NPRs Python Notes

Nicholas P. Ross

September 30, 2020

Abstract

The is my (NPR's) set of `matplotlib` notes.

You will be able to find the latest version of these notes and indeed the .tex file at:
`https://github.com/d80b2t/Research_Notes`.

1 matplotib

From `http://matplotlib.org/`

matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB* or Mathematica), web application servers, and six graphical user interface toolkits.

matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code. For a sampling, see the screenshots, thumbnail gallery, and examples directory

For simple plotting the pyplot interface provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

The real introduction is here::
`https://www.labri.fr/perso/nrougier/teaching/matplotlib/`
and here::
`https://www.oreilly.com/learning/simple-line-plots-with-matplotlib`

From http://stackoverflow.com/questions/12987624/confusion-between-numpy-scipy-matplotlib-and-pylab:

- pylab is part of matplotlib (in matplotlib.pylab) and tries to give you a MatLab like environment. matplotlib has a number of dependencies, among them numpy which it imports under the common alias np. scipy is not a dependency of matplotlib.

- If you run ipython –pylab an automatic import will put all symbols from matplotlib.pylab into global scope. Like you wrote numpy gets imported under the npalias. Symbols from matplotlib are available under the mpl alias.

```
> ipython --pylab
```

```
import matplotlib.pyplot as plt
```

From http://stackoverflow.com/questions/16522380/python-pandas-plot-is-a-no-show: Put

```
import matplotlib.pyplot as plt
```

at the top, and

```
plt.show()
```

at the end.

1.1 matplotlibrc

ttps://matplotlib.org/users/customizing.html#matplotlibrc-sample
ttps://matplotlib.org/users/dflt_style_changes.htmlhttps://matplotlib.org/users/dflt_style_changes.ht
ttps://github.com/matplotlib/matplotlib/blob/master/matplotlibrc.templatehttps://github.com/ma

1.2 ColorMaps

https://www.youtube.com/watch?v=xAoljeRJ3lU (a.k.a. "introducing viridis").

2 Avoid-overlapping-in-scatterplot-with-2d-density

From:: Python Graph Gallery.

```
# Libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import kde

# Create data: 200 points
data = np.random.multivariate_normal([0, 0], [[1, 0.5], [0.5, 3]],
    200)
x, y = data.T

# Create a figure with 6 plot areas
fig, axes = plt.subplots(ncols=6, nrows=1, figsize=(21, 5))

# Everything sarts with a Scatterplot
axes[0].set_title('Scatterplot')
axes[0].plot(x, y, 'ko')
# As you can see there is a lot of overplottin here!

# Thus we can cut the plotting window in several hexbins
nbins = 20
axes[1].set_title('Hexbin')
axes[1].hexbin(x, y, gridsize=nbins, cmap=plt.cm.BuGn_r)

# 2D Histogram
axes[2].set_title('2D Histogram')
axes[2].hist2d(x, y, bins=nbins, cmap=plt.cm.BuGn_r)
```

```
# Evaluate a gaussian kde on a regular grid of nbins x nbins over
    data extents
k = kde.gaussian_kde(data.T)
xi, yi = np.mgrid[x.min():x.max():nbins*1j,
    y.min():y.max():nbins*1j]
zi = k(np.vstack([xi.flatten(), yi.flatten()]))

# plot a density
axes[3].set_title('Calculate Gaussian KDE')
axes[3].pcolormesh(xi, yi, zi.reshape(xi.shape), cmap=plt.cm.BuGn_r)

# add shading
axes[4].set_title('2D Density with shading')
axes[4].pcolormesh(xi, yi, zi.reshape(xi.shape), shading='gouraud',
    cmap=plt.cm.BuGn_r)

# contour
axes[5].set_title('Contour')
axes[5].pcolormesh(xi, yi, zi.reshape(xi.shape), shading='gouraud',
    cmap=plt.cm.BuGn_r)
axes[5].contour(xi, yi, zi.reshape(xi.shape) )
```

3 Good links

https://www.machinelearningplus.com/plots/top-50-matplotlib-visualizations-the-master-plots-python/