

The Matplotlib User's Guide

John Hunter

January 21, 2005

Contents

1	Introduction	5
1.1	Migrating from matlab	6
1.2	Numerix	7
1.2.1	Choosing Numeric or numarray	8
1.3	Backends	9
1.4	Integrated development environments	9
1.5	Interactive	10
1.6	Customization using .matplotlibrc	11
1.6.1	RC file format	11

2.10 Spectral analysis	36
2.11 Axes properties	38
2.12 Legends and tables	38
2.13 Navigation	38
2.13.1 Classic toolbar	38
2.13.2 toolbar2	39
2.14 Event handling	39
2.15 Customizing plot defaults	41
3 Font finding and properties	43
4 Collections	45
5 Tick locators and formatters	47
5.1 Tick locating	47
5.2 Tick formatting	48
5.3 Example 1: major and minor ticks	48
5.4 Example 2: date ticking	49
6 Cookbook	53
6.1 Plot elements	53
6.1.1 Horizontal or vertical lines/spans	53
6.1.2 Fill the area between two curves	53
6.2 Text	54
6.2.1 Adding a ylabel on the right of the axes	54
6.3 Data analysis	54
6.3.1 Linear regression	54
6.3.2 Polynomial regression	55
6.4 Working with images	56
6.4.1 Loading existing images in .ng im716	55

Chapter 1

Introduction

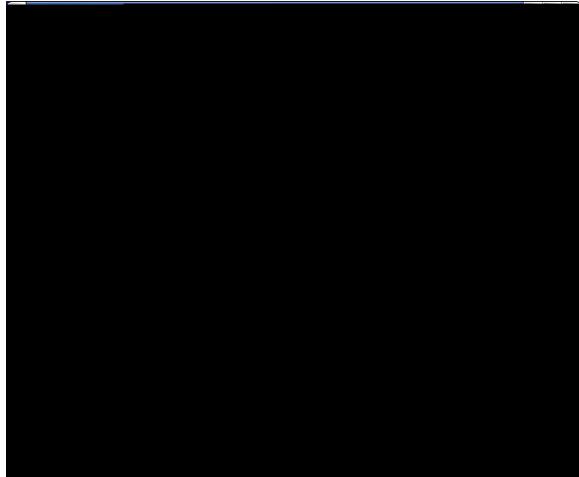
matplotlib is a library for making 2D plots of arrays in python. Although it has its origins in emulating the matlab™ graphics commands, it does not require matlab, and can be used in a pythonic, object oriented

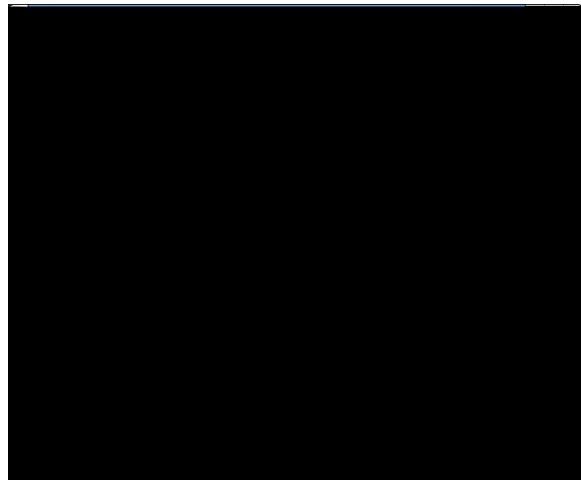
with. There are many GUIs for python: pygtk, wxpython, Tkinter, PyQT, pyfltk, and more, and matplotlib supports most of them.

IDE		GUI	
-----	--	-----	--

Chapter 2

The pylab interface





<i>Ab112b112re11</i>	<i>Fu112i112at112on</i>
aa	ant112i112al112i112ased
c	col112or
i112s	i112i112nesty112le
i112w	i112i112newi112d112t112h
mec	mar112ker112ed112gecol112or
mew	mar112ker112ed112gewi112d112t112h
mf112c	mar112ker112f112acecol112or
ms	mar112ker112si112ze

```
ylabel('voltage (m)' , color=0.5)    # ylabel is light gray  
title(
```

2.4.2 Loading and saving binary data

```
>>> from pylab import  
>>> figure()  
>>> subplot(111)  
>>> plot([1,2,3])
```

2.5.1 figure

You can create and manage an arbitrary number of figures using the `figure` command. The standard way to create a figure is to number them from 1... N . A call to `figure(1)` creates figure 11!te5(si)16tuf,tato

2.5.2 subplot

axes and subplot are both used to create axes in a figure. subplot is used more commonly, and creates axes assuming a regular grid of axes *numRows* by *numCols*. For example, to create two rows and one column

`subplot(3, 4, 12)`. matplotlib uses matlab style indexing in creating figures and axes, so `subplot(3, 4, 1)` is the first subplot, not `subplot(3, 4, 0)`.

ordinates. `left, bottom = (0, 0)`

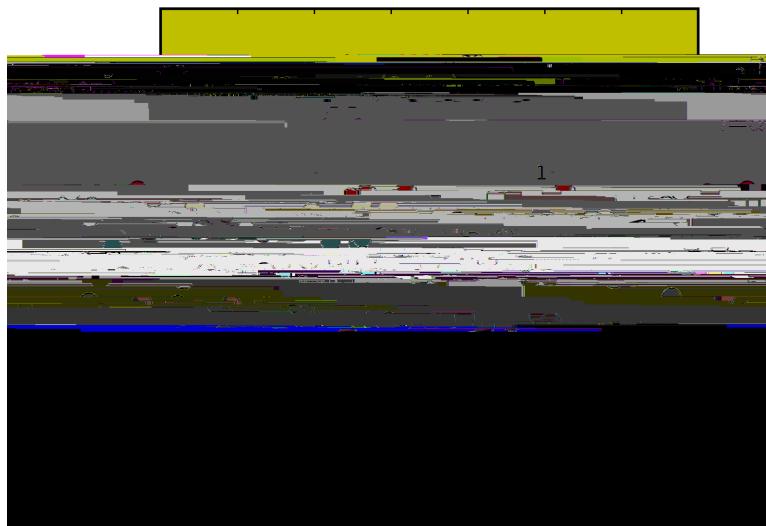


Figure 2.10: Incorporating T_E

Once the luminance data are normalized, they color mapper transforms the normalized data to RGBA

- ml ab. wi ndow_none - no windowing
- ml ab. wi ndow_hannin g - multiply each segment by a Hanning window

2.13.2 toolbar2

The toolbar2 buttons (see Figure 2.15) behave very differently from the classic matplotlib toolbar

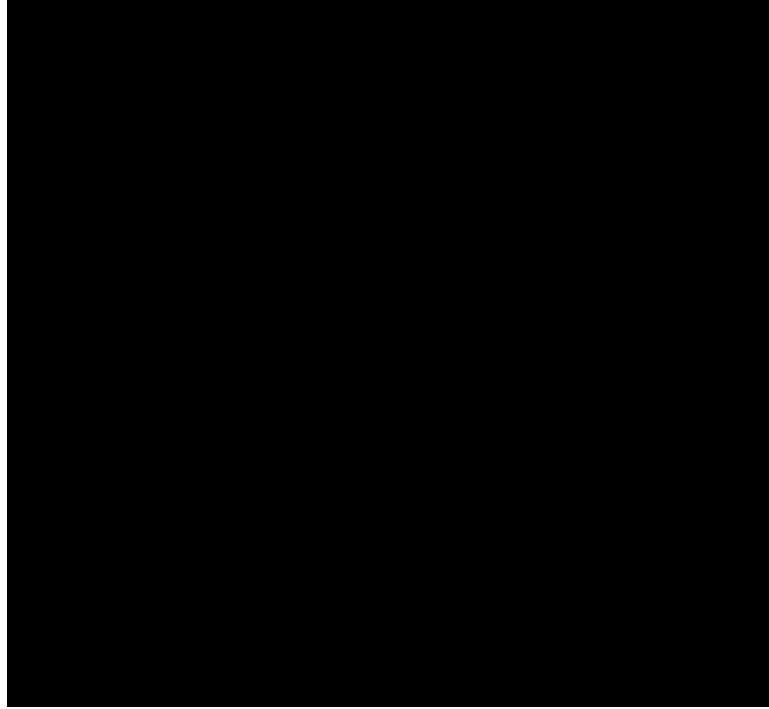


Figure 2.15: The newfangled toolbar2, discussed in Section 2.13.2

You can connect to the following events: 'button_

```
# Connect to
```



```
family = ['serif', 'sans-serif', 'cursive', 'fantasy', 'monospace']  
t =
```

Chapter 4

Collections

Chapter 5

Tick locators and formatters

The `matplotlib.ticker` module contains classes to support completely configurable tick locating and formatting. Although the locators know nothing about major or minor ticks, they are used by the `Axis` class to support major and minor tick locating and formatting. Generic tick locators and formatters are provided,



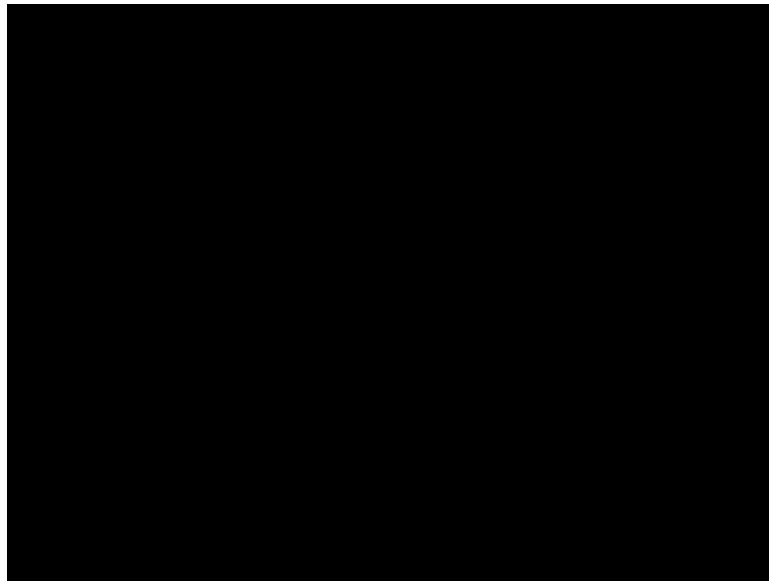


Figure 5.1: Creating custom major and minor tick locators and formatters; see Listing 5.1

```
# import the tick locator and formatter classes
from
```



```
# but we'll override the default with
```


Chapter 6

Cookbook

6.1 Plot elements

6.1.1 Horizontal or vertical lines/spans

It is often useful to draw a line that stretches from the left to the right side of the axes at a given height, e.g. to represent a y-axis threshold. In this case, the left and right are plotted in axes coordinates (0 and 1 respectively), and the y coordinate is in data coordinates. Plotted this way, the horizontal extent of the line will not change if you interactively change the xlimits, e.g. by using the pan navigation tool. Although you can create these lines yourself using `matplotlib.lines.Line2D` instances and setting the appropriate transforms, several helper functions are provided to make this easier.

6.1.2 Fill the area between two curves

The `fill` command takes a list of vertices and draws a polygon. A filled area between two curves is simply a

89he

Listing 6.1: Fill the area between two curves; see Figure 6.1

```
from pylab import
x1 = arange(0, 2, 0.01)
y1 = sin(2 pi x1)
y2 = sin(4 pi x1) + 2

# reverse x and y2 so the polygon fills in order
x = concatenate( (x1,x1[::-1]) )
y = concatenate( (y1,y2[::-1]) )

p =
```

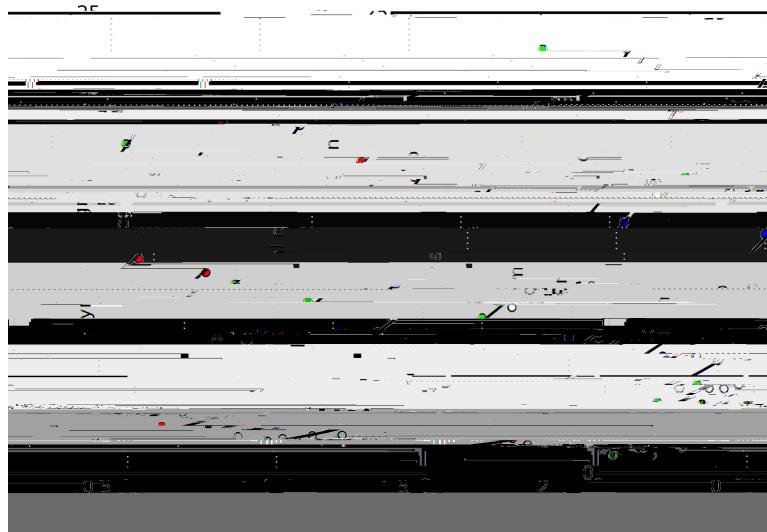



Figure 6.3: Estimating a best fit cubic for some random data; see Listing 6.3

```
x = arange(0.0, 2.0, 0.05)
nse = 0.6 randn(len(x))
y = 1.1 + 3.2 x + 0.1 x2 + 2 x3 + nse
# the bestfit line
```


6.4.4 Defining your own colormap

Perry Greenfield has provided a nice framework with `matplotlib.colors.LinearSegmentedColormap` to de-

Chapter 7

Matplotlib API

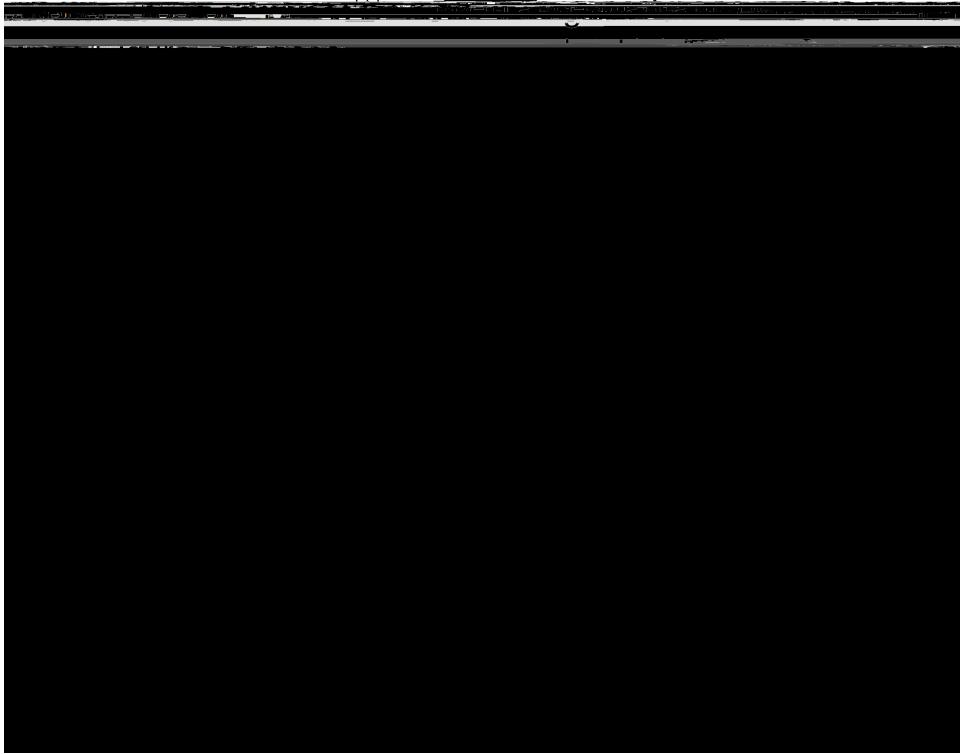


Figure 7.2: The inheritance diagram for The FigureCanvas hierarchy. The FigureCanvas is a backend dependent class which contains a figure instance. For GUI kend

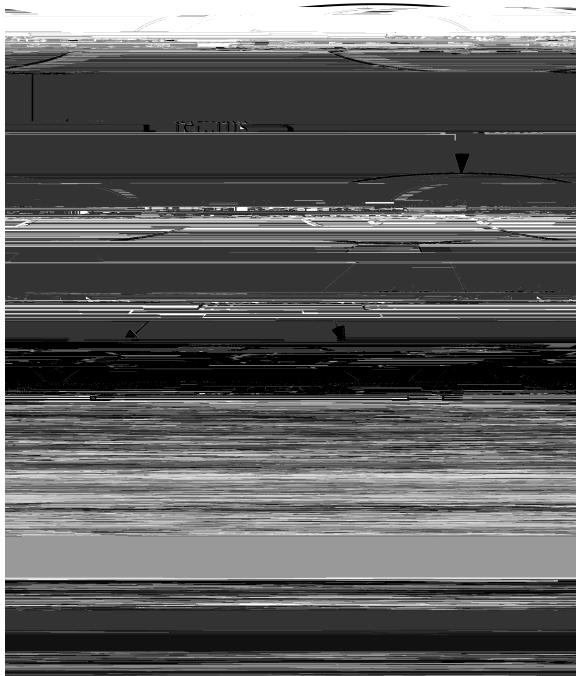


Figure 7.5: The pylab interface function `new_figure_manager` returns a backend dependent concrete implementation of `matplotlib.libbackend_bases.FigureManagerBase`

```
manager.window.show()  
  
if gtk.main_level() == 0 and mainloop:  
    if gtk.pygtk_version >
```


Appendix A

A sample .matplotlibrc

```
#### MATPLOTLIBRC
```

```
lines.markeredgewidth : 1.0
```

```
font.stretch      : normal  
font.size        : medium
```

```
### SAVING FIGURES
# the default savefig params can be different for the
```

Appendix B

mathtext symbols

THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between JDH and Licensee. This License Agreement does not grant permission to use JDH trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

Bibliography

Julius S. Bendat and Allan G. Piersol.