- Getting set up to look at astronomical data
- Reading some file formats (FITS & ASCII)
- Fundamentals of astronomical data
- SDSS Images
- SDSS Spectroscopy
- Data distribution tools

Focus is on getting your feet wet. We won’t learn everything there is to know about:

- *astronomy, or*
- *programming, or*
- *software engineering, or*
- *the SDSS*

- Monday
  - *Getting your tools ready*
  - *Read and plot the SDSS z-catalog*
- Tuesday
  - *SDSS photometry*
  - *Compare catalog to SDSS images*
- Wednesday
  - *SDSS spectroscopy*
  - *Spectroscopic measurements*
- Thursday
  - *Data tools (CAS, API, etc)*
  - *Measure needle in haystack*

Presentation in mornings (9:30am to 12pm)

Small projects in afternoons, working in pairs (1:30pm to 4:30pm)

- Is your laptop ready?
  - *python*
  - *matplotlib*
  - *astropy*

Install latest version from astropy.org

- Do you know Unix?
  - *life will be easier if you have it*
- Can you use git?
- Do you have a text editor?
- Test case:

http://data.sdss3.org/sas/dr9/boss/spectro/redux/v5_4_45/spectra/3588/spec-3588-55184-0511.fits

```
import astropy.io.fits
import matplotlib.pyplot as plt
hdulist= astropy.io.fits.open('spec-3588-55184-0511.fits')
loglam= hdulist[1].data.field('LOGLAM')
flux=hdulist[1].data.field('FLUX')
plt.plot(loglam, flux)
plt.savefig('spectrum-plot.png', format='png')
```

- Using git.
- E.g. to get the presentations for this workshop:

```
git clone git://github.com/blanton144/sdss-workshop.git
```

- If I make changes or put examples, you can update with this command (run within the sdss-workshop directory):

```
git pull
```

- Recommendation: do all of your work in a git (or svn, or something) repository: backup and versioning

- Use sensible directory structures, especially within a git or svn repo

```
redqso/
  Makefile
  bin/
  data/
  pro/
  python/
  src/
  tex/
```

binary executables

small bits of data or results (e.g. data in figures)

software

papers

python as a command line language

```
rain.[blanton].20 % python
Enthought Python Distribution -- www.enthought.com
Version: 7.3-2 (64-bit)

Python 2.7.3 |EPD 7.3-2 (64-bit)| (default, Apr 12 2012, 11:14:05)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "credits", "demo" or "enthought" for more information.
>>> print "hello."
hello.
```

python scripts

hello.py

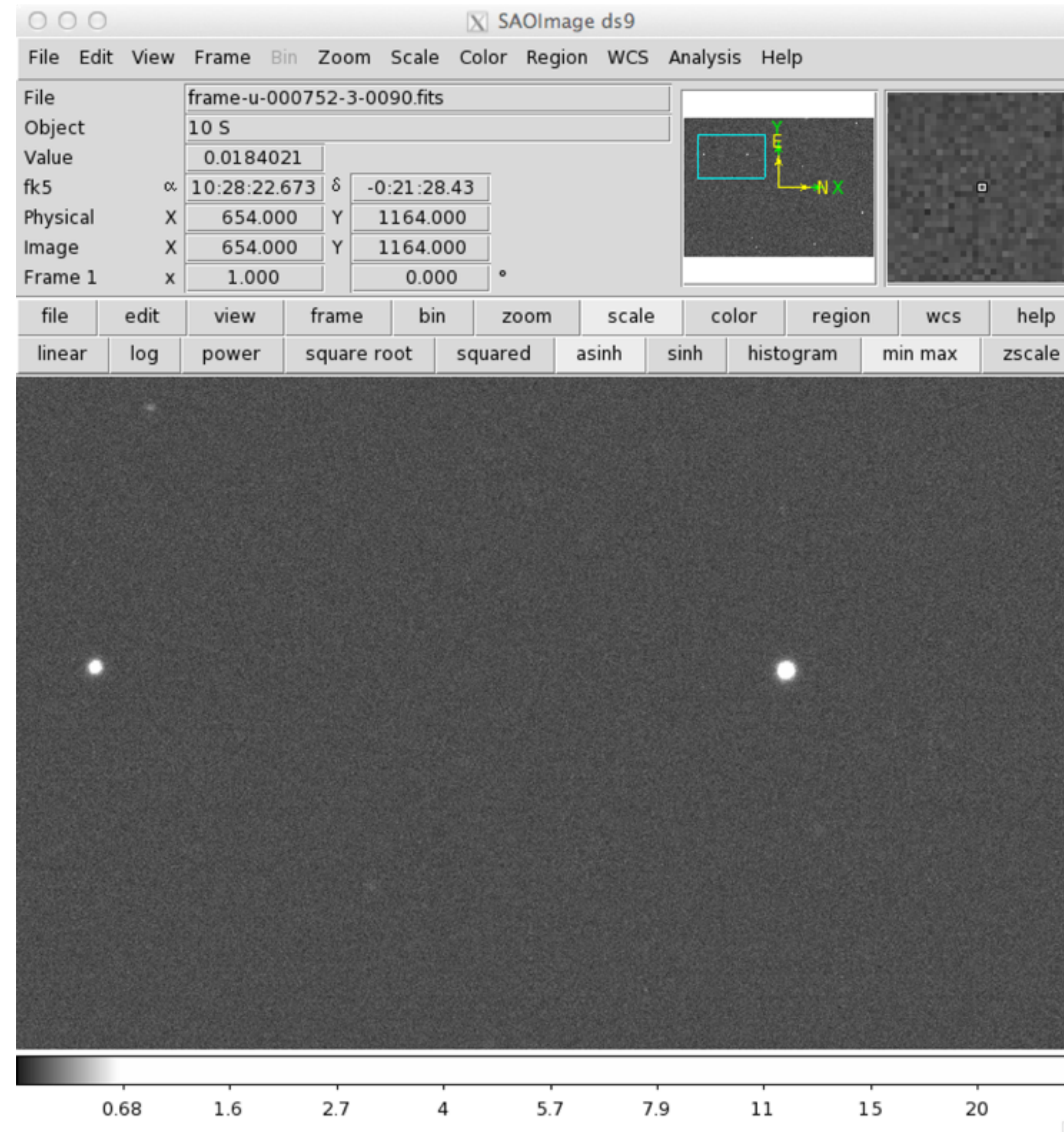
```
#!/usr/bin/python

print "hello."
```

```
rain.[blanton].23 % chmod u+x hello.py
rain.[blanton].24 % ./hello.py
hello.
```

python as a programming language: for the moment, beyond our scope

- SciCoder Material
  - *Python*
  - *FITS and Python*
  - *numpy*
  - *scipy*
  - *matplotlib*

- Looking at FITS images
- python can show them (matplotlib, aplpy)
- “ds9” is a useful stand-alone tool
  - *http://hea-www.harvard.edu/RD/ds9/site/Home.html*



http://data.sdss3.org/sas/dr9/boss/spectro/data/55325/

[sdR-r2-00114947.fit.gz]()

[sdR-r2-00114948.fit.gz]()

[sdR-r2-00114949.fit.gz]()

- Some basic basics:
  - *Right Ascension and Declination*
  - *bitmasks*
  - *Modified Julian Day (MJD)*
- I assume you already know:
  - *Redshifts*
  - *Magnitudes*

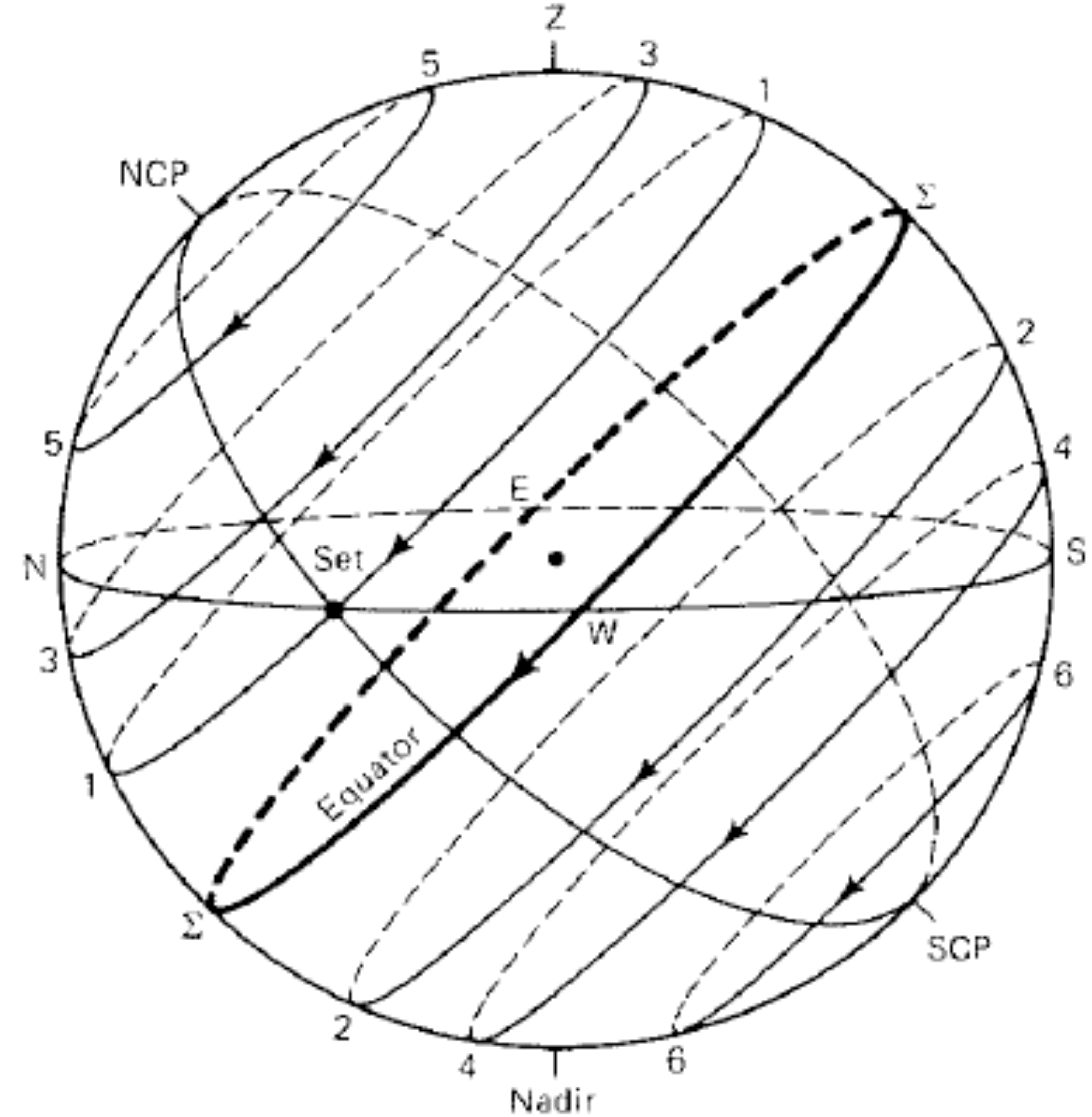
Kaler

Chapter 2 of Binney & Merrifield is a good starting point

Right ascension and declination are aligned with longitude and latitude on the Earth.

At latitude X, the zenith direction is always towards declination X

Of course, the RA of zenith changes during night and year



2.5. The rising and setting of stars. The heavy circle is the celestial equator. The daily paths of stars 1 and 2 are the same as before. The northern one rises and sets (crosses the horizon) in the northeast and northwest respectively, the southern one in the southeast and southwest. The farther north or south the star (stars 3 and 4) the closer to the north and south points are the risings and settings. If the star is closer to the pole than the pole is to the horizon (δ greater than $90° - \phi$), it (star 5) cannot set; if δ is less than $-(90° - \phi)$ (star 6) it cannot rise. A star on the equator is up for 12 hours and down for 12 hours. The more northerly the star, the longer it is above the horizon and vice versa. The hour circles of the setting point of star 1 and of the rising point of star 2 are drawn and give hour angles of about 7.5 hours west and 4.5 hours east respectively. The northern star is then up for 15 hours, the southern for only 9.

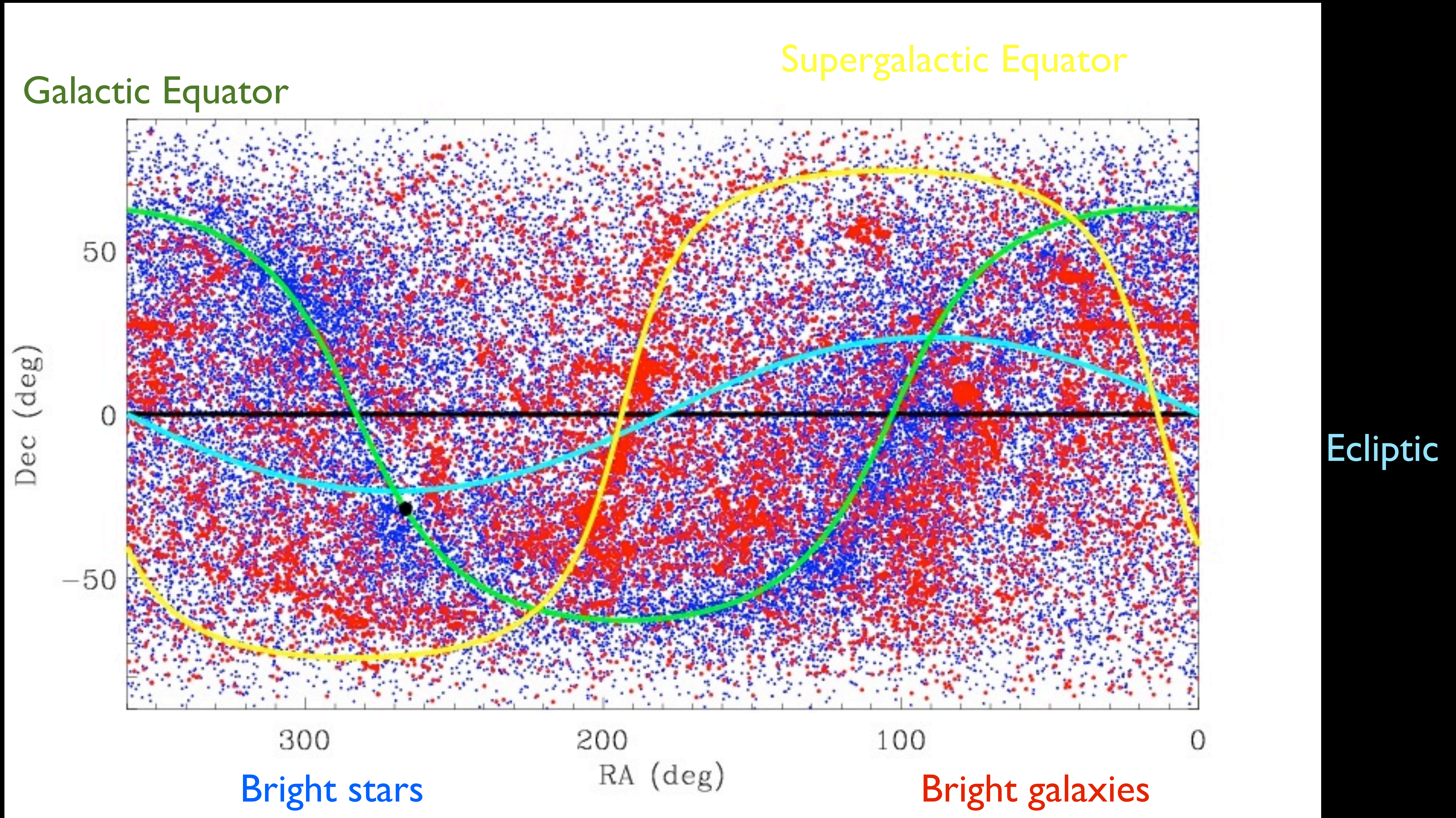
Right ascension is often expressed in terms of hours:

RA = HH:MM:SS.ss
Dec = DD:MM:SS.ss

Why? Because the Earth is turning.

Just remember that 1 hr = 15 deg and 1 deg = 4 min.

Also, because the rotation axis of the Earth precesses with a 26,000 yr period, the RA and Dec have to be referenced to a particular equinox. Currently this is using to the year 2000 (known as "J2000" coordinates)



astropy.coordinates is useful

- Bitmasks
  - *(http://sdss3.org/dr9/algorithms/bitmasks.php)*

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0		integer value
0	1	0	0	1	0	1	1	=	75
1	0	0	0	0	0	1	0	=	130
0	0	0	0	0	1	1	1	=	7

```
if((myflag & 4) != 0) {
  printf("Bit 2 is set\n");
} else {
  printf("Bit 2 is not set\n");
}
```

		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
myflag	=	0	0	0	0	1	1	0	1
4	=	0	0	0	0	0	1	0	0
myflag & 4	=	0	0	0	0	0	1	0	0

- Bitmasks

RESOLVE_STATUS

Resolve status for an SDSS catalog entry. Only one of bits RUN_PRIMARY, RUN_RAMP, RUN_OVERLAPONLY, RUN_IGNORE, and RUN_DUPLICATE can be set. RUN_EDGE can be set for any object. To get a unique set of objects across the whole survey, search for objects with SURVEY_PRIMARY set. To get a unique set of objects within a run, search for objects with RUN_PRIMARY set.

Bit name	Binary digit	Description
RUN_PRIMARY	0	primary within the objects own run (but not necessarily for the survey as a whole)
RUN_RAMP	1	in what would be the overlap area of a field, but with no neighboring field
RUN_OVERLAPONLY	2	only appears in the overlap between two fields
RUN_IGNORE	3	bright or parent object that should be ignored
RUN_EDGE	4	near lowest or highest column
RUN_DUPLICATE	5	duplicate measurement of same pixels in two different fields
SURVEY_PRIMARY	8	Primary observation within the full survey, where it appears in the primary observation of this part of the sky
SURVEY_BEST	9	Best observation within the full survey, but it does not appear in the primary observation of this part of the sky
SURVEY_SECONDARY	10	Repeat (independent) observation of an object that has a different primary or best observation
SURVEY_BADFIELD	11	In field with score=0
SURVEY_EDGE	12	Not kept as secondary because it is RUN_RAMP or RUN_EDGE object

- Modified Julian Date (MJD)
  - *(Julian Date) - 2400000.5*
  - *Start of the JD count is from 0.0 at 12 noon 1 JAN -4712 (4713 BC)*
  - *Start of MJD count is from 0.0 at 12am on 17 Nov 1858*
  - *MJD rolls over at midnight UT*
  - *Often used in lieu of calendar date, for obvious computational reasons*
  - *June 3, 2013 is MJD = 56446*
  - *MJD = 55555 is December 25, 2010*

```
from astropy.time import Time
mjds= [55555, 56446]
t= Time(mjds, format='mjd', scale='utc')
print t.iso
```

- Download the SDSS spectroscopic catalog, in flat file form:
  - *http://data.sdss3.org/sas/dr9/sdss/spectro/redux/specObj-dr9.fits*
  - *http://data.sdss3.org/datamodel/files/SPECTRO_REDUX/specObj.html*
- Read it into memory
- How many objects are there?
- Plot RA & Dec distribution
- Plot redshift distribution:
  - *of galaxies (CLASS = 'GALAXY')*
  - *of quasars (CLASS = 'QSO')*
  - *of stars (CLASS = 'STAR')*

- Take objects from the file, and look at their spectra and images, using the tools at:
    - *skyserver.sdss3.org*
    - *data.sdss3.org*
- Do this for a dozen objects each with the classifications:
    - *GALAXY*
    - *STAR*
    - *QSO*
- Just to give you a sense of what we are dealing with.