COGS 14P: Scientific Python for Research

Professor Ramesh Srinivasan, Cognitive Sciences

Version 0 9/21/21

E-mail: r.srinivasan@uci.edu
Course Web: **https://canvas.eee.uci.edu/courses/37161**
Office Hours: **W 2-3 pm, F 11-12 and by appointment**
Office: **Professor Srinivasan's Zoom Office Hours - https://uci.zoom.us/j/92844739334**

Teaching Assistants:
Discussion 1: F 2-3 Jenny (Qinhua) Sun
E-mail: qinhuas@uci.edu
Office Hours **M 9-10, Th 5-6 and by appointment**
Office: **Jenny's Zoom Office Hours - https://uci.zoom.us/j/91511758283**

Discussion 2: F 3-4 Minkyu Kim
E-mail: m.kim@uci.edu
Office Hours **T 2-3, Th 3-4 and by appointment**
Office: **Minkyu's Zoom Office Hours - https://uci.zoom.us/j/95254426634**

Course Description

Introduces Scientific Python for research applications. Emphasizes data analysis, data visualization, simulation, and experimental control with applications in cognitive sciences, psychology, and neuroscience. Topics include data structures, execution control, graphical visualization, and interaction with sound and display interfaces.

Hybrid Course Organization

This class is a hybrid course. Each week you will attend **one** assigned class day on Tuesday or Thursday and one discussion section

Discussion	Lecture	Teaching Assistant
F 2-3	T 12:30-1:45	Jenny (Qinhua) Sun
F 3-4	Th 12:30-1:45	Minkyu Kim

Each of the lectures will be recorded on zoom and available to **both** sections of the class.

In addition, each week a Tutorial worksheet will be provided which emphasizes the new syntax and coding concepts discussed during the week. The best way to do this class is to attempt the Tutorial *before* your lecture to familiarize yourself with the topics. If necessary, complete the Tutorial after the lecture.

Discussion sections are available for your benefit. Attendance is *not* required. The purpose of the Discussion section is to have a place where you can meet your TA and ask questions about the Tutorials, Homeworks, or Projects. Its also place to meet with your classmates to work on the class work.

Course Structure

Topics

1. Introduction - Software Organization
2. Variables and Expressions
3. Computing with Numpy variables and methods
4. Lists and Dictionaries
5. Principles in visualizing data
6. Making plots using Matplotlib
7. Review of logical operations
8. Data organization using logical operations
9. Applications: Data analysis and visualization
10. Execution control - branching
11. Execution control - loops

12. Applications: Model simulation and analysis

13. Functions and Modules

14. Managing Python environments

15. Good programming practices

16. Working with images and sounds

17. Applications: Control of experiments

Assignments

Tutorials

Each Friday a Tutorial will be assigned and will be due at 11:59 pm the following Friday.

Homeworks

Homework will be assigned regularly in the class. You will be given at least a week to complete the Homework assignments.

Projects

Three major coding Projects will be assigned during the class. These projects will take the methods you have learned through and apply them to three major applications of programming we will discuss in the class: (1) Data analysis and visualization, (2) Model simulation and analysis, and (3) Control of experiments. You will be given at least 2 weeks to complete each coding Project.

Software

Anaconda Python

There are three options to get software tools you need to do the homework for this class.

(1) Use this computer lab.

This computer lab is available on a variable schedule on weekdays (typically after 4 pm up to 10 pm), and on Saturdays. All of the course software is available on these computers
The advantage of this option is that the computer labs are already set up with Python and the

software options for the course. One disadvantage is the limited hours. The other is that customization of the software is linked to the physical machine and not your account.

(2) Use the virtual computer labs:

Apporto Computer Labs - https://uci.apporto.com/

The advantage of this option is that the computer labs are already set up with course software. In addition, customization to the software are saved to your account. The disadvantage is that the usability of the virtual computer labs depends on your home network speed.

(3) Install Anaconda Python Libraries on your (Windows, Mac, or Linux) computer.

Anaconda Python - https://www.anaconda.com/products/individual

Why Anaconda Python?

Your computer may already have Python installed on it, which in theory you could configure to use for this class. It is much easier to simply install the Anaconda distribution of Python which is free and comes with an nearly complete library of software for scientific computing. Please be aware of the possibility that there are two versions of python on your computer and make sure you are using Anaconda Python

Software Tools - Integrated Development Environment

The Lecture Notes and Tutorials in this course will be in the form of a Python notebook (colloquially a Jupyter notebook, with extension **.ipynb**). A python notebook integrates instructional materials written in markdown (like html on websites) with blocks of code that can be executed.

Python programs that you write for the Projects in this course (and in the future) are usually written as Python programs (with extension **.py**). These are just programs and do not include any markdown instructional materials.

It is very helpful to make use of an **IDE** (Integrated Development Environment) to work on your programs.

The choice of IDE is a deeply personal one. There is no objective reason to prefer one over the other, although I can assure you everyone who uses Python has strong views about this.
For this class, I plan to use the VSCode IDE which can be used with both python notebooks and python programs. You may download the VSCode IDE at

Visual Studio Code - https://code.visualstudio.com/download

You are free to use whatever IDE you like. A popular IDE for Python notebooks is Jupyter Notebooks, which is part of Anaconda Python, and runs in your browser. For python programs

a simple easy to use IDE is Spyder which is similar to MATLAB or R studio. There are also more complicated, feature rich IDEs like PyCharm.

Final Exam

There is no Final Exam for this course. **The final coding Project is due Friday, Dec 10 at the Finals time, 12:30 pm.**

Grading Policy

Tutorials: 30% of grade.
Homework: 20% of grade.
Coding Projects: 50% of grade.

Late Work Policy

It is essential that you complete all the work of the class. You cannot learn this material skipping over a lesson.

It is essential that you keep pace with the class, especially with the Tutorials. The material in the class builds on previous weeks, so if you fall behind by a week it is very difficult to recover.

I also understand that you have competing classes, work, and other obligations. If you are not able to meet the deadline posted for each assignment, you should still submit the assignment late.

Tutorials are always due Friday at 11:59 pm. A 10% penalty will apply if the Tutorial is submitted by Sunday 11:59 pm. if you have not submitted by that time, you must contact me to make arrangements and discuss your progress in the course. Or I will contact you!

All other late submissions of Homeworks and Projects will be penalized by 10% for each week late.

It is better to get 60% on assignment than 0%.

If you submit late work please notify your TA by email as they may not be aware of your late submission.

Disability services, academic dishonesty, and copyright policy

Disability Services link: https://dsc.uci.edu/
Academic Dishonesty link: https://aisc.uci.edu/students/academic-integrity/index.php
Copyright policy link: http://copyright.universityofcalifornia.edu/use/teaching.html