Welcome to week 14!

A quicky review!

Emphasis on Final will be material listed here & in todays lecture.

Hypothesis Testing Framework (Ch. 4-6)

The general outline of the process:

English

1. Set the hypotheses.
   For a single proportion this will look like:
   H_0: p = null value
   H_A: p < or > or ≠ null value
2. Check assumptions and conditions
3. Calculate a test statistic and a p-value
4. Make a decision, and interpret it in context

Provides a rigorous way to determine the answer with a specific level of confidence.

Math

- If p-value < α, reject H_0,
  there is sufficient evidence for [H_A]

- If p-value > α, do not reject H_0,
  there is not sufficient for evidence for [H_A]

English

Hypothesis Testing Framework (Ch. 4-6)

The general outline of the process:

1. Set the hypotheses.

   For a single proportion this will look like:

   H_0: p = null value

   H_A: p < or > or ≠ null value

2. Check assumptions and conditions
3. Calculate a test statistic and a p-value
4. Make a decision, and interpret it in context

- If p-value < α, reject H_0,

  there is sufficient evidence for [H_A]

- If p-value > α, do not reject H_0,

  there is not sufficient for evidence for [H_A]

(a) normal, large sample

(b) normal?, small sample

(c) observations & theory

<u>Test Statistics</u>

(a) Z-score -> P(Z)

(b) T-Score -> P(T)

(c) χ^2 -> P(χ^2)

Anatomy of a test statistic

The general form of a test statistic is

$$\frac{\text{point estimate} - \text{null value}}{\text{SE of point estimate}}$$

Only tricks are:

(1) picking what the point and null values are based on our hypotheses

(2) what the form of the standard errors is based on what our underlying distribution looks like (normal, t-distribution, χ^2)

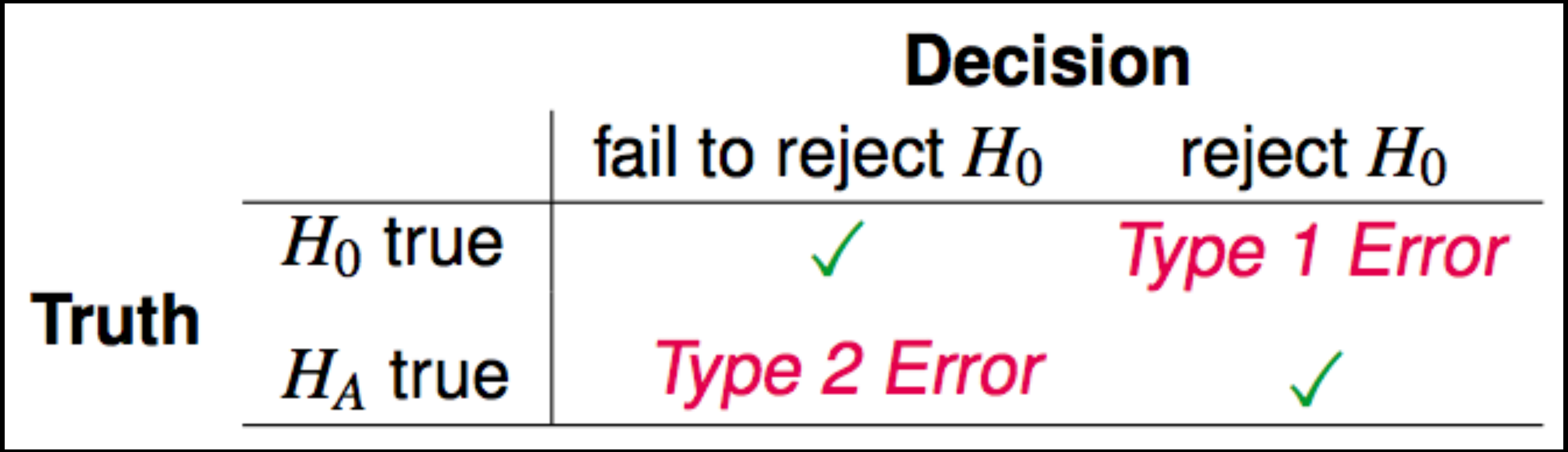
This construction is based on

- identifying the difference between a point estimate and an expected value if the null hypothesis was true, and
- standardizing that difference using the standard error of the point estimate.

These two ideas will help in the construction of an appropriate test statistic for count data.

Decision errors (cont.)

There are two competing hypotheses: the null and the alternative. In a hypothesis test, we make a decision about which might be true, but our choice might be incorrect.



		Decision	
		fail to reject H_0	reject H_0
Truth	H_0 true	✓	*Type 1 Error*
	H_A true	*Type 2 Error*	✓

A Type 1 Error is rejecting the null hypothesis when H_0 is true.
A Type 2 Error is failing to reject the null hypothesis when H_A is true.

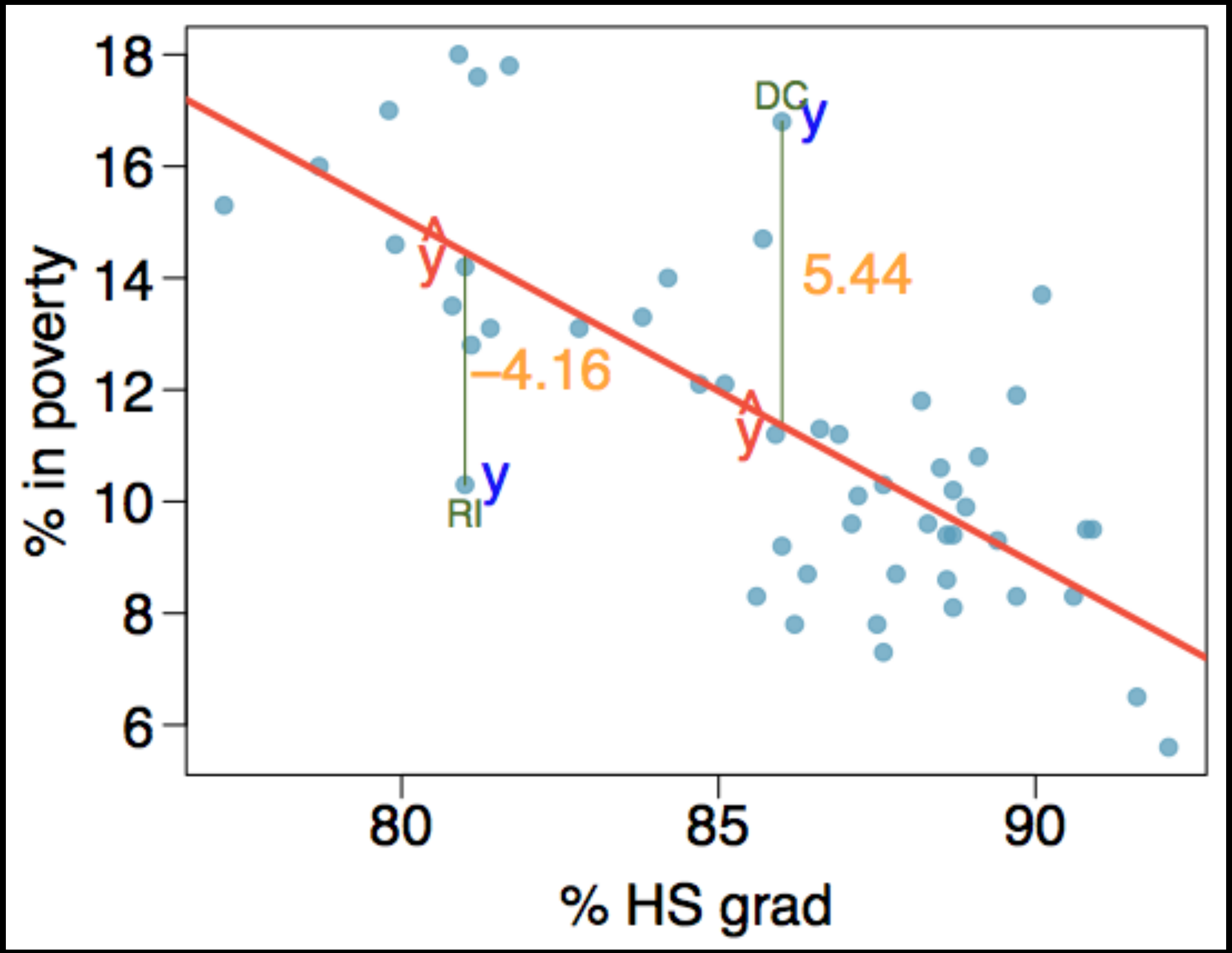
We (almost) never know if H_0 or H_A is true, but we need to consider all possibilities.

Residuals

Residuals are the leftovers from the model fit: Data = Fit + Residual

Aka a residual is the difference between the observed (y_i) and predicted $\hat{y}_i$.

$e_i = y_i - \hat{y}_i$

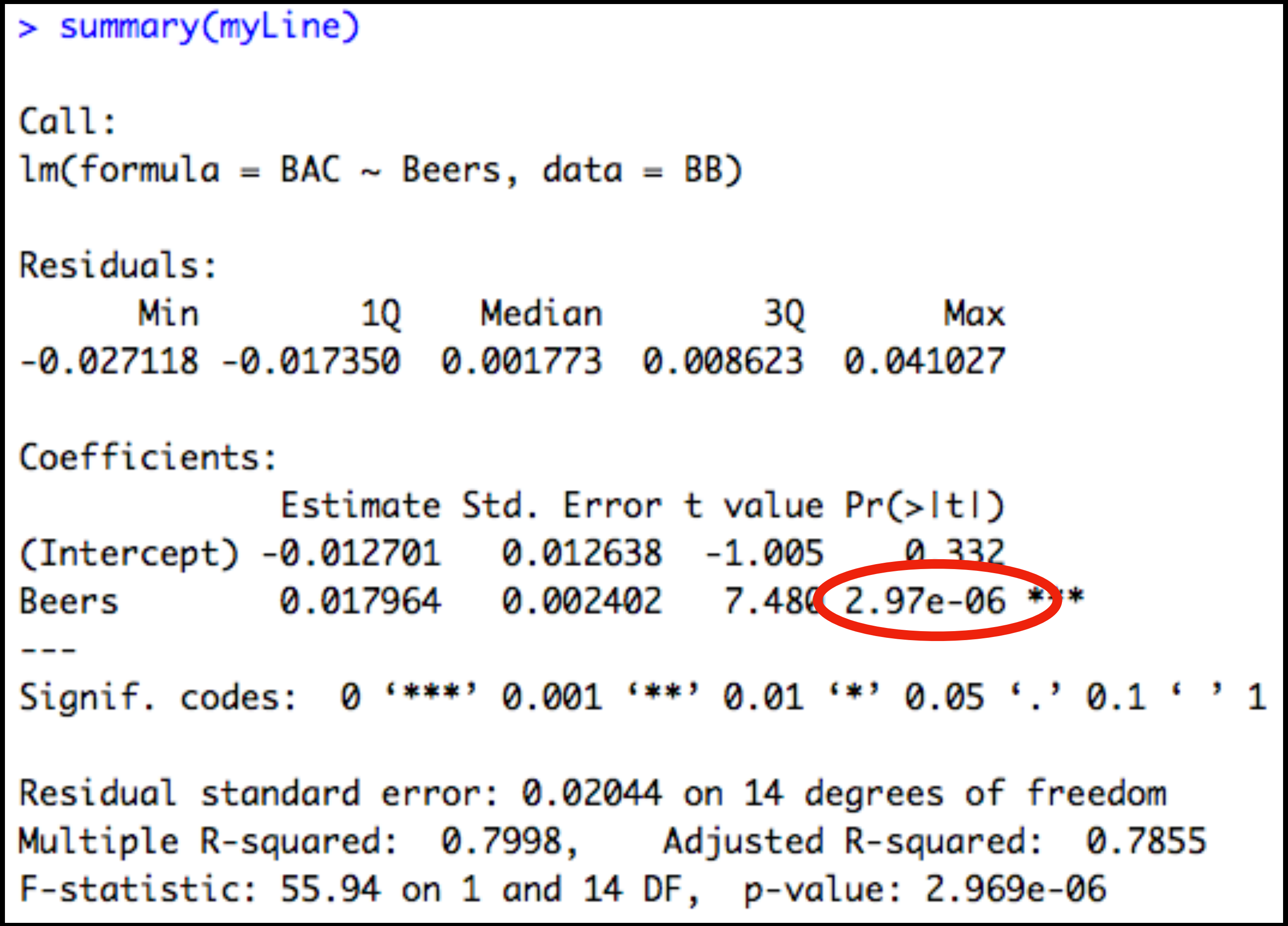


Here is a depiction of the residuals - how far each point is from our fitted line.

p-values for Linear Regression

What's really going on here? Just the same calculations we've been doing the past few weeks!

p-value > 0.05 so we fail to reject H_0



```
> summary(myLine)

Call:
lm(formula = BAC ~ Beers, data = BB)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.027118 -0.017350  0.001773  0.008623  0.041027 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.012701   0.012638  -1.005    0.332    
Beers        0.017964   0.002402   7.480 2.97e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02044 on 14 degrees of freedom
Multiple R-squared:  0.7998,	Adjusted R-squared:  0.7855 
F-statistic: 55.94 on 1 and 14 DF,  p-value: 2.969e-06
```

H_0: There is no relation between Beers and BAC - slope = 0
H_A: There is a relationship between Beers and BAC - slope != 0

Conditions to use MLR

1. Independence of observations of responses

2. Linearity of *all* variables - linear relationship between response variable and each of the explanatory variables

3. Multicollinearity checked for - does not mean we cannot use MLR, but we should be aware of how predictor/explanatory variables are related when quoting our results

4. Constant variance

5. Normality of Residuals

6. No influential points (outliers with strong leverage)

Logistic Regression: A Morbid Example

Logistic regression is a GLM used to model a binary categorical variable using numerical and categorical predictors.

We assume a binomial distribution produced the outcome variable and we therefore want to model *p* the probability of success for a given set of predictors.

To finish specifying the Logistic model we just need to establish a reasonable link function that connects η to *p*. There are a variety of options but the most commonly used is the logit function.

$$logit(p) = \log\left(\frac{p}{1-p}\right), \text{ for } 0 \le p \le 1$$

Logistic Regression: A Morbid Example

Ok, so what does the totality of our model look like?

$$y_i \sim \text{Binom}(p_i)$$

$$\eta = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

$$\text{logit}(p) = \eta$$

From which we back out the probability of survival based on parameters 1-n, for the *i*th observation:

$$p_i = \frac{\exp(\beta_0 + \beta_1 x_{1,i} + \cdots + \beta_n x_{n,i})}{1 + \exp(\beta_0 + \beta_1 x_{1,i} + \cdots + \beta_n x_{n,i})}$$

So far…

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

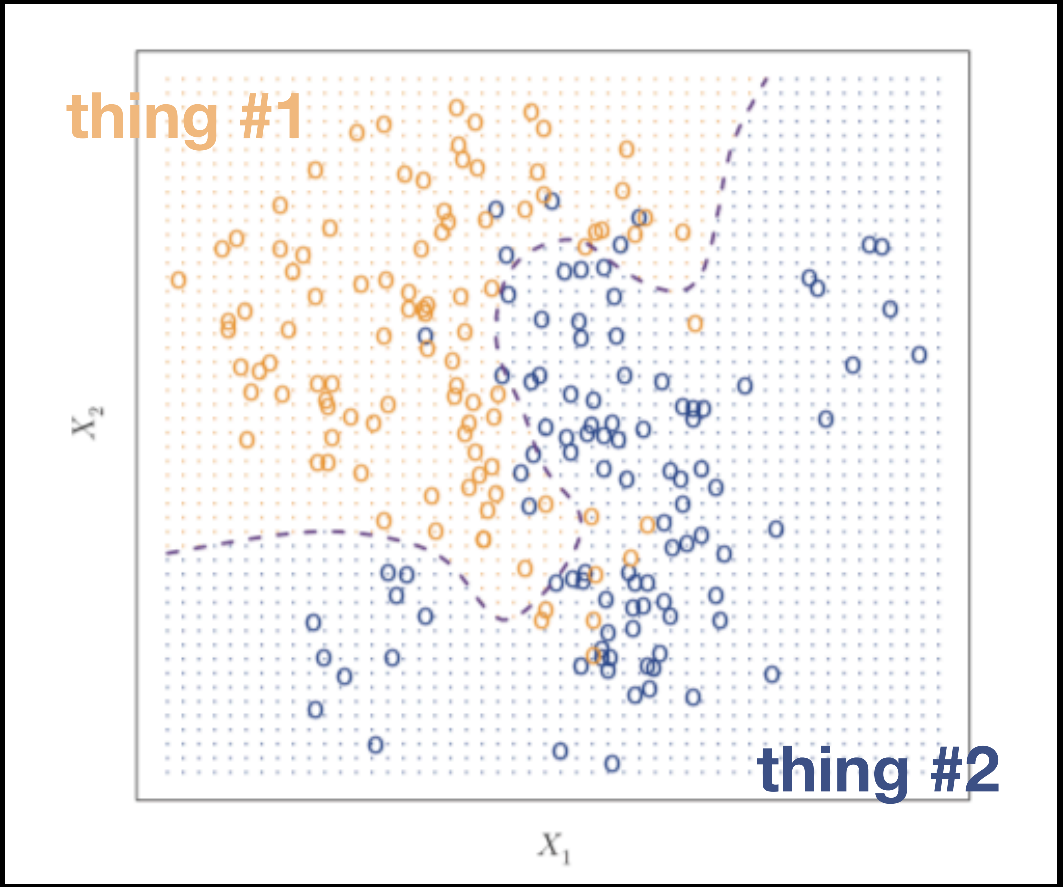
predicted y

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.

Bias-Variance Trade-Off (First Glance)

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

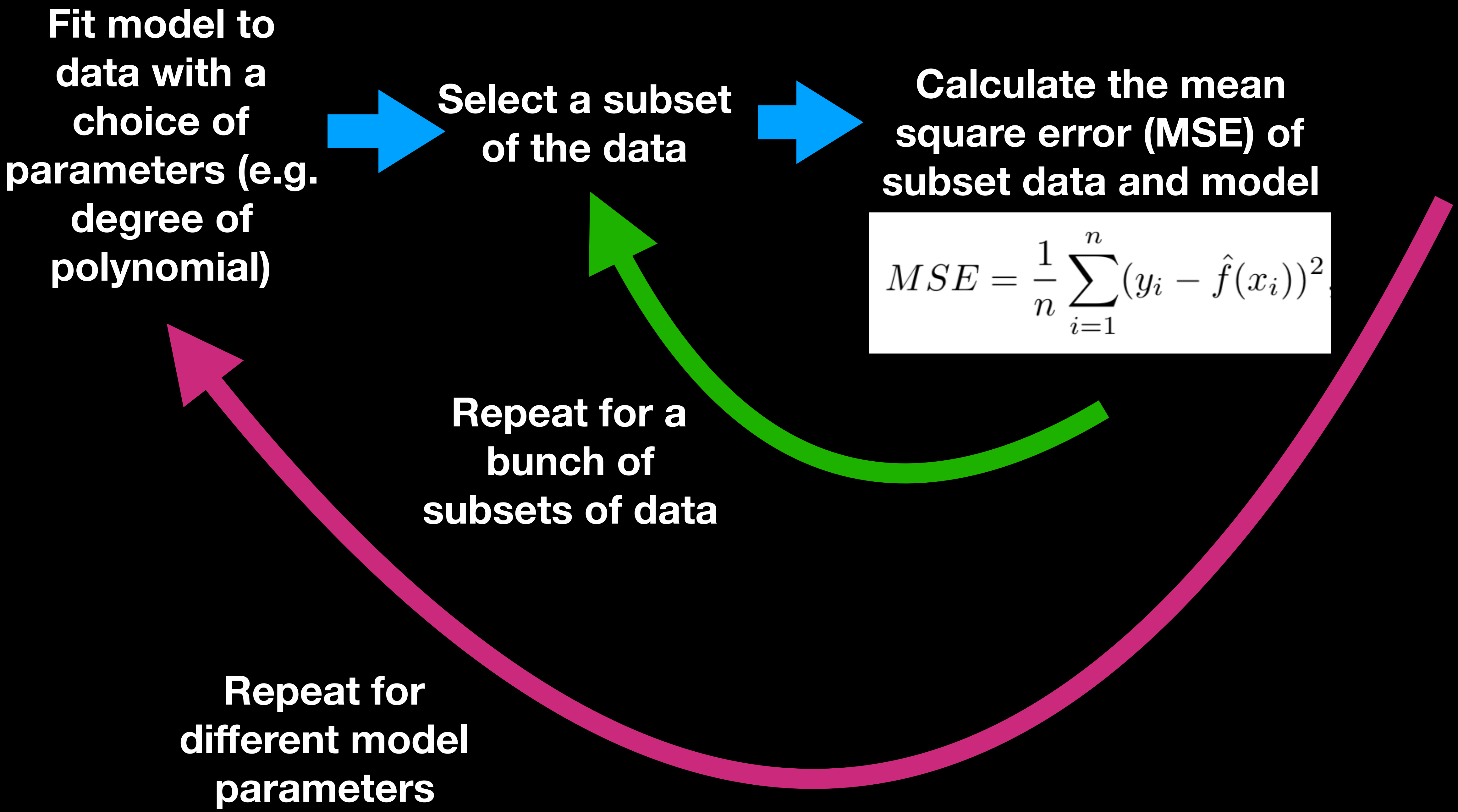
mean square error if we kept estimating response variable y by our fitted function of our explanatory variables, f(x) with different sample datasets at point x_0

how much our function, f, changes if we use a different random sample (variance)

Inherent error (bias) in the fact that any model is only an approximation to reality

Inherent error in our measurements

Cross-Validation Methods



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

Select a subset of the data

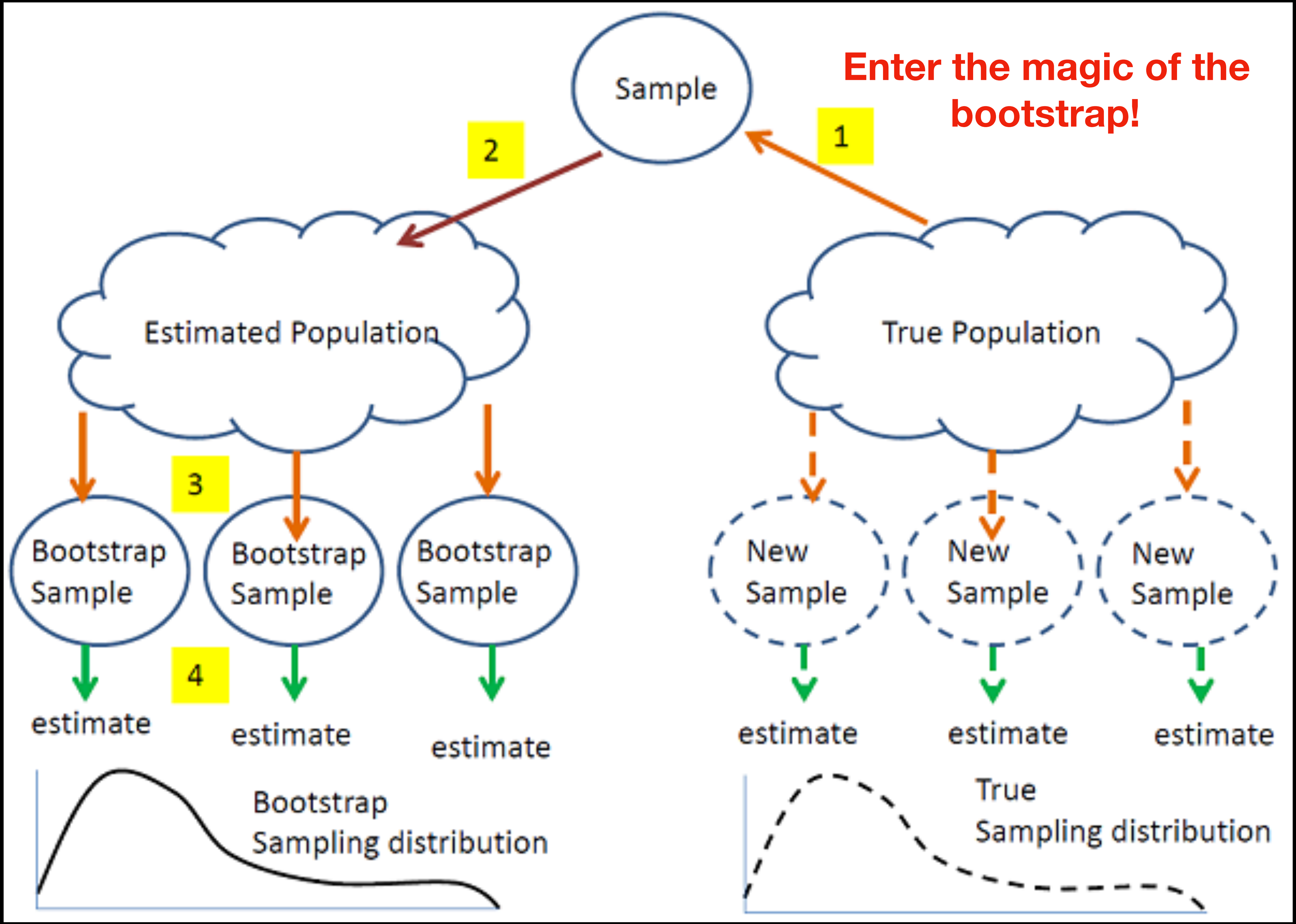
Calculate the mean square error (MSE) of subset data and model

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

Repeat for a bunch of subsets of data

Repeat for different model parameters

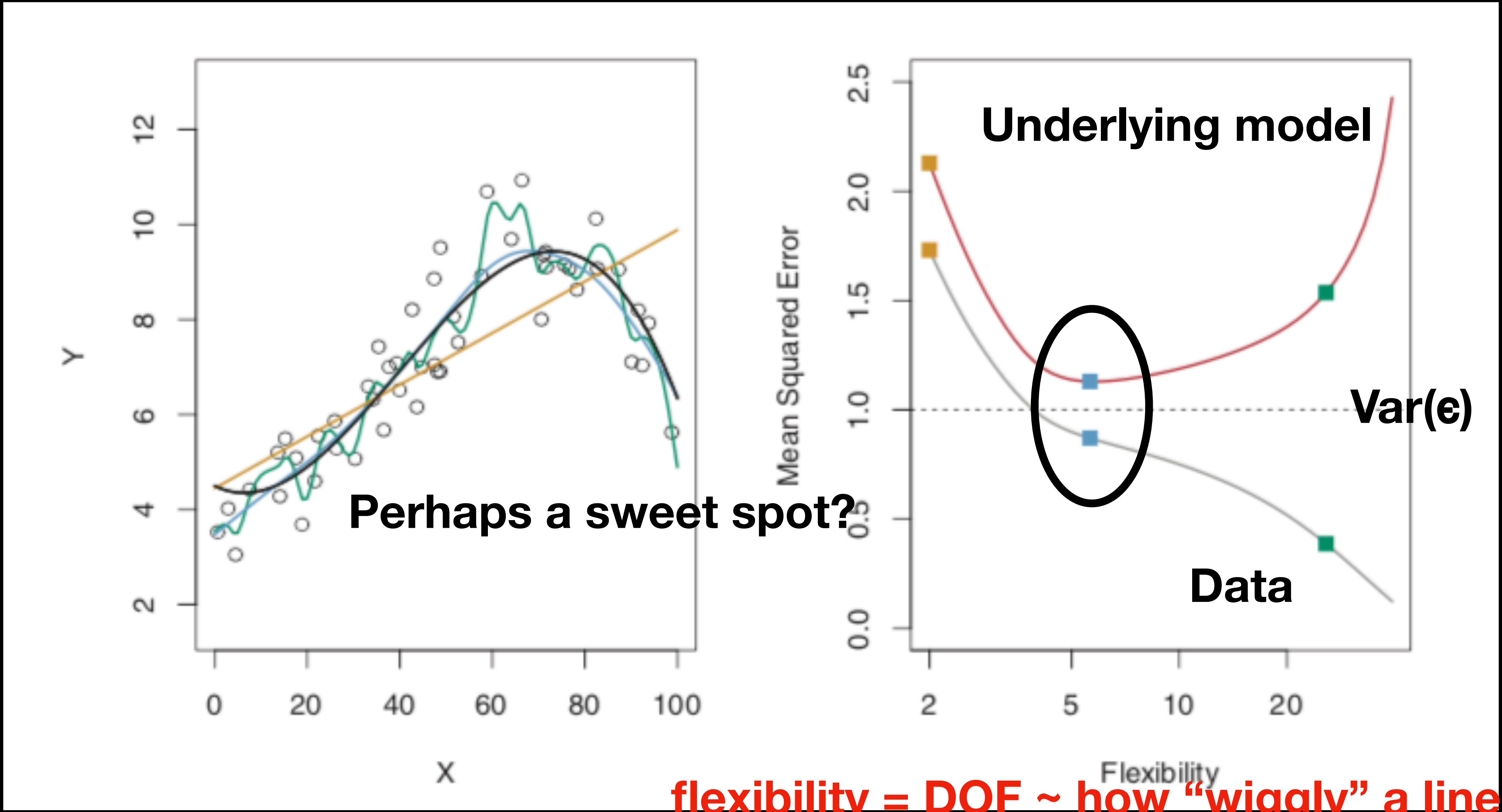
Bootstrapping



Distribution of means, proportions, etc

Bias-Variance Trade-Off (First Glance)

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



flexibility = DOF ~ how "wiggly" a line is

— Actual underlying function - y
o Simulated data with added error (ϵ)
— Linear fit
— Low "flexibility" smooth spline
— High "flexibility" smooth spline

fits data well, but underlying model badly

So far…

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

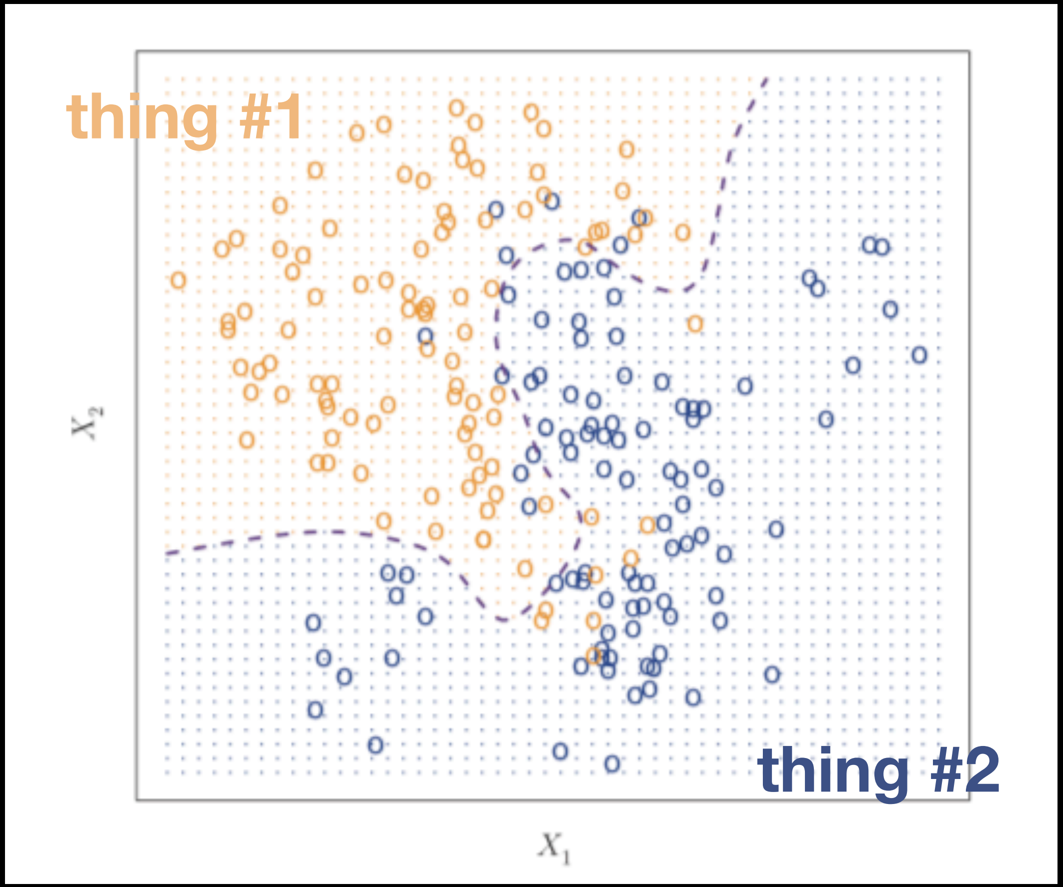
predicted y

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…

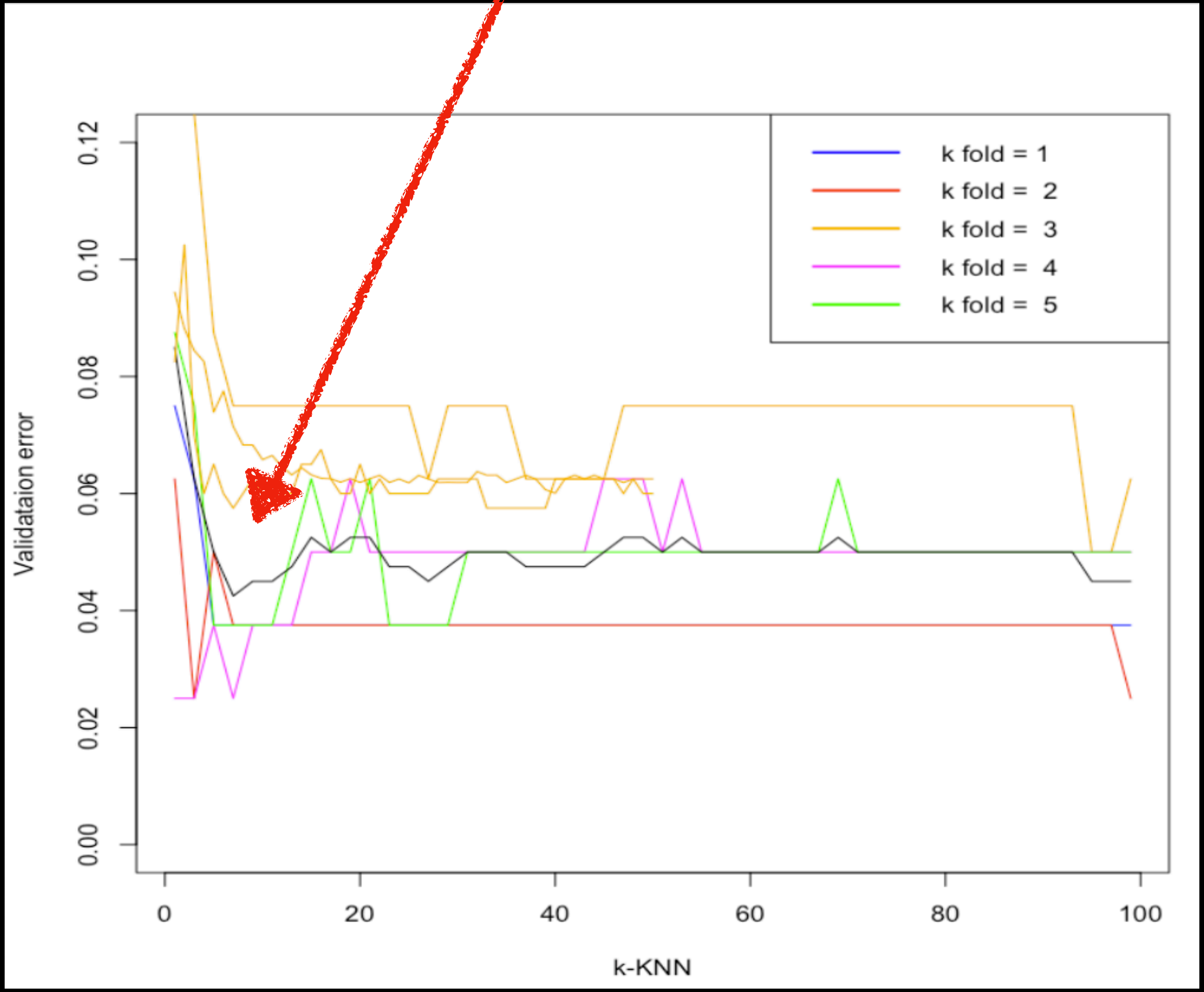


"I want to know where thing #1 and thing #2 live in some 2D space - computer, go figure out the boundary between these two things and let me know"

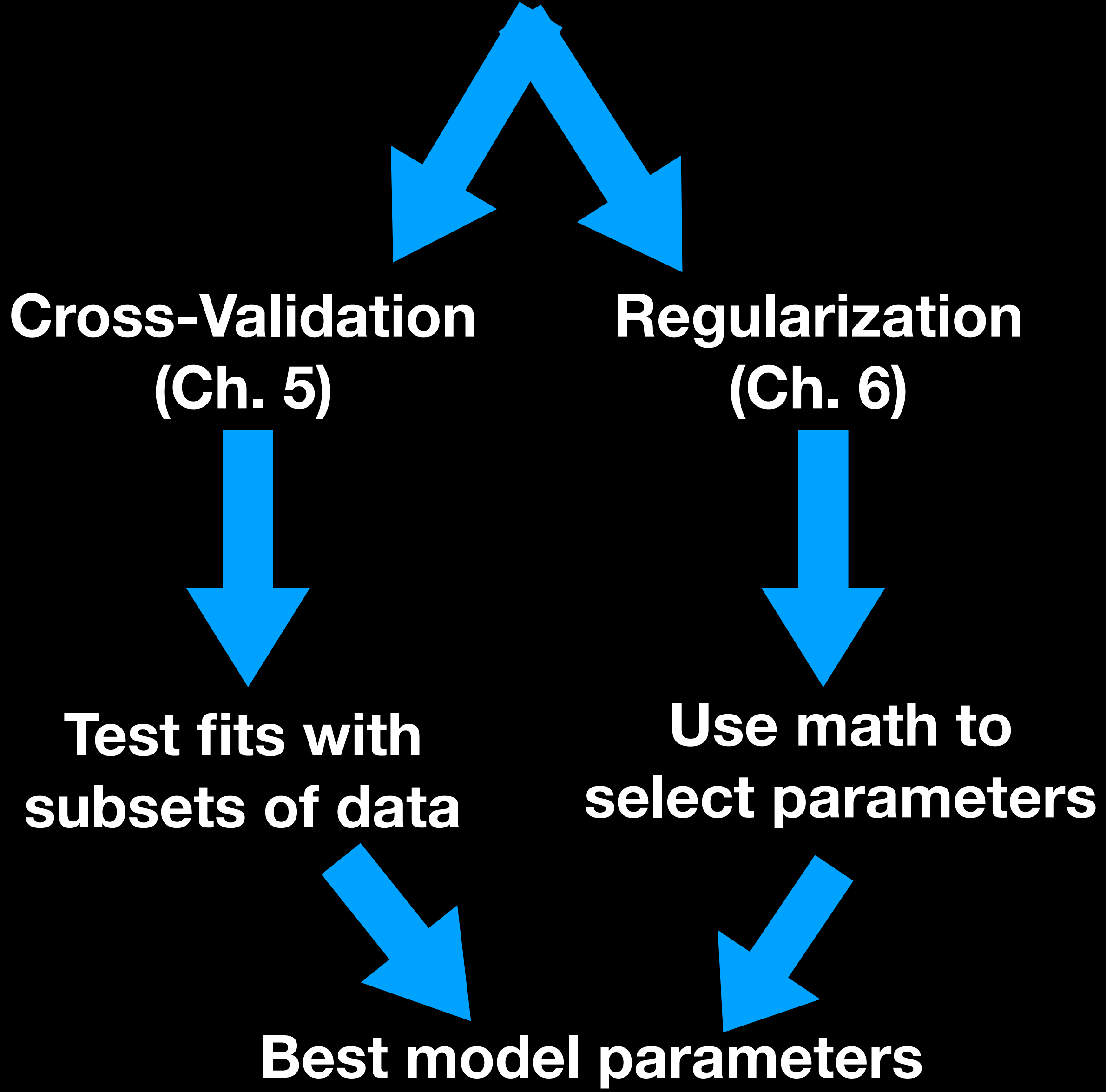
This is nice because we don't have to assume some model beforehand.

Cross-Validation Methods

Best KNN k is about here

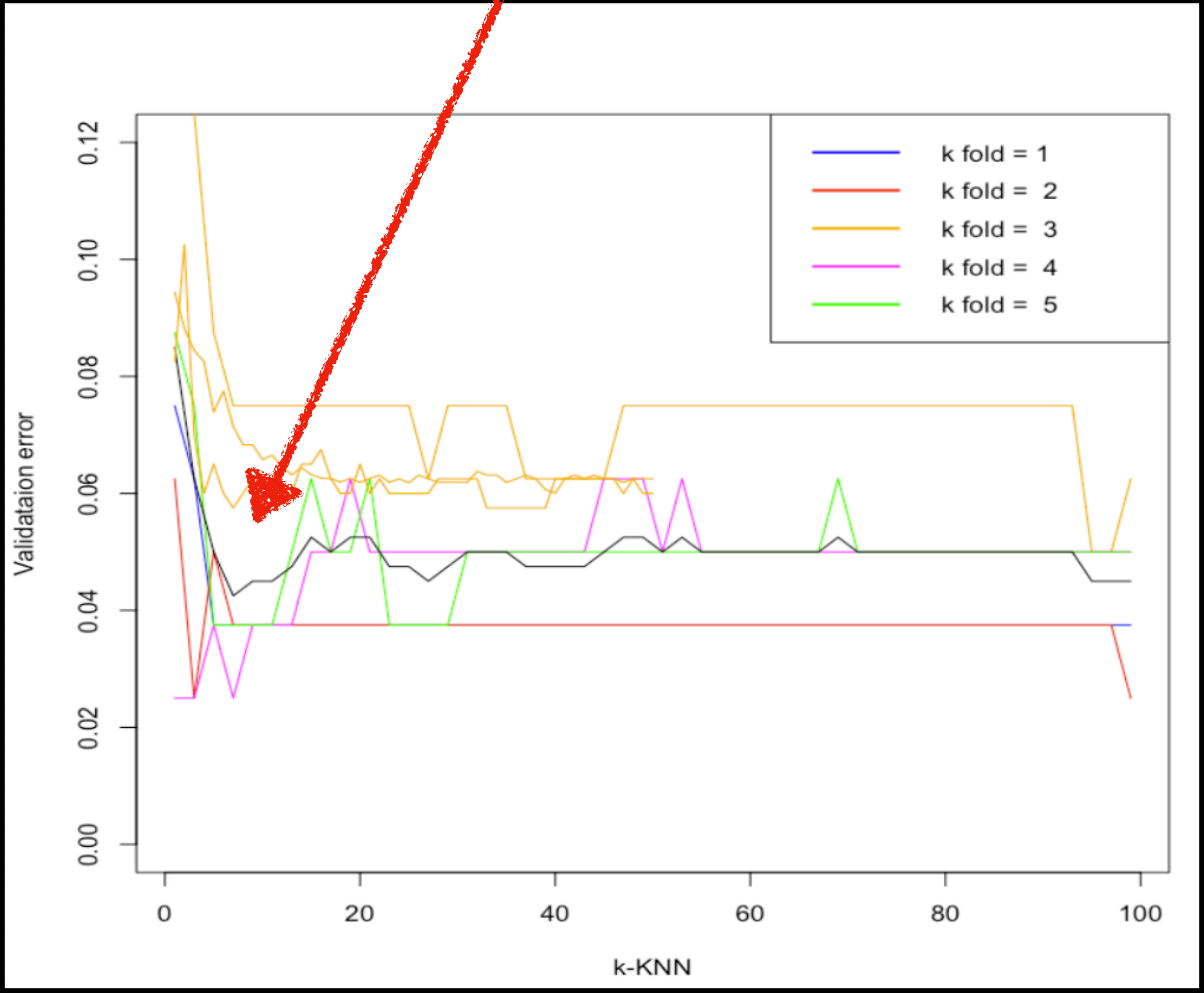


But we are only able to calculate the test error because we know the background distribution.

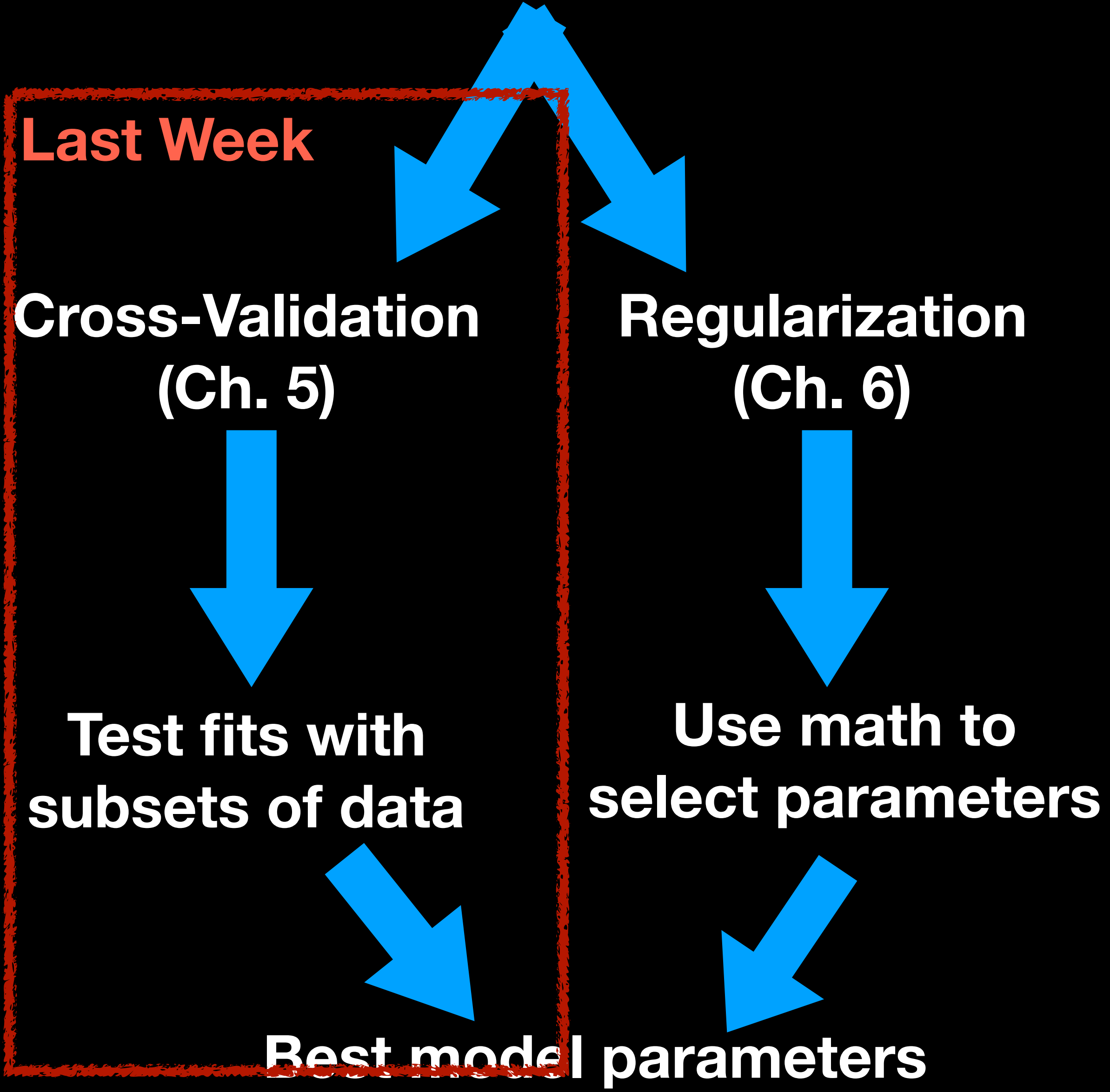


Cross-Validation Methods

Best KNN k is about here

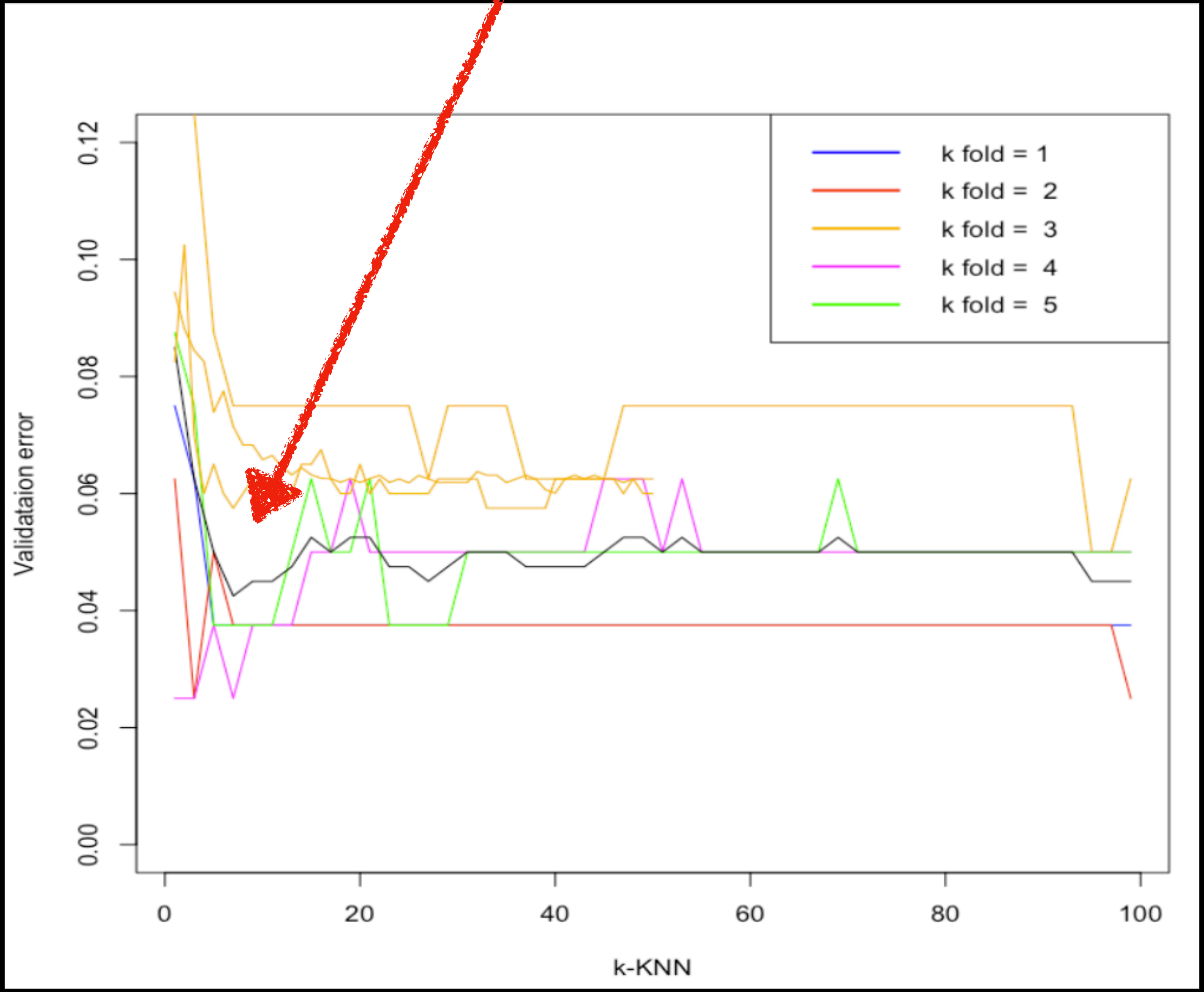


But we are only able to calculate the test error because we know the background distribution.

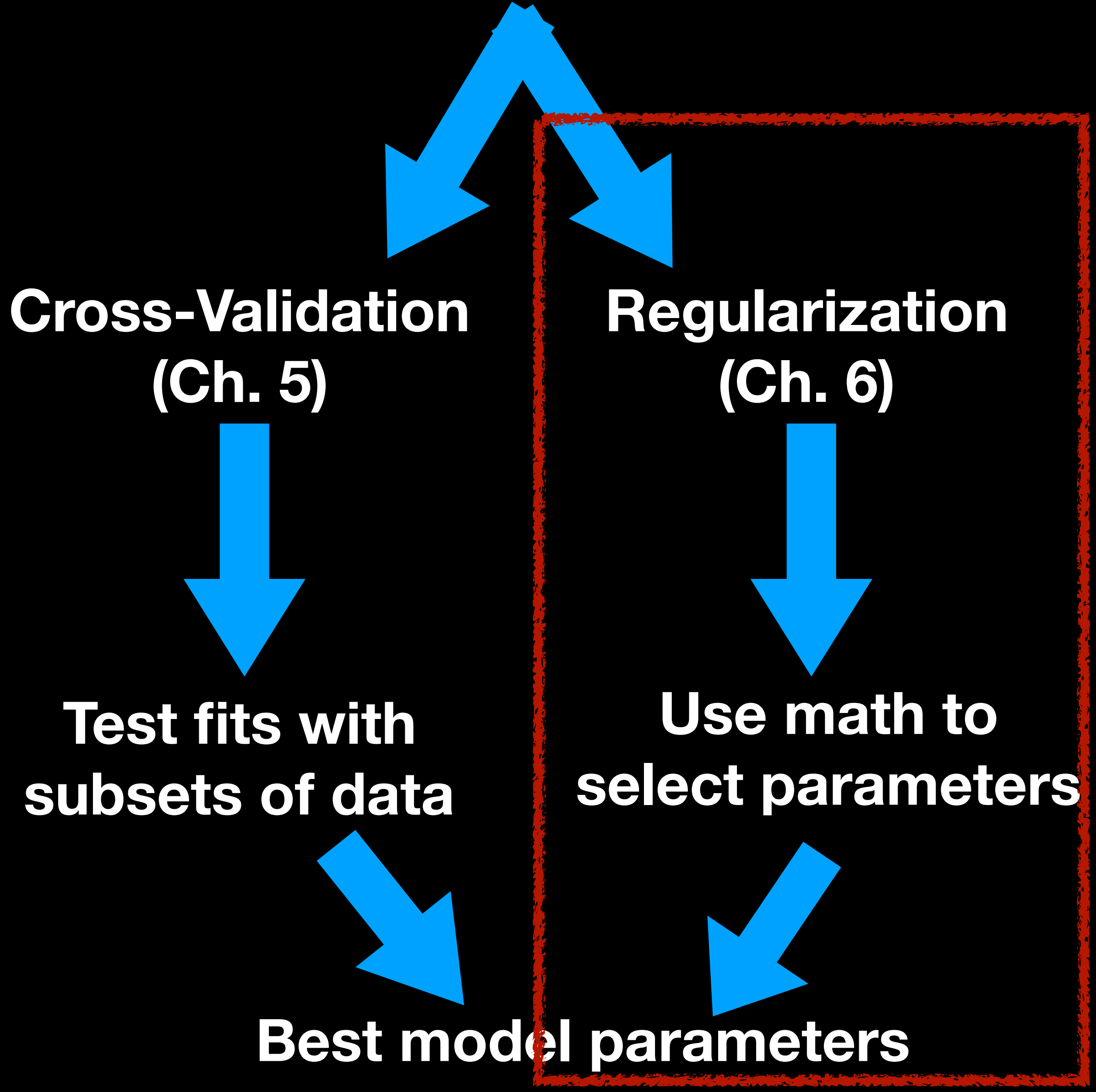


Cross-Validation Methods

Best KNN k is about here

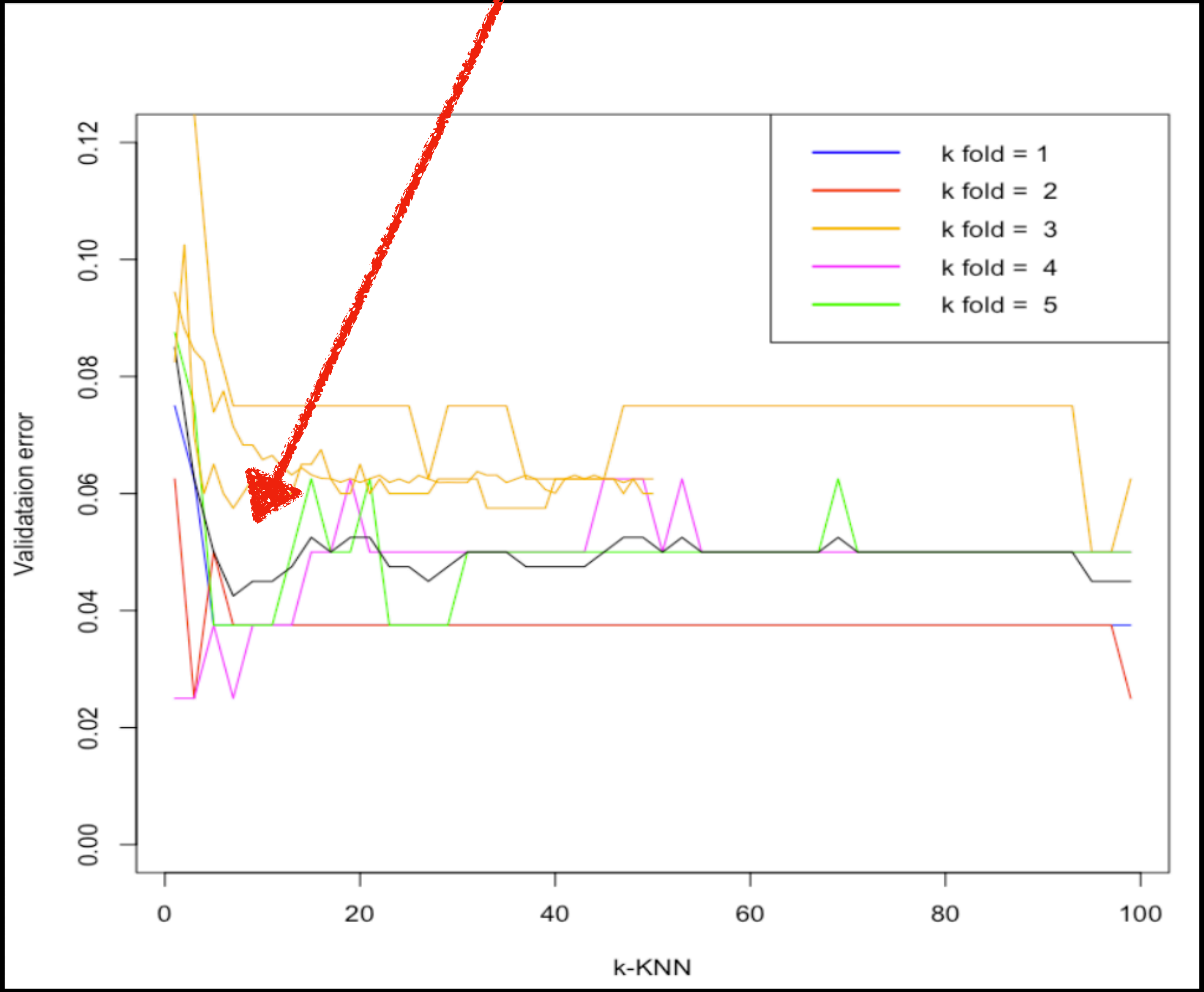


But we are only able to calculate the test error because we know the background distribution.

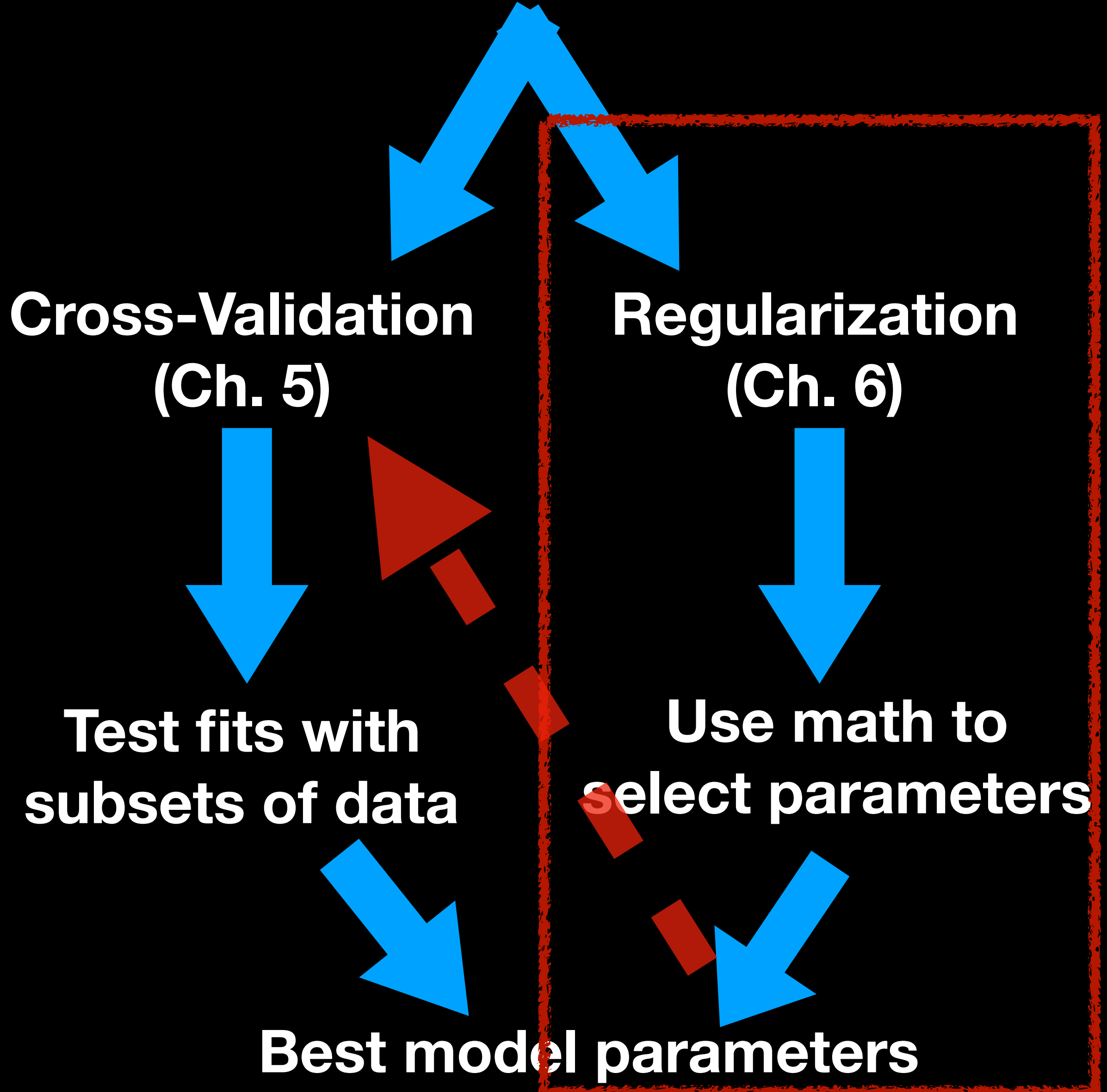


Cross-Validation Methods

Best KNN k is about here



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



Improvements on the Linear Model

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Linear models are great! (Interpretability & predictive performance)

But we can enhance them by modifying the process of choosing parameters - alternatives to least squares fitting

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Linear models are great! (Interpretability & predictive performance)

But we can enhance them by modifying the process of choosing parameters - alternatives to least squares fitting

Prediction Accuracy: especially when $p > n$, to control the variance.

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Linear models are great! (Interpretability & predictive performance)

But we can enhance them by modifying the process of choosing parameters - alternatives to least squares fitting

Prediction Accuracy: especially when $p > n$, to control the variance.

Model Interpretability: By removing irrelevant features — that is, by setting the corresponding coefficient estimates to zero — we can obtain a model that is more easily interpreted. We will present some approaches for automatically performing feature selection.

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Linear models are great! (Interpretability & predictive performance)

But we can enhance them by modifying the process of choosing parameters - alternatives to least squares fitting

Subset selection

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Linear models are great! (Interpretability & predictive performance)

But we can enhance them by modifying the process of choosing parameters - alternatives to least squares fitting

Subset selection

… we actually essentially already covered this!

Improvements on the Linear Model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

Linear models are great! (Interpretability & predictive performance)

But we can enhance them by modifying the process of choosing parameters - alternatives to least squares fitting

Subset selection

… we actually essentially already covered this!

A few more details…

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For k = 1,2,...p:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these p models, and call it M_k. Here k best is defined as having the smallest RSS, or equivalently largest R^2.

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For k = 1,2,...p:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these p models, and call it M_k. Here k best is defined as having the smallest RSS, or equivalently largest R^2.

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For k = 1,2,...p:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these p models, and call it M_k. Here k best is defined as having the smallest RSS, or equivalently largest R^2.

3. Select a single best model from among $M_0, \ldots, M_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For k = 1,2,...p:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these p models, and call it M_k. Here k best is defined as having the smallest RSS, or equivalently largest R^2.

3. Select a single best model from among $M_0, \ldots, M_p$ using cross-validated prediction error, $\boxed{C_p \text{ (AIC), BIC, or adjusted } R^2.}$

Different measures of goodness of fit - accounting how many parameters are fit

Subset selection

We identify a subset of the p predictors (of k possible) that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Brute Force Way:

1. Let M_0 denote the null model, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For k = 1,2,...p:

   (a) Fit all $\binom{p}{k}$ models that conta

   (b) Pick the best among these M_k. Here k best is defined as having the smallest RSS, or equivalently largest R^2.

this number can get big ~ 2^p
p=2: 4 (including null)
p=3: 8
...
p=10: 1024

3. Select a single best model from among M_0, . . . , M_p using cross-validated prediction error, $\boxed{C_p \text{ (AIC), BIC, or adjusted } R^2.}$

Different measures of goodness of fit - accounting how many parameters are fit

Subset selection

- For computational reasons, best subset selection cannot be applied with very large p.

Subset selection

- For computational reasons, best subset selection cannot be applied with very large p.
- Best subset selection may also suffer from statistical problems when p is large: *larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.*

Subset selection

- For computational reasons, best subset selection cannot be applied with very large p.
- Best subset selection may also suffer from statistical problems when p is large: *larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.*
- Thus an enormous search space can lead to overfitting and high variance of the coefficient estimates.

Subset selection

- For computational reasons, best subset selection cannot be applied with very large p.
- Best subset selection may also suffer from statistical problems when p is large: *larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.*
- Thus an enormous search space can lead to overfitting and high variance of the coefficient estimates.
- For both of these reasons, stepwise methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Subset selection

- For computational reasons, best subset selection cannot be applied with very large p.

- Best subset selection may also suffer from statistical problems when p is large: *larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.*

- Thus an enormous search space can lead to overfitting and high variance of the coefficient estimates.

- For both of these reasons, stepwise methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Great news: we already covered this!

Subset selection: Forward & Backward (Review)

Subset selection: Forward & Backward (Review)

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.

Subset selection: Forward & Backward (Review)

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.

Subset selection: Forward & Backward (Review)

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.
- Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection.

Subset selection: Forward & Backward (Review)

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.
- Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection.
- However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Subset selection: Forward & Backward (Review)

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.
- Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection.
- However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.
- *Neither are guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.*

Subset selection: Forward & Backward (Review)

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.
- Like forward stepwise selection, backward stepwise selection provides an efficient alternative to best subset selection.
- However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.
- *Neither are guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.*
- Backward selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

Subset selection: Forward & Backward (Review)

How do we quantify how “good” each of our forward/backward selected models are?

Subset selection: Forward & Backward (Review)

How do we quantify how “good” each of our forward/backward selected models are?

Find models that minimize:

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

Subset selection: Forward & Backward (Review)

Find models that minimize:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

Subset selection: Forward & Backward (Review)

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Find models that minimize:

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

d predictors, n data points

estimate from SE²/n = σ² or RSE

Subset selection: Forward & Backward (Review)

Find models that minimize:

$$\text{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

$$C_p = \frac{1}{n}\left(\text{RSS} + 2d\hat{\sigma}^2\right)$$

d predictors, n data points

estimate from $SE^2/n = \sigma^2$ or RSE

Idea here is to get a measurement of the mean square error that accounts for # of parameters in a model

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
- von Neumann

Subset selection: Forward & Backward (Review)

Find models that minimize:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

d predictors, n data points

estimate from $SE^2/n = \sigma^2$ or RSE

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

Idea here is to get a measurement of the mean square error that accounts for # of parameters in a model

$$\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + \log(n)d\hat{\sigma}^2\right)$$

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
- von Neumann

Or: large values as R_{adj}^2

Subset selection: Forward & Backward (Review)

Find models that minimize:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

d predictors, n data points

estimate from $SE^2/n = \sigma^2$ or RSE

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

Idea here is to get a measurement of the mean square error that accounts for # of parameters in a model

$$\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + \log(n)d\hat{\sigma}^2\right)$$

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
- von Neumann

Or: large values as R_{adj}^2

Optional R notes.

Shrinkage Methods

Shrinkage Methods

Lets recall how we find a linear model:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Least squares minimization (i.e. minimize RSS)

Shrinkage Methods

Lets recall how we find a linear model:

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Least squares minimization (i.e. minimize RSS)

In contrast, the Ridge or Lasso regression coefficient estimates β^R and β^L are the values that minimize

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Ridge

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \mathrm{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Lasso

where $\lambda \geq 0$ is a tuning parameter, to be determined separately.

Shrinkage Methods

Lets recall how we find a linear model:

$$\mathrm{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

Least squares minimization (i.e. minimize RSS)

In contrast, the Ridge or Lasso regression coefficient estimates β^R and β^L are the values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

Don't panic, its just a bit more math (and we'll get R to do it for us)

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

where $\lambda \geq 0$ is a tuning parameter, to be determined separately.

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\text{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\text{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

Why would we make our lives more complicated like this?

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

Shrinkage Methods

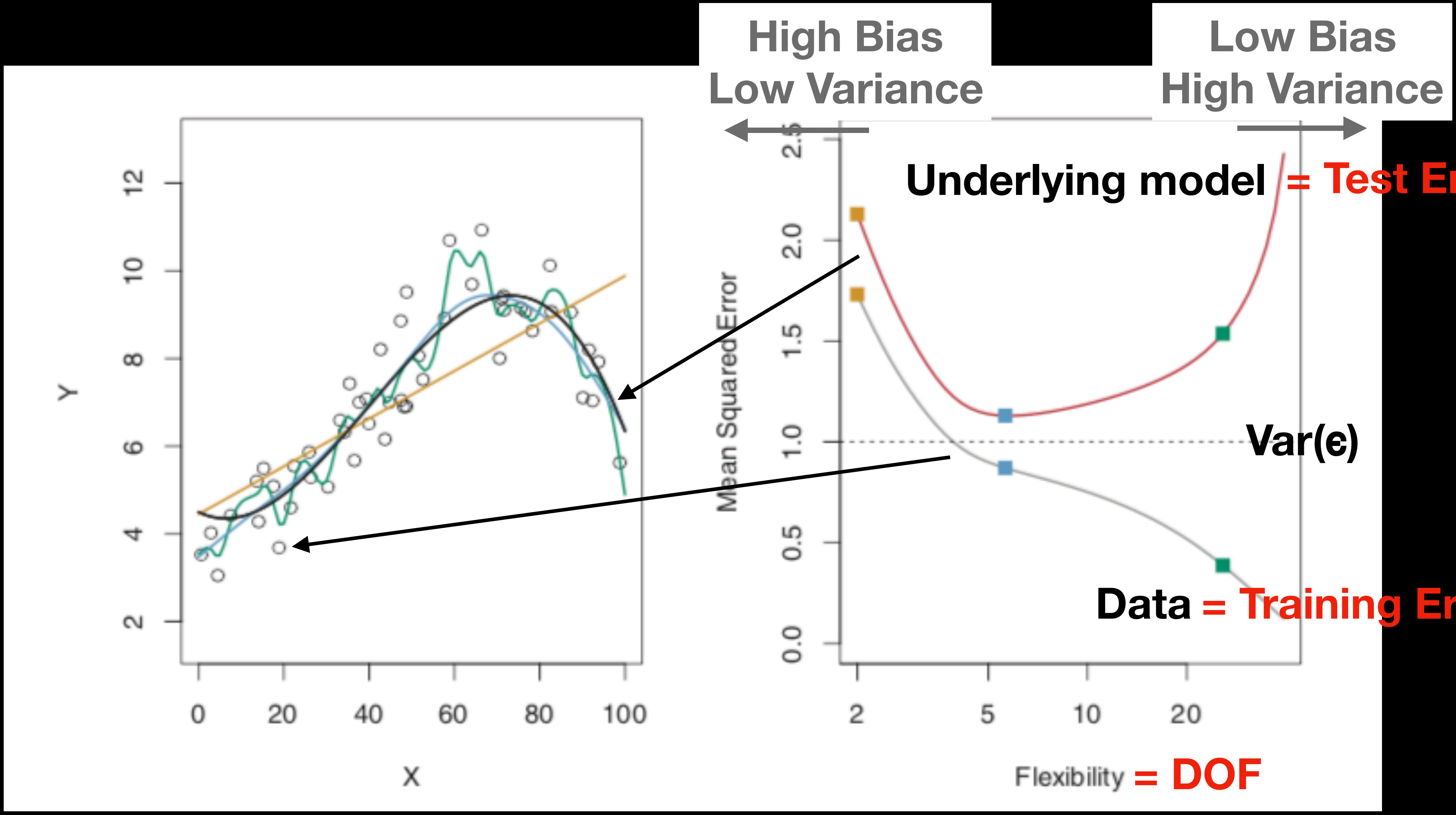
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

Why would we make our lives more complicated like this?



Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

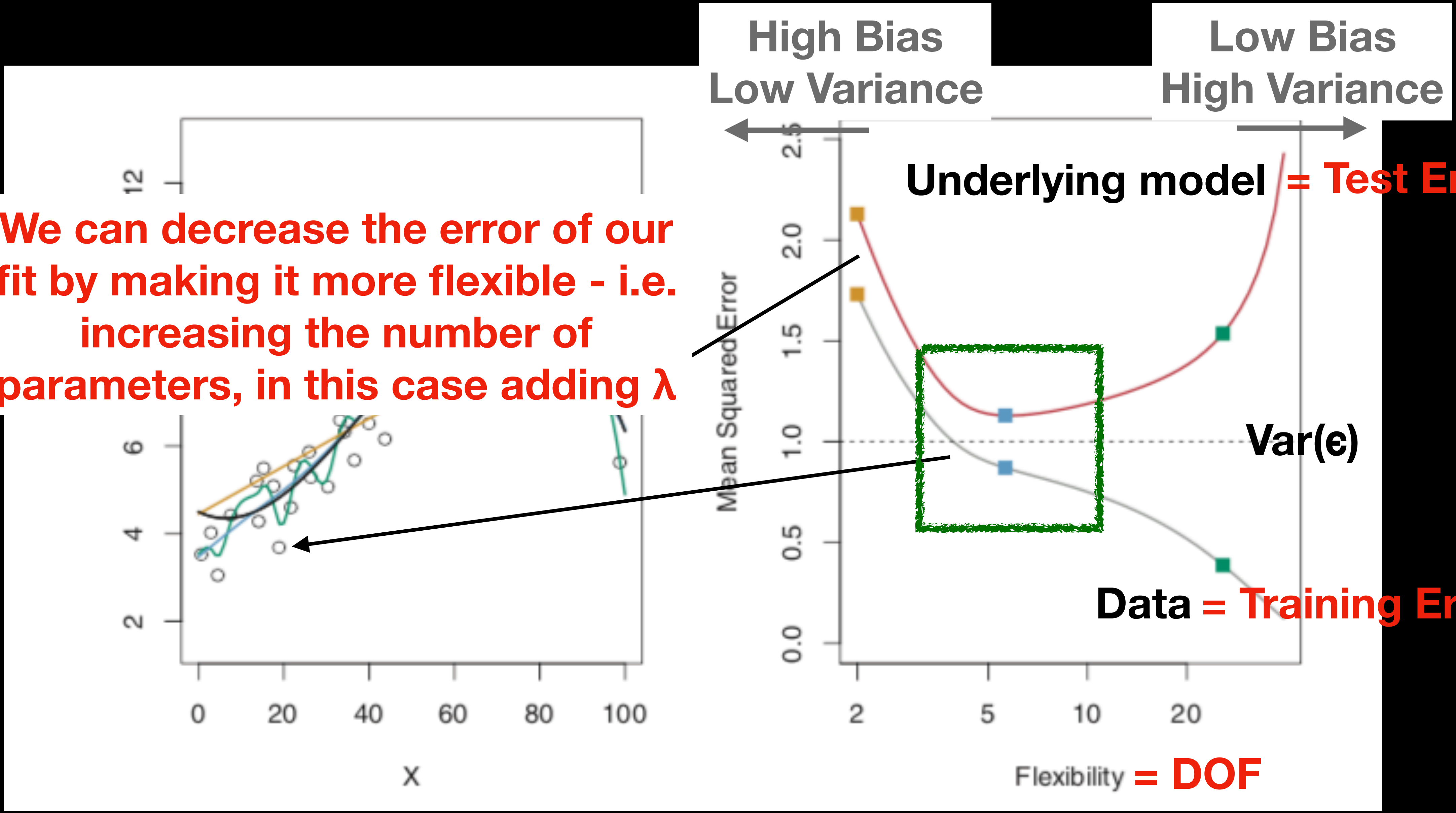
Ridge

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

Why would we make our lives more complicated like this?

We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding λ



Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

Why would we make our lives more complicated like this?

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding λ

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

Why would we make our lives more complicated like this?

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding λ
(2) Natural way to perform variable selection

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

Why would we make our lives more complicated like this?

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding λ

(2) Natural way to perform variable selection $\longrightarrow$ "Shrinkage" of less important parameters to zero

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

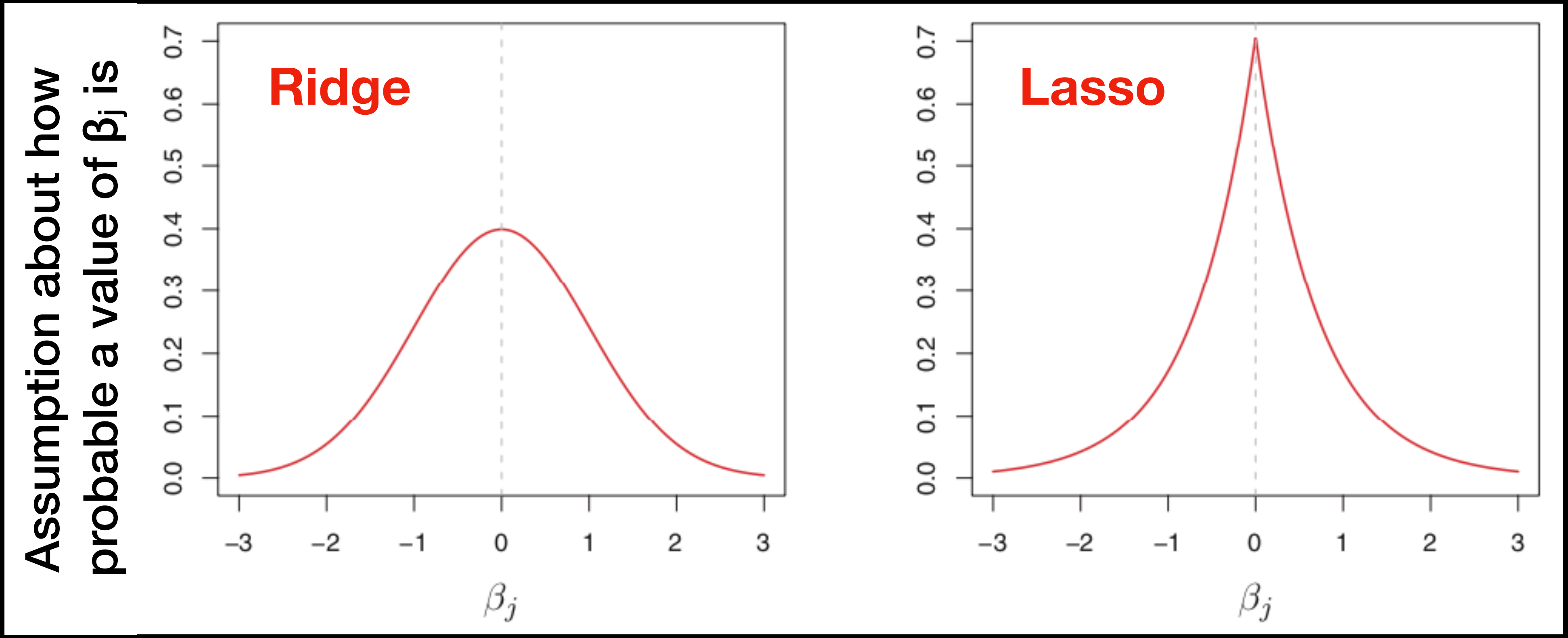
Why would we make our lives more complicated like this?

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding λ

(2) Natural way to perform variable selection → "Shrinkage" of less important parameters to zero



Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

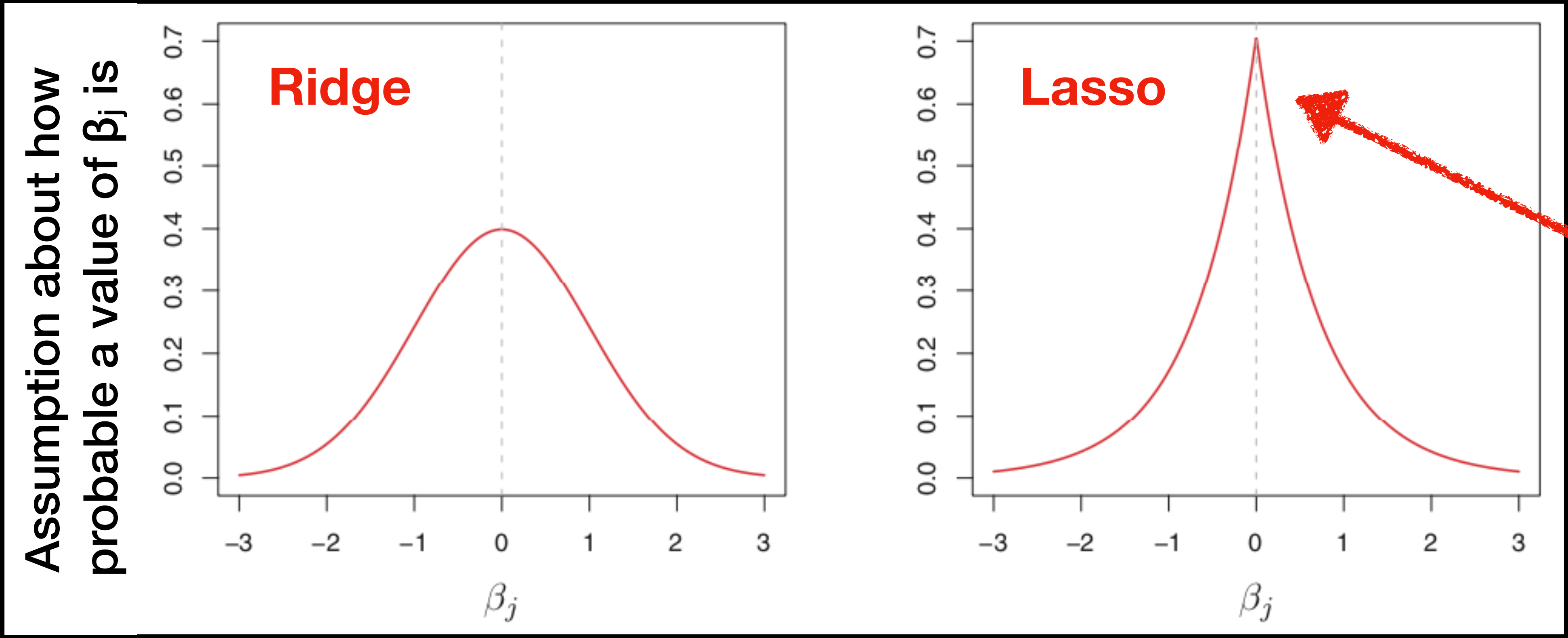
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

Why would we make our lives more complicated like this?

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding λ

(2) Natural way to perform variable selection → **"Shrinkage" of less important parameters to zero**



Parameters are more likely to be zero

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|$$

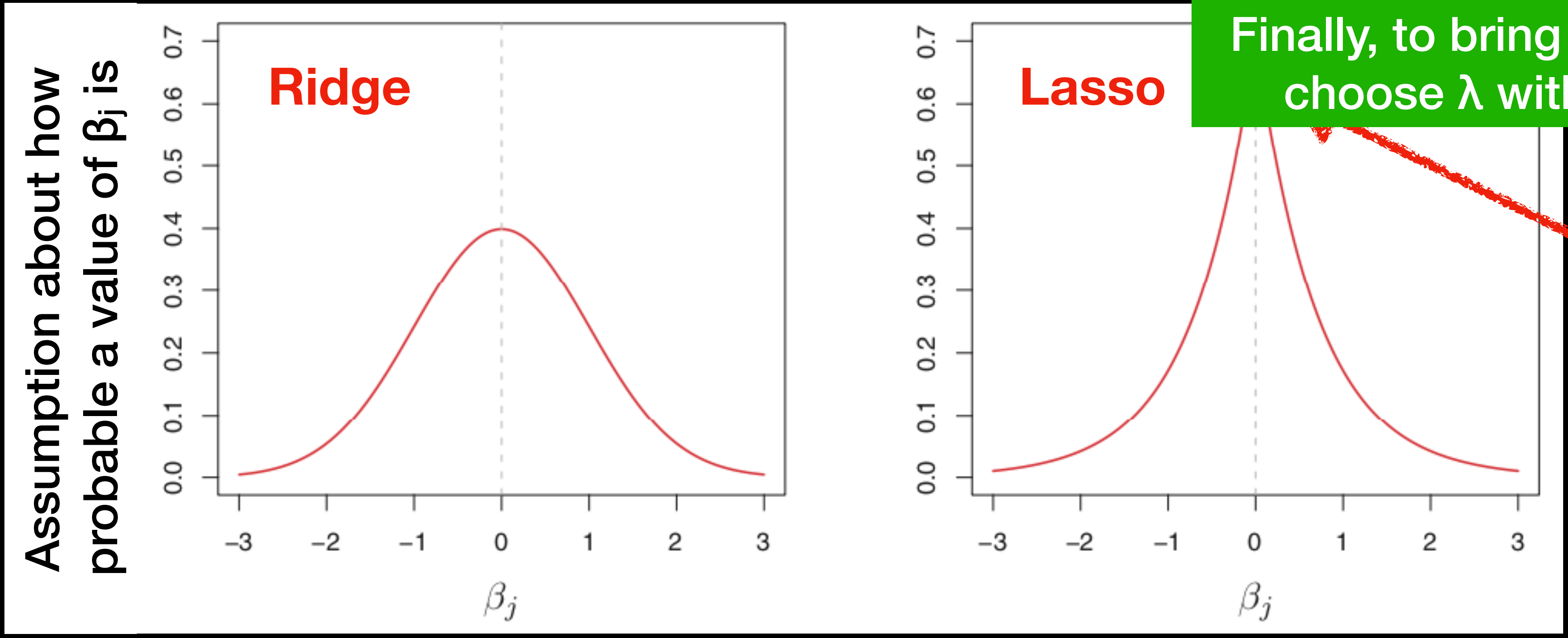
Lasso

Why would we make our lives more complicated like this?

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding $\boxed{\lambda}$

(2) Natural way to perform variable selection → **"Shrinkage" of less important parameters to zero**

Assumption about how probable a value of β_j is



Finally, to bring it full circle - we can choose λ with Cross-Validation

Parameters are more likely to be zero

Shrinkage Methods

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\text{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge

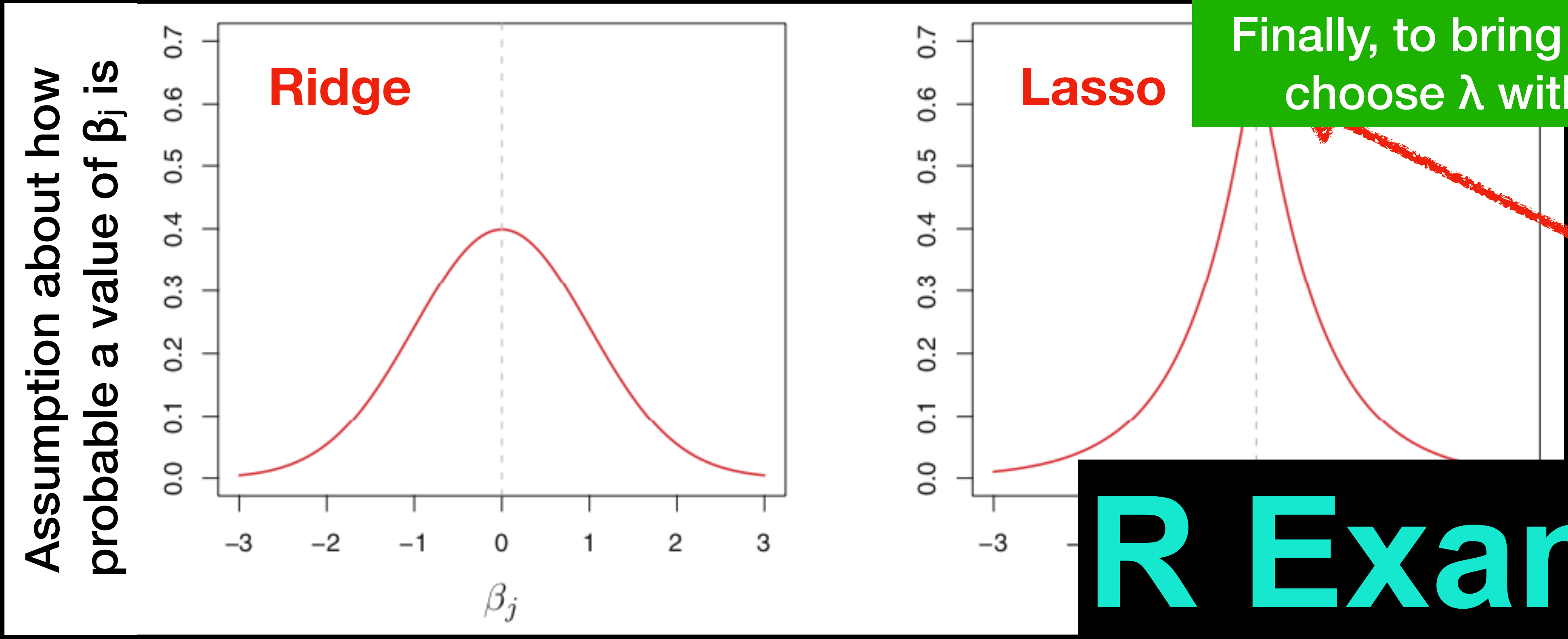
Why would we make our lives more complicated like this?

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\text{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso

(1) We can decrease the error of our fit by making it more flexible - i.e. increasing the number of parameters, in this case adding $\boxed{\lambda}$

(2) Natural way to perform variable selection → "Shrinkage" of less important parameters to zero



Finally, to bring it full circle - we can choose λ with Cross-Validation

Parameters are more likely to be zero

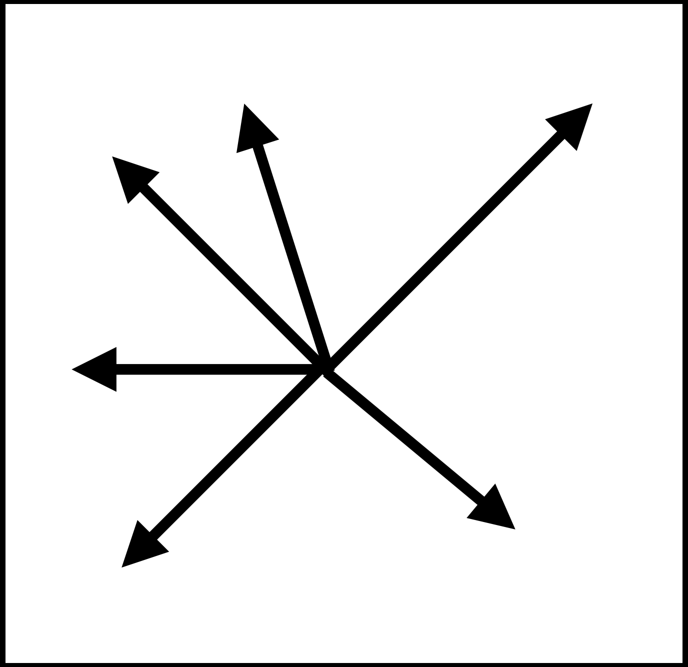
R Examples!

Unsupervised Learning:
An intro to Principle Component Analysis

Unsupervised Learning:
An intro to Principle Component Analysis

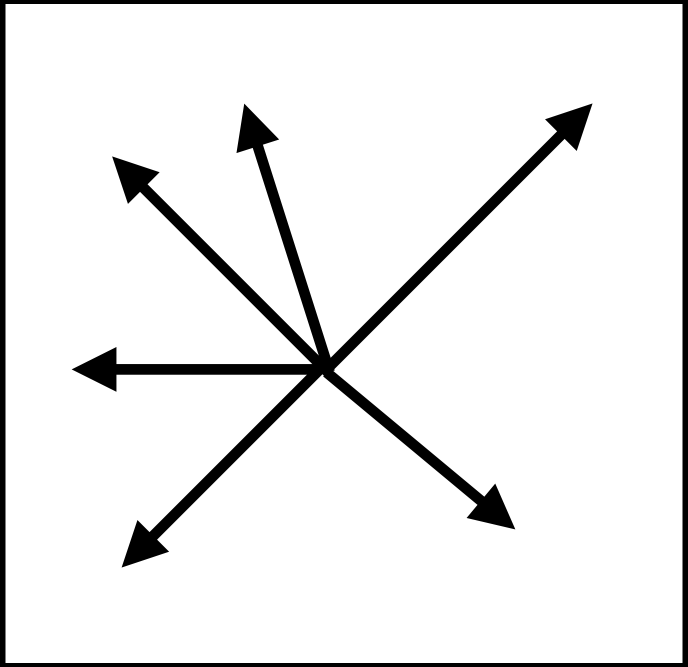
… what do we do when we don’t know anything

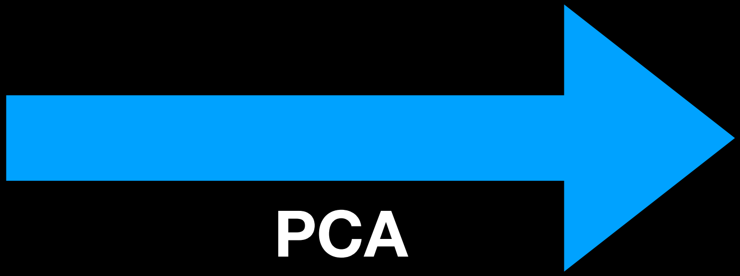
Principle Component Analysis: In Pictures

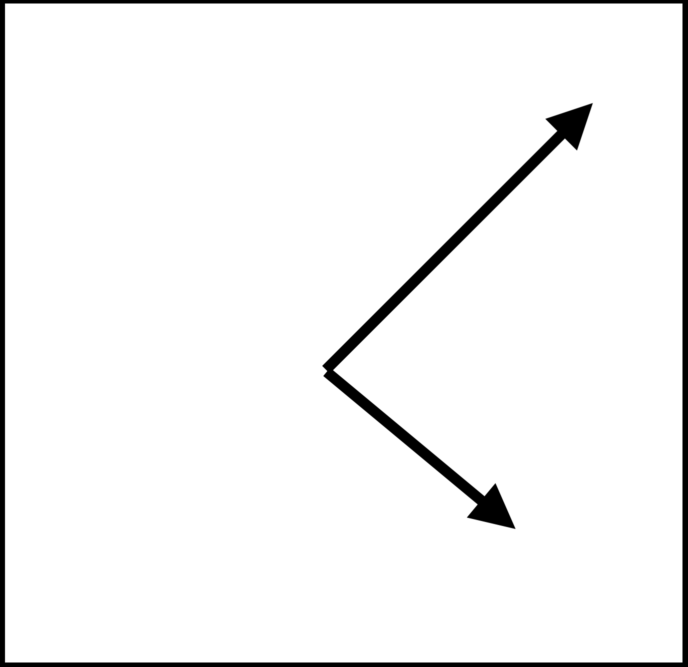


How many vectors do I need to define a 2D space?

Principle Component Analysis: In Pictures



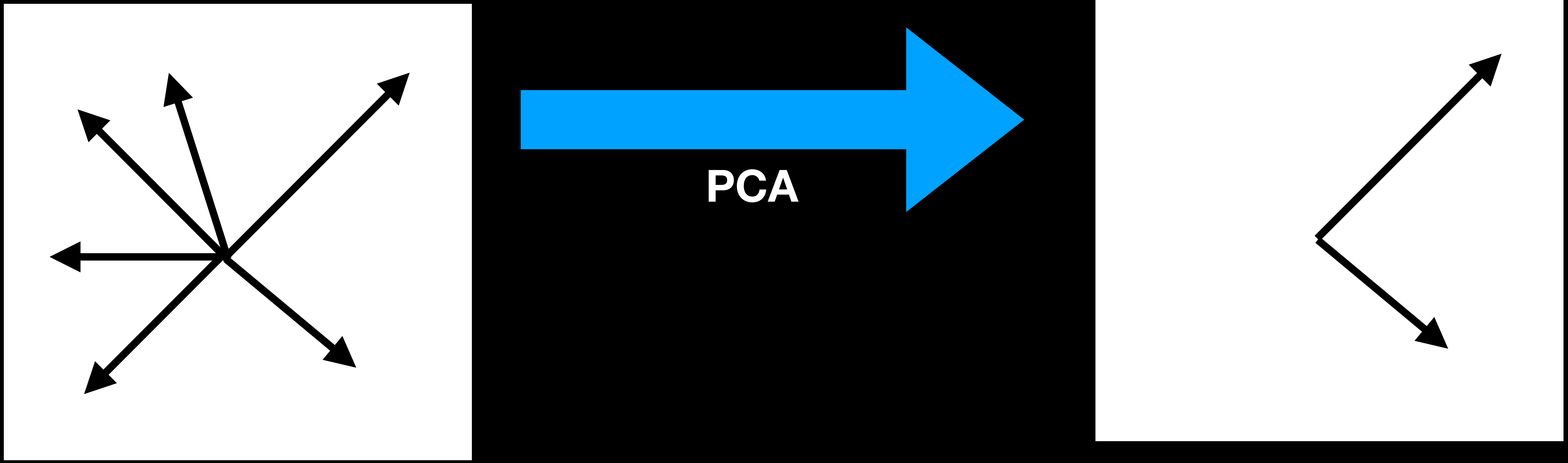




How many vectors do I need to define a 2D space?

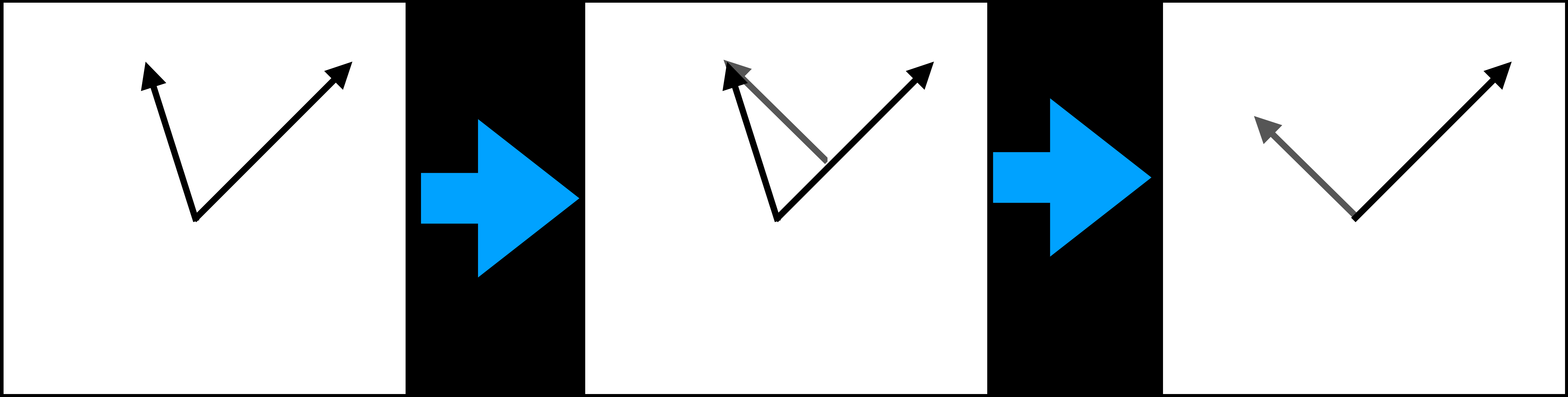
Minimum number of vectors to define a space in a certain number of dimensions

Principle Component Analysis: In Pictures



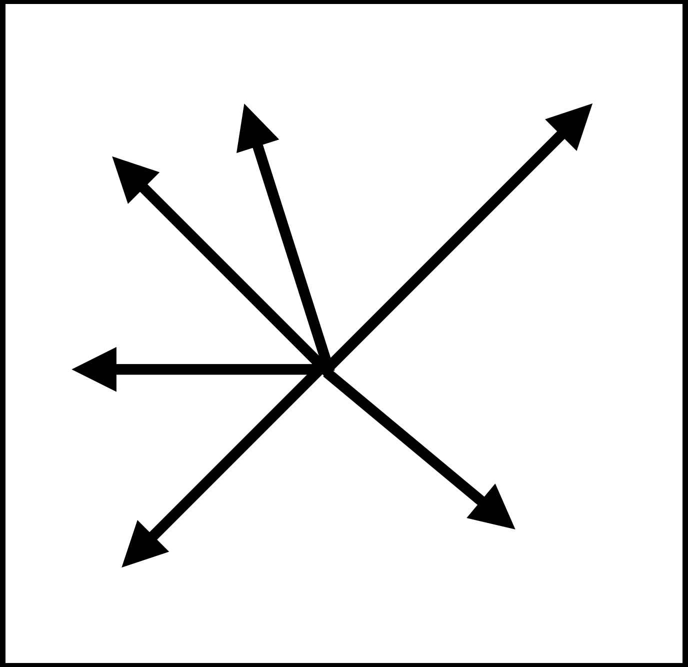
How many vectors do I need to define a 2D space?

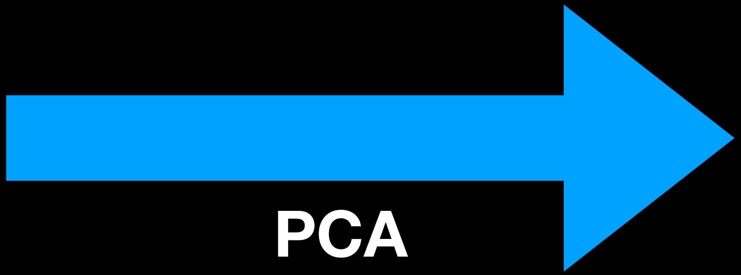
Minimum number of vectors to define a space in a certain number of dimensions

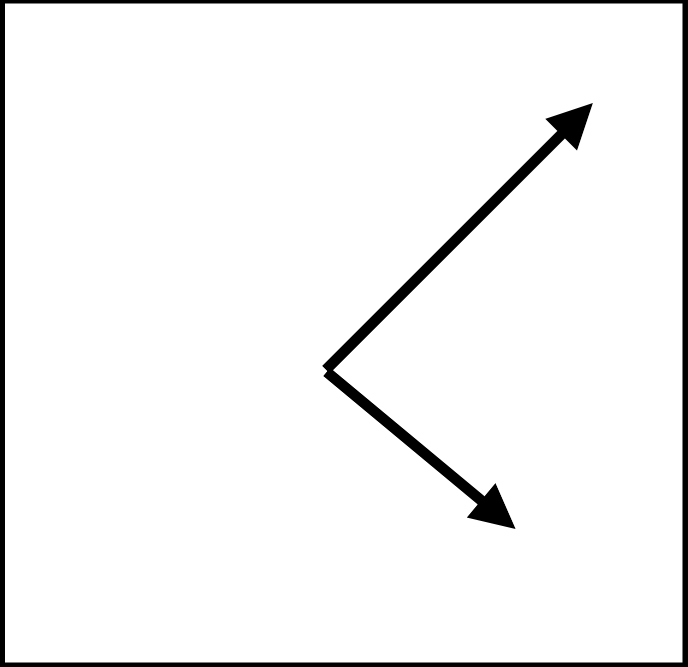


Constructing orthogonal vectors

Principle Component Analysis: In Pictures



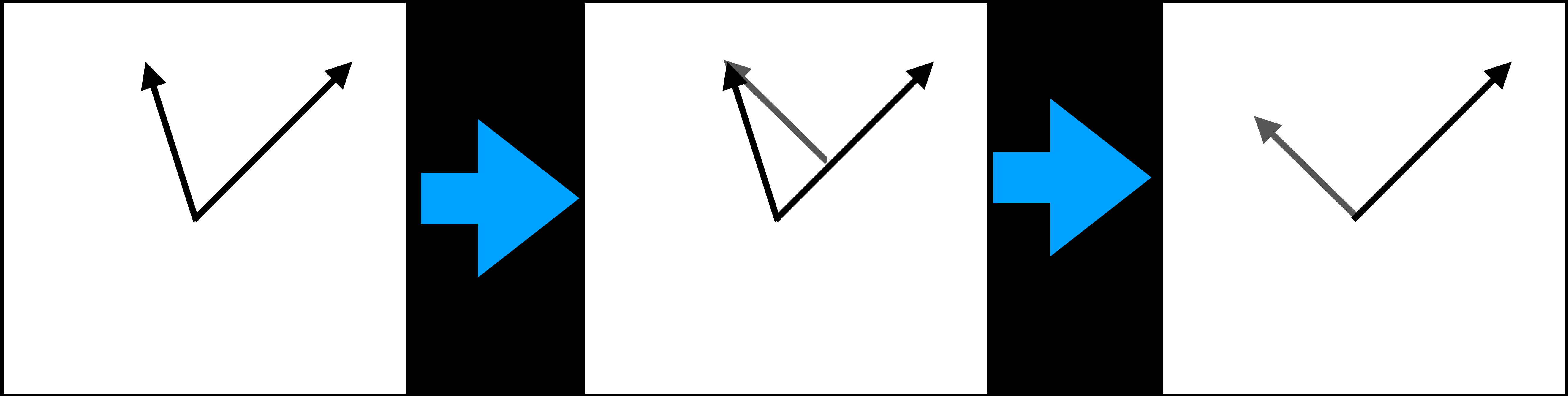




How many vectors do I need to define a 2D space?

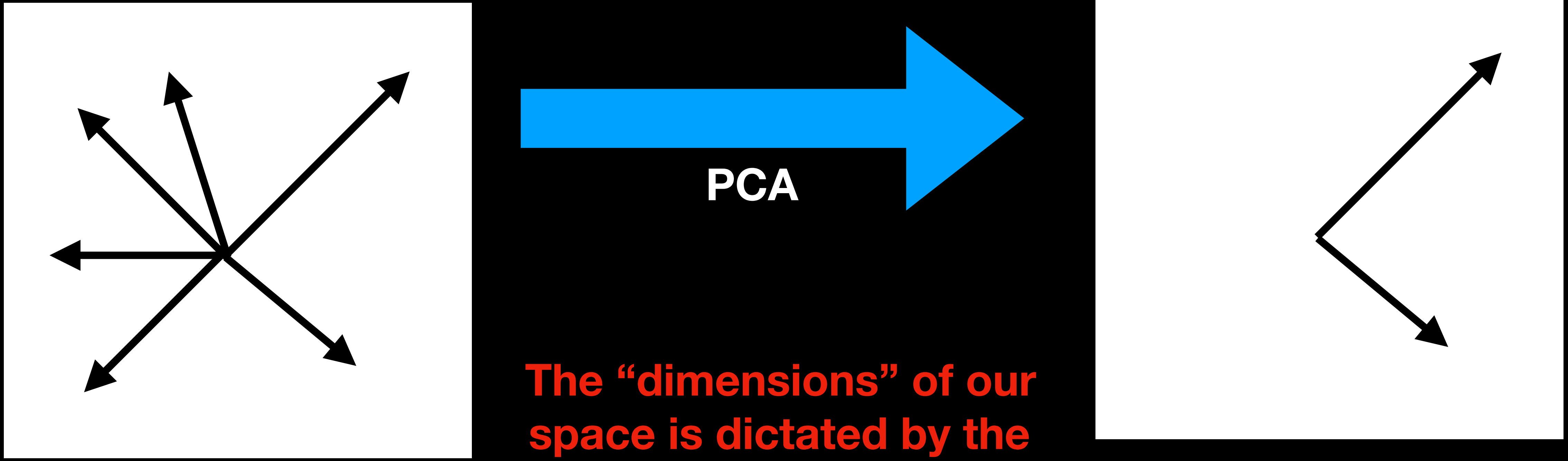
The “dimensions” of our space is dictated by the number of parameters we have

Minimum number of vectors to define a space in a certain number of dimensions



Constructing orthogonal vectors

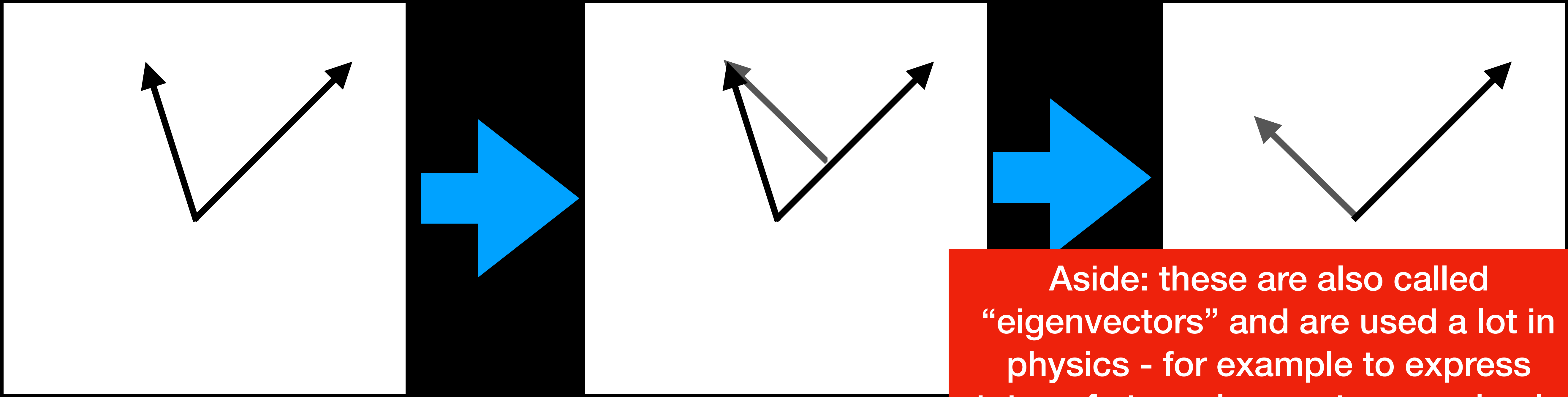
Principle Component Analysis: In Pictures



How many vectors do I need to define a 2D space?

The "dimensions" of our space is dictated by the number of parameters we have

Minimum number of vectors to define a space in a certain number of dimensions



Aside: these are also called "eigenvectors" and are used a lot in physics - for example to express states of atoms in quantum mechanics

Constructing orthogonal vectors

Principle Component Analysis

Can also think about PCA in terms of variance:

Principle Component Analysis

Can also think about PCA in terms of variance:

- The first principal component is that (normalized) linear combination of the variables with the largest variance.

Principle Component Analysis

Can also think about PCA in terms of variance:

- The first principal component is that (normalized) linear combination of the variables with the largest variance.
- The second principal component has largest variance, subject to being uncorrelated with the first.
- And so on.

Principle Component Analysis

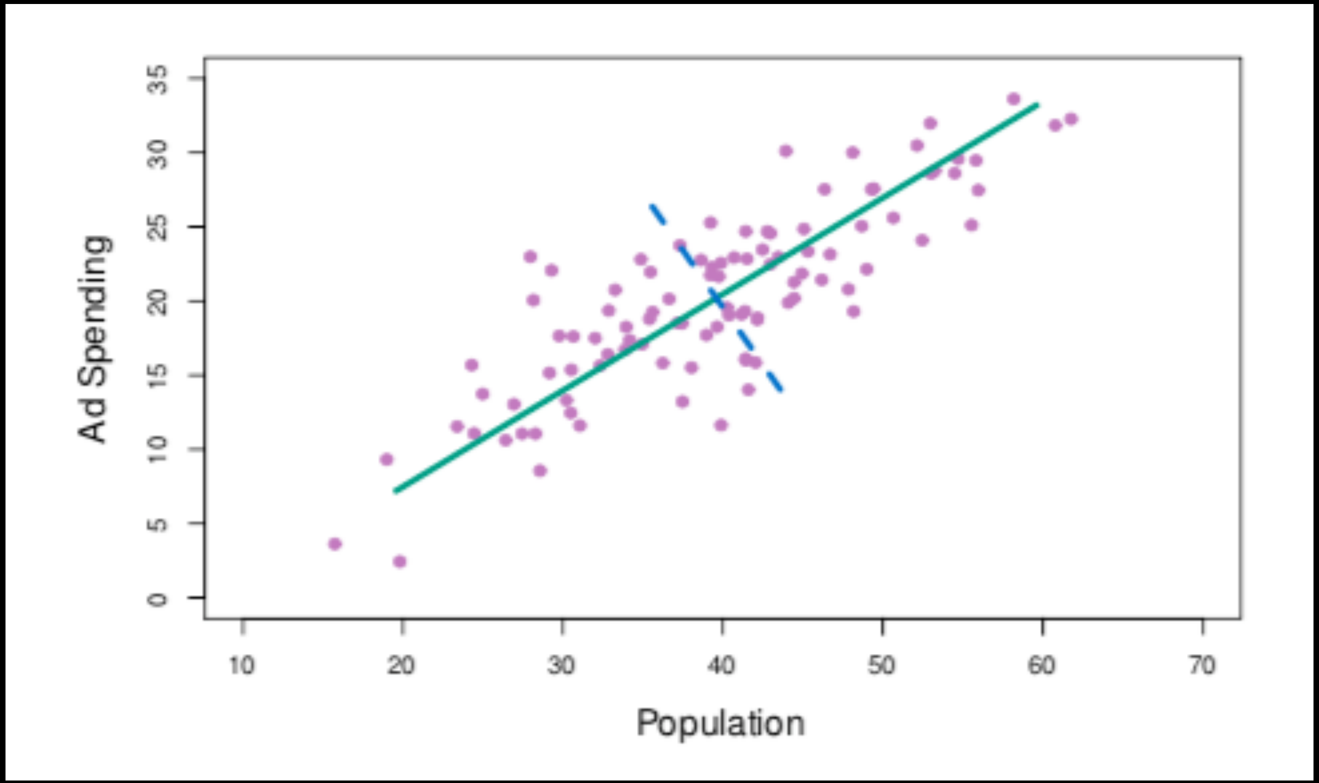
Can also think about PCA in terms of variance:

- The first principal component is that (normalized) linear combination of the variables with the largest variance.
- The second principal component has largest variance, subject to being uncorrelated with the first.
- And so on.
- Hence with many correlated original variables, we replace them with a small set of principal components that capture their joint variation.

Principle Component Analysis

Can also think about PCA in terms of variance:

- The first principal component is that (normalized) linear combination of the variables with the largest variance.
- The second principal component has largest variance, subject to being uncorrelated with the first.
- And so on.
- Hence with many correlated original variables, we replace them with a small set of principal components that capture their joint variation.

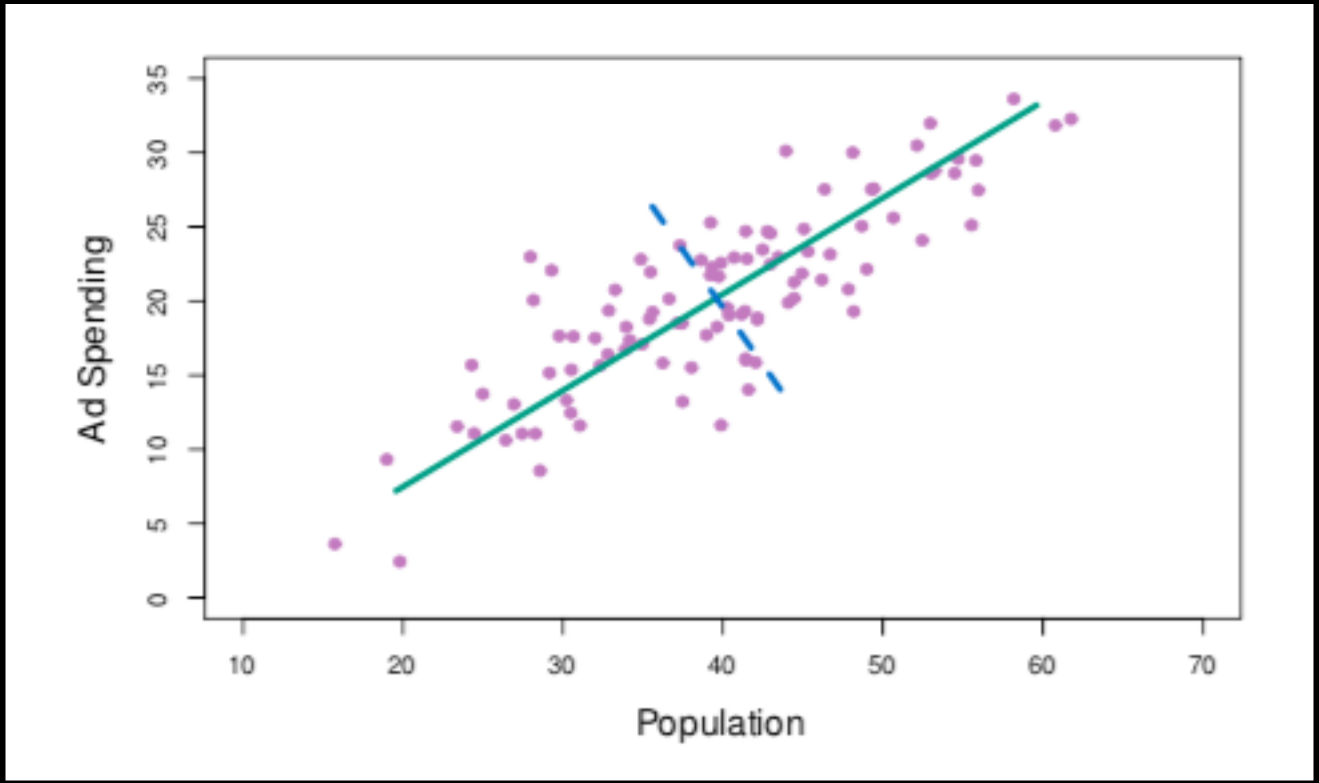


We've essentially already fitted the first PC by fitting lines!

Principle Component Analysis

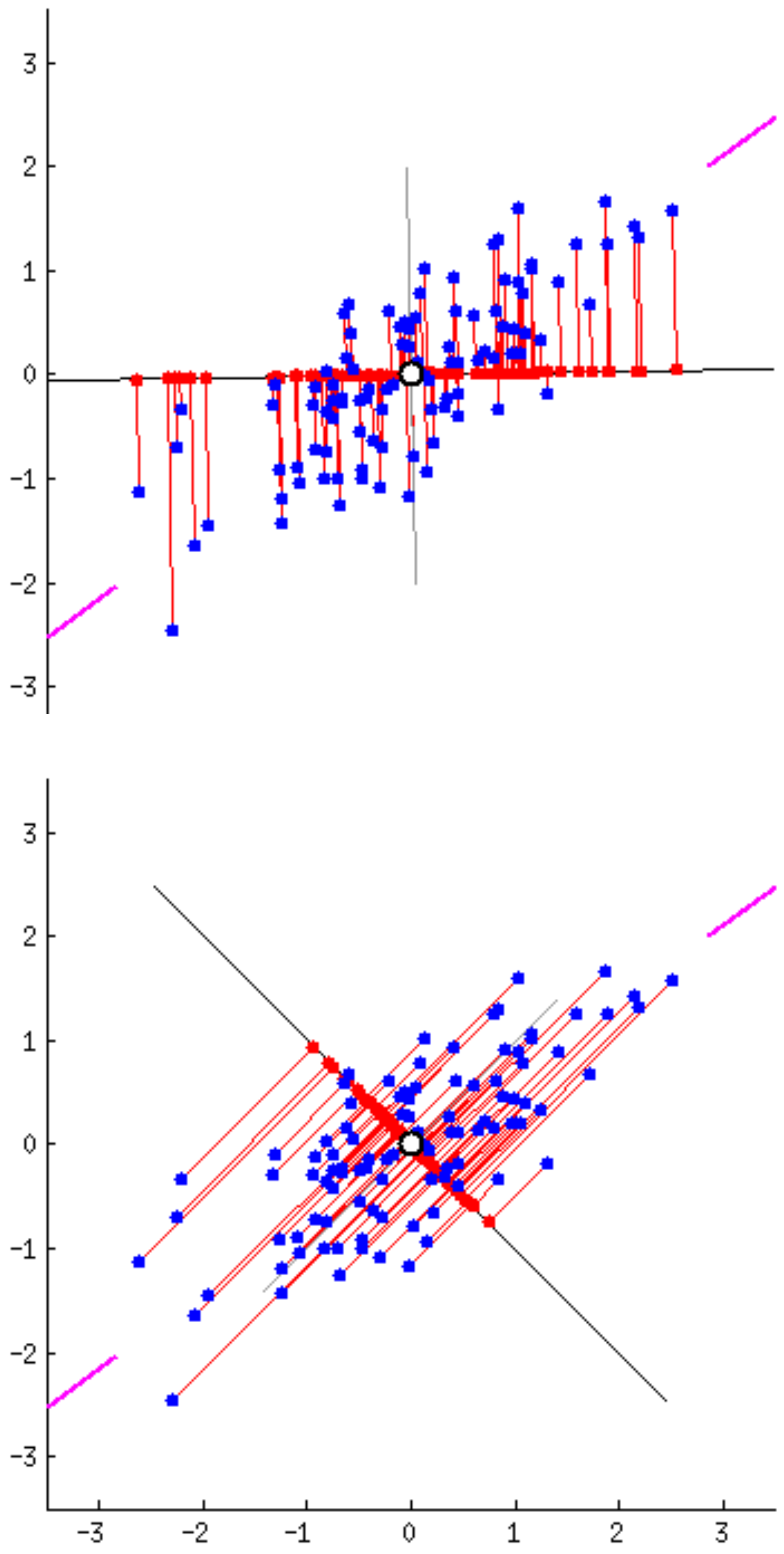
Can also think about PCA in terms of variance:

- The first principal component is that (normalized) linear combination of the variables with the largest variance.
- The second principal component has largest variance, subject to being uncorrelated with the first.
- And so on.
- Hence with many correlated original variables, we replace them with a small set of principal components that capture their joint variation.



We've essentially already fitted the first PC by fitting lines!

Quick R example!


3
2
1
0
-1
-2
-3
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3