ASTR469: Project #3, Astronomical Catalogs

Due 21 April by class time.

We will have three in-class sessions to work on this assignment.

You will spend time outside of class writing up the project and reviewing relevant literature and class materials.

Concepts practiced: *Astronomical data; Magnitudes; Extinction; Wave properties; Blackbody Radiation; Linear least squares fitting; Unix; Python; LaTeX.*

Description:

We are continuing now on your journey to becoming independent researchers in astronomy topics! This project will further increase your independence through the use of publicly available data. A fundamental data product in astronomy is catalogs. These have the advantage that the basic data reduction (imaging, making spectra, measuring fluxes) has already been performed, and basic information about all observed objects has already been extracted. The data can therefore more easily be utilized for bulk properties of a sample of objects, and importantly all researchers can use the same data in their analyses. You are going to be working with infrared point-source catalogs in this project; such catalogs contain sources that are unresolved compared to the telescope resolution, and most detected objects are stars. All of the catalogs below cover the entire sky.

Project tasks:

1. **In this project you will have to:**
   - Use one or more of the infrared catalogs listed below.
   - Form and test some hypothesis involving a least-squares fit.

   Previously, we discussed least-squares fitting of a spectral line; here you will most likely be doing linear least-squares fitting for two different properties of a set of infrared galaxies or stars. As noted in class, least-squares analysis is very common in astronomy!

2. **Consider your intended measurement and hypothesis:** Below are a few project ideas, however you are greatly encouraged to try something else, and/or to talk with me to refine an idea. For many of these you will have to restrict the magnitudes (or fluxes) for this to work properly, such that only the brightest stars/objects are shown.
   - "Reddening" is caused by dust absorption and scattering (recall our discussion in class of extinction and reddening in the atmosphere — for example the increasing redness of the setting Sun). You can see the effects of this reddening by plotting a color-color diagram of a random field using WISE and 2MASS data. A color-color diagram just takes the color index between two bands (e.g. $J-H$, $H-K$) and plots them against one another. The color indicates to you how reddened the star is (you'll have to consider here what it means to be "redder" in terms of magnitudes). Plot a color-color diagram for a direction toward the Galactic center (Galactic longitude from 10 to $-10\,^\circ$, Galactic latitude $< 1\,^\circ$) and for another field in the outer Galaxy (Galactic longitude from 170 to $190\,^\circ$, Galactic

latitude $< 1^\circ$). Fit a straight line to the diagrams and assess/comment on any differences you find.

- Determine the relationship between IRAS flux densities and/or IRAS colors of some small number of galaxies observed by IRAS ("large galaxies catalog" or "cataloged galaxies and quasars"). Compare this to that of an individual star forming region (e.g., W3, Mon R2, Sgr B2; note we used tools to check properties of individual objects in class throughout the semester, don't feel shy about doing that here). Does this tell you anything about the emission from galaxies at IRAS wavelengths?
- Determine the radial distribution of 2MASS or WISE point sources from a globular cluster (M22, M5, M13 [probably the best], M71, M4, M15, M2). You will have to determine an appropriate angular size for the search. Fit this (1D) radial distribution with a function of your choice.
  (a) Compare this profile to that of an open cluster (Pleides, M11, M67, NGC188, NGC604).
  OR
  (b) Rather than the radial fit, look for variations in properties as a function of radius, i.e. flux, color, etc?. Can you create a fit to one of these relationships?
  OR
  (c) Compare this fit to a Sersic profile, which is generally used for elliptical galaxies. You'll have to look up this functional form.
- Something else! You will be tempted to choose on of these ideas, but I will be most impressed if you go your own way.

3. **Write up your project.** Your writeup should include:
   - A brief description of the relevant background (e.g. catalog, relevant science), the hypothesis you want to test, and your expected outcome.
   - A description of your process: how you aimed to test your hypothesis, results, and conclusions. Include relevant equations and figures with captions that work to clarify your procedure and how you got to your conclusions.
   - Grades for this project will be weighted more heavily toward the rigor and accuracy of your methods (i.e. did you state clearly what measurement you wanted to make, did you describe conceptually how one could make that measurement, and did you actually make that measurement correctly?), and your conclusions (interpret the results to the best of your ability, perhaps reviewing our class notes from the first half of the semester).

Excluding figures, a suitable length for the words in this project in \documenclass{article} format would be around 2 pages. These projects may be slightly wordier than the previous projects since your hypothesis should be clear to any reader.

Below are a few procedures that will help you with data access and processing. Try all bullets below.

- **Decide what kind of analysis you might want to do.** These catalogs are huge, so you should have an idea of your project at the outset (although feel free to play around). Ask yourself: do you want to just analyze all the stars above some magnitude? Do you want to measure the properties of some particular object (if so, what is the object's name/position, and is it resolved enough to have multiple point sources in the below catalogs)?

- **Access the data.** We are going to be using the resource called "Gator," which provides access to infrared data contained in the Infrared Processing Archive Center (IPAC). The site is here:
  `http://irsa.ipac.caltech.edu/applications/Gator/`
  or, for simpler queries, here:
  `http://irsa.ipac.caltech.edu/applications/BabyGator/`

  In class we will go over how to use this site. There are three surveys that I would like you to focus on; note there are several different catalogs (galaxy, point source catalogs, etc.) in each of these surveys:

  - Two Micron All Sky Survey (2MASS) has around 470 million point sources, the vast majority of which are stars. 2MASS has three infrared bands, J, H, and K.[1] The 2MASS telescope resolution is $\sim 3''$. The table data are given in magnitudes. See information about the survey and all-sky catalog here:
    `https://irsa.ipac.caltech.edu/data/2MASS/docs/releases/allsky/doc/explsup.html`

  - Widefield Infrared Survey Explorer (WISE) has 564 million point sources, again most of which are stars. WISE has four mid-infrared bands, at 3.4, 4.6, 12, and $22\,\mu$m. The WISE resolution is $\sim 10''$ (depends on wavelength of course). The data are given in magnitudes. More information about the WISE all-sky release can be found here:
    `https://wise2.ipac.caltech.edu/docs/release/allwise/expsup/`

  - Infrared Survey (IRAS) has 246 thousand point sources. There are many stars in the IRAS catalog, but it also has many star-forming regions and some galaxies. IRAS has four infrared bands, at 12, 25, 60, and $100\,\mu$m. The IRAS resolution is $\sim 1'$ (but also depends on wavelength). The data are given in Janskys ($10^{-26}\,\mathrm{W\,m^{-2}\,Hz^{-1}}$. If you're interested there's more information about IRAS here:
    `https://irsa.ipac.caltech.edu/IRASdocs/exp.sup/`

- **Read the Data into Python**
  Python has numerous ways to read in data. One way is with Astropy: Simply save

---

[1] In class we focused on discussions using the optical Johnson Filters UBVRI; these are just infrared filter band names. $J \simeq 1.25\,\mu$m, $H \simeq 1.65\,\mu$m, $K \simeq 2.2\,\mu$m

your text file, then

```
>>> from astropy.io import ascii
>>> data = ascii.read(‘‘your filename’’)
```

At this point the “data” variable contains the entire table. You can specify the columns with: data[“column name”]
e.g.
data[“fnu_12”] is the 12 μm flux for IRAS data.

You can also import the data into a pandas data frame: `>>> import pandas as pd`

```
>>> data = pd.read_csv(‘‘your filename’’)
```

Same as before, the “data” variable contains the entire table. You can specify the columns with: data[“column name”]
e.g.
data[“fnu_12”] is the 12 μm flux for IRAS data.

- **Do a Least-squares fit.** Least squares fitting minimizes the sum of the squared residuals between the data and the fit. As we discussed in class, in a perfect fit, this sum will be zero. In a poor fit, the fit will be large. Python has a number of least squares fitting modules that a quick Google search will let you learn about.

  One method that works only for *linear* fits is using the Scipy linear regression function:
  `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html`
  Which you can use as follows:

  ```
  >> import scipy
  >> slope, intercept, r_val, p_val, std_err = scipy.stats.linregress(x, y)
  ```

  This has the benefit of giving the “R” value, and uncertainties. The `r_val` reported gives you a measure of whether the data do actually follow a linear trend (i.e. it answers the question: are these two things I’m plotting against each other actually correlated?). This is a useful value to know if you’re trying to fit a line to data that looks truly scattered. The first two figures here are revealing of how this value works:
  `https://en.wikipedia.org/wiki/Pearson_correlation_coefficient`

  Here is another way to fit if your data seem to be correlated but non-linear:
  `https://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html`
  Have a look at that function and note its inputs before moving on to my suggested usage below. Note that below I am showing a *linear* fit, whereas you might opt to do something quadratic, cubic, etc. if your data looks like it needs more complexity.
  Usage example:

  ```
  >>> import numpy.polyfit
  >>> fit, cov = polyfit(x, y, 1, cov=true)
  ```

  This fits a linear (1st order) polynomial of the form `y = fit[0] + x`$^{\mathrm{fit}[1]}$, in other words it’s the $y = mx + b$ linear equation. The `cov` variable is the covariance matrix, which tells you the wiggle room the fit outcome had in all the dimensions of the fit (in this case there are two fitting dimensions: m and b. The diagonal terms of the covariance matrix are the standard deviations of the fit parameters. For example, the `[0,0]` index is the standard deviation of `fit[0]`. The off-diagonal terms are the covariance standard deviations. When we did error propagation using partial derivatives, we assumed those terms were zero (if you don't remember this, review the online notes on error propagation, fitting, and regression!).