Physics 266. Fall 2016. Homework 1

Due on 6 October, 2016

SHOW ALL WORK AND HAND IN ANY CODE YOU WROTE TO DO THE CALCULATIONS.

1. Introduction to python – working with fits files: This problem is intended to introduce you to python, if you have not already used it, and have you do some very simple operations on data in fits files.

On the course site on canvas (`canvas.ucdavis.edu`), under the "Week 2" module (select "Modules" from the navigation bar at the left side of the screen) you will find several files which may be of use to you:

- An example python program (example.py) showing some of the aspects of the language. If you haven't programmed in python before, you should look carefully at this program. You should also download the program and the associated polyfit_data.txt file, and run the program by typing (in the directory containing the program), "python example.py".
- A page with useful links to the scipy and matplotlib (the plotting library, also called pylab) web sites.
- The pyfits handbook. The pyfits library is used to read, write, and manipulate FITS files. Pyfits is no longer being actively developed since its functionality has been incorporated into astropy. Thus, if you want the functions, which have the same names in both pyfits or astropy, you can type `from astropy.io import fits`

For this homework problem, you will find 3 fits files (`bias1.fits, bias2.fits, bias3.fits`) on the course site on canvas under Files → Data

(a) Calculate the mean and standard deviation of the data in each of the 3 files.

(b) Each of the 3 files contains a 2048×2048 data array. Create a new 2048×2048 data array that is the mean of the three input data arrays. In other words, each pixel in the new array should have a data value that is the mean of the data values in that same pixel in each of the three input arrays. Calculate the mean and standard deviation of the data in this new array.

2. More python intro – working with ascii tables and plotting:

In the Files→Data area on the course site on canvas you will find two text files containing tables. These are called `1608_p60_comb_i.cat` and `1608_p60_comb_r.cat`. These represent measurements made of galaxies and stars from imaging data that were taken using the i and r filters, respectively. The files were created such that, e.g., the 12th row in one file represents measurements of the same object as the 12th row in the other file. Use the `read` function in `astropy.io.ascii` to load the data from each table.

NOTE The read function should automatically figure out that this table was created by a commonly-used program called SExtractor. You will know that it has done so successfully if the table that you create with the `read` function has columns called XWIN_IMAGE, DELTA_J2000, DTOT, etc. If instead, you get columns called col_45, etc., then the code did not recognize the format. In that case, you will have to call read in a way that tells it the proper format. For example, if you are reading from a file called `mydata.cat`, then you should type something like this:

```
rcat = read('mydata.cat',guess=False,format='sextractor')
```

(a) The column that you will used in each file is called MAG_AUTO. These represent the i and r-band magnitudes of the stars and galaxies. There are some bad points in each data set, so create cleaned versions of each catalog by selecting only rows in which *both* the r and i magnitudes have values <50 (call the new variables that contain the cleaned r and i magnitudes `rmag` and `imag`, respectively) . Then make a plot of i-band magnitude (x-axis) vs. $(r - i)$ (y-axis) where the points are blue circles.

(b) On the same figure, plot with red triangles the same two quantities but only for objects where $r < 22$.

3. First steps as Bayesians – blocks in a box: This problem is similar to a "classic" statistics problem, but we'll also look at it from a Bayesian perspective. You have a box containing $N_{\rm tot} = 9$ wooden blocks, N_B of which are blue and N_R of which are red. You will do the classic thing, which is to randomly draw one block at a time, each time replacing the block that you have drawn back into the box before picking the next block.

NOTE: For this question, it will be much less tedious if you write some code to do some of the parts, in particular (c)-(e). Write the code in python.

(a) On any given draw, what is the probability of getting a red block? Note that this won't be a number, but rather an expression.

(b) If you draw M times, with replacement, what is the probability of getting R red blocks? Here R obviously has to have a value between 0 and M. (This is the classical statistics question). Here again, just write down an expression.

(c) Remember that the likelihood is defined as $P(\text{data}|\text{model})$. If you draw six times and get two red blocks, what value of N_R maximizes the likelihood?

(d) Now we get to be Bayesians. You draw six blocks, of which two are red. Do the following two things: (i) Calculate and report the values of the *posterior* probability, i.e., $P(N_R|\text{data})$ for each value $N_R = 0, 1, 2, \ldots 9$. Here "data" refers to the fact that you got 2 red balls out of 6 draws. Assume a uniform prior on N_R, i.e., that N_R is equally likely to have any of the values between 0 and 9. Note that this posterior needs to be normalized such that

$$\sum_i P(i|\text{data}) = 1$$

(ii) Plot the results, with N_R along the x-axis and $P(N_R|\text{data})$ on the y-axis.

(e) Do the same thing as in (d), but now you have gotten 20 red blocks in 60 draws.