6.00 Handout, Lecture 18
(Not intended to make sense outside of lecture)

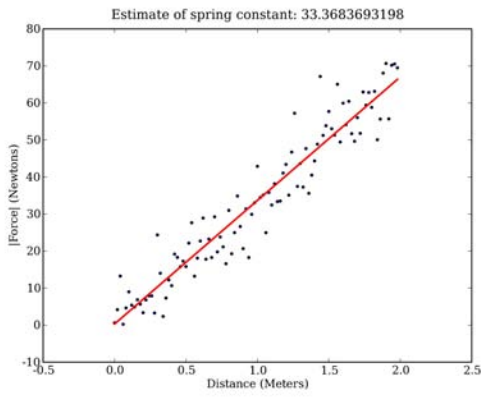
```
springData = open('springData.txt', 'r')
distances = []
forces = []
for line in springData:
    line = line[:-1]
    elems = line.rsplit(':')
    distances.append(float(elems[0]))
    forces.append(float(elems[1]))
```

```
distances = pylab.array(distances)
pylab.scatter(distances, forces)
pylab.xlabel('Distance (Meters)')
pylab.ylabel('|Force| (Newtons)')
a, b = pylab.polyfit(distances, forces, 1)
yVals = a*distances + b
pylab.plot(distances, yVals, c = 'r', linewidth = 2)
pylab.title('Estimate of spring constant: ' + str(a))
print 'Linear fit to linear data:', rSquare(forces, yVals)
```

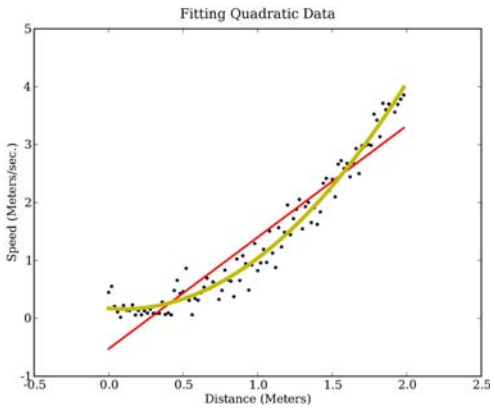
```
distances = pylab.array(distances)
pylab.figure()
pylab.scatter(distances, speed)
pylab.xlabel('Distance (Meters)')
pylab.ylabel('Speed (Meters/sec.)')
a, b = pylab.polyfit(distances, speed, 1)
yVals = a*distances + b
pylab.plot(distances, yVals, c = 'r', linewidth = 2)
print 'Linear fit to quadratic data:', rSquare(speed, yVals)
```

```
a, b, c = pylab.polyfit(distances, speed, 2)
yVals = a*(distances**2) + b*distances + c
pylab.plot(distances, yVals, c = 'y', linewidth = 4)
pylab.title('Fitting Quadratic to Quadratic Data')
print 'Quadratic fit to quadratic data:', rSquare(speed, yVals)
```

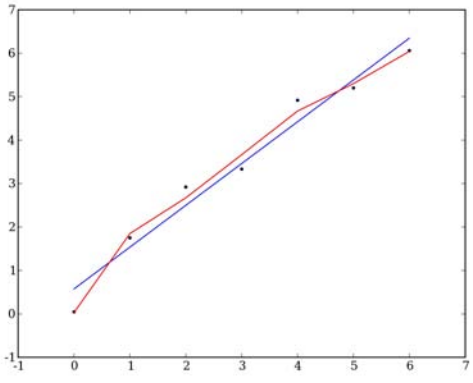
```
def rSquare(measured, estimated):
    m = pylab.array(measured)
    e = pylab.array(estimated)
    mMean = m.sum()/float(len(m))
    diffs = (e - m)**2
    var = (mMean - m)**2
    return 1 - diffs.sum()/var.sum()
```



$R^2 = .915$



$R^2 = .896$, $R^2 = .971$



$R^2 = .980$, $R^2 = .991$