

# The Matplotlib User's Guide

John Hunter

October 17, 2004







# Chapter 1

antigrain library that ships with matplotlib -







## 1.5 Interactive

## 1.17. INSTALLING

A complete sample rc file is shown in Appendix A.

The matplotlib rc values are read into a dictionary `rcParams` contains the key/value pairs. You can changes these values within a script by importing this dictionary. For example, to require that a given script uses numarray, you could do

```
from matplotlib import rcParams
rcParams['numerix'] = 'numarray'
from matplotlib.matlab import
```

Additionally, the commands

numerix settings in setup.py and in your .matplotlibrc file are the same, and are the same as the array package you typically use.

Note that if you install matplotlib anywhere other than the default location, you will need to set the MATPLOTLIBDATA environment variable to point to the install base dir. Eg, if you install matplotlib with `python setup.py build --prefix=/home/jdhunter` then set MATPLOTLIBDATA to `/home/jdhunter/share/matplotlib`.

### 1.7.2 Installing on windows

If you don't already have python installed, you may want to consider using the enthought edition of python, which has scipy, Numeric, and wxpython, plus a lot of other goodies, preinstalled - <http://www.enthought.com/python>. With the enthought edition of python + matplotlib installer, the following backends should work out of the box: agg, wx, wxagg, tkagg, ps and svg.





## Chapter 2

# The matlab<sup>TM</sup> interface

Although matplotlib has a full object oriented API (see Chapter 7), the primary way people create plots is



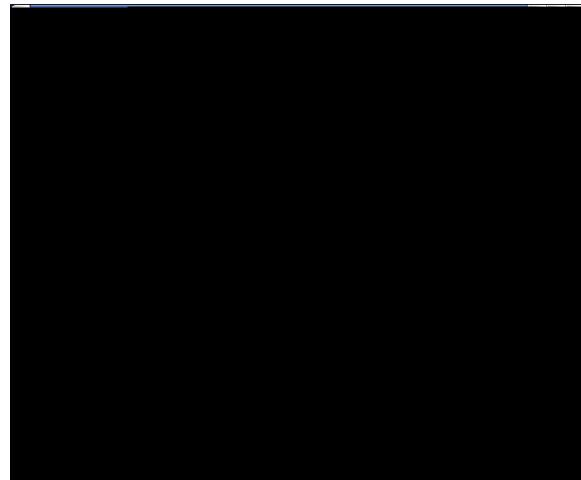


Figure 2.2: A simple plot decorated with some text labels and an axes grid



Figure 2.3: A sine wave added to the simple plot. This may not be what we wanted. Because the `hold` state was on, the two plots were superimposed.

#### Using keyword arguments to control line properties





```
title("Don't try this at home, folks", color='#afeeee')
```

b	blue
g	green
r	red
c	cyan
m	magenta
y	yellow
k	black
w	white
0.75	

### 2.4.2 Loading and saving binary data

ASCII is bloated and slow for working with large arrays, and so binary data should be used if performance is a consideration. To save the array X in binary form, use the numerix `tostring` method

```
# open the
```





```
ylabel('Undamped')
```

Likewise, to create two columns and one row of axes, you would use subplot(121) to create the left axes and subplot(122) to create the right axes. If the total number of axes exceeds single digits, use comma

### 2.5.3 axes

When you need a finer grained control over axes placement than afforded by subplot, use the funcaxes command. The axes

## *2.6. TEXT*

```
# instance methods
>>> l = ylabel(
```

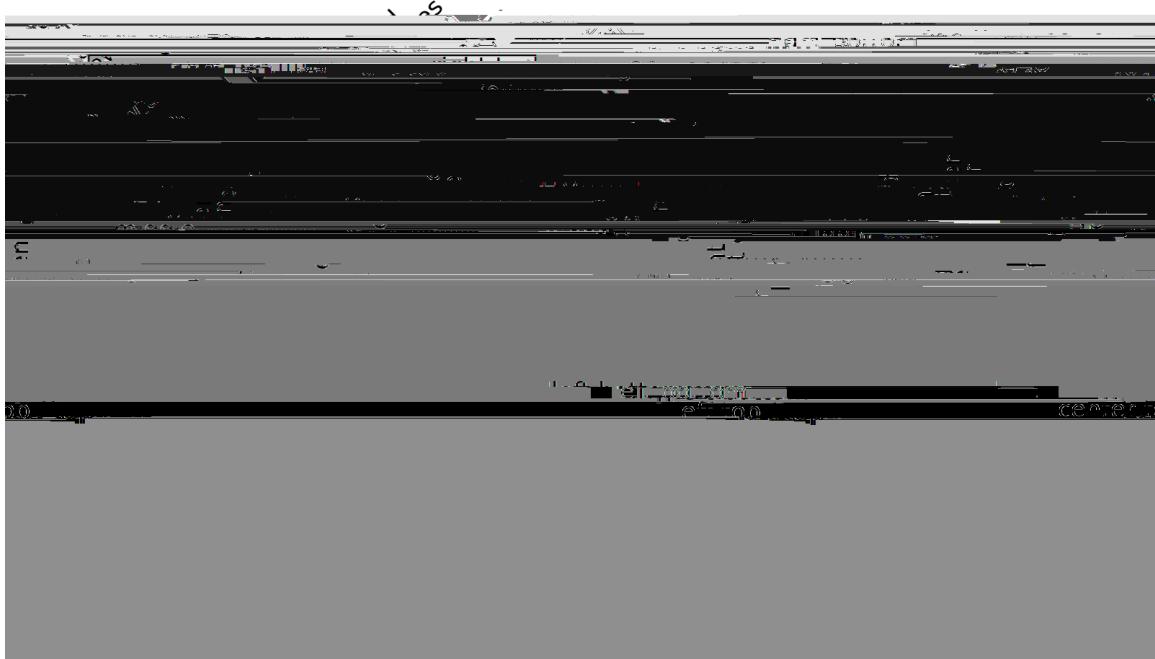


Figure 2.9: Aligning text with *horizontalalignment*, *verticalalignment*, and *multialignment* options to the `text` command; see Listing 2.4

```
ax.text(left, bottom, 'left top',
        horizontalalignment='left',
        verticalalignment='top',
        transform=ax.transAxes)

ax.text(left, bottom, 'left bottom',
        horizontalalignment='left',
        verticalalignment='bottom',
        transform=ax.transAxes)

ax.text(right, top,
```

```
horizontalalignment='center' ,
```

















### 2.13.2 toolbar2

The toolbar2 buttons (see Figure 2.15) behave very differently from the classic matplotlib toolbar (else why introduce a new one!) despite the visual similarity of the forward and back buttons.

The Forward

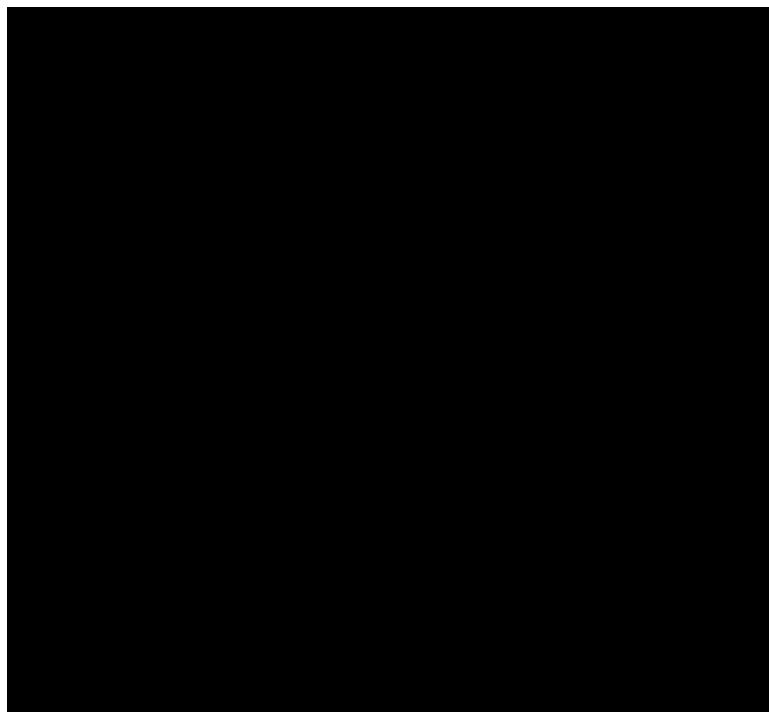


Figure 2.15: The newfangled toolbar2, discussed in Section 2.13.2

Then whenever the user clicks anywhere on the figure canvas, your function will be called and passed a `matplotlib.backend`





## Chapter 3

# Font finding and properties

`matplotlib.lib.fonts.font_manager` is module for finding, managing, and using fonts across-platforms. This module provides a single `FontManager` that can be shared across backends and platforms. The `findfont()`

```
family = ['serif', 'sans-serif', 'cursive', 'fantasy',
```

Chapter 4

## Collections



# Chapter 5

<i>Class</i>	<i>Summary</i>
MinuteLocator	locate minutes
HourLocator	locate hours
DayLocator	locate specified days of the month

*5.4. EXAMPLE 2: DATE TICKING*

stock price, a string formatter (`FormatStrFormatter`) is used to place dollar signs on the y tick labels.

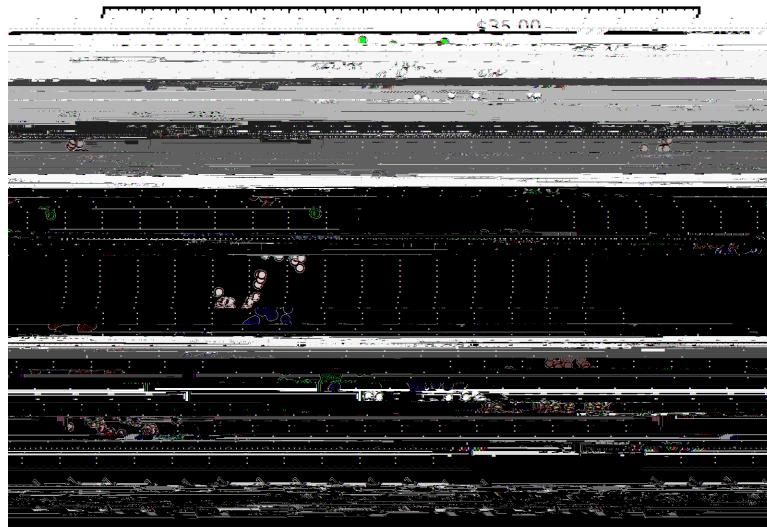


Figure 5.2: Providing custom tick locators and formatters for financial/date plots; see Listing 5.2.

Listing 5.2: Custom date tick locators and formatters; see Figure 5.2

```
import datetime
from matplotlib.matlab import
from matplotlib.dates import MONDAY, SATURDAY
from matplotlib.finance import quotes_historical_yahoo
from matplotlib.dates import MonthLocator, WeekdayLocator, DateFormatter
from matplotlib.ticker import FormatStrFormatter

# the start and end date range for the first plts
```

```
# but we'll override the default with our custom locators and
# formatters
ax.xaxis.set_major_locator(months)
ax.xaxis.set_major_formatter(monthsFmt)
ax.xaxis.set_minor_locator(mondays)

# format the y axis in dollars
ax.yaxis.set_major_formatter(dollarFmt)

# call autoscale to pick intelligent view limits based on our
```





```
y = concatenate( (y1,y2[::-1]) )  
p = fill
```







```
im2 = imshow(Z2, cmap=cm.jet , alpha=.9,  
            interpolation='bilinear'
```

.







```
gc.set_foreground(
```



Figure 7.3: Matplotlib Artist hierarchy. The primitive Artists are the Patches, Line2D, Text, AxesImage,

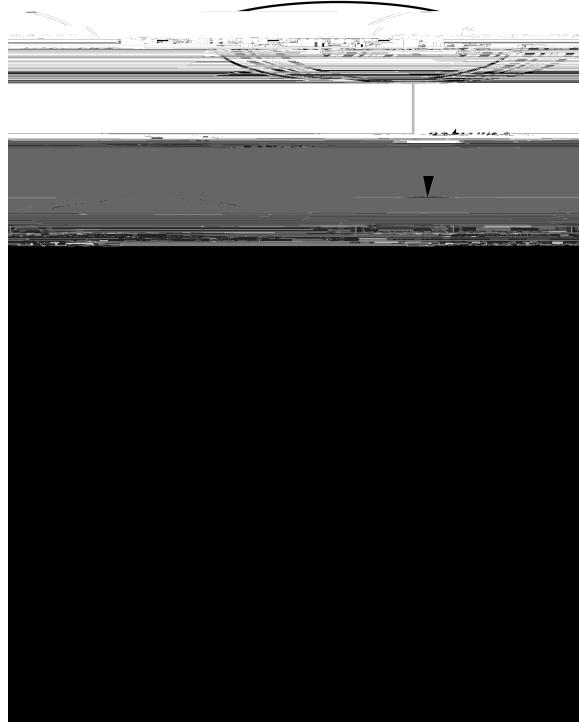


Figure 7.5: The matlab interface function `new_figure_manager` returns a backend dependent concrete implementation of `matplotlib.backend_bases.FigureManagerBase`, which contains the figure canvas and figure window. The attribute names are shown in lower case, and the backend dependent classes are shown in upper case. The standard attribute naming system allows the matlab interface to make calls across backends to





# Appendix A

## A sample .matplotlibrc

```
#### MATPLOTLIBRC FORMAT
#
# This is a
```

```
lines.markeredgewidth : 1.0
lines.markersize : 6      #
```

```
font.stretch      : normal  
font.size        : medium  
font.serif       : New Century Schoolbook, Century Schoolbook L, Utopia, ITC
```

```
### SAVING FIGURES
# the
```





## Appendix C

# matplotlib source code license

All of the matplotlib src code is distributed under the Python Software Foundation (PSF) license, which permits commercial and noncommercial free use and redistribution as long as the conditions below are met. The VERSION string below is replaced by the current matplotlib version number with each release.

LICENSE AGREEMENT FOR MATPLOTLIB(of)-4nwi th eacB(of)-4nnnac23. 91Td[(  
matpl otl ib i n sourceacB(of)-4documentation.eac23. 91Td[(2. (of)-4Subj ect)-fs Agreement, JDkacB(of)-4hereby  
majDtest-faceacB(of)-4I nrheru eacB(of)-4nnanti onveacB(of)-4nncense Agreement, eacB(of)-4nn2002-2004fofe

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. *Random Data: Analysis and Measurement Procedures*. John Wiley & Sons, New York, 1986.

Eric W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman & Hall/CRC, second edition edition, 2002.