INSY 6500
Information Systems for Operations

Introduction to Matplotlib

Fall 2019

Jeffrey S. Smith

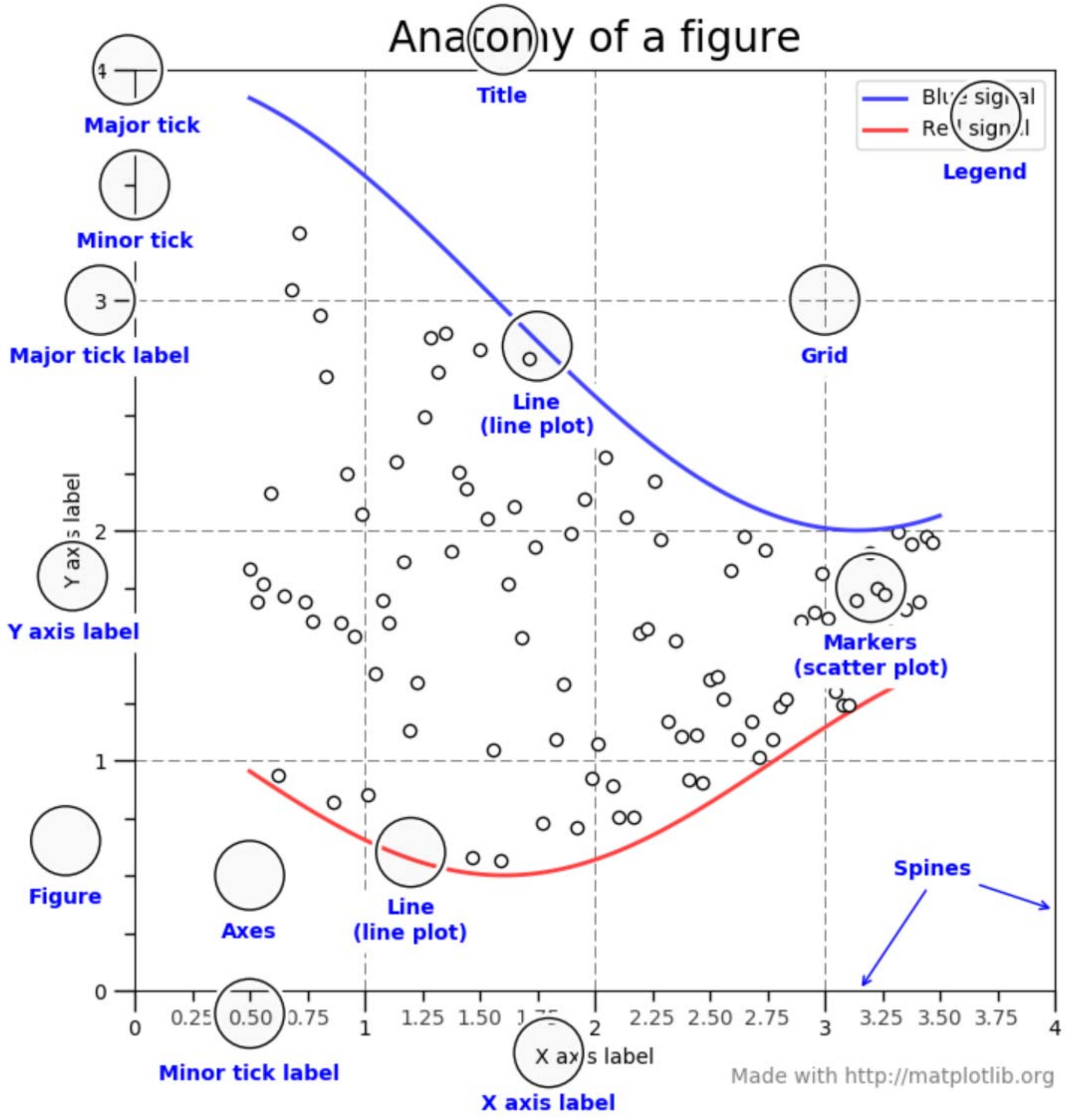
Industrial and Systems Engineering Department
Auburn University
jsmith@auburn.edu

What is Matplotlib?

Matplotlib is a multi-platform data visualization library built on NumPy arrays, and designed to work with the broader SciPy stack.

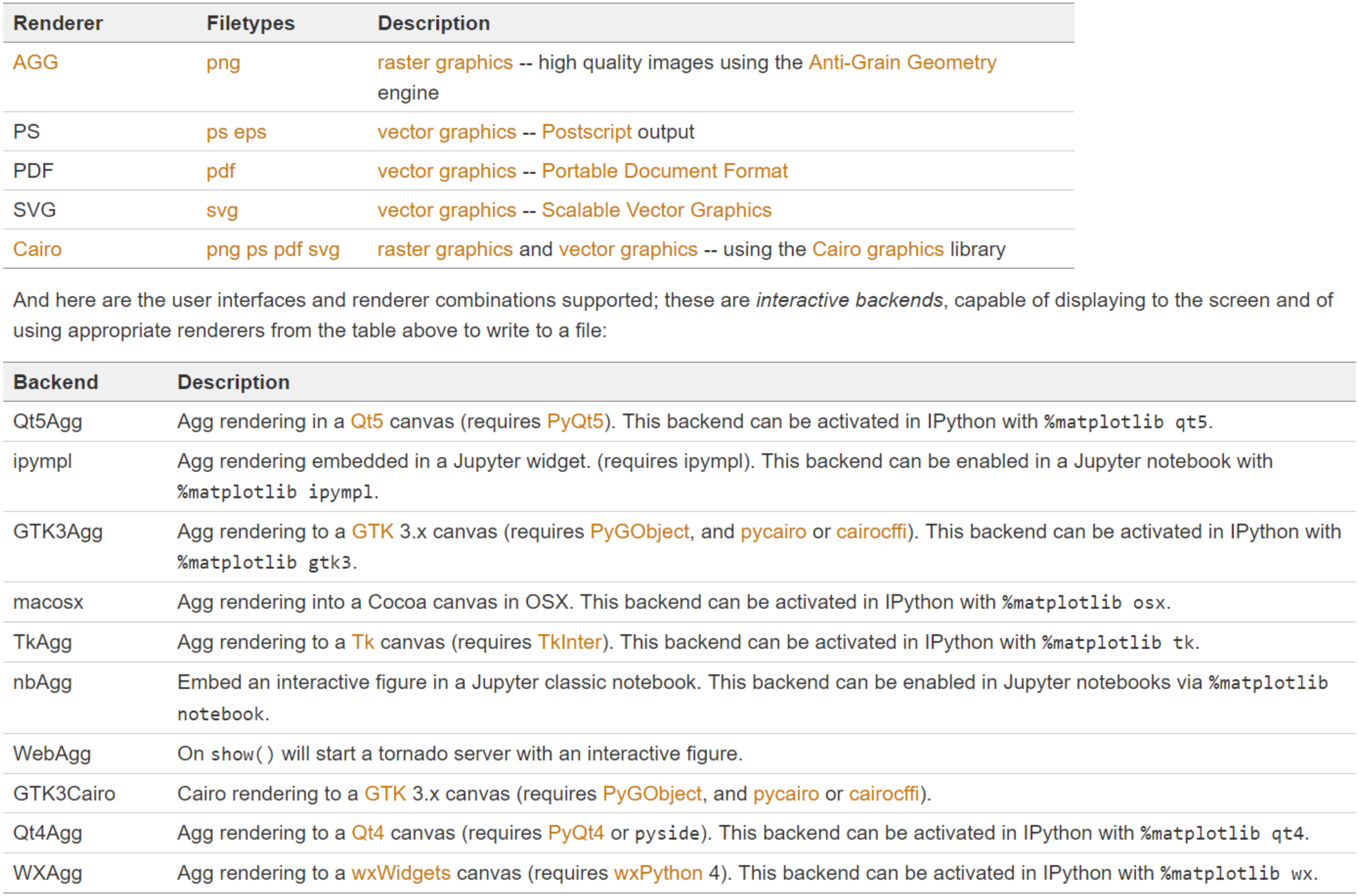
One of Matplotlib's most important features is its ability to play well with many operating systems and graphics backends. Matplotlib supports dozens of backends and output types, which means you can count on it to work regardless of which operating system you are using or which output format you wish.

VanderPlas, Notebook 04.00



https://matplotlib.org/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py

(next slide also)



Renderer	Filetypes	Description
AGG	png	raster graphics -- high quality images using the Anti-Grain Geometry engine
PS	ps eps	vector graphics -- Postscript output
PDF	pdf	vector graphics -- Portable Document Format
SVG	svg	vector graphics -- Scalable Vector Graphics
Cairo	png ps pdf svg	raster graphics and vector graphics -- using the Cairo graphics library

And here are the user interfaces and renderer combinations supported; these are *interactive backends*, capable of displaying to the screen and of using appropriate renderers from the table above to write to a file:

Backend	Description
Qt5Agg	Agg rendering in a Qt5 canvas (requires PyQt5). This backend can be activated in IPython with `%matplotlib qt5`.
ipympl	Agg rendering embedded in a Jupyter widget. (requires ipympl). This backend can be enabled in a Jupyter notebook with `%matplotlib ipympl`.
GTK3Agg	Agg rendering to a GTK 3.x canvas (requires PyGObject, and pycairo or cairocffi). This backend can be activated in IPython with `%matplotlib gtk3`.
macosx	Agg rendering into a Cocoa canvas in OSX. This backend can be activated in IPython with `%matplotlib osx`.
TkAgg	Agg rendering to a Tk canvas (requires TkInter). This backend can be activated in IPython with `%matplotlib tk`.
nbAgg	Embed an interactive figure in a Jupyter classic notebook. This backend can be enabled in Jupyter notebooks via `%matplotlib notebook`.
WebAgg	On `show()` will start a tornado server with an interactive figure.
GTK3Cairo	Cairo rendering to a GTK 3.x canvas (requires PyGObject, and pycairo or cairocffi).
Qt4Agg	Agg rendering to a Qt4 canvas (requires PyQt4 or `pyside`). This backend can be activated in IPython with `%matplotlib qt4`.
WXAgg	Agg rendering to a wxWidgets canvas (requires wxPython 4). This backend can be activated in IPython with `%matplotlib wx`.

Our Matplotlib-related Resources

- Course Jupyter Notebook(s?)
  - 3.6.0 – Introduction to Matplotlib.ipynb
- VanderPlas Jupyter Notebooks
  - GitHub address: https://github.com/jakevdp/PythonDataScienceHandbook
  - Notebooks 04.00 – 04.15
- Matplotlib web page: https://matplotlib.org/
  - User Guide: https://matplotlib.org/users/index.html
  - Examples: https://matplotlib.org/gallery/index.html - Note that you can download the entire gallery of examples in Jupyter Notebook format.

Two Matplotlib Environments/Interfaces

- Matlab-style “stateful” interface
- Pyplot (plt) functions are used to add plot elements to the current *axes* in the current *figure*.

```
plt.figure()
plt.subplot(2, 1, 1)
plt.plot(x, np.sin(x))

plt.subplot(2, 1, 2)
plt.plot(x, np.cos(x))
plt.title("Cosine Function")
```

- Object-oriented interface – plotting functions are *methods* of explicit Figure and Axes objects.

```
fig, ax = plt.subplots(2)

ax[0].plot(x, np.sin(x))
ax[1].plot(x, np.cos(x))
ax[0].set_title('Sine function')
fig.savefig('first_figure.png')
```