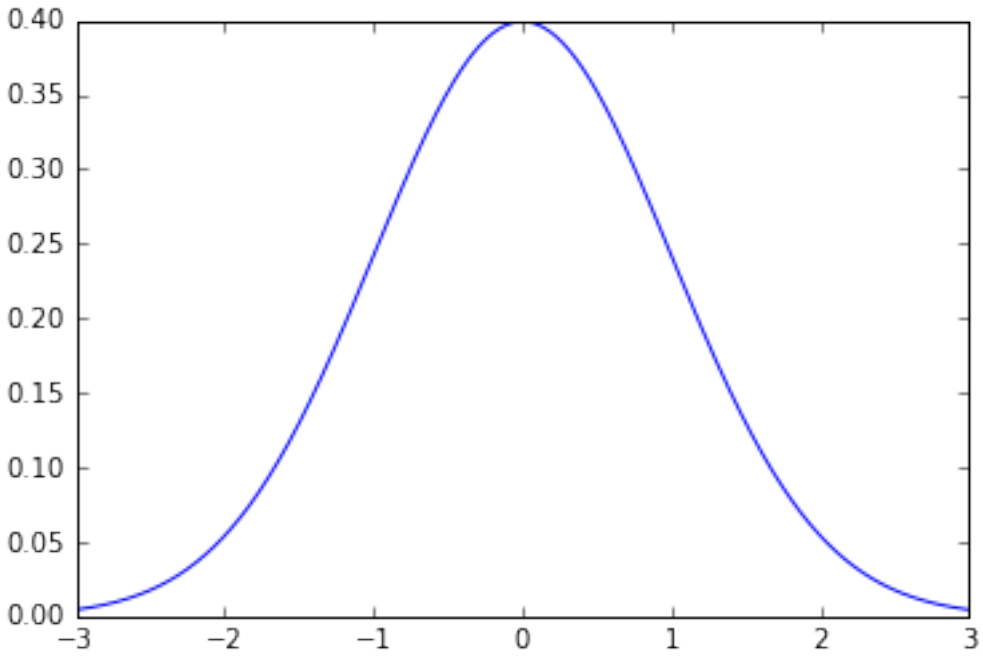
MatPlotLib

December 20, 2016

1 MatPlotLib Basics

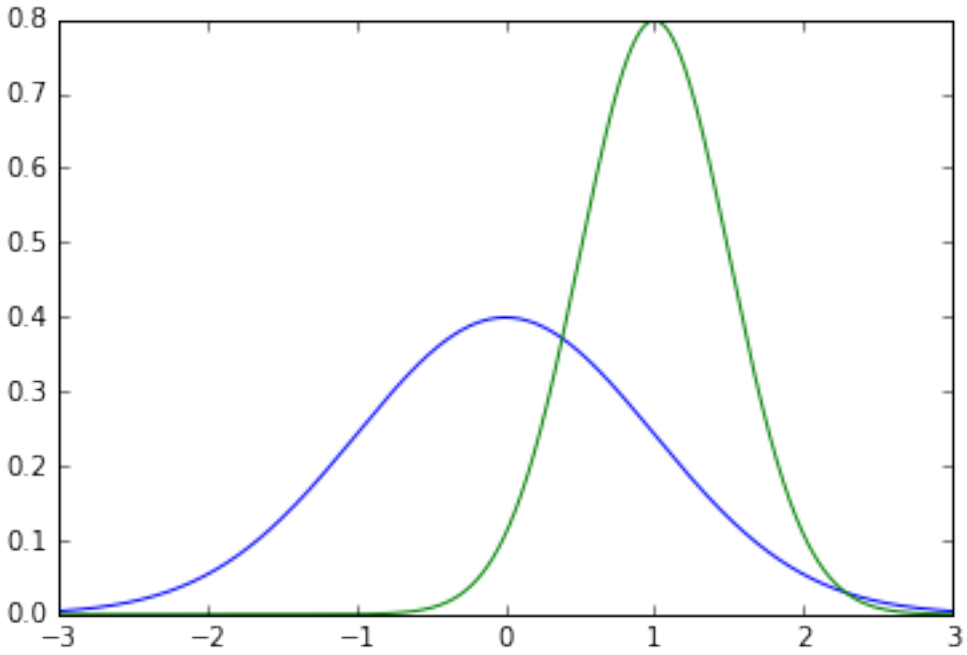
1.1 Draw a line graph

```
In [1]: %matplotlib inline
        from scipy.stats import norm
        import matplotlib.pyplot as plt
        import numpy as np

        x = np.arange(-3, 3, 0.01)

        plt.plot(x, norm.pdf(x))
        plt.show()
```



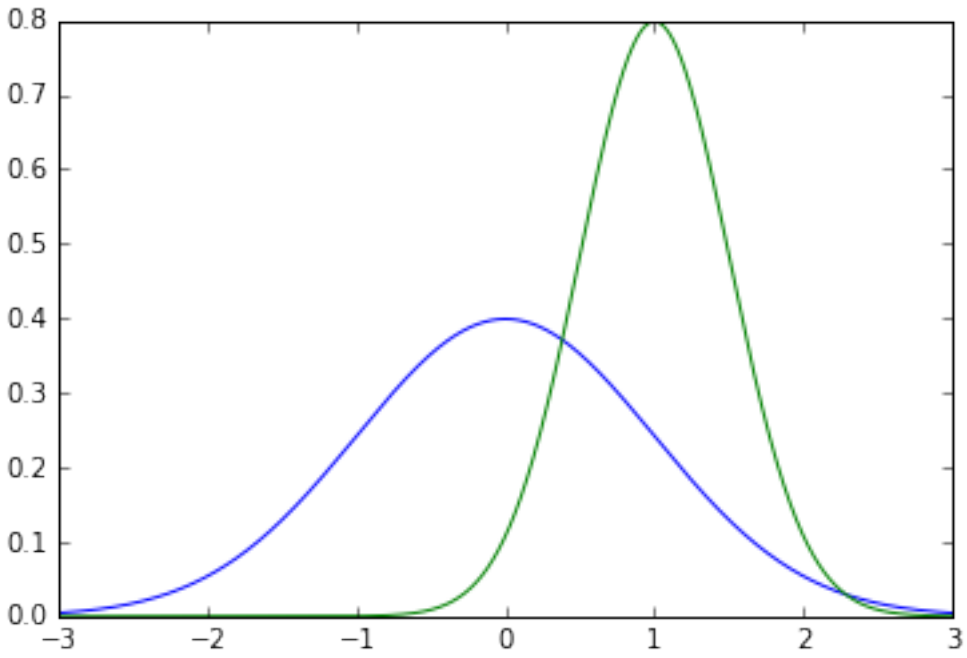
1.2 Mutiple Plots on One Graph

```
In [3]: plt.plot(x, norm.pdf(x))
        plt.plot(x, norm.pdf(x, 1.0, 0.5))
        plt.show()
```



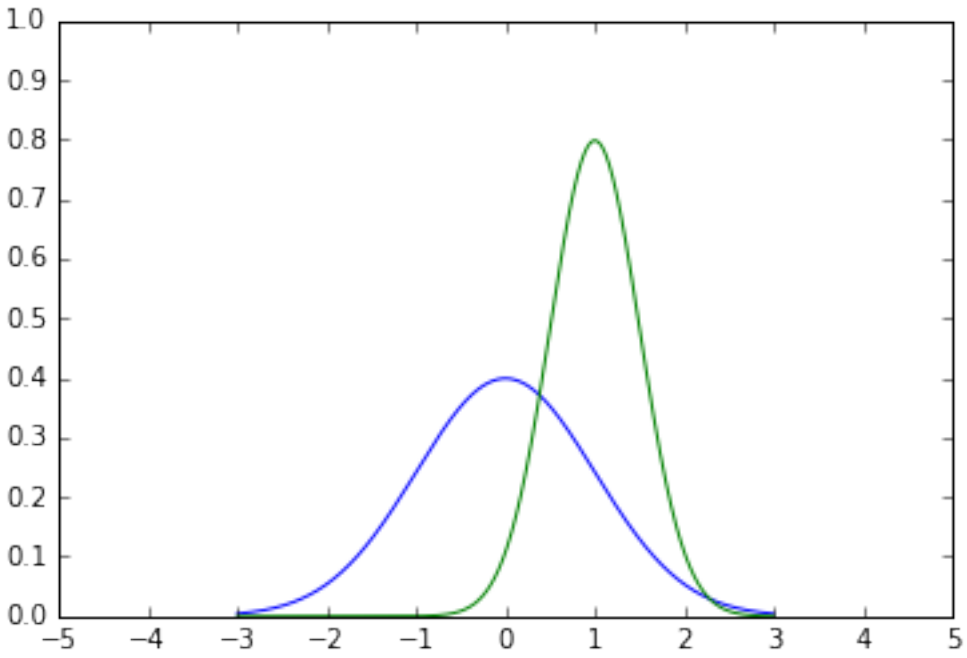
1.3 Save it to a File

```
In [4]: plt.plot(x, norm.pdf(x))
        plt.plot(x, norm.pdf(x, 1.0, 0.5))
        plt.savefig('C:\\Users\\Frank\\MyPlot.png', format='png')
```



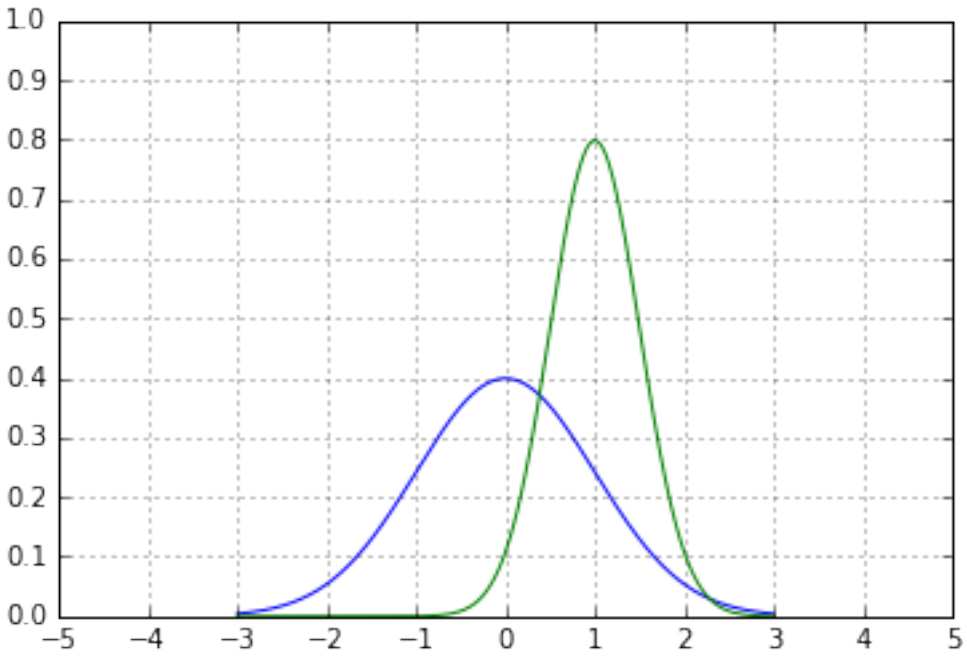
1.4 Adjust the Axes

```
In [5]: axes = plt.axes()
        axes.set_xlim([-5, 5])
        axes.set_ylim([0, 1.0])
        axes.set_xticks([-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5])
        axes.set_yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
        plt.plot(x, norm.pdf(x))
        plt.plot(x, norm.pdf(x, 1.0, 0.5))
        plt.show()
```



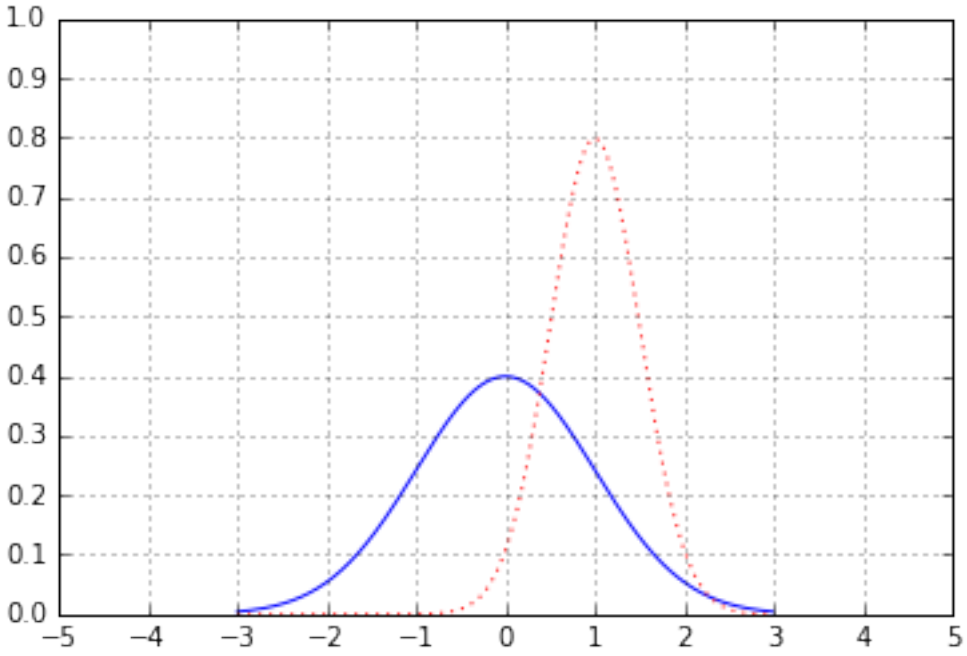
1.5 Add a Grid

```
In [6]: axes = plt.axes()
        axes.set_xlim([-5, 5])
        axes.set_ylim([0, 1.0])
        axes.set_xticks([-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5])
        axes.set_yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
        axes.grid()
        plt.plot(x, norm.pdf(x))
        plt.plot(x, norm.pdf(x, 1.0, 0.5))
        plt.show()
```



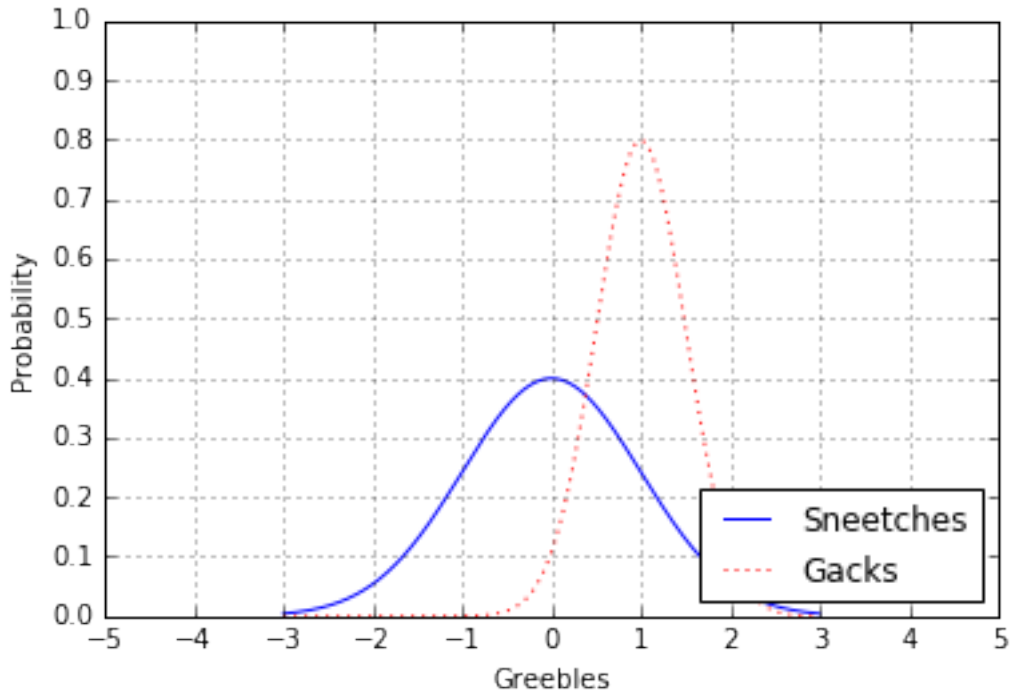
1.6 Change Line Types and Colors

```
In [3]: axes = plt.axes()
        axes.set_xlim([-5, 5])
        axes.set_ylim([0, 1.0])
        axes.set_xticks([-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5])
        axes.set_yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
        axes.grid()
        plt.plot(x, norm.pdf(x), 'b-')
        plt.plot(x, norm.pdf(x, 1.0, 0.5), 'r:')
        plt.show()
```



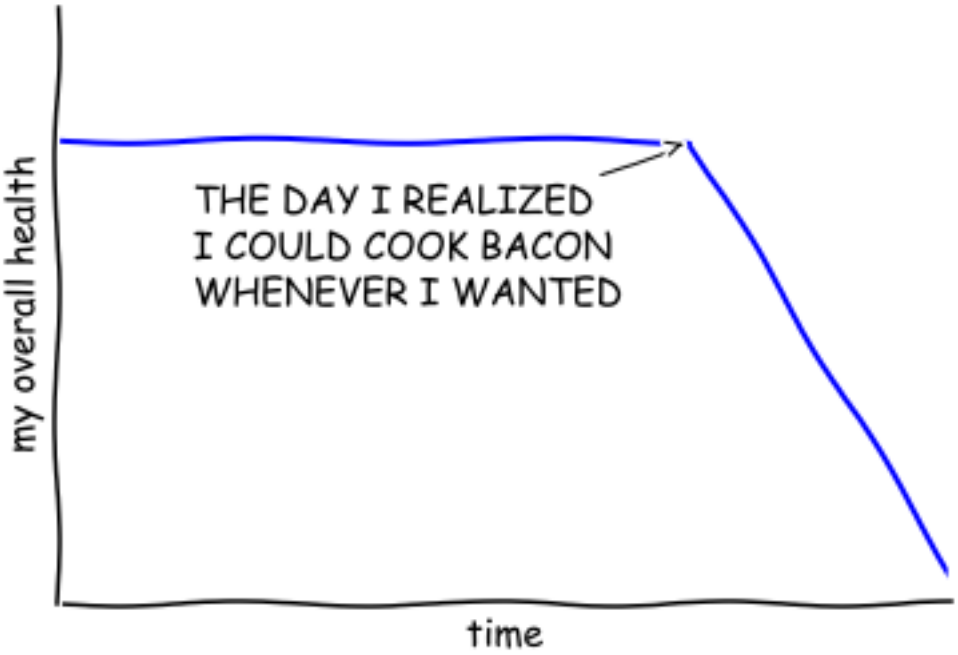
1.7 Labeling Axes and Adding a Legend

```
In [8]: axes = plt.axes()
        axes.set_xlim([-5, 5])
        axes.set_ylim([0, 1.0])
        axes.set_xticks([-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5])
        axes.set_yticks([0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
        axes.grid()
        plt.xlabel('Greebles')
        plt.ylabel('Probability')
        plt.plot(x, norm.pdf(x), 'b-')
        plt.plot(x, norm.pdf(x, 1.0, 0.5), 'r:')
        plt.legend(['Sneetches', 'Gacks'], loc=4)
        plt.show()
```



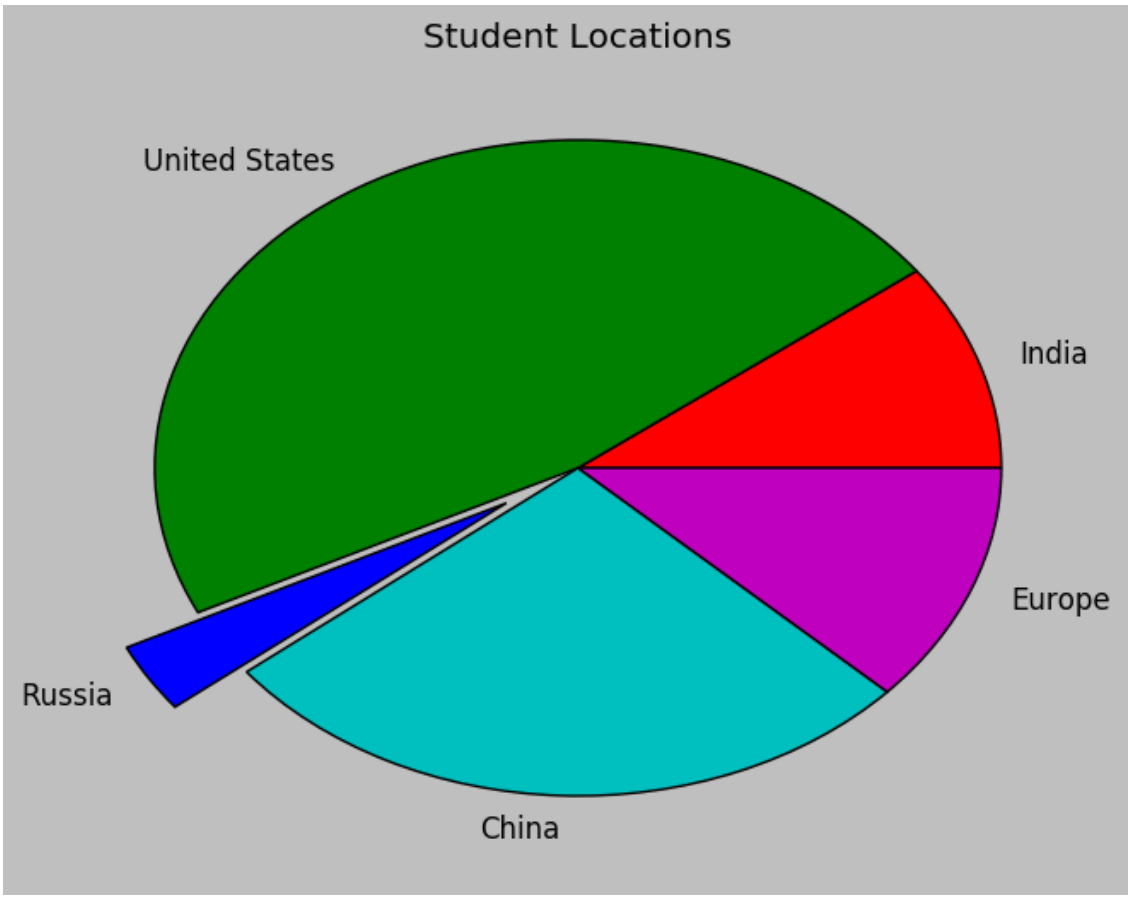
1.8 XKCD Style :)

```
In [9]: plt.xkcd()

        fig = plt.figure()
        ax = fig.add_subplot(1, 1, 1)
        ax.spines['right'].set_color('none')
        ax.spines['top'].set_color('none')
        plt.xticks([])
        plt.yticks([])
        ax.set_ylim([-30, 10])

        data = np.ones(100)
        data[70:] -= np.arange(30)

        plt.annotate(
            'THE DAY I REALIZED\nI COULD COOK BACON\nWHENEVER I WANTED',
            xy=(70, 1), arrowprops=dict(arrowstyle='->'), xytext=(15, -10))

        plt.plot(data)

        plt.xlabel('time')
        plt.ylabel('my overall health')

Out[9]: <matplotlib.text.Text at 0xa84cdd8>
```



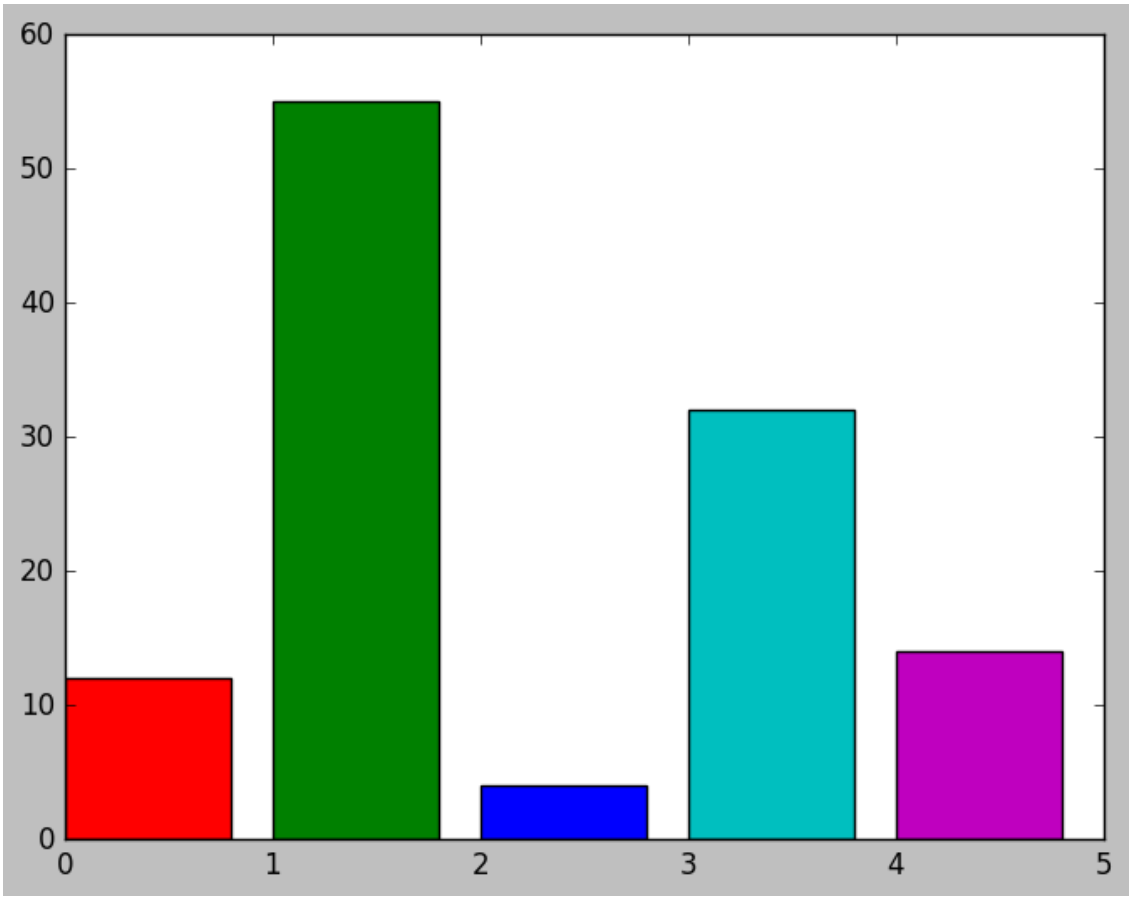
1.9 Pie Chart

```
In [10]: # Remove XKCD mode:
         plt.rcdefaults()

         values = [12, 55, 4, 32, 14]
         colors = ['r', 'g', 'b', 'c', 'm']
         explode = [0, 0, 0.2, 0, 0]
         labels = ['India', 'United States', 'Russia', 'China', 'Europe']
         plt.pie(values, colors= colors, labels=labels, explode = explode)
         plt.title('Student Locations')
         plt.show()
```



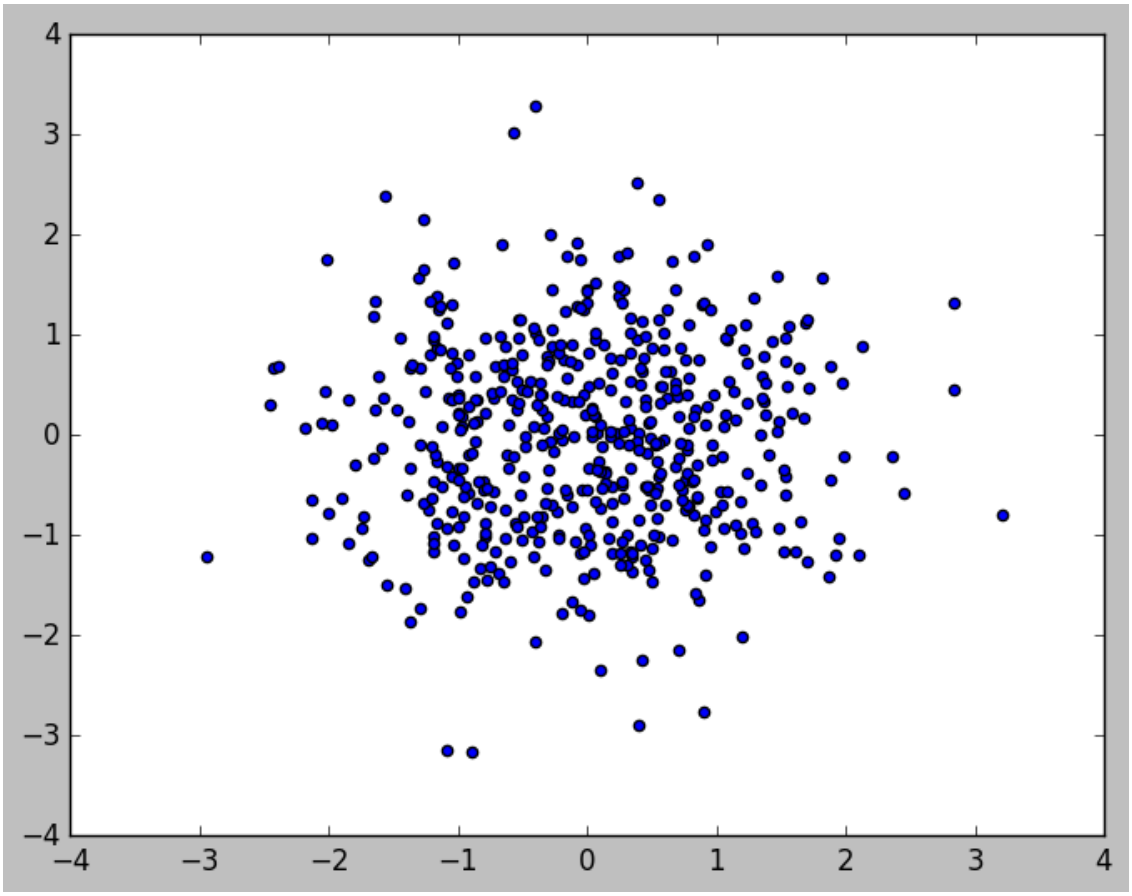
1.10 Bar Chart

```
In [11]: values = [12, 55, 4, 32, 14]
         colors = ['r', 'g', 'b', 'c', 'm']
         plt.bar(range(0,5), values, color= colors)
         plt.show()
```



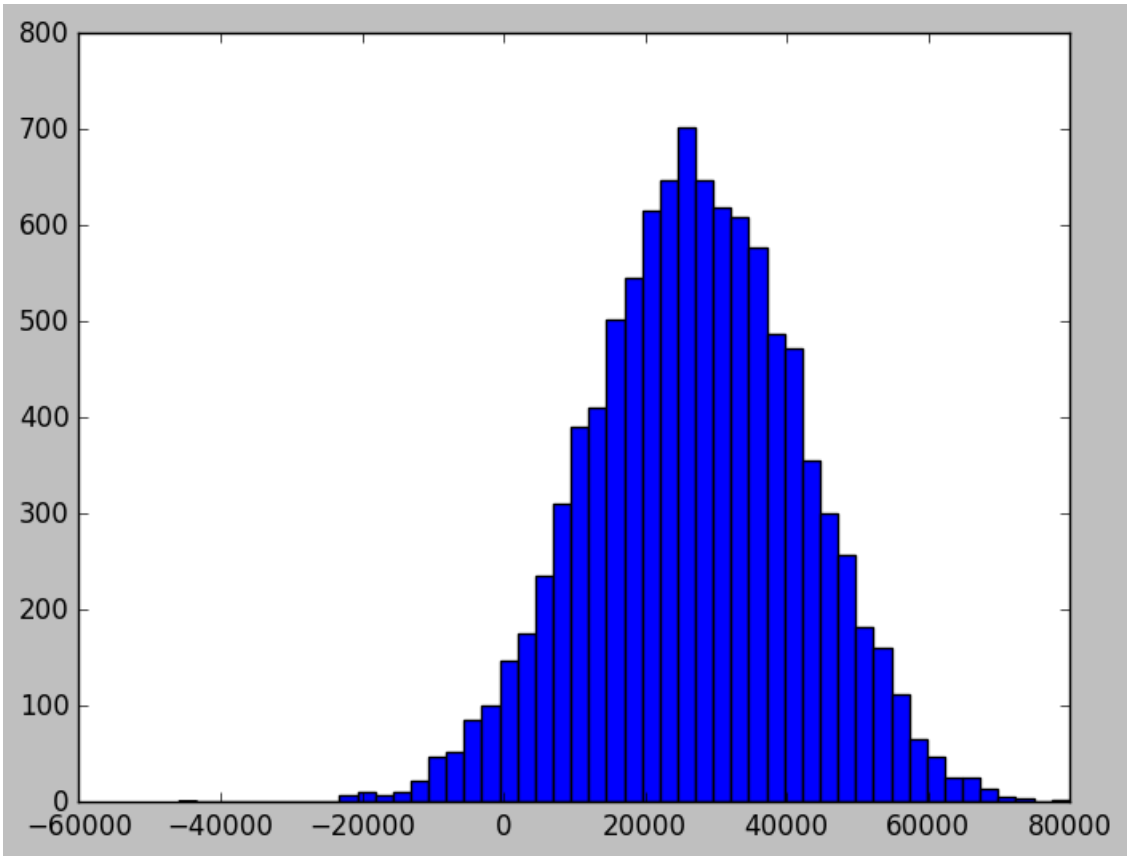
1.11 Scatter Plot

```
In [14]: from pylab import randn

         X = randn(500)
         Y = randn(500)
         plt.scatter(X,Y)
         plt.show()
```



1.12 Histogram

```
In [15]: incomes = np.random.normal(27000, 15000, 10000)
         plt.hist(incomes, 50)
         plt.show()
```



1.13 Box & Whisker Plot

Useful for visualizing the spread & skew of data.

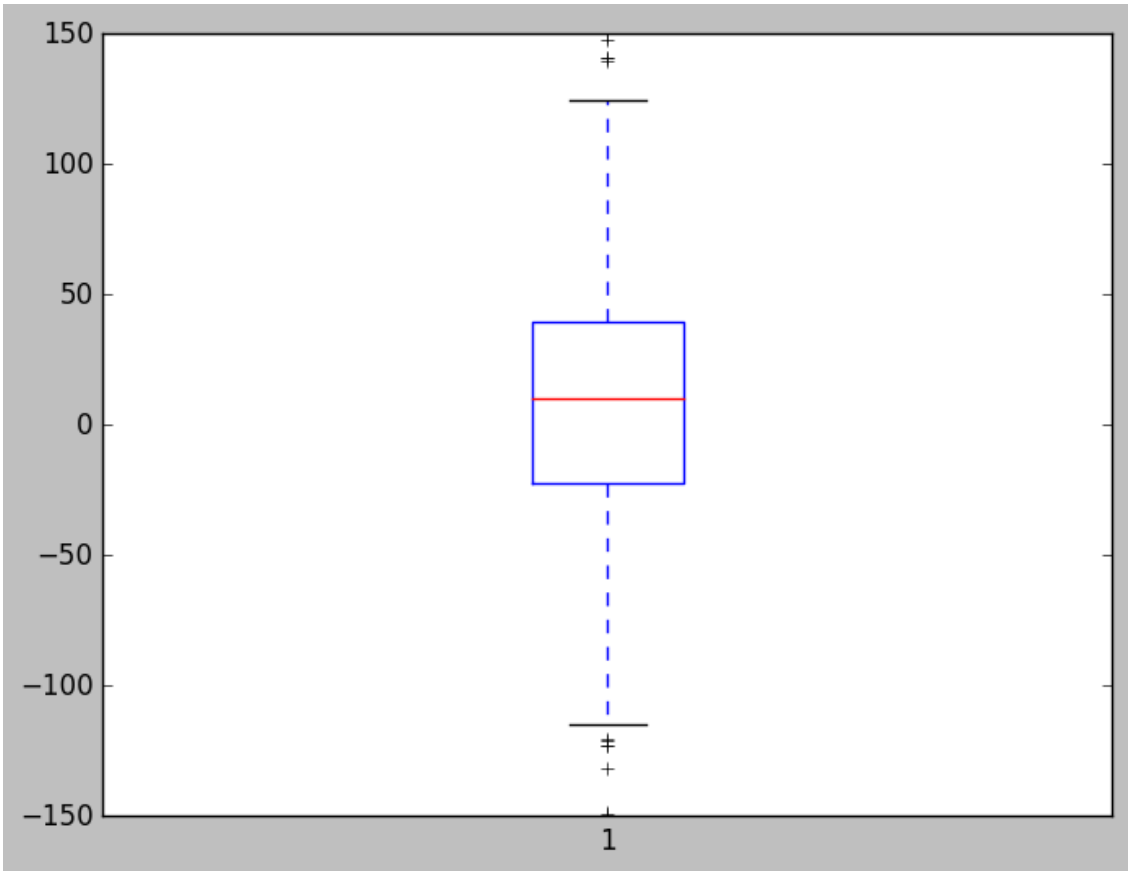
The red line represents the median of the data, and the box represents the bounds of the 1st and 3rd quartiles.

So, half of the data exists within the box.

The dotted-line "whiskers" indicate the range of the data - except for outliers, which are plotted outside the whiskers. Outliers are 1.5X or more the interquartile range.

This example below creates uniformly distributed random numbers between -40 and 60, plus a few outliers above 100 and below -100:

```
In [16]: uniformSkewed = np.random.rand(100) * 100 - 40
         high_outliers = np.random.rand(10) * 50 + 100
         low_outliers = np.random.rand(10) * -50 - 100
         data = np.concatenate((uniformSkewed, high_outliers, low_outliers))
         plt.boxplot(data)
         plt.show()
```



1.14 Activity

Try creating a scatter plot representing random data on age vs. time spent watching TV. Label the axes.

In [ ]: