

**Welcome to Week 16!**

**You made it!!**

**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:

1. Set the hypotheses.

For a single proportion this will look like:

$H_0: p = \text{null value}$

$H_A: p < \text{or } > \text{ or } \neq \text{ null value}$

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  $< \alpha$ , reject  $H_0$ ,  
there is sufficient evidence for  $[H_A]$
- If p-value  $> \alpha$ , do not reject  $H_0$ ,  
there is not sufficient for evidence for  $[H_A]$

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

English



Math



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:

$H_0: p = \text{null value}$

$H_A: p < \text{or } > \text{ or } \neq \text{ null value}$

2. Check assumptions and conditions

3. Calculate a test statistic and a p-value

4. Make a decision, and interpret it in context

- If  $p\text{-value} < \alpha$ , reject  $H_0$ ,  
there is sufficient evidence for  $[H_A]$
- If  $p\text{-value} > \alpha$ , 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

## Test Statistics

(a) Z-score  $\rightarrow P(Z)$

(b) T-Score  $\rightarrow P(T)$

(c)  $\chi^2 \rightarrow P(\chi^2)$

# Anatomy of a test statistic

The general form of a test statistic is

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

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

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

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.

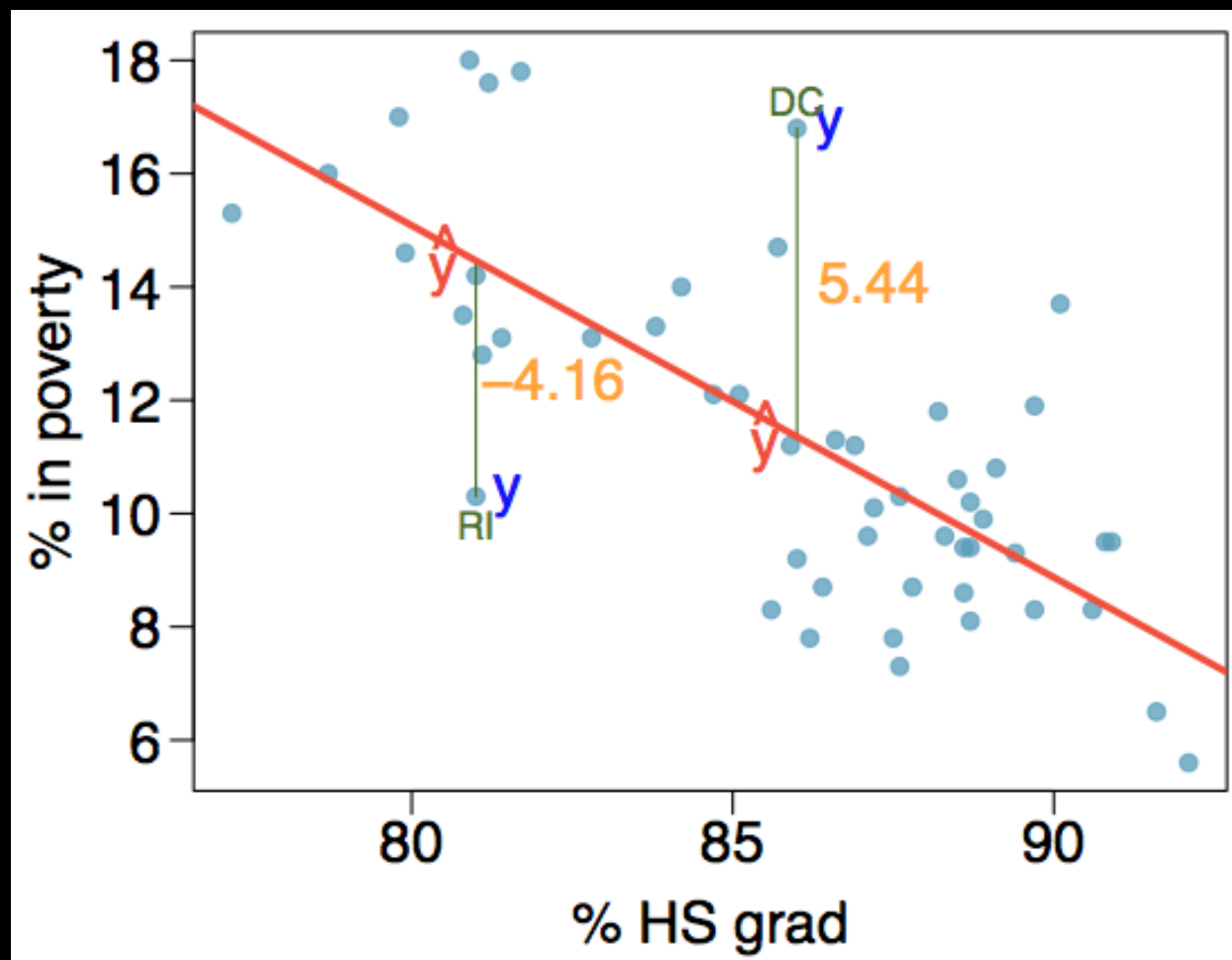
We (almost) never know if  $H_0$  or  $H_A$  is true, but we need to consider all possibilities.

# Residuals

**Residuals** are the leftovers from the model fit:  $\text{Data} = \text{Fit} + \text{Residual}$

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  $\neq 0$



# 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  $\eta$  to  $p$ . There are a variety of options but the most commonly used is the logit function.

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

# Logistic Regression: A Morbid Example

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

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

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

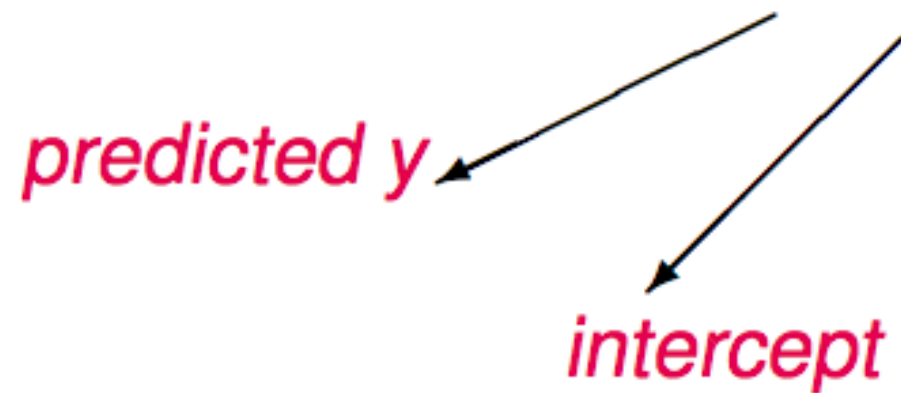
$$\text{logit}(p) = \eta$$

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...

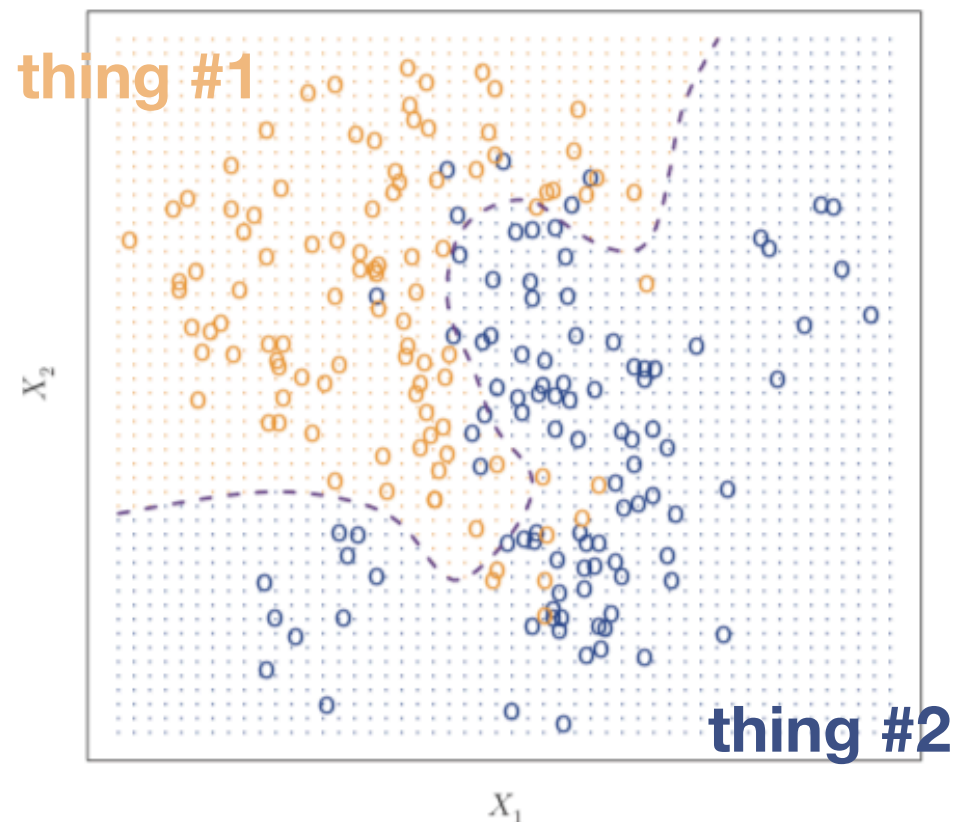
$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_n x_n$$



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)

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

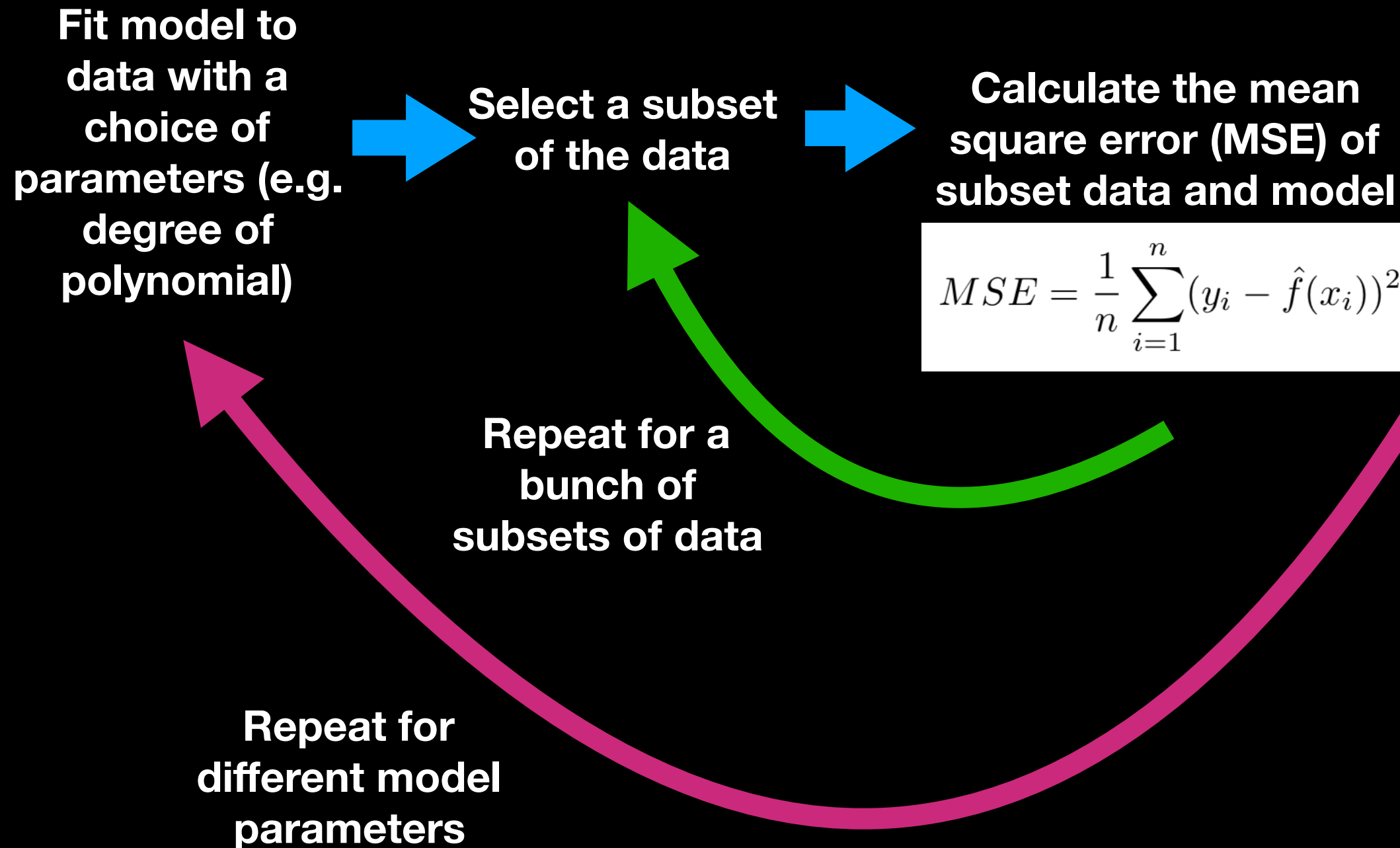
**mean square error** if we kept estimating response variable  $y$  by our fitted function of our explanatory variables,  $f(x)$  with different sample datasets at point  $x_0$

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

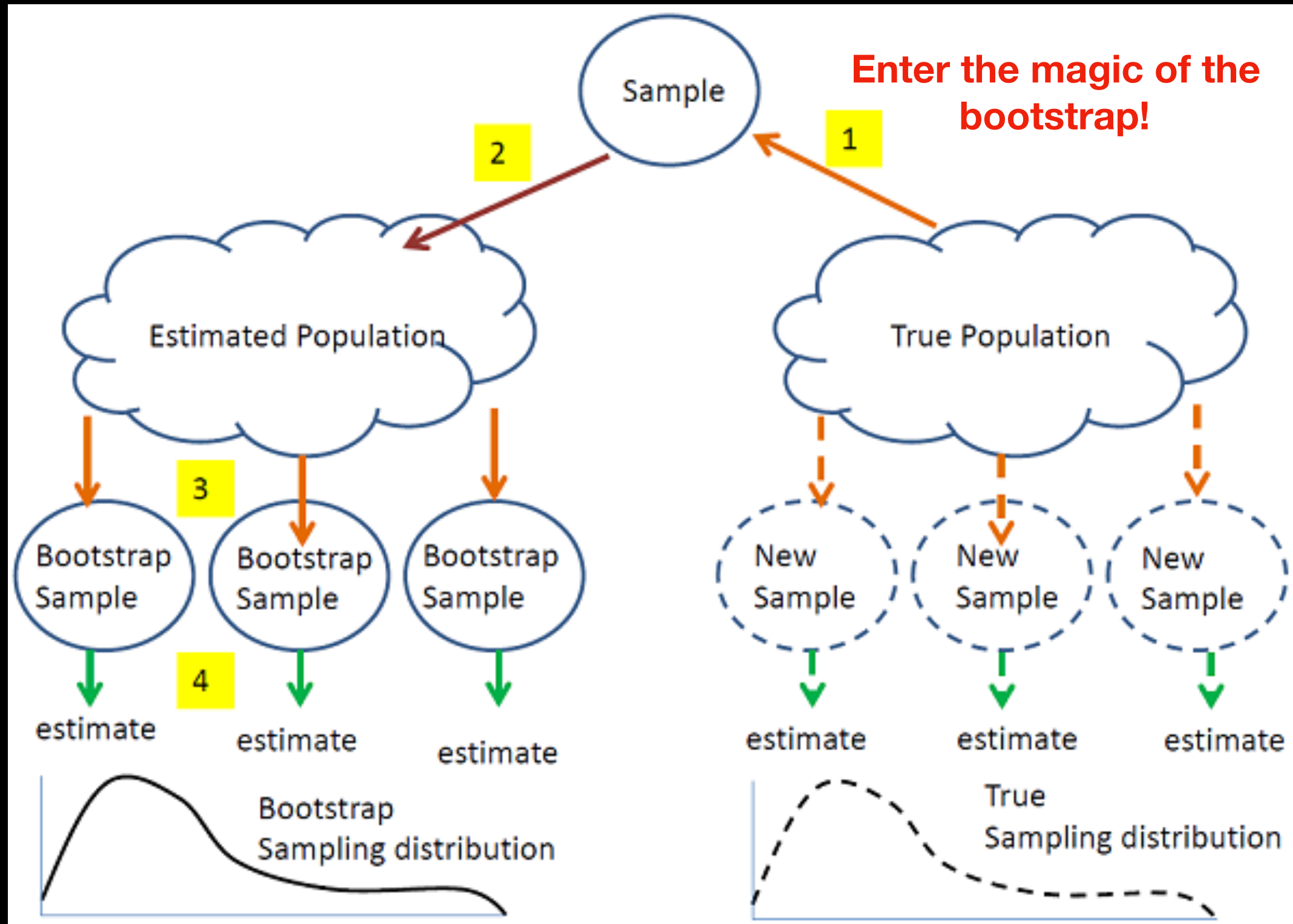
Inherent error (**bias**) in the fact that any model is only an approximation to reality

Inherent error in our measurements

# Cross-Validation Methods



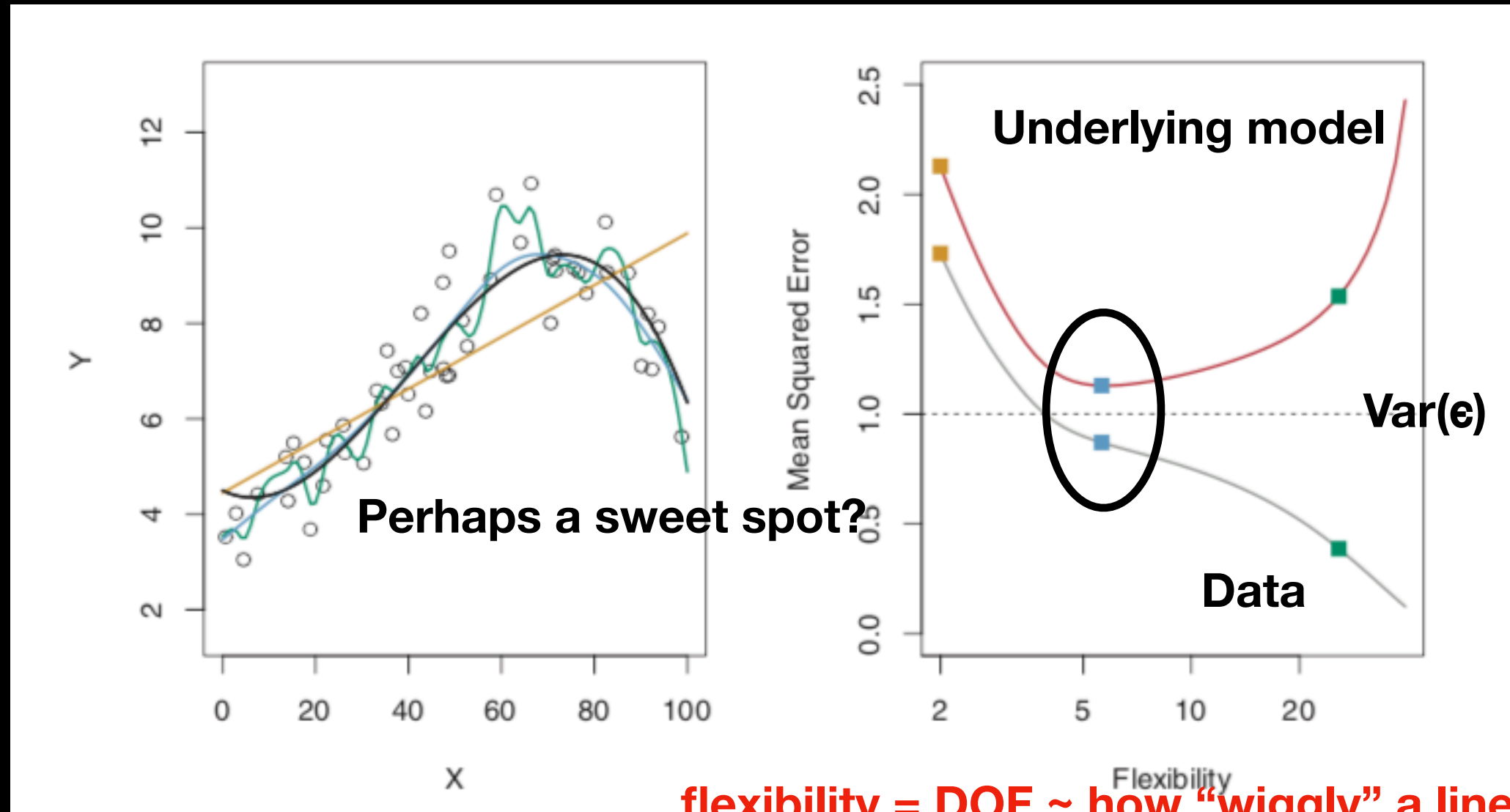
# Bootstrapping



**Distribution of means, proportions, etc**

# Bias-Variance Trade-Off (First Glance)

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



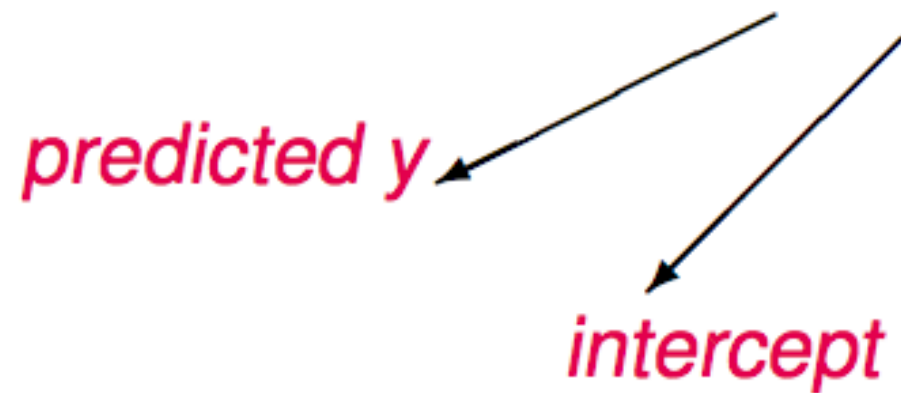
- Actual underlying function -  $y$
- o Simulated data with added error ( $\epsilon$ )
- Linear fit
- Low “flexibility” smooth spline
- High “flexibility” smooth spline

fits data well, but underlying model badly



# So far...

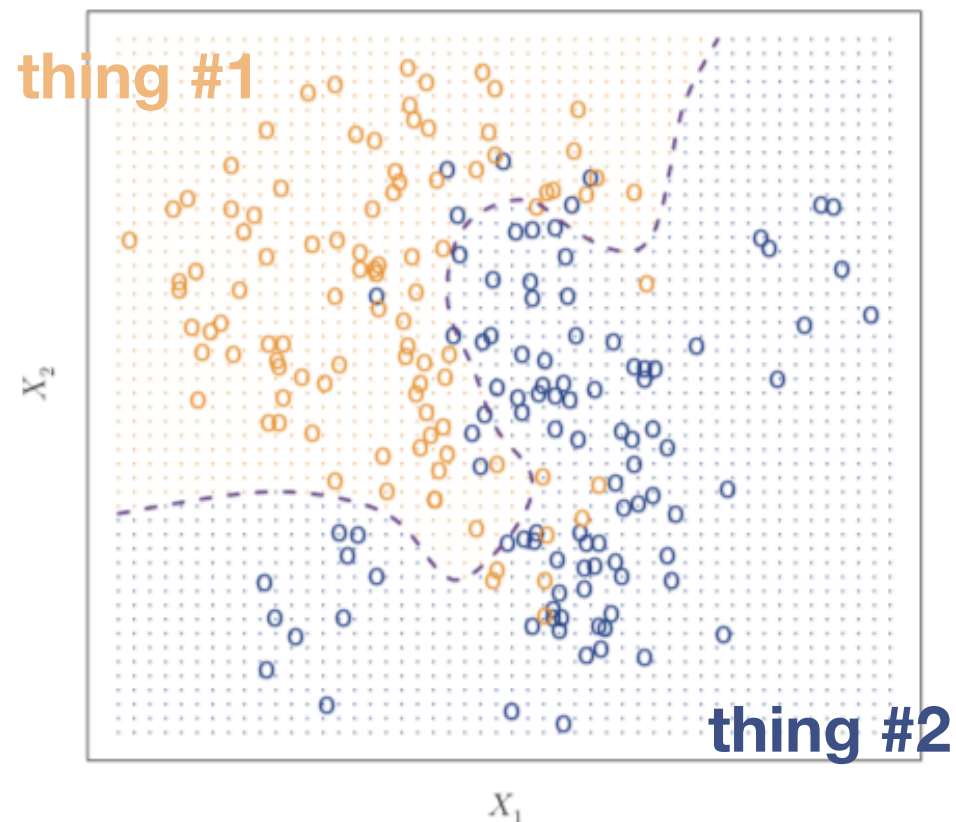
$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_n x_n$$



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...

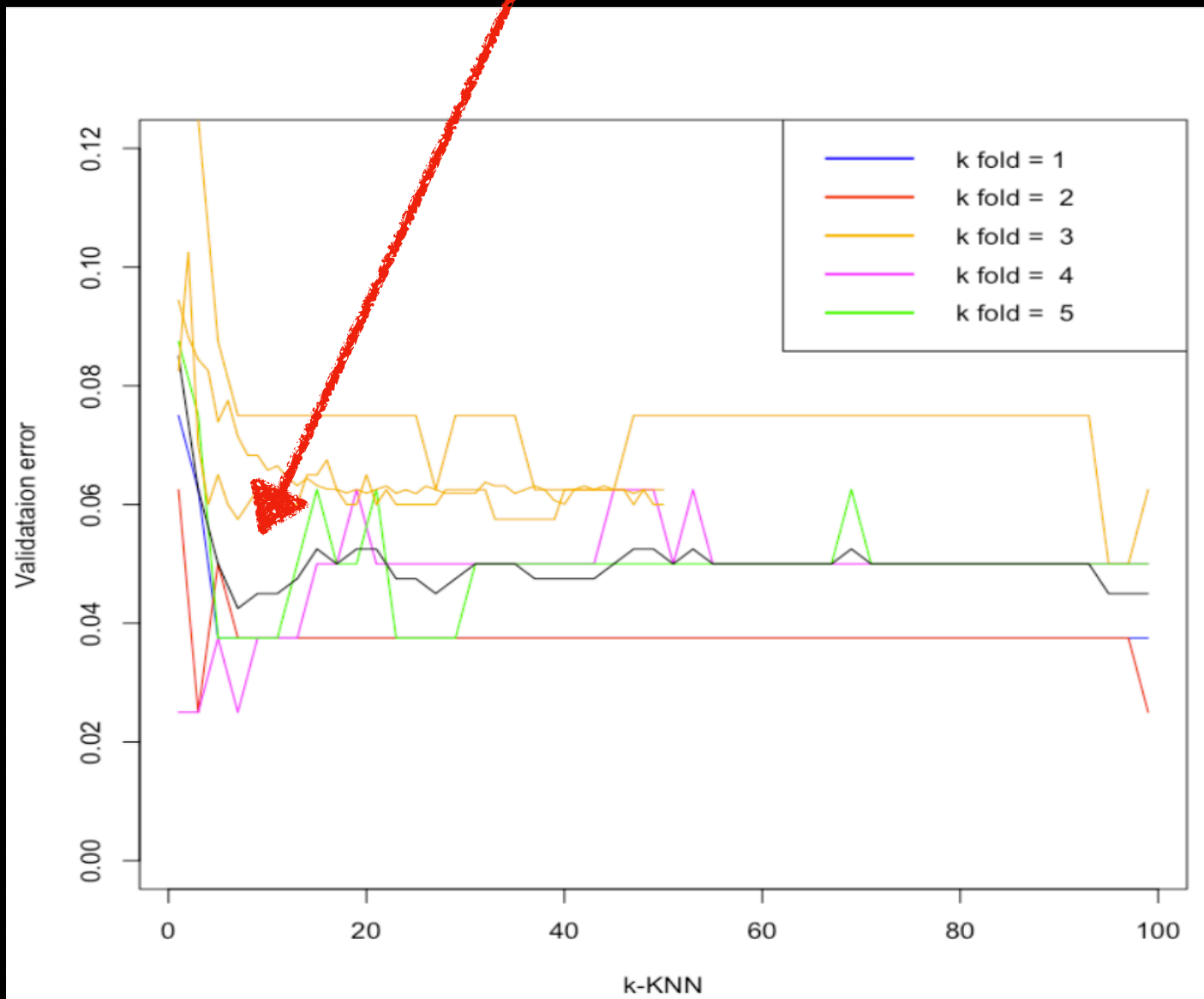


“I want to know where thing #1 and thing #2 live in some 2D space - computer, go figure out the boundary between these two things and let me know”

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  
(Ch. 5)

Regularization  
(Ch. 6)

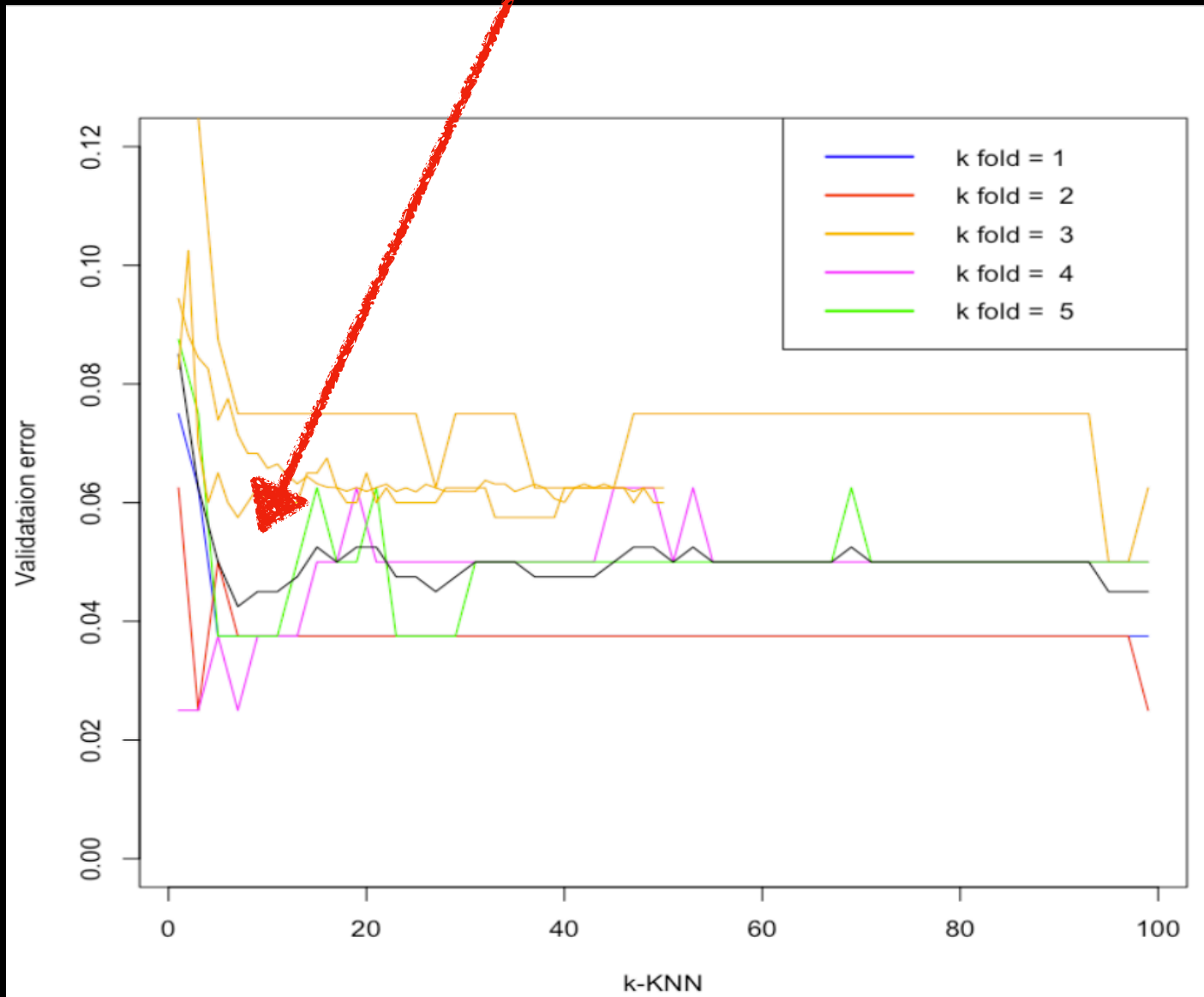
Test fits with  
subsets of data

Use math to  
select parameters

Best model parameters

# 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.

Last Week

Cross-Validation  
(Ch. 5)

Regularization  
(Ch. 6)

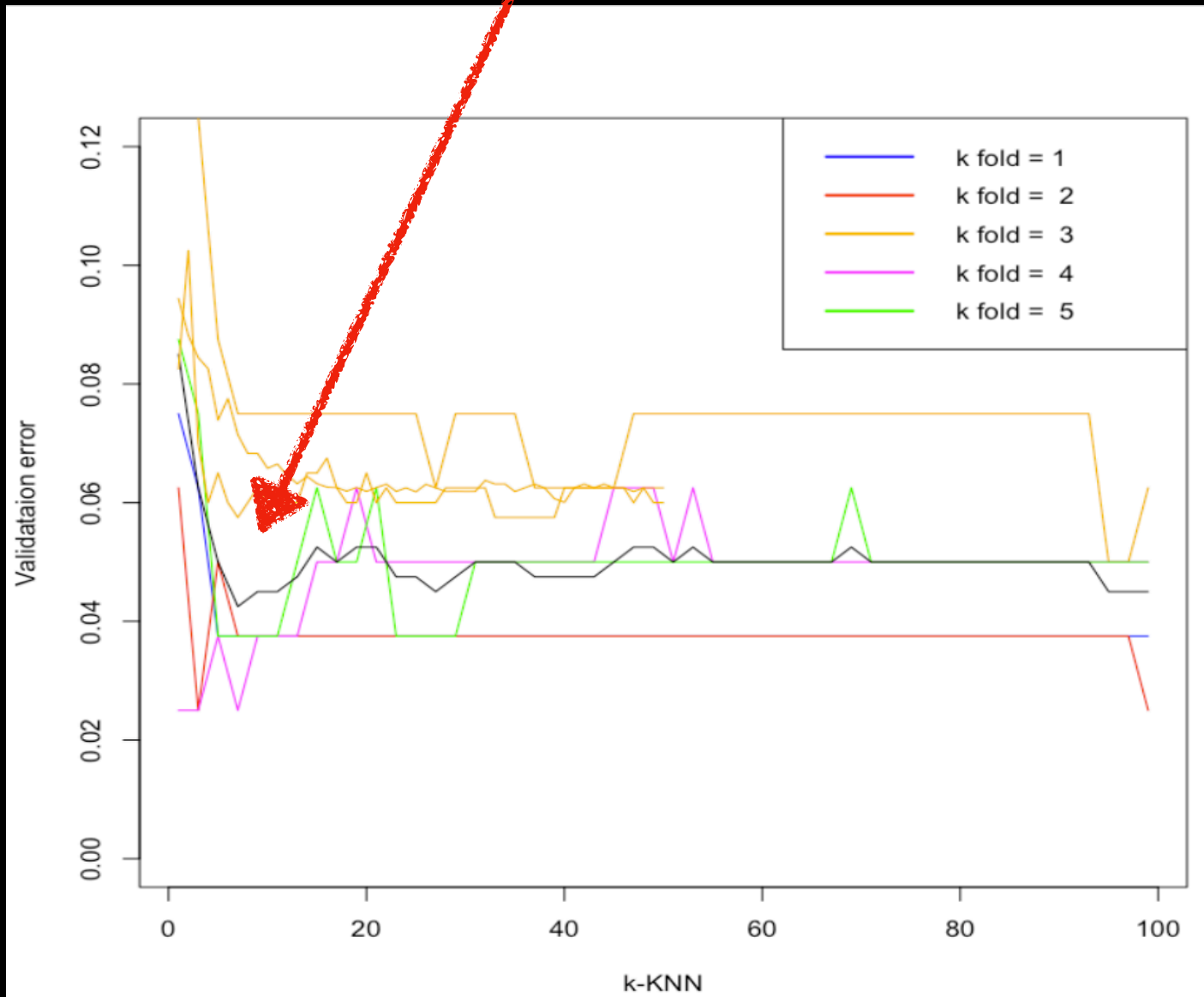
Test fits with  
subsets of data

Use math to  
select parameters

Best model parameters

# 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  
(Ch. 5)

Test fits with  
subsets of data

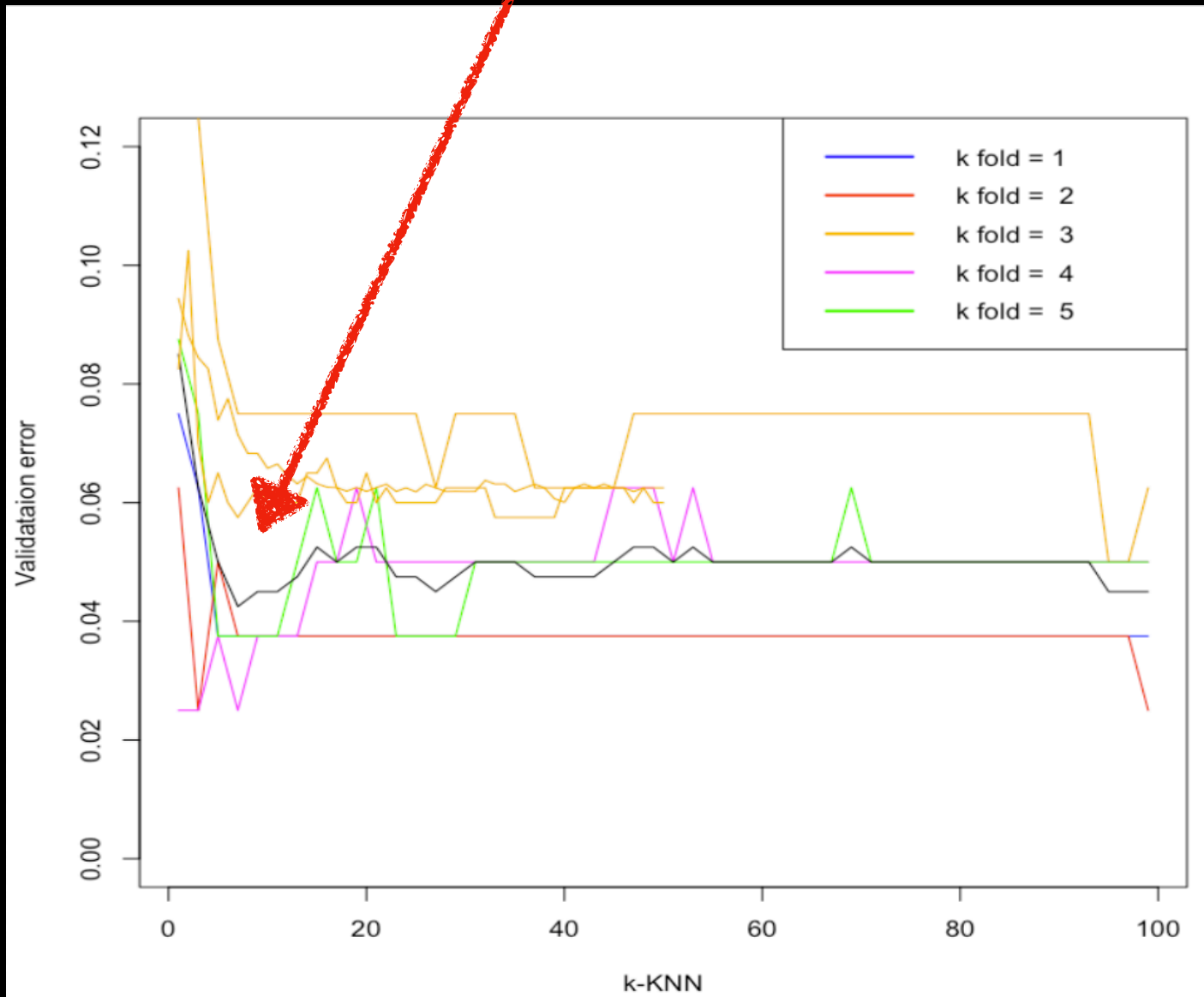
Best model parameters

Regularization  
(Ch. 6)

Use math to  
select parameters

# 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  
(Ch. 5)

Regularization  
(Ch. 6)

Test fits with  
subsets of data

Use math to  
select parameters

Best model parameters

# Improvements on the Linear Model

Linear model is a good approximation of the relationship between the input and output variables, but it is not always the best choice.

There are several reasons why a linear model might be inadequate:

- The relationship between the input and output variables is non-linear.

- The data is noisy or contains outliers.

- The model is overfitted to the training data.

There are several ways to improve the linear model:

- Use a non-linear model, such as a polynomial or neural network.

- Use regularization to prevent overfitting.

# 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  $k = 1, 2, \dots, p$ :
  - (a) Fit all  $\binom{p}{k}$  models that contain exactly  $k$  predictors.
  - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $M_k$ . Here **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  $k = 1, 2, \dots, p$ :
  - (a) Fit all  $\binom{p}{k}$  models that contain exactly  $k$  predictors.
  - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $M_k$ . Here **best** is defined as having the smallest RSS, or equivalently largest  $R^2$ .

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

# 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  $k = 1, 2, \dots, p$ :
  - (a) Fit all  $\binom{p}{k}$  models that contain exactly  $k$  predictors.
  - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $M_k$ . Here **best** is defined as having the smallest RSS, or equivalently largest  $R^2$ .
3. Select a single best model from among  $M_0, \dots, M_p$  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  $k = 1, 2, \dots, p$ :
  - (a) Fit all  $\binom{p}{k}$  models that contain exactly  $k$  predictors.
  - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $M_k$ . Here **best** is defined as having the smallest RSS, or equivalently largest  $R^2$ .
3. Select a single best model from among  $M_0, \dots, M_p$  using cross-validated prediction error,  $C_p$  (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  $k = 1, 2, \dots, p$ :
  - (a) Fit all  $\binom{p}{k}$  models that contain  $k$  predictors. 

this number can get big  $\sim 2^p$   
 $p=2$ : 4 (including null)  
 $p=3$ : 8  
...  
 $p=10$ : 1024
  - (b) Pick the best among these  $\binom{p}{k}$  models, and call it  $M_k$ . Here **best** is defined as having the smallest RSS, or equivalently largest  $R^2$ .
3. Select a single best model from among  $M_0, \dots, M_p$  using cross-validated prediction error,  $C_p$  (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)

Forward selection: start with an empty model and add variables one by one until no more variables improve the model.

Backward selection: start with a full model and remove variables one by one until no more variables can be removed without worsening the model.

Stepwise selection: a combination of forward and backward selection, allowing for both adding and removing variables.

Criteria for variable selection: Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Adjusted R-squared, etc.

Limitations of subset selection: can lead to overfitting, biased coefficient estimates, and poor predictive performance on new data.

Regularization methods: Ridge regression, Lasso regression, and Elastic Net, which penalize large coefficients to prevent overfitting.

Cross-validation: used to evaluate the performance of subset selection methods and regularization methods.

Model selection: choosing the best model from a set of candidate models based on a specific criterion.

Model comparison: comparing the performance of different models to determine the best one.

Model evaluation: assessing the performance of a model on new data to estimate its predictive accuracy.

Model interpretation: understanding the relationship between the variables in the model and the response variable.

Model validation: verifying the results of the model selection process and ensuring the model's performance is stable.

# 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:

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

# Subset selection: Forward & Backward (Review)

Find models that  
minimize:

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


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

# Subset selection: Forward & Backward (Review)

Find models that  
minimize:

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

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

d predictors, n data points

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

# Subset selection: Forward & Backward (Review)

Find models that minimize:

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

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

$d$  predictors,  $n$  data points

estimate from  $\text{SE}^2/n = \sigma^2$  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:

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

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

d predictors, n data points

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

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

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

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

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

Or: large values as  $R_{\text{adj}}^2$

# Subset selection: Forward & Backward (Review)

Find models that minimize:

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

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

d predictors, n data points

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

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

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

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

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

Or: large values as  $R_{\text{adj}}^2$

## Optional R notes.



# Shrinkage Methods

# Shrinkage Methods

Lets recall how we find a linear model:

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

**Least squares minimization (i.e. minimize RSS)**

# Shrinkage Methods

Lets recall how we find a linear model:

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

Least squares minimization (i.e. minimize RSS)

In contrast, the **Ridge** or **Lasso** regression coefficient estimates  $\beta^R$  and  $\beta^L$  are the values that minimize

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

**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**

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

# Shrinkage Methods

Lets recall how we find a linear model:

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

Least squares minimization (i.e. minimize RSS)

In contrast, the **Ridge** or **Lasso** regression coefficient estimates  $\beta^R$  and  $\beta^L$  are the values that minimize

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

**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**

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

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

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

**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**

# Shrinkage Methods

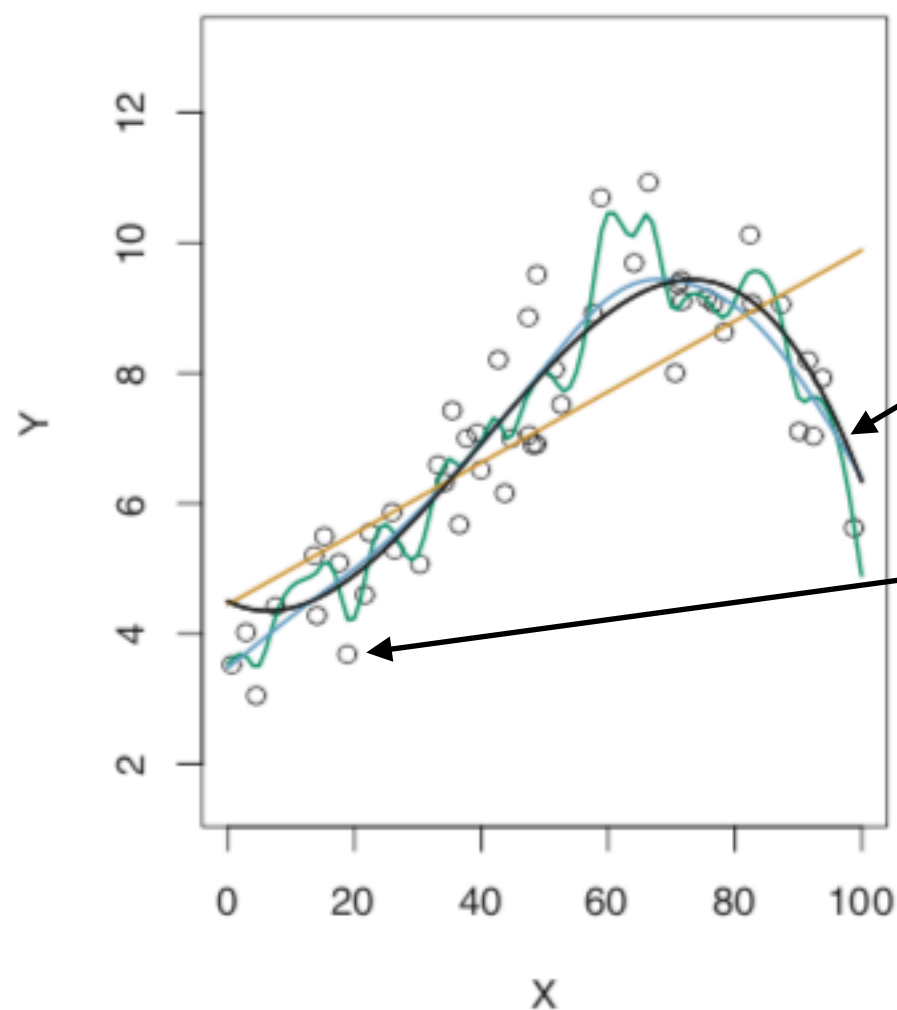
$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2$$

**Ridge**

Why would we make our lives more complicated like this?

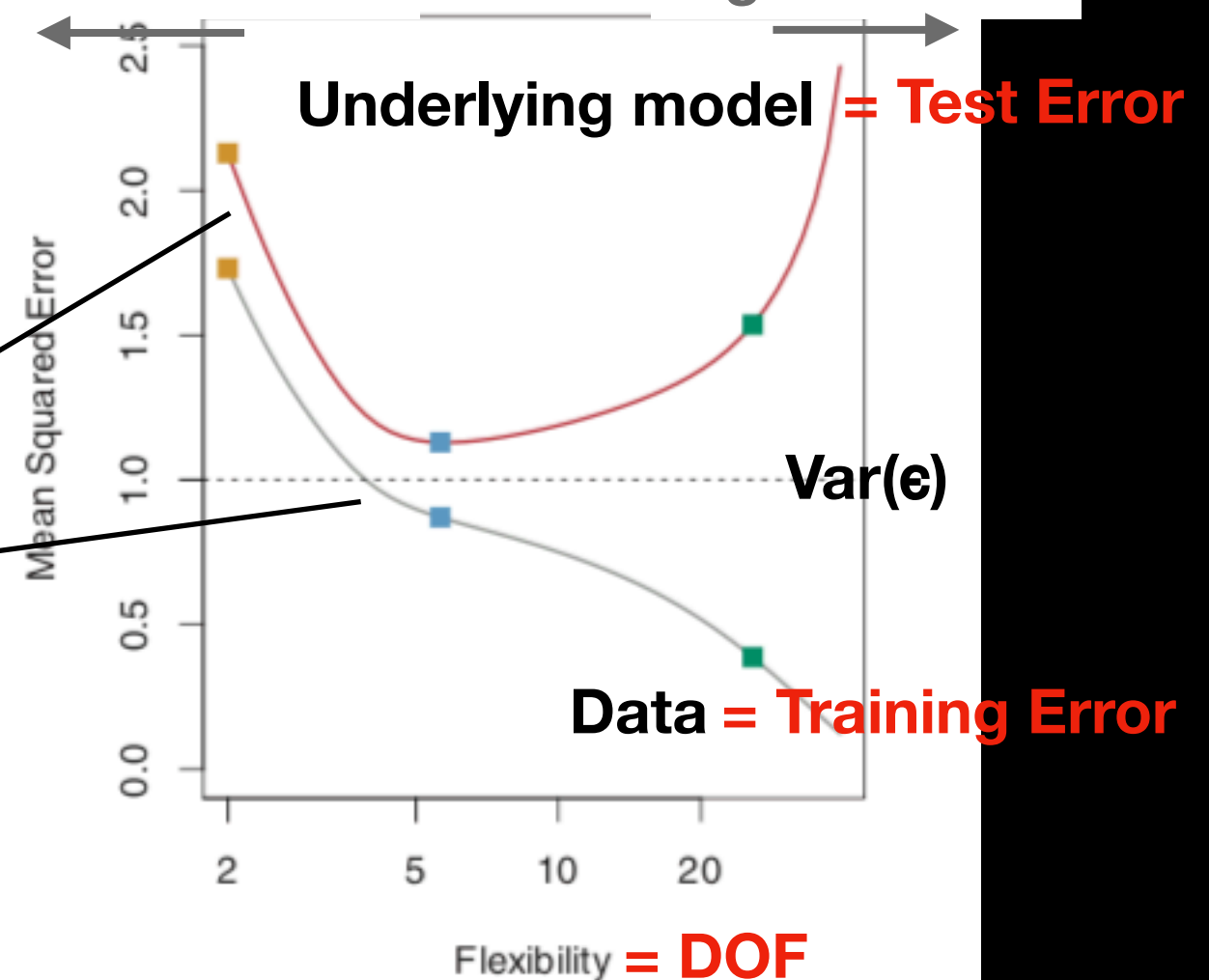
$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^p |\beta_j|$$

**Lasso**



High Bias  
Low Variance

Low Bias  
High Variance



# Shrinkage Methods

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

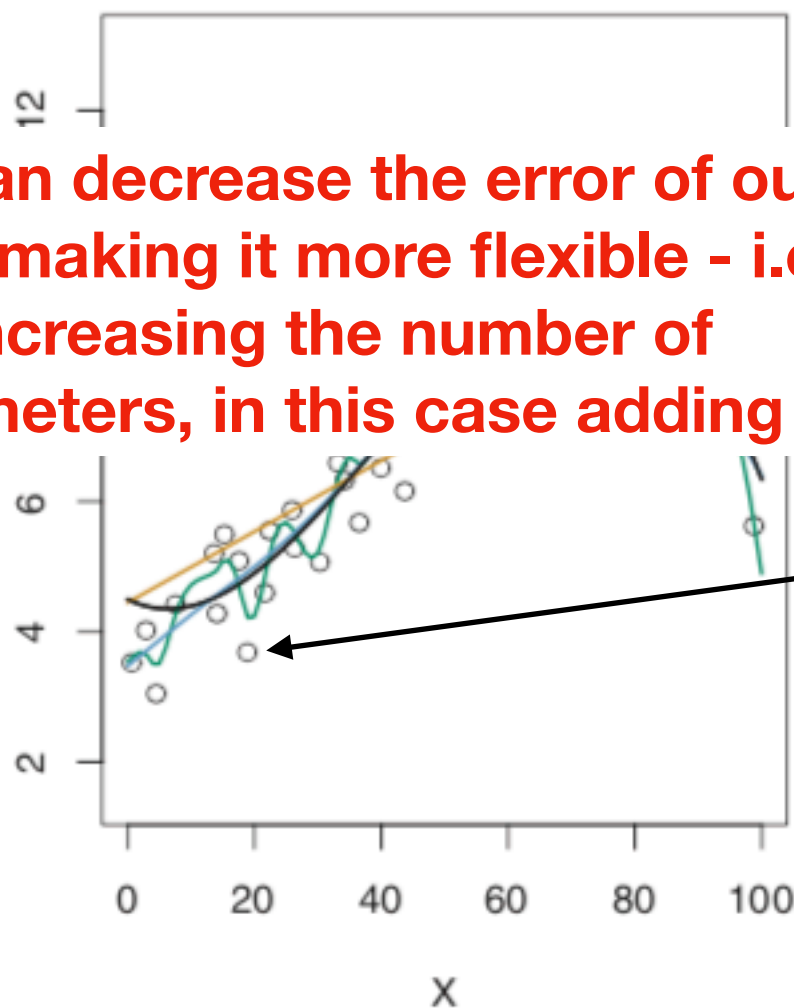
**Ridge**

Why would we make our lives more complicated like this?

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

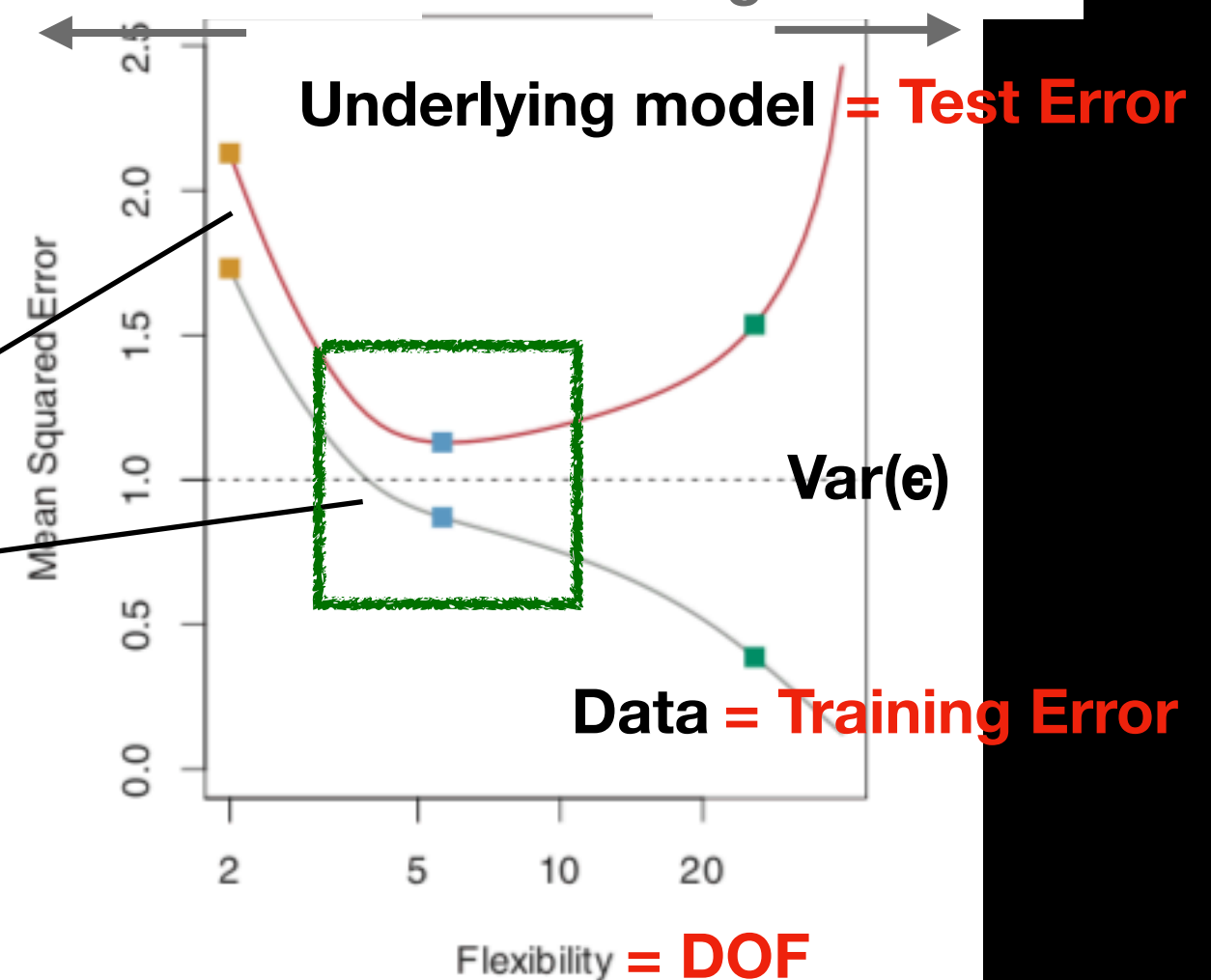
**Lasso**

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



High Bias  
Low Variance

Low Bias  
High Variance





# 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  $\lambda$**

# 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  $\lambda$**
- (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  $\lambda$**
- (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**

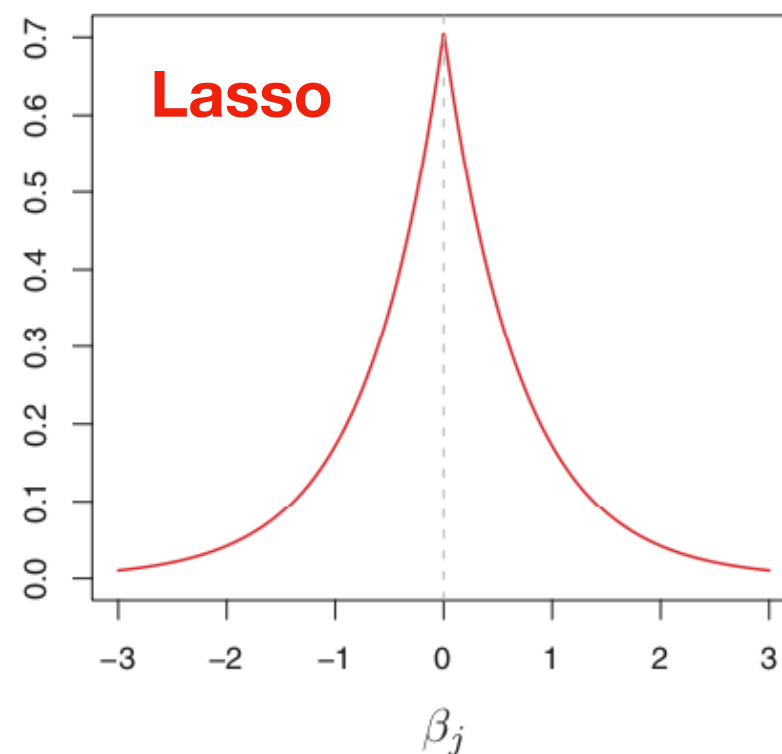
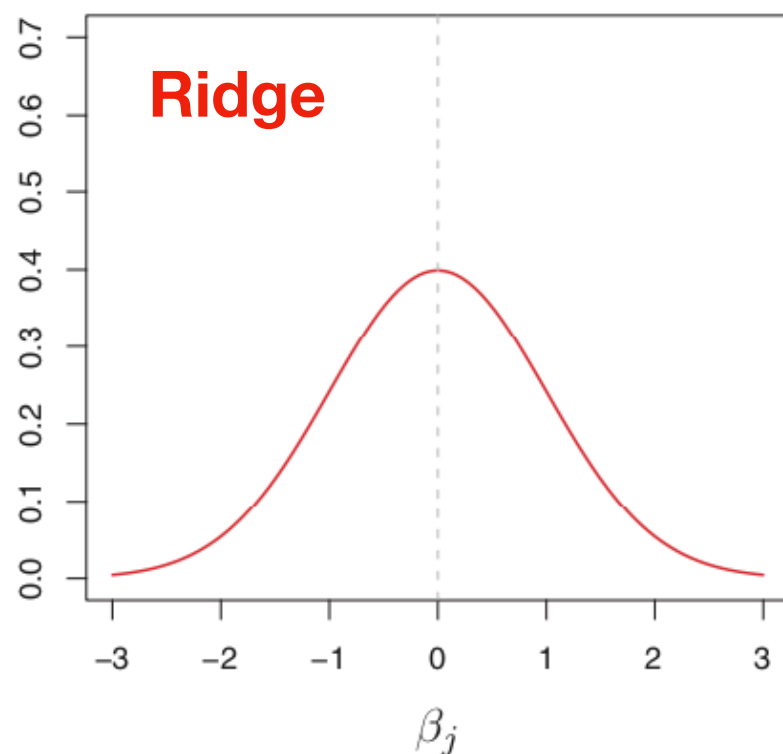
Why would we make our lives more complicated like this?

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

**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  $\lambda$
- (2) Natural way to perform variable selection  “Shrinkage” of less important parameters to zero

Assumption about how probable a value of  $\beta_j$  is



# 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**

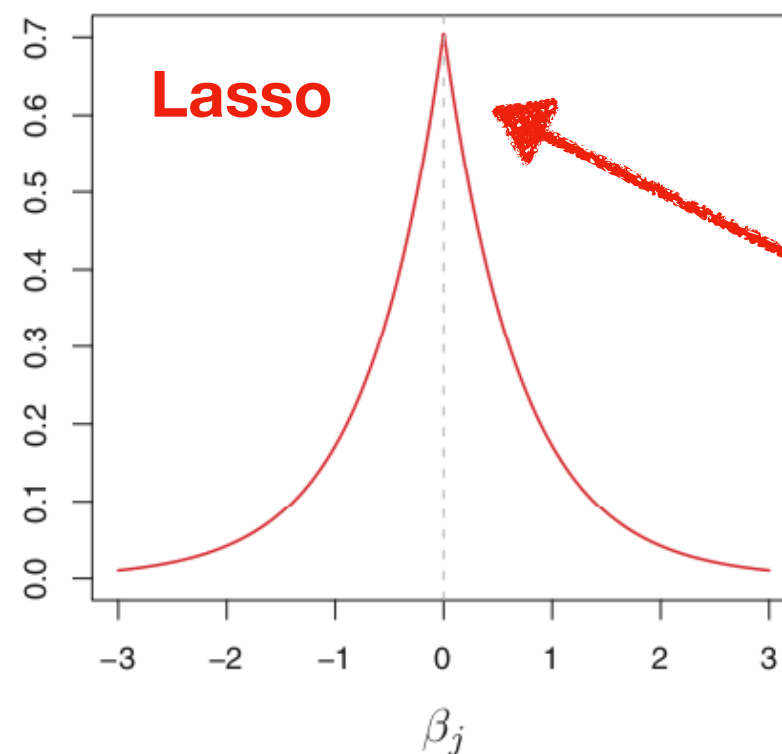
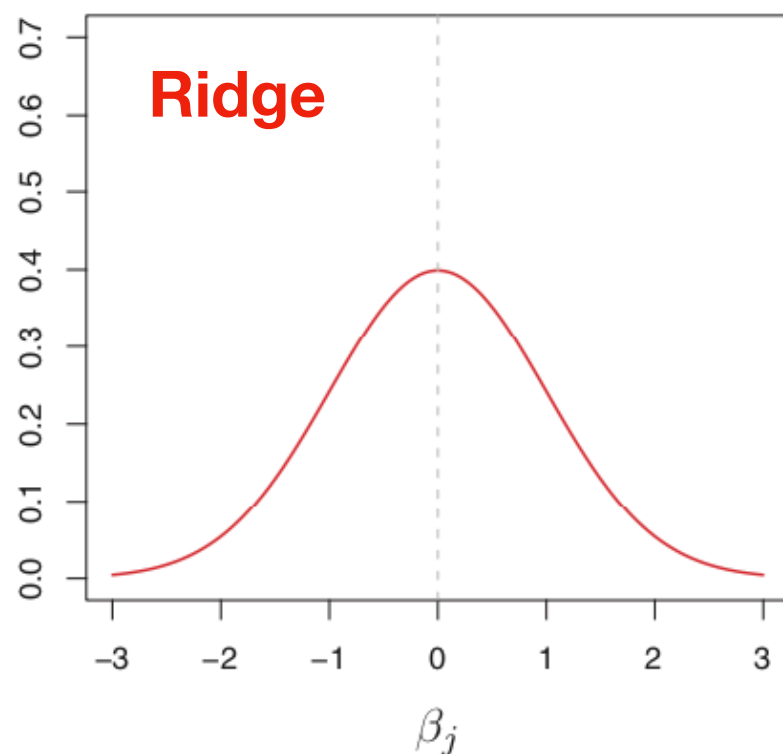
Why would we make our lives more complicated like this?

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

**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  $\lambda$
- (2) Natural way to perform variable selection  “Shrinkage” of less important parameters to zero

Assumption about how probable a value of  $\beta_j$  is



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**

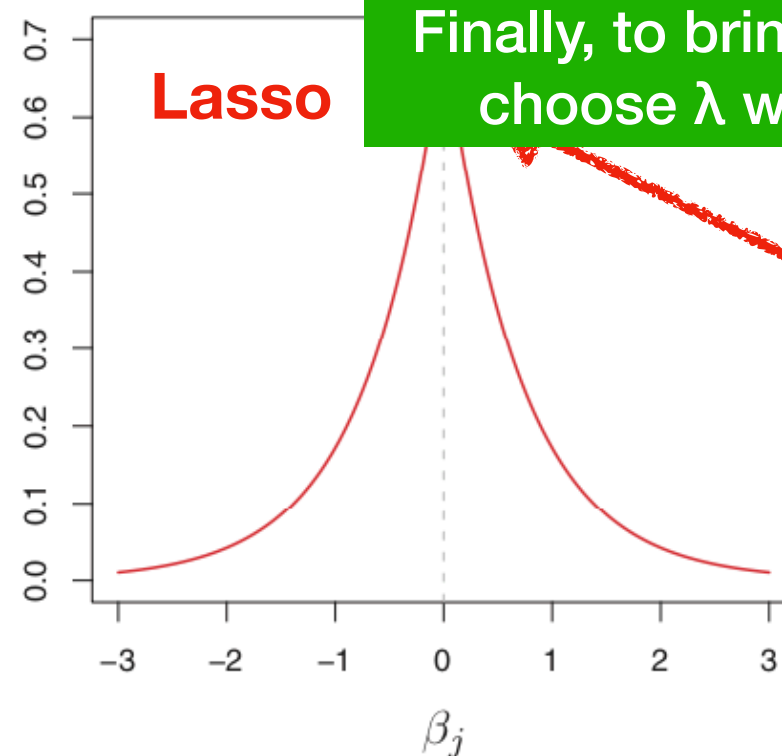
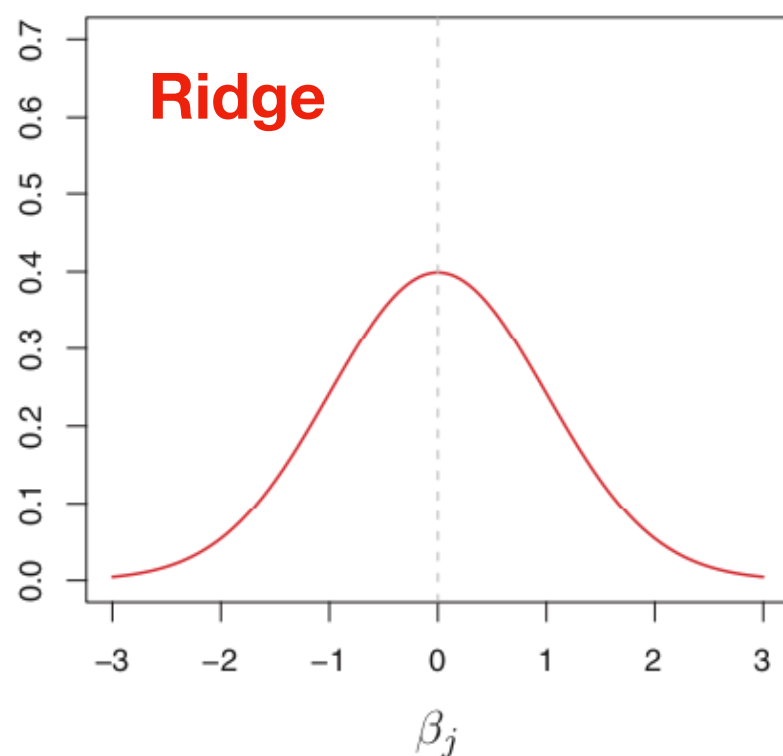
Why would we make our lives more complicated like this?

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

**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  $\lambda$
- (2) Natural way to perform variable selection  $\rightarrow$  “Shrinkage” of less important parameters to zero

Assumption about how probable a value of  $\beta_j$  is



Finally, to bring it full circle - we can choose  $\lambda$  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?

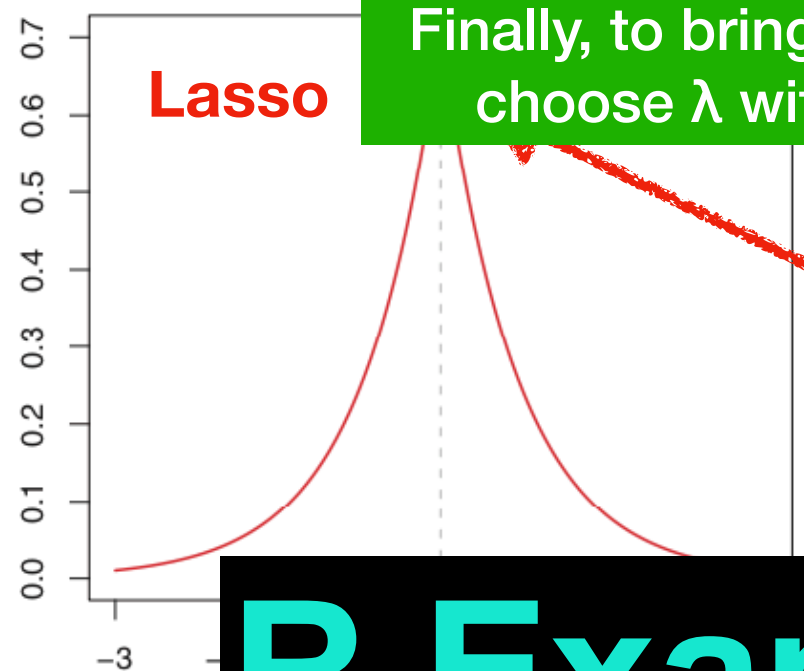
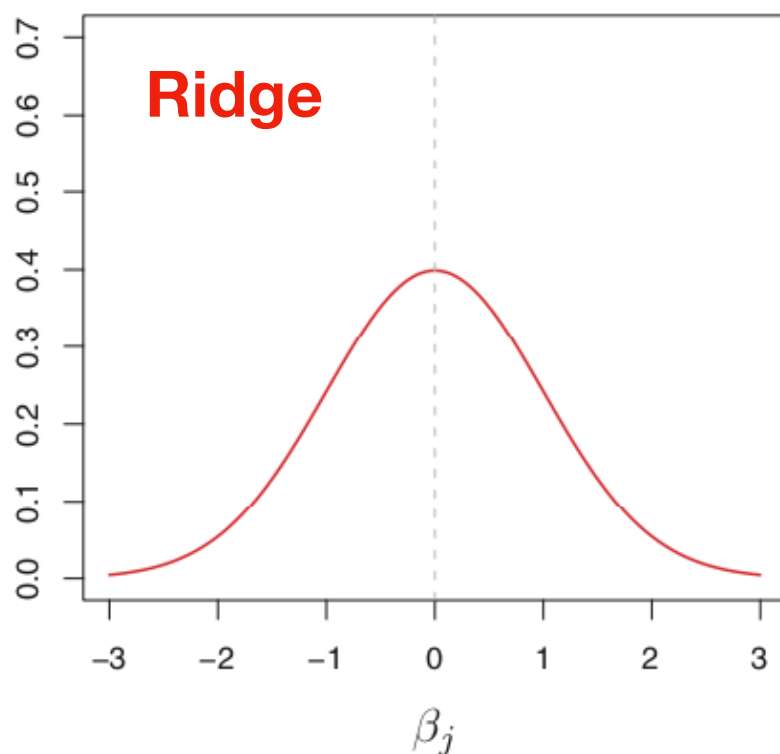
$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^p |\beta_j|$$

**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  $\lambda$
- (2) Natural way to perform variable selection

“Shrinkage” of less important parameters to zero

Assumption about how probable a value of  $\beta_j$  is



Finally, to bring it full circle - we can choose  $\lambda$  with Cross-Validation

Parameters are more likely to be zero

# R Examples!











# **Unsupervised Learning:**

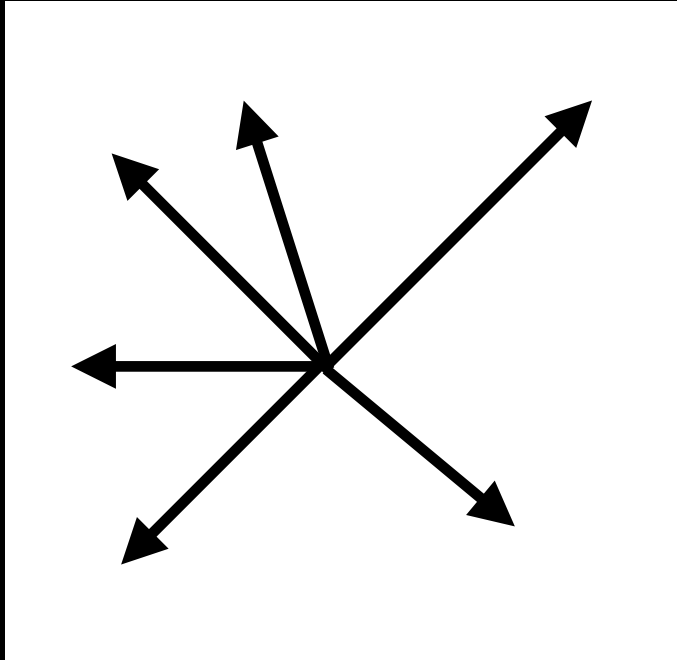
## **An intro to Principle Component Analysis**

# **Unsupervised Learning:**

## **An intro to Principle Component Analysis**

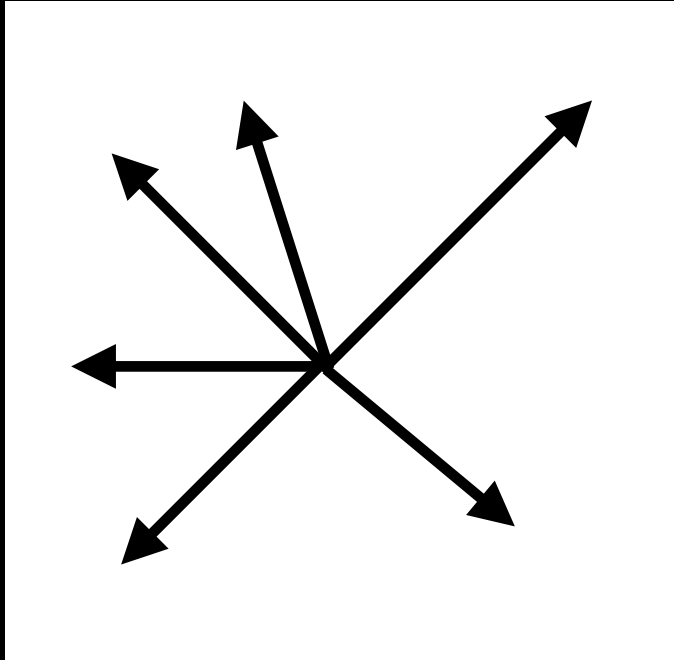
**... what do we do when we don't know anything**

# Principle Component Analysis: In Pictures

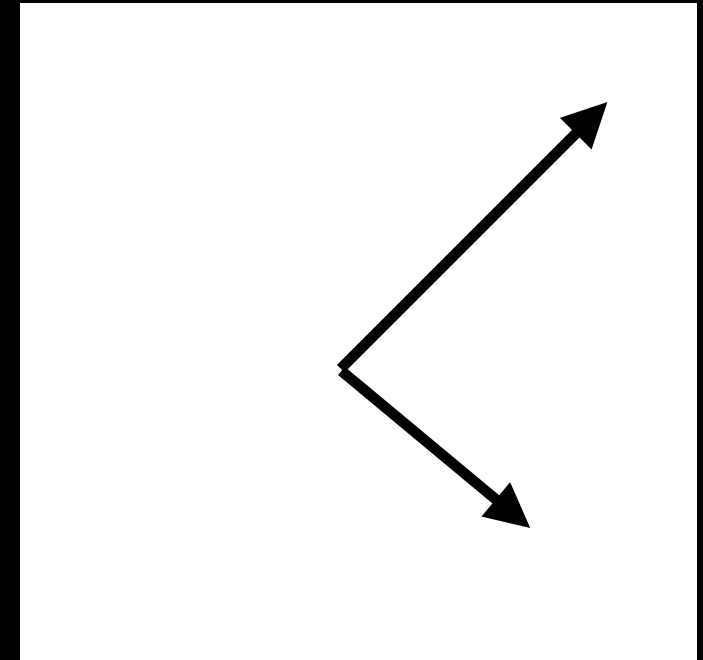
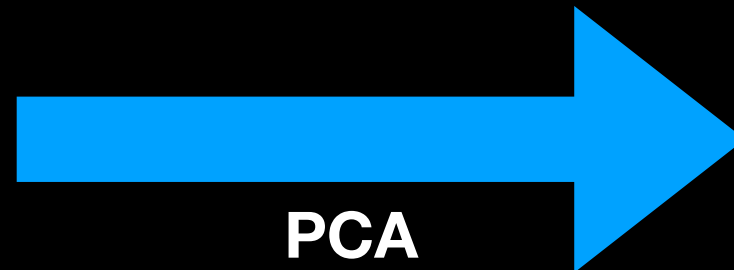


How many vectors do I  
need to define a 2D  
space?

# Principle Component Analysis: In Pictures

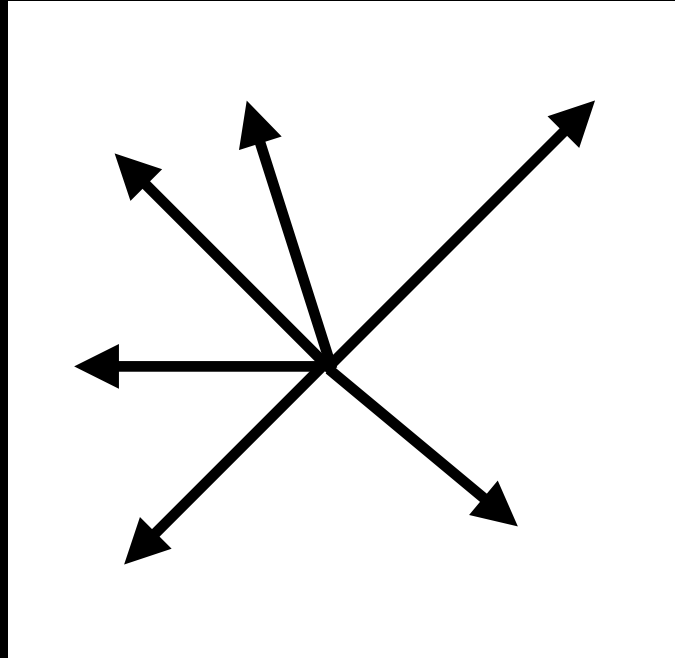


**How many vectors do I  
need to define a 2D  
space?**

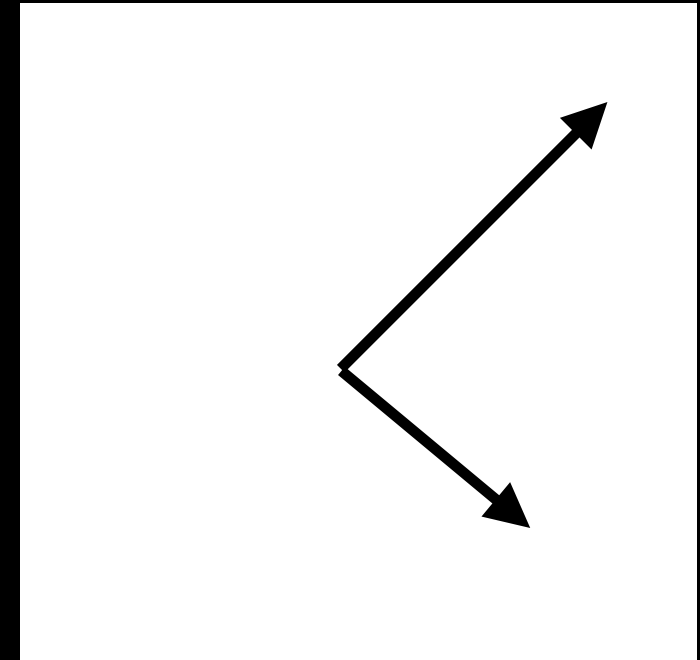
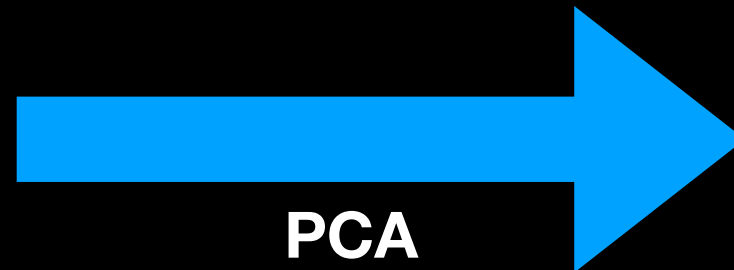


**Minimum number of  
vectors to define a space  
in a certain number of  
dimensions**

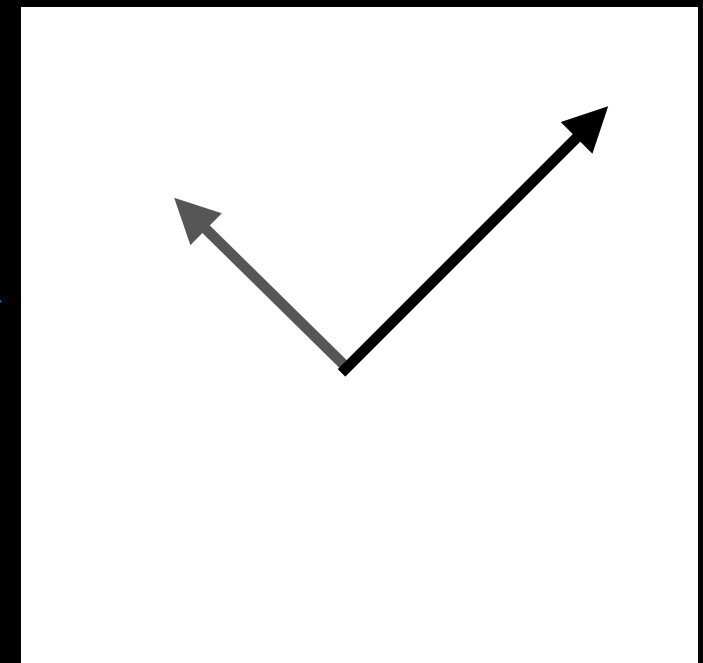
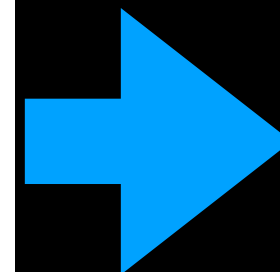
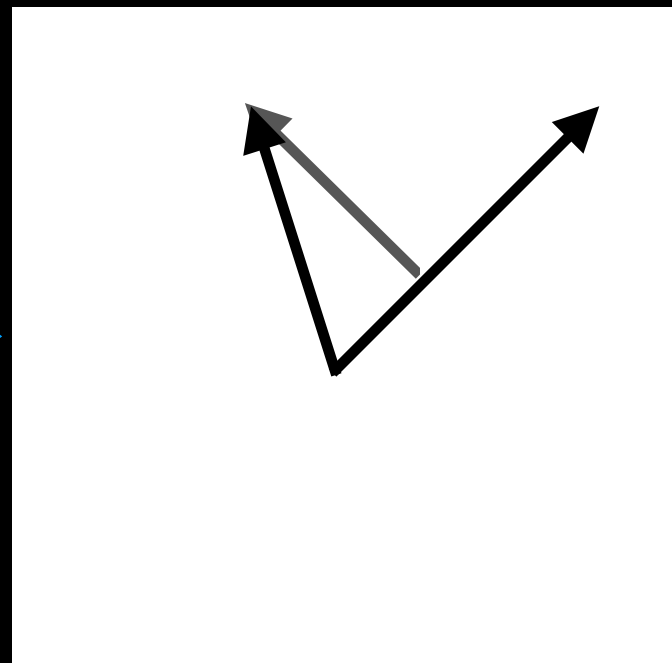
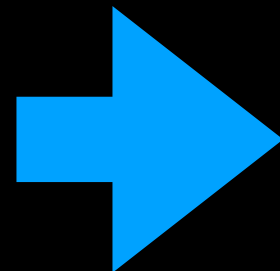
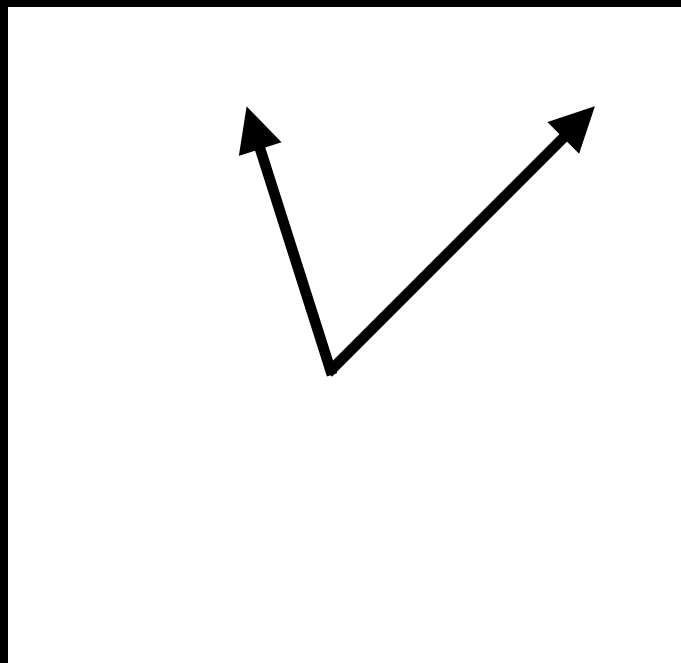
# Principle Component Analysis: In Pictures



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



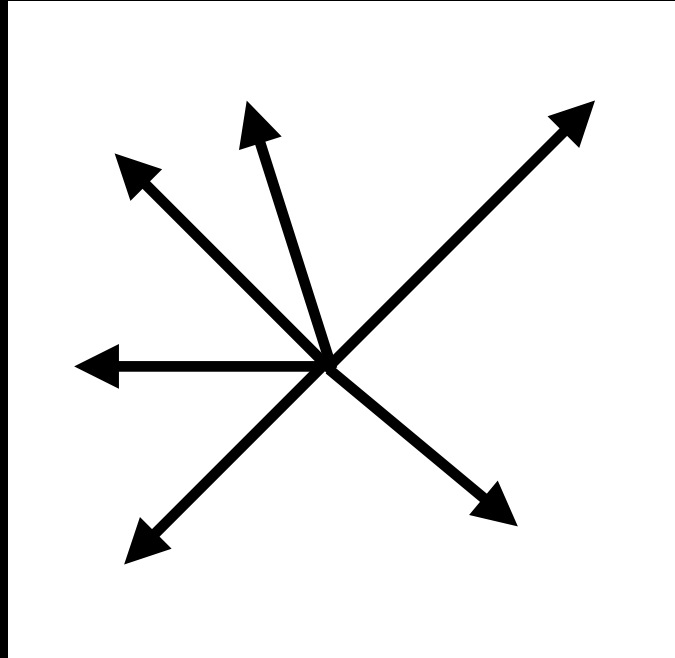
Minimum number of  
vectors to define a space  
in a certain number of  
dimensions



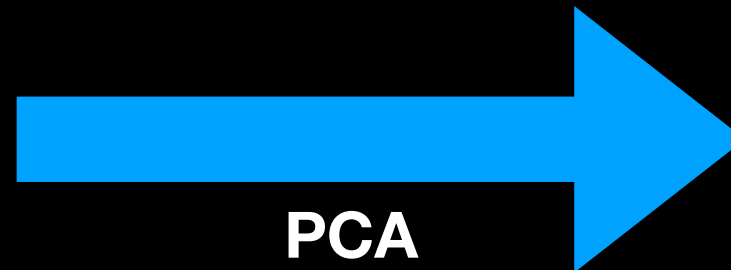
Constructing orthogonal vectors



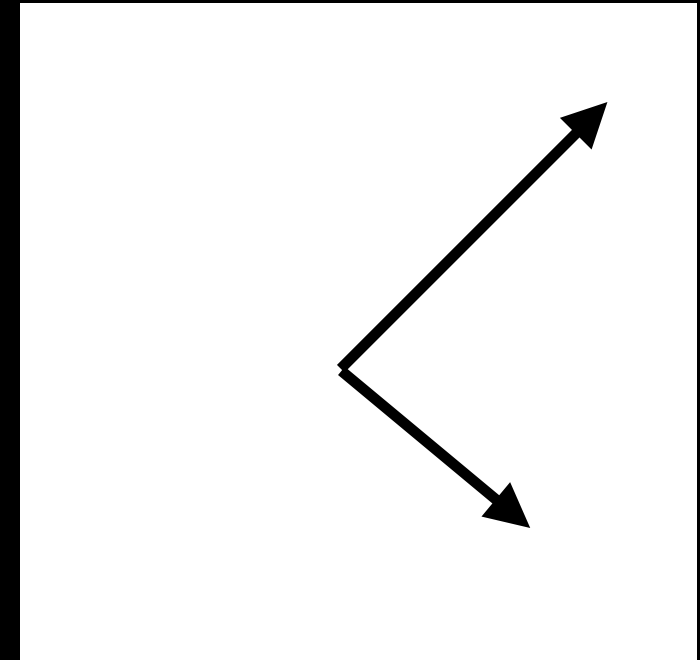
# Principle Component Analysis: In Pictures



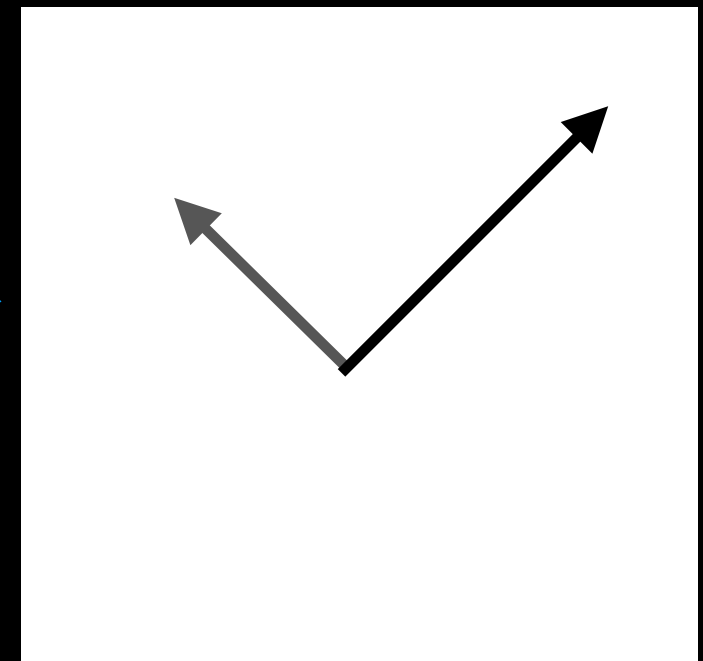
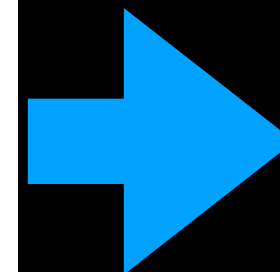
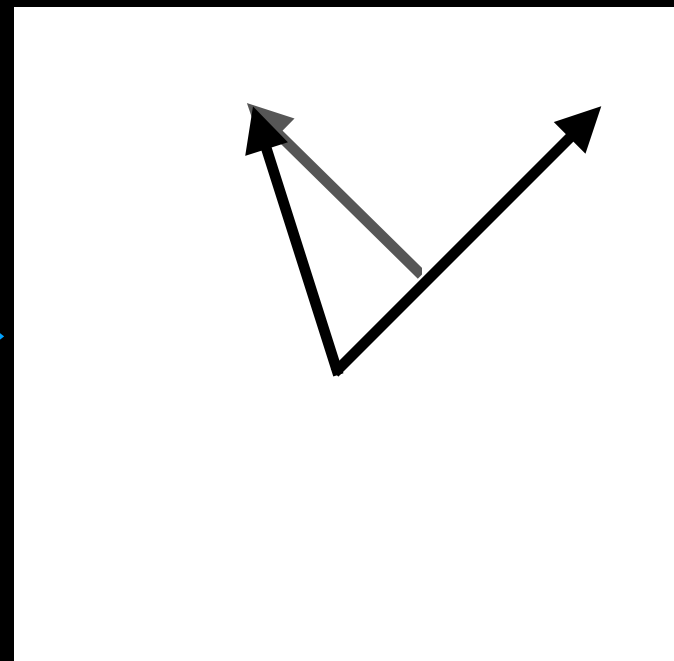
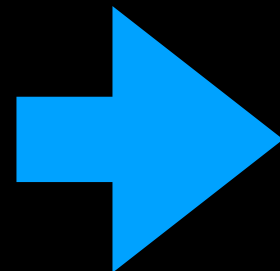
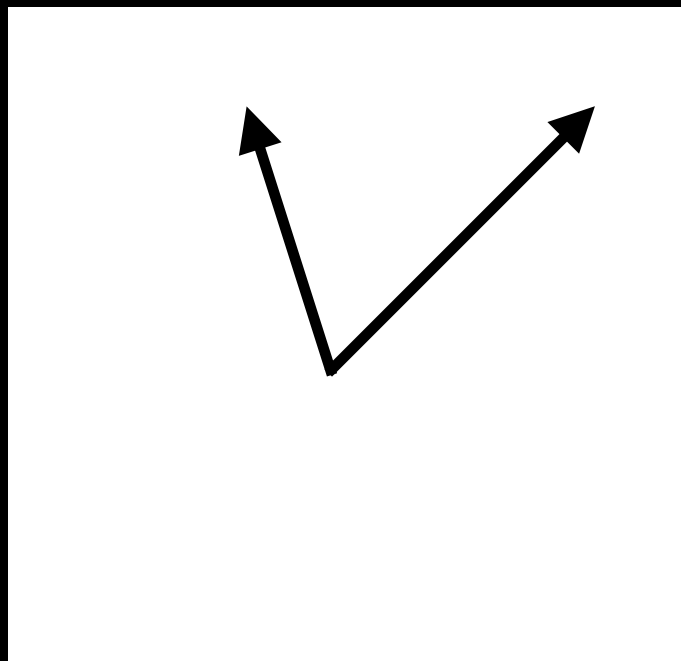
How many vectors do I need to define a 2D space?



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

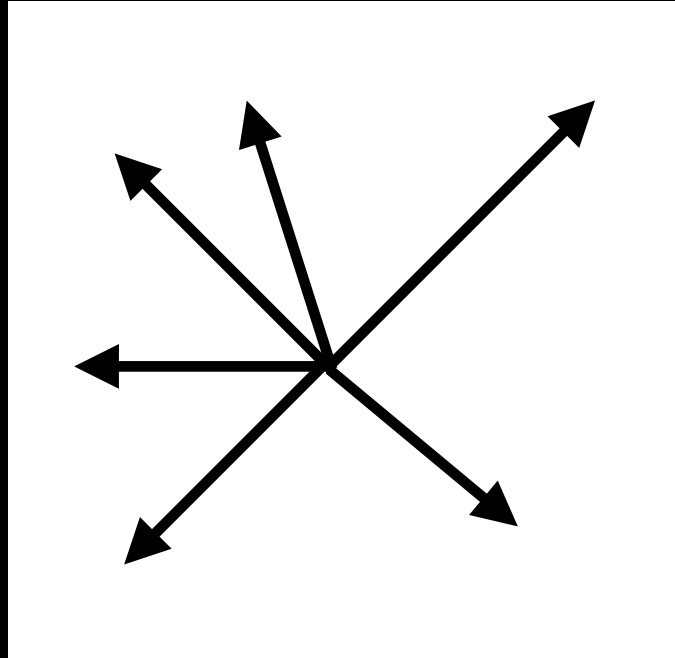


Minimum number of vectors to define a space in a certain number of dimensions

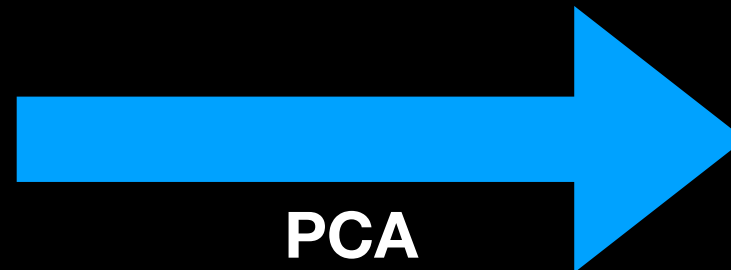


Constructing orthogonal vectors

# Principle Component Analysis: In Pictures

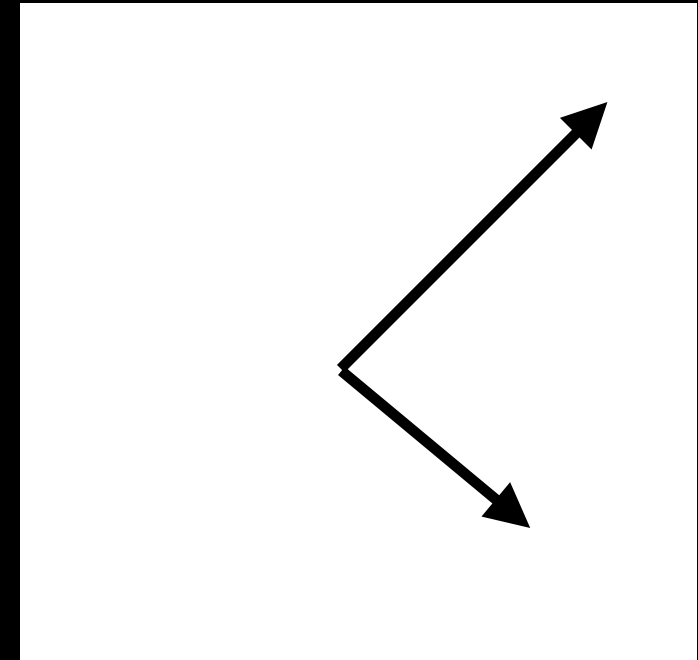


How many vectors do I need to define a 2D space?

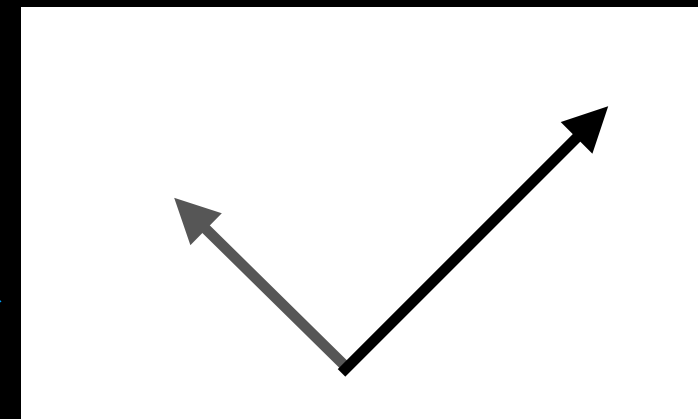
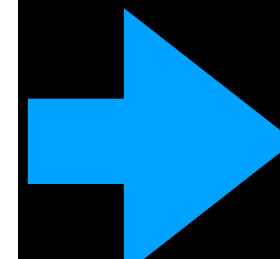
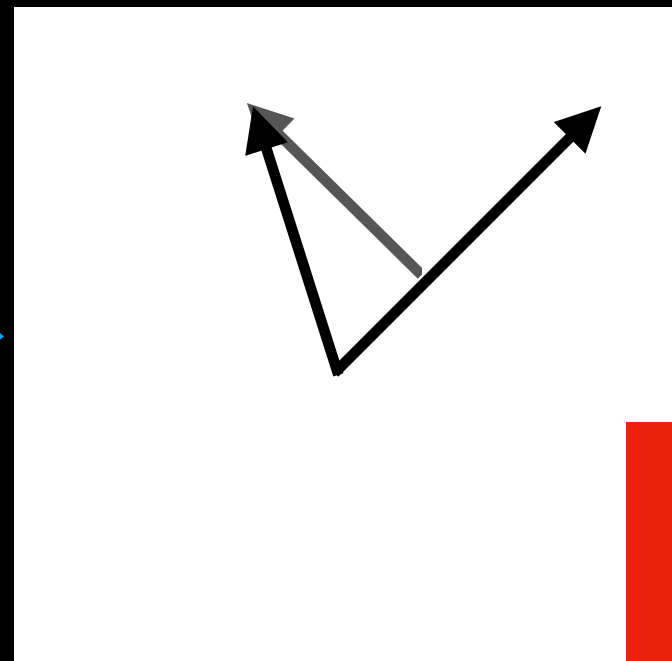
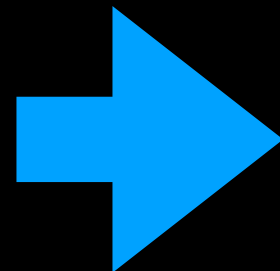
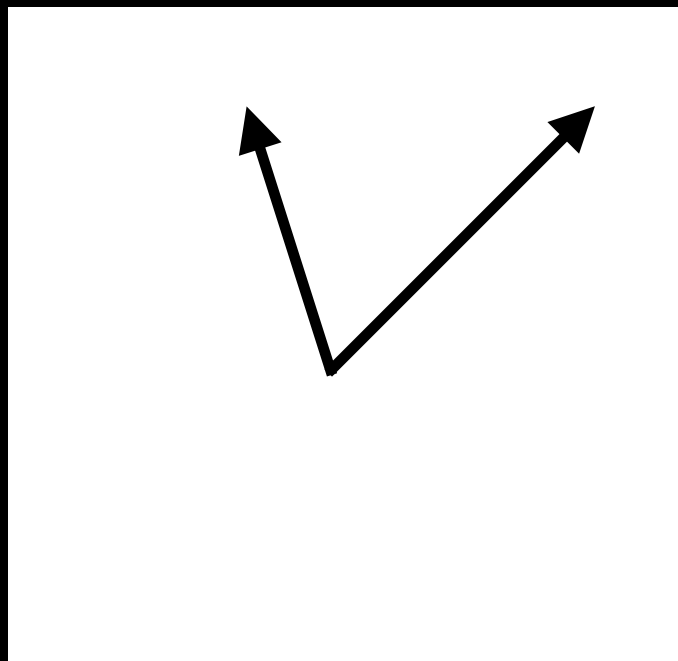


PCA

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

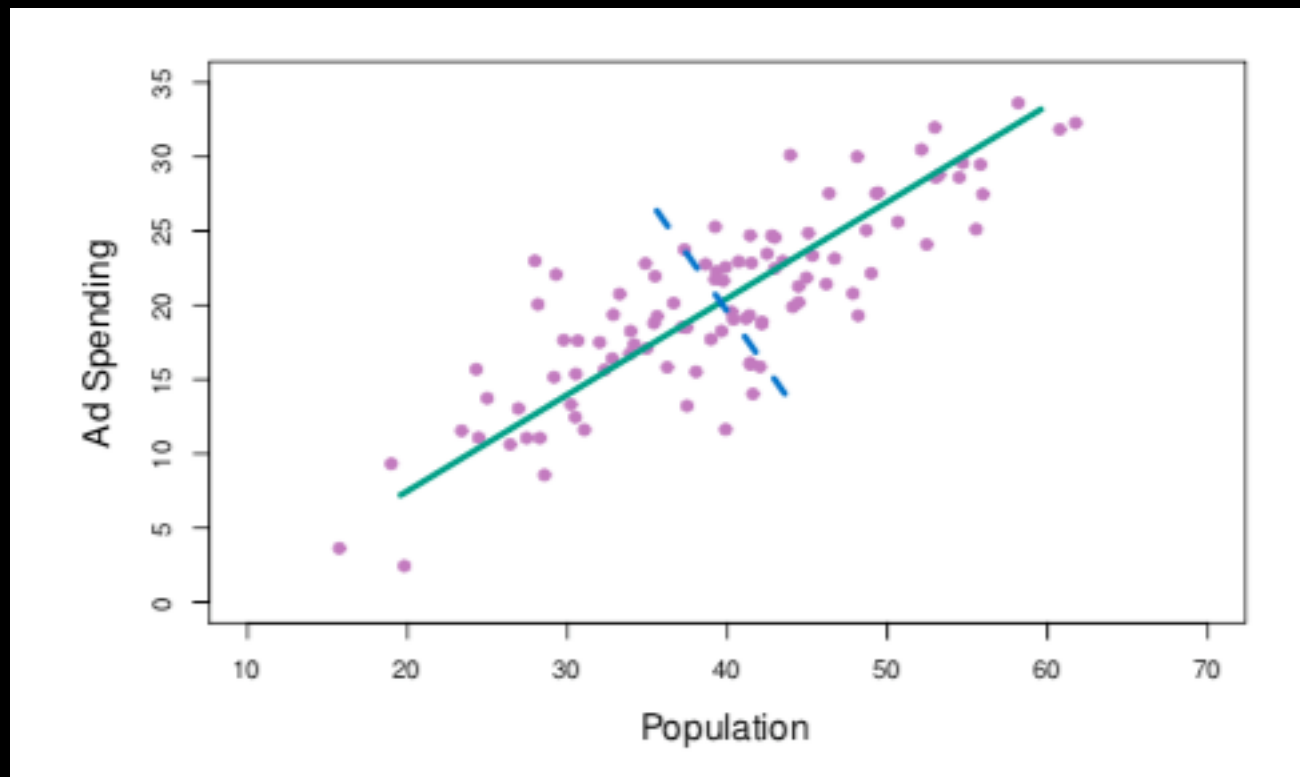
Can also think about PCA in terms of variance:

- 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.

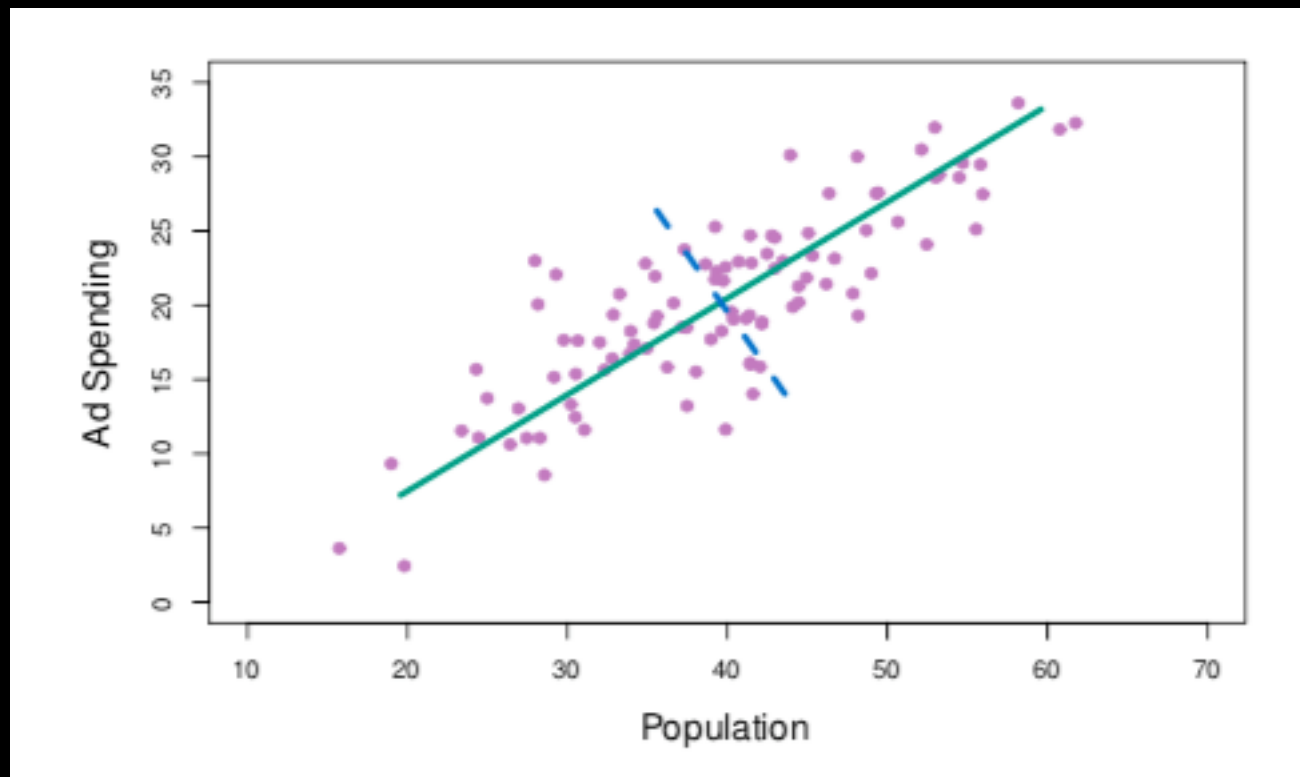


**We've essentially already fitted the first PC by fitting lines!**

# 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!**



