PSYC 32: Introduction to Programming for Psychological Scientists (Winter 2020)

Meeting times: MWF 12:50 — 1:55 PM
X-hour: Tu 1:20 — 2:10 PM
Classroom: TBA

Instructor: Dr. Jeremy R. Manning
Email: jeremy@dartmouth.edu
Office Location: 349 Moore Hall
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.

Distribution requirement satisfied: TAS (Technology or Applied Science)

Course Goals

This course is intended to train students to:

- Write and understand computer programs written in Python and JavaScript
- 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

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 1 should speak with me prior to enrolling.

Course Materials

We will be working through a variety of free online books and tutorials throughout the term. This will include excerpts from the following sources (in addition to others selected throughout the term, based on specific lecture content, and students' interests and backgrounds):

- How to Think Like a Computer Scientist, by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers
- Whirlwind Tour of Python, by Jake VanderPlas
- Python Data Science Handbook, by Jake VanderPlas

You will also need a laptop computer (Mac, Windows, or Linux) capable of displaying and outputting graphics and running Python and a web browser. (A Chromebook will not work, nor will an iPad or similar.) *We will use our laptops nearly every day in class to do demos, hands-on exercises, and presentations.*

If obtaining these materials presents a financial or logistical hardship for you, please come speak to me as early in the term as possible.

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 also 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.

The first part of the course (roughly weeks 1 — 4) will serve as a highly compressed version of a standard introductory programming course. You will learn what a computer program is and how to write your own computer programs. We will use tutorials and problem sets to solidify the main concepts. Whereas a traditional introductory programming course might provide background on the theory, data structures, computational complexity, algorithm design, etc., here the focus will be on developing a "minimum viable set" of computational skills needed to carry out sophisticated computational work in a psychological (or similar) research setting.

In the second part of the course (roughly weeks 5 — 10), we will focus on applying the programming methods used in Part 1 to psychological research problems:

- We will explore tools for creating customized experiments. Students will implement experiments in small groups and collect data from other students in the class.
- After the data collection phase, we will learn about "data wrangling" tools (for extracting and organizing data) and different approaches to data analysis. We will cover some basic concepts from machine learning (e.g., regression, clustering) and statistics (e.g., hypothesis testing). The focus will be on training students to use and (conceptually) understand techniques commonly employed in laboratory research, rather than delving deeply into the mathematical or theoretical underpinnings.
- Finally, we will explore data visualization approaches, whereby students will learn to display the results of their analyses visually.

- Students will present their work to the class in two parts: in one presentation students will focus on explaining the underlying methods, and in a second (shorter) presentation students will focus on presenting their research findings and interpretations.

Jupyter

We will use Jupyter to develop and program analyses. This software provides an easy means of organizing notes, code, and graphics in a single cohesive format ("notebook"). We will also use Jupyter notebooks for class presentations. (You will learn how to set them up and run the notebooks at the beginning of the course.)

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.

Slack

We will use Slack (a tool for organizing notes, files, and conversations) to provide a forum for asking and answering questions, posting demos, etc. You will need to join the class team at the beginning of the term by creating a (free) Slack account. We'll set up a series of channels (one for each topic we decide to explore) and we will use the tool during and outside of class to keep track of our thoughts and ideas.

Grading

All course assignments 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): A (93–100); A- (90–92); B+ (87–89); B (83–86); B- (80–82); C+ (74–79); C (57–73); C- (50–56); D+ (44–49); D (37–43); D- (30–36); or F (0–29).

Programming Exercises (40%)

At the beginning of the course (during weeks 1—4), you'll build experience programming in Python, plotting and manipulating data, managing code, etc., via a series of tutorial-style problem sets. You'll start these in class, working through the problems in groups (with instructor supervision and assistance), and then you'll finish up the problem sets outside of class as needed. You'll work on a total of 4 problem sets, each worth 10% of your total grade.

Project 1: Experiment (20%)

In weeks 5 and 6, you'll design and implement a psychological experiment. This project may be carried out individually or in small groups. You will also collect data from your classmates, anonymize it, and share it with the class via GitHub, along with appropriate documentation.

Project 2: Data Wrangling (10%)

In week 7, you will apply data organization and "preprocessing" techniques to the data collected by you and your classmates during Project 1. You will share your code (and the organized/preprocessed data) with the class via GitHub, along with appropriate documentation.

Project 3: Analysis (20%)

In weeks 8—9 you will analyze one or more datasets prepared during Project 2. You will fit basic models to your data, generate figures, and compile your findings into a brief report. You will share all code and results with the class via GitHub, along with appropriate documentation.

Project 4: Project Presentations (10%)

You will present your work in two in-class presentations (one will be methods-focused and the other will be results-focused). You will critically evaluate yourself and your classmates using brief written responses.

The Academic Honor Principle

I expect you to abide by Dartmouth's Academic Honor Principle at all times. I encourage (and expect) you to discuss your assignments with your classmates. The class will be heavily collaborative, and I encourage group presentations and collaboration on assignments. However, it's important that you clearly indicate which work was done by you. Further, you cannot "re-use" projects from other courses without modifying them, although some projects will allow you to (optionally) build on prior work. Put simply, you should hand in your own (new) work, even if you collaborated or discussed your assignment with a classmate. If you have any questions about the Academic Honor Principle and how it applies generally to this course, or specifically to a particular assignment, please ask me.

Scheduling Conflicts

This class requires you to be physically present (e.g. to take part in the in-class discussions, give presentations, participate in lab exercises, etc.). I expect you to attend *and be on time for* every class. However, I also understand that in rare circumstances (e.g., religious observances, other fixed-schedule activities) you may need to occasionally miss class. ***If you know you will have a scheduling conflict with this course during the term, please meet with me before the end of Week 2 to discuss appropriate arrangements.***

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.