“Amongst our weaponry are such diverse elements as: fear, surprise, ruthless efficiency, an almost fanatical devotion to the Pope, and nice red uniforms!”

(and savvy plotting packages….)
What is matplotlib?

matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in Python scripts, the Python and iPython shell (ala matlab or mathematica), web application servers, and six graphical user interface toolkits.

http://matplotlib.sourceforge.net
What can it do?

- You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc, with just a few lines of code.
The Many (Inter)faces of matplotlib

- **pyplot**
  - Pyplot provides a Matlab-style state-machine interface to the underlying object-oriented plotting library in matplotlib.
  - Preferred method of access for interactive plotting.
  - [http://matplotlib.sourceforge.net/users/pyplot_tutorial.html](http://matplotlib.sourceforge.net/users/pyplot_tutorial.html)

- **pylab**
  - pylab combines the pyplot functionality (for plotting) with the numpy functionality (for mathematics and for working with arrays) in a single namespace, making that namespace (or environment) even more Matlab-like.
  - Formerly preferred method of access for interactive plotting, but still available.

- **matplotlib API (object-oriented)**
  - Used to embed matplotlib within an application, e.g. GFM
  - [http://matplotlib.sourceforge.net/api](http://matplotlib.sourceforge.net/api)
pyplot Vs. pylab

pyplot:

```python
import matplotlib.pyplot as plt
import numpy as np

x = np.arrange(0, 10, 0.2)
y = np.sin(x)

plt.plot(x, y)
plt.show()
```

pylab:

```python
from pylab import *

x = arange(0, 10, 0.2)
y = sin(x)

plot(x, y)
show()
```
import matplotlib.pyplot as plt
plt.plot([1,2,3])
plt.ylabel('some numbers')
plt.show()
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16], 'ro')
plt.axis([0, 6, 0, 20])
import numpy as np
import matplotlib.pyplot as plt
mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='g', alpha=0.75)
plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)

Note: You can also format text with LaTeX markup
from __future__ import division
from matplotlib.patches import Patch
from pylab import *

def func3(x,y):
    return (1- x/2 + x**5 + y**3)*exp(-x**2-y**2)

dx, dy = 0.05, 0.05
x = arange(-3.0, 3.0001, dx)
y = arange(-3.0, 3.0001, dy)
X,Y = meshgrid(x, y)

Z = func3(X, Y)
pcolor(X, Y, Z)
colorbar()
axis([-3,3,-3,3])
show()
And a bit more fancy

from numpy.random import uniform, seed
from matplotlib.mlab import griddata
import matplotlib.pyplot as plt
import numpy as np
seed(-1)
npts = 200
x = uniform(-2,2,npts)
y = uniform(-2,2,npts)
z = x*np.exp(-x**2-y**2)
xi = np.linspace(-2.1,2.1,100)
yi = np.linspace(-2.1,2.1,100)
zi = griddata(x,y,z,xi,yi)
CS = plt.contour(xi,yi,zi,15,linewidths=0.5,colors='k')
CS = plt.contourf(xi,yi,zi,15,cmap=plt.cm.jet)
plt.colorbar()
plt.scatter(x,y,marker='o',c='b',s=5)
plt.xlim(-2,2)
plt.ylim(-2,2)
plt.title('griddata test (%d points)' % npts)
plt.show()}
Saving Your Plot for Posterity

Syntax:

```
savefig(fname,  
    dpi = None,  
    facecolor = 'w',  
    edgecolor = 'w',  
    orientation = 'portrait',  
    papertype = None,  
    format = None,  
    transparent = False)
```

An Example:

```
import matplotlib.pyplot as plt
plot.savefig('example.png',  
    format = 'png')
```
Customizing matplotlib

- matplotlib uses matplotlibrc configuration files to customize all kinds of properties

- Where is matplotlibrc?
  - matplotlibrc in the current working directory, usually used for specific customizations that you do not want to apply elsewhere.
  - .matplotlib/matplotlibrc, for the user’s default customizations.
  - INSTALL/matplotlib/mpl-data/matplotlibrc, where INSTALL is something like /opt/local/lib/python2.5/site-packages

- [http://matplotlib.sourceforge.net/users/customizing.html](http://matplotlib.sourceforge.net/users/customizing.html)