

MIT OpenCourseWare
<http://ocw.mit.edu>

6.00 Introduction to Computer Science and Programming
Fall 2008

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

6.00 Handout, Lecture 21 (Not intended to make sense outside of lecture)

```
def getSpringData(fname):
    springData = open(fname, 'r')
    distances = []
    forces = []
    for line in springData:
        if line[0] == '#': continue
        line = line[:-1]
        elems = line.rsplit(':')
        distances.append(float(elems[0]))
        forces.append(float(elems[1]))
    return pylab.array(distances), pylab.array(forces)
```

```
distances, forces = getSpringData('springData.txt')
pylab.scatter(distances, forces)
pylab.xlabel('Distance (Meters)')
pylab.ylabel('|Force| (Newtons)')
pylab.title('Force vs. Distance for Spring')
```

```
k, b = pylab.polyfit(distances, forces, 1)
yVals = k*distances + b
pylab.plot(distances, yVals, c = 'r', linewidth = 2)
pylab.title('Force vs. Distance, k = ' + str(k))
```

```
a, b, c = pylab.polyfit(times, speeds, 2)
yVals = a*(times**2) + b*times + c
pylab.plot(times, yVals, c = 'y', linewidth = 4)
```

$R^2 = 1 - EE/DV$, EE is errors in estimation and DV variance in data

EE computed by comparing estimates to measured data

DV computed by comparing mean of measured data to measure data

```
def rSquare(measured, estimated):
    diffs = (estimated - measured)**2
    mMean = measured.sum()/float(len(measured))
    var = (mMean - measured)**2
    return 1 - diffs.sum()/var.sum()
```

