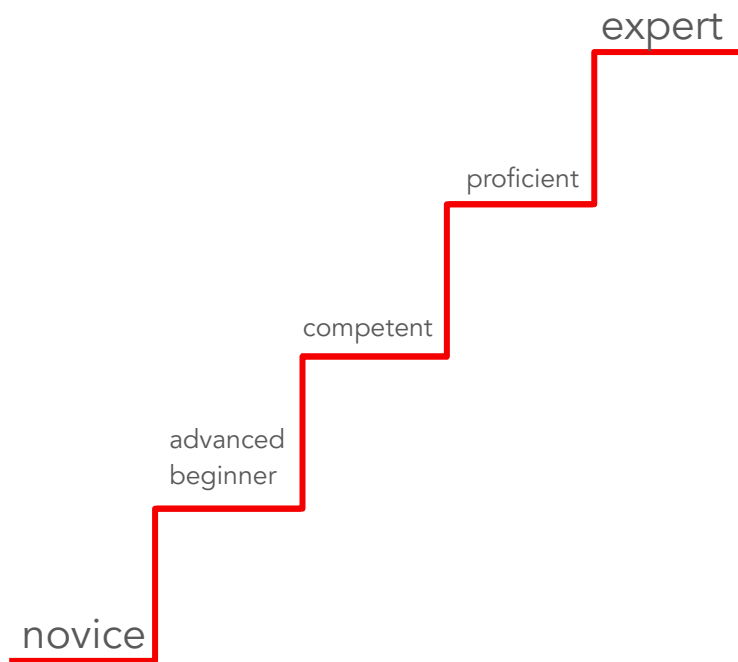
Intro Programming courses
are often **thought of as**
difficult and are courses with
the **highest dropout rates**



....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



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

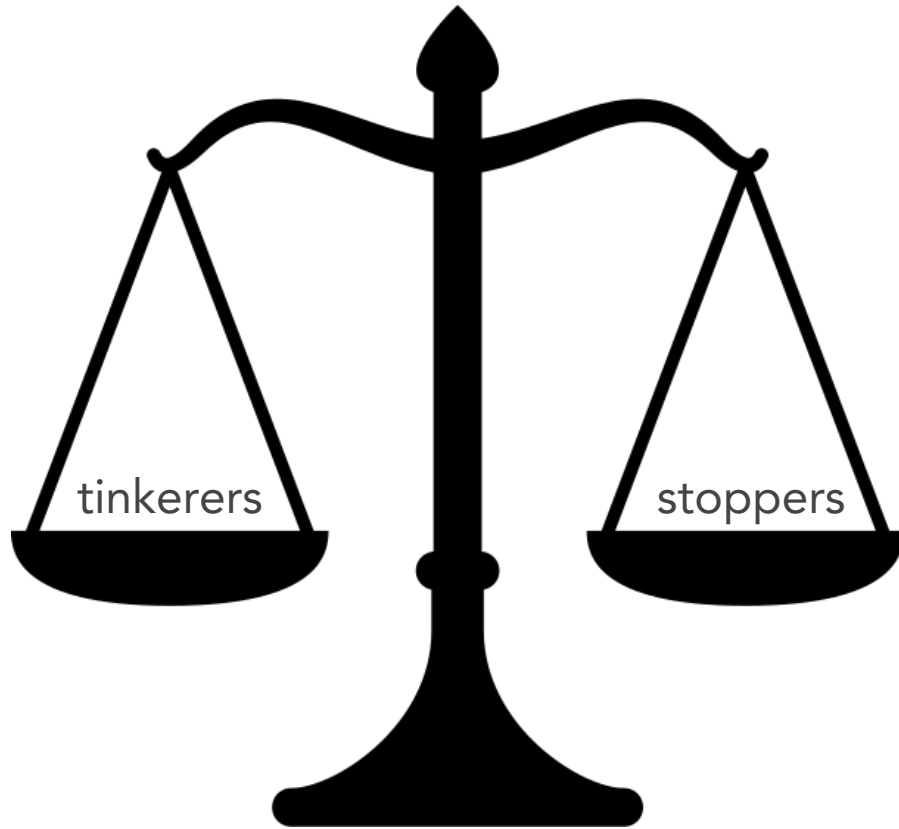
It's generally accepted that it takes people **10 years to move from novice to expert programmer**. But, there are lots of steps in between! We're working to move you further away from novice (& in the direction of expert) than you are right now.



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 Campuswire). If you're not even sure what your question is, include what information that you do have - what you're stuck on, what you've tried, error messages you've received, etc.

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"*

-Brad Voytek





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:

1. First 2/3 of course: basic concepts
2. In-class practice (no stakes)
 - a. iclicker questions for comprehension
 - b. time to apply what was just explained
3. Coding Labs (low stakes)
 - a. Notebooks provided
 - b. Staff/classmates there to help
 - c. Checked for effort, not correctness
4. Assignments (mid stakes)
 - a. Completed individually (*can* work together)
 - b. Programmatically graded
5. Midterms (high stakes)
 - a. Completed totally individually

COGS 18: How You'll Be Evaluated

	% of Grade	Requirement
Coding Labs	16%	Participate In 8 Coding Labs
Assignments	40%	Complete 5 assignments
Midterms	25%	2 midterms
Final	19%	Complete final project or final exam

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
- Can 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.

(5) Assignments
(40%) : Jupyter
notebooks that are
completed
individually &
graded
programmatically.

Assignments always be due @ 11:59PM.

Assignment	Week	Median Time Spent (hours)
A1	wk2	2
A2	wk3	4
A3	wk5	4
A4	wk7	5
A5	wk8	5

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

DATA SCIENCE / MACHINE LEARNING PLATFORM

UC San Diego

Information Technology Services - Educational Technology Services

Help Options ▾



Log In

Registered Users
"username@ucsd.edu"

UC San Diego Jupyterhub (Data Science) Platform

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. Campuswire
2. Email
- ~~3. Canvas~~

In technical
classes,
Campuswire 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.

Sign up: <https://campuswire.com/c/G3CF63BDA>

(2) Midterms (25%): Exams are open-book/open Google but **completed on your own**. Each will include a **combination of types of questions**. There will be a **flexible time window** when these exams can be taken/submitted.

(1) Final Project or
Exam (19%): will be
completed
individually and
submitted
electronically on the
day of the final.

It will be up to you which you do. The **project** will help you learn more and has the opportunity for EC and an A+ in the course, but takes longer. The **exam** takes less time, but the highest grade you can earn in the course is an A and must be completed on your own. You do not have to show up anywhere on the day of the actual final.

All exam
and due
dates are
all listed
on the
course
syllabus
and are in
Canvas

COURSE SCHEDULE

Date	Week	Lecture	Day	Topic	Assignment (11:59 PM)	CodingLab (11:59 PM)
9/24	0	1	F	Introduction		
9/27	1	2	M	Tooling & Integrity		
9/29	1	3	W	Variables		CL1: Tooling & Integrity
10/1	1	4	F	Operators		
10/4	2	5	M	Conditionals		
10/6	2	6	W	Debugging		CL2: Programming I
10/8	2	7	F	Collections	A1: Getting-Started	
10/11	3	8	M	Loops		
10/13	3	9	W	Dictionaries [*]		CL3: Programming II
10/15	3	-	F	Review	A2: Ciphers	
10/18	4	-	M	Midterm I		
10/20	4	10	W	Functions I		CL4: Loops
10/22	4	11	F	Algorithms		



Your point of contact for COGS 18
will be the course website:

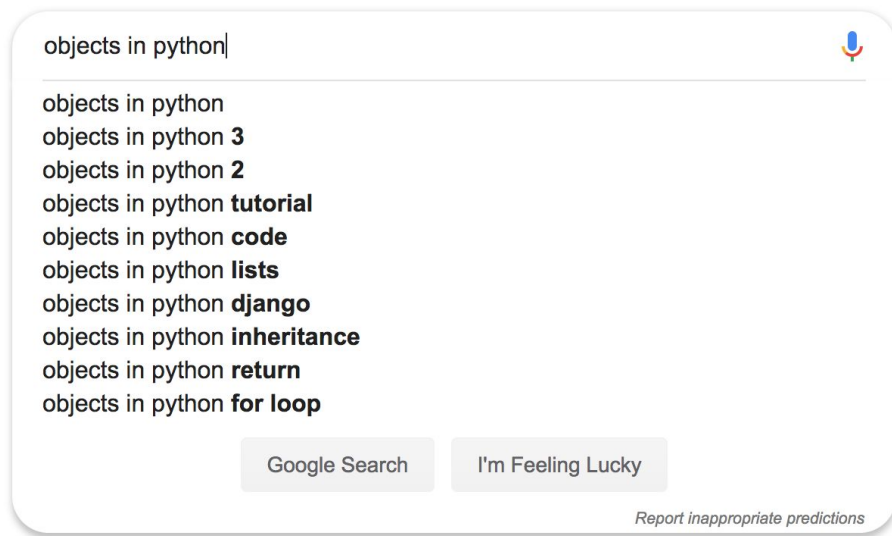
<https://cogs18.github.io>

Course Website	https://cogs18.github.io	syllabus, Coding Lab Answers (& lecture notes)
Campuswire	https://campuswire.com/c/G3CF63BDA (course code on canvas)	questions, discussion, regrades
Canvas	https://canvas.ucsd.edu/courses/29538	grades, lecture videos, zoom links
Datahub	https://datahub.ucsd.edu/	coding labs, assignments, exams, (& lecture notes)
Lecture Slides	Link also on Canvas	Syncing to get most recent lecture slides
Daily Lecture Survey	Link also on Canvas	Daily feedback after lecture (extra credit toward final exam/project)
Anonymous Feedback	Submit via Google Form	if I ever offend you, use an example you hate, or to provide general feedback

Any questions about
course logistics?

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

Including “in python” in your
Google search can be magic

A mockup of a Google search interface. At the top, a search bar contains the text "objects in python|". To the right of the search bar is a microphone icon. Below the search bar, a list of search suggestions is displayed: "objects in python", "objects in python 3", "objects in python 2", "objects in python **tutorial**", "objects in python **code**", "objects in python **lists**", "objects in python **django**", "objects in python **inheritance**", "objects in python **return**", and "objects in python **for loop**". At the bottom of the suggestions list are two buttons: "Google Search" and "I'm Feeling Lucky". Below these buttons is a link that says "Report inappropriate predictions".

objects in python|

objects in python
objects in python 3
objects in python 2
objects in python **tutorial**
objects in python **code**
objects in python **lists**
objects in python **django**
objects in python **inheritance**
objects in python **return**
objects in python **for loop**

Google Search I'm Feeling Lucky

[Report inappropriate predictions](#)

StackOverflow probably has the answer to your question

[Home](#)[PUBLIC](#)[Stack Overflow](#)[Tags](#)[Users](#)[Jobs](#)

Teams
Q&A for work

[Learn More](#)

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Popular](#) [Name](#) [New](#)[python](#) × 1137913

a multi-paradigm, dynamically typed, multipurpose programming language, designed to be quick (to learn, to use, and to

1085 asked today, 6241 this week

[python-3.x](#) × 151128

For questions about Python programming that are specific to version 3+ of the language. Use the more generic [python] tag

273 asked today, 1641 this week

[python-2.7](#) × 89413

the last major version in the 2.x series. Do not use this tag simply to convey the version of Python you're using, unless the question

40 asked today, 219 this week

[python-requests](#) × 9229

a full-featured Python HTTP library with an easy-to-use, logical API.

8 asked today, 81 this week

[wxpython](#) × 6191

a Python wrapper for the cross-platform C++ GUI API wxWidgets.

20 asked this week, 52 this month

[ipython](#) × 6036

a feature-rich interactive shell for Python, and provides a kernel for frontends such as IPython Notebook and Jupyter Notebook.

15 asked this week, 65 this month

[python-imaging-library](#) × 4495

The Python Imaging Library (PIL) provides the Python language with a de-facto standard foundation for image work. PIL's

23 asked this week, 113 this month

[python-3.6](#) × 3882

Version of the Python programming language released in December 2016. For issues specific to Python 3.6. Use more

10 asked today, 43 this week

[python-3.5](#) × 3260

The version of the Python programming language released on September 13, 2015. For issues that are specific to Python 3.5.

9 asked this week, 34 this month

[python-import](#) × 3150

For questions about importing modules in Python

18 asked this week, 58 this month

[python-3.4](#) × 2594

The version of the Python programming language released on March 16, 2014. For issues that are specific to Python 3.4. Use

6 asked this month, 126 this year

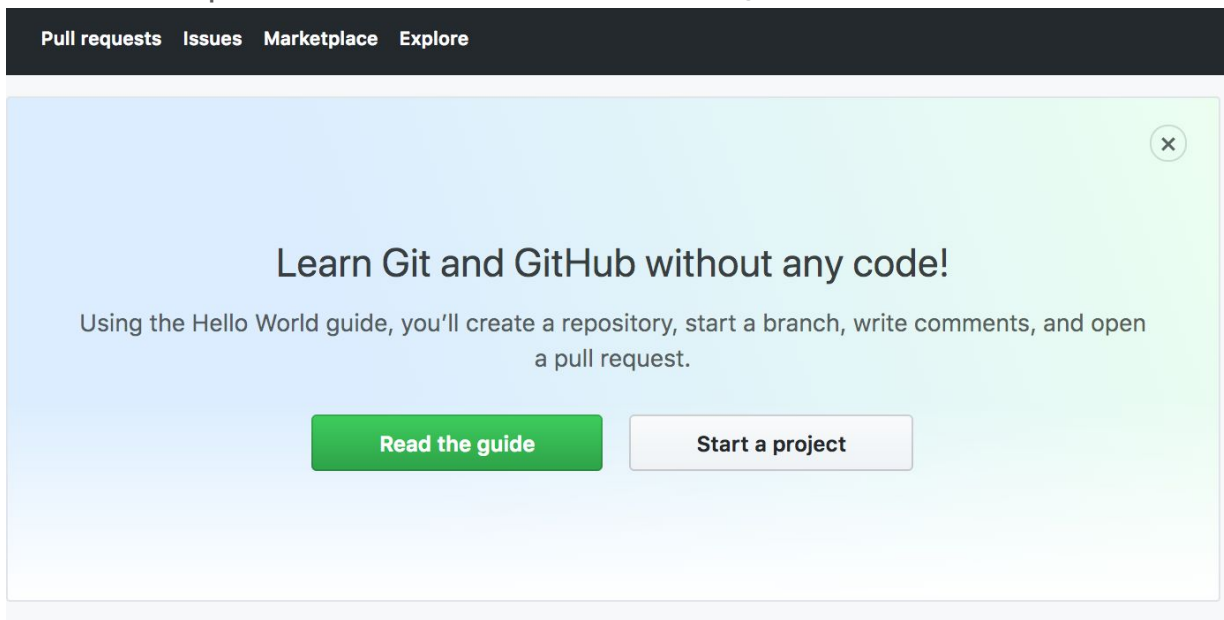
[python-sphinx](#) × 2365

a tool that makes it easy to create intelligent and beautiful documentation. Sphinx is especially suitable for Python

9 asked this week, 34 this month

GitHub: programmers' social media platform

Code is shared on GitHub. In the beginning, it may be intimidating, but I encourage you to familiarize yourself with the platform and share code you write on GitHub.



There are also
COGS18-specific
avenues when
looking for help

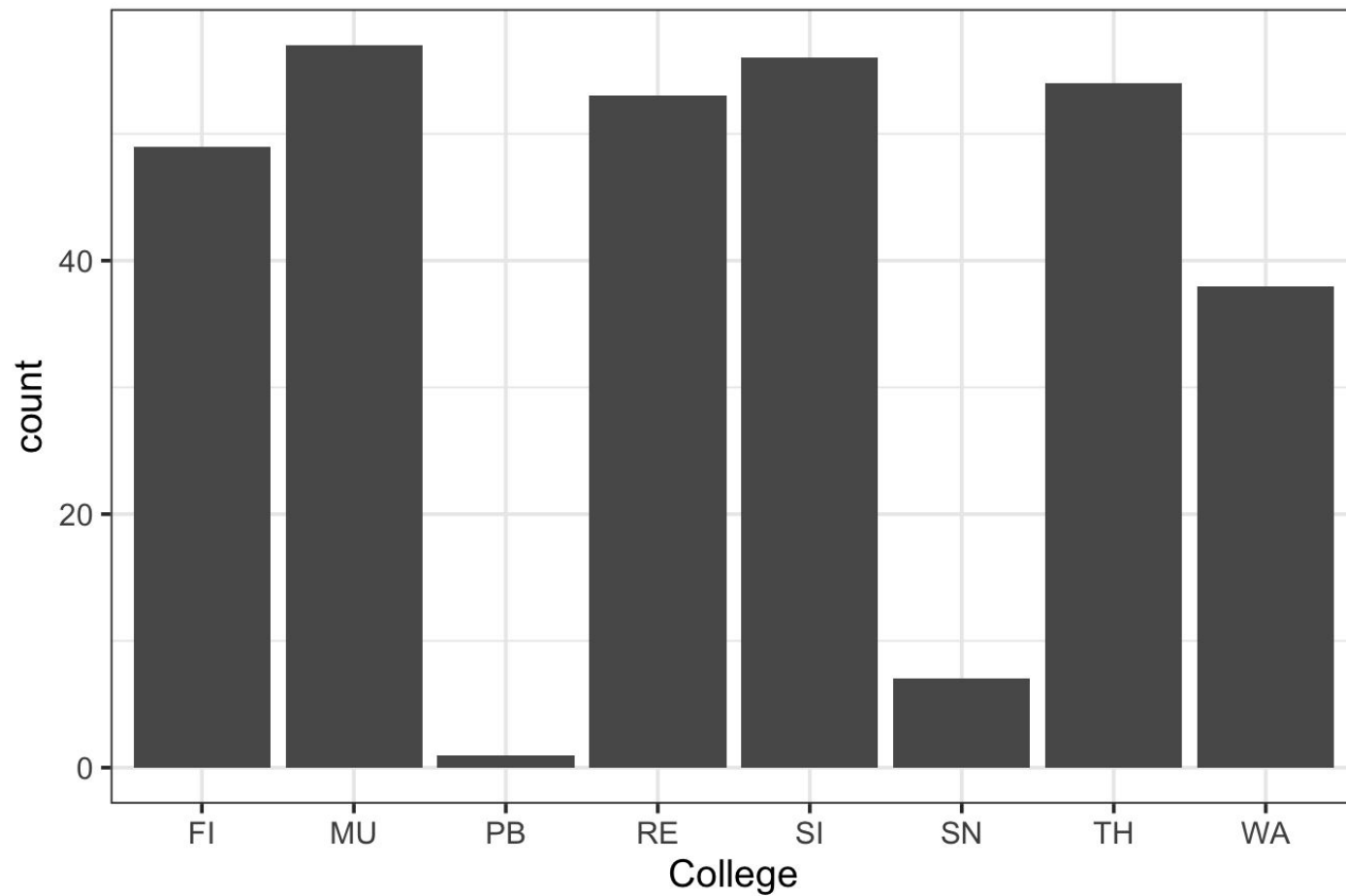
Questions in **CodingLabs**, coming to
office hours, talking to your **classmates**,
or reaching out for help on
Campuswire are all options for you.
You're encouraged to help one another
on Campuswire!



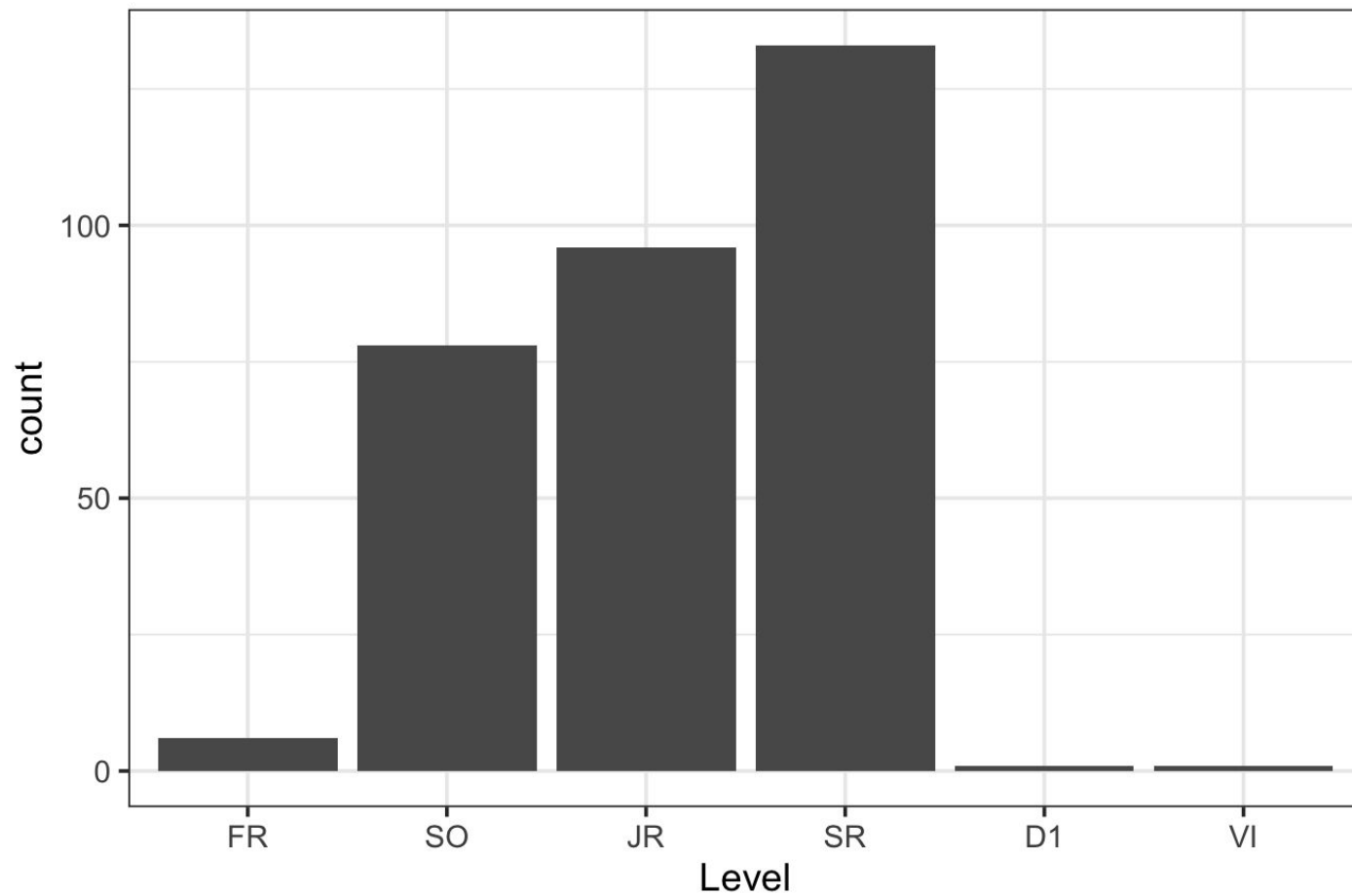
Today I used a PDF slideshow,
but every other day of class,
lecture notes will be presented
in a **Jupyter notebook**



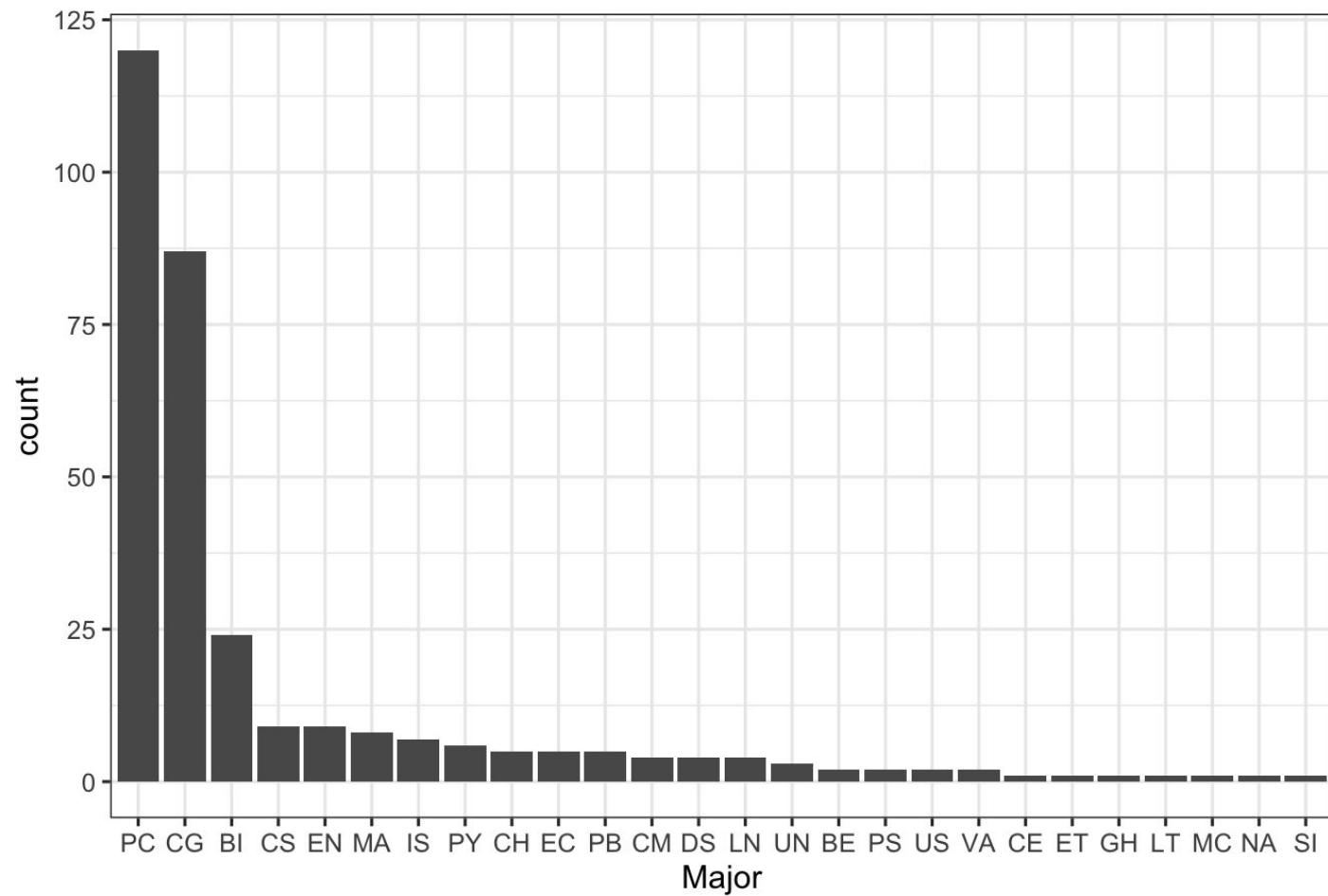
COGS 18



COGS 18



COGS 18





I'm excited to have you all in
COGS 18 this quarter & I'd love
to learn more about you: [Link to
Survey](#) (links also on Canvas)

...and reminder for [daily lecture survey](#)