Outline

1. Introduction and Motivation
2. Getting Started
3. 3D
4. Flight Work and Other Interesting Concepts
The first step in using data is understanding it.

Numbers are complicated and ugly. Colors are pretty.

Properly visualized data is effective communication on its own.

- A scientific paper with well-crafted figures is much more effective than one with dreaded "Data Appendices"
Consider something like this...

**Figure:** Lookin’ good.
...over its original form.

Figure: YIKES
Data Comes in Many Forms

- **CSV**
  - Use native `csv` library from Python
  - Simple, robust
  - Stands for Comma Separated Values
  - Can also read Tab-Delimited Files

- **Excel Spreadsheets**
  - Install with: `pip install xlrd`
  - Plays nicely with the Excel models of Books, Sheets, and Cells

- **Fixed Width Data Files**
  - Use native `struct` library from Python
  - Similar to CSVs but lacking a specific data separator.
  - Implemented in C rather than Python (Cython): very fast!

- **JSON**
  - Use native `requests` library from Python
  - Get data straight from the web.
Simple types of plots to plot

1. `plot()` is a marked scatter plot with the individual data points unenumerated by default.
2. `bar()` is a bar plot.
3. `hist()` is a histogram bar plot.
4. `hbar()` is a *horizontal* bar plot.
5. `boxplot()` is a box and whisker plot.
6. `scatter()` is a scatter plot with line markings turned off by default.
Methods of changing the appearance of a plot

1. `subplot(int x)` allows you to choose a section of a figure that you want to plot on. For example, `subplot(311)` means that you have a 3-row 1-column plot and you will plot in the 1st (top) section.

2. `title()` gives the graph a title.

3. `xlim()`, `ylim()` allow for the setting of the ranges of the axes.

4. `xticks()`, `yticks()` allow for the placement of tick marks and labels on the graph’s axes.

5. `legend()` generates a legend for your graph. You can specify names for the plotted figures in plotting order or use labels passed in at the time of plotting.

6. `annotate()` allows for the highlighting of a specific value or region.
Unlocking Your Full Matplotlib Potential

- This goes much deeper than the above.
- Visit matplotlib.org to check out all optional parameters for each of the above functions.
  - color and colormaps
  - thickness
  - background coloring
  - location on plot
  - formatting modes
Removing outliers

- If you know what behavior your data should follow, you can remove outliers to make the picture better.

Smoothing

- Sometimes in data presentation, it’s better to show the big idea rather than all the minute details.
- Can use median filters (`matplotlib.signal.medfilt()`) or averaging boxes (`convolve()`).
Don’t misrepresent your data! Use the previous tricks to clarify rather than obfuscate.
1. Introduction and Motivation
2. Getting Started
3. 3D
4. Flight Work and Other Interesting Concepts
3D Plotting

- Use `mpl_toolkits.mplot3d`, which features the following classes:
  1. `axes3d` is a 3D plotting library that works very similar to typical `matplotlib` 2D plotting
  2. `axis3d` is an outdated 3D plotting library that apparently suffers from being buggy and poorly designed. Avoid!
  3. `art3d` is a 3D art class which is used to build components of `axes3d`, but has some interesting features of its own right.
  4. `proj3d` is the background class for these others.

- When plotting in 3D, you must always be careful to specify your dimensions.
Axes3D.plot() gives a marked scatter
Axes3D.scatter() gives an unmarked scatter plot
Axes3D.plot_wireframe() plots a transparent mesh of a surface.
Axes3D.plot_surface() plots a solid surface
Axes3D.plot_trisurf() plots a solid surface made from a Triangulation object
Axes3D.contour() plots a 3D contour
Others, like quivers, 2D plots, bar plots, polygon plots.
Outline

1. Introduction and Motivation
2. Getting Started
3. 3D
4. Flight Work and Other Interesting Concepts
There are many other projects and implementations that you can consider incorporating into your data visualization.

scikits.audiolab allows you to analyze sound files and plot their frequencies.

- `pip install scikits.audiolab`

If you’re feeling confident with your HTML and JSON vocabulary, you can look into Google Visualization API for plotting to the web.

Basemap is a library that makes coordinate generation easy. It’s great if you’re looking to plot with respect to space.

PIL and Images (as you might remember) are excellent libraries for reading in images and using them as data.
Now we can take a page out of FiveThirtyEight’s book.

We can download some airline delay data from http://www.transtats.bts.gov/ and play around