Problem Set #2

Due 21 Apr 2024

Note, you are submitting all your problems as Jupyter notebooks, so make your code, figures and text readable in a single file.

1) Often you have some kind of model and you want to compare the model predictions to some actual data. For this exercise, we have some simple models of how stellar populations change with time (this has two components, age and metallicity).

Basically, one takes a normalized amount of gas mass and turns it into stars with a range of mass, then uses stellar evolution isochrones to age those stars. At every timestep, the luminosity of the stars (through a given filter) is added and colors are calculated. At the same time, as supernovae go off, the mean metallicity of the next generation of stars goes up (this is called chemical enrichment). Rule of thumb, young stellar populations are blue, old ones are red. Also, metal-poor populations are blue and metal-rich ones are red.

Use the two models (`colors_low.csv` and `colors_high.csv`, both have time, g-r and r-i colors) as representative of two simple models, one for a low metallicity gas, then second for a high metallicity gas. First, plot the colors are a function of time (log years is an astronomers favorite choice of time). Notice how the colors redden with time, and how the metal-rich model is redder overall.

Now get some data, plot the SDSS galaxy data as shown in class (sdss.dat) Separate the red and blue cloud samples from the magnitude-color diagram in class (color them red and blue for clarity). Use the `np.logical_and` function and just draw by eye a dividing line to separate blue and red galaxies.

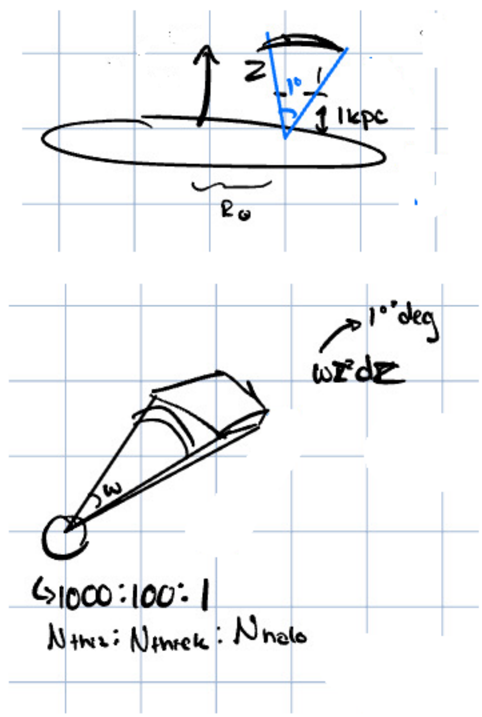
Now plot $r - i$ versus $g - r$ for the two subsamples. Set your axes so blue galaxies are in the lower left, red galaxies in the upper right (remember these are magnitudes, not luminosities). Make sure your symbols size allows you to see the depth of data.

Lastly, draw those models on top of the color-color data from SDSS. Note again, color in galaxies is a function of two effects; age and metallicity. Young objects are blue (rich in high mass, blue stars), old objects are red (rich in evolved red giant branch stars). Stars that are metal-poor are blue (less line absorbtion in the blue), metal-rich stars are red (a cooler, redder red giant branch).

Comment on the different predictions of the models, how accurate do they fit the data, would you choose one model over they other?

2) Sometimes an analytic solution is difficult to visualize, and even harder to add various components. For example, in the Milky Way, there are three components of stars; the thin disk, thick disk and the halo. The thin and thick disks have a distribution of stellar density that goes as $\rho \sim e^{-z/h}$ where z is the distance from the Sun in the Galactic North direction and h is the scale height 0.3 kpc for the thin disk, 1.0 kpc for the thick disk, see Figure below). The halo has a density distribution that goes as $\rho \sim \rho_o/r^3$, where r is measured from the Galactic Center and ρ_o is 35 stars per kpc^3. The ratio of thin to thick to halo stars at the Sun's location is 1000:100:1 (for normalization).

Assume you are observing stars in a square degree straight up out of the disk plane (i.e., z). Write a python script that calculates (and plots) the number of stars as a function of z (in intervals of 0.01 kpc) from $z = 0$ to 15 kpc for each component.



Note: you will need to account for the variation in volume element along the line of sight (see bottom panel, Figure above) as well as the variation in stellar density. If the thin disk stars are young, thick disk stars slightly older and halo stars the oldest, comment on what your distributions might mean to a scenario of galaxy formation.

3) Get the FITS file `Gaia_bLT10_pGT5.fits` from the class website. The Gaia dataset that contains proper motion and distance data for stars with parallaxes > 5 milliarcsec that also live within 10 degrees of the disk plane. Use the `Table` module from `astropy` to read the file (`from astropy.table import Table`) where `gaia=Table.read('Gaia_bLT10_pGT5.fits')` is the command you want.

There are close to 300,000 stars in that database, and if you say `gaia.colnames`, you will see the information contained in the table. The file contains:

```
ra, dec:  coordinates in right ascension and declination, both in degrees.
l,b:  coordinates in Galactic longitude and latitude, both in degrees.
parallax:  parallax in milli-arcsec
pmra,pmdec:  proper motion (milli-arcsec/yr) in ra and dec coordinates
pml,pmb:  proper motion (milli-arcsec/yr) in Galactic l and b coordinates
phot_g_mean_mag, bp_rp:  apparent Gp magnitude and Bp-Rp color (these are the Gaia filter names)
star_type:  1=main sequence, 2=red giant, 3=white dwarf, 0=other
```

And remember you access the data by saying for example `gaia['parallax']`. If you wanted the parallax data only for RGB stars, you could say

```
rgb_stars = gaia['star_type']==2
p = gaia['parallax'][rgb_stars]
```

Calculate the distance to each star from its parallax and convert apparent magnitude into absolute magnitude. Then make a color-magnitude diagram and label the key regions. Use the `hist2d` option in `matplotlib` to make a Hess diagram of the data. Comment on the stars that make up the Galactic disk.

Combine distance and proper motion in the Galactic latitude direction (b) to calculate each star's W velocity (up/down out of the plane). Use the following formula:

$$W = 4.74\,\mathrm{pmb} - D$$

where pmb is the proper motion in the Galactic latitude direction (note it needs to be converted from milli-arcsecs to arcsecs) and D is distance in parsecs. Divide the sample into 1) all MS stars, 2) all RGB stars, 3) blue MS stars ($B_p - R_p < 0.5$) and plot the W velocities. Comment on the differences and use `np.std` to determine the standard deviation of each sample. What is your guess at the astrophysical meaning of the differences.