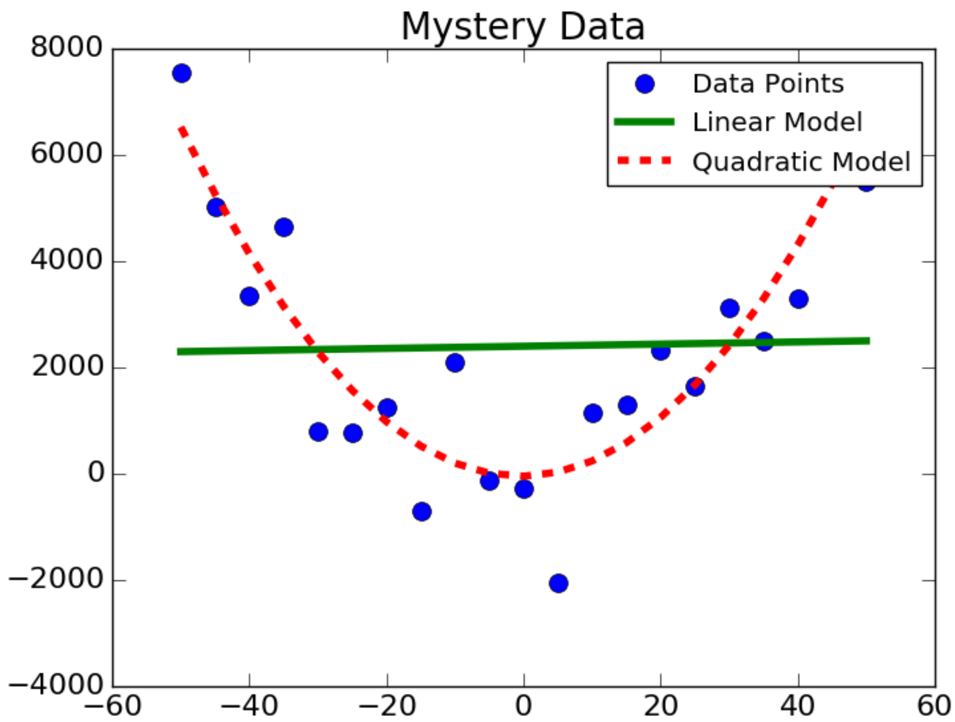
Understanding Experimental Data

How Good Are These Fits



- Relative to each other
- In an absolute sense

Relative to Each Other

- Fit is a function from the independent variable to the dependent variable
- Given an independent value, provides an estimate of the dependent value
- Which fit provides better estimates

Comparing Mean Squared Error

```
def aveMeanSquareError(data, predicted):
    error = 0.0
    for i in range(len(data)):
        error += (data[i] - predicted[i])**2
    return error/len(data)

estYVals = pylab.polyval(model1, xVals)
print('Ave. mean square error for linear model =',
      aveMeanSquareError(yVals, estYVals))
estYVals = pylab.polyval(model2, xVals)
print('Ave. mean square error for quadratic model =',
      aveMeanSquareError(yVals, estYVals))
```

Ave. mean square error for linear model = 9372.73078965

Ave. mean square error for quadratic model = 1524.02044718

In an Absolute Sense

- Mean square error useful for comparing two different models for the same data
- Useful for getting a sense of absolute goodness of fit?
  - Is 1524 good?
- Hard to know, since there is no upper bound and not scale independent
- Instead we use coefficient of determination, R^2,

$$R^2 = 1 - \frac{\sum_i (y_i - p_i)^2}{\sum_i (y_i - \mu)^2}$$

If You Prefer Code

$$R^2 = 1 - \frac{\sum_i (y_i - p_i)^2}{\sum_i (y_i - \mu)^2}$$

```
def rSquared(observed, predicted):
    error = ((predicted - observed)**2).sum()
    meanError = error/len(observed)
    return 1 - (meanError/numpy.var(observed))
```

R^2

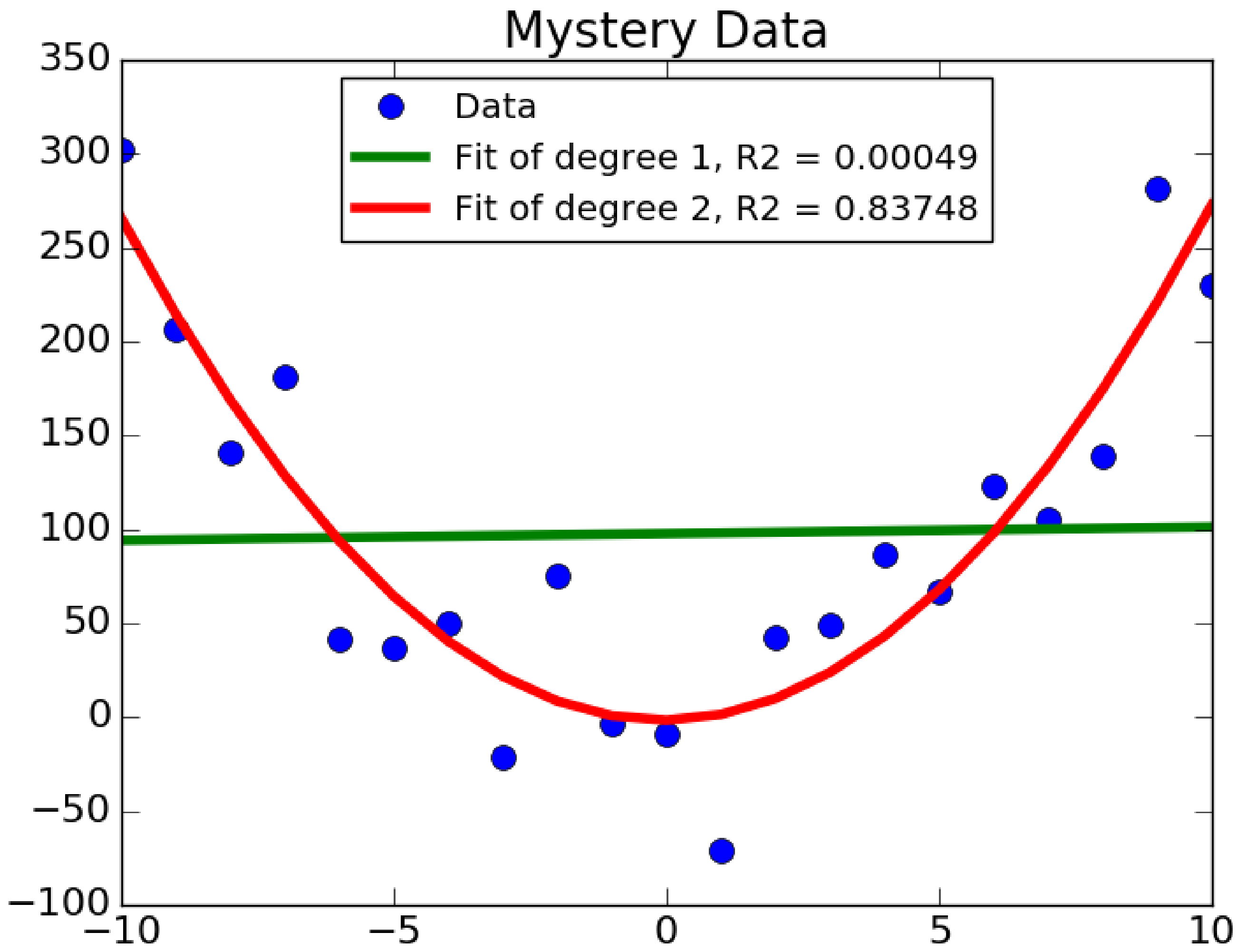
- By comparing the estimation errors (the numerator) with the variability of the original values (the denominator), R^2 is intended to capture the proportion of variability in a data set that is accounted for by the statistical model provided by the fit
- Always between 0 and 1 when fit generated by a linear regression and tested on training data
- If $R^2 = 1$, the model explains all of the variability in the data. If $R^2 = 0$, there is no relationship between the values predicted by the model and the actual data. If $R^2 = 0.5$, the model explains half the variability in the data.

Testing Goodness of Fits

```
def genFits(xVals, yVals, degrees):
    models = []
    for d in degrees:
        model = pylab.polyfit(xVals, yVals, d)
        models.append(model)
    return models

def testFits(models, degrees, xVals, yVals, title):
    pylab.plot(xVals, yVals, 'o', label = 'Data')
    for i in range(len(models)):
        estYVals = pylab.polyval(models[i], xVals)
        error = rSquared(yVals, estYVals)
        pylab.plot(xVals, estYVals,
                   label = 'Fit of degree '\
                   + str(degrees[i])\
                   + ', R2 = ' + str(round(error, 5)))
    pylab.legend(loc = 'best')
    pylab.title(title)
```

How Well Fits Explain Variance



Can We Get a Tighter Fit?