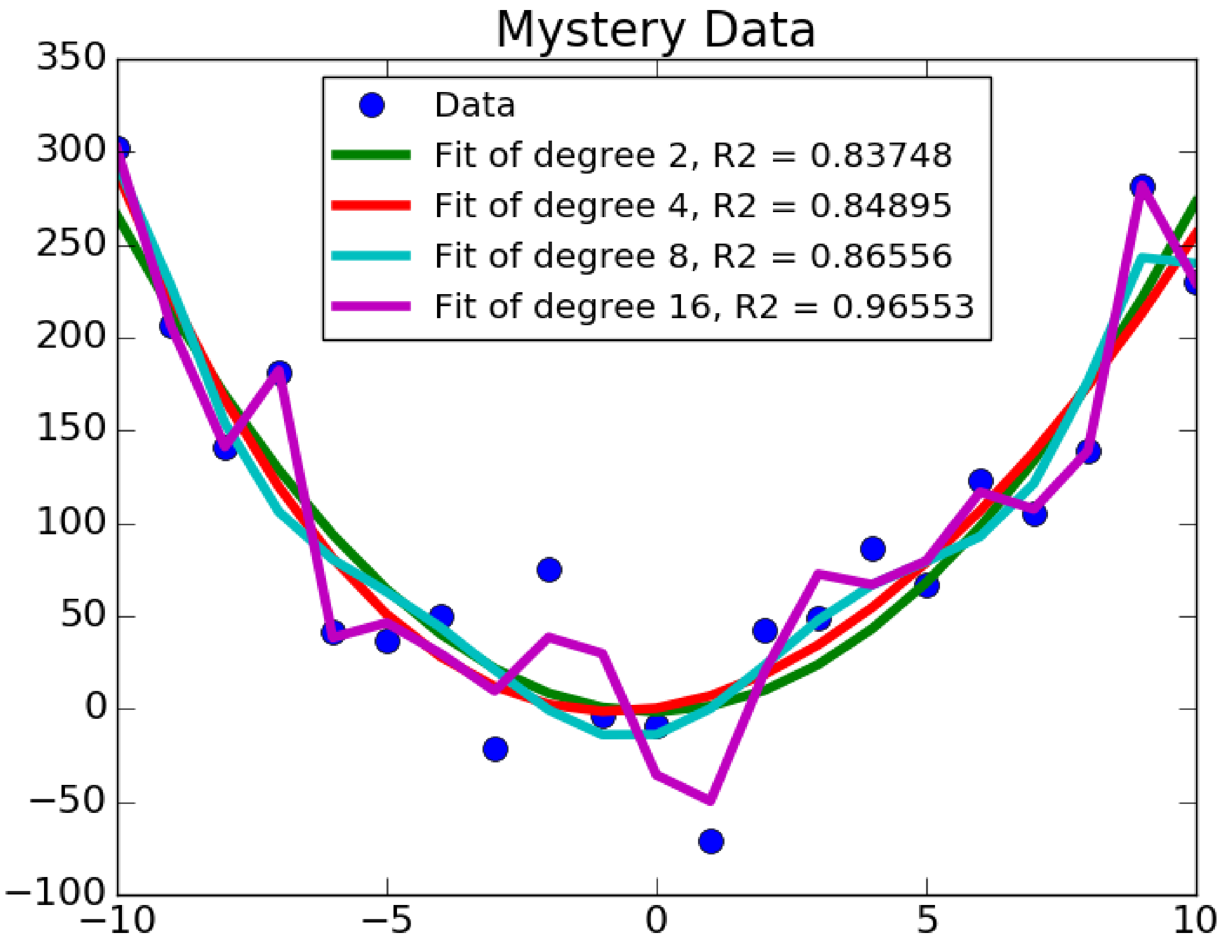
Understanding Experimental Data, cont.

Can We Get a Tighter Fit?



Why We Build Models

- Help us understand the process that generated the data
  - E.g., the properties of a particular linear spring
- Help us make predictions about out-of-sample data
  - E.g., predict the displacement of a spring when a force is applied to it
  - E.g., predict the effect of treatment on a patient
- A good model helps us do these things

How Mystery Data Was Generated

```
def genNoisyParabolicData(a, b, c, xVals, fName):
    yVals = []
    for x in xVals:
        theoreticalVal = a*x**2 + b*x + c
        yVals.append(theoreticalVal\
        + random.gauss(0, 35))
    f = open(fName,'w')
    f.write('x    y\n')
    for i in range(len(yVals)):
        f.write(str(yVals[i]) + ' ' + str(xVals[i]) + '\n')
    f.close()

#parameters for generating data
xVals = range(-10, 11, 1)
a, b, c = 3, 0, 0
genNoisyParabolicData(a, b, c, xVals,  ’Mystery Data.txt')
```

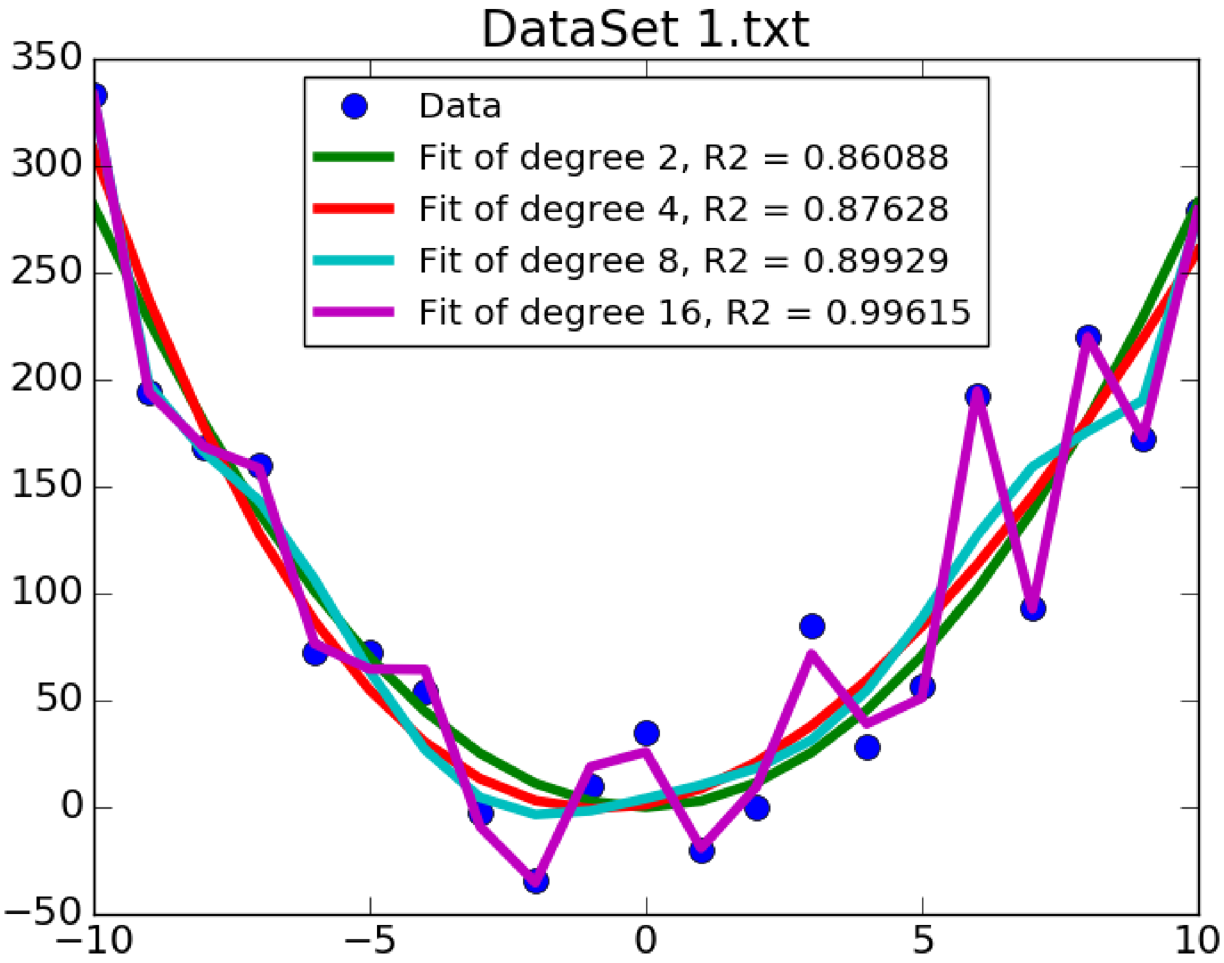
generate from parabolo

add some noise

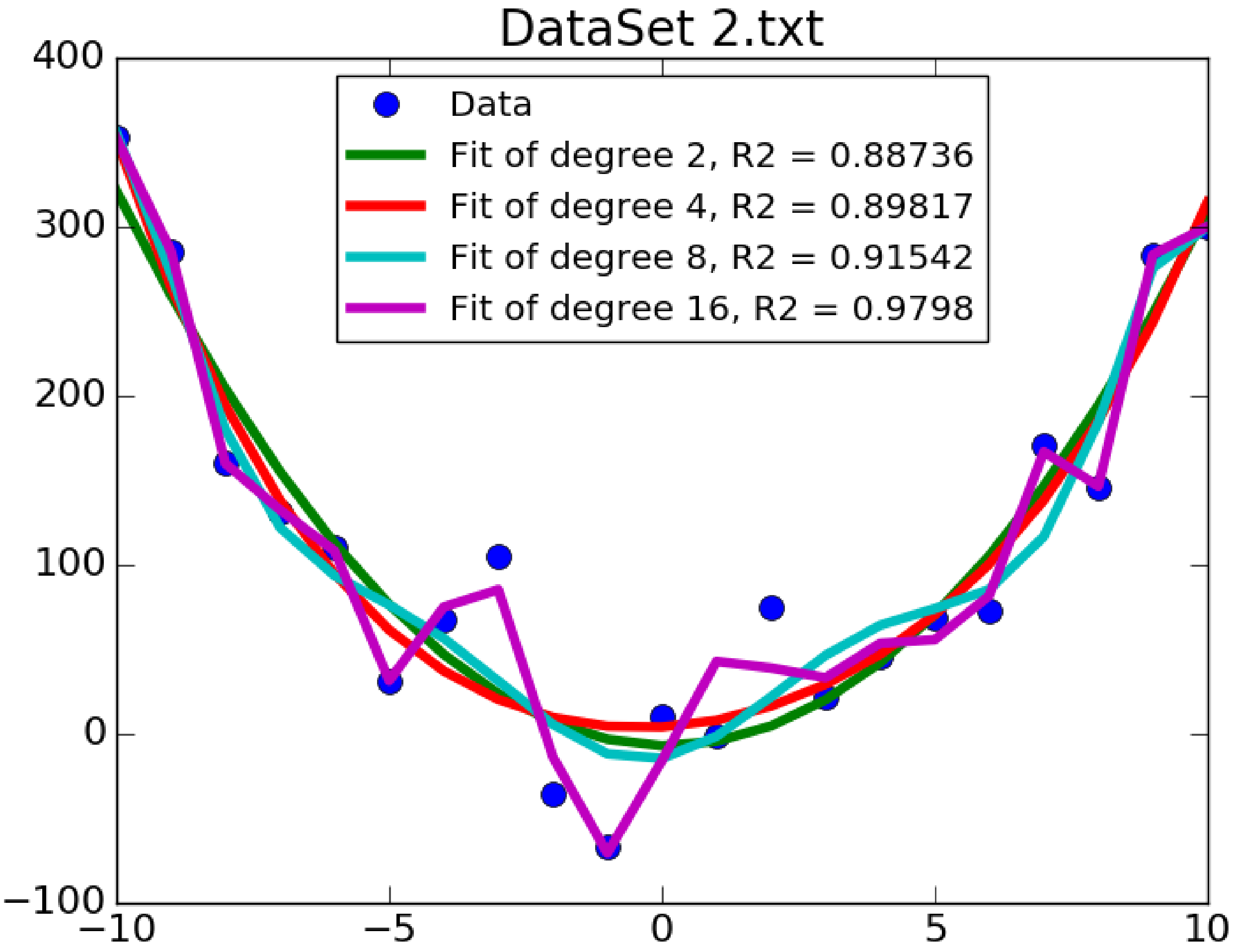
Let’s Look at Two Data Sets

```
degrees = (2, 4, 8, 16)

~~random.seed(0)~~
xVals1, yVals1 = getData('Dataset 1.txt')
models1 = genFits(xVals1, yVals1, degrees)
testFits(models1, degrees, xVals1, yVals1,
        'DataSet 1.txt')

pylab.figure()
xVals2, yVals2 = getData('Dataset 2.txt')
models2 = genFits(xVals2, yVals2, degrees)
testFits(models2, degrees, xVals2, yVals2,
         'DataSet 2.txt')
```

Fits for Dataset 1



Fit for Dataset 2



Hence Degree 16 Is Tightest Fit

- What we are looking at is training error
- How well the model performs on the data from which it was learned
- Small training error is a necessary condition for a great model, but not a sufficient one
- We want model to work well on other data generated by the same process
  - Measurements for other weights on the spring
  - Voters other than those surveyed
  - Etc.
- I.e., it needs to generalize

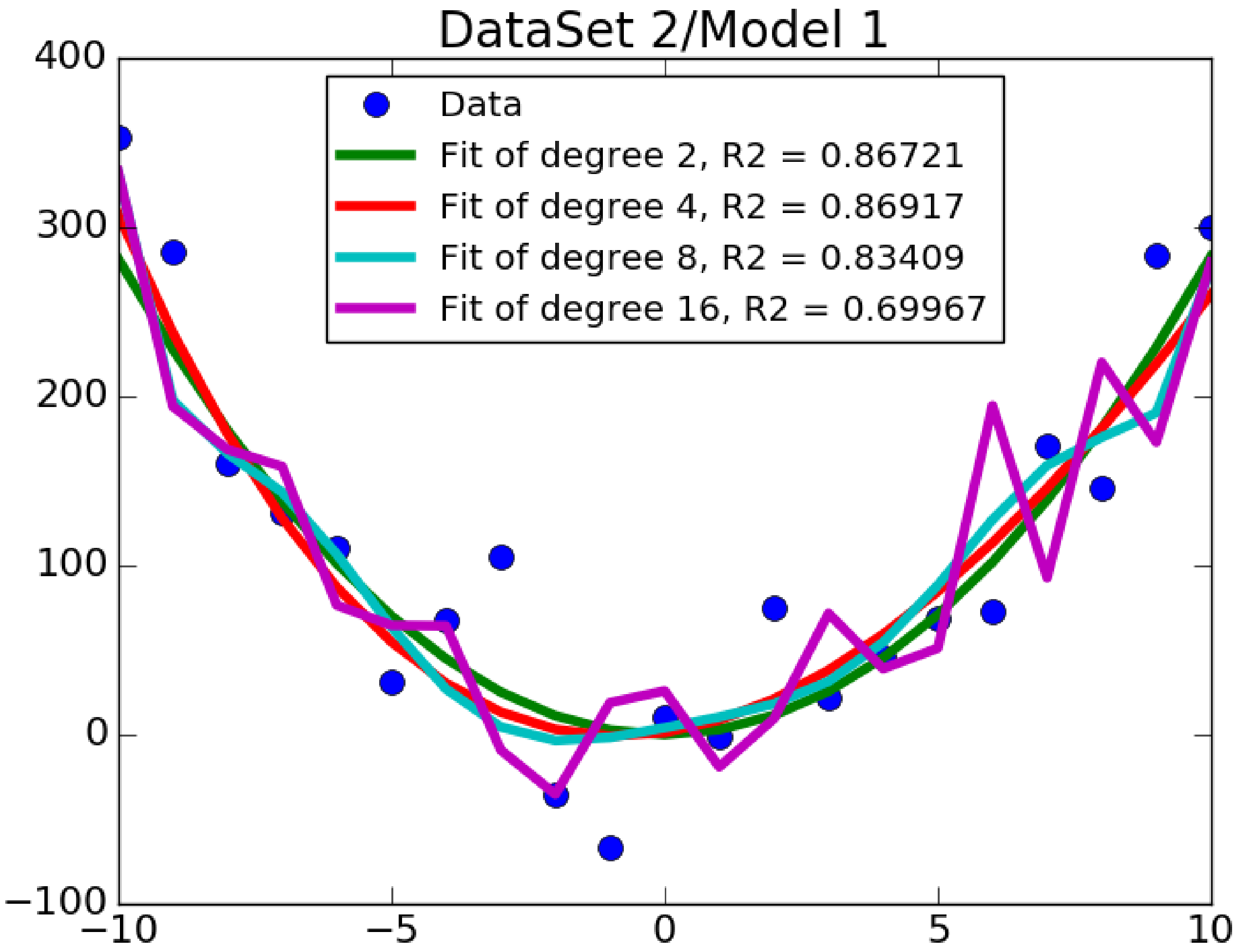
Cross Validate

- Generate models using one dataset, and then test it on the other
  - Use models for Dataset 1 to predict points for Dataset 2
  - Use models for Dataset 2 to predict points for Dataset 1
- Expect testing error to be larger than training error
- A better indication of generalizability than training error

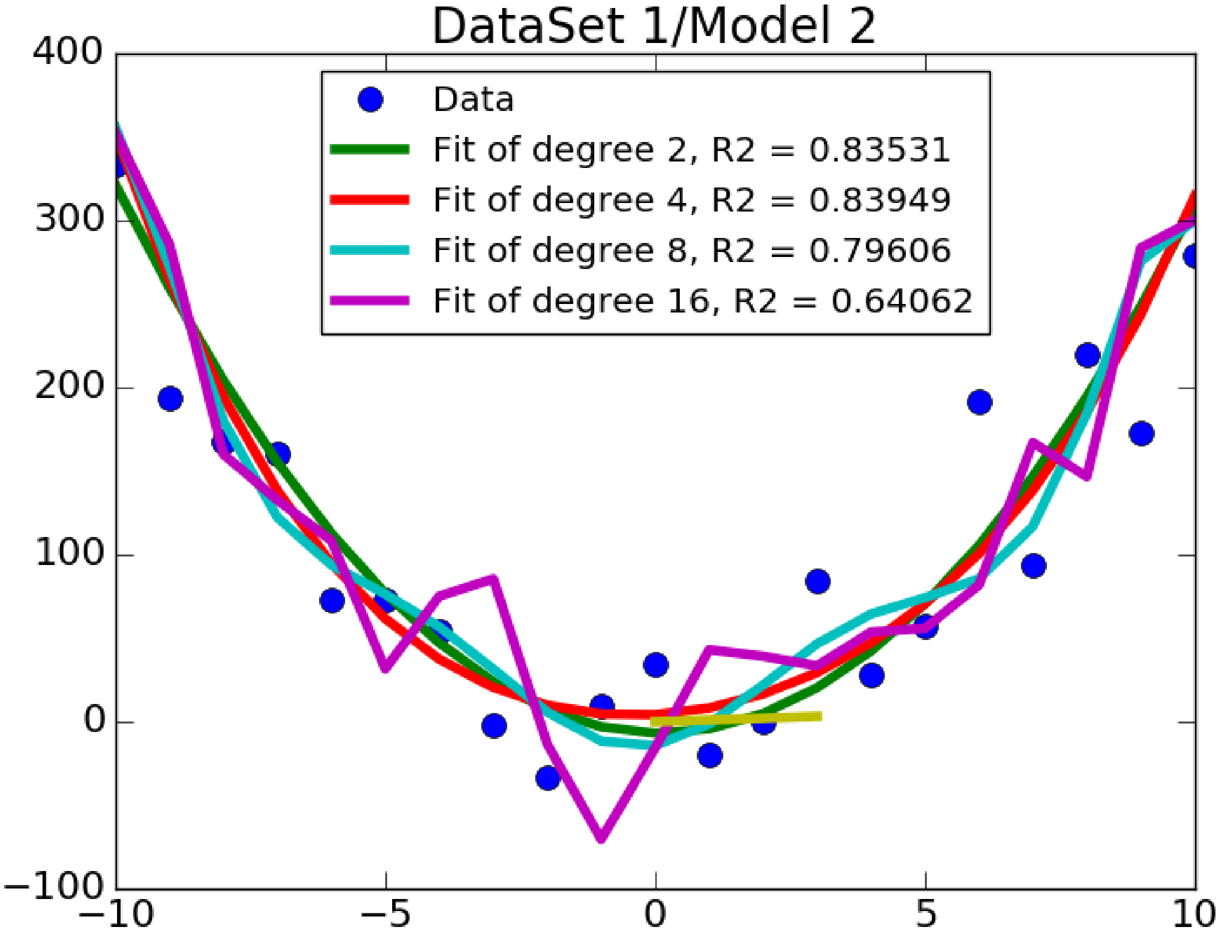
Test Code

```
pylab.figure()
testFits(models1, degrees, xVals2, yVals2,
         'DataSet 2/Model 1')
pylab.figure()
testFits(models2, degrees, xVals1, yVals1,
         'DataSet 1/Model 2')
```

Train on Dataset 1, Test on Dataset 2



Train on Dataset 2, Test on Dataset 1


DataSet 1/Model 2
● Data
Fit of degree 2, R2 = 0.83531
Fit of degree 4, R2 = 0.83949
Fit of degree 8, R2 = 0.79606
Fit of degree 16, R2 = 0.64062