Introduction to Computational Earth Science

Authors: Lisa Tauxe, ltauxe@ucsd.edu & Hanna Asefaw, hasefaw@ucsd.edu

Computers in Earth Science

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 of manuscripts and so on. In this class you will learn computer programming with special applications useful to Earth Sciences. We will learn Python, an object-oriented programming language.

Python

So why Python?

- Flexible, freely available, cross platform
- Easier to learn than many other languages
- It has many numerical, statistical and visualization packages
- It is well supported and has lots of online documentation
- The name 'Python' refers to 'Monty Python' - not the snake and many examples in Python documentation use jokes from the old Monty Python skits. If you have never heard of Monty Python, look it up on youtube; you are in for a treat.

Which Python?

- Python is undergoing a transition from 2.7 to 3. The notebooks in this class, apart from a few exceptions, are compatible with both.
- If you are not using the university computers for this class, we recommend that you install the most recent version of Anaconda python for your operating system : https://www.anaconda.com/download/ (https://www.anaconda.com/download/)

Class Structure:

- There will usually be three lectures a week and one discussion session.
- Students are expected to read the lecture prior to attending class
- Each lecture begins with a quick review (~5 min) and proceed to the topic of the day. Lecture time will be mostly devoted to practicing the skills covered in the lecture, assuming that the students have read the lecture prior to class.
- At the end of every lecture, students will turn in their lecture (jupyter) notebooks with the practices filled in. Each student will have the opportunity to present a practice solution to the class, but will be informed of their assignment ahead of time. Lecture notebooks will count toward 25% of the final grade (approximately one point per lecture).
- There will be a programming assignment every week, due BEFORE CLASS one week from the

assignment. Assignments will count for 50% of the grade (approximately 5 points per assignment).
- Help with assignments and the solutions will be disussed during weekly discussion section, both before they are due and after they have been graded.
- In lieu of a final exam, there will be a final project - a program of your own design. There is a great deal of flexibility in the choice of what the program will do but there are some compulsory elements to it, which we will discuss in more detail later. Final projects will count for 25% of the final grade.

Class Expectations

- Attendence is strongly suggested and weekly homework assignements are mandatory as is the final project.
- Homework will not be accepted late.
- You may consult any online resources to help you solve your problem as well as your fellow students. This is encouraged. But do NOT copy verbatim what you find there. You must re-work anything through your own brain and in your own words and style or you will not learn how to program. Copying programs does not help you learn and in fact it is "cheating". Cheating will be reported to the authorities and will result in unpleasantness all around.
- The best way to learn how to program is to attend the lecture, do the practice problems and assignments and attend the discussion section where your TA can help.

Lectures

```
Out[8]:
```

	Date	Topic	Application
1	1/8/18	Intro to the class	
2	1/10/18	Variables and Operations	
3	1/12/18	Data structures	
4	1/17/18	Dictionaries, program loops (if, while and for)	
5	1/19/18	functions and modules	
6	1/22/18	NumPy and matplotlib	seismic record
7	1/24/18	NumPy arrays	
8	1/26/18	file systems and paths	
9	1/29/18	Pandas, file I/O	P-S wave arrival times
10	1/31/18	object oriented programming	objects and classes
11	2/2/18	recursions and exceptions	fibonacci spiral
12	2/5/18	lambda, map, filter reduce, list comprehension	
13	2/7/18	data wrangling with Pandas	seismic travel time plots

14	2/9/18	subplots, bar charts pie charts	elemental abundances
15	2/12/18	histograms and cumulative distribution functions	hypsometric curve
16	2/14/18	statistics 101	Univariate data
17	2/16/18	hypothesis testing t, F	
18	2/21/18	application to grain sizes	grain sizes
19	2/23/18	line and curve fitting	Bivariate data & Hubble plot
20	2/26/18	maps	spatial data; earthquake locations/ depths
21	2/28/18	gridding and contouring	IGRF
22	3/2/18	rose diagrams and equal area projections	glacial striations
23	3/5/18	matrix math - dot and cross products	poles to planes and more
24	3/7/18	plotting great and small circles	
25	3/9/18	3D plots of points and surfaces	benioff zone
26	3/12/18	Time series - periodograms	temporal data
27	3/14/18	Animations	Indian plate motion
28	3/16/18	Wrap up	
29	3/23/118	Student Presentations	