Python for Scientific Data Analysis

1. Syllabus

2. Course Topical Overview

We will cover the Basics of Python and Python Data Structures first, then we will discuss specific packages -- NumPy, SciPy, Matplotlib, and AstroPy. We will include discussion of advanced data analysis techniques -- e.g. numerical linear algebra, PCA, Markov Chain Monte Carlo methods and Bayesian inference -- and will *try* to cover Machine Learning.

Below are some more details about these topics.

Basics of Python

- Printing/variables
- Prompting, Type Conversions, Argument Passing, and Reading/Writing
- Functions
- If-Then Statements, Looping

Data Structures

- Tuples, Lists, Arrays, and Dictionaries

`d=[5,6,7]` # a list

```
dtuple=tuple(d) #now a tuple
```

```
c=np.array([[1,2,3],[4,5,6]])
```

- Slicing

```
a[3,4,5,6]; b=a[1:2]
```

- Sequence Functions, Comprehensions, and Lambda Functions

```
zipped=zip(seq1,seq2)
```

NumPy

- NumPy Arrays

```
a=np.array([np.e,np.pi],dtype='float128')
```

- Array Arithmetic and Universal Functions

```
arr = np.array([[1., 2., 3.], [4., 5., 6.]]); arr2=1/arr
```

- Array Slicing and Reshaping

```
print(arr2d[:2, 1:]); arr.reshape(2,3)
```

- More Array Operations and Array Broadcasting

```
np.vstack([newarr,newarr2]); np.tile(arr,(2,1))
```

- Basic Linear Algebra with NumPy and SciPy

```
linalg.solve(a,b)
```

- More Advanced Linear Algebra with NumPy and SciPy: SVD, PCA

SciPy

- Stats
- Optimization
- Root Finding
- Interpolation
- Signal Processing
- Hypothesis Testing (frequentist perspective)

Matplotlib

- Plotting Basics
- Subplots and Axes Configurations

- Shadings, Histograms, Contour Plots, and Images
- Other Plottable Things

AstroPy

- FITS Files and Image Display (with Matplotlib)
- Units, Constants, and Coordinates
- AstroQuery (Working with Databases)
- Simple Photometry and Spectroscopy
- Time Series Data (time permitting)

Markov Chain Monte Carlo Methods and Bayesian Inference

??? (Machine Learning?)

3. What is needed for this Class from You

- a laptop computer
- Python-3.7 or later installed on your laptop under the Anaconda Python distribution with the following packages installed: Jupyter Notebooks, NumPy, SciPy, Matplotlib, Pandas, Seaborn, AstroPy, Astroquery, Corner, and Emcee and other packages as announced. My recommendation: create a separate Anaconda Python environment for this class to avoid clashes with your normal codes. **DO NOT PROCRASTINATE WITH THE ANACONDA PYTHON INSTALLATION!**
- Zoom installed on your laptop.
- Markdown (recommended but not required)
- A Github Account (recommended but not required)

4. How The Class Will Work

- Lecture Notes - will be on Github in at least the Jupyter Notebook format (also likely in Markdown and PDF). Before class, I will upload the day's lecture notes to Github.

- Homework (80% of grade) - to be submitted in the form of Jupyter Notebooks or Python scripts (i.e. *.py) with clear and easy instructions for execution. We will have a significant amount of reserved class time to work on homework.
- Class Projects (20% of grade) - You will devise and work on a focused Python coding project relevant to your own interests that goes beyond material covered in this class. This could be original code you write yourself or analysis of data with a sophisticated pipeline/analysis package not covered in the class. The subject matter will be approved by me. You are allowed to work in groups of up to 3 for this assignment, provided that each member directly contributes an equal amount to the project.

5. Other Notes ...

- This is a (still) new course!
- Computer Setup/Anaconda Installation

MacOS - https://docs.anaconda.com/free/anaconda/install/mac-os/

Linux - https://docs.anaconda.com/free/anaconda/install/linux/

Windows - ... go buy a Mac instead

OR

https://docs.anaconda.com/free/anaconda/install/windows/

How do you know it worked???

Go into Terminal (if you are using Mac/Linux) or perhaps Anaconda powershell (if you are using Windows), type "python" (which should bring up Python prompt), and then type (on successive lines)

```
import numpy as np
print(np.random.rand(10))
import matplotlib.pyplot as plt
plt.scatter(np.random.rand(10),np.random.rand(10))
plt.show()
```

- Feedback is strongly encouraged
- Introductions ...