Lec15-astropy2-RyanSponzilli

October 15, 2024

1 ASTR 310 Lecture 15 - astropy 2

1.0.1 1. Reading a table and converting coordinates/times (10 points total)

- Download the file `fermi_lat_grbs.dat` from today's exercise page on the website. It contains 1000 gamma-ray bursts observed by the Fermi satellite between 2010 and 2018.
- Use the `astropy.io.ascii.read()` routine to read the file into a table. You need to specify the `header_start`, `data_start`, and `delimiter` arguments.
- Display the column headers using the `columns` or `colnames` method. Print the units on the `ra` and `dec` columns.

[3 pts]

[112]:
```
import astropy.io.ascii
import astropy.coordinates
import astropy.units as u
import numpy as np
```

[113]:
```
table = astropy.io.ascii.read("fermi_lat_grbs.dat", guess=False,␣
 ↪header_start=2, data_start=3, delimiter='|')
print(table.colnames)
print(table['ra'].unit)
print(table['dec'].unit)
```

```
['col0', 'name', 'ra', 'dec', 'trigger_time', 't90', 't90_error', 't90_start',
'fluence', 'fluence_error', 'flux_1024', 'flux_1024_error', 'flux_1024_time',
'flux_64', 'flux_64_error', 'flnc_band_ampl', 'flnc_band_ampl_pos_err',
'flnc_band_ampl_neg_err', 'flnc_band_epeak', 'flnc_band_epeak_pos_err',
'flnc_band_epeak_neg_err', 'flnc_band_alpha', 'flnc_band_alpha_pos_err',
'flnc_band_alpha_neg_err', 'flnc_band_beta', 'flnc_band_beta_pos_err',
'flnc_band_beta_neg_err', 'flnc_spectrum_start', 'flnc_spectrum_stop',
'pflx_best_fitting_model', 'pflx_best_model_redchisq',
'flnc_best_fitting_model', 'flnc_best_model_redchisq', '_1']
None
None
```

- Notice that RA and DEC are in sexagesimal format, represented as strings. Use `astropy.coordinates.Angle` to convert them to decimal degrees. Print the table or try the `show_in_notebook` method. (The method sometimes gets carried away, so you might like to restrict it to printing the first 10 rows.) [3 pts]

[114]:
```
table['ra'] = astropy.coordinates.Angle(table['ra'], unit=u.degree)
table['dec'] = astropy.coordinates.Angle(table['dec'], unit=u.degree)
table
```

[114]:
```
<Table length=1000>
 col0     name             ra          … flnc_best_model_redchisq   _1
                           deg         …
int64    str12           float64       …         float64          int64
----- ------------ ------------------- … ------------------------ -----
   -- GRB120830212  22.524666666666665 …                    0.891    --
   -- GRB140323433   23.79727777777778 …                    1.131    --
   -- GRB181007737  14.081333333333333 …                       --    --
   -- GRB100325246  13.942666666666668 …                    1.003    --
   -- GRB100507577 0.19333333333333333 …                    0.998    --
   -- GRB171206122               0.632 …                    1.001    --
   -- GRB140518709  16.269333333333332 …                    1.025    --
   -- GRB130623488   1.381527777777778 …                    1.061    --
   -- GRB110212550  20.755333333333333 …                    1.169    --
  …           …                  … …                       …     …
   -- GRB120121101  15.711333333333332 …                    0.979    --
   -- GRB170510217  10.660722222222223 …                    1.205    --
   -- GRB170802638   3.486666666666667 …                    0.993    --
   -- GRB150523690  13.484666666666666 …                    1.121    --
   -- GRB140619490  20.330666666666666 …                     1.08    --
   -- GRB170607946  16.444055555555554 …                    1.297    --
   -- GRB120926426  3.9813333333333336 …                    0.886    --
   -- GRB130627372  12.296333333333333 …                    0.894    --
   -- GRB101102840  18.978666666666665 …                    0.987    --
```

- Do some brief analysis:
  a) Construct a boolean mask array that selects only those bursts with durations $t_{90} < 2$ seconds and relative errors in the duration $\frac{t_{90\_error}}{t_{90}} < 50\%$.

  b) Construct a second boolean mask array that selects $t_{90} > 2$ seconds with the same relative error.

  c) Compare the median peak energy (flnc_band_epeak, in keV) of the two samples. Use the NumPy `median` function to compute the values. [4 pts]

[115]:
```
table['t90'].unit = u.second
table['t90_error'].unit = u.second
table['flnc_band_epeak'].unit = u.keV

mask1 = (table['t90'] < 2*u.second) & ((table['t90_error'] / table['t90']) < 0.
↪5)
```

```
mask2 = (table['t90'] > 2*u.second) & ((table['t90_error'] / table['t90']) < 0.
 ↪5)

print(np.median(table['flnc_band_epeak'][mask1].quantity))
print(np.median(table['flnc_band_epeak'][mask2].quantity))
```

```
359.3643 keV
135.48020000000002 keV
```

1.0.2 2. Working with and modifying a table (10 points total)

Download the files "mcxc.dat" and "mcxc.readme" from today's exercise page on the website. These files contain a catalog of X-ray-detected clusters of galaxies from Piffaretti et al. (2011) obtained through the VizieR service at the University of Strasbourg.

1. Use `astropy.io.ascii.read()` to read the table and its metadata into Python. This table is in "CDS" format, and you specify the metadata file using the `readme` argument.
2. Extract the log of L500 (luminosity in units of 10^{44}erg/s), log of M500 (mass in 10^{14} solar masses), and z (redshift) columns from the data into 1D arrays. Create a mask array selecting those clusters with redshift < 0.1.
3. Now construct the array r containing $r = \log L - 1.64 \log M$. The X-ray luminosity and mass of galaxy clusters are correlated roughly such that $L \propto M^{5/3}$, so the range in r is small.
4. Modify the table to add a masked column (`MaskedColumn` object) for r, and use the description "$r = \log L - 1.64 \log M$" for that column. Use the mask array you created to mask those clusters you don't want to store r values for. Print the summary information for the table, and print the table to check whether your column was added correctly. You should see "–" for masked-out rows.
5. Write the modified table to a file named "mcxc_new.csv" in comma-separated value (CSV) format.

[116]:
```
import astropy.table



t = astropy.io.ascii.read("mcxc.dat", readme="mcxc.readme", format='cds')
log_l500 = np.log10(t['L500'])
log_m500 = np.log10(t['M500'])
z = t['z']

z_mask = z < 0.1

r = log_l500 - 1.64*log_m500

t['r'] = astropy.table.MaskedColumn(r, mask=~z_mask)
t['r'].description = "r = log(L) - 1.64 * log(M)"
```

[117]:
```
t
```

[117]:

```
<Table length=1743>
    MCXC          OName       …  L500r4          r
                              …               1e+37 W
   str12          str18       … float64       float64
------------ ---------------- … ------- --------------------
J0000.1+0816 RXC J0000.1+0816 …      -- -0.49006049429220355
J0000.4-0237 RXC J0000.4-0237 …      --  -0.4908979504552985
J0001.6-1540 RXC J0001.6-1540 …      --                   --
J0001.9+1204 RXC J0001.9+1204 …      --                   --
J0003.1-0605  RXCJ0003.1-0605 …      --                   --
J0003.2-3555  RXCJ0003.2-3555 …      --  -0.4855196142700461
J0003.8+0203  RXCJ0003.8+0203 …      -- -0.46430026883136716
J0004.9+1142 RXC J0004.9+1142 …      --  -0.4723642366254167
J0005.3+1612 RXC J0005.3+1612 …      --                   --
         …                … …      …                    …
J2355.1-1500      BVH2007 242 …      --  -0.4676040941865678
J2355.6+1120 RXC J2355.6+1120 …      --   -0.474121679853096
J2355.7+1138            A2678 …      --  -0.4739568466752166
J2355.8+3423 RXC J2355.8+3423 …      --                   --
J2357.0-3445  RXCJ2357.0-3445 …      --  -0.4862703253095651
J2359.3-6042  RXCJ2359.3-6042 …      --  -0.4610630708726696
J2359.4-3418    MS2356.9-3434 …      --                   --
J2359.5-3211  RX J2359.5-3211 …      --                   --
J2359.9-3928  RXCJ2359.9-3928 …      --                   --
```

[118]:

```python
t.write("mcxc_new.csv", format='csv', overwrite=True)
```