Phys 60441
Techniques of Radio Astronomy
Part 1: Python Programming
LECTURE 4

Tim O’Brien

Room 3.214 Alan Turing Building

tim.obrien@manchester.ac.uk

An exercise

- Write a Python script that carries out the following:
  - Prompts the user for the name of a data file
  - Reads from the file a list of (x,y,z) groups of real numbers (one group of 3 per line) and stores these in three arrays.
  - Prints to the screen the number of lines read from the file.
  - Calculates the minimum, maximum and mean of x and y and prints these to the screen.
- Employ tidy, structured programming, appropriate use of functions (e.g. to calculate min, max and mean) and ensure your program can deal with datafiles of variable length and including comments.

Matplotlib example

(ex12.py) See http://www.scipy.org/Cookbook/Matplotlib

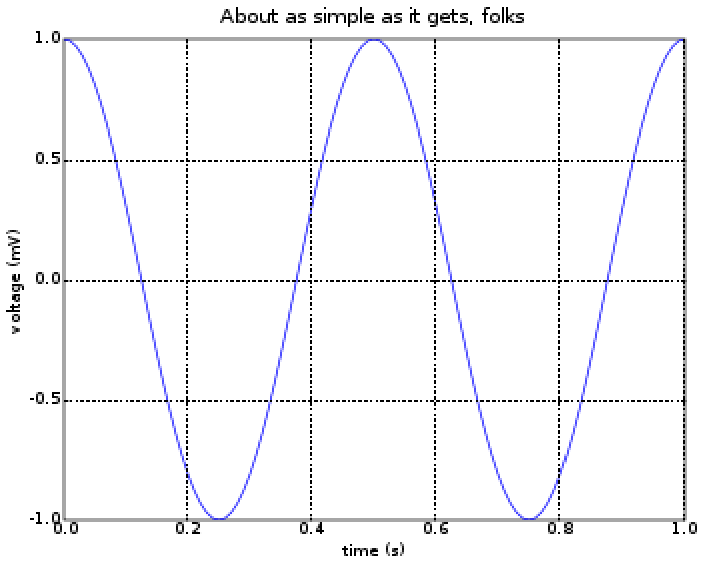
```
from numpy import *
import pylab as plt

t = arange(0.0, 1.0, 0.01)
s = cos(4*pi*t)

print t,s

plt.plot(t, s)

plt.xlabel('time (s)')
plt.ylabel('voltage (mV)')
plt.title('About as simple as it gets, folks')
plt.grid(True)
plt.savefig('simple_plot')
plt.show()
```



Produces png graphics file called simple_plot.png

Cookbook for SciPy

- http://www.scipy.org/Cookbook
- Lots of worked examples.
- Also includes cookbooks for matplotlib etc

Linear Regression Example

(ex13.py)

http://docs.scipy.org/doc/numpy/reference/generated/numpy.polyval.html

http://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html

http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.linregress.html

```
import scipy as sp
from scipy import stats
import pylab as plt
n=50                          # number of points
x=sp.linspace(-5,5,n)         # create x axis data
a, b=0.8, -4
y=sp.polyval([a,b],x)
yn=y+sp.randn(n)              #add some noise
(ar,br)=sp.polyfit(x,yn,1)
yr=sp.polyval([ar,br],x)
err=sp.sqrt(sum((yr-yn)**2)/n) #compute the mean square error
print('Linear regression using polyfit')
print(‘Input parameters: a=%.2f b=%.2f’ % (a,b))
print(‘Regression: a=%.2f b=%.2f, ms error= %.3f' % (ar,br,err))
plt.title('Linear Regression Example')
plt.plot(x,y,'g--')
plt.plot(x,yn,'k.')
plt.plot(x,yr,'r-')
plt.legend(['original','plus noise', 'regression'])
plt.show()
(a_s,b_s,r,xx,stderr)=stats.linregress(x,yn)
print('Linear regression using stats.linregress')
print('parameters: a=%.2f b=%.2f’ % (a,b))
print(‘regression: a=%.2f b=%.2f, std error= %.3f' % (a_s,b_s,stderr))
```

Note format of points/lines

(ex14.py)

Least squares fit example

From scipy example list for leastsq: http://www.scipy.org/scipy_Example_List#head-4c436ae0085d9a56056425d11abff4ccdd3d3620

Also see page 19 of the documentation http://docs.scipy.org/doc/scipy/scipy-ref.pdf

v is vector of parameter values,
x is independent variable

```
from pylab import *
from numpy import *
from scipy.optimize import leastsq

fp = lambda v, x: v[0]/(x**v[1])*sin(v[2]*x)   # parametric function

v_real = [1.7, 0.0, 2.0]
fn = lambda x: fp(v_real, x)                   # fn to generate noisy data

e = lambda v, x, y: (fp(v,x)-y)                # error function

n, xmin, xmax = 30, 0.1, 5                     # Generate noisy data to fit
x = linspace(xmin,xmax,n)
y = fn(x) + rand(len(x))*0.2*(fn(x).max()-fn(x).min())

v0 = [3., 1, 4.]                               # Initial parameter values
v, success = leastsq(e, v0, args=(x,y), maxfev=10000)      # perform fit
print ‘Fit parameters: ', v
print ‘Original parameters: ', v_real

X = linspace(xmin,xmax,n*5)                    # plot results
plot(x,y,'ro', X, fp(v,X))
show()
```

PyFITS: Handling FITS Images

(ex15.py)

- FITS – Flexible Image Transport System, data format widely used in astronomy
- http://www.stsci.edu/resources/software_hardware/pyfits

hdulist contains a list of Header Data Units each consisting of a header (comprising multiple “cards”) and data

hdulist[0] contains the primary HDU

```
import pyfits
import pylab as plt

hdulist = pyfits.open('testimage.fits')
hdulist.info()
print hdulist[0].header[13]
prihdr = hdulist[0].header
print prihdr.ascardlist().keys()
print prihdr.ascardlist()[:51]

scidata = hdulist[0].data
print scidata.shape
print scidata[10,10]

imgplot = plt.imshow(scidata)
imgplot.set_cmap('hot')
imgplot.set_clim(5000,6000)
plt.colorbar()
plt.show()
```