PSYC 132: Introduction to Programming for Psychological Scientists (Spring 2023)

Meeting times: T/Th 2:25 — 4:15 PM
Classroom: Moore 202

Instructor: Dr. Jeremy R. Manning (jeremy@dartmouth.edu)
Office Location: Moore 349 (but most office hours will be held over Zoom)
Office Hours: By appointment

Course Description

Studying the mind is an increasingly computational endeavor. Modern psychological laboratories use computers to administer experiments, collect data, analyze data, create figures, write papers, and share their work with the world. Related and analogous approaches are used in fields as diverse as finance, art, biomedical science, law, and many others. In this course we will use hands-on training experiences, problem sets, and mini research projects to introduce students to a sampling of the computational tools employed in cutting-edge psychological research. A focus of the course will be on "open science" practices that enable scientists to share and clearly document each aspect of the scientific process.

Note: The content of the PSYC 132 (graduate) offering of this course is equivalent to the PSYC 32 (undergraduate) offering. Both graduate students and undergraduates may enroll in either the undergraduate or graduate offering. (Undergraduates must meet the pre-requisites listed below, or receive instructor permission.)

Undergraduate distribution requirement satisfied: TAS (Technology or Applied Science)
Psychological and Brain Sciences graduate curriculum requirement satisfied: methods

Undergraduate Pre-Requisites

You must have taken PSYC 11 in order to enroll in this course. I also recommend that you take MATH 1 or 3 (Introductory Calculus) or an equivalent course prior to enrolling. Students who have already taken CS 10 should speak with me prior to enrolling.

Course Goals

This course is intended to train students to:

- Gain hands-on experience with computational approaches and tools used in modern psychology research
- Write and understand computer programs written in Python
- Design and implement (using computer code) psychological experiments
- Carry out data analyses on real experimental data
- Build basic computational models of experimental data
- Generate compelling figures to display analysis results visually

Course Materials

We will be working through a variety of free online resources throughout the term. Some of the materials have been written by me, and many of the materials are excerpted from other freely available materials.

All course materials may be found here, and an approximate list of course topics may be found here. Note that the course schedule may change in any given term according to student interests, goals, abilities, and needs.

You will also need a laptop computer (ideally) or another web-enabled device (in a pinch). *You should come prepared to use your computer nearly every class day to do demos, hands-on exercises, and presentations.* You should bring your laptop to class every day.

Format and Overview

This course follows an "experiential learning" model, whereby nearly every aspect of the course will involve hands-on exercises, projects, and discussions. We will use tutorial-style "lectures" to solidify the key concepts, where students will follow along with the material by working through short exercises (and asking questions or solving problems in small groups) as the lectures progress. We will also use (roughly bi-weekly) coding assignments to help you gain a more in-depth understanding of the material. The coding assignments vary substantially in difficulty. Some assignments will take only a few minutes to complete, and others will be more involved.

Although we'll be working from a template used in prior offerings of the course, the precise content will vary by year according to students' backgrounds and interests. In general, the first portion of the course is intended to introduce you to the fundamentals, and in later parts of the course we'll apply the fundamental ideas and tools to specific problems or topics. These later portions of the course will evolve somewhat organically, so it is helpful for students to maintain an open and flexible mindset.

Colaboratory

We will use Google Colaboratory to develop and program analyses. This resource provides an easy means of organizing notes, code, and graphics in a single cohesive format ("notebook"). In addition to being a generally useful tool, Colaboratory has the additional benefit that all code will run on Google Cloud servers (rather than your personal computer). This substantially reduces setup costs and difficulties.

GitHub

We will use GitHub to manage and share data and code. GitHub provides an easy way of managing multiple versions of data and code that may be easily shared and tracked. You will need to create a (free) GitHub account at the beginning of the term.

GitHub Classroom

We will use GitHub Classroom to distribute, submit, and receive feedback on assignments. You'll interact with GitHub Classroom using your GitHub account.

Discord

We will use Discord to provide a forum for asking and answering questions, posting demos, etc. You can join Discord by creating a free account (or using an existing Discord account). Note that our Discord workspace will be publicly visible, so please take care to limit your posts to material you would be comfortable sharing with the world.

Grading

All course assignments (problem sets) will be assigned a point value, added together, and converted to the nearest equivalent letter grade as follows (all scores in parentheses are percentages of the total possible number of points).

Undergraduate students: A (93–100); A- (90–92); B+ (87–89); B (83–86); B- (80–82); C+ (77–79); C (73–76); C- (70–72); D (60–69); or E (0–59)

Graduate students: HP (93–100); P (80–92); LP (60-79); NC (0-59)

Most assignment grading in this course is *automatic*— i.e., done without manual instructor intervention. Automatic grading works as follows:

- You can submit or modify your assignment up until the assignment deadline
- When the assignment deadline is reached, a series of tests will be automatically applied to your assignment:
    - Each test case will be associated with a specific point value
    - You will know some test cases (and their point values) in advance. These are called "public" tests.
    - You will *not* know other test cases (or their specific point values) in advance. These are called "private" tests. You can figure out the total point value across all private tests by subtracting the total point values of the public tests from the number of points the assignment is worth. However, you won't know how many private tests there are or how many points individual tests are worth. The point of these private tests is to get you thinking carefully about how to anticipate and test for edge cases, and to help you develop better and more reliable code.
    - If your code "passes" a given test, you get the points for that test. If it "fails" a test, you don't get the points for that test. There is no partial credit for individual tests.
    - Sometimes the test are simple (like "does your code exist?" or "are your files named correctly?") and sometimes the tests are complicated (e.g., testing a tricky edge case or condition).

- If you haven't submitted your assignment by the deadline, you'll receive a default grade of 0 for the assignment, since no tests will pass.
- Different assignments are worth different amounts, according to their difficult and/or complexity
- If you notice an error in the automatic feedback you receive, please send me a note or tell me (otherwise I won't know there's a problem).
- You may re-submit any assignment up to one additional time after the deadline has passed. Note that new tests may be run when you re-submit, so your grade may change even if you submit exactly the same assignment twice. You can resubmit an assignment using the appropriate GitHub Classroom link (just like you use for the initial submission) and sending me a note so that I know to manually trigger an automatic re-grade.

You'll receive *n* assignments (problem sets and projects) in total. The "standard" (modal) assignment list may be found here. However, the precise value of *n* (and the specific set of assignments) will vary by year according to how much of the course material we get through (in turn, this will depend on students' background and interest). As you might expect, the total possible number of points you receive will depend on how many assignments are given, and how many points each assignment is worth.

Undergraduate students may drop 2 assignments (i.e., only the top *n*-2 assignments will count towards your final grade). **Graduate students may drop 1 assignment** (i.e., only the top *n*-1 assignments will count towards your final grade). In other words, if you choose to hand in all of the assignments, the lowest grade(s) will be dropped. If you choose *not* to hand in all of the assignments, your missing assignments will (by definition) be tied for your lowest point value, and will therefore be dropped.

Additional comments and implementation notes:

- You don't get any bonus points (or penalty) for handing in all of the assignments versus simply not doing the assignments you'll end up dropping. The least risky strategy (grading-wise) is to hand in all of your assignments, but you could consider:
    - Whether your grade will meaningfully change depending on whether you do versus don't hand in a given assignment
    - Whether you're interested in the concept or approach featured in the assignment
    - What other things you could be doing with your time
- Because each assignment has a different point value, not all dropped assignments will affect your grade equally. For example, dropping a low value assignment will affect your final grade very little, whereas dropping a high value assignment could affect your final grade substantially. If two or more assignments are tied for the lowest point value, I'll drop the combination of assignments (up to 2 for undergraduate students or up to 1 for graduate students) that will yield the highest final grade.
- In general, if you put in a consistent effort to engage with the course material, do the assignments on time, etc., you'll likely have no problem getting a good grade (e.g., a P or better as a graduate student, or an A- or better as an undergraduate student). On the other hand, if you find yourself "falling through the cracks" and feel that you're in danger of being lost or doing poorly, I encourage you to reach out to me proactively so that we can strategize how to get you back on track. While it is technically "allowed," it would be very difficult to leave all of the assignments to the last possible moment and attempt to submit them all in one final heroic effort.

The Academic Honor Principle

I expect you to abide by Dartmouth's Academic Honor Principle at all times. However, how that works in my classroom might be a bit different from more traditional settings. At a high level, the guiding principle is that you should behave with integrity, honesty, and fairness (to yourself and others). Here's how that breaks down in the context of this course:

First, I encourage (and expect) you to discuss your assignments with your classmates. We'll often even discuss ongoing assignments in class, and you can directly use materials developed collectively in your own assignments. You can also use online resources (web searches, ChatGPT, GitHub Co-Pilot, etc.) in whatever ways you deem appropriate and useful. That said, *you* need to take final responsibility for whatever you turn in. For example, you should be able to explain any code you turn in (if/when asked). And if you choose work with someone else (or with a tool like ChatGPT or Co-Pilot), you are still responsible for whatever grade the code you hand in earns you.

Second, if you *do* choose to work with others (human or AI), it is still critical that you indicate what work was done by you versus others. Similarly, you cannot "re-use" projects from other courses (or from your other non-course activities) without modifying them. Put simply, you should hand in your own (new) work, even if you collaborated or discussed your assignment with a classmate (or me), and you should cite your sources and/or explain your process if your efforts were collaborative.

If you have any questions about the Academic Honor Principle and/or how it applies generally to this course, or specifically to a particular assignment, please ask me.

Scheduling Conflicts

This class works best when students are participating "live" (synchronously). When you are able, I expect you to attend and be on time for every class meeting. There are two exceptions to this. First, I understand that your personal circumstances may occasionally limit your ability to participate in person. That's fine and you don't need my permission to miss class for any reason *you* deem appropriate. Second, if you are feeling sick (or if you have reason to believe you have COVID, the flu, or some other highly contagious illness), please do **not** attend class in person until you've recovered.

If you are generally able to attend the in-person course meetings, I would appreciate a heads up (as far in advance as you're able to tell me) if you know you will be missing class. That will help me prepare (and, if necessary, reconfigure) group activities, etc.

If you are generally *not* able to attend the in-person course meetings, please reach out to me as soon as possible (and before the end of Week 2 at the latest) so that we can figure out a plan that will help you to still get as much as you can out of the course.

Importantly, **no part of your grade will depend on attendance**. I trust that you will make the best decision for yourself, based on your personal goals and circumstances, about whether or not to attend each class meeting. By the same token, if you do miss a class meeting, it is your responsibility to make up the material. Fortunately this is typically straightforward to do, since all of the course materials (including recordings of lectures and class meetings from previous years' offerings) may be found on the course GitHub page.

Student Needs

I strive to maintain a welcoming and accessible classroom environment. I want you to be an active participant and contributor to ongoing discussions and activities, and that means that every student should feel comfortable in my classroom. If you would like me to be aware of any issues that arise during the term, or any personal needs that may require adjusting how I run my class or how you participate, I encourage you to see me privately. Dartmouth's Student Accessibility Services Office can also help assist with setting up disability-related accommodations.