Basic Plotting

Adam Richards

Galvanize, Inc

Last updated: September 15, 2017

Goals

1. Understand how figures, subplots, axes work together in matplotlib, seaborn, and pandas
2. Use plots and subplots effectively to explore a dataset
3. Distinguish between different categories on the same plot
4. Plot inside and outside of ipython notebooks
5. Understand MPL fundamentals well enough to learn effectively

For the morning, we will go through a gentle introduction to Matplotlib setting the stage for the afternoon assignment. The goal for this morning is to get more practice with EDA and Pandas.

matplotlib

The most frequently used plotting package in Python, matplotlib, is written in pure Python and is heavily dependent on NumPy.

- Plots should look great i.e. publication quality
- Text should look great (antialiased, etc.)
- Postscript output for inclusion with TEX documents
- Embeddable in a GUI for application development
- Code should be easy to understand and extend
- Making plots should be easy

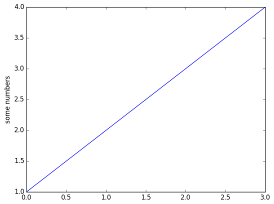
When using in publications or white papers $\rightarrow$ [1]

Matplotlib is conceptually divided into three parts:

- pylab interface (similar to MATLAB) - pylab tutorial
- Matplotlib frontend or API - artist tutorial
- backends - drawing devices or renderers

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.set_ylabel('some numbers')
ax.plot([1,2,3,4])
plt.show()
```



Note that the x-axis was automatically generated

On Jupyter and data visualization...

Who is in your audience?

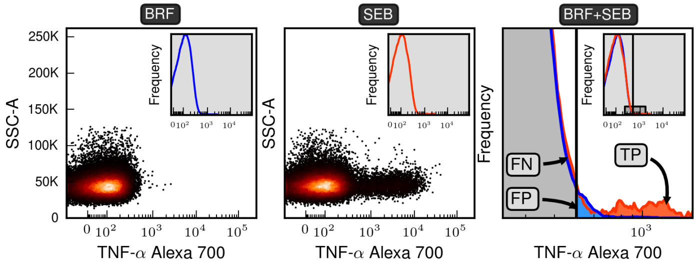
- How much customization do I need?
- How fast does it need to be done?
- Is it complicated?
- Version control, reproducibility

So many choices...

1. Environment - IPython, Jupyter, scripts, Sphinx, reportlab
2. Plotting tool - Pandas, Seaborn, Plotly, Bokeh, MPL-pylab, MPL-artist
3. Deliverable - webpage, report, presentation, white-paper, publication, dashboard

What is it that is I expect to see?

It can get complicated...

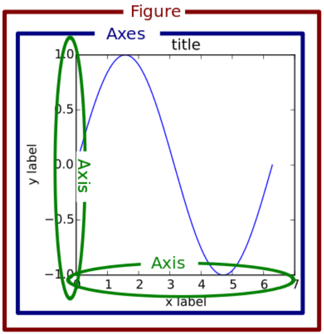


http://www.sciencedirect.com/science/article/pii/S0022175914001185

mpl basics

The shell...

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,6))
ax = plt.add_subplot(1,1,1)
...
ax.set_title('foo')
ax.set_ylabel('y')
ax.set_xlabel('x')
plt.savefig('foo.png',dpi=400)
```



If you are in Jupyter then use `%matplotlib inline`

Most backends support png, pdf, ps, eps and svg.

The supported file formats depend on the selected backend...

Backends

```
import matplotlib as mpl
mpl.use('PS')
```

Backend	Description
GTKAgg	Agg rendering to a GTK 2.x canvas (requires PyGTK and pycairo or cairocffi; Python2 only)
GTK3Agg	Agg rendering to a GTK 3.x canvas (requires PyGObject and pycairo or cairocffi)
GTK	GDK rendering to a GTK 2.x canvas (not recommended) (requires PyGTK and pycairo or cairocffi; Python2 only)
GTKCairo	Cairo rendering to a GTK 2.x canvas (requires PyGTK and pycairo or cairocffi; Python2 only)
GTK3Cairo	Cairo rendering to a GTK 3.x canvas (requires PyGObject and pycairo or cairocffi)
WXAgg	Agg rendering to to a wxWidgets canvas (requires wxPython)
WX	Native wxWidgets drawing to a wxWidgets Canvas (not recommended) (requires wxPython)
TkAgg	Agg rendering to a Tk canvas (requires TkInter)
Qt4Agg	Agg rendering to a Qt4 canvas (requires PyQt4 or `pyside`)
Qt5Agg	Agg rendering in a Qt5 canvas (requires PyQt5)
macosx	Cocoa rendering in OSX windows (presently lacks blocking show() behavior when matplotlib is in non-interactive mode)

Start simple and build

```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(8,4))
ax = fig.add_subplot(111)

ax.set_ylabel('something')
ax.set_title('something')

t = np.arange(0.0, 1.0, 0.01)
s = np.sin(2*np.pi*t)
line, = ax.plot(t, s, color='blue', lw=2)
plt.show()
```

Useful plotting functions

command	description
`plot`	plot lines and/or markers
`bar`	bar plot
`error bar`	error bar plot
`boxplot`	boxplot
`histogram`	histogram
`pie`	pie charts
`imshow`	heatmaps/images
`scatter`	scatter plots

The gallery will be your new friend

To the Notebooks

Goals

1. Understand how figures, subplots, axes work together in matplotlib, seaborn, and pandas
2. Use plots and subplots effectively to explore a dataset
3. Distinguish between different categories on the same plot
4. Plot inside and outside of ipython notebooks
5. Understand MPL fundamentals well enough to learn effectively

So much we can do with the axes

Plot 1

```
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(222)
ax4 = fig.add_subplot(222)
```

Plot 2

```
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(212)
```

Plot 3
Hint: `add_axes(left,bottom,width,height`

```
ax1 = fig.add_subplot(211)
ax2 = fig.add_axes([0.25,0.1,0.5,0.3])
```

Useful customization functions

command	description
`text`	add text to an axis
`table`	embed a table in the axes
`suptitle`	figure title
`ylim/xlim`	get/set the limits of x and y
`imshow`	heatmaps/images
`xticks/yticks`	get/set limits of tick locations
`tight_layout`	tries to make whitespace look right

style_sheets

```
with plt.style.context('fivethirtyeight'):
   plt.plot(x, np.sin(x) + x + np.random.randn(50))
   plt.plot(x, np.sin(x) + 0.5 * x + np.random.randn(50))
   plt.plot(x, np.sin(x) + 2 * x + np.random.randn(50))
```

or

```
plt.style.use('dark_background')
```

But be careful with the latter in Jupyter!

Higher level interfaces

- seaborn
- holoviews
- ggplot

Where do we go from here?

- 3D plots in MPL
- Interactive widgets in MPL
- Embedded plots
- Image analysis - PIL
- Mayavi - 3D data viz

Useful links

- list of plotting commands
- matplotlib howtos
- pylab tutorial
- artist tutorial
- FAQs

References I

J.D. Hunter, *Matplotlib: A 2D graphics environment*, Computing In Science & Engineering **9(3)** (2007), 90–95.