Lab #3

NumPy and Matplotlib

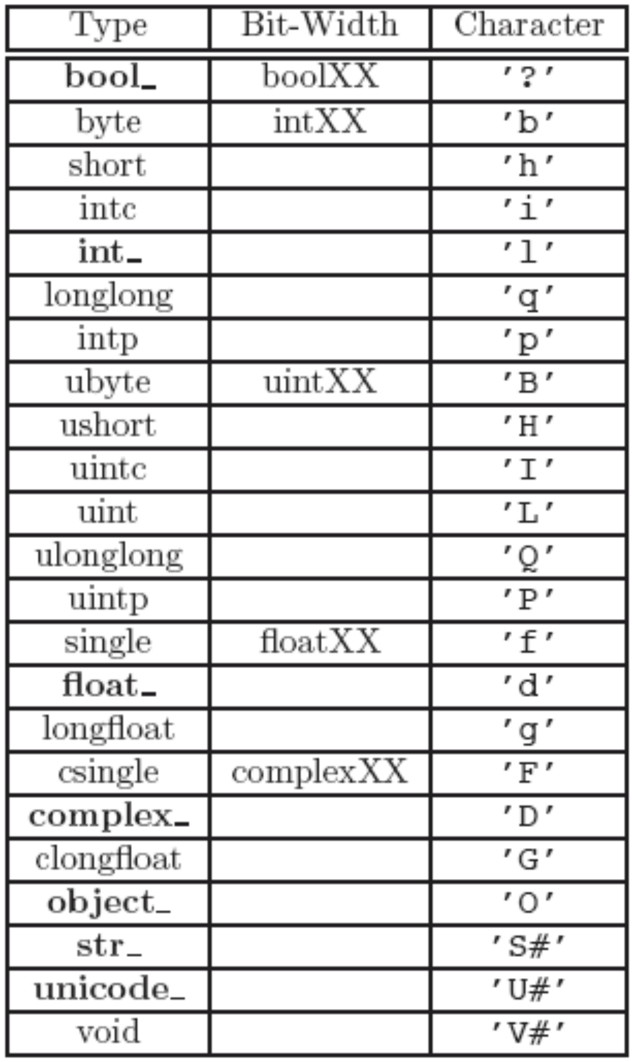
Prof. Steven Ludtke

N410.07, sludtke@bcm.edu

NumPy

- **http://csc.ucdavis.edu/~chaos/courses/nlp/Software/NumPyBook.pdf** # numpy book

- from numpy import *
- a=arange(60)
- b=a.reshape(10,6) # make 2-D matrix
- c=a.reshape(3,4,5) # make 3-D (tensor)
- b.shape # current dimensions
- b.size # total number of elements
- b.ndim # dimensionality
- b.dtype # type of value stored
- b.astype("")



Type	Bit-Width	Character
bool_	boolXX	'?'
byte	intXX	'b'
short		'h'
intc		'i'
int_		'l'
longlong		'q'
intp		'p'
ubyte	uintXX	'B'
ushort		'H'
uintc		'I'
uint		'L'
ulonglong		'Q'
uintp		'P'
single	floatXX	'f'
float_		'd'
longfloat		'g'
csingle	complexXX	'F'
complex_		'D'
clongfloat		'G'
object_		'O'
str_		'S#'
unicode_		'U#'
void		'V#'

NumPy

- a=zeros((nx,ny,...))
- a=fromfunction(lambda i,j:i+j,(4,5))
- a=loadtxt(filename)   # read multicolumn data from a text file
- savetxt(filename,array)   # write multicolumn data to a text file
- a.transpose()   # transpose an array (swap rows with columns)
- a=arange(0,20,.1)   # fractional version of range()
- a*10   # multiply each element !
- b=sin(a) # sin() of each element
- c=a[c>0]   # condition, returns elements >0
- c.sort()   # sort values in-place
- c.mean(),var(),std(),prod()   # average, variance, standard dev, product
- inner(a,b), outer(a,b)   # inner and outer matrix products
- dot(a,b), cross(a,b)   # dot and cross products (similar to above)
- histogram(a,bins,range)   # compute a histogram of ‘a’

Lambda

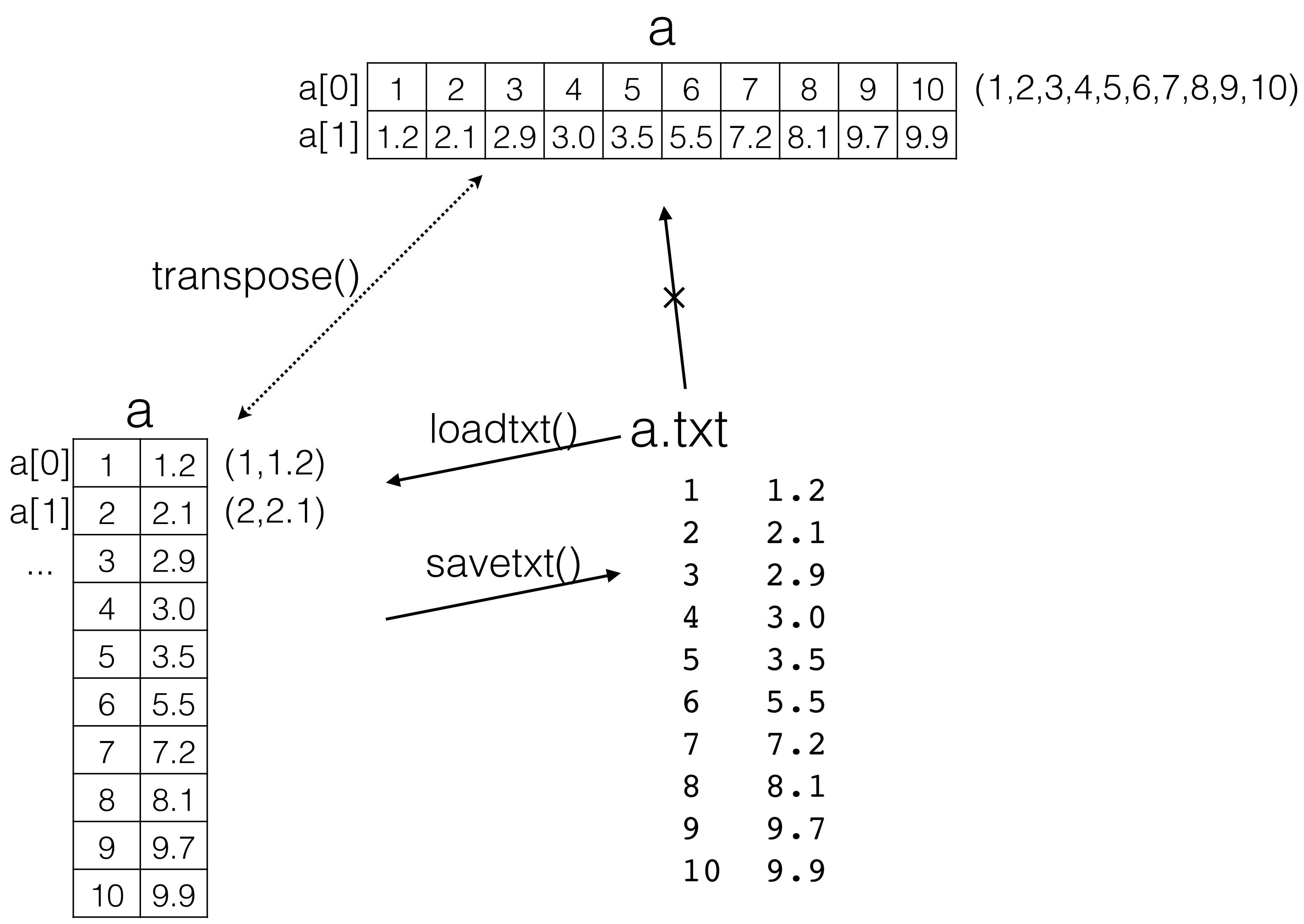
- Unnamed single-use functions:

```
lambda x,y: x*y*23
```

- equivalent to:

```
def f(x,y) : return x*y*23
```

NumPy.array



a[0]	1	2	3	4	5	6	7	8	9	10
a[1]	1.2	2.1	2.9	3.0	3.5	5.5	7.2	8.1	9.7	9.9

a[0]	1	1.2
a[1]	2	2.1
...	3	2.9
	4	3.0
	5	3.5
	6	5.5
	7	7.2
	8	8.1
	9	9.7
	10	9.9

```
1   1.2
2   2.1
3   2.9
4   3.0
5   3.5
6   5.5
7   7.2
8   8.1
9   9.7
10  9.9
```

matplotlib (pylab)

- Matlab-like plotting library
- **http://matplotlib.sourceforge.net/users/pyplot_tutorial.html**

```
ipython --pylab          # special mode for interaction with pylab

x=arange(0,4*pi,0.05)  # from numpy

y=sin(x)                 # easy to apply a function to a list of values

plot(x,y)                # plot x,y and open a display window
```

OR

```
python

from pylab import *      # <-- only if you don’t use ipython

x=arange(0,4*pi,0.05)

y=sin(x)                 # easy to apply a function to a list of values

plot(x,y)                # plot x,y and open a display window

show()                   # opens the plot window (blocks on some machines)
```

matplotlib (pylab)

```
ipython --pylab   # special mode for interaction with pylab
x=arange(0,4*pi,0.05) # from numpy
y=sin(x)        # easy to apply a function to a list of values
y2=cos(x)
figure(1)       # start a new figure
subplot(211)    # make a 1x2 set of plots and move to 1st
plot(x,y)       # plot x,y and open a display window
subplot(212)    # start on the 2nd subplot
plot(x,y2)      # second plot
```

Lab 3

- There is a sample program called plot.py on the class website. There are also two sample .txt files. You will need all three for this lab.

- Begin by discussing the program among yourselves line by line. Make sure you all understand what the program is doing. Add print statements if necessary to understand the data at each step.

- There are three exercises below. Each person should select one, as before, and your group should turn in a single aggregate program. Decide together whether you will use argv[] or raw_input() for user options.:

    1. The first sample file has two columns of numbers. The second file has 9 columns. Modify the program so the user can select which column should be the X-axis and which should be the Y-axis in the plot.

    2. Instead of displaying the plot on the screen interactively, have the plot saved to a .pdf or .png file (user's choice) instead. Allow the user to select between lines and points when the plot is displayed.

    3. Allow the user to enter labels for the X and Y axes and and overall title for the plot.

Note: if you need to pass strings with spaces into a program with argv[], you must have " around the string on the command-line when you run the program (exercise 3), or each word will be treated as a separate argument in argv[].

Homework 4

- Starting with the program from lab (either the original, or the final version after your group's modifications). Modify the program:
  - Display the data points as points rather than a line (one of the lab problems)
  - Compute a linear fit (least squares) through the x/y data points
  - Draw the fit line (as a line) on the plot, and label it in some fashion with the y=mx+b from the fit.