Lec16-matplotlib1-RyanSponzilli

October 17, 2024

1 ASTR 310 Lecture 16 - matplotlib 1

1.0.1 Exercise 1: a scatter plot

- Download the file `halos.dat` from today's exercise page. This is a text file containing x, y, z positions and masses for dark matter halos in a cosmological simulation. Positions are in Mpc and masses are in $M_{\odot}$.
- Create a scatter plot of the (x, y) coordinates of the halos. Represent each halo by a circle whose size is proportional to the logarithm of the halo's mass. For example, you might try using a size given by $70[\log(M/M_{\odot}) - 11.3]$.
- Using your favorite color map, make the color of each circle proportional to the z-coordinate of the corresponding halo. If you don't like the default colors, you can change the color map using `plt.set_cmap('Spectral')` or something similar. See the gallery for examples.
- Include an appropriate title and axis labels.

[10 pts]

[136]:
```
import numpy as np
import astropy.io.ascii
import astropy.table
import matplotlib.pyplot as plt
```

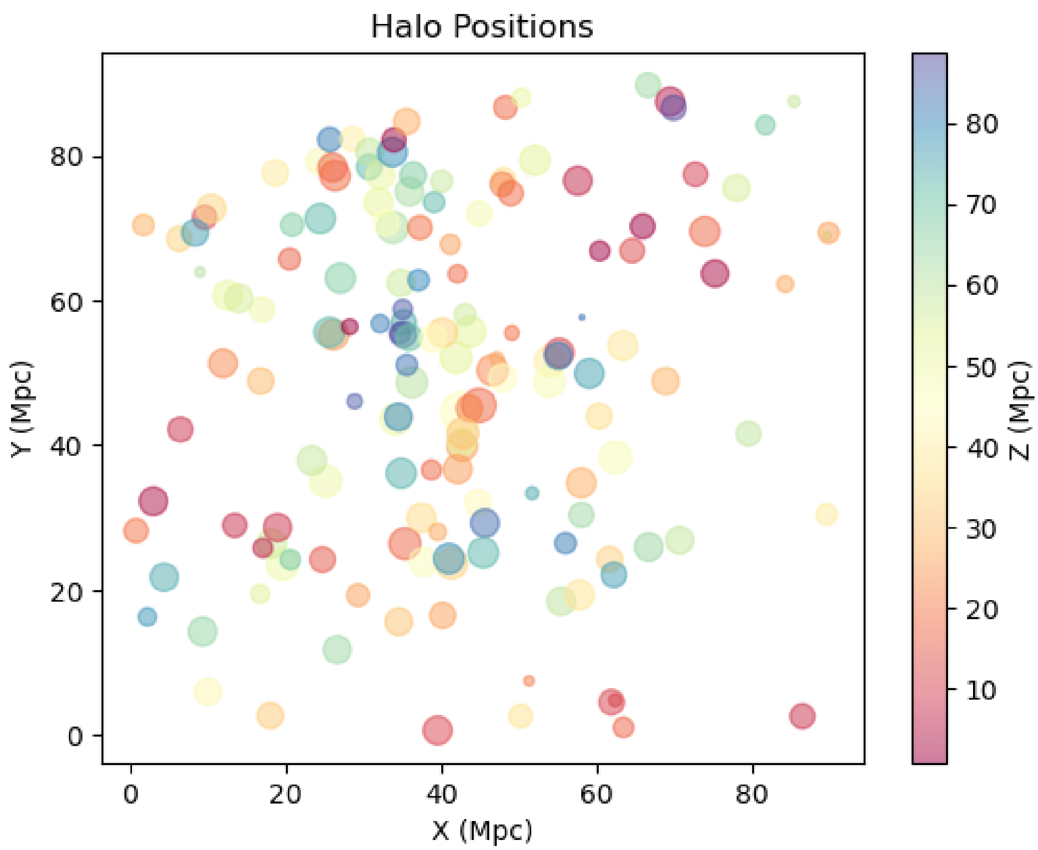
[137]:
```
table = astropy.io.ascii.read("halos.dat")
table.rename_column('col1', 'x')
table.rename_column('col2', 'y')
table.rename_column('col3', 'z')
table.rename_column('col4', 'mass')
table['size'] = 70*(np.log10(table['mass']) - 11.3)
table
```

[137]:
```
<Table length=154>
   x       y       z           mass                size
float64 float64 float64      float64             float64
------- ------- ------- ----------------- ------------------
 42.584  44.506  45.621 505300000000000.0 238.24845087667615
 46.593  50.376  22.243  16440000000000.0  134.1131269242821
  0.767  28.209  19.144   2735000000000.0  79.58701314686137
```

20.833	70.433	67.554	2111000000000.0	71.7141763315359
4.379	21.797	74.507	7180000000000.0	108.92871109696102
19.595	23.582	50.171	19390000000000.0	139.13044663550932
1.719	70.401	27.654	1515000000000.0	61.628884298682664
63.351	53.788	38.193	10720000000000.0	121.11363497497258
2.228	16.323	77.722	875600000000.0	44.96140270271255
…	…	…	…	…
79.513	41.611	59.114	3121000000000.0	83.60056380657326
77.966	75.502	55.398	5232000000000.0	99.30674147213475
45.404	25.18	72.749	13570000000000.0	128.2805893361816
81.678	84.217	68.545	1028000000000.0	49.839518026147914
84.206	62.265	26.243	677100000000.0	37.14569696581968
85.365	87.463	58.246	371600000000.0	18.905299372512285
86.43	2.603	7.778	3151000000000.0	83.89138822217987
89.603	68.929	57.235	278000000000.0	10.083135714265286
89.79	69.34	25.309	1422000000000.0	59.70297174756233
89.539	30.478	38.98	1500000000000.0	61.326388133897645

[138]:
```
plt.scatter(table['x'], table['y'], c=table['z'], s=table['size'],␣
↪cmap='Spectral', alpha=0.5)
plt.colorbar().set_label("Z (Mpc)")
plt.title("Halo Positions")
plt.xlabel("X (Mpc)")
plt.ylabel("Y (Mpc)")
```

[138]: Text(0, 0.5, 'Y (Mpc)')



1.0.2 Exercise 2: pulsars

Download the file "atnf_pulsars_20200112.dat" from the exercise page. This is a fixed-width ASCII table containing data from the Australia Telescope National Facility (ATNF) catalog of pulsars in our Galaxy known as of 1/12/20.

- Use `astropy.io.ascii` to read the table. You might need to give the reader some hints about the format or delimiters and that the header starts on line 2.
  We will make use of the pulsar periods P (column `period`), rate of change of periods $\dot{P}$ (column `period_dot`), and companion types (column `companion_type`).
- The companion type will be masked out if the pulsar is an isolated pulsar, or else it will be one of 'MS', 'NS', 'CO', 'He', or 'UL' depending on the type of binary companion. Make two boolean mask arrays to select the pulsars that are in binaries vs. not in binaries. *Hint*: use the `mask` attribute of the `companion_type` column object.

[4 pts]

[139]:
```
t2 = astropy.io.ascii.read("atnf_pulsars_20200112.dat", guess=False,␣
↪delimiter='|', header_start=2, data_start=3)
```

```
nonbinaries = t2['companion_type'].mask
binaries = ~nonbinaries
```

- Make a log-log scatter plot of $\dot{P}$ vs. P. Make binary pulsars and isolated pulsars two separate colors, and produce an appropriate legend. Make appropriate axis labels (P is in units of seconds, while $\dot{P}$ is dimensionless).
- Finally, overplot the "pulsar death line." This is (roughly) the function

$$\dot{P}_{\mathrm{death}} = 7.14 \times 10^{-18} \left(\frac{P}{\mathrm{sec}} \right)^{3.2}$$

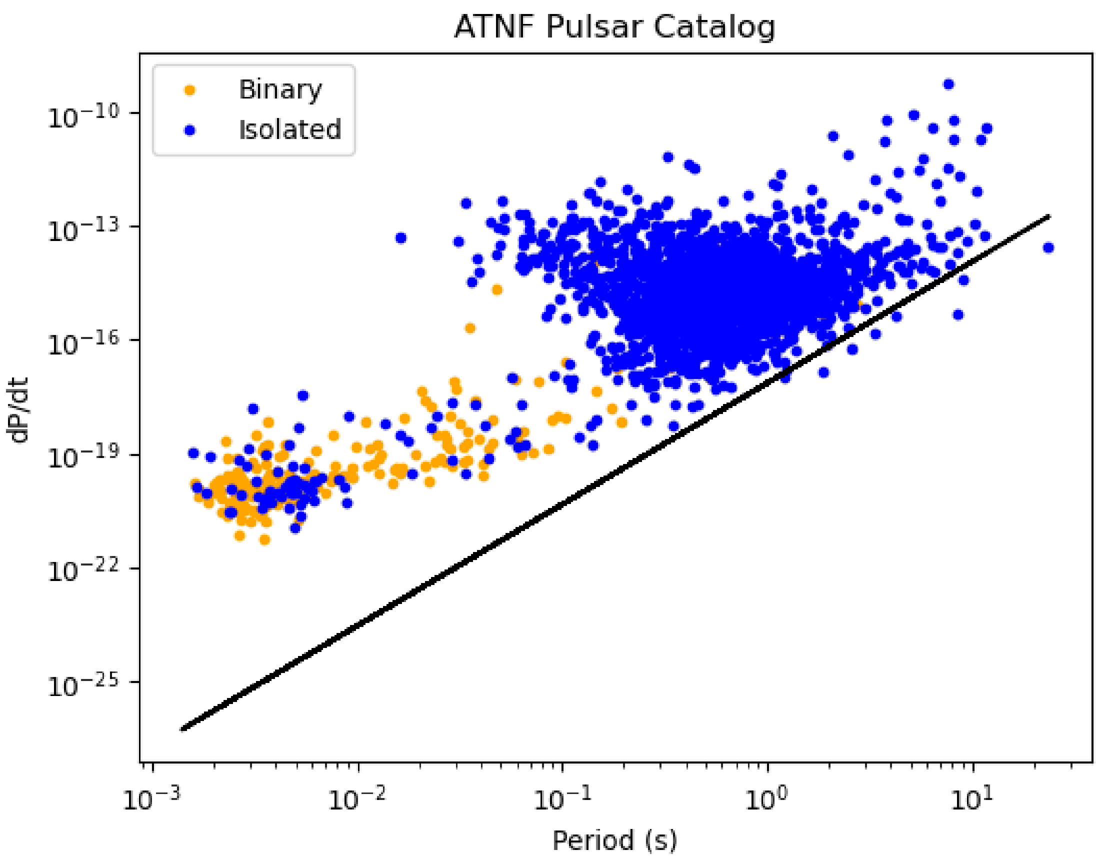
and corresponds to the line beyond which pulsars stop producing pulses. You can draw a line by defining two points on the line and connecting them.

[6 pts]

[143]:
```
plt.scatter(t2['period'][binaries], t2['period_dot'][binaries], c='orange',␣
 ↪s=10)
plt.scatter(t2['period'][nonbinaries], t2['period_dot'][nonbinaries], c='blue',␣
 ↪s = 10)
plt.xlabel("Period (s)")
plt.ylabel("dP/dt")
plt.title("ATNF Pulsar Catalog")
plt.legend(["Binary", "Isolated"])
plt.xscale("log")
plt.yscale('log')

t2['deathline'] = 7.14e-18 * (t2['period']**3.2)
plt.plot(t2['period'], t2['deathline'], c='k')
```

[143]:
```
[<matplotlib.lines.Line2D at 0x17a8ed220>]
```



[141]:
```
t2
```

[141]:
```
<Table length=2800>
 col0      name            ra       …  _1        deathline
int64     str16          str13      … int64       float64
----- -------------- ------------- … ----- ----------------------
   -- PSR J1036-8317 10 36 00.0000 …    --  9.071871444050527e-26
   -- PSR J0211-8159 02 11 59.0000 …    --  9.061867506822521e-18
   -- PSR J1900-7951 19 00 24.4000 …    -- 1.5699790581930042e-17
   -- PSR J1119-7936 11 19 40.8000 …    --  9.987428592146507e-17
   -- PSR J1057-7914 10 57 27.4000 …    -- 1.8539131226343153e-17
   -- PSR J1159-7910 11 59 35.7100 …    --  9.086664421394126e-19
   -- PSR J1841-7845 18 41 25.9000 …    -- 2.5641733786118627e-19
   -- PSR J1403-7646 14 03 04.1000 …    -- 1.6785213049829142e-17
   -- PSR J1651-7642 16 51 07.8700 …    -- 4.3214681163905236e-17
  …          …            … …   …                  …
   --    PSR J0645+80 06 46 00.0000 …    -- 1.8696177067216754e-18
   --    PSR J0100+80 01 00 18.0000 …    -- 2.5777926887357078e-17
   -- PSR J0849+8028 08 49 01.5000 …    --  3.227126604385572e-17
```

```
-- PSR J1641+8049 16 41 20.8381 …    --  1.704644272348171e-26
-- PSR J0653+8051 06 53 15.0900 …    -- 1.3295424224935145e-17
-- PSR J1942+8106 19 42 54.6800 …    -- 4.3802960516618106e-20
--   PSR J0614+83 06 14 00.0000 …    --  8.074943002832524e-18
-- PSR J1321+8323 13 21 45.6315 …    --   1.98250911635482e-18
-- PSR J2351+8533 23 51 03.2961 …    --  7.411415524568623e-18
-- PSR J1624+8643 16 24 32.7200 …    --  3.676976848406149e-19
```

[ ]: