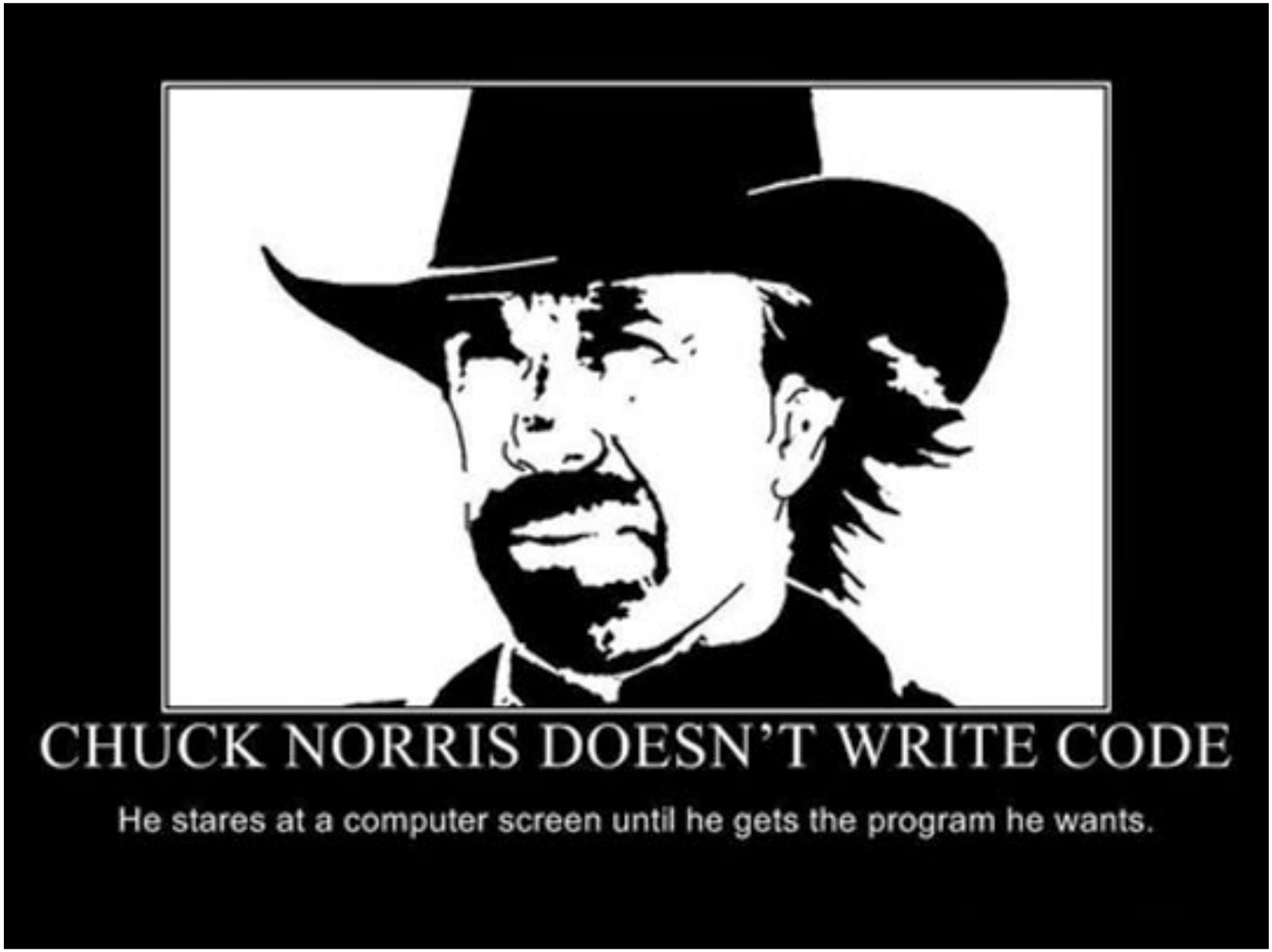
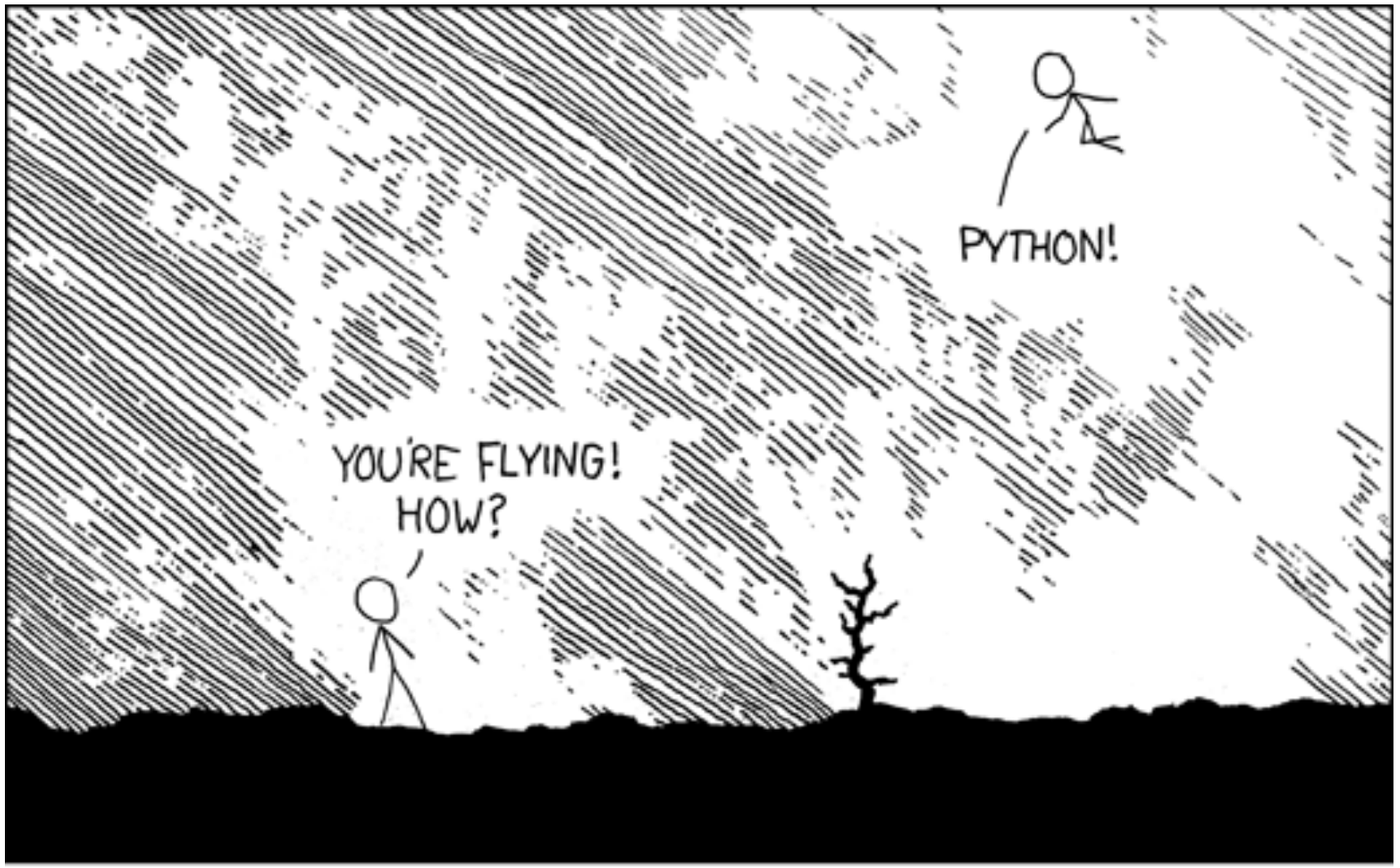
python 101

Nadia Blagorodnova
Caltech, 22nd January 2016

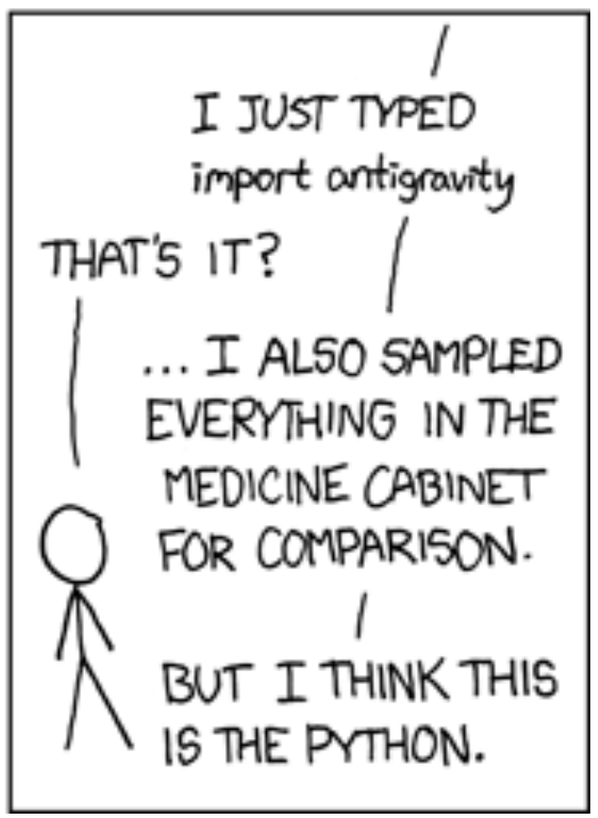
Why do we need to program?




PYTHON!
YOU'RE FLYING!
HOW?


I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"


I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?


I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Getting python on your machine

Distribution	Mac	Linux	Windows
Anaconda	Y	Y	Y
MacPorts	Y	–	–
Homebrew	Y	–	–
RPM, APT	–	Y	–
ActiveState CE	Y	Y	Y
Enthought Canopy	Y	Y	Y
STSci_Python	Y	[8]	Y
yt Project	Y	Y	–
Ureka	Y	Y	–

https://python4astronomers.github.io/installation/recommended_options.html

Basic python packages

- ipython
- numpy
- scipy
- matplotlib

>Many already included in your distribution!

adding more (third party) packages

pip install --upgrade astropy
pip install --upgrade aplpy

…or download the source code and (generally):
python setup.py install

Basic structures (I) - lists

Creating a list:

```
In [527]: l = range(10)
In [528]: l
Out[528]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
In [542]: 5*[1]
Out[542]: [1, 1, 1, 1, 1]
```

Viewing portions:

```
In [530]: l[6:]
Out[530]: [6, 7, 8, 9]
In [531]: l[6:-1]
Out[531]: [6, 7, 8]
```

Concatenating:

```
In [535]: l2 = [3,2,1]
In [536]: l3 = l + l2
In [537]: l3
Out[537]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 3, 2, 1]
```

Sorting + set:

```
In [538]: l3.sort()
In [539]: del l3[0]
Out[539]: [1, 1, 2, 2, 3, 3, 4, 5, 6, 7, 8, 9]
In [540]: set(l3)
Out[540]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Basic structures (II) - tuple

- Creating a tuple:

```
In [550]: t = "hi", 1, 0.1, 1.2e4

In [551]: t
Out[551]: ('hi', 1, 0.1, 12000.0)
In [552]: a,b,c,d = t
In [553]: print a,b,c,d
hi 1 0.1 12000.0
```

- Creating a tuple of two elements:

```
In [555]: l1 = range(0,10,1)

In [556]: l2 = range(10,20,2)

In [557]: len(l1)
Out[557]: 10

In [558]: len(l2)
Out[558]: 5

In [559]: zip(l1,l2)
Out[559]: [(0, 10), (1, 12), (2, 14), (3, 16), (4, 18)]
```

Basic structures (III) - dictionary

- Creating a dictionary:

```
In [565]: phones = {"Ana":617493843, “Ben”:7423945384}
```

- Keys/values:

```
In [566]: phones.keys()
Out[566]: ['Ben', 'Ana']
In [567]: phones.values()
Out[567]: [7423945384, 617493843]
In [571]: phones.has_key("Joe")
Out[571]: False
```

- Retrieving a value:

```
In [573]: phones["Ana"]
Out[573]: 617493843
```

- Looping over all keys:

```
In [574]: for k in phones.keys():
   .....:     print k
   .....:
Ben
Ana
```

More on python data structures!
https://docs.python.org/2/tutorial/datastructures.html

Comprehension lists

```
In [852]: S = [x**2 for x in range(10)]
```

```
In [853]: V = [2**i for i in range(13)]
```

```
In [854]: M = [x for x in S if x % 2 == 0]
```

```
In [855]: print S, V,  M
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81] [1, 2, 4, 8, 16, 32, 64, 128, 256, 512,
1024, 2048, 4096] [0, 4, 16, 36, 64]
```

Basic structures (III) - numpy array

- Create arrays with arange:

```
In [585]: import numpy as np
In [585]: a = np.arange(10)
In [586]: b = np.arange(20,30,1)
In [587]: a
Out[587]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
In [588]: b
Out[588]: array([20, 21, 22, 23, 24, 25, 26, 27, 28, 29])
```

- Create arrays with linespace:

```
In [583]: c = np.linspace(0,10,4)

In [584]: c
Out[584]: array([  0.        ,   3.33333333,   6.66666667,  10.        ])
```

- Create arrays with repeat, ones and zeroes!

```
In [598]: np.repeat(1,4)
Out[598]: array([1, 1, 1, 1])

In [599]: np.zeros(3)
Out[599]: array([ 0.,  0.,  0.])

In [600]: np.zeros(5)
Out[600]: array([ 0.,  0.,  0.,  0.,  0.])

In [601]: np.ones(5)
Out[601]: array([ 1.,  1.,  1.,  1.,  1.])
```

Random module

```
In [596]: r = np.random.rand(5)

In [597]: r
Out[597]: array([ 0.17549771,  0.21230048,  0.14701362,  0.66392852,
0.04367731])
```

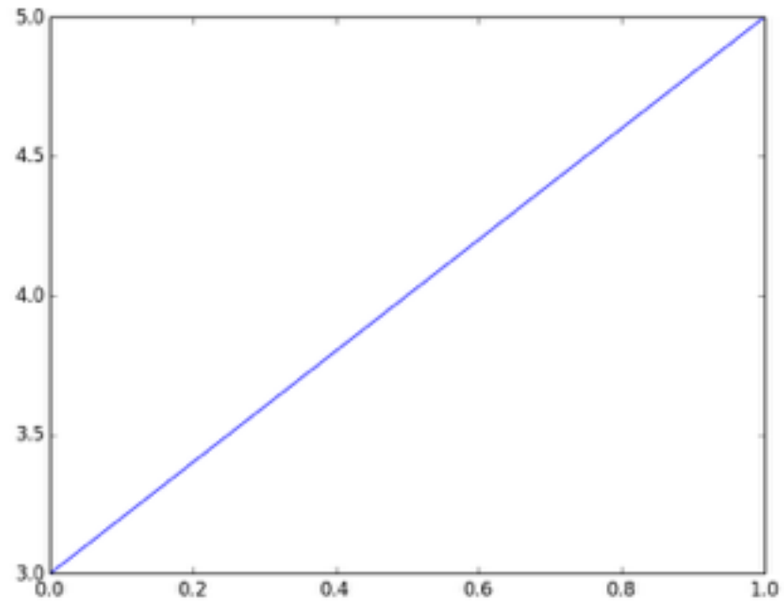
- Using different distributions: uniform & normal:

```
In [603]: r = np.random.rand(100)

In [604]: n = np.random.norm(0,1,100)
```

Plotting - plot & scatter

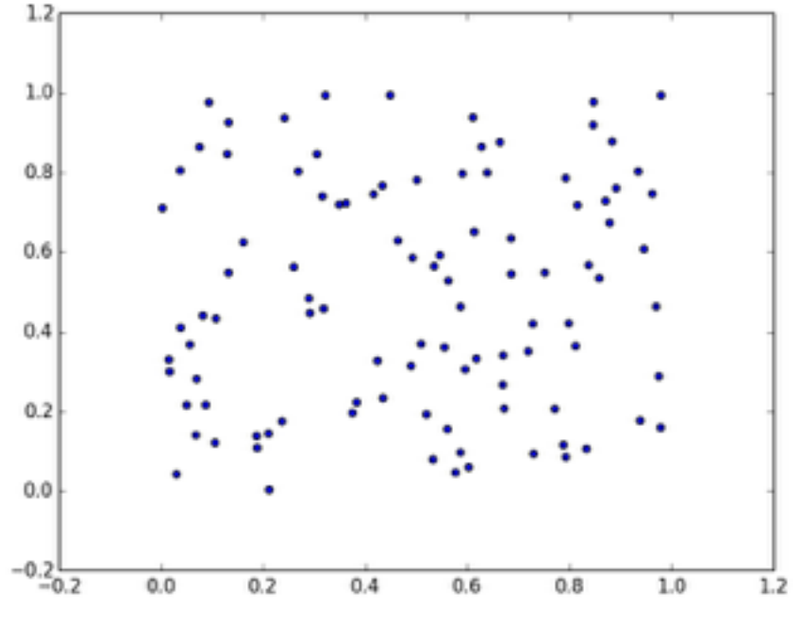
- Simple line plot:

```
In [642]: x = np.linspace(0,1,10)

In [643]: y = 3 + 2*x

In [644]: plt.plot(x,y)
Out[644]: [<matplotlib.lines.Line2D at 0x18c35e510>]

In [645]: plt.show()
```



- Simple scatter plot:

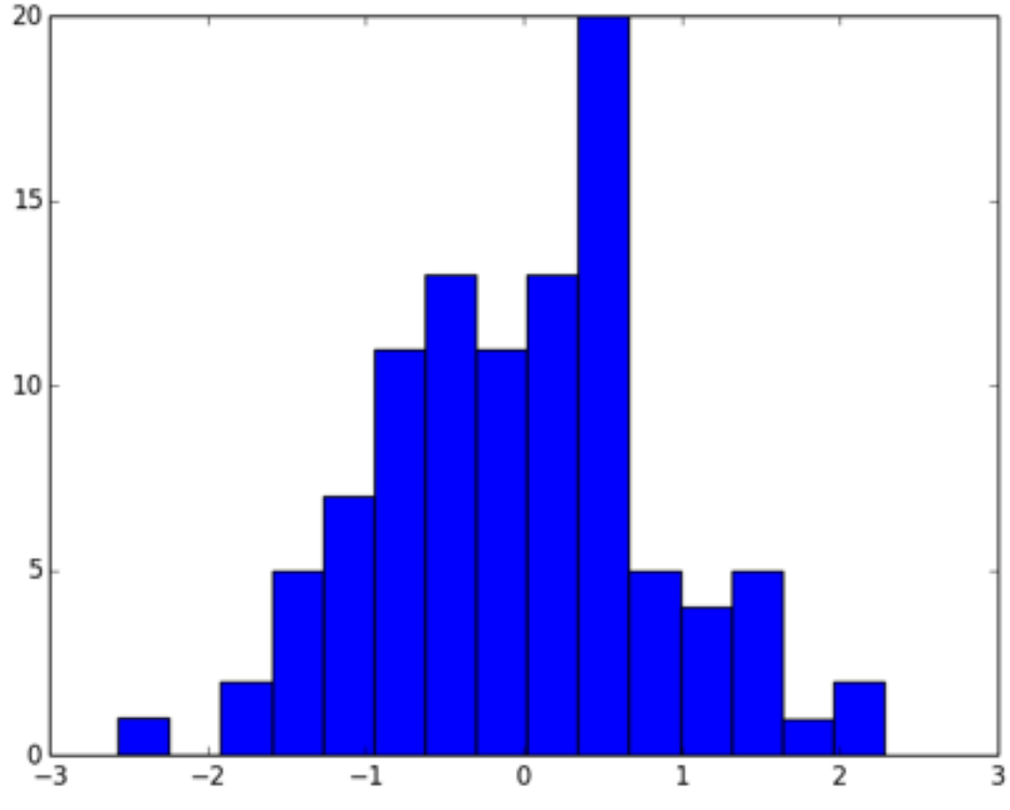
```
In [651]: x = np.random.rand(100)

In [652]: y = np.random.rand(100)

In [653]: plt.scatter(x,y)
Out[653]: <matplotlib.collections.PathCollection at 0x1

In [654]: plt.show()
```



Plotting - histogram

- Creating a 1D histogram:

```
In [660]: x = np.random.normal(0,1,100)

In [661]: plt.hist(x, bins=15)
Out[661]:
(array([  1.,   0.,   2.,   5.,   7.,  11.,  13.,  11.,  13.,  20.,   5.,
          4.,   5.,   1.,   2.]),
 array([-2.56514229, -2.24189882, -1.91865536, -1.59541189, -1.27216843,
        -0.94892496, -0.6256815 , -0.30243803,  0.02080543,  0.3440489 ,
         0.66729236,  0.99053583,  1.31377929,  1.63702276,  1.96026622,
         2.28350969]),
 <a list of 15 Patch objects>)

In [662]: plt.show()
```



Plotting - 2d histogram

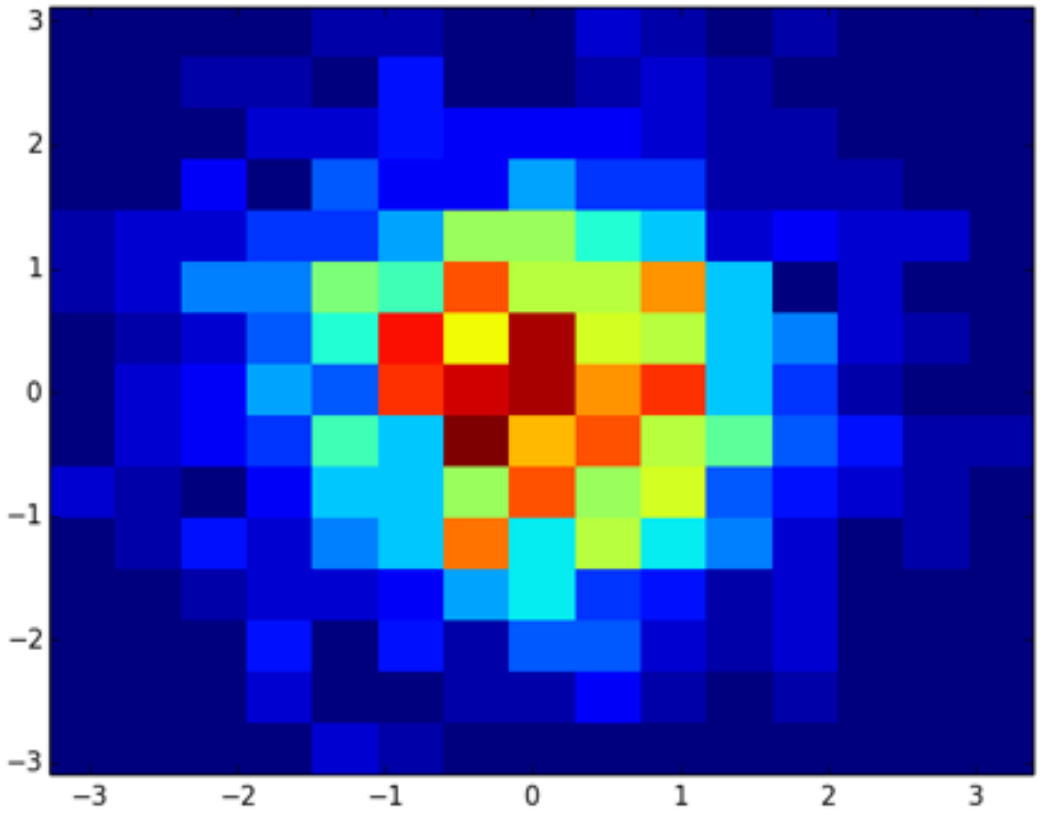
- Creating the data:

```
In [676]: x = np.random.normal(0,1,1000)

In [677]: y = np.random.normal(0,1,1000)
```

- Plotting:

```
In [680]: h = plt.hist2d(x,y, bins=15)

In [681]: plt.show()
```

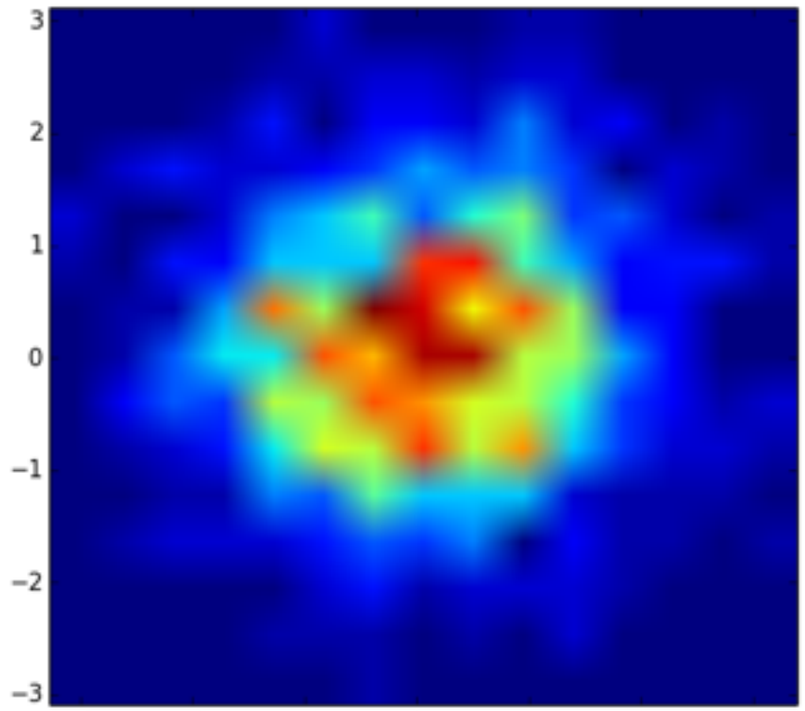
```
In [684]: h[0].shape
Out[684]: (15, 15)

In [685]: h[1].shape
Out[685]: (16,)
```



Visualising 2d arrays

- 2d visualisation of 2D matrix:

```
In [687]: plt.imshow(h[0], extent=(np.min(h[1]),
np.max(h[1]), np.min(h[2]), np.max(h[2]) ) )
Out[687]: <matplotlib.image.AxesImage at 0x1a8a551d0>
In [688]: plt.show()
```

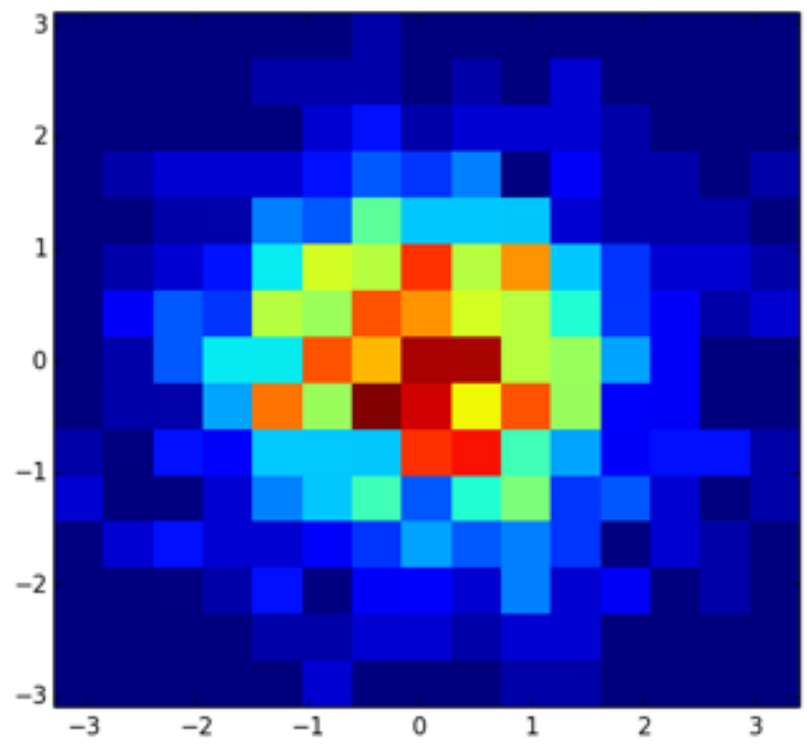


- Adjusting some parameters

```
In [691]: plt.imshow(h[0].T, extent=(np.min(h[1]),
np.max(h[1]), np.min(h[2]), np.max(h[2]) ),
origin='lower',  interpolation="none")
Out[691]: <matplotlib.image.AxesImage at 0x1a92adcd0>
In [692]: plt.show()
```

- Transposing:

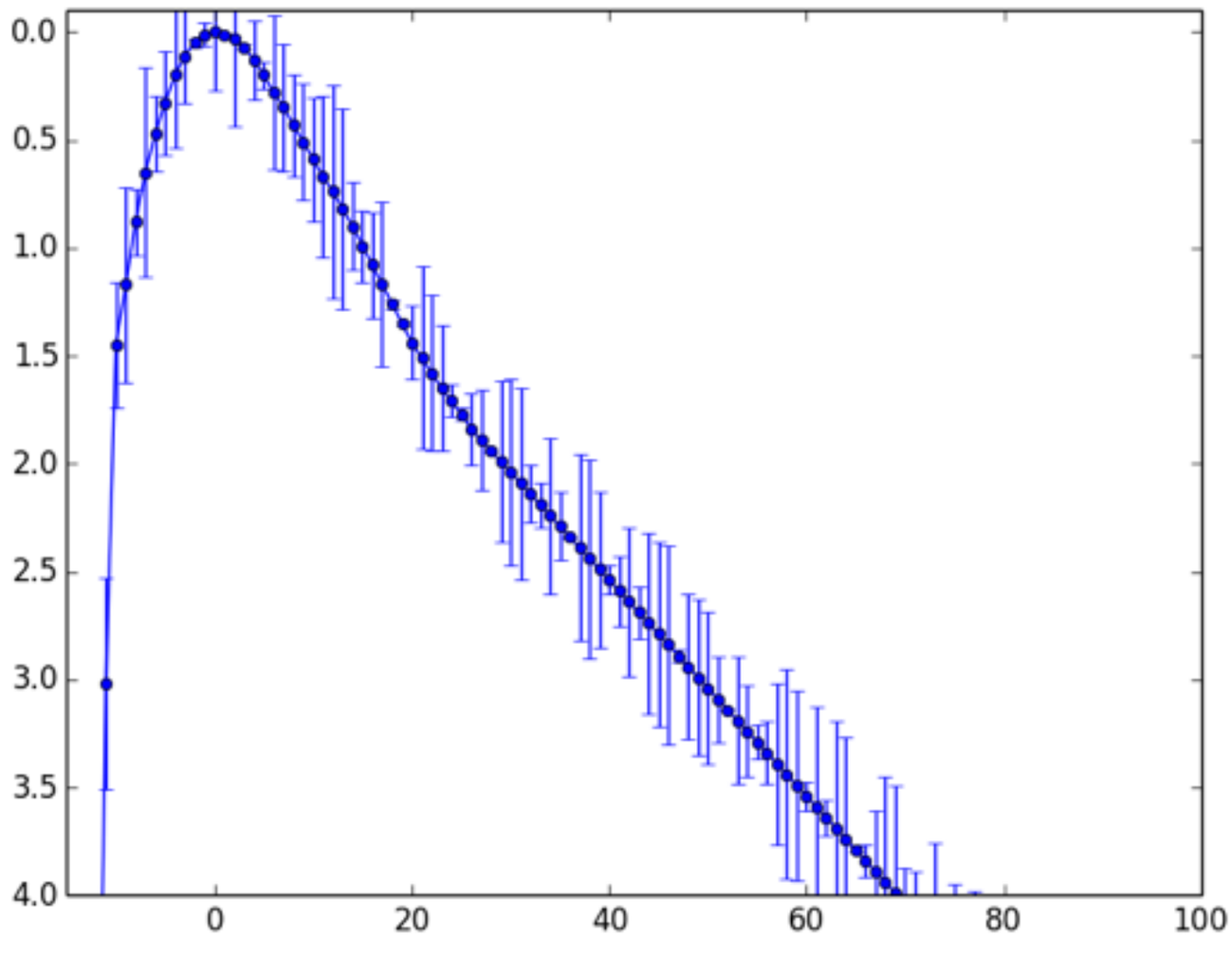
```
In [691]: plt.imshow(h[0].T, extent=(np.min(h[1]),
np.max(h[1]), np.min(h[2]), np.max(h[2]) ),
origin='lower',  interpolation="none")
Out[691]: <matplotlib.image.AxesImage at 0x1a92adcd0>
In [692]: plt.show()
```



Wide range of examples!
http://matplotlib.org/gallery.html

Plot with error bars

```
In [28]: plt.errorbar(t["t"], t["Ia01"], yerr=0.5*np.random.rand(len(t["t"])),
marker="o", markersize=5)
Out[28]: <Container object of 3 artists>

In [29]: plt.ylim(-0.1, 4)
Out[29]: (-0.1, 4)

In [30]: plt.xlim(-15,100)
Out[30]: (-15, 100)

In [31]: plt.gca().invert_yaxis()

In [32]: plt.show()
```



Read-write text files

- Simple reading:

>>> f = open(“myfile.txt”)

>>> f.readline()

'This is the first line of the file.\n'

>>> f.readline()

'Second line of the file\n'

>>> f.readline()

‘’

- Write into the file:

f.write('This is a test\n')

Read-Write files with bumpy

```
In [774]: a = np.random.rand(100,20)
In [775]: np.savetxt("afiles.txt", a, fmt="%.4f")
In [776]: !head afiles.txt
0.6842 0.6230 0.6502 0.5345 0.9231 0.8113 0.4722 0.3676 0.3579 0.5632 0.2795 0.6899 0.1953 0.4386 0.8165
0.9060 0.9265 0.7979 0.9923 0.8408
0.8122 0.1068 0.5586 0.0152 0.4537 0.5150 0.1251 0.2294 0.5752 0.7959 0.6981 0.4607 0.2639 0.2015 0.1509
0.2989 0.8676 0.2874 0.2990 0.8379
0.7895 0.4547 0.9783 0.6174 0.8935 0.8999 0.5449 0.8822 0.2617 0.3804 0.4811 0.1550 0.7510 0.3349 0.6661
0.8966 0.7543 0.0269 0.0866 0.2956
0.3585 0.8565 0.5698 0.3823 0.1994 0.3323 0.5822 0.6549 0.8974 0.8267 0.9773 0.1303 0.5781 0.9498 0.8000
0.7365 0.1053 0.6306 0.9691 0.4925
0.9960 0.4525 0.3093 0.4706 0.8481 0.4256 0.2313 0.0002 0.6143 0.4288 0.2234 0.7273 0.9232 0.7253 0.4777
0.8721 0.2123 0.0216 0.9878 0.1468
0.7418 0.9010 0.1175 0.4356 0.0711 0.8349 0.5789 0.2117 0.1179 0.5412 0.2255 0.6813 0.8148 0.0029 0.0192
0.8643 0.6133 0.7319 0.8732 0.5213
0.8207 0.5646 0.1381 0.9785 0.0530 0.4352 0.8516 0.0067 0.3660 0.1594 0.1099 0.1099 0.8897 0.5403 0.6525
0.5392 0.6145 0.6104 0.0264 0.2695
0.8458 0.9927 0.5094 0.6141 0.6677 0.9605 0.6974 0.7914 0.4803 0.8598 0.0314 0.2633 0.0301 0.6486 0.2594
0.3004 0.7752 0.1286 0.8656 0.4295
0.4658 0.7521 0.2370 0.3236 0.7577 0.6969 0.6303 0.5935 0.4391 0.3268 0.1223 0.6844 0.0665 0.5837 0.2059
0.2625 0.3278 0.1349 0.6113 0.4084
0.4253 0.6639 0.9311 0.4422 0.0458 0.2845 0.3937 0.5348 0.1794 0.2933 0.3356 0.5348 0.1298 0.3075 0.8994
0.9720 0.4812 0.8874 0.5851 0.4995

In [777]: b = np.loadtxt("afiles.txt")
In [778]: b
Out[778]:
array([[ 0.6842,  0.623 ,  0.6502, ...,  0.7979,  0.9923,  0.8408],
       [ 0.8122,  0.1068,  0.5586, ...,  0.2874,  0.299 ,  0.8379],
       [ 0.7895,  0.4547,  0.9783, ...,  0.0269,  0.0866,  0.2956],
       ...,
       [ 0.836 ,  0.6438,  0.2801, ...,  0.8968,  0.391 ,  0.6056],
       [ 0.7177,  0.7613,  0.5227, ...,  0.9654,  0.4554,  0.783 ],
       [ 0.2941,  0.1584,  0.2628, ...,  0.9219,  0.7274,  0.1068]])
```

Read numpy arrays

- Read a table ()

```
In [779]: t = np.genfromtxt(“table_snlc.dat", dtype=None, names=True)
In [784]: t[0:3]
Out[784]:
array([ (-30.0, 57.53, 56.23, 54.93, 53.64, 52.34, 51.04, 49.74, 48.44, 47.14, 45.84, 44.54, 43.24, 41.94, 40.64,
39.34, 38.04, 36.75, 35.45, 34.15, 32.85, 31.55, 3.84, 10.19, 10.87, 5.09, 10.19, '----', '----', '----', '----',
2.96, '----'),
       (-29.0, 54.66, 53.41, 52.16, 50.91, 49.66, 48.41, 47.16, 45.91, 44.66, 43.41, 42.16, 40.91, 39.66, 38.41,
37.16, 35.91, 34.66, 33.41, 32.16, 30.91, 29.66, 3.66, 9.58, 10.45, 4.78, 9.58, '----', '----', '----', '----', 2.83,
'----'),
       (-28.0, 51.8, 50.59, 49.39, 48.19, 46.99, 45.79, 44.59, 43.39, 42.19, 40.99, 39.79, 38.59, 37.39, 36.18,
34.98, 33.78, 32.58, 31.38, 30.18, 28.98, 27.78, 3.49, 8.97, 10.03, 4.47, 8.97, '----', '----', '----', '----', 2.69,
'----')],
      dtype=[('t', '<f8'), ('Ia01', '<f8'), ('Ia02', '<f8'), ('Ia03', '<f8'), ('Ia04', '<f8'), ('Ia05', '<f8'),
('Ia06', '<f8'), ('Ia07', '<f8'), ('Ia08', '<f8'), ('Ia09', '<f8'), ('Ia10', '<f8'), ('Ia11', '<f8'), ('Ia12',
'<f8'), ('Ia13', '<f8'), ('Ia14', '<f8'), ('Ia15', '<f8'), ('Ia16', '<f8'), ('Ia17', '<f8'), ('Ia18', '<f8'),
('Ia19', '<f8'), ('Ia20', '<f8'), ('Ia21', '<f8'), ('Ia05hk', '<f8'), ('Ibcave', '<f8'), ('Ibcfast', '<f8'),
('Ibcslow', '<f8'), ('Ibc05E', '<f8'), ('IIP', 'S5'), ('IIL', 'S5'), ('IIb', 'S5'), ('IInave', 'S5'), ('IInfast',
'<f8'), ('IInslow', ‘S5')])
In [818]: t = np.genfromtxt("/Users/nadiablago/Documents/workspace/TDE/data/sn/J_MNRAS_412_1441/table_snlc.dat",
dtype=None, names=True, missing_values="----")
```

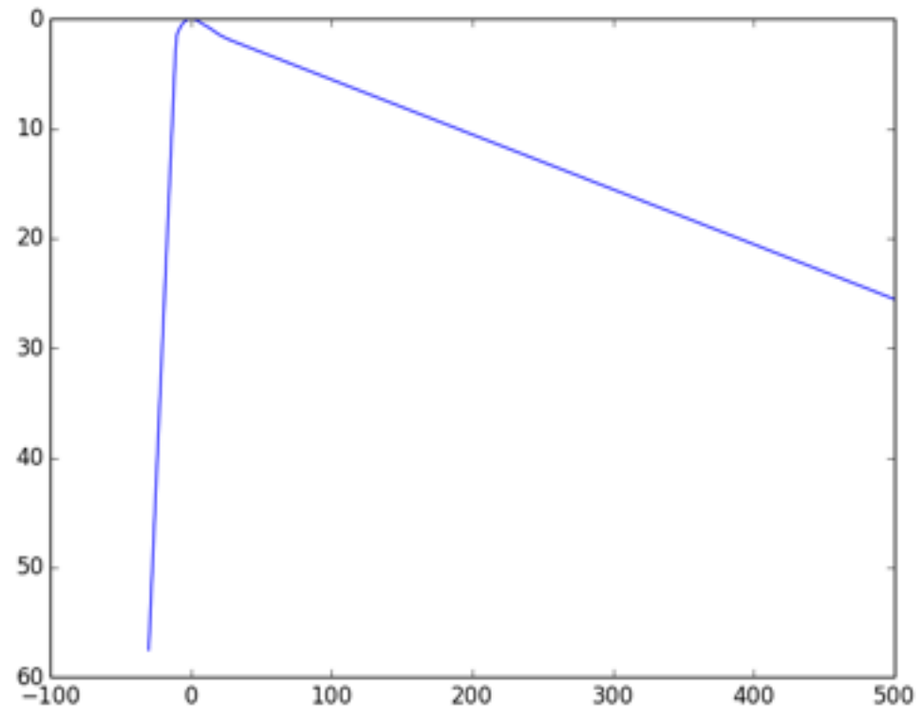
- With missing values

```
In [819]: t[0:3]
Out[819]:
array([ (-30.0, 57.53, 56.23, 54.93, 53.64, 52.34, 51.04, 49.74, 48.44, 47.14, 45.84, 44.54, 43.24, 41.94, 40.64,
39.34, 38.04, 36.75, 35.45, 34.15, 32.85, 31.55, 3.84, 10.19, 10.87, 5.09, 10.19, nan, nan, nan, nan, 2.96, nan),
       (-29.0, 54.66, 53.41, 52.16, 50.91, 49.66, 48.41, 47.16, 45.91, 44.66, 43.41, 42.16, 40.91, 39.66, 38.41,
37.16, 35.91, 34.66, 33.41, 32.16, 30.91, 29.66, 3.66, 9.58, 10.45, 4.78, 9.58, nan, nan, nan, nan, 2.83, nan),
       (-28.0, 51.8, 50.59, 49.39, 48.19, 46.99, 45.79, 44.59, 43.39, 42.19, 40.99, 39.79, 38.59, 37.39, 36.18,
34.98, 33.78, 32.58, 31.38, 30.18, 28.98, 27.78, 3.49, 8.97, 10.03, 4.47, 8.97, nan, nan, nan, nan, 2.69, nan)],
      dtype=[('t', '<f8'), ('Ia01', '<f8'), ('Ia02', '<f8'), ('Ia03', '<f8'), ('Ia04', '<f8'), ('Ia05', '<f8'),
('Ia06', '<f8'), ('Ia07', '<f8'), ('Ia08', '<f8'), ('Ia09', '<f8'), ('Ia10', '<f8'), ('Ia11', '<f8'), ('Ia12',
'<f8'), ('Ia13', '<f8'), ('Ia14', '<f8'), ('Ia15', '<f8'), ('Ia16', '<f8'), ('Ia17', '<f8'), ('Ia18', '<f8'),
('Ia19', '<f8'), ('Ia20', '<f8'), ('Ia21', '<f8'), ('Ia05hk', '<f8'), ('Ibcave', '<f8'), ('Ibcfast', '<f8'),
('Ibcslow', '<f8'), ('Ibc05E', '<f8'), ('IIP', '<f8'), ('IIL', '<f8'), ('IIb', '<f8'), ('IInave', '<f8'), ('IInfast',
'<f8'), ('IInslow', '<f8')])
```

Write numpy array

- Plot

```
In [787]: plt.plot(t["t"], t["Ia01"])
Out[787]: [<matplotlib.lines.Line2D at 0x1d4351b50>]
In [788]: plt.gca().invert_yaxis()
In [789]: plt.show()
```



- Write

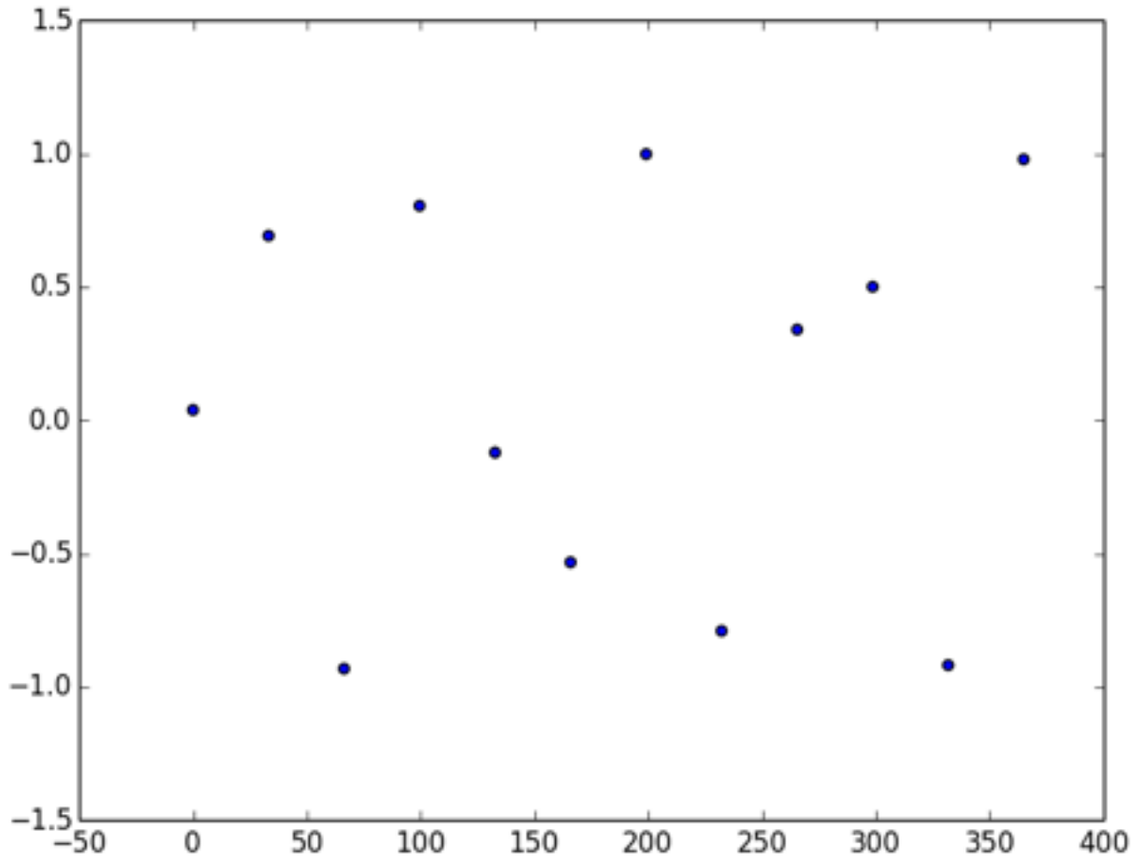
```
In [832]: np.savetxt("saved_lc.txt", t, fmt="%.4f",
header=str(t.dtype.names).replace("'","").replace(",",
```

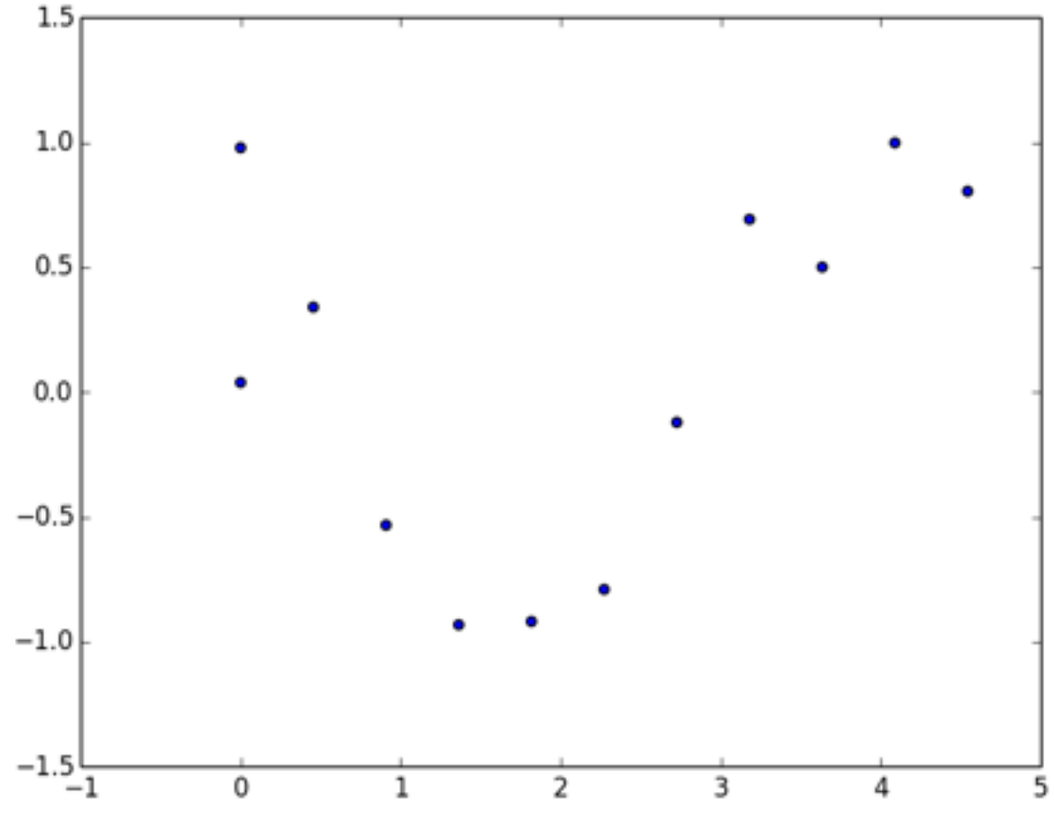
```
In [833]: !head saved_lc.txt
# (t Ia01 Ia02 Ia03 Ia04 Ia05 Ia06 Ia07 Ia08 Ia09 Ia10 Ia11 Ia12 Ia13 Ia14 Ia15
Ia16 Ia17 Ia18 Ia19 Ia20 Ia21 Ia05hk Ibcave Ibcfast Ibcslow Ibc05E IIP IIL IIb
IInave IInfast IInslow)
-30.0000 57.5300 56.2300 54.9300 53.6400 52.3400 51.0400 49.7400 48.4400 47.1400
45.8400 44.5400 43.2400 41.9400 40.6400 39.3400 38.0400 36.7500 35.4500 34.1500
32.8500 31.5500 3.8400 10.1900 10.8700 5.0900 10.1900 nan nan nan nan 2.9600 nan
-29.0000 54.6600 53.4100 52.1600 50.9100 49.6600 48.4100 47.1600 45.9100 44.6600
43.4100 42.1600 40.9100 39.6600 38.4100 37.1600 35.9100 34.6600 33.4100 32.1600
30.9100 29.6600 3.6600 9.5800 10.4500 4.7800 9.5800 nan nan nan nan 2.8300 nan
-28.0000 51.8000 50.5900 49.3900 48.1900 46.9900 45.7900 44.5900 43.3900 42.1900
40.9900 39.7900 38.5900 37.3900 36.1800 34.9800 33.7800 32.5800 31.3800 30.1800
28.9800 27.7800 3.4900 8.9700 10.0300 4.4700 8.9700 nan nan nan nan 2.6900 nan
```

Fold an array

- Create initial data:

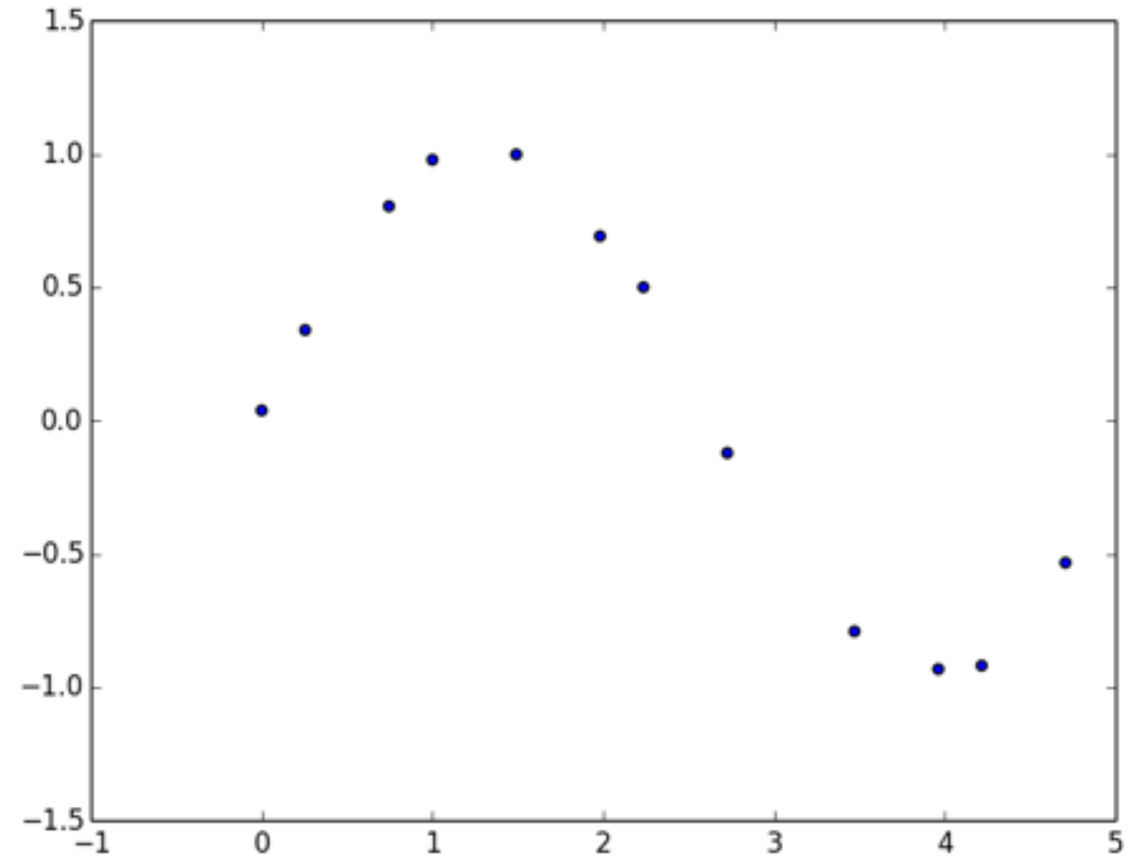
```
In [746]: x = np.linspace(0,365,12)

In [747]: y = np.sin(2*np.pi*x/5.2) +
np.random.rand(12)*0.1

In [748]: plt.scatter(x,y)
Out[748]:
<matplotlib.collections.PathCollection
at 0x1c270a5d0>

In [749]: plt.show()
```





- Finding the period:

```
In [750]: plt.scatter(x%5.,y)
Out[750]: <matplotlib.collections.PathCollection at 0x1c274

In [751]: plt.show()

In [752]: plt.scatter(x%5.2,y)
Out[752]: <matplotlib.collections.PathCollection at 0x1c9b0

In [753]: plt.show()
```



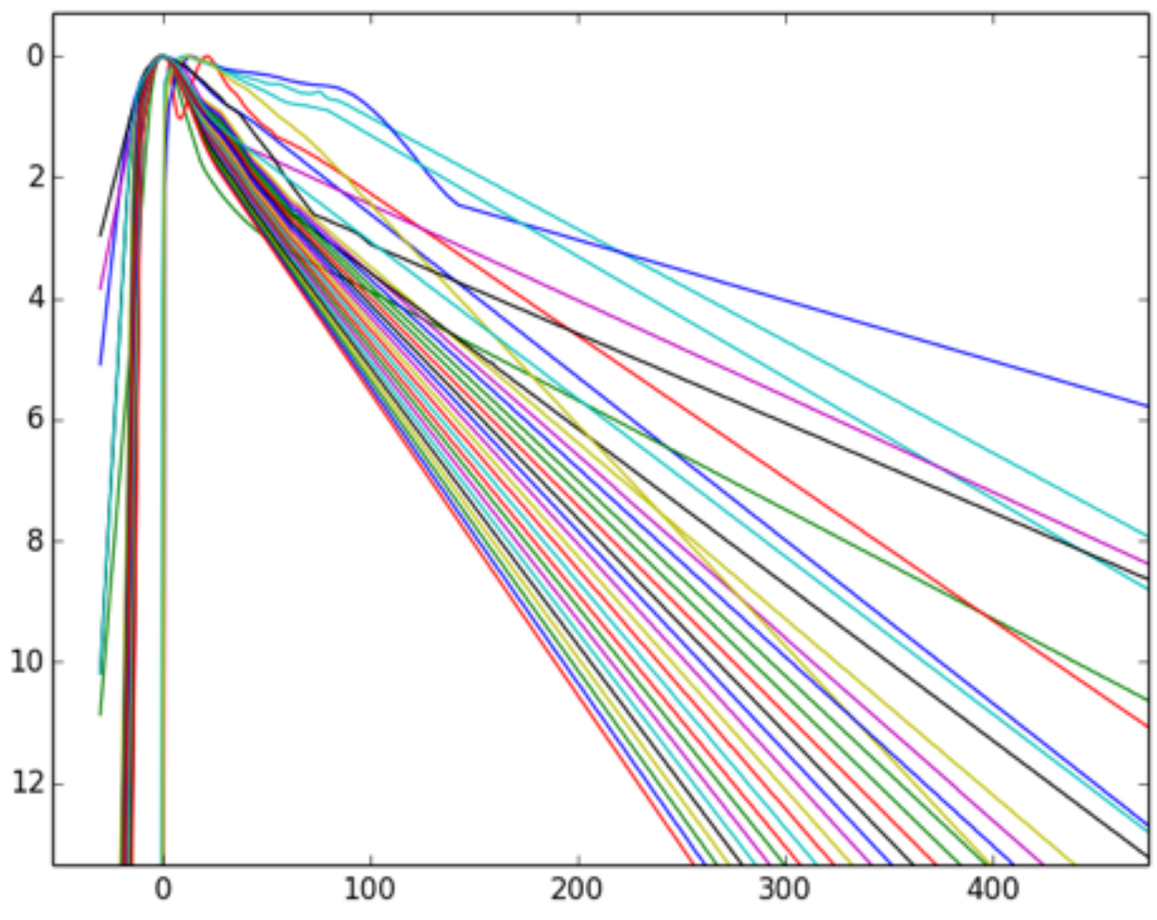
Exercise:

plot all the light curves form the file

(one possible) Solution

- Obtain all the names and loop trough them:

```
In [846]: for n in set(t.dtype.names) - set("t"):
              plt.plot(t["t"], t[n])
          ......:

In [847]: plt.gca().invert_yaxis()

In [848]: plt.show()
```



Thank you!

nblago@caltech.edu