PSY4219 / PSY6219
Scientific Computing for Psychological and Brain Sciences
Tue/Thu 9:35-10:50am
Online Course
Fall 2020

Instructor

Thomas Palmeri
Department of Psychology
507 Wilson Hall
615-343-7900
thomas.j.palmeri@vanderbilt.edu
http://catlab.psy.vanderbilt.edu
Office Hours: by appointment on Zoom

Course Overview

This course is an introduction to scientific computing for psychological and brain sciences. The goal of the course is for students to develop some proficiency in designing, writing, and debugging computer programs to control experiments, perform data analyses, and simulate simple neural and psychological mechanisms. Topics include computer programming methods, algorithms and data structures, graphing and visualization techniques, computational and numerical methods, data science methods, image and signal processing, optimization, simulation methods, and high performance computing.

Prerequisites

For graduate students, although no previous formal coursework in computer programming is required, I will assume that students have familiarity with computer programming and understand basic programming concepts common to many programming language. For undergraduates, an introductory course in programming is required as a prerequisite (CS 1101, 1103, or 1104). I will try to adjust the pace of the course depending on the amount of computer programming and mathematics background students have had. To do that, I encourage everyone to let me know if the material is going by too quickly or too slowly. Whether graduate student or undergraduate, no prior experience with Python is required.

Online Course Logistics

This course will be taught synchronously online. Lectures will be delivered through Zoom. Lectures will also be recorded and shared on Brightspace after class for students who cannot join live lectures online.

I will sometimes distribute example code before class on Brightspace that will be used during class. I recommend creating a setup for yourself where you can view the course presentation through Zoom on one computer screen and use your computer to go through

example code on another computer screen. This could be by using an external monitor; it could be by viewing Zoom on a tablet and using your computer to go through code. More details of the course logistics will be discussed in class.

Course Requirements and Grading

Homework assignments each week will be used throughout the course to allow students the opportunity to put the scientific computing tools into practice. There will be no exams. Final letter grades will based on percentages as follows:

A 92.5 – 100%		A- 90.0 – 92.5%
B+ 87.5 – 90.0%	B 82.5 – 87.5%	B- 80.0 – 82.5%
C+ 77.5 – 80.0%	C 72.5 – 77.5%	C- 70.0 – 72.5%
D+ 67.5 – 70.0%	D 62.5 – 67.5%	D- 60.0 – 62.5%
	F 0.0 – 60.0%	

While I encourage students to help each other out with conceptual confusions, all homework assignments must be completed individually. Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.

You will turn in homework assignments using Brightspace (see below). I ask two things: First, that you submit a ***single ZIP file*** rather than multiple files. Second, that you make sure that you send me ***everything*** that's needed for the program to run successfully, which includes not only files you created but copies of any files I might have given you as part of the assignment (unless stated otherwise).

Any student officially auditing the course is expected to attend class and can participate in a way commensurate with the amount of work they do on class assignments.

Recommended Free Python Textbooks

A Whirlwind Tour of Python by Jake VanderPlas, available as a free ebook at https://jakevdp.github.io/WhirlwindTourOfPython/

Python Data Science Handbook, also by Jake VanderPlas, available as a free ebook at https://jakevdp.github.io/PythonDataScienceHandbook/

See class slides for how to access (interactive) Jupyter Notebook versions of both textbooks and run on Google Colab (or locally).

Introduction to Computation and Programming in Python, by John V. Guttag

Python

The Python programming language will be used for all assignments in this course. Python is a high-level computer programming language particularly well suited to scientific computing in psychology and neuroscience. It is free, open software that runs on multiple platforms (Windows, Mac, and Linux). It is highly extensible with thousands of libraries and modules written and shared by scientists and engineers from around the world. It allows easy interface with programs written in languages like C, Fortran, Java, or Matlab.

We will be using the Anaconda Python distribution. Details of how to download and install Python will be given on slides uploaded to Brightspace.

Brightspace

All class materials will be posted on Brightspace (readings, class slides, example code, homework assignments). You will turn homework assignments in on Brightspace. Grades will be posted on Brightspace.

Piazza

We use Piazza to facilitate discussions about assignments, Python coding, Jupyter notebooks, PyCharm, and Python setup. Rather than emailing questions to the teaching staff, I encourage you to post questions on Piazza. Answers we give on Piazza might help other students with similar questions. I also encourage students to help other students on Piazza – doing so will not only help you learn the material more deeply, but may earn you some extra credit at the end of the semester for doing so. The course site on Piazza is here: http://piazza.com/vanderbilt/fall2020/psy42196219

Possible Course Topics

Python and Jupyter Notebooks
Numeric and Non-numeric Types
Array, Vector, and Matrix Operations
Python Language Fundamentals
Programming Techniques
Graphing, Plotting, and Visualization
Random Numbers and Monte Carlo Methods
Scientific Computing
Simulation Methods
Images and Image Processing
Signals and Signal Processing
Experimental Control Programming
Pandas and Data Science Methods
Web-Based Experiments
High-Performance Computing
Debugging Techniques