COGS 106: Computational Lab Skills

Professor Ramesh Srinivasan, Cognitive Sciences

Version 0 09/28/23

Course meetings: TTh 2:00-3:20
E-mail: r.srinivasan@uci.edu
CANVAS PAGE: **https://canvas.eee.uci.edu/courses/58601**
Office Hours: **By Appointment**
Office: SBSG 2336

Course Description

Introduces computational methods for laboratory science. Topics include code management, data handling, exploratory data analysis, data visualization, linear and nonlinear data models, supervised and unsupervised machine learning. By design, this class is an introductory level and will emphasize foundational ideas that make data science methods accessible to further study.

Prerequisites

COGS/PSYC 10A/B/C OR STATS 110 OR STATS 7
COGS/PSYC 14P or ICS 31 or MATH 9

All of the Lecture Notes I will provide incorporate Python code for you to execute. I will spend more time at the start of the class explaining the programming details which will facilitate reviving dormant Python skills or mapping from Matlab to Python.

In addition, I will provide Lecture Notes from my course introductory python course (COGS/PSYC 14P), for your review.

Weekly Lecture Notes and Exercises will be provided as Jupyter Notebooks using the Python programming language. Jupyter Notebooks combine text (written in markdown) and code to create **interactive** Lecture Notes that allow you to execute and modify computer programs to build your intuition about the simulation and data analysis methods covered in this course.

All the Lecture Notes in this class will provide example code which you will execute and create the figures within the Lecture Notes, matching the figures I present in the Lectures. In-class assignments and homework Exercises will ask you to write Python code in order to solve practical simulation and data analysis problems modeled after the Lecture Note examples.

Data Analysis Project

The major component of this course (and your grade) is a data analysis project. The goal of this project is to take the methods developed in the course and apply them to new data and to interpret the results of these analysis in a final report, submitted at the end of the quarter.

Data Sets

I will provide some options for the data analysis projects consisting of data sets I have curated.

However, I recognize that my interests are not the same as yours, so I am quite open to a final project that you define. This may come from data sets that you have access to because of your participation in research labs on campus. It may also come from data sets you identify from a search through data archives such as kaggle.com or openneuro.org.

If you choose to use your own data, you must get your proposed data set approved by me. If the data comes from a research lab, you must also ask for permission to use the data for this class, and for me to have access to the data for evaluating your project.

Scope of Work

The final project should incorporate some basic visualization of the data, and some of the techniques discussed in the class customized to the specific data analysis goals of your project. An important goal of this class is *designing data analysis that has explanatory power*.

Collaboration

You are allowed to collaborate with a group (preferably no more than 3) in the data analysis project and submit a single Github repository and Paper/Presentation.

Project Proposal

The first assignment will be defining the scope of work for your project, your collaborative team, and getting it approved by me. This will be submitted as a written project proposal (< 2 pages). The scope of work will depend partly on group size. Part of the goals of this course is to gain the skills to identify *What are the important questions to answer for my data set?* So, in addition to identifying the analysis you would like to do with the data set, your proposal should explain what you believe you will learn from the analysis.

Final Report

The final report should be submitted in the form of a short paper (5 pages) and a project Github repository that provides: (1) a brief background and explanation of the data (1-2 paragraphs) (2) an explanation of the objective of each analysis (3) The repository should contain easy to run CODE I can run on your data that runs the analysis and makes each graph or returns each quantitative result. The code should be well documented. (4) an explanation of the meaning of the graph or result. (5) A discussion of what your analysis learned and what further data/analysis would be useful.

Final Presentation

During the last week of class and finals week, each group will make a presentation on their progress on their final project. I'm thinking about this still.

Software

Anaconda Python

All of the exercises and projects in this class make use of the Python programming language. 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 principle 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 as well as a number of other useful tools like the *Jupyter Lab* IDE.

VS CODE

My preferred IDE for Jupyter Notebooks is Visual Studio Code.

https://code.visualstudio.com/

There is no requirement to use VS code. You can use Jupyter Notebooks or Jupyter Lab, if you are already used to them and like them. Some people like PyCharm. All of the materials in this class should work in any IDE.

GitHub

This course is organized from Github Classroom.

What would make this the most effective is if you were to make use of git software to interact with the repository. This can either be done with commmand line tools (better) or using GitHub Desktop or inside VS CODE.

https://desktop.github.com/

Inside VS Code there are some instructions of how to set up git.

Command Line Interface (CLI) is actually the best way to work with github. if there is interest I would be happy to show you how.

Course Structure

Course Topics

1. Orientation to Python, Jupyter Notebooks, Github Classroom
2. Python Fundamentals for Data Analysis
3. Fitting Models to Data (Least-Squares)
4. Fitting Models to Data (Maximum Likelihood Estimates)
5. Parameter Estimation (Numerical Optimization)
6. Supervised Learning: Classification and Logistic Regression
7. Unsupervised Learning: Clustering
8. Latent Variable Models
9. Foundations of Machine Learning
10. Get started with Neural Networks

Homework

Homework will be assigned weekly, due about one week later.

Final Exam

There is no Final Exam for this course. **Final Project is due on Thursday Dec 14, 5 pm.**

Grading Policy

In-class work and Homework: 50% of grade.
Final Project Proposal: 5% of grade
Final Project Repository/Code: 20% of grade Final Project Paper/Presentation: 25% of grade.

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