Multiple Linear Regression

(Chapters 12-13 in Montgomery, Runger)

12-1: Multiple Linear Regression Model

12-1.1 Introduction

- Many applications of regression analysis involve situations in which there are more than one regressor variable X_k used to predict Y.

- A regression model then is called a **multiple regression model**.

Multiple Linear Regression Model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots \beta_k x_k + \varepsilon$$

One can also use powers and products of other variables or even non-linear functions like $\exp(x_i)$ or $\log(x_i)$ instead of $x_3, \ldots x_k$.

Example: the general two-variable quadratic regression has 6 constants:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1)^2 + \beta_4 (x_2)^2 + \beta_5 (x_1 x_2) + \varepsilon$$

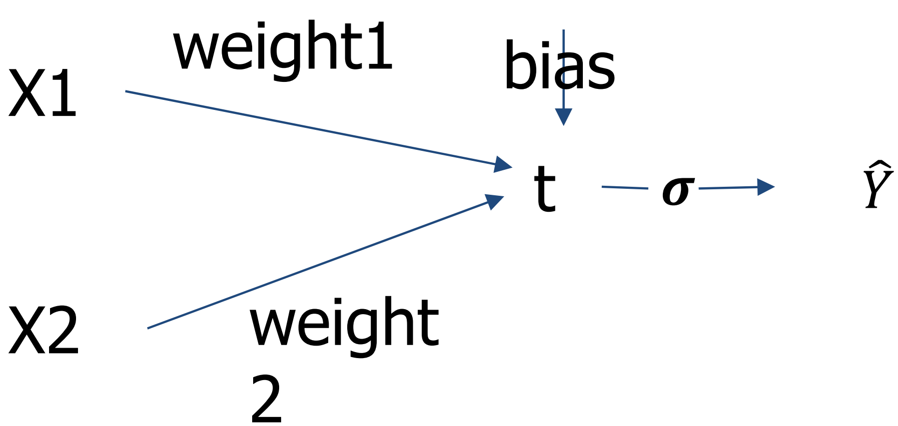
Nonlinear Regression

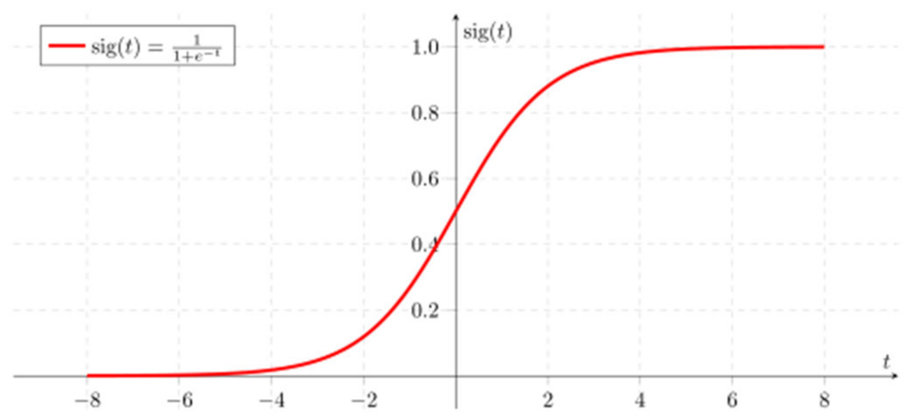
Example: Logistic Regression

$\hat{Y} = \boldsymbol{\sigma}(X1*weight1 + X2*weight2 + bias)$

Linear regression analog

$\hat{Y} = X1*beta1 + X2*beta2 + beta0$





12-1: Multiple Linear Regression Model

12-1.3 Matrix Approach to Multiple Linear Regression

Suppose the model relating the regressors to the response is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i \qquad i = 1, 2, \ldots, n$$

In matrix notation this model can be written as

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \qquad (12\text{-}6)$$

12-1: Multiple Linear Regression Model

12-1.3 Matrix Approach to Multiple Linear Regression

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} \text{ and } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon$$

12-1.3 Matrix Approach to Multiple Linear Regression

We wish to find the vector $\hat{\beta}$ that minimizes the sum of squares of error terms:

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \varepsilon'\varepsilon = (y - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$$

$$0 = \frac{\partial L}{2\partial \beta} = -\boldsymbol{X}'(\mathbf{y} - \mathbf{X}\beta) = -\boldsymbol{X}'\boldsymbol{y} + (\boldsymbol{X}'\boldsymbol{X})\boldsymbol{\beta}$$

The resulting least squares estimate is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \qquad (12\text{-}7)$$

Analog of $\frac{1}{Var(X)}$

Analog of $Cov(X,Y)$

Multiple Linear Regression Model

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{y}$$

H is an idempotent matrix

$$\hat{y} = X\hat{\beta} = X(X'X)^{-1}X'y,$$

$$\hat{y} = Hy, \quad \text{and} \quad e = (I - H)y.$$

$H = H'$; $H^2 = X(X'X)^{-1}X'X(X'X)^{-1}X' = X(X'X)^{-1}X' = H$

($(X'X)^{-1}X'X = I$)

Vectors $\hat{y}$ & e are orthogonal since

$\hat{y}' \cdot e = y'H(I-H)y = 0$ since

$$H(I - H) = H - H^2 = H - H = 0.$$

12-1: Multiple Linear Regression Models

12-1.4 Properties of the Least Squares Estimators

Unbiased estimators:

$$
\begin{aligned}
E(\hat{\boldsymbol{\beta}}) &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}] \\
&= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon})] \\
&= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\epsilon}] \\
&= \boldsymbol{\beta}
\end{aligned}
$$

Covariance Matrix of Estimators:

$$
\mathbf{C} = (\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix}
$$

12-1: Multiple Linear Regression Models

12-1.4 Properties of the Least Squares Estimators

Individual variances and covariances:

$$V(\hat{\beta}_j) = \sigma^2 C_{jj}, \qquad j = 0, 1, 2$$

$$\text{cov}(\hat{\beta}_i, \hat{\beta}_j) = \sigma^2 C_{ij}, \qquad i \neq j$$

In general,

$$\text{cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2 \mathbf{C}$$

R^2 and Adjusted R^2

The **coefficient of multiple determination R^2**

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T}$$

The **adjusted R^2** is

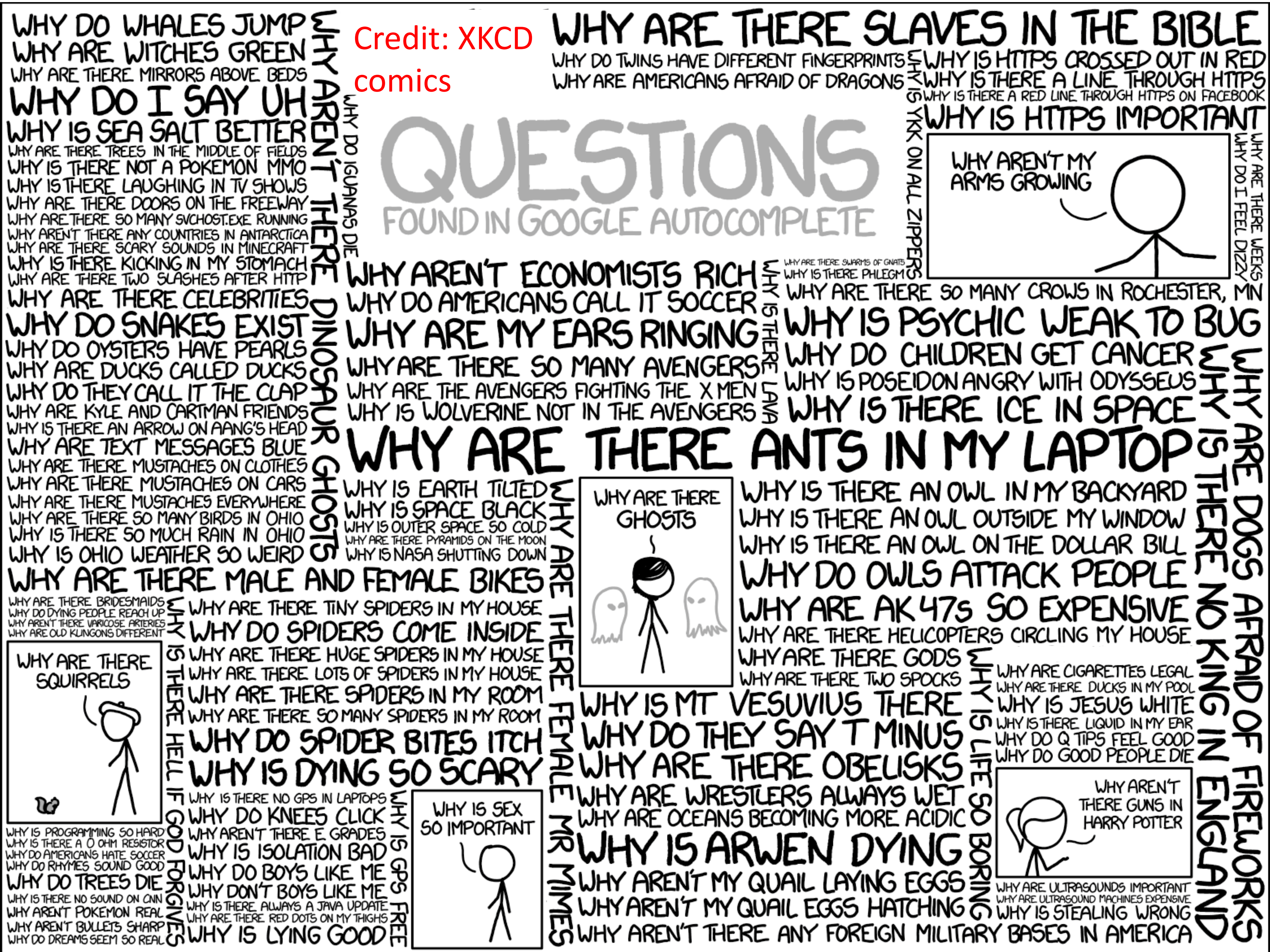
$$R^2_{adj} = 1 - \frac{SS_E/(n-p)}{SS_T/(n-1)} \approx 1 - \frac{\hat{\sigma}_E^2}{\hat{\sigma}_y^2} \qquad (12\text{-}23)$$

- The adjusted R^2 statistic penalizes adding terms to the MLR model.
- It can help guard against overfitting (including regressors that are not really useful)

How to know where to stop adding variables?

- Adding new variables x_i to MLR
  watch the adjusted R^2

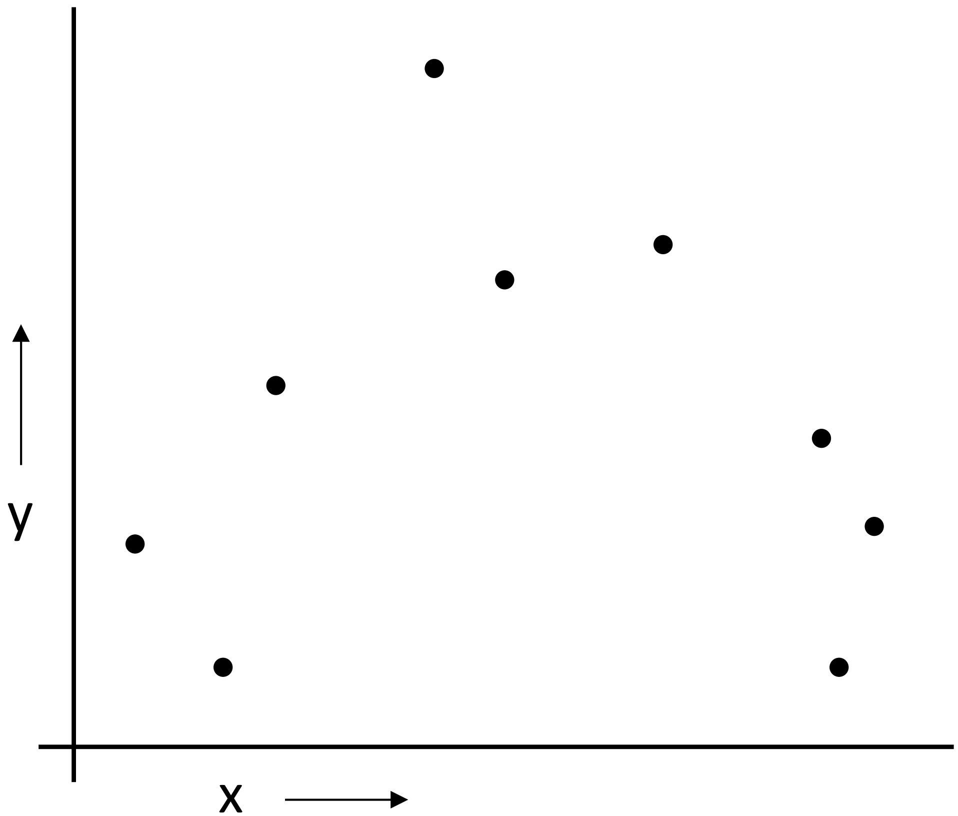
- Once the adjusted R^2
  no longer increases = stop.
  Now you did the best you can.



Credit: XKCD comics

How to know where to stop adding new variables or powers of old variables?

A Regression Problem

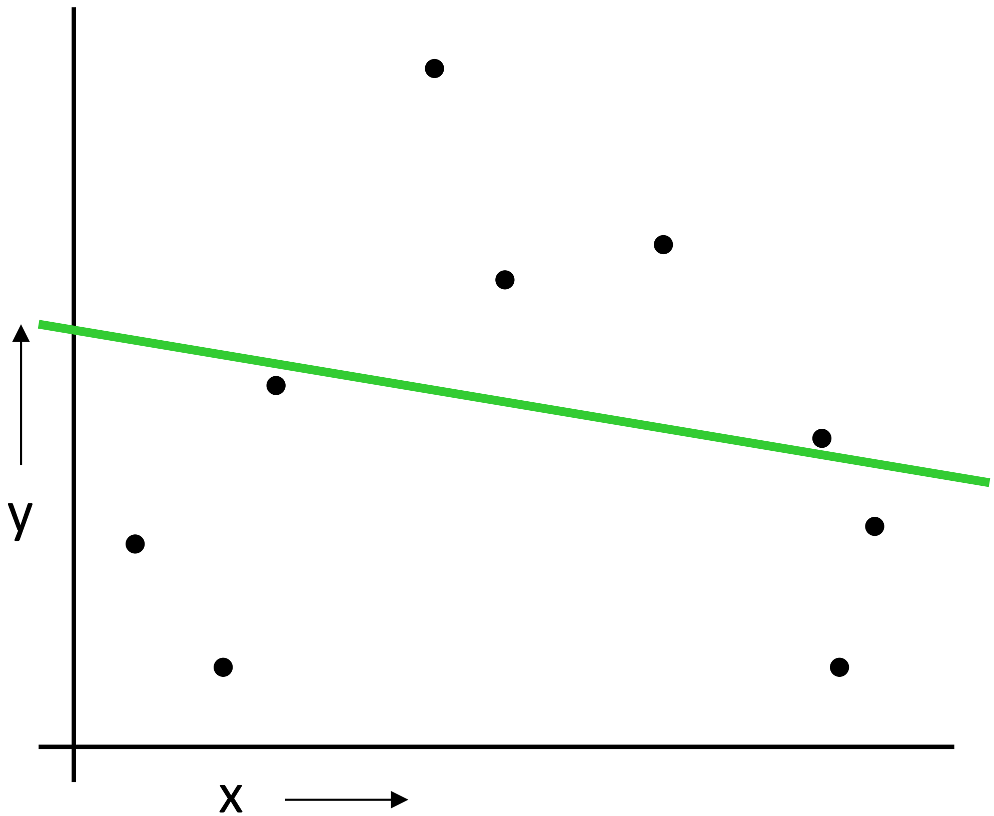


y = f(x) + noise

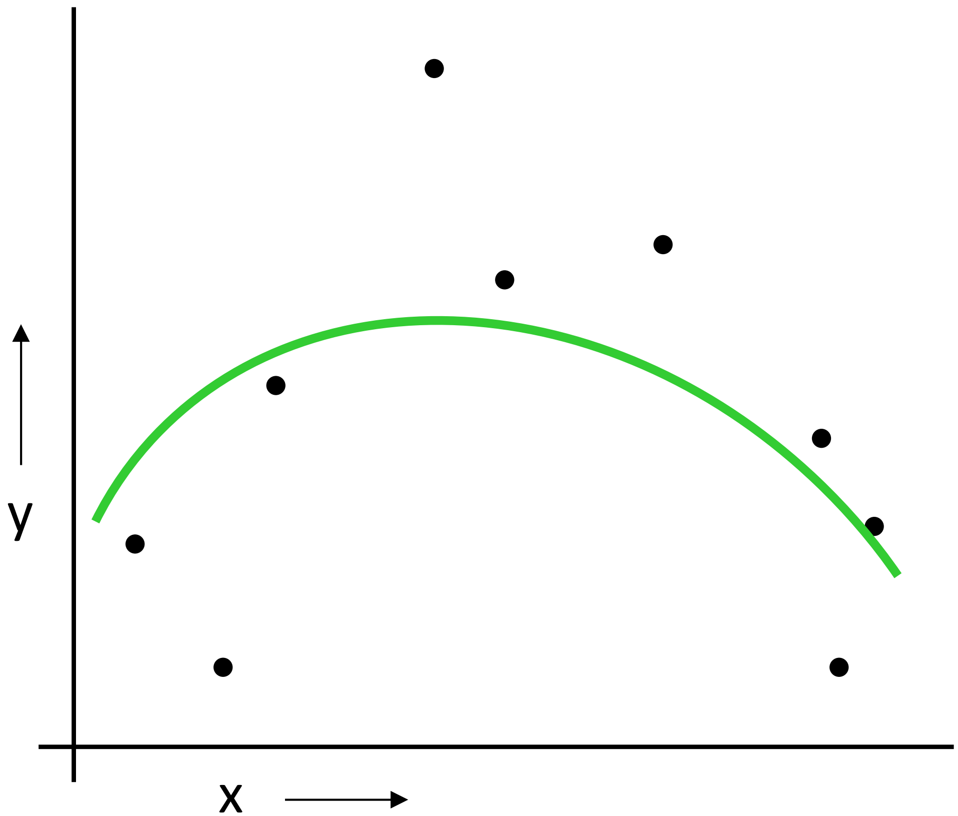
Can we learn f from this data?

Let's consider three methods…

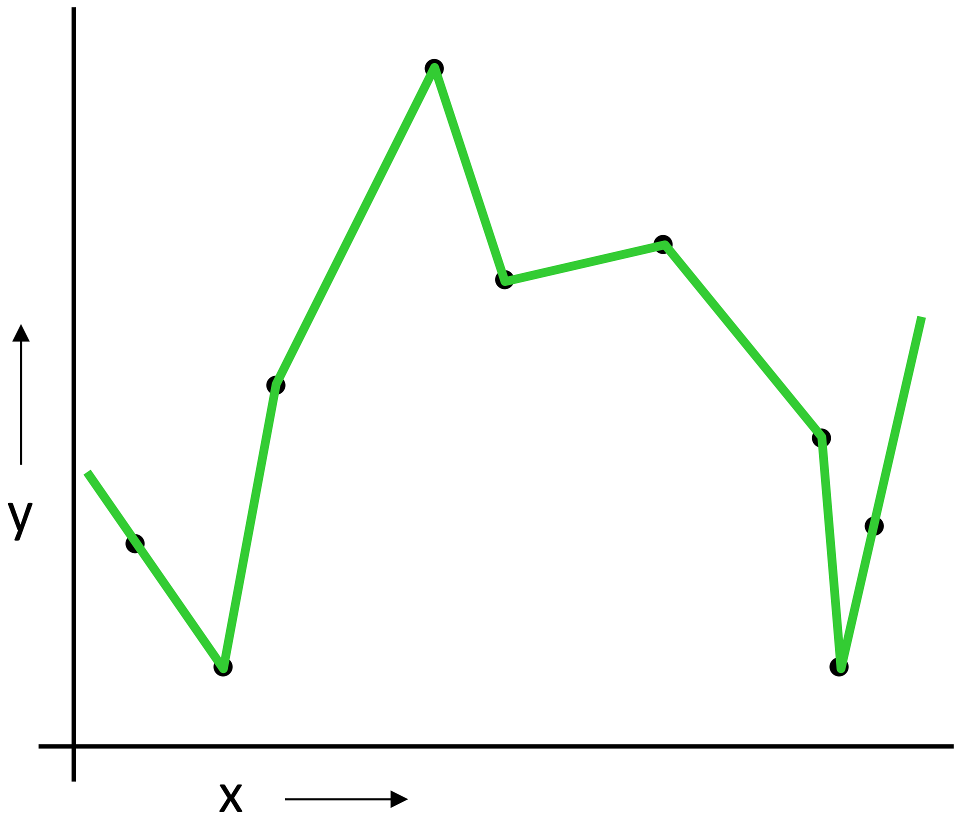
Linear Regression



Quadratic Regression

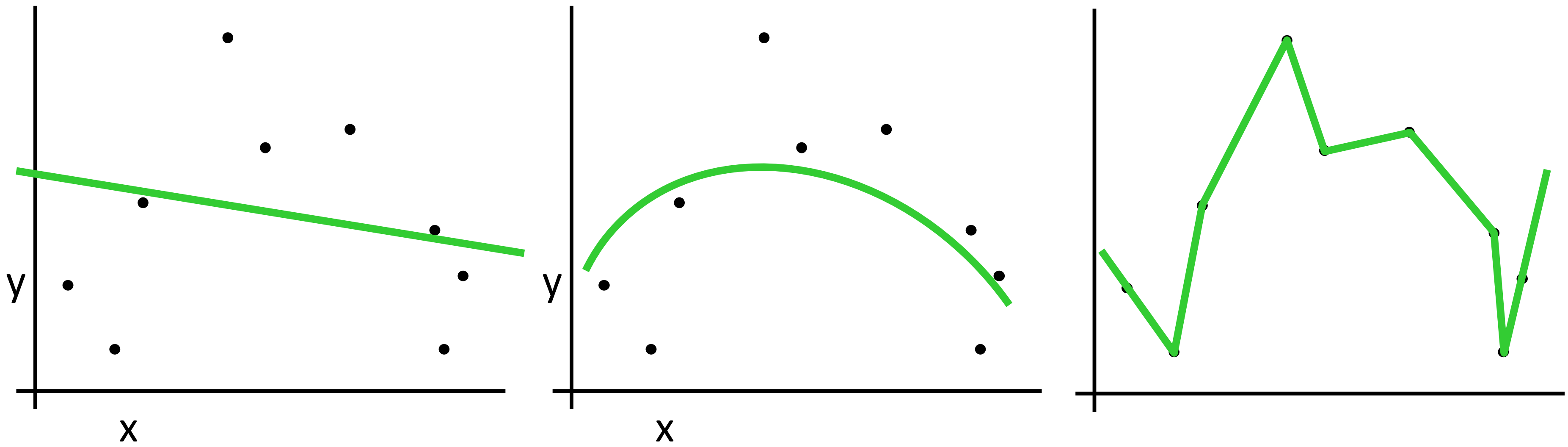


Join-the-dots



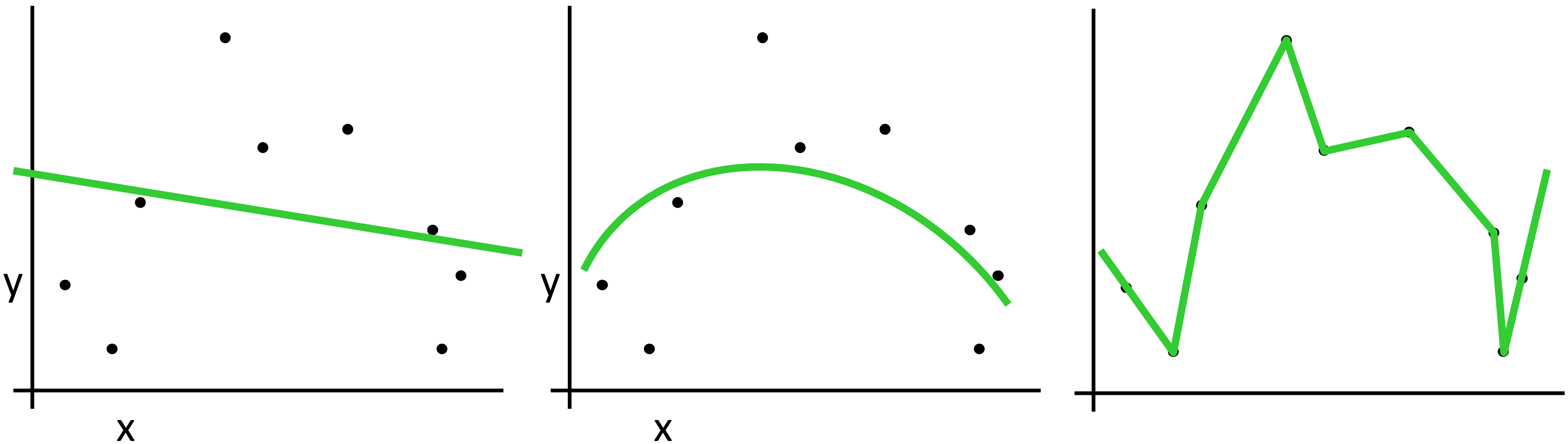
Also known as piecewise linear nonparametric regression if that makes you feel better

Which is best?



Why not choose the method with the best fit to the data?

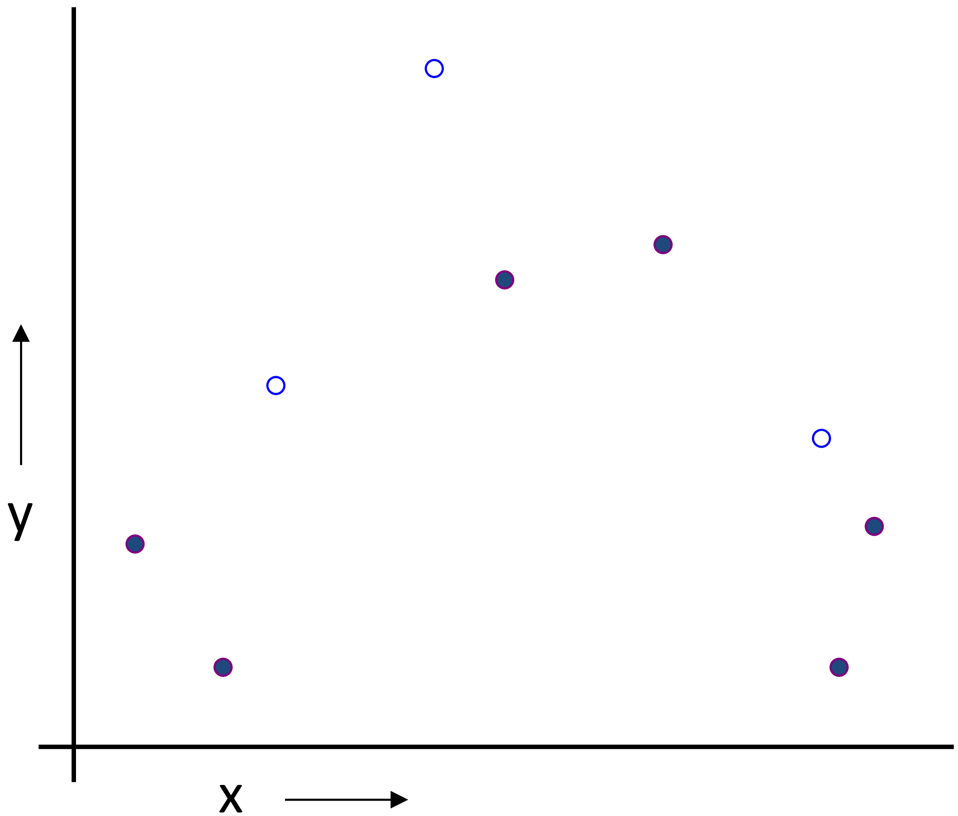
What do we really want?



Why not choose the method with the best fit to the data?

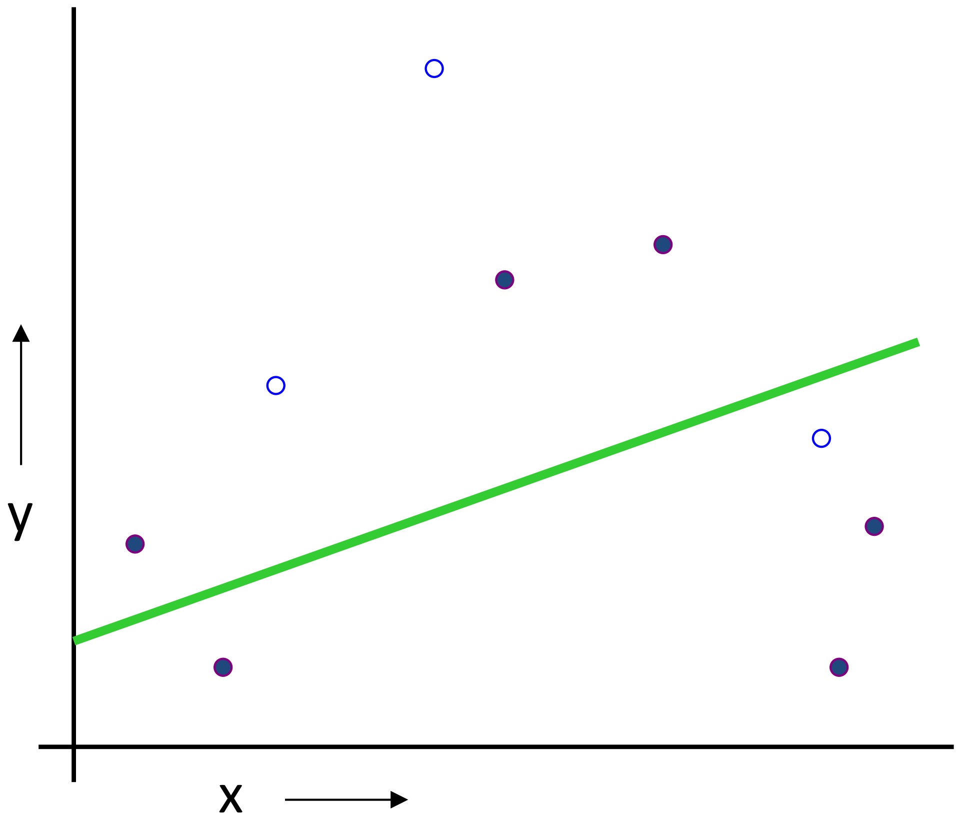
"How well are you going to predict future data drawn from the same distribution?"

The test set method



1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set

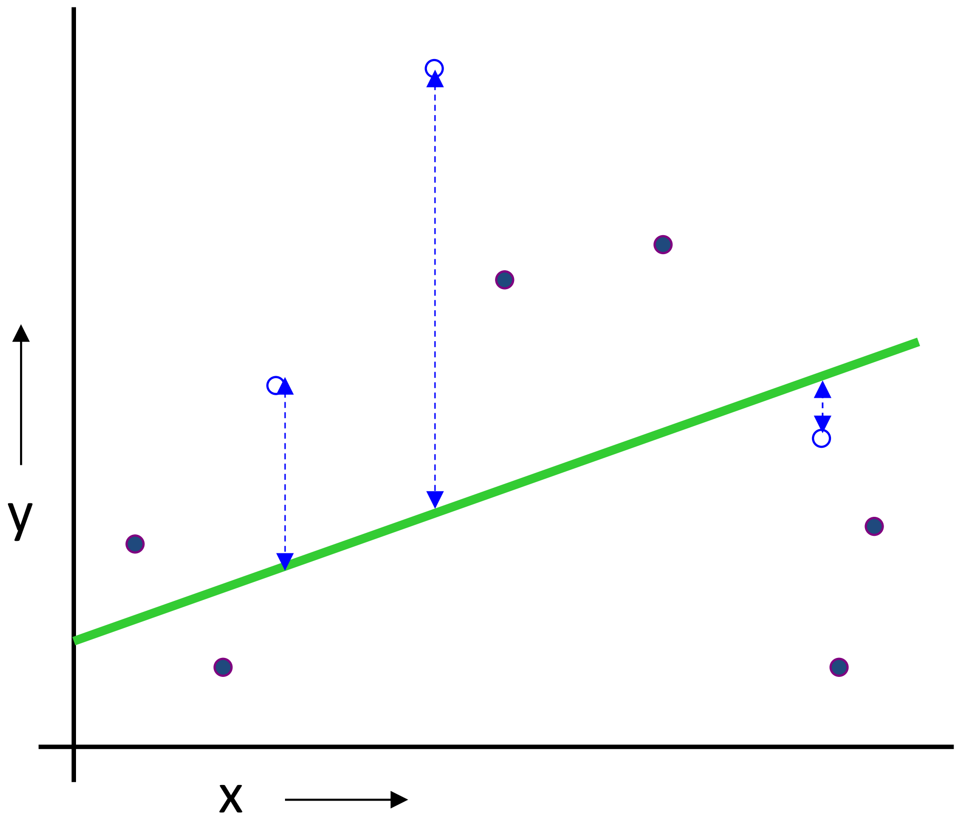
The test set method



1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set

(Linear regression example)

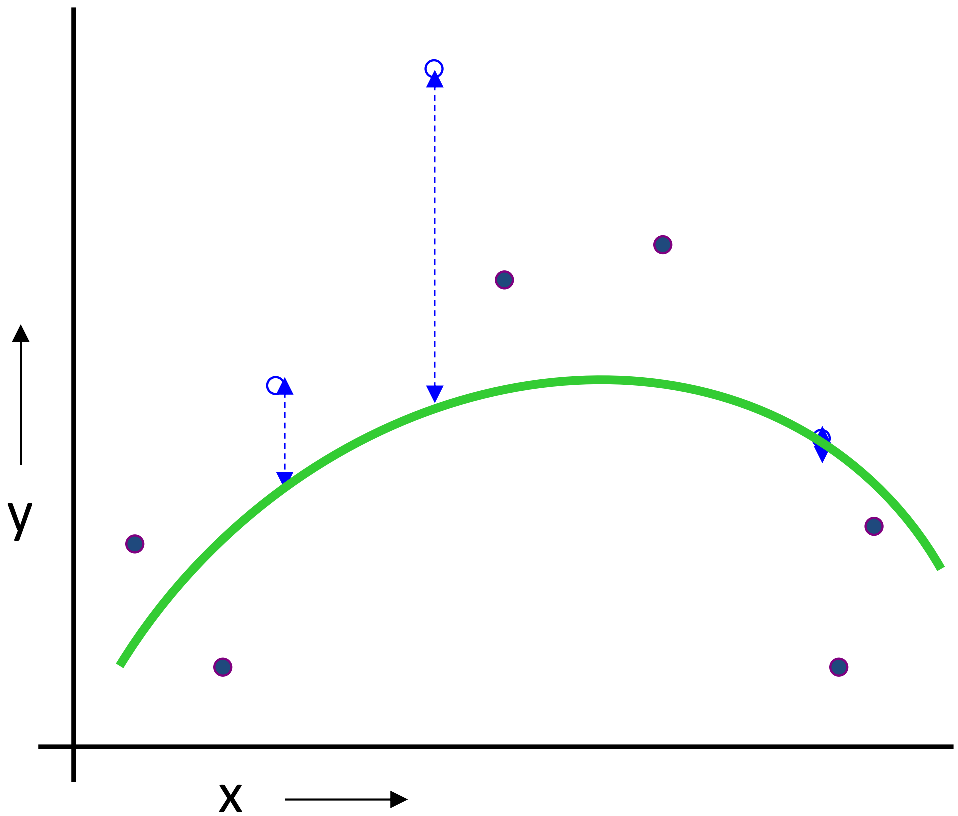
The test set method



(Linear regression example)
Mean Squared Error = 2.4

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

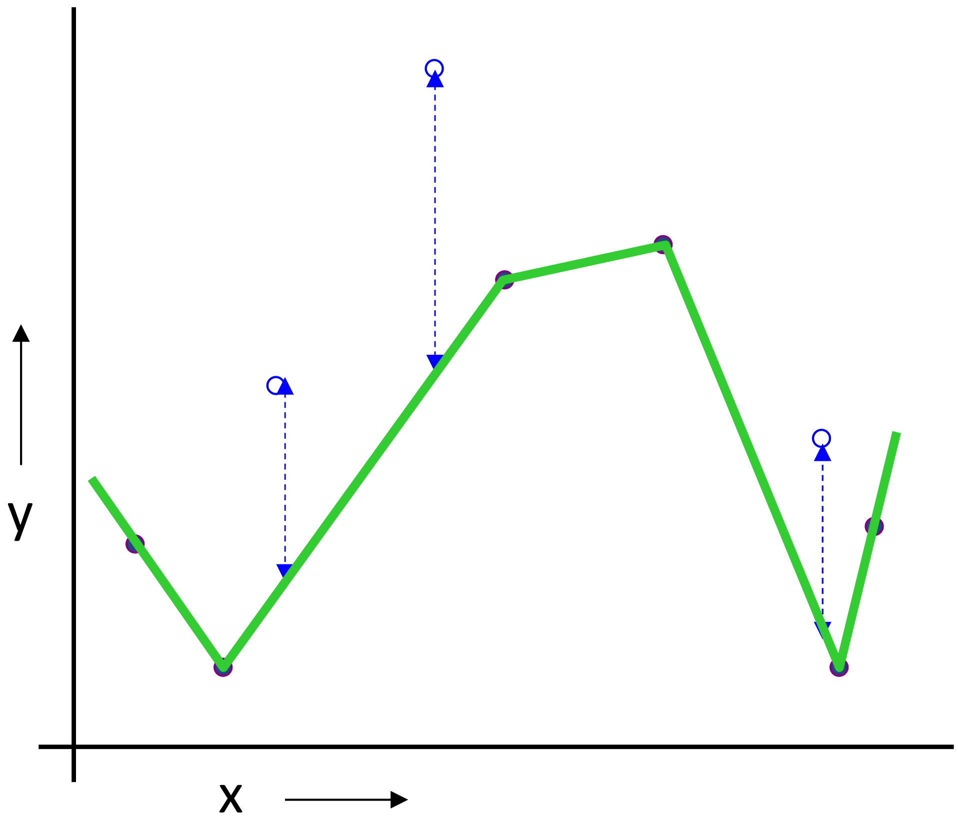
The test set method



(Quadratic regression example)
Mean Squared Error = 0.9

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

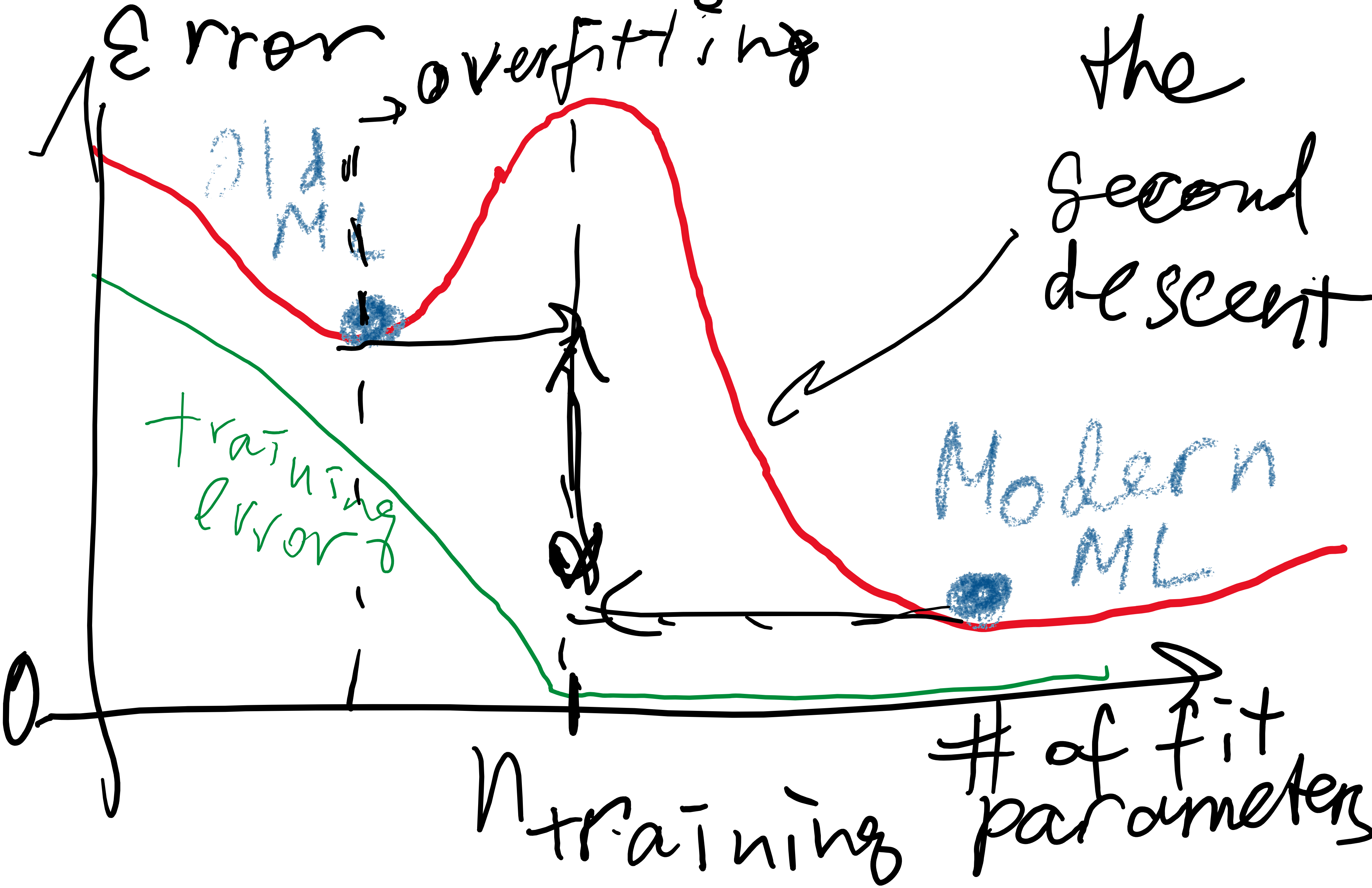
The test set method



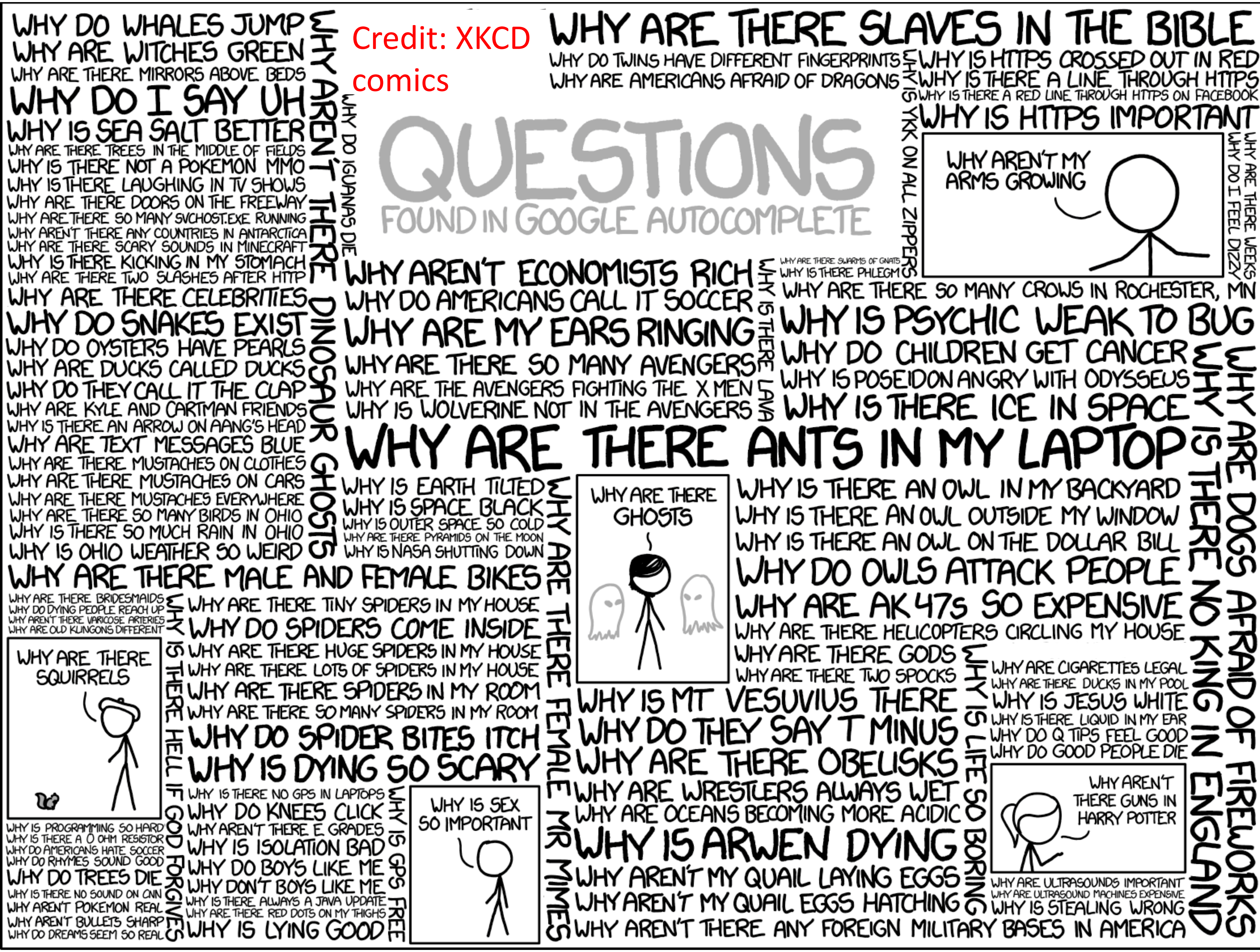
(Join the dots example)
Mean Squared Error = 2.2

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

Double descend- the main reason modern Machine Learning works so well



Credit: XKCD comics



WHEEL OF FORTUNE

Matlab exercise #1: “Wheel of Fortune”

- Each group gets a pair of genes that are known to be correlated.
- Each group also gets a random pair of genes selected by the “Wheel of Fortune”. They may or may not be correlated
- Download (log-transformed) expression_table.mat
- Run command fitlm(x,y) on assigned and random pairs
- Record β_0, β_1, R^2, P-value of the slope β_1 and write them on the blackboard
- Validate Matlab result for R^2 using your own calculations
- Look up gene names (see gene_description in your workspace) and write down a brief description of biological functions of genes. Does their correlation make biological sense?

Correlated pairs

plausible biological connection based on short description

1, 6 g1=1994; g2=188;

2, g1=2872; g2=1269;

3, g1=1321; g2=10;

4, g1= 886; g2=819;

5, g1=2138; g2=1364;

no obvious biological common function

```
g1=1+floor(rand.*3000); g2=1+floor(rand.*3000);
disp([g1, g2])
```