UCF Physics: AST 5765/4762: (Advanced) Astronomical Data Analysis

Fall 2016 Learning Python for Data Analysis

Python is an interpreted language that is used in many areas of computing, including data analysis. Python is very easy to learn, yet it is among the most powerful of all languages. Because many style mistakes are syntax errors in Python, it forces you to learn good coding habits that will carry over to other languages.

Python is also designed to run code written in other languages, like C, C++, or FORTRAN, so if something needs to be super-fast, you can write in a compiled language and then use Python for the high-level user interface. It also means that existing code, including many numerical packages, is or can easily be made available in Python.

Most of the documents below are in the class `python/doc` directory. Be prepared to read each one as that topic comes up during the course.

For our purposes, you need to know:

- How to use Python interactively to manipulate arrays of numbers
- How to do a variety of astronomy-specific tasks
- How to plot data and view images
- How to write simple programs

Basic Python:

You can run the raw Python interpreter (just type `python3`), but try the more friendly interactive interpreter called `ipython3`. Once inside Python, see the documentation below.

Most Python books focus on teaching you to program, for which see below. Books with good introductory sections include (in order from easiest to most advanced):

- *ThinkPython* by Allen B. Downey. This one is for beginners. See `http://www.greenteapress.com/thinkpython/`.
- *A Byte of Python* by C. H. Swaroop. See `http://www.swaroopch.com/notes/python/`. Intermediate level.
- *Python Tutorial* by Guido van Rossum. `http://docs.python.org/`. By the author of Python.
- *Learn to Program Using Python* by Alan Gauld, sections 1 and 2.
- *Internet Programming in Python* by Aaron Watters, Guido Van Rossum, and James C. Alhstrom.
- *Dive Into Python* by Mark Pilgrim. For programmers experienced in other languages. Available locally in `/usr/share/doc/diveintopython/README.Debian` or at diveintopython.net.

Numerical Arrays:

Images are arrays of numbers. Even photometric time series are 1D arrays. To manipulate arrays mathematically, we use the `numpy` and `scipy` packages. These are hosted at scipy.org. `Numpy` provides the basic array datatype and a moderate selection of routines to manipulate them. These are mostly enough for our purposes, but `scipy` provides a much larger suite of numerical routines. Reference documentation for these is fairly good, and is included with the packages (as is the case for all well written Python packages). Resources:

- A list of tutorials and other documents is on scipy.org under Documentation.
- A series of worked examples is on scipy.org under Cookbook.

Astronomy:

We use `ds9` to view astronomical images on screen. This is not a Python program, but a standalone utility you can get by typing "`ds9`" to a shell prompt. We use the Python `RO` package to send data to `ds9` and to control it. There is a user manual under the `ds9` Help tab, and also as a PDF in our online space. To round out our trio of astronomy specialty packages, we use `AstroPy` for things like reading and writing astronomical data in FITS format.

- *SAOImage DS9 Reference Manual*.
- *PyFITS User's Manual*.
- `ds9_python.py` file. This summarizes how to send images to `ds9` from Python, and how to read coordinates back from a mouse click in `ds9`.
- *Using Python for Interactive Data Analysis* by Perry Greenfield and Robert Jedrzejewski. This pretty much covers it, including a lot of what is missing in other sections of this handout. Copy the examples into your own space and read the tutorial while doing the examples. Note that we now use `RO`, not `numdisplay`, to access `ds9`; see `python/ds9_python.py`.

Plotting:

We use the Matplotlib package for plotting. It is a 2D graphics package with several user interfaces. We use the one called `pyplot`. It has a very nice web site (matplotlib.org), with full documentation `http://matplotlib.org/Matplotlib.pdf` (the PDF is also in the class directory) and numerous examples. Note particularly the Pyplot tutorial in the Matplotlib User Guide.

Help:

The SciPy community invites you to join its mailing lists and ask questions about Python. The lists are archived, so search first before posting (you can search without joining, and the digest format reduces email traffic if you join). Go to scipy.org and click "Mailing Lists" to learn more.