SIO 113 Computation & Data Analysis in the Geosciences

Spring 2024

Instructor: Dave May (dmay@ucsd.edu)

Instructional assistant: Gabrielle Hobson (ghobson@ucsd.edu)

1. Introduction

Computers are essential to all modern Earth Science research. We use them for compiling and analyzing data, preparing illustrations like maps or data plots, writing manuscripts, and so on. In this class, you will learn to write computer programs with special applications useful to Earth Scientists. We will learn **Python**, an object-oriented programming language, and use Jupyter notebooks to write our Python programs.

2. Python

2.1 What is Python?

From wikipedia

> Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
>
> Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

The name 'Python' refers to 'Monty Python' - not the snake - and many examples in the Python documentation use jokes from the old Monty Python skits. If you have never heard of Monty Python, look it up on youtube.

2.2 Why Python?

- Easier to learn than many other languages.
- Extremely flexible and versatile.
    - Many numerical, statistical and visualization packages.

- Freely available and cross platform (works any system).
- It is well supported, actively developed and has lots of online documentation.
- It can help you improve how you conduct your science (currently), and enable new scientific enterprises to begin.
- Python programmers are in demand.

2.3 How will we use Python?

We will learn to write code in Python and learn how to develop Jupyter notebooks. A Jupyter notebook is a development environment where you can write, debug, and execute your Python programs. Notebooks allows you to create documents that contain live code, equations, visualizations and narrative text.

3. Realities of Programming

1. There is no such thing as theoretical computer programming - it is a practical skill. The only way to learn the art of programming is thus to practice it.
2. You must practice, practice, practice. Practice makes perfect.
3. Do not be deterred or discouraged by immediate failures (i.e. “code does not work”).
4. Programming mistakes (bugs) are inevitable.
5. Writing code = introducing bugs, diagnosing bugs and fixing bugs (debugging). Be prepared that 1% of your time may be spent on programming and 99% of your time may be spent debugging.
6. You should only believe code you can run.
7. There is no such thing as a perfect code. Write code for a specific design objective.
8. The learning cycle is different for everybody, as is the time-scale over which you will learn to program. Everyone can learn to program.
9. The only way to learn and improve is to practice. See 1 and 2.

4. More Realities of Programming

"Bugs are inevitable"

1. If your code works first time you run it - be suspicious. Test it again under different conditions (i.e. inputs).
2. Every line of code you write can introduce a bug.
3. The most difficult bugs to identify are often trivial to fix. Don’t give up. Persevere.
4. If you see something strange in your results there is probably a bug. Be suspicious - test everything.
5. A single bug can ruin your beautiful code. Be motivated to carefully debug and test your code. A single small error in the code cannot be ignored. Laziness never pays off.
6. Creating a correct and working code is possible. Perseverance pays off.

5. Course Format

Three lectures per week (25 lectures in total):

- Monday 10:00 - 10:50
- Wednesday 10:00 - 10:50
- Friday 10:00 - 10:50

One practical session per week:

- Friday 11:00 - 11:50

5.1 Lectures

For each lecture, there is an accompanying PDF and or Jupyter notebook. You should read and familiarize yourself with the lecture material before the class.

Every lecture will be structured as follows:

- A recap from the previous lecture, followed by a short introduction to the new material (~5-10 mins).
- In the next 30-35 mins, we adopt a flipped classroom format. You will be working the preprepared lecture material and working through a lecture specific problem set. This format is proven as a why to learn Python via "hands-on" experience.
- A final wrap-up in the last 5 minutes.

5.2 Practical Sessions

The format of the practical sessions is left intentionally open. It is your time. The time is intended for you to consolidate your understanding of programming and master your new Python skills.

The instructor and IA will be present to assist with any questions you may have regarding lecture material, lecture exercises or homework questions.

We expect you will use this time to:

- work on your exercises, homework and or the final project (see Setion 7);
- ask questions you may have regarding any lecture material, lecture specific problem sets, assignments or the final project.

No Python topic is off-limits - if you have quesions - please ask. We are here to help.

There will be weekly assignments distributed on Monday (expect in week 1).

6. Schedule

Below is a tentative schedule.

"L" = lecture; "P" = practical; "HW" = homework

Class type	Date	Topic	Action
L	01-Apr	L1: Course introduction, File systems & paths	
P	03-Apr	Getting setup with Python on your computer	
L	05-Apr	L2: Variables and operations	
P		Practical 1	
L	08-Apr	L3: Data structures	HW#1 distributed
L	10-Apr	L4: Dictionaries, loops	
L	12-Apr	L5: Functions & modules	

P		Practical 2	
L	15-Apr	L6: Numpy and matplotlib	HW#2 distributed
L	17-Apr	L7: Numpy arrays	
L	19-Apr	L8: Numpy, matplotlib, pandas	
P		Practical 3	
L	22-Apr	L9: Pandas filtering	HW#3 distributed
L	24-Apr	L10: Object oriented programming	
L	26-Apr	L11: Lambda, list & dict comprehension, exceptions	
P		Practical 4	
L	29-Apr	L12: Data wrangling with Pandas	HW#4 distributed
L	01-May	L13: More Pandas, subplots, bar charts and pie charts	
L	03-May	L14: Histograms, distributions	
P		Practical 5	
L	06-May	L15: Line and curve fitting, scikit-learn	HW#5 distributed
L	08-May	L16: Jupyter notebooks	
L	10-May	L17: cartopy	
P		Practical 6	

L	13-May	L18: geoplot & geopandas	HW#6 distributed
L	15-May	L19: Rose diagrams, projections	
L	17-May	L20: Matrix math / linear algebra, Plotting great and small circles	
P		Practical 7	
L	20-May	L21: scikit-learn: Cluster analysis	HW#7 distributed
L	22-May	L22: scikit-learn: Classification	
L	24-May	L23: SVD and principal component analysis	
P		Practical 8	
	27-May	Memorial Day observance	
L	29-May	L24: Gridding & contouring	
L	31-May	L25: 3D scatter plots and surfaces	
P		Practical 9	
P	03-Jun	Final project work	
P	05-Jun	Final project work	
P	07-Jun	Final project work	

Please note that the course material is dynamic - we improve the material over the duration of the quarter (and fix bugs).

7. Assessment

There is no final exam. The assement is progressive and distributed over the quarter. The breakdown of the assessment is as follows.

- **70% Homework assignments (7 in total)**
- **30% Final project**
    - Project proposal (ungraded).
    - Code submission.

All assessments are to be submitted via Canvas.

Detailed instructions will be provided within each homework specifying how the submission should be organized.

Please remember the rule "You should only believe code you can run it" - we will be applying this rule when grading your work. If we cannot run the Python code (or Notebook) that you have submitted you will not receive full marks. You must submit all data files, inputs required to reproduce your results. We will not chase if your submission is incomplete - this is your responsibility. Please verify that your Python submission executes without errors before uploading it into Canvas.

7.1 Problem sets

For lectures 2 through 25, there is an accompanying set of exercises you should complete. These are ungraded. The expectation is that you work on these during each lecture. The instructor and IA will assist you answering the excercises.

7.2 Homework Assignments

A homework assignment will be distributed every week on Monday.

- Each homework assignment is due by 6 PM on Monday the following week.

7.3 Final Project

The final project has two important deadlines:

- Project proposal due Friday 03-May 4 PM (end of week 5);
- Final submission due Friday 07-Jun 4 PM (before exam week).

The final project is your opportunity to be creative and showcase the Python skills you have learned during this course by applying them to an Earth science application. There is a great deal of flexibility on the exact nature of the final project but it must be related to geoscience in same way (e.g. the Earth, Atmosphere, Oceans, Space).

For example:

- Make a movie of lightning strikes across the continental US, or of volcanic eruptions across the western US, or of plate motion over the last 200 Ma;
- Make a 2D image of the solar system with orbiting planets;
- Design your own project.

As you learn new concepts, start thinking about what you'd like to create for your final project.

Whatever the specifics of your project, at the bare minimum your project must:

- be presented in the form of a Jupyter notebook;
- include at least one class;
- include at least one module with three (non-trivial) functions (or class methods) you wrote;
- read in at least one data file;
- make at least one plot;
- use at least three markdown blocks covering (i) a description of what the program does; (b) instructions on how to use the program, (c) a summary of the scientific conclusions.

You will submit the final project to Canvas as a zipped directory with all the parts required to run it (the data files, figures, modules, etc.).

In week 5, you will be asked to turn in a proposal for the final project. In the proposal, you'll describe the final project and how it relates to geocience. At that stage, you still may not have all the skills required to complete your project, but we can let you know if it is possible and substantial enough for the final project.

8. Expectations

- Read the lecture material before coming to the lecture.
- Late submissions will not be accepted unless prior approval has been sought.
- Code submissions must be complete - all required data files must be provided.
- You may consult any online resources, your fellow students & your instructor / IA to help you solve your problems. When utilizing online resources or when working in a group, please refer to Section 10 Academic Integrity (below) to ensure you give the appropriate acknowledgement and attribution to the work which you are not the sole author of.

9. Course Resources

All course material (handouts, notebookes, example code, homework) will be available on Canvas.

10. Academic Integrity

All coursework that you will produce in this unit should be original work that is yours alone. Copying work or the re-use of any material without proper acknowledgement or attribution is considered plagiarism. At UCSD, plagiarism is not acceptable.

Guidelines for giving proper acknowledgement or attribution to others work is provided below.

Group work

- If you worked in a group, you need to indicate all people involved in the group work and clearly state what your contributions were.
- If you use material from another student, you need to indicate who the student was and detail exactly what material was re-used and state clearly any modifications you made to their work.

Online resources

- If you used a tutorial in any capacity, you must provide the URL (web address) of the original material and indicate that you used an online resource as part of your submitted work. You must indicate which parts of the tutorial you used.
- If you used a script, or piece of computer code (e.g. taken from a tutorial, or a site such as StackOverflow) you must provide the URL of the original material. You must submit the original script / code in the form of an appendix with your submission. You must also indicate if you did, or did not, modify the original material. If you modified the original material, you must detail the modifications made. Care must be taken to ensure that the website and or authors copyright policy on the script / code is not violated. If in doubt please consult your instructor.
- Complete answers to any practice problem or homework assignment which are generated by ChatGPT are not considered "original work that is yours alone". You may use ChatGPT to debug you own code, and as a help resource (like you would a tutorial). You may also use ChatGPT to verify the correctness of your own code. If you do use ChatGPT in any capacity for a submission, please indicate this in your submission and provide the exact query you used.

11. Students with Disabilities

Students requesting accommodations for this course due to a disability must provide a current Authorization for Accommodation (AFA) letter issued by the Office for Students with Disabilities (OSD) which is located in University Center 202 behind Center Hall. Students are required to present their AFA letters to faculty (please make arrangements to contact the instructor privately) and to the OSD Liaison in the department in advance so that accommodations may be arranged. Contact the OSD for further information [phone] 858-534-4382; [email] osd@ucsd.edu, [website] https://disabilities.ucsd.edu.

12. Student Affairs

Throughout your time at UC San Diego, you may experience a range of issues that can negatively impact your learning. These may include physical illness, housing or food insecurity, strained relationships, loss of motivation, depression, anxiety, high levels of stress, alcohol and drug problems, interpersonal or sexual violence, or grief. Such issues may lead to diminished academic performance and affect your ability to participate in day-to-day activities. If there are issues related to coursework that are a source of particular stress or challenge, you may speak with your instructor, so that they are able to support you. UC San Diego provides several resources available all enrolled students, including:

- Counseling and Psychological Services: [phone] 858-534-3755, [website] https://caps.ucsd.edu
- Student Health Service: [phone] 858-534-3300, [website] https://studenthealth.ucsd.edu
- CARE at the Sexual Assault Resource Center: [phone] 858-534-5793, [website] https://care.ucsd.edu
- The Hub Basic Needs Center: [phone] 858-246-2632, [website] https://basicneeds.ucsd.edu

13. Copyright

All course materials (class lectures and discussions) and the intellectual content of the course itself are protected by United States Federal Copyright Law, the California Civil Code. The UC Policy 102.23 expressly prohibits students (and all other persons) from recording lectures or discussions and from distributing or selling any course materials without the prior written permission of the instructor. Students are permitted to make notes solely for their own private educational use. Exceptions to accommodate students with disabilities may be granted with appropriate documentation. See https://policy.ucop.edu/doc/2710530/PACAOS-100.