Matplotlib

Diane Trout

Matplotlib

Matplotlib is a matlab inspired 2d plotting package for python. It,

- has "two APIs".
- supports multiple backends.
- is (sometimes) easy to use from the interpreter.
- has many, many plot types.
- can embed plots in gui toolkits.

APIs

- Matplotlib’s original purpose in life was to make python more like matlab.
- Provides interactive commands, like:
  - subplot(121)
  - plot([1,2,3],[2,3,4], ’r-’)
  - subplot(122)
  - scatter([2,1,3],[1,3,4])
  - gca()
  - show()

API

- However being written in python, those commands are backed by a class library.
  - f = figure.Figure()
  - canvas = FigureCanvasTkAgg(f)
  - p = f.add_subplot(111)
  - p.plot([1,2,3],[1,2,3])
- in the interpreter I tend to grab the results of the matlab like commands (which return python objects) to customize my plot.

API

- Has 17 "backends"
- cocoa, fltk, gd, gdk, gtk, paint, postscript, qt, svg, tk, wsx
- (several of those come in "agg" and non-"agg" flavors)
- (agg is a 2d rendering engine (http://antigrain.com)

Interpreter Issues

- Some of the backends are less amenible to use from an interpreter.
- The GTK one is rather cranky, and will only let you see a single plot.
- TkAgg and QtAgg work better.
- matplotlib.use("TkAgg") (or "QtAgg", etc).
- or change the default in matplotlibrc
- The problem is that once the first Gtk window closes, all subsequent calls block, and unfortunately Gtk is the default on linux.

Plots

They have quite a few examples...

Simple Line Plot

```
# keep namespaces clean
import pylab as p

#plot our data
def main(argv=None):
    series1 = [1,3,1,5,6,7,0,2,4,]
    # standard line plot
    p.plot(series1)
    p.show()
```

Matplotlib

- I usually just look at the examples when trying to figure out what other plots to make.
- For more information, http://matplotlib.sf.net