Plotting/Analysis tips

- When plotting mags and colors, **don't autoscale**, or you'll get unreadable plots. For CMDs, for example, reasonable limits on the magnitude range would be r = 13-24, and limits on the color range would be g-r = -1 to +2.
- When plotting lots of data points, **make the marker sizes small** so that the density of points doesn't make it so you can't see all the data. try something like `scatter(x,y,s=1)`
- Also, **to select subsamples**, an easy way to do this is to set a selection flag like this:
  - `want = (SDSS['g']<20)` *# for selecting objects with a g mag brighter than 20*

    or
  - `want = (np.abs(SDSS['redshift']-0.1)<0.05)` *# for selecting objects in the redshift range 0.005 to 0.015*

    or
  - `want = (SDSS[‘type’]==3)` *# for selecting spatially extended source (i.e., not point sources)*

  followed by (for example):

```
# plot the whole sample
scatter(SDSS['r'], SDSS['g']-SDSS['r'], s=1)
# overplot the subsample
scatter(SDSS['r'][want], SDSS['g'][want]-SDSS['r'][want],s=20,color='red')
```

*Note the **double** equal sign (‘==‘) when do an “is it equal to?’’ in a selection.*

- **You can also "stack" selections** like this:

```
bright = SDSS['g']<18 # has a g-mag brighter than 18
red = (SDSS['g']-SDSS['r']>0.7) # is redder than g-r=0.7
has_redshift = (SDSS['redshift'] != -999) # has a measured redshift
want = np.logical_and(bright,red)
want = np.logical_and(want,has_redshift)
```

which would give you a "want" selection that is bright red galaxies with measured redshifts

Pro tip: you can remind yourself of what data columns exist by saying:

`SDSS.colnames`

Astropy Data Tables

Remember that in our notebooks we are keeping the SDSS data in an astropy data table called **`SDSS`**. A data table has rows and columns.

Rows are the objects (stars, galaxies), while columns are the properties of the objects (RA, Dec, g-mag, etc).

> **`SDSS[‘g’]`** is the column of g-magnitudes for all the objects. *Something in quotes means a column!*
>
> Make new columns simply by defining them: **`SDSS[’g–r’] = SDSS[‘g’] – SDSS[‘r’]`**

Each row is all the data for a single object, which you can access by adding a bracketed number

> **`SDSS[’g’][12]`** # the g-magnitude for the 13th object in the table (remember python numbering starts at 0)
>
> **`SDSS[99]`** # show all the information for the 100th object in the table
>
> **`SDSS[‘r’][10,24,19]`** # show the the r-magnitude for the 11th, 25th, and 20th objects in the table, *in that order.*

We can access subsets of full data table using selection flags like this:

```
has_redshift = SDSS[‘redshift’] != –999
SDSS[has_redshift] # astropy data table for just the things that have a redshift.
```