Welcome to Week #13!

Admin:

- No class/Office Hours next week (email for availability)
- EC options over break

Review: K-Nearest Neighbors

So far…

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n$$

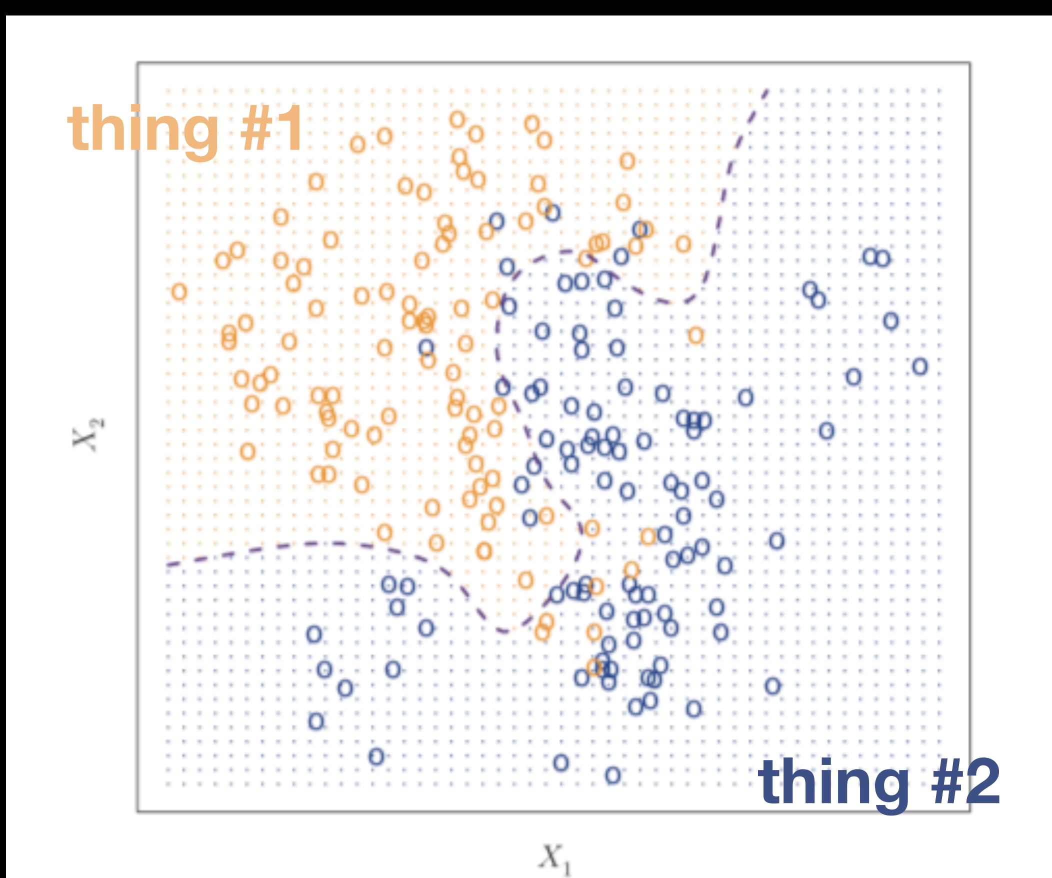
predicted y

intercept

So far we've been saying:
"I have a basic idea of what the functional form of y looks like (a line or a logit) - computer go find the parameters of that functional form.

This is nice because we have some hope of gaining intuition from our models.

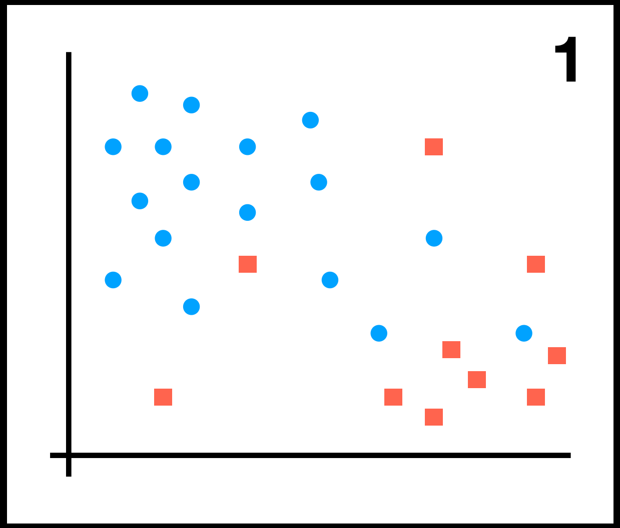
Now we classify…



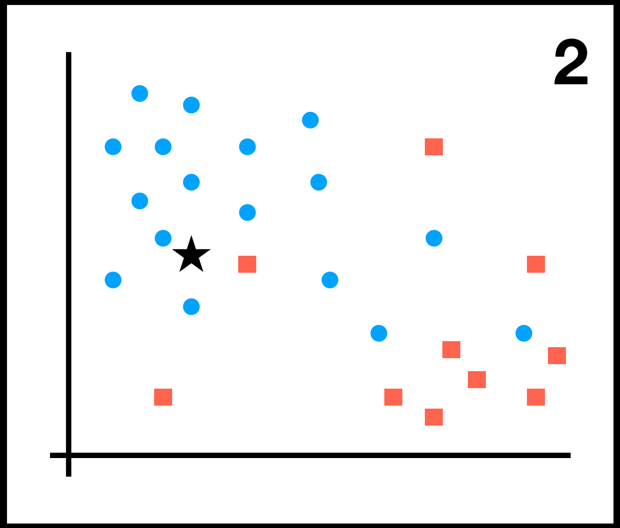
"I want to know where thing #1 and thing #2 live in some 2D space - computer, go figure out the boundary between these two things and let me know"

This is nice because we don't have to assume some model beforehand.

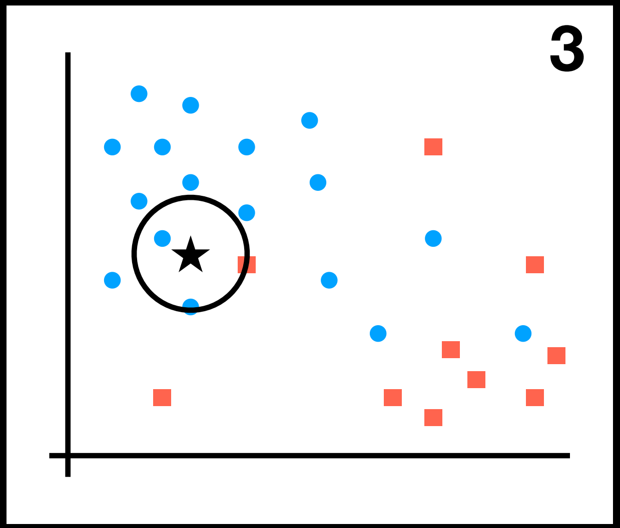
K Nearest Neighbors, in pictures



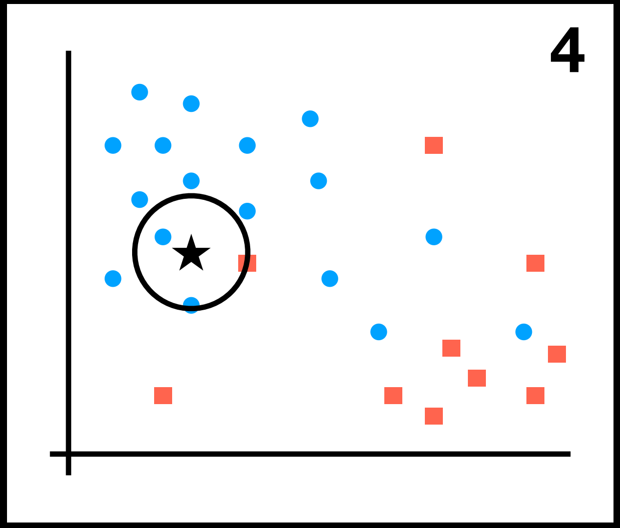
Sample (training) data representing underlying population



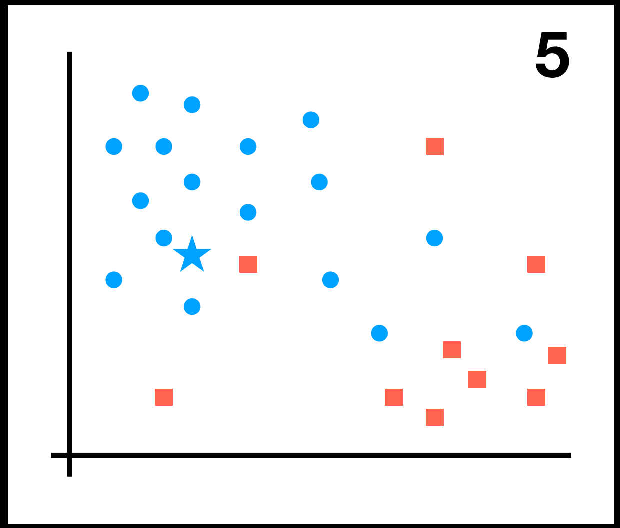
New point of interest



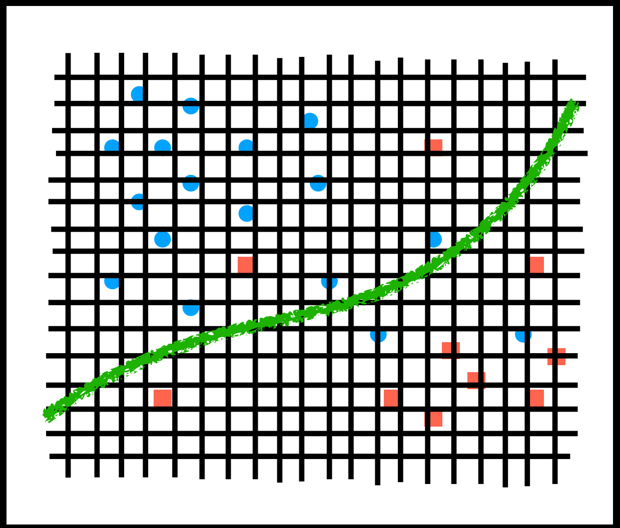
Find k nearest neighbors (here k = 3)



count “types” - here 2/3 points are blue
P(blue) = 2/3
P(red) = 1/3



if P > cut off say new point is in that group here: P(blue) > 0.5



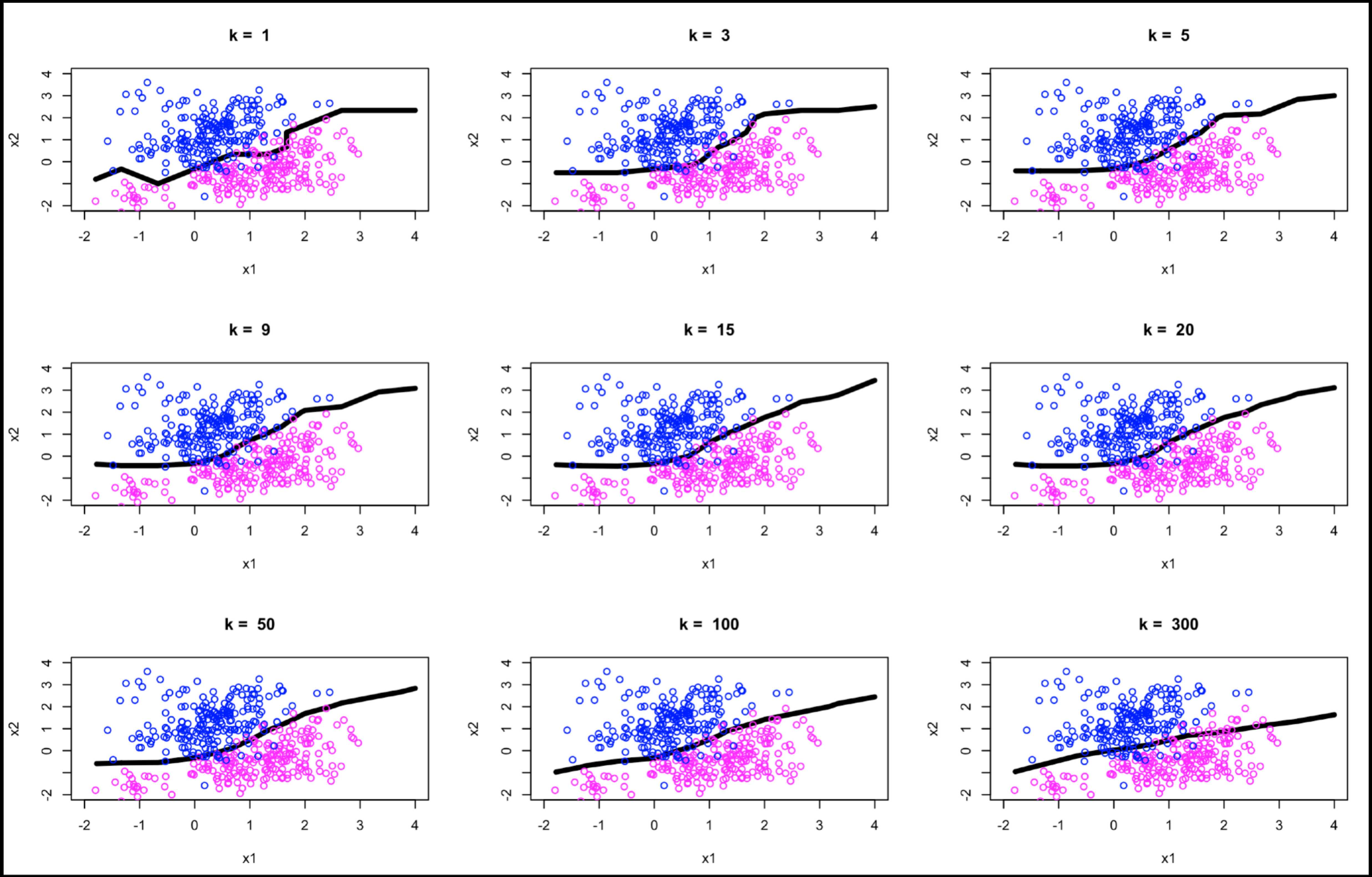
Repeat 2-3 on a grid & draw a separating line

K Nearest Neighbors (and MLR) → Cross-Validation & Bootstrap

p-values quantify the fit of individual parameters

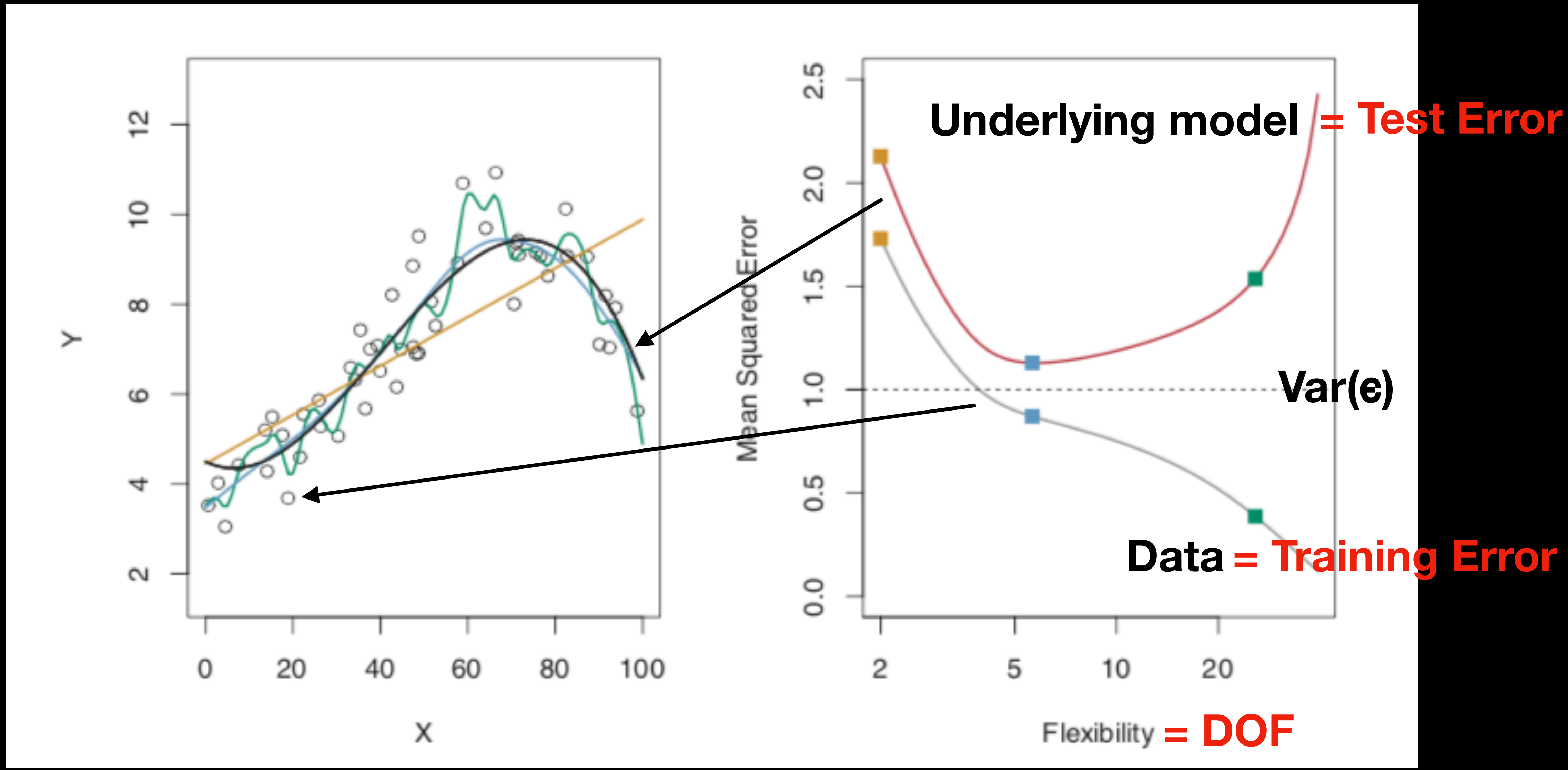
But first: some definitions!

quantify how good the *model* is



Bias-Variance Trade-Off (Second Glance)

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$



Actual underlying function - y

o Simulated data with added error (ϵ)

Linear fit

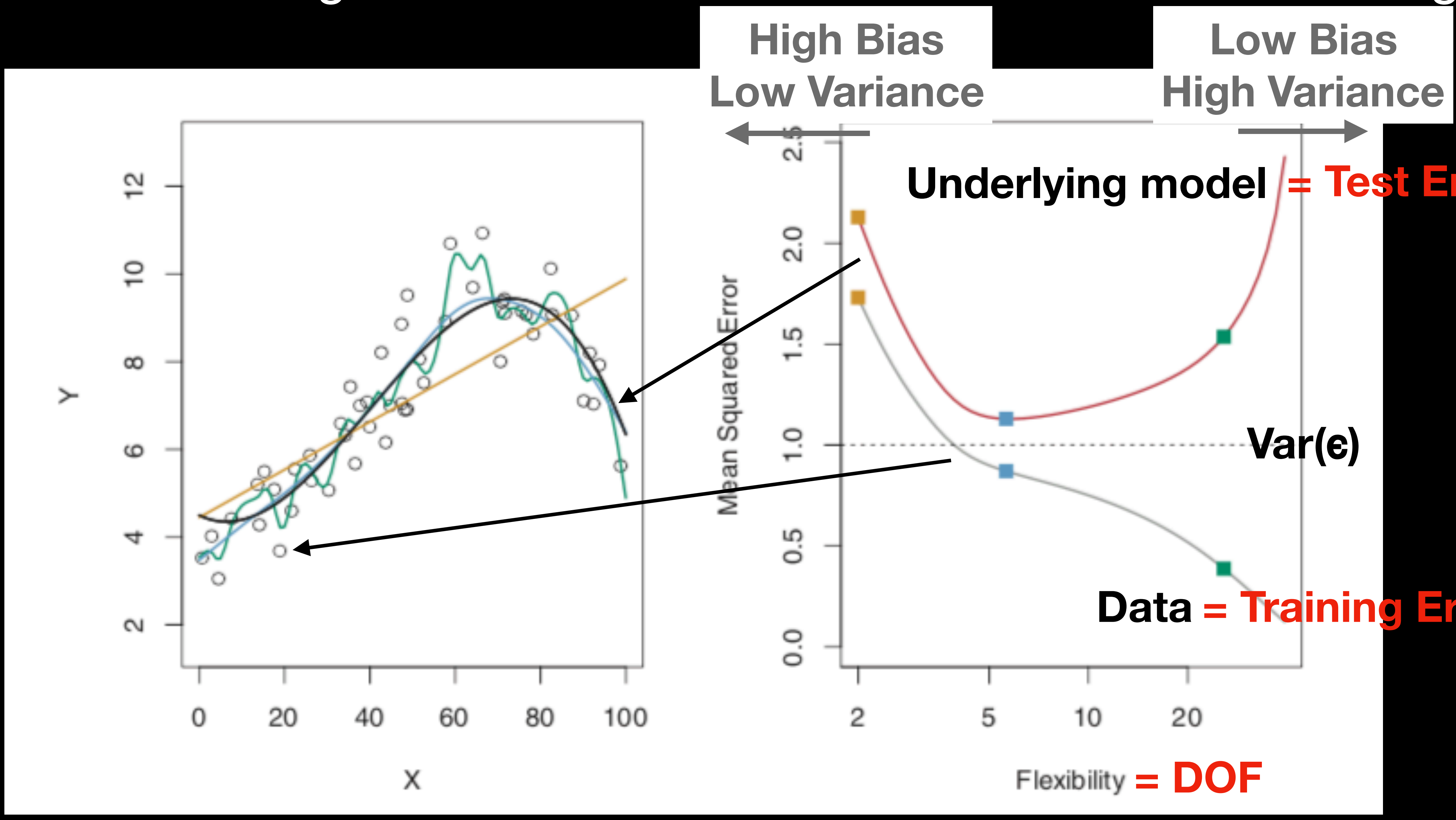
Low “flexibility” smooth spline

High “flexibility” smooth spline

Bias-Variance Trade-Off (Second Glance)

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

- The test error is the average error that results from using a statistical learning method to predict the response on a new observation, one that was *not* used in training the method. (Hard to calculate w/o an underlying model.)
- In contrast, the training error can be easily calculated by applying the statistical learning method to the observations used in its training.



Test & Training Error in KNN: With Math!

- The test error is the average error that results from using a statistical learning method to predict the response on a new observation, one that was *not* used in training the method. (Hard to calculate w/o an underlying model.)

$$\text{Ave}\left(I(y_0 \neq \hat{y}_0)\right)$$

new observation, requires we know what that would be from an underlying model (or more observations)

What our calculated fit/ model would predict

- In contrast, the training error can be easily calculated by applying the statistical learning method to the observations used in its training.

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

an observation in our current dataset

What our fit/model would predict

Here I = 1 if y_i != $\hat{y}_i$ and I = 0 if y_i = $\hat{y}_i$, so larger I means worse model

K Nearest Neighbors (and MLR) → Cross-Validation & Bootstrap

p-values quantify the fit of individual parameters

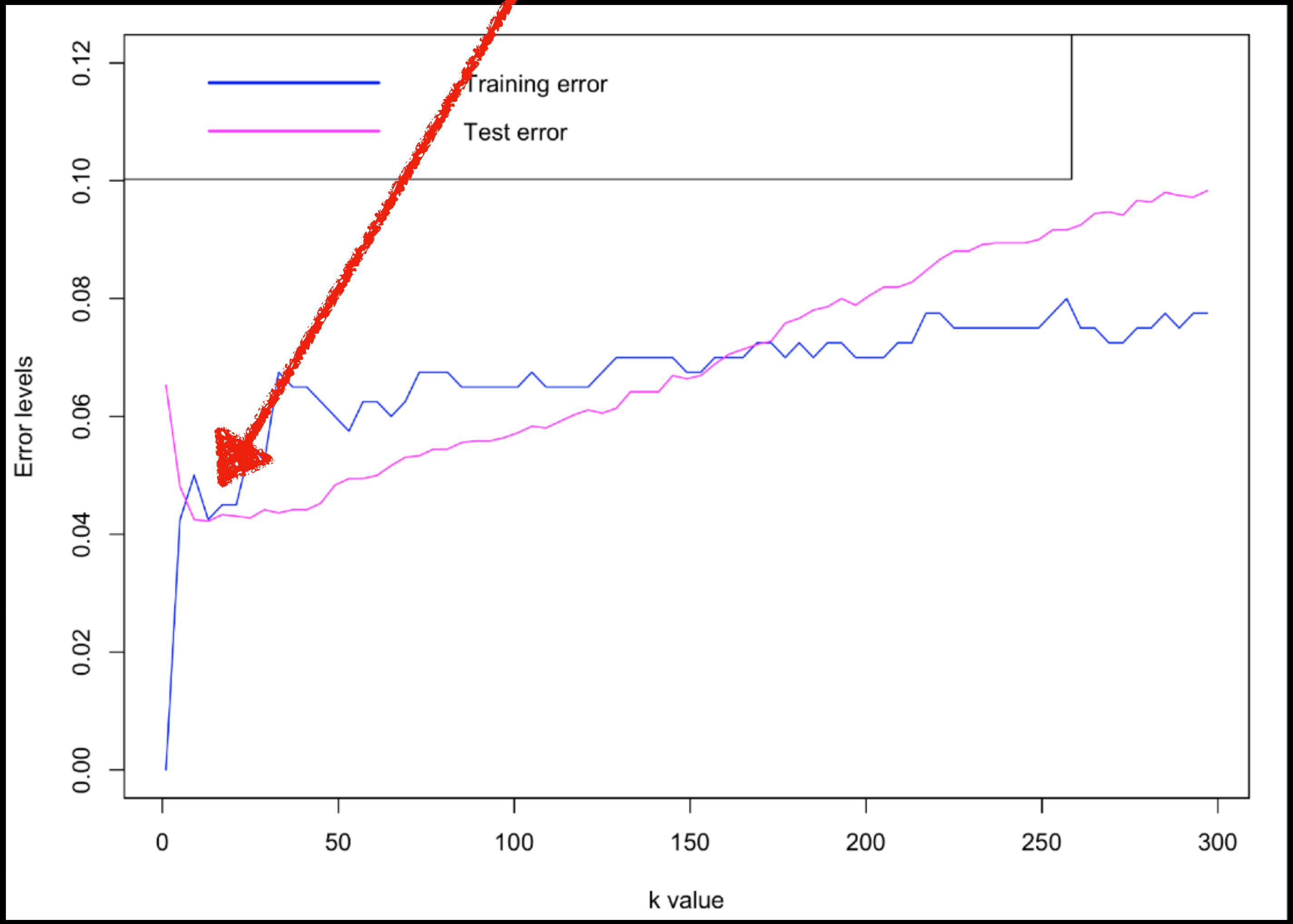
quantify how good the *model* is

But first: some definitions!

Using our KNN example in R with an underlying model!

Cross-Validation Methods

We can see that k~10 should minimize both types of errors



But we are only able to calculate the test error because we know the background distribution.

Cross-Validation (Ch. 5)

Regularization (Ch. 6)

Test fits with subsets of data

Use math to select parameters

Best model parameters

Cross-Validation Methods

CV: break sample into “test” and “training” datasets

Cross-Validation Methods

Fit model to data with a choice of parameters (e.g. degree of polynomial)

Cross-Validation Methods

Store each MSE_i for subset i

After all subsets are fit, we store AVERAGE(MSE_i) for this particular set of model parameters

Fit model to data with a choice of parameters (e.g. degree of polynomial)

Select a subset of the data

Calculate the mean square error (MSE) of "left out" data and model

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

Repeat for a bunch of subsets of data

Repeat for different model parameters

Output is: MSE(different values of model parameters) ~ TEST ERROR

Cross-Validation Methods

For classification problems

"tr" only: $I(y_i \neq \hat{y}_i)$

Fit model to data with a choice of parameters (e.g. degree of polynomial)

Select a subset of the data

Calculate the error of subset data and model

$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i$$

Repeat for a bunch of subsets of data

Repeat for different model parameters

Cross-Validation Methods

LOOCV
(Leave-One-Out Cross-Validation)

k-fold CV
(k-fold Cross-Validation)

fit on n-1 points, n-1 times

fit on k<n subsets, k times

In R!

shortcut MSE calculation (eq. 5.2 in ISL) for some fits, but otherwise can be computationally expensive

less computationally expensive than LOOCV

because subsets are similar - very similar output fits

less similar outputs

very easy to code

a little more complex to code, but not by much

Cross-Validation Methods: Some issues

- As we have seen, the validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.

- In the validation approach, only a subset of the observations — those that are included in the training set rather than in the validation set — are used to fit the model.

- This suggests that the validation set error may tend to overestimate the test error for the model fit on the entire data set.

Bootstrapping

- The bootstrap is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- For example, it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.
- The use of the term bootstrap derives from the phrase to pull oneself up by one's bootstraps, widely thought to be based on one of the eighteenth century "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:

The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.

- It is not the same as the term "bootstrap" used in computer science meaning to "boot" a computer from a set of core instructions, though the derivation is similar.

Boo

- Th ble and powerful statistical tool that can be us certainty associated with a given estimator or st od.

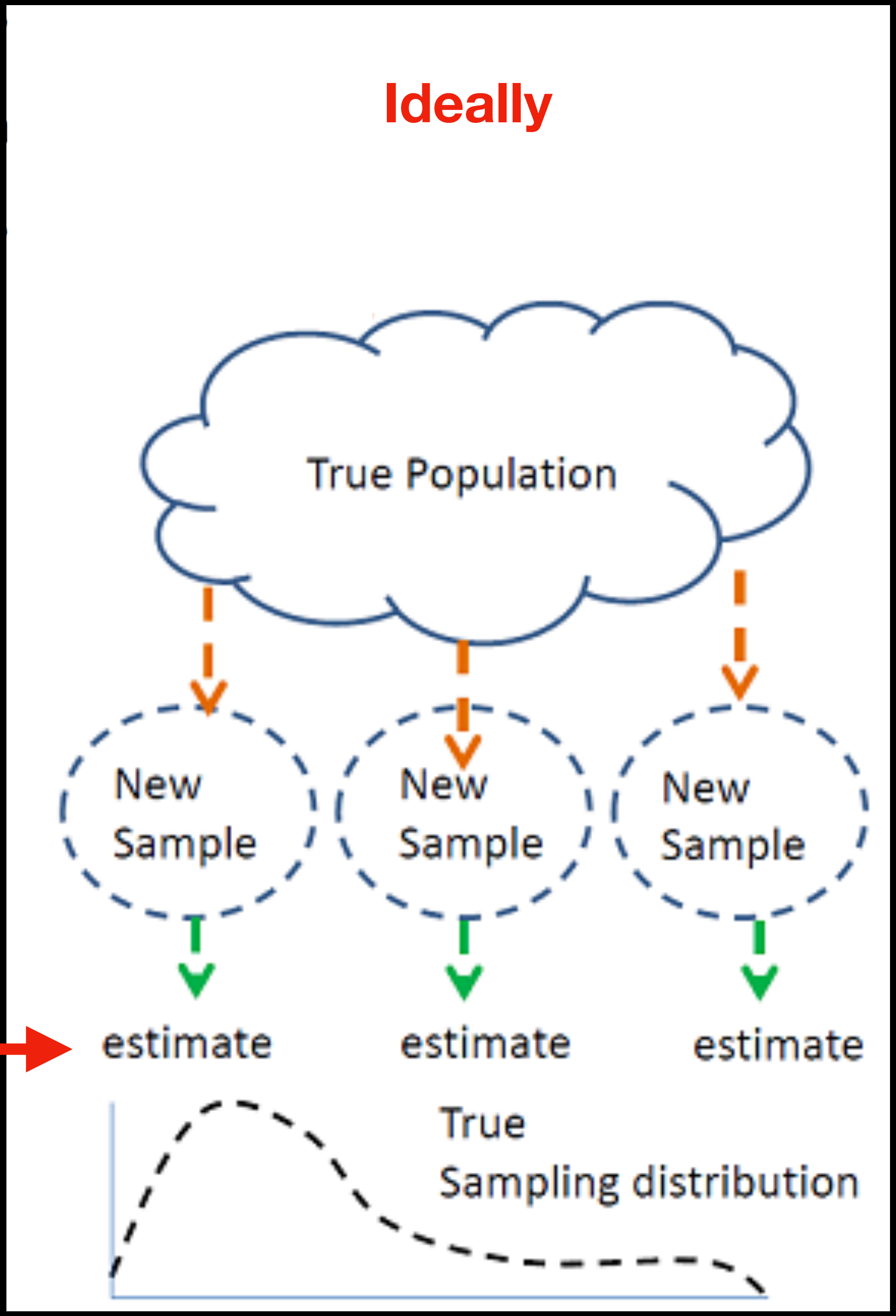
- Fo vide an estimate of the standard error of a co nce interval for that coefficient.

- Th otstrap derives from the phrase to pull or tstraps, widely thought to be based on one of y “The Surprising Adventures of Baron Munchausen” by Rudolph Erich Raspe:

The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thought to pick himself up by his own bootstraps.

- It is not the same as the term “bootstrap” used in computer science meaning to “boot” a computer from a set of core instructions, though the derivation is similar.

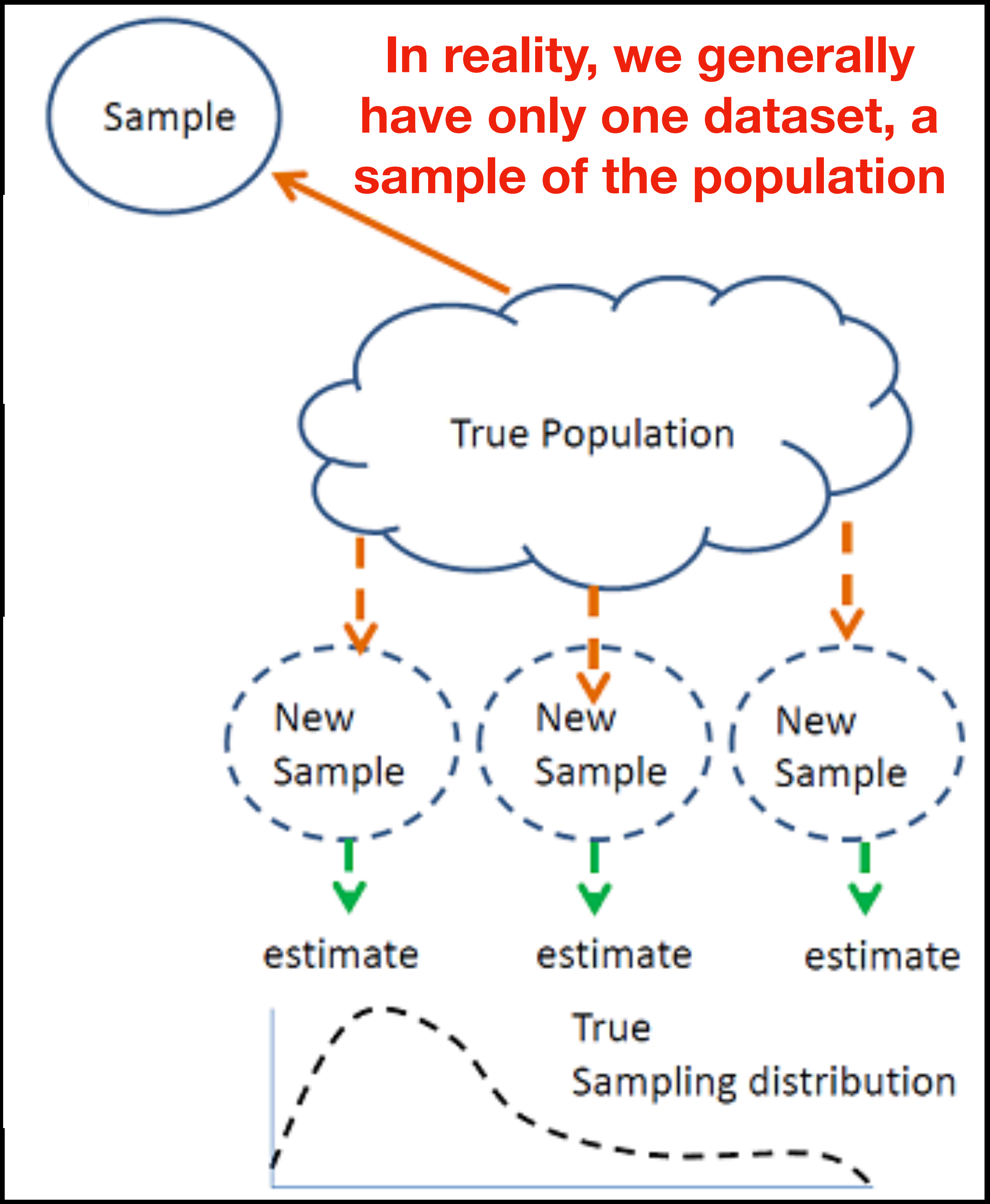
Bootstrapping



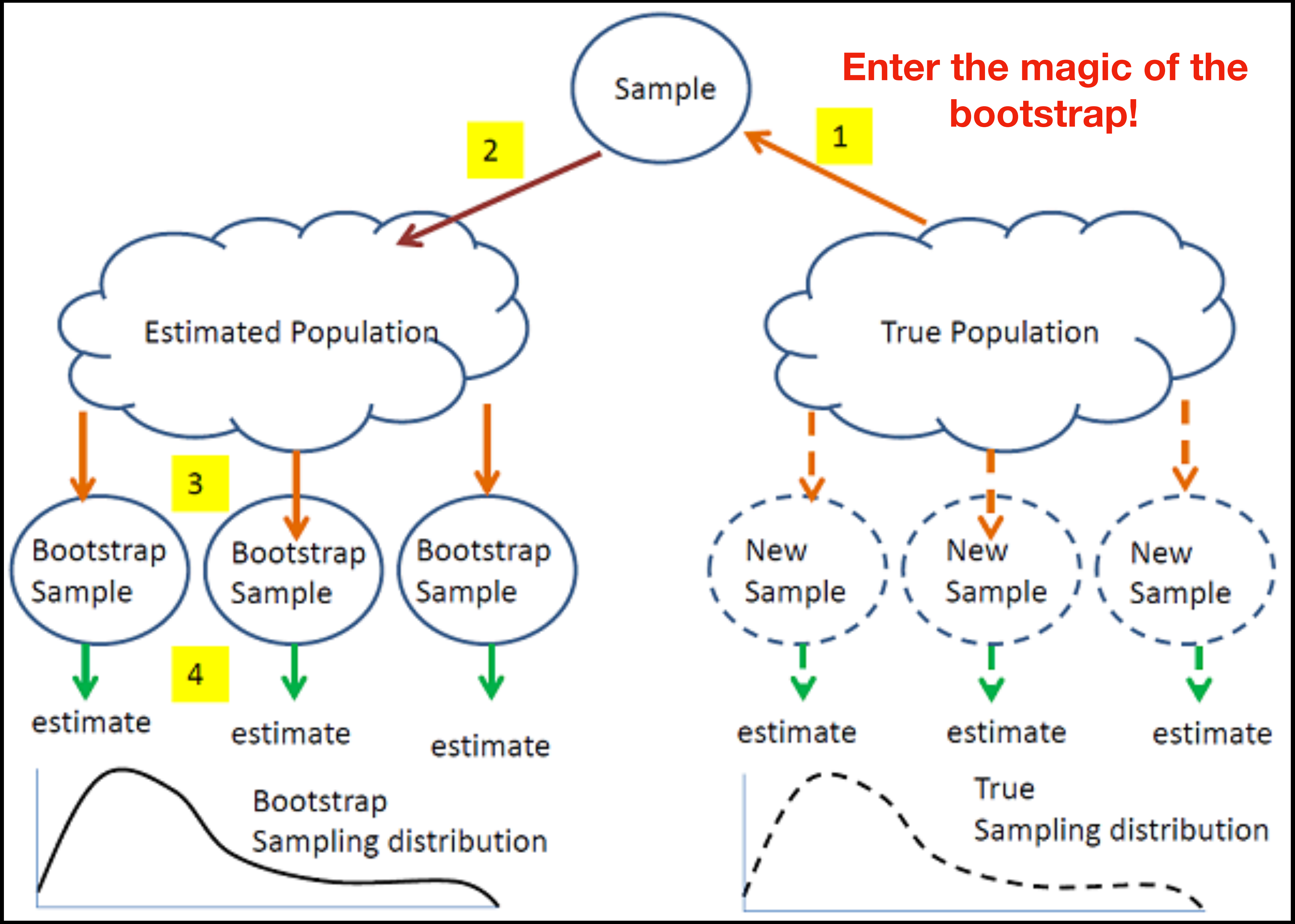
A mean, or proportion, etc

Distribution of means, proportions, etc

Bootstrapping

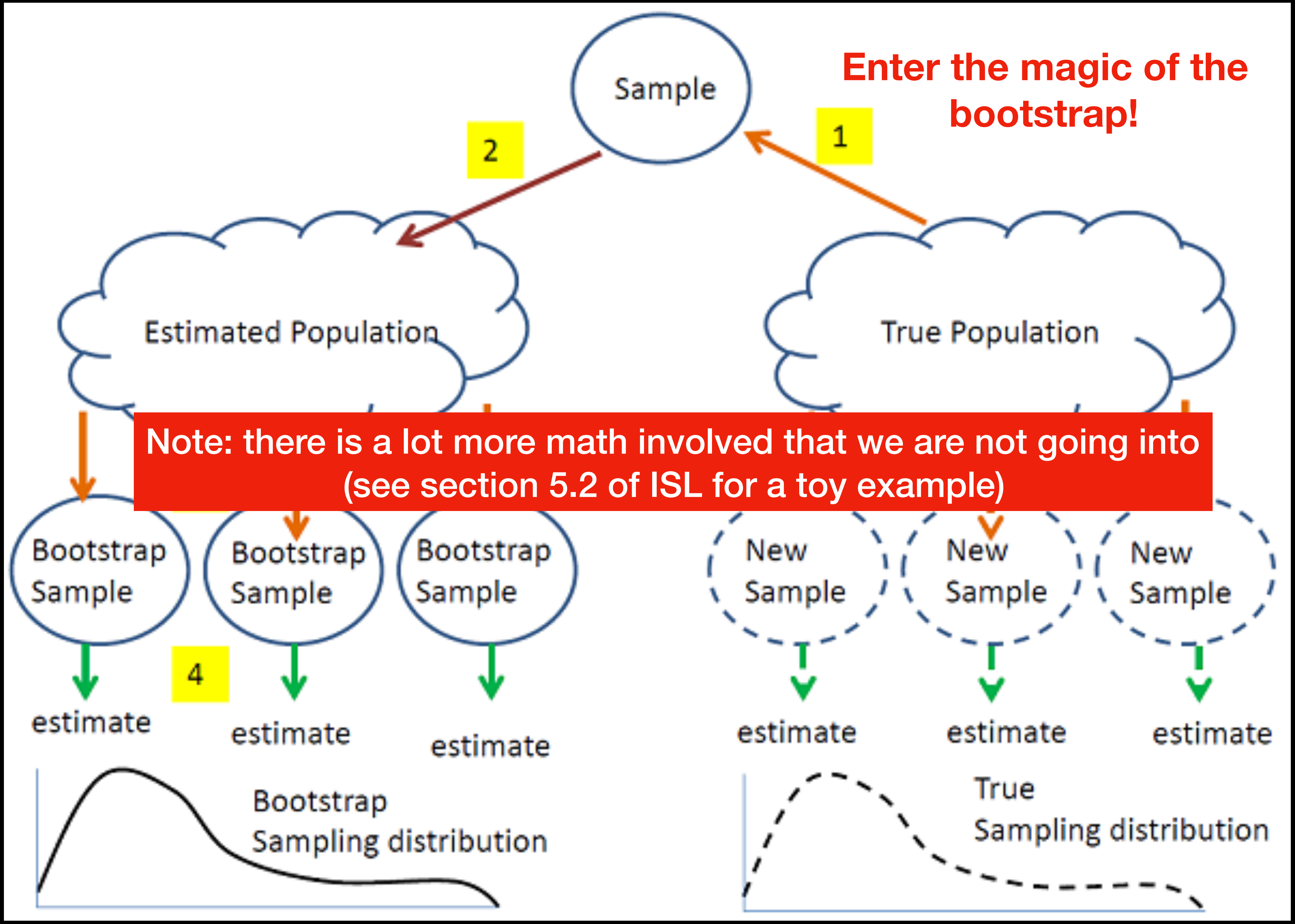


Bootstrapping



Distribution of means, proportions, etc

Bootstrapping



Distribution of means, proportions, etc