Matplotlib

Nick Thompson

April 21, 2015

Table of contents

Installation

Non-interactive Plots and Gotchas

Interactive Matplotlib

3D Apps

Ubuntu

```
sudo apt-get install tk-dev
sudo apt-get install python3-tk # For tkinter backend
sudo apt-get install python3-pyqt4
sudo apt-get install libblas-dev
sudo apt-get install liblapack-dev # For SciPy
pip3 install -r requirements.txt
```

Mac

```
pip3 install -r requirements.txt
```

Minimal Working Example

```
>>> import matplotlib
>>> from pylab import *
>>> plot([1, 2, 3, 2, 1])
[<matplotlib.lines.Line2D object at 0x7f4bf8ad14a8>]
>>> show()
```

Seriously Annoying; Probably indispensable

- The call to `show()` is very often a no-op.
- This occurs whenever `Matplotlib` doesn't have access to system graphics libraries (virtualenvs?)
- `show()` is a blocking call; `Matplotlib` scripts can be run in batch-mode by setting the *backend* to agg

Matplotlib Backend

```
>>> import matplotlib
>>> matplotlib.get_backend()
'TkAgg'
```

The backend is useful for embedding `Matplotlib` in other applications; unsurprisingly `TkAgg` is for embedding in `Tkinter` GUIs.

Setting Matplotlib Backend

Direct editing of `matplotlibrc`:

```
>>> import matplotlib
>>> matplotlib.matplotlib_fname()
'/home/nthompson/matplotlib_talk/lib/python3.4/site-
    packages/matplotlib/mpl-data/matplotlibrc'
```

```
$$ head -40 /home/nthompson/matplotlib_talk/lib/
    python3.4/site-packages/matplotlib/mpl-data/
    matplotlibrc
#### CONFIGURATION BEGINS HERE

# The default backend; one of GTK GTKAgg GTKCairo
    GTK3Agg GTK3Cairo
# CocoaAgg MacOSX Qt4Agg Qt5Agg TkAgg WX WXAgg Agg
    Cairo GDK PS PDF SVG
# Template.
# You can also deploy your own backend outside of
    matplotlib by
# referring to the module name (which must be in the
    PYTHONPATH) as
# 'module://my_backend'.
backend      : tkagg
```

Available Backends

```
$ python3 -q
>>> import matplotlib
>>> matplotlib.rcsetup.interactive_bk
['GTK', 'GTKAgg', 'GTKCairo', 'MacOSX', 'Qt4Agg', '
    Qt5Agg', 'TkAgg', 'WX', 'WXAgg', 'CocoaAgg', '
    GTK3Cairo', 'GTK3Agg', 'WebAgg', 'nbAgg']
>>> matplotlib.rcsetup.non_interactive_bk
['agg', 'cairo', 'emf', 'gdk', 'pdf', 'pgf', 'ps', '
    svg', 'template']
>>> matplotlib.rcsetup.all_backends
['GTK', 'GTKAgg', 'GTKCairo', 'MacOSX', 'Qt4Agg', '
    Qt5Agg', 'TkAgg', 'WX', 'WXAgg', 'CocoaAgg', '
    GTK3Cairo', 'GTK3Agg', 'WebAgg', 'nbAgg', 'agg', '
    cairo', 'emf', 'gdk', 'pdf', 'pgf', 'ps', 'svg', '
    template']
```

None of these are guaranteed to work!

Setting Matplotlib Backend

Choose backend at runtime:

```
>>> import matplotlib
>>> matplotlib.use(’agg’)
>>> from pylab import *
>>> plot([1, 2, 3, 2, 1])
>>> show()
```

What is the Matplotlib backend?

The backend chooses the rendering engine (vector or raster).
The most common is the anti-grain geometry library.
More backends are described in the Matplotlib FAQs.

Super simple example

```
./super_simple.py
```

Keyboard commands on Default plots

- Click “Pan and zoom” (or p); hold x and y with right or left mouse-buttons
- Click “Zoom to rectangle” (or o); then push left and right arrow keys for back/forwards.
- Click “Configure subplots” to control spacing.
- Ctrl-f for toggling fullscreen
- Mouse over axes + g: Toggle grid
- (Awesome) k and l log scale the axes!

Beginner's Examples

```
$ ./tex_on_plots.py
$ ./two_plots.py
$ ./polar_plot.py
$ ./daub4.py
$ ./fibonacci.py
```

Available Default Styles

```
$ python3 -q
>>> import matplotlib.pyplot as plt
>>> plt.style.available
['fivethirtyeight', 'dark_background', 'ggplot', 'bmh
    ', 'grayscale']
>>> exit()
$ ./wiggle_trace.py
```

For awesome examples, check out the Matplotlib gallery.

Interactive Matplotlib

Matplotlib becomes interactive via a call to `ion()`.
This makes calls to `show()` non-blocking.

```
$$ python
>>> import matplotlib.pyplot as plt
>>> plt.ion()
>>> plt.plot([1,2,3,2,1])
>>> plt.title(‘‘Hello’’)
>>> plt.title(‘‘This is a tent’’)
>>> plt.xlabel(‘‘X axis’’)
>>> plt.ylabel(‘‘Y axis’’)
>>> plt.ioff()
>>> plt.title(‘‘Goodbye’’)
```

On Mac, replace `python` by `ipython` or set your backend to `TkAgg` to avoid a known bug.

Matplotlib Response to Keyboard Clicks

```
./mpl_connect.py
```

Matplotlib Response to Keyboard Clicks

```
./point_plotter.py
```

From the Matplotlib Documentation

```
./line_builder.py
```

Three dimensional plots

```
./torus.py
```