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

Fall 2018 Homework 2

Due Tuesday 10 September 2018

Work:

Become sufficiently familiar with Python to:

1. Start Python,
2. Allocate, manipulate, and get information about arrays,
3. Print the value of any cell in an array,
4. Change values in an array according to given mathematical or positional criteria (e.g., zero every 5th pixel, set all values greater than 100 to -2000, set a given rectangular sub-array to a given value, insert one array into another),
5. View images and manipulate image colormaps interactively from Python,
6. Make line plots and figures with images, write them in standard formats,
7. Use available resources to find functions, packages, and any instructions you need.

Resources:

1. See `learnpython.pdf` handout for other suggested readings.
2. Do the Matplotlib `pyplot` tutorial.
   See `https://matplotlib.org/tutorials/introductory/pyplot.html`.
3. Do sections 1 and 2 of the *Using Python for Interactive Data Analysis* tutorial. The exercises are for your edification (don't hand them in). See `Files/python/doc/pydatatut/`.
4. https://github.com/rougier/numpy-100
   The exercises are for your edification (don't hand them in).
5. See `make_program_executable.txt` for making a text program file into an executable program.
6. See `Files/python/ds9.py` for displaying and manipulating images with `ds9`.

Hand in:

Do not use loops for any problem on this assignment. The files you hand in should include a text file that provides all the commands to do the exercises, named as instructed in the `homework+coding.pdf` handout. Put images (plots, screendumps, etc.) in separate files as described in the handout. Finding the right documentation is part of the assignment!

Be sure that your main homework file is executable and runs as a program when we type its name. See lecture notes, the demo `01shell`, and the handout `make_program_executable.txt` for how to do this.

NOTE: The problems here can be done with much less than the full reading assignment above. Be sure you have *mastered* the material needed to do the problems, and then get as far as you can in the reading assignment and its exercises. You will be using all of that material, and more, in the rest of the course. This is the only time budgeted for learning Python.

1. (10 points) Write a set of Python commands that:

   (a) Create a Float64 array, *x*, of integers from 0 to 1000 (how many elements do you need?). Print the datatype of the elements and the array minimum and maximum. Remember to answer all questions raised in the assignment, such as "how many elements do you need"!).

   (b) Re-scale *x* to contain values from 0 to 2π. Print the minimum and maximum values.

   (c) Put the sine of the values in a new array, *y*.

   (d) Print the value of `y[234]`.

2. (10 points) Write a set of Python commands that:

   (a) Plot *y vs. x* from the last problem with reasonable axis labels and plot title (remember to capitalize all titles!).

   (b) Save your plot as a PNG file (no screenshots or window dumps from outside Python).

3. (10 points) Make a "ramp" array, with 101 evenly-spaced elements going from -1 to +1. "Clip" the array so that any value greater than 0.5 is set to 0.5 and any value less than -0.5 is set to -0.5, and plot it on the screen. Save a screen shot that includes both the Python and graphics windows (use `gnome-screenshot` or the same program from the panel menu). Do not save from within Python. Remember, no loops!

4. (10 points) Make a 200×300 Float64 array where each element contains its own y coordinate, starting with $y = 0$ for row 0 and ending with $y = 299$ for row 299. Display with `ds9` (see the file `Files/python/ds9.py`). When you use `print`, row 0 is the first row printed, row 1 appears below that, etc., and row 299 is last, at the bottom. When you display with `ds9`, however, row 0 is the bottom and row 299 is the top. Examine several randomly-placed elements from within Python to be sure the values match the y coordinate in the print statements. Be careful that the *final* array has the correct data type. Take a screen shot showing both your Python and `ds9` windows.

5. (10 points) Give the URLs of two web sites outside of UCF that provide free astronomical software that is written in Python. Write a paragraph about each package in your own words. Put the two paragraphs as an extended string (between sets of triple single-quotes) in your main homework file.

6. (10 points) Include a copy of your class log file in your handin.