matplotlib
Intro to matplotlib

- Matplotlib is the standard plotting library for scientific python
  - **Design objectives** (from the matplotlib documentation):
    - Plots should look great - publication quality. One important requirement for me is that the text looks good (antialiased, etc.)
    - Postscript output for inclusion with TeX documents
    - Embeddable in a graphical user interface for application development
    - Code should be easy enough that I can understand it and extend it
    - Making plots should be easy
  - Mostly it is for 2-d data (including surface plots of f(x,y), etc.)
  - Active development with lots of new features
  - Best way to figure out how to do something: look at the gallery
Importing

- There are several interfaces to matplotlib that provide varying amounts of access to its underlying functionality
  - See http://matplotlib.org/faq/usage_faq.html
  - matplotlib is the entire package
  - matplotlib.pyplot is a module within matplotlib that provides easy access to the core plotting routines
  - pylab combines pyplot and numpy into a single namespace to give a MatLab like interface
    - This is best for interactive work
- A number of toolkits extend the functionality
  - basemap and cartopy: mapping (e.g. projecting onto a globe, geographical boundaries)
  - mplot3d: basic 3-d plotting
  - AxesGrid: high-level methods for arranging multiple plots together in a figure
Figures vs. Axes

- Figures are the highest level object and can include multiple axes (see [http://matplotlib.org/users/pyplot_tutorial.html](http://matplotlib.org/users/pyplot_tutorial.html))
  - There are many matplotlib routines to subdivide a figure into multiple subplots
Backends

- matplotlib can output to a number of different devices—the backends provide this functionality

- Interactive backends:
  - pygtk, wxpython, tkinter, q3, macosx
  - These allow for plotting to the screen, and updates with each command (if desired)

- Hardcopy backends:
  - PNG, SVG, PDF, PS

- To select a backend:

  ```python
  import matplotlib
  matplotlib.use('PS')
  import matplotlib.pyplot
  ```
IPython and matplotlib

- IPython supports matplotlib:
  - `ipython --pylab` will pop up a window in interactive mode
  - `%pylab inline` magic in notebooks
There are a lot of new projects, some built upon matplotlib, others independent.

A common goal for a lot of these is to allow for interactive data exploration in the web browser. Many use the javascript library d3.js to do this.

Examples:

- mpld3: http://mpld3.github.io/ (based on matplotlib; see his blog post here: http://jakevdp.github.io/blog/2014/01/10/d3-plugins-truly-interactive/)
- plot.ly: https://plot.ly/
- D3PO: http://d3po.org/
- d3py: https://github.com/mikedewar/d3py (inactive?)
- Seaborn: http://web.stanford.edu/~mwaskom/software/seaborn/ (based on matplotlib)
- ggplot: https://github.com/yhat/ggplot/ (for you R users)
Some Examples

- There are far more examples than we can cover—we'll see more as the class goes on.