### Visualization

UW CSE 160 Winter 2017

# matplotlib

- Strives to emulate MATLAB
  - Pro: familiar to MATLAB users
  - Pro: powerful
  - Con: not the best design for a plotting library
- One important function for HW6:

```
plot(xvalues, yvalues)
```

## Plot

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

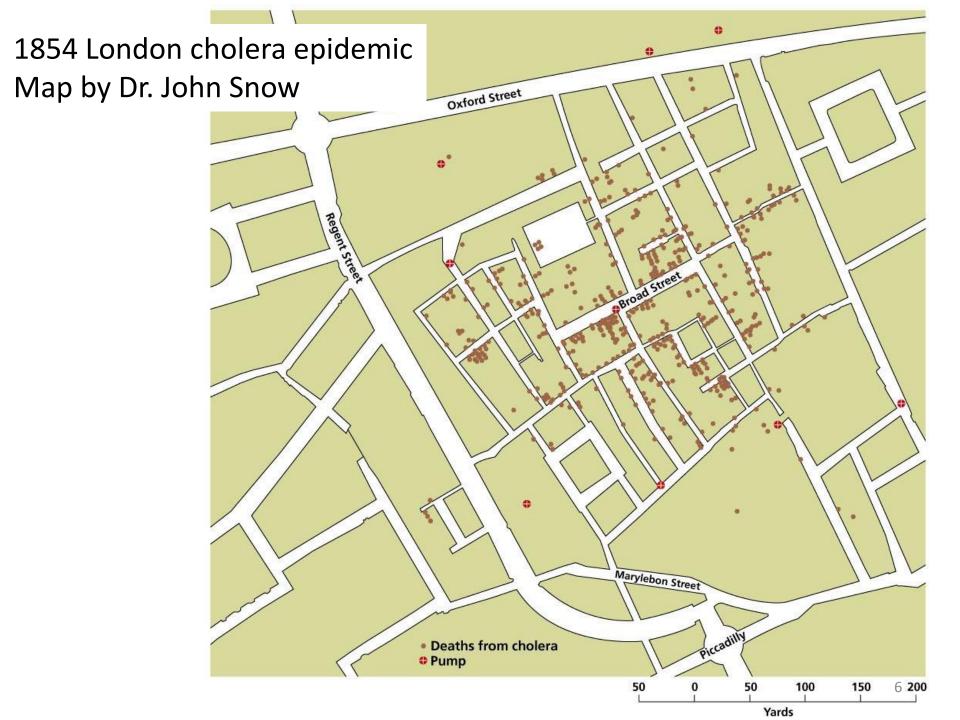
Has a side effect on the figure (like "print" statement)

import matplotlib.pyplot as plt

```
xs = range(-100, 110, 10)
x^2 = [x^* + 2 \text{ for } x \text{ in } xs]
neqx2 = [-x**2 \text{ for } x \text{ in } xs]
plt.plot(xs, x2)
plt.plot(xs, negx2)
plt.xlabel("x")
                                          Incrementally
plt.ylabel("y")
                                          modify the figure.
plt.ylim(-2000, 2000)
plt.axhline(0) # horiz line
plt.axvline(0) # vert line
plt.savefig("quad.png")
                                         Save your figure to a file
plt.show()
                                          Show it on the screen
```

```
def myplot(xs, ys, description):
  plt.plot(xs, ys, linewidth=2, color='green', linestyle='-', marker='s', label=description)
def setup_plot():
 plt.xlabel("x")
  plt.ylabel("y")
  plt.axhline(0,linestyle=':',color='red')
  plt.axvline(0,linestyle=':',color='red')
def finish_plot():
  plt.legend()
  plt.show()
setup_plot()
myplot(xs,x2,"x**2")
finish_plot()
setup_plot()
myplot(xs,negx2,"-x**2")
finish_plot()
```

We can group these options into functions as usual, but remember that they are operating on a global, hidden variable (the figure)



## Napoleon's Russian Campaign of 1812

#### Graphic by Charles Joseph Minard

