Programming and Data Analysis
Geol 497D/GeoSci 697D - Spring 2020
Monday and Wednesday at 2:30 PM - 3:45 PM in Morrill 271

Instructor: William P. Clement
Office: Morrill, Room 269D
Office Hours: Tuesday, 10:00 – 12:00, or by appointment.
Contact: wclement@geo.umass.edu

We will use the textbook **Learning Scientific Programming with Python** by Christian Hill.

Course Summary

Modern Earth and Environmental Scientists deal with complex and often very large quantitative data sets that are typically not useful or understandable in raw form. Thus, quantitative data analysis skills are highly desired and useful in quantitative Earth science subdisciplines. This course provides an introduction to processing, visualizing, and interpreting quantitative Earth and Environmental Science data using scientific computing techniques widely used in the Earth sciences. We will use python, a widely-use, free, open-source programming language. I do not expect you to have any previous programming experience. I will use part of the class time to introduce the necessary computational background and python syntax. After the introduction, we will use the class time to start the python assignments. Working together really helps the learning process. Expect to work on a new python assignment weekly. Topics covered include Earth science applications of: conditional statements, loops, vector operations, automated data analysis & visualization, differentiation, interpolation, curve fitting, image processing, and three-dimensional visualization.

Course Goals

The main goal of this course is to provide a computational and quantitative skill set relevant for processing, filtering, analyzing, and visualizing quantitative Earth Science data efficiently and accurately. By completing this course, you will gain experience in the basics of computer programming, data visualization, and mathematical principles relevant to Earth sciences. You will learn to make your own custom tools that automate computation and visualization tasks so that a problem need only be solved once. At the end of this class, you should be able to: write your own computer programs, break a problem down into smaller, more easily coded chunks, appreciate the power of computers, and develop a broader perspective on "ways of thinking".

The overarching goal is that the course will demonstrate the wide applicability of computation in the Earth sciences and provide students with the confidence to pursue quantitative research projects during their academic and professional careers.

Homework Assignments

Homework assignments will be mostly computer-based; however traditional pencil and paper planning and calculations will be performed to validate algorithms and to test knowledge of computational tasks. Thus, homework assignments will be a mix of basic python methodology questions and applications of computational methods to Earth Science data. Homework exercises have been

designed to reinforce programming methods taught in lecture and are key to understanding course material. Students should come to every class with paper, pencils, and a USB flash memory stick (often called thumb drives, or flash drives). The purpose of the USB devices will be to save your homework assignments so they don't accidentally get erased.

I will also have you look over one of your classmates computer code. This will give you the chance to perhaps see different ways to code the same problem. You will choose an alias so that your assignments are anonymous. All your assignments should use this alias to identify you. You will look over the python code and see if you can understand what the code does. Well documented code is important and we often don't document code, or do it poorly, because we are in a hurry. Somebody should be able to figure out what your code does by reading the code and the documentation within the code. Especially since we are moving to an era of reproductive research.

Grading

Your final grade will be based on completion of the homework assignments.

Coding tips

Below is some hopefully helpful advice:

1. The absolute best way to learn a programming language is to take simple examples, and then play around with various commands to see how things work. When you are reading through the text and any lecture notes, have python open so you can try the examples. My homework assignments assume that this is how you are reading the text every week.

2. Don't forget to read the built-in python documentation. It is very complete and comes with simple examples that you can experiment with. This is how I learned python. I did not have a formal python course in college or grad school.

3. Stay in class for the entire period and ask questions to make sure you have made significant progress before class ends. Class periods are your chance to get guided help in getting your code started.

4. Try to finish your homework several days before they are due. This way, if you get stuck, you will have time to ask questions to me or your classmates. If you try to do the assignments on the night before they are due, you will likely become very stressed and frustrated.

5. If you are stuck on the same problem for more than an hour, take a break. I cannot count how many times I have realized a mistake, or figured out the solution to a problem while hiking or biking. If you give it some time, and still cannot figure out a problem, come see me.

6. Always remember that there are usually many ways to code up the same result. So, just getting the correct result is only part of the game. A good algorithm is mathematically correct, easy to understand, well-documented, and computationally efficient. This is how you will be graded in this course.

7. *This is the first time I have taught this course, so there will certainly be mistakes. If you are working on a problem and suspect that there is a mistake, let me know. I don't want you to get frustrated trying to do the impossible!*