PLS206 Applied Multivariate Modeling in Agricultural and Environmental Sciences

Regularization:
- Ridge regression
- The Lasso

Cross validation:
- "Validation"
- K-fold CV
- Leave one out (LOOCV)
- Measuring Accuracy
  - Classification
  - Continuous response
- Selecting models

Parameter	Exploration	Inference	Prediction
Purpose	generate hypotheses	test hypotheses	forecast the future accurately
Priority	thoroughness	avoid false positives	minimize error
A priori hypotheses	not necessary	essential	not necessary, but may inform model specification
Emphasis on model selection	important	minimal	important
Key statistical tools	any	null hypothesis significance tests	AIC; regularization; machine learning; cross-validation; out-of-sample validation
Pitfalls	fooling yourself with over-fitted models with spurious covariate effects	misrepresenting exploratory tests as tests of a priori hypotheses	failure to rigorously validate prediction accuracy with independent data

JOURNAL ARTICLE

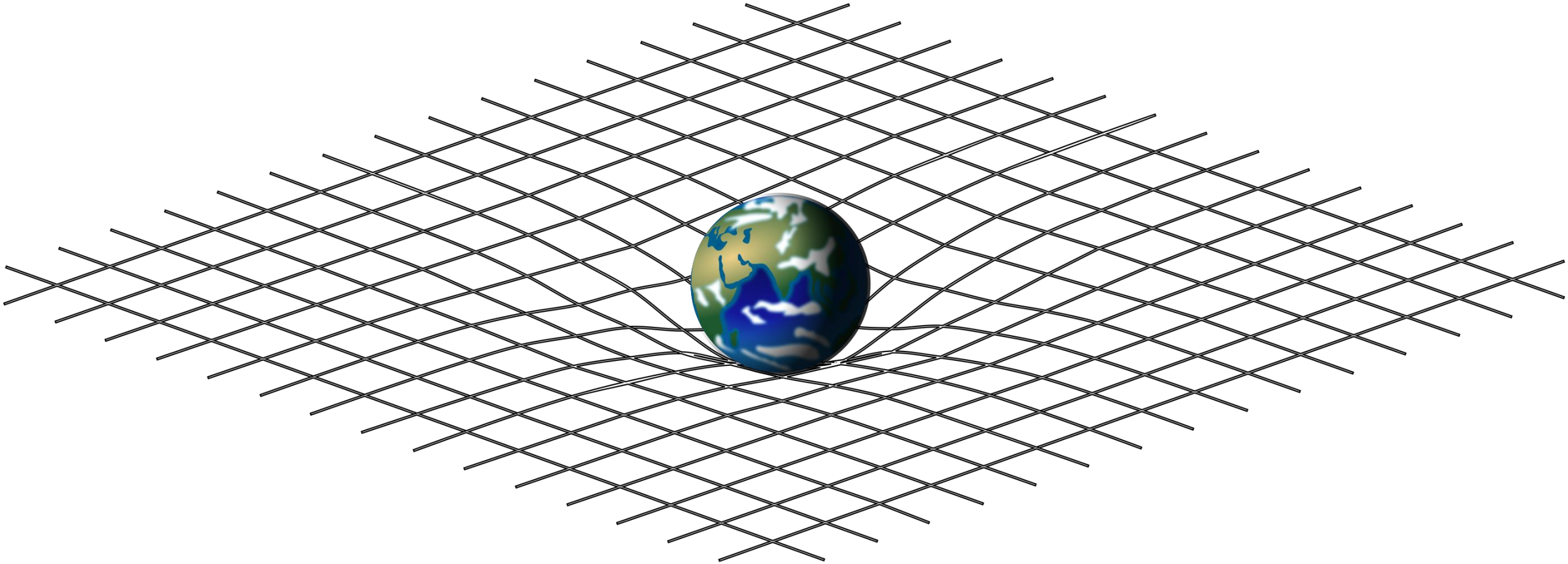
Prediction in ecology and evolution

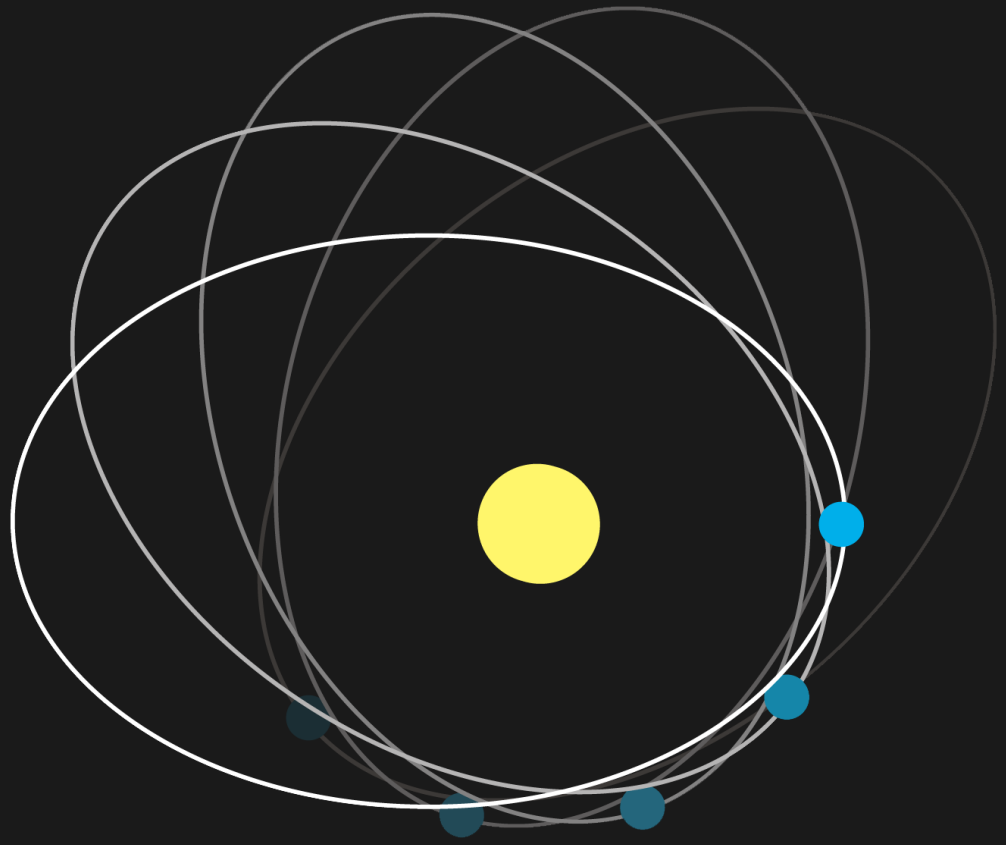
Andrew P Hendry

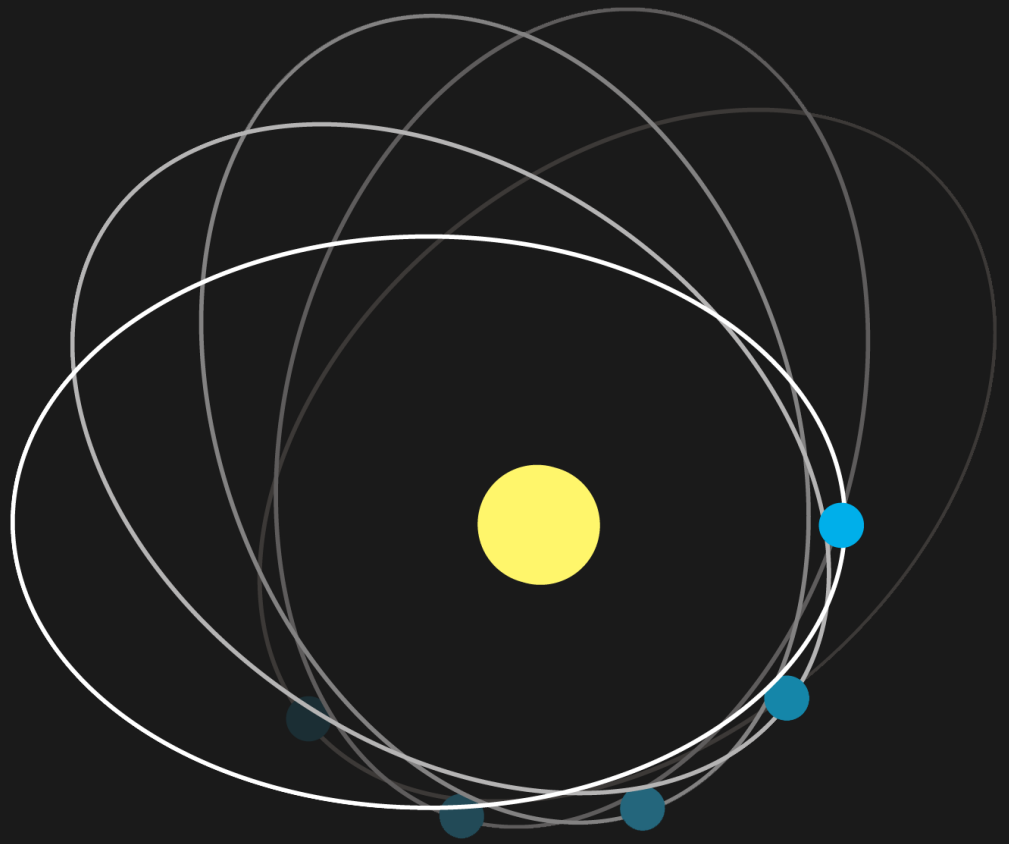
BioScience, biad083, https://doi.org/10.1093/biosci/biad083

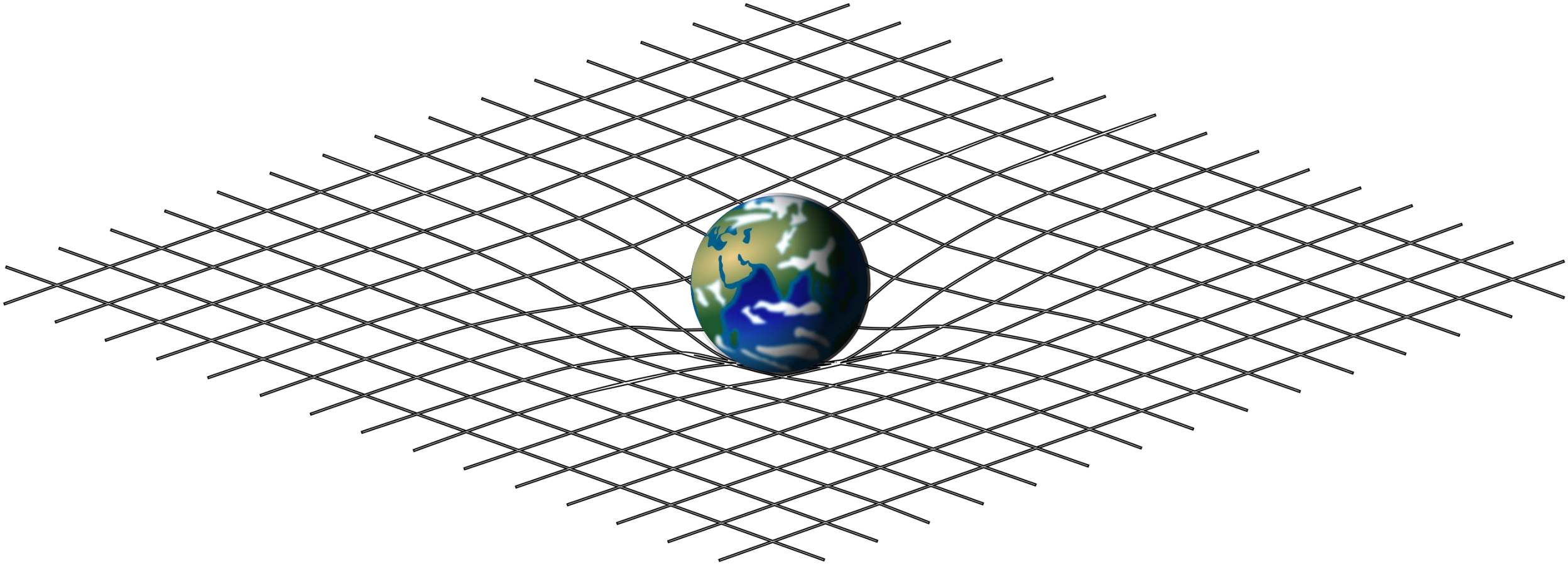
Published: 27 October 2023 **Article history** ▾

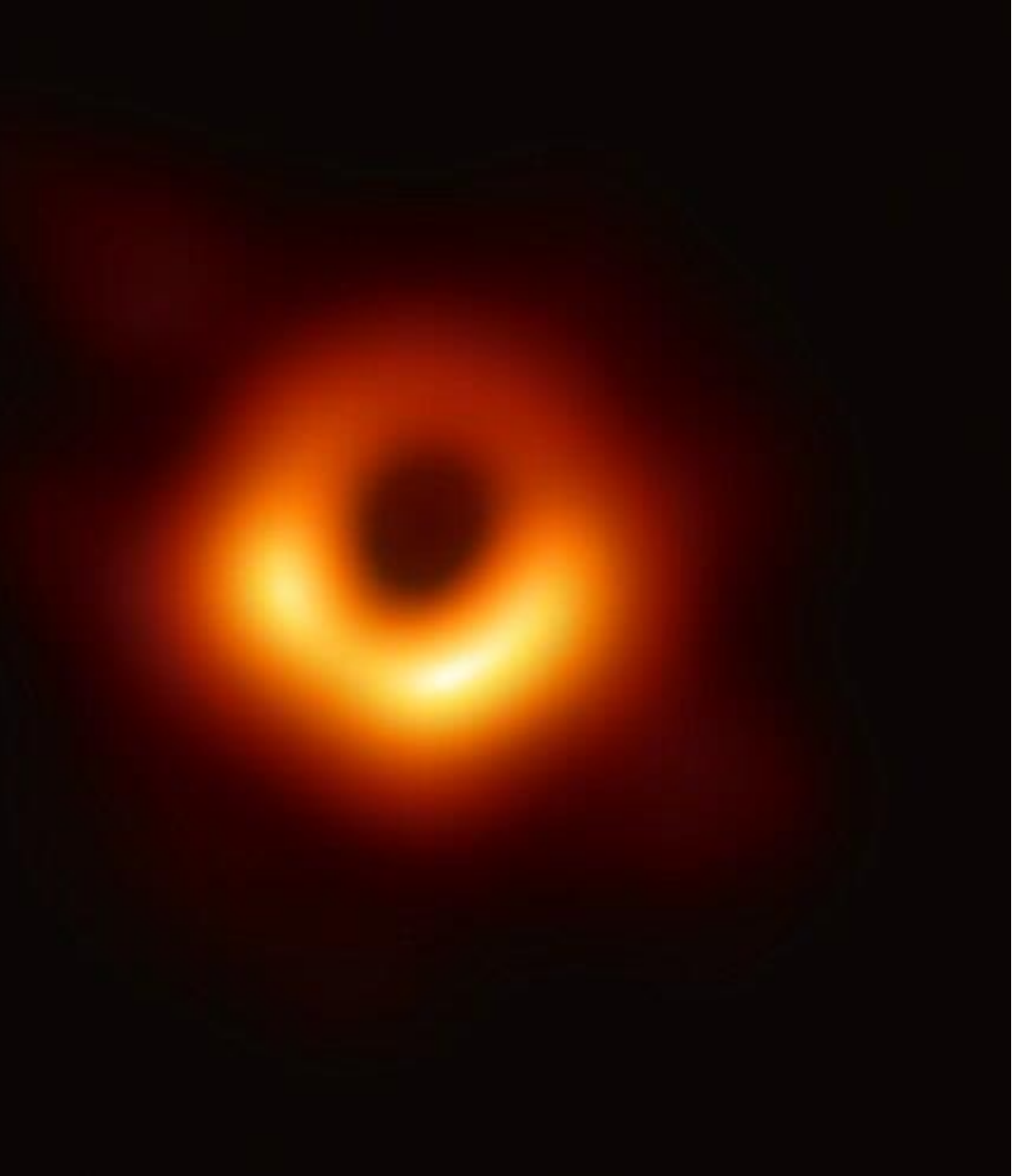
"Prediction is frequently asserted to be the *sine qua non* of science"

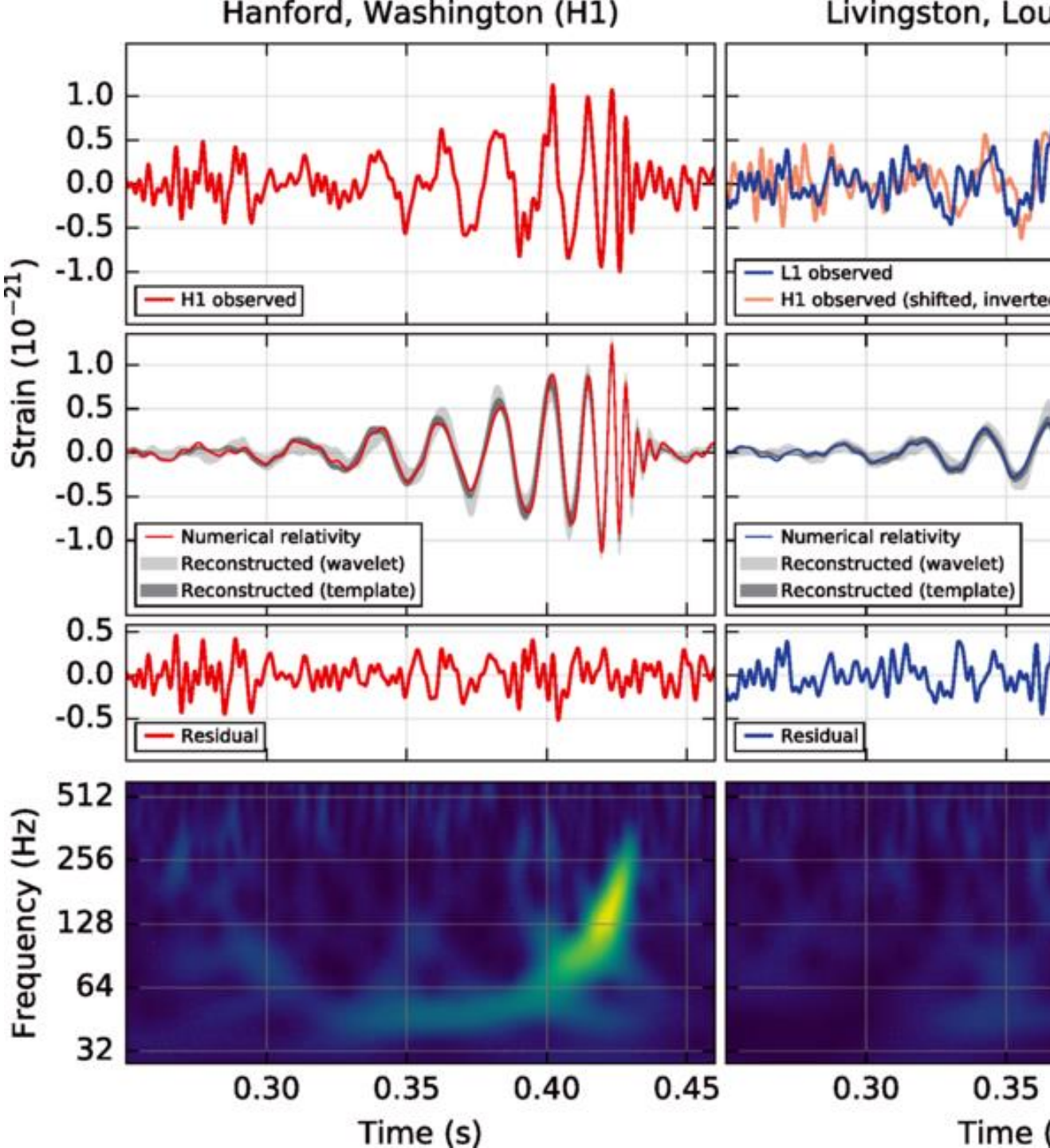




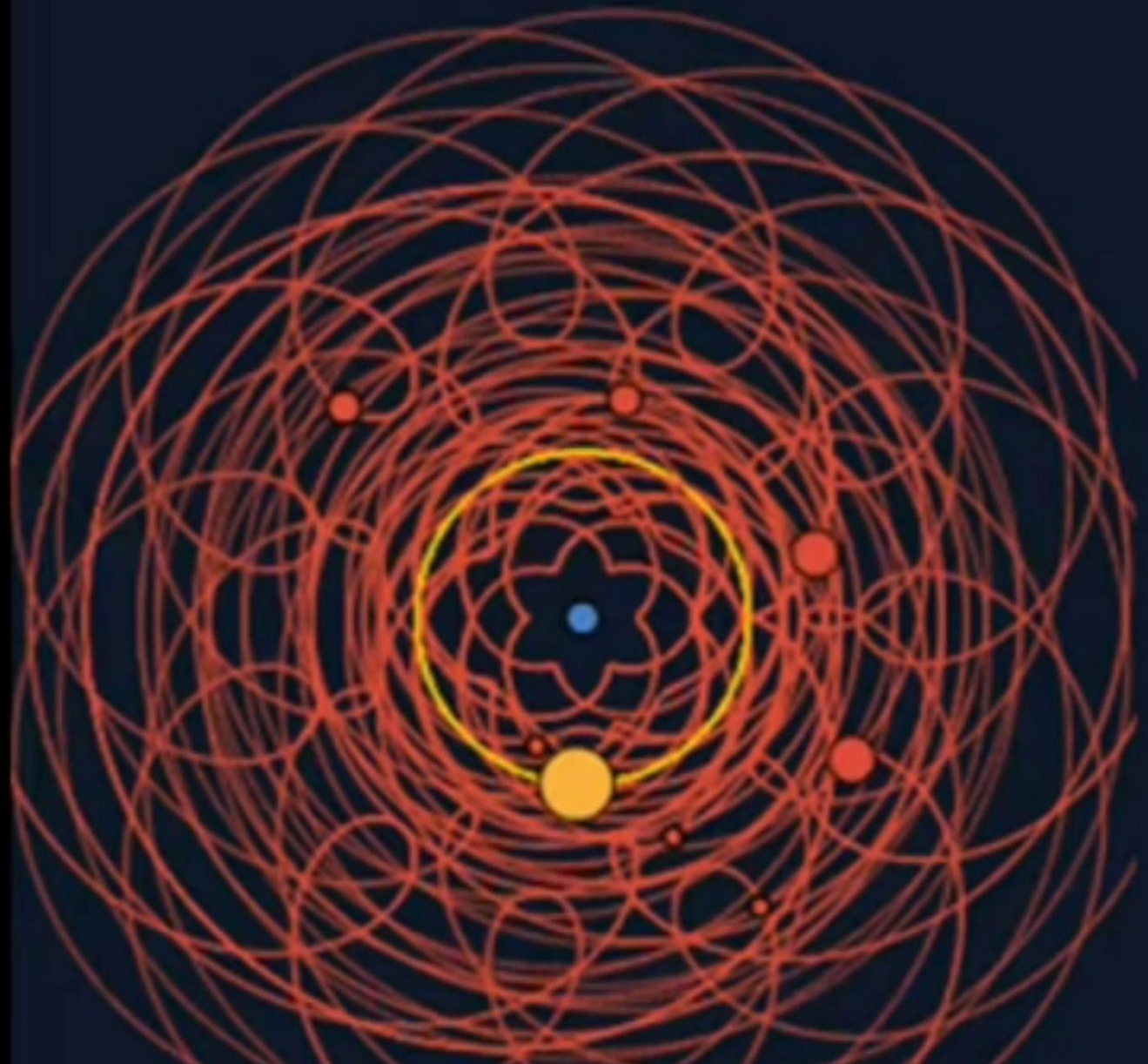




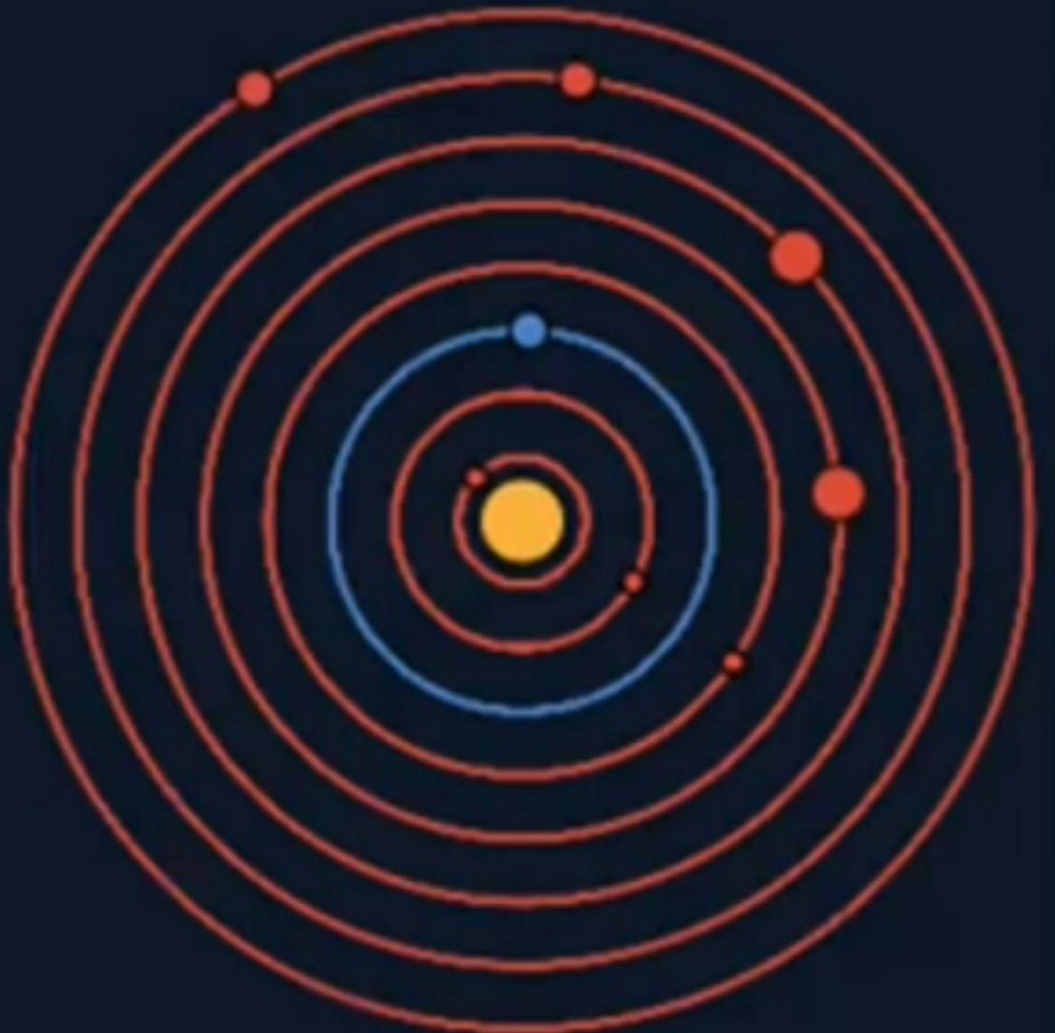


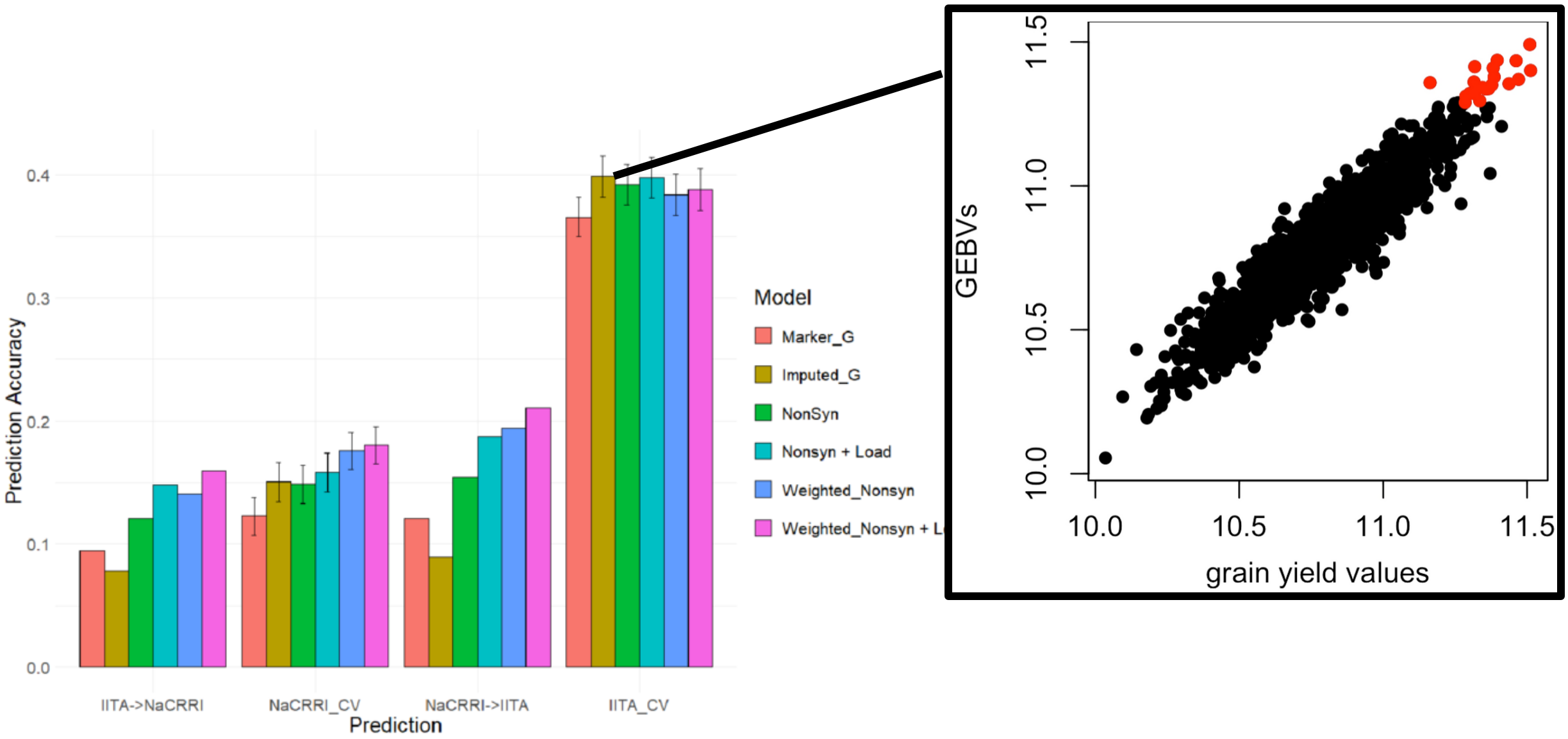

Hanford, Washington (H1)
Livingston, Lou
1.0
0.5
0.0
-0.5
-1.0
H1 observed
L1 observed
H1 observed (shifted, inverte
Strain (10^{-21})
Numerical relativity
Reconstructed (wavelet)
Reconstructed (template)
Residual
512
256
128
64
32
Frequency (Hz)
0.30
0.35
0.40
0.45
Time (s)

Geocentrism



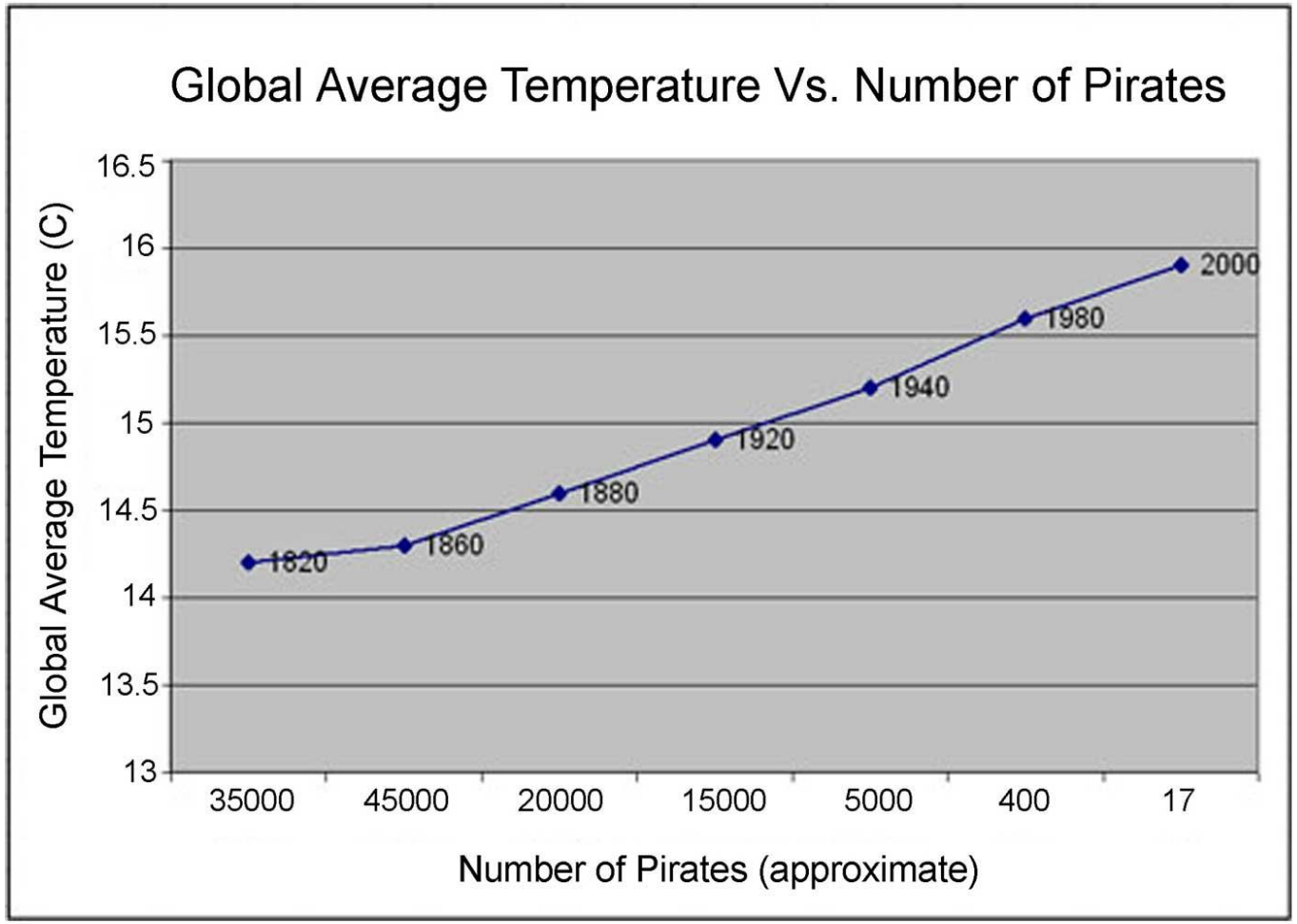
Heliocentrism




Prediction Accuracy
0.0
0.1
0.2
0.3
0.4
IITA->NaCRRI
NaCRRI_CV
NaCRRI->IITA
IITA_CV
Prediction
Model
Marker_G
Imputed_G
NonSyn
Nonsyn + Load
Weighted_Nonsyn
Weighted_Nonsyn + L
GEBVs
10.0
10.5
11.0
11.5
grain yield values

Caution: correlation does not equal causation!

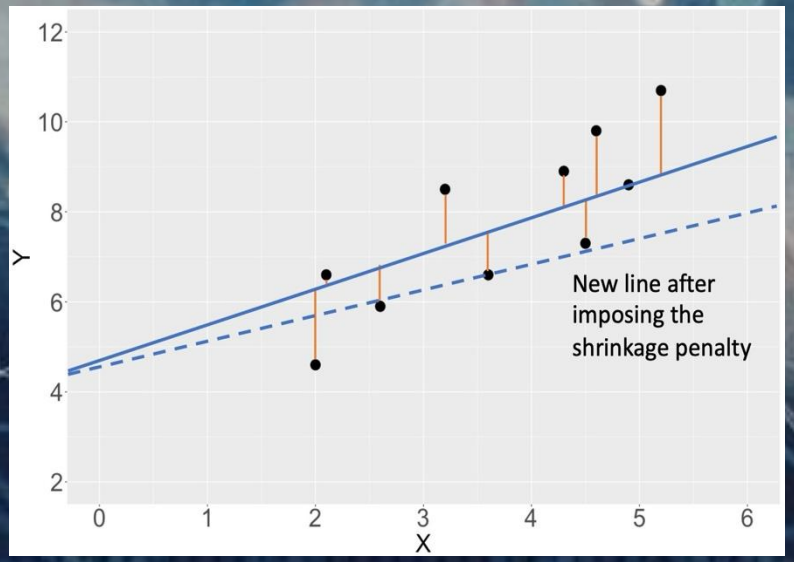
STOP GLOBAL WARMING: BECOME A PIRATE



WWW.VENGANZA.ORG

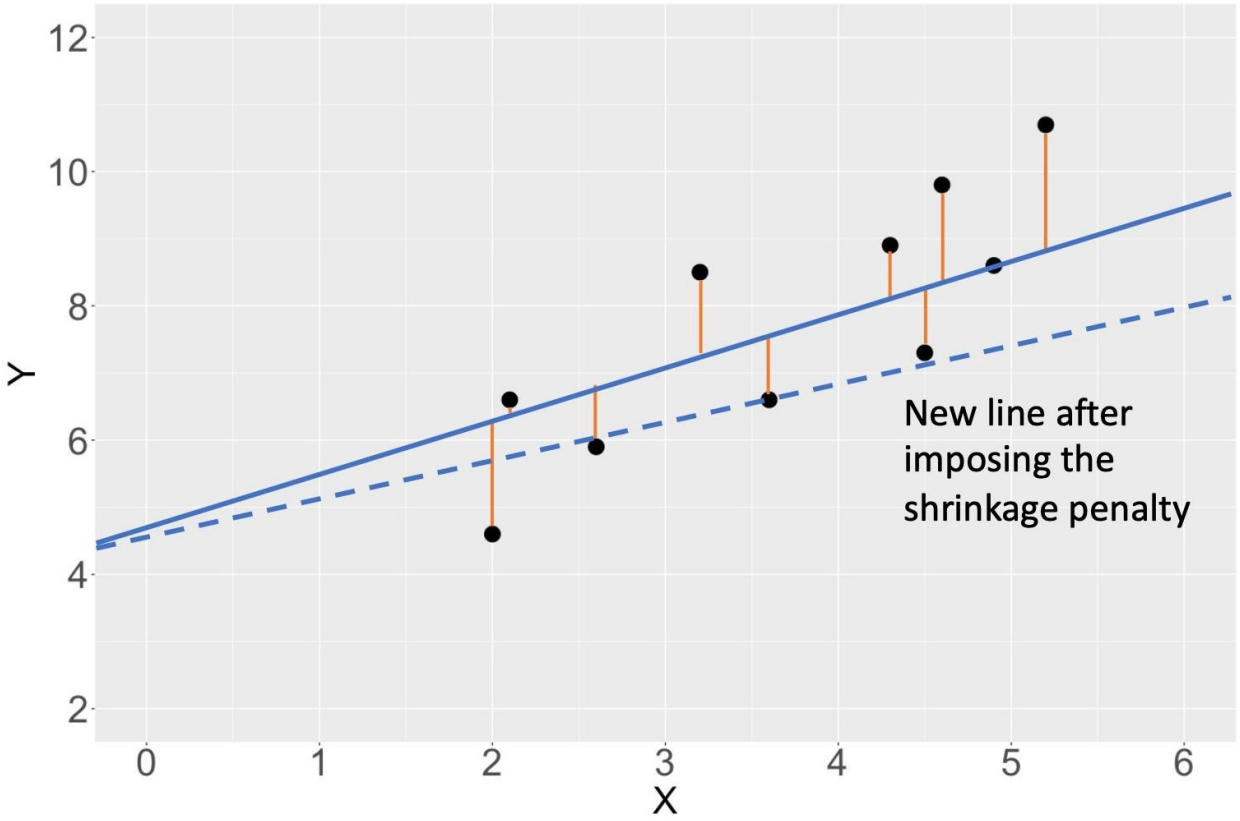
PLS206 **Applied Multivariate Modeling in Agricultural and Environmental Sciences**

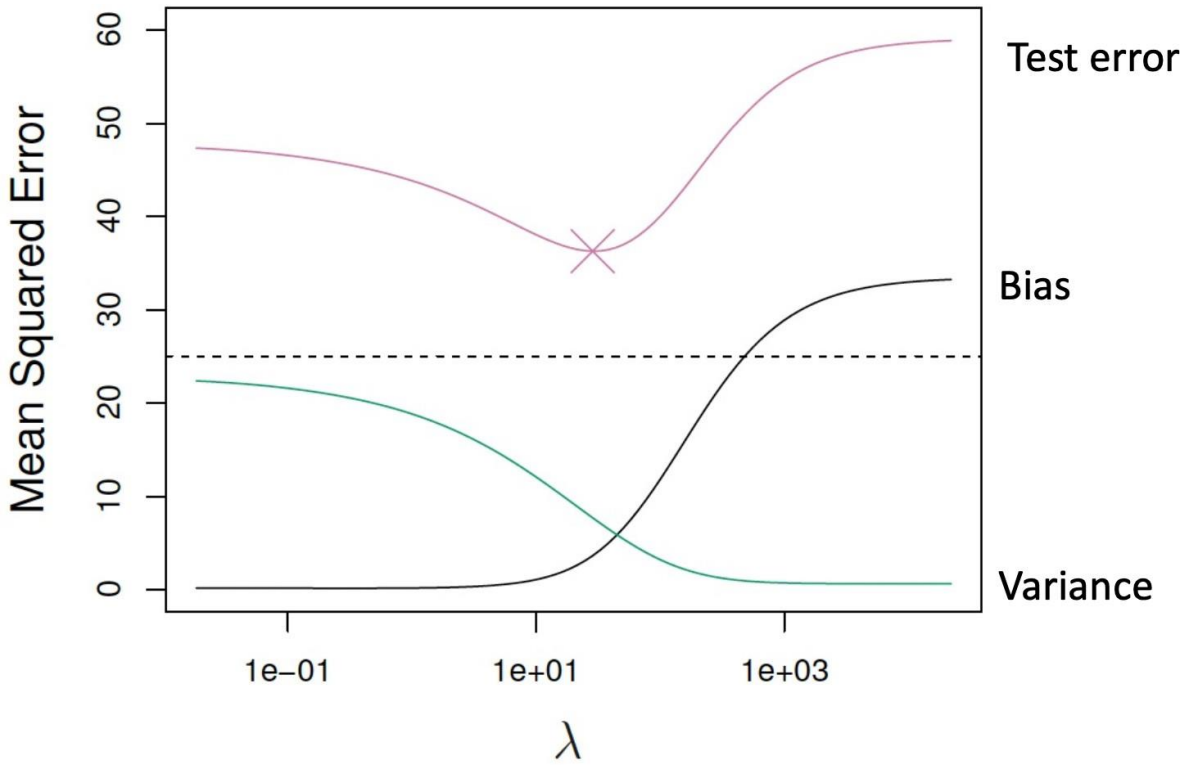
Regularization



Regularization

Reducing slopes can give better accuracy to predict future data

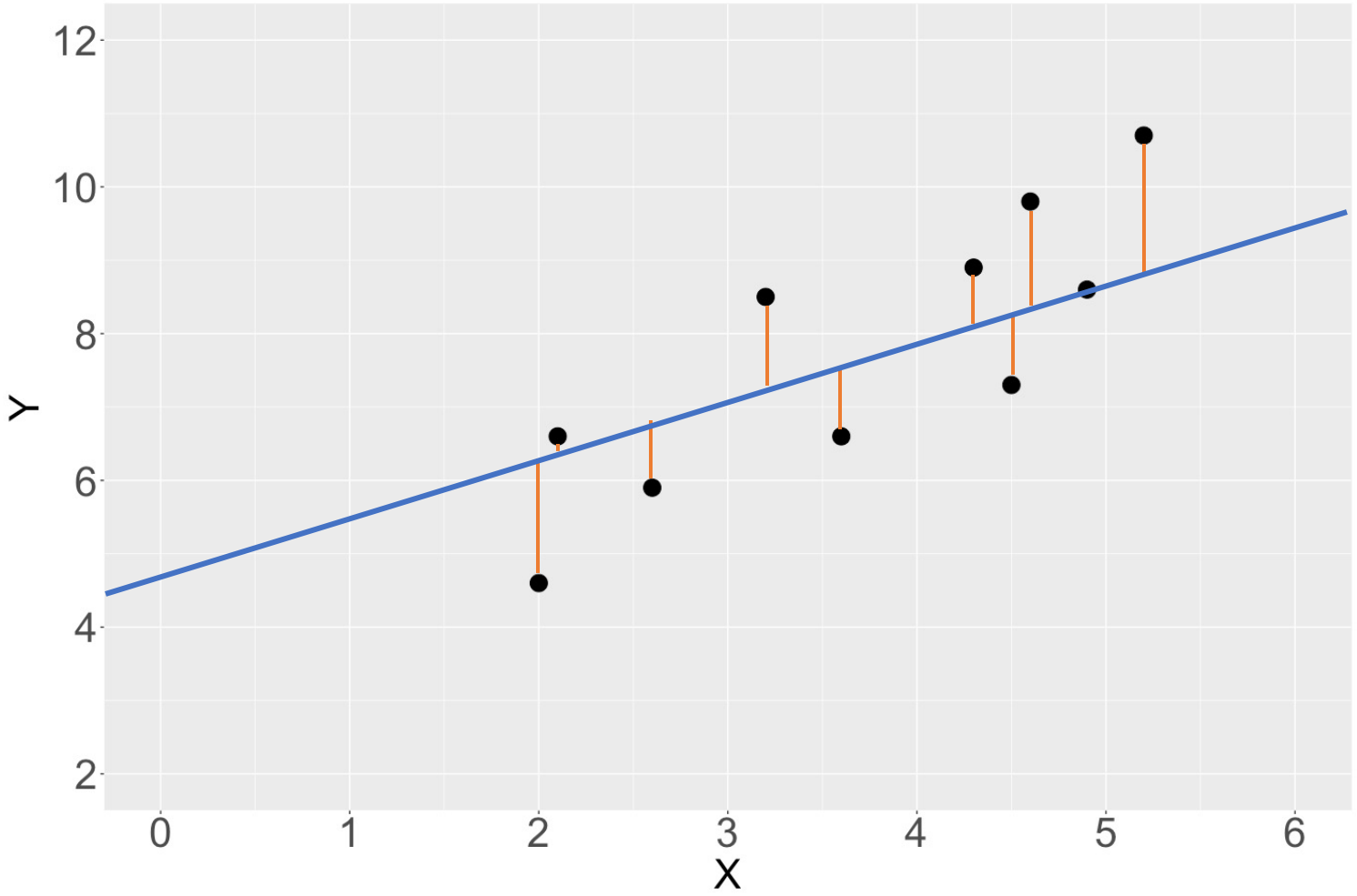




Regularization or Shrinkage Methods

- 3 related methods: Ridge regression, Lasso, and Elastic net
- These methods shrink or regularize the coefficients (aka the betas or slopes) of a linear model towards zero
- This constraint can improve the model fit by reducing the variance of the model, which gives it better predictive ability
- Remember: variance refers to the amount by which our model would change if we estimated it using a different training set
- These are great methods for datasets with huge numbers of predictors
- Historically, they haven't been used a lot in ecological/environmental sciences, but they are gaining popularity

What does it mean to shrink acoefficient?



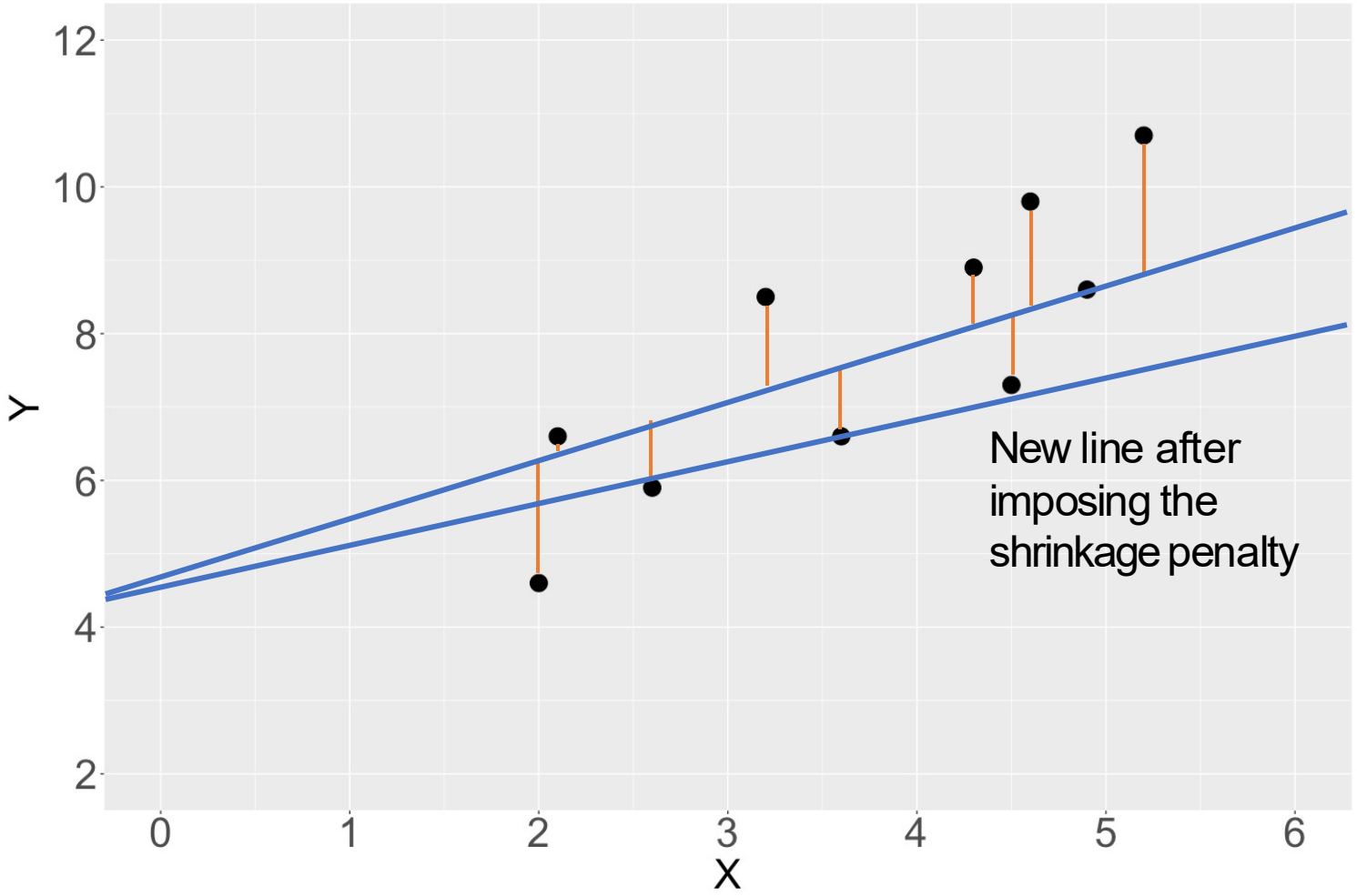
When we fit a linear regression, we find the equation for the line based on least squares

The goal of least squares is to find the line that minimizes the residual sum of squares

This line has a slope and an intercept

Regularization methods shrink the slope of the line by imposing a penalty on the least squares solution

What does it mean to shrink acoefficient?



When we fit a linear regression, we find the equation for the line based on least squares

The goal of least squares is to find the line that minimizes the residual sum of squares

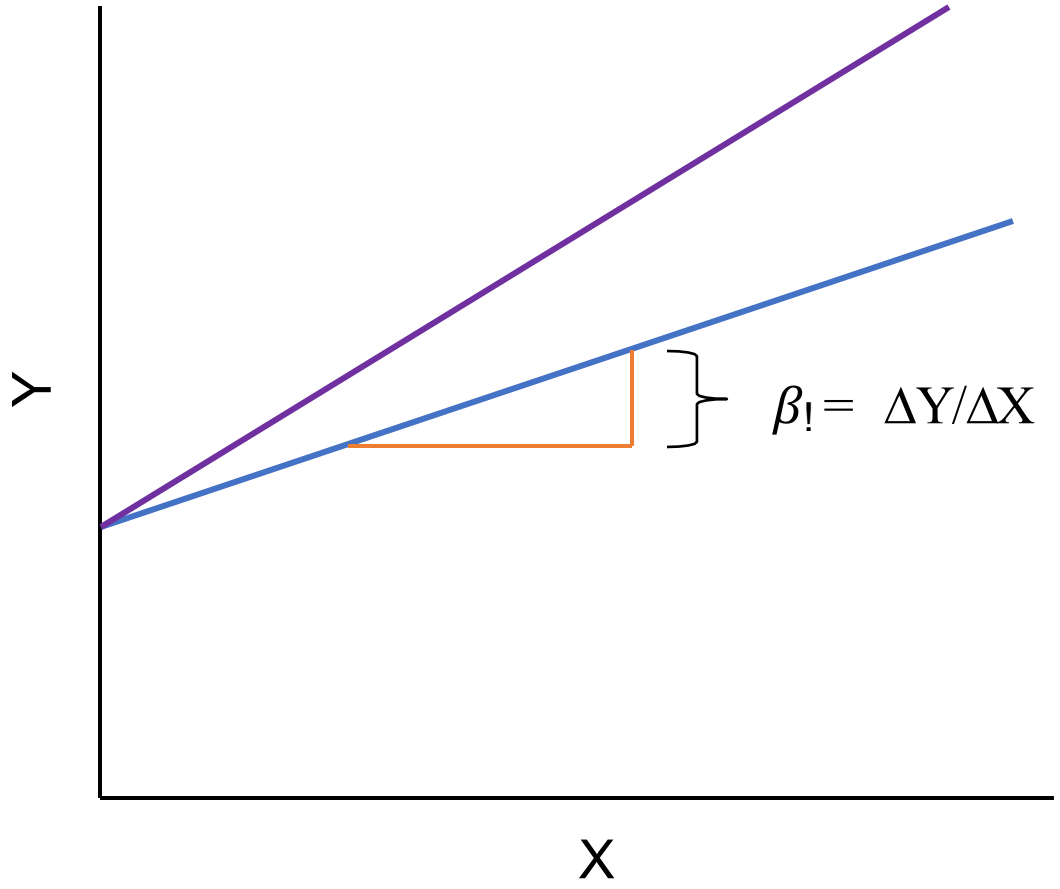
This line has a slope and an intercept

Regularization methods shrink the slope of the line by imposing a penalty on the least squares solution

Why do this?

In fitting the least squares line, we have minimized bias (the difference between predicted and observed values). By shrinking the slope slightly, we increase the bias of the model a little, but in doing so, we can often see a large decrease in the variance of the model

Why does the variance decrease when we shrink the coefficient or slope?



When the slope is steep, the predictions for Yare more sensitive to changes in X

As the slope decreases slightly, the predictions for Yare less sensitive to changes in X

So by decreasing the slope, we decrease the variance or the amount the model would change if we estimated Y using a different set of Xs

Ridge Regression

In Ridge regression, we look for the coefficients and the value of lambda that minimize the value of this equation

Shrinkage Penalty

$$RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

Must scale your predictors!

Residual sum of squares

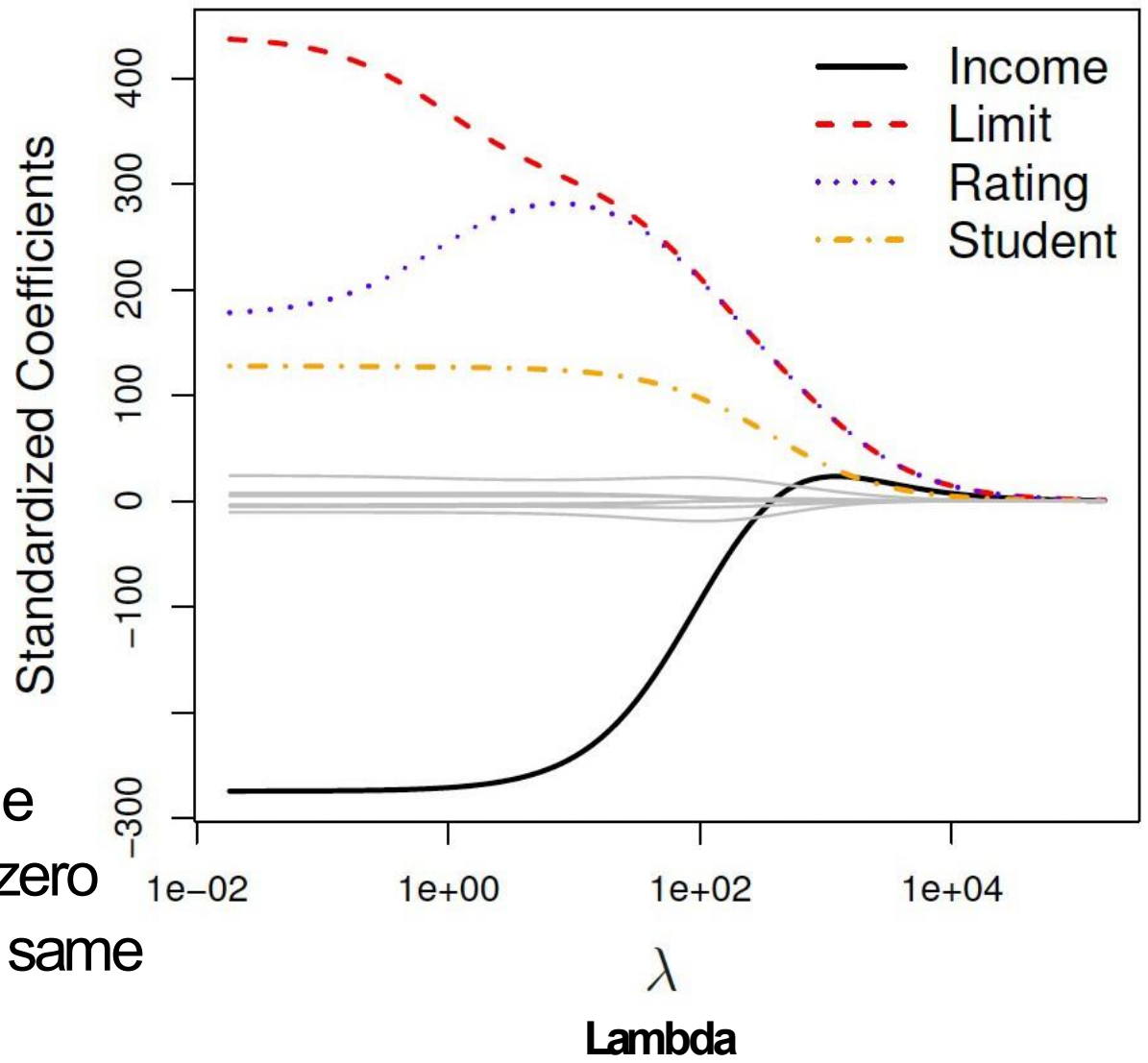
Lambda is the tuning parameter

Each of the slopes (betas) in the model squared and summed

This does not include the y-intercept, just the slopes

It can take on values from 0 to positive infinity
The size of lambda determines how large the penalty is and how much the coefficients are shrunk
As we increase lambda, the slopes become smaller, and the predictions become less and less sensitive (we reduce variance)
We determine the best value for lambda that balances the trade off between bias and variance using 10-fold cross-validation

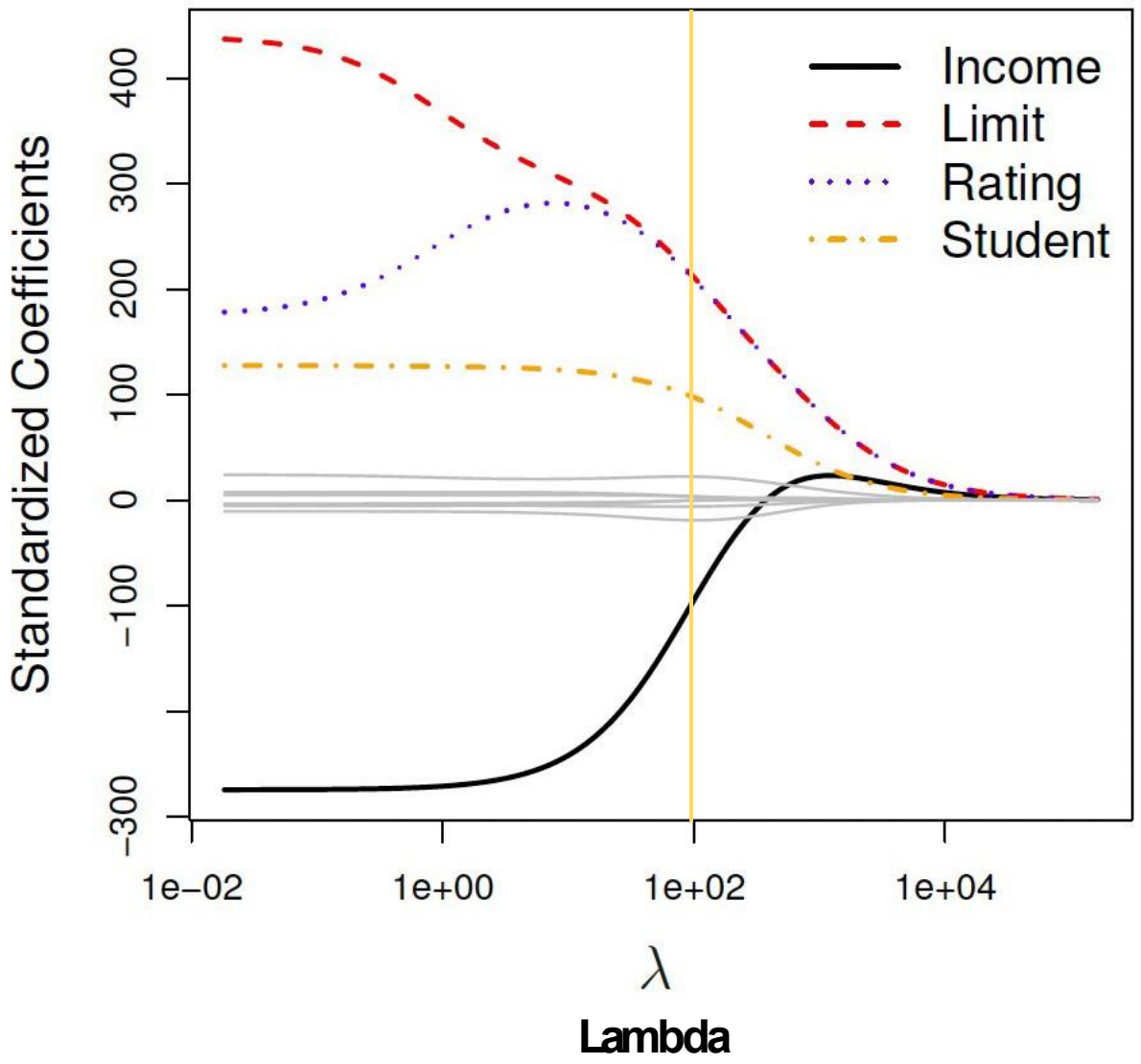
Fig 6.4 from Introduction to Statistical Learning



When lambda = 0, the shrinkage penalty is zero and the model is the same as with least squares

When lambda is large, the model coefficient estimates approach zero

Fig 6.4 from Introduction to Statistical Learning



There is a set of coefficient estimates for each value of lambda

For example, when lambda = 100, the coefficients for each predictor can be determined by measuring their y-axis values

They grey horizontal-ish lines are a group of predictors with slopes near zero

Best value for lambda is determined using cross validation

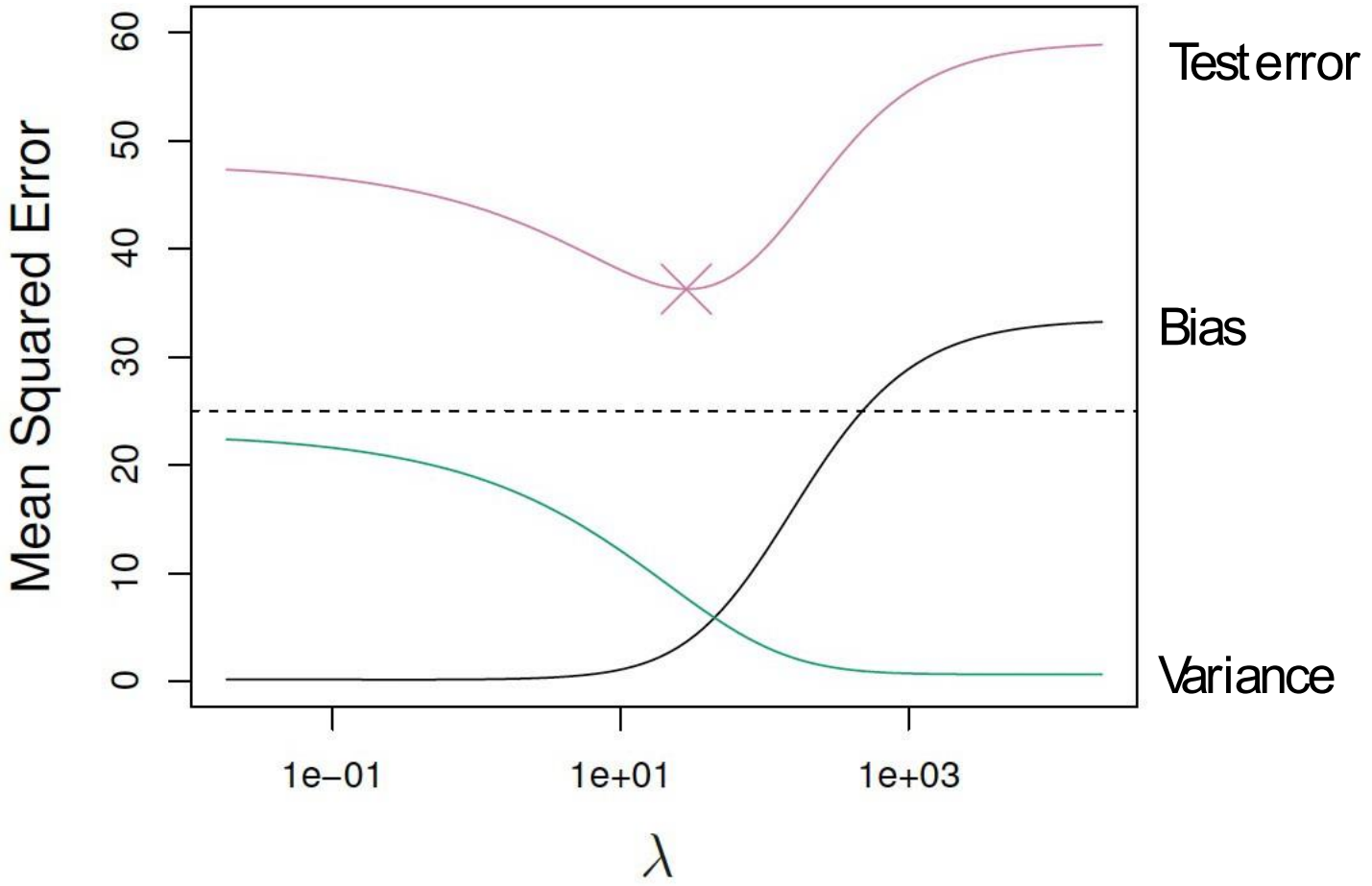


Fig 6.5 from Introduction to Statistical Learning

Optimal value of lambda marked by Xon the test error line

Value for lambda is determined using cross validation

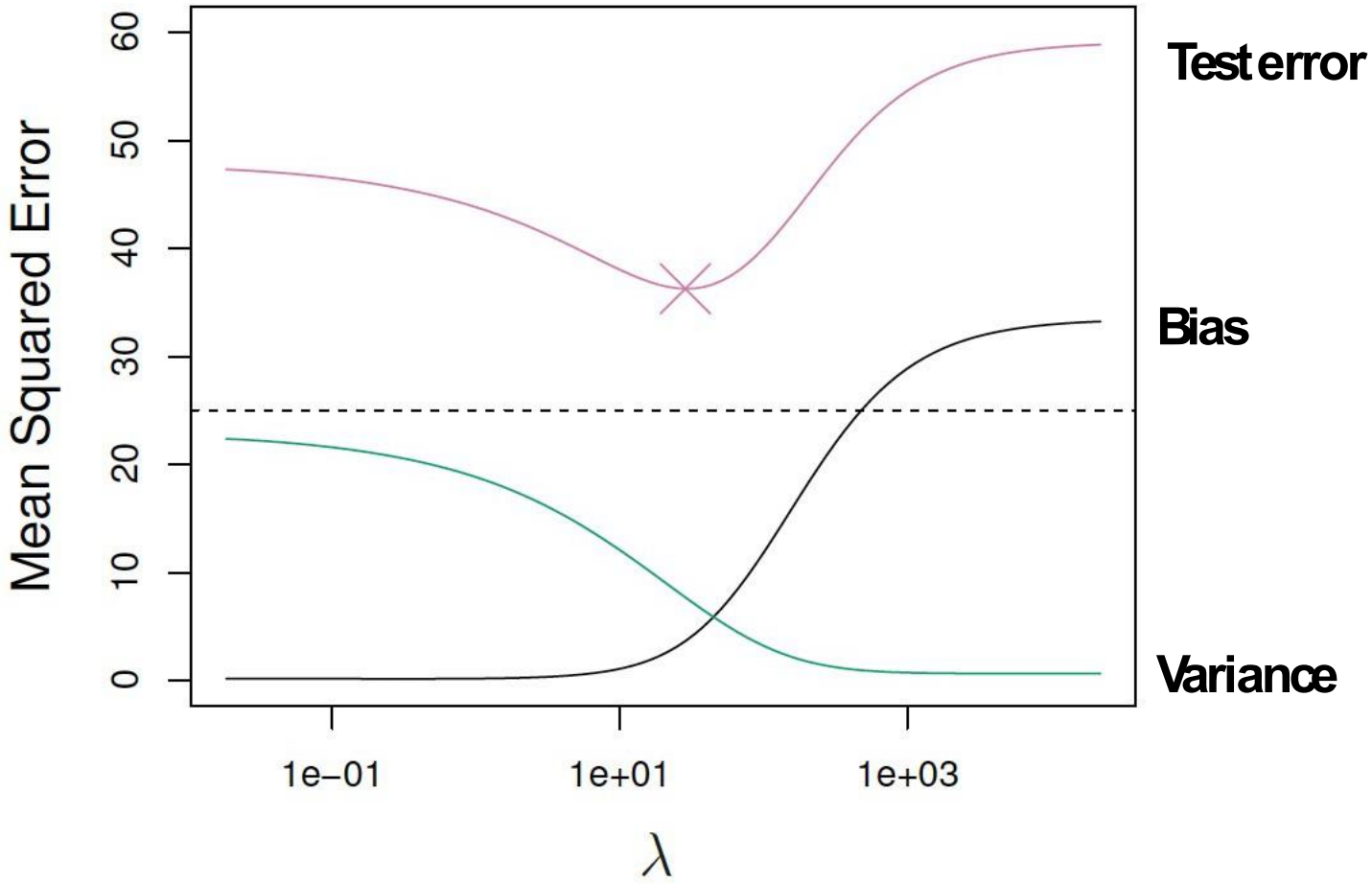


Fig 6.5 from Introduction to Statistical Learning

Any value of lambda only has a single solution, so this procedure is less computationally intensive than all subsets approach. The main package in Rfor regularization, glmnet, uses 100 different values of lambda as the default, but it will stop before that if test error is not sufficiently changing with increasing lambda.

Ridge Regression continued

- You can run ridge regression even when the number of parameters is larger than the number of observations
- You should standardize your predictor variables before running ridge regression so that they all have a standard deviation of 1
- The coefficients are never exactly zero, even when lambda is large
- It shrinks coefficients in a continuous way, but does not remove them entirely from the model
- Helping us find a good predictive model, but not simplifying the model by actually removing predictor variables

The Lasso

Very similar to Ridge but with a slightly different shrinkage penalty

When lambda is large, it will shrink coefficients to zero if the variable is not important -> removing them from the model

Lasso yields sparse models because it can fully remove variables from the model

Shrinkage Penalty

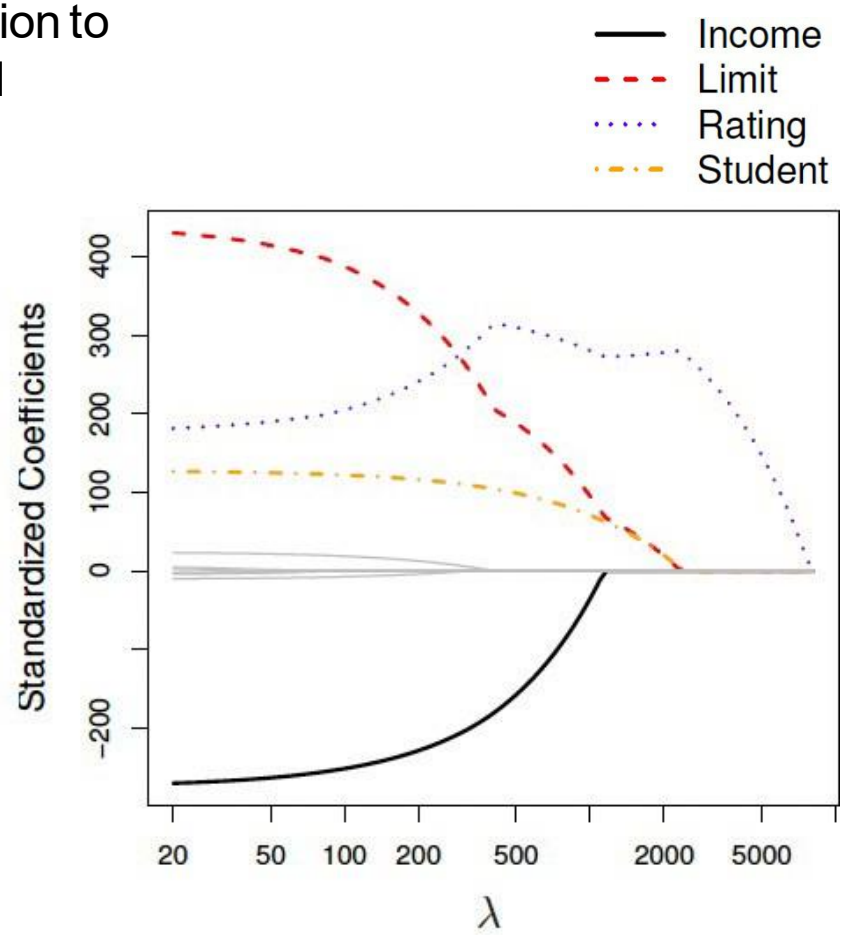
$$\text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Lambda still is the tuning parameter with same properties as for Ridge

The sum of the absolute value of the slopes

Again, this does not include the y-intercept

Fig 6.6 from Introduction to Statistical Learning

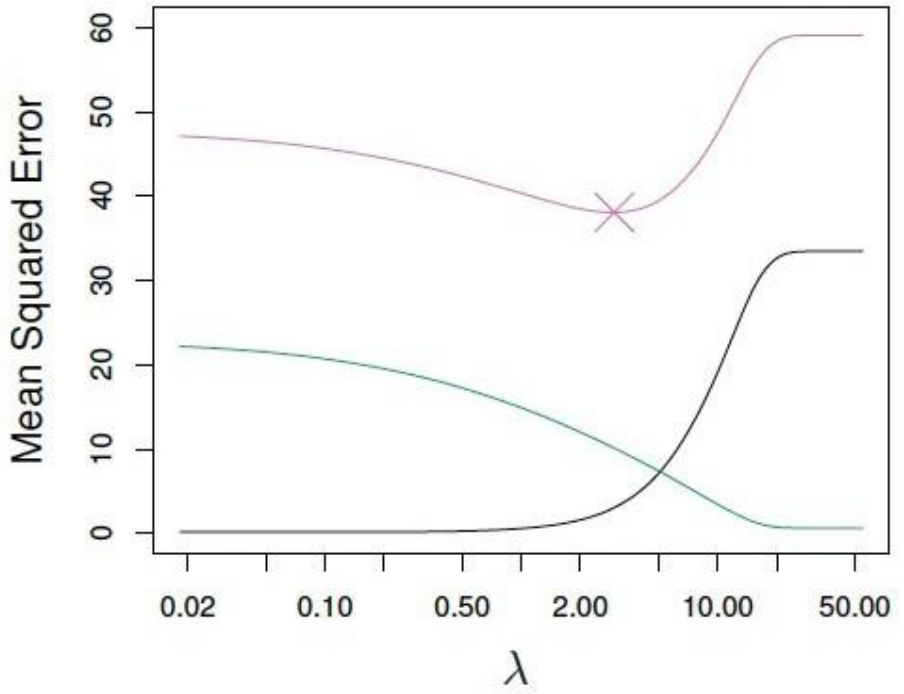


Certain variables drop out of the model when lambda is large because their slopes become 0

(Credit balance ~ Income + Limit + Rating + Student)

Note: these two plots do NOT use the same x-axis scale! (just a plot to remind us)

Fig 6.8 from Introduction to Statistical Learning



The optimal value for lambda is found using k-fold cross-validation just like for Ridge regression

Ridge vs Lasso

Ridge

Produces models that contain all the variables, some with coefficients near zero

It performs best when the response variable is related to a large number of the predictor variables with coefficients of similar sizes

Lasso

Produces simpler and more interpretable models

It performs best in situations where a few of the predictor variables have substantial coefficients (slopes) and the remaining variables have coefficients that are very small or equal to zero

The glmnet package makes it very easy to run multiple regularization methods and use cross validation to identify which is best for a particular dataset

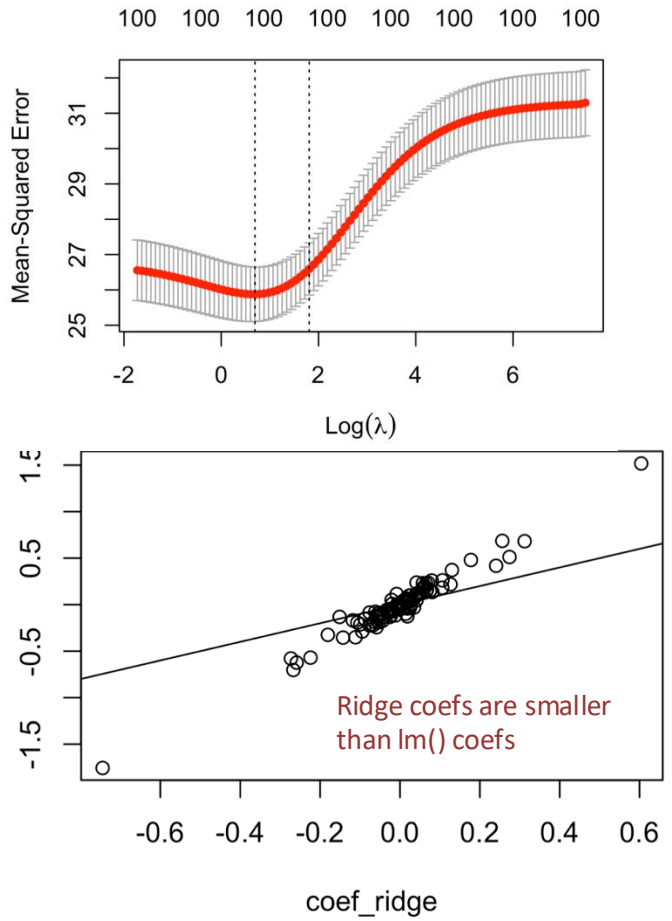
There is a 3rd method called elastic net that combines the penalties of Ridge and Lasso. Again, it's very easy to implement all 3 methods at once in R.

```
> head(y)
[1] -2.9725403  3.4945331 -0.5826752  6.1906238 -0.4704869 -8.6613093
```

```
> dim(x)
[1] 1000  100
```

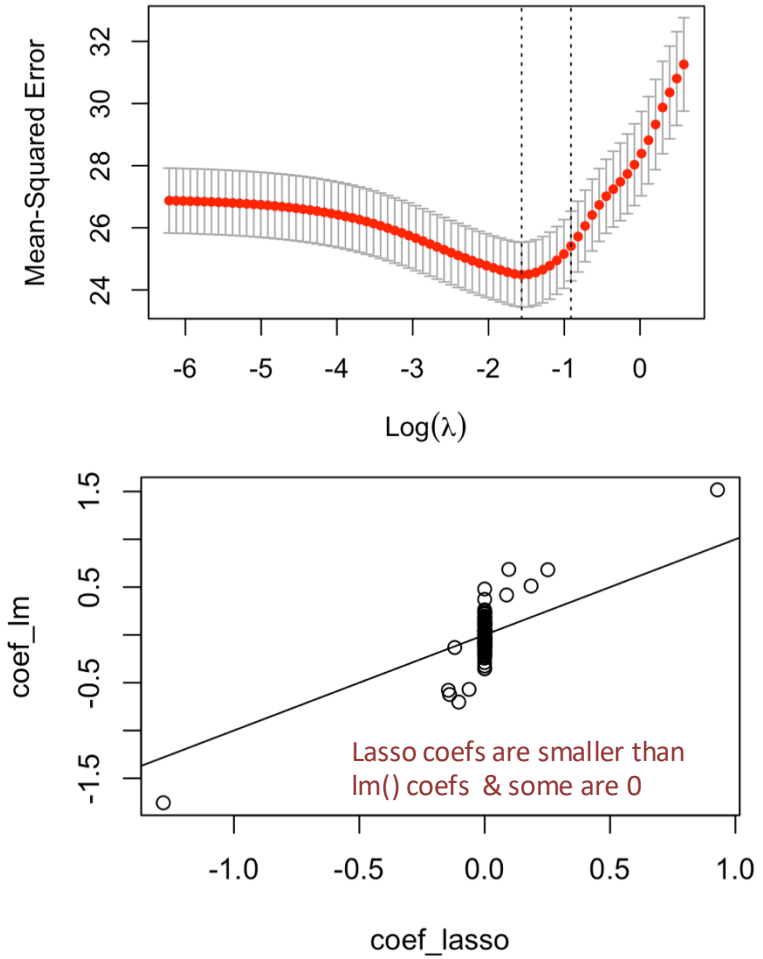
Ridge

```
cvo_ridge = cv.glmnet(x, y, alpha=0)
plot(cvo_ridge)
```



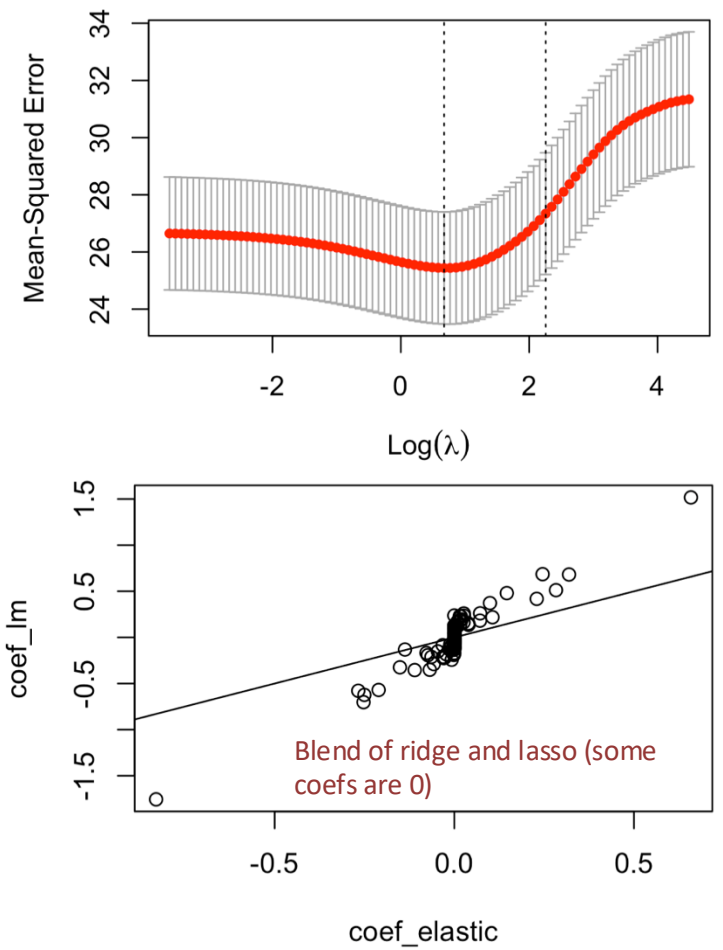
Lasso

```
cvo_lasso = cv.glmnet(x, y, alpha=1)
plot(cvo_lasso)
```



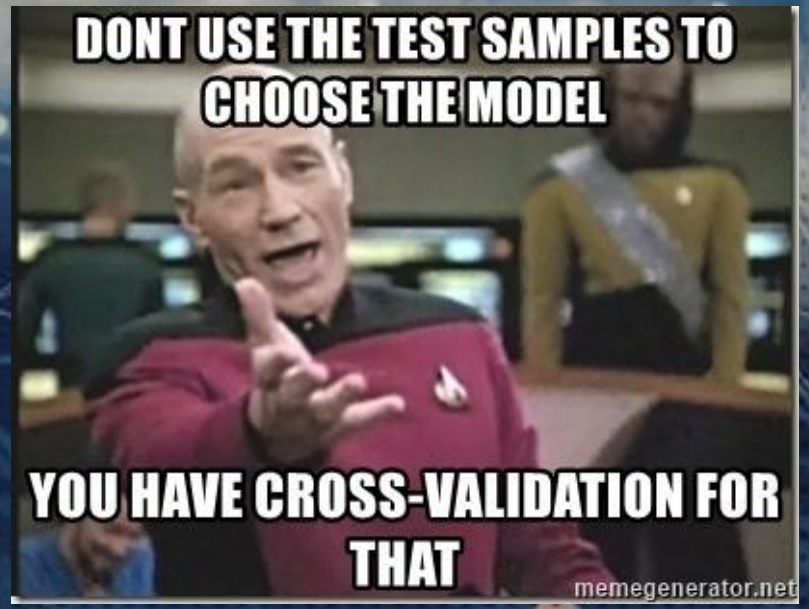
Elastic Net

```
cvo_elastic = cv.glmnet(x, y, alpha=0.02)
plot(cvo_elastic)
```



PLS206 **Applied Multivariate Modeling in Agricultural and Environmental Sciences**

Cross Validation



Training and Testing Models

We often want to know how useful the model we have constructed is for explaining our biological phenomenon of interest.

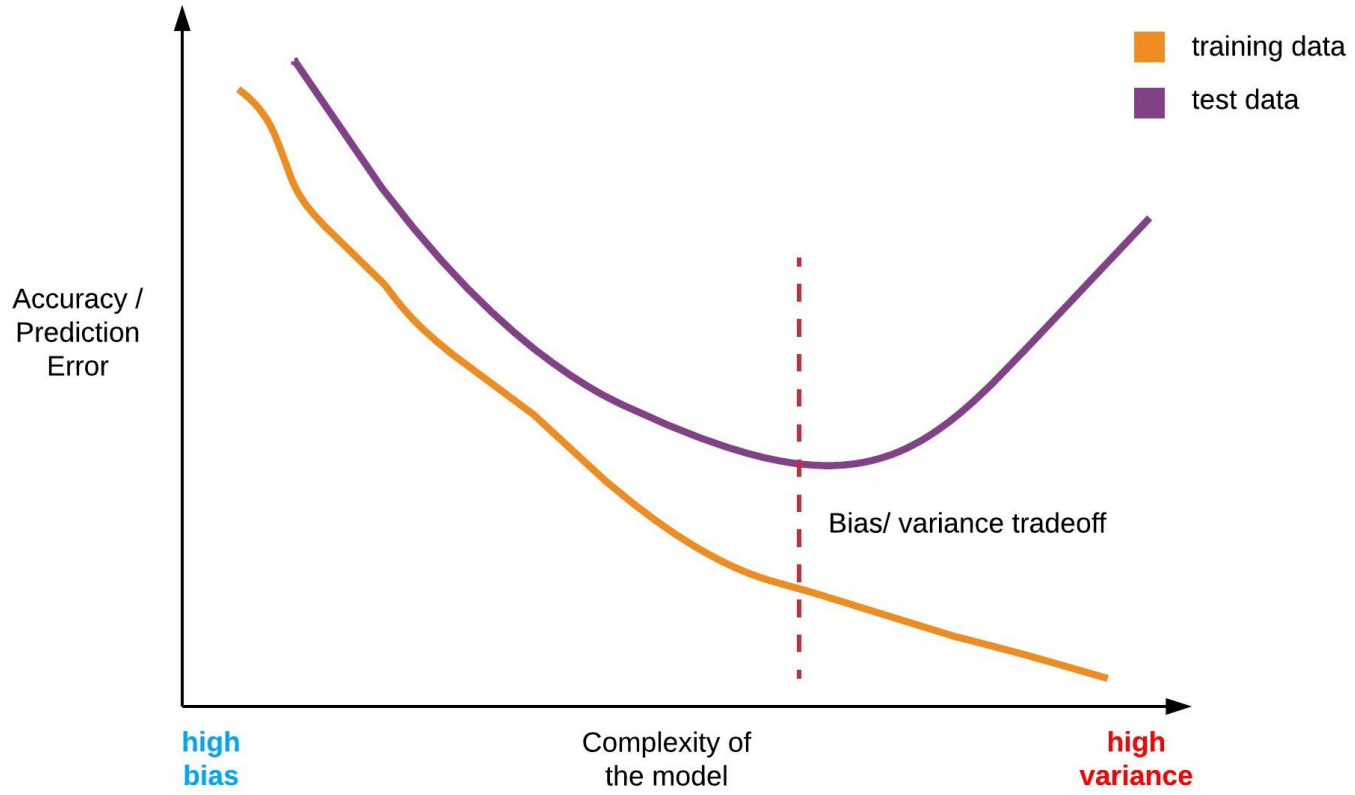
In an ideal world, we would:
Collect one set of data to construct ("train") the model
Collect another data set to evaluate ("test") the model to confirm it works

If the model is useful, the differences between the predicted values and response values in the test data (called the test error) will be low.

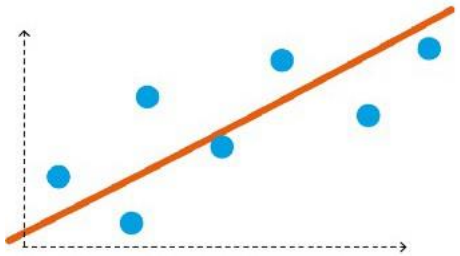
But we all know that funding cycles and the timelines for completing graduate degrees do not allow for this. So, what do we do?

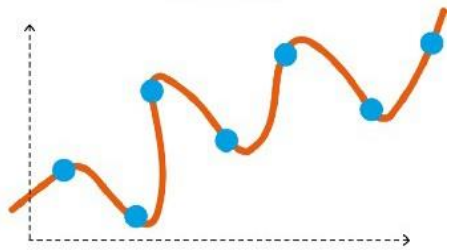
Can we use our single dataset to both test and train our model? Yes… but only if we proceed carefully. We can't train and test a model using the same data.

Variance-Bias Tradeoff



High bias
Low variance





Low bias
High variance

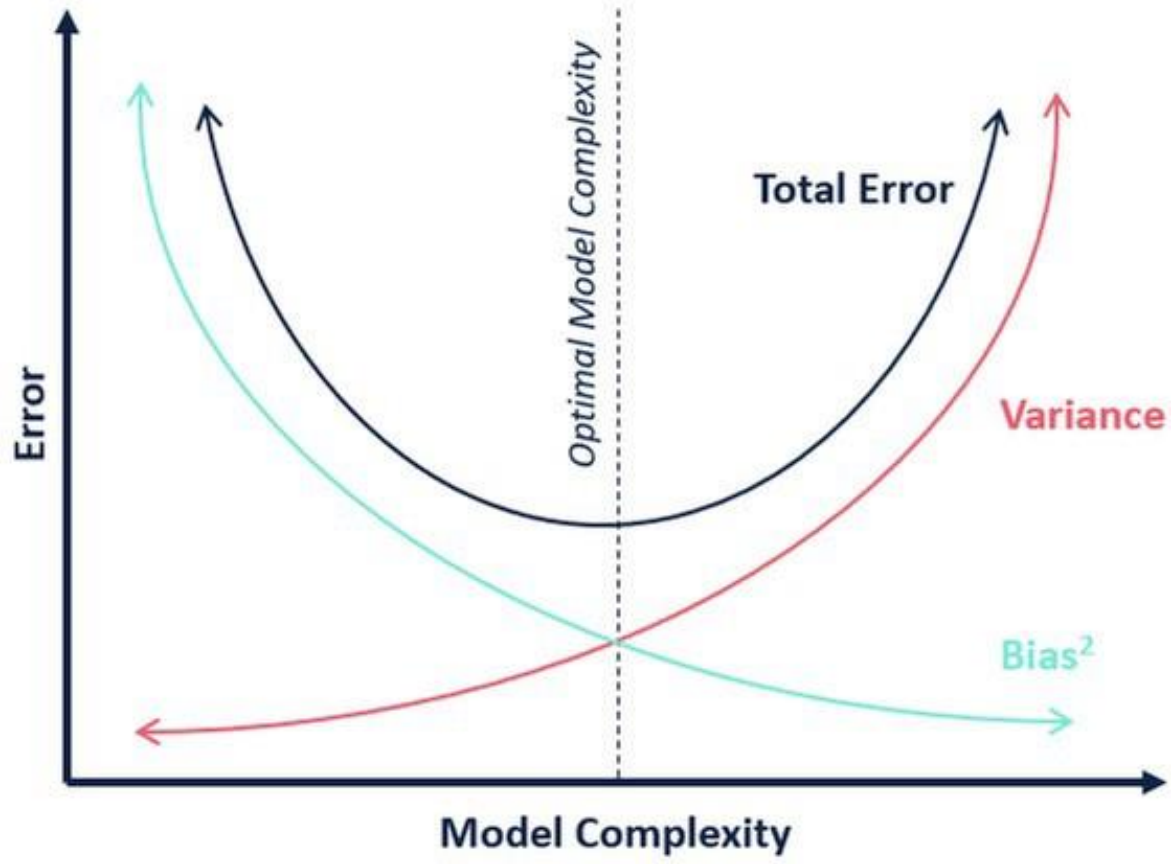
Variance-Bias Tradeoff

Bias: refers to the difference between predicted and observed values. Refers to the error that is introduced by approximating a complex real-life problem with a much simpler model.

Variance: refers to the amount by which our model would change if we estimated it using a different training set.

High bias and low variance: the model does not consider the training data enough. Model is underfit and oversimplified. Training and test error will be high.

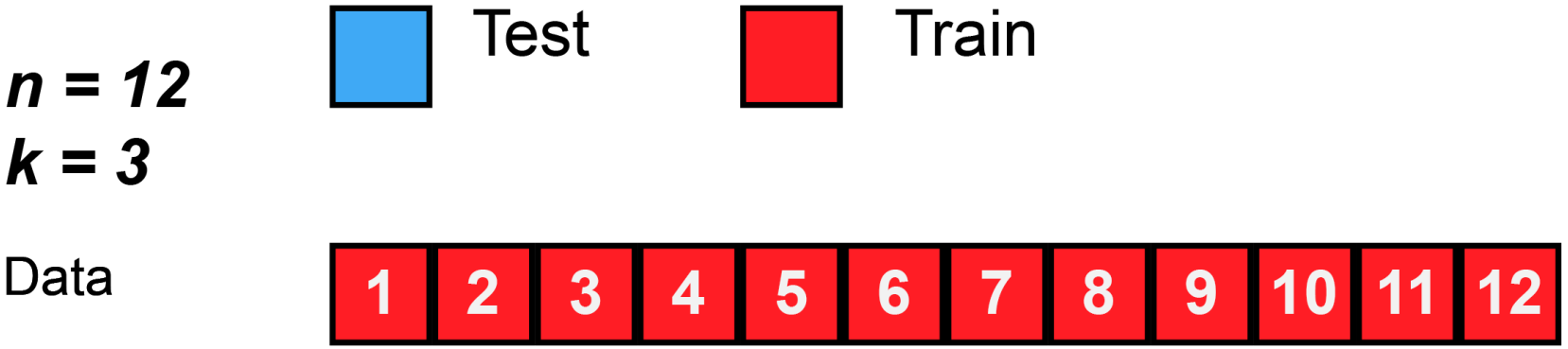
Low bias and high variance : the model follows the training data too closely. Model is overfit and too complex. Training error will be low, but test error will be high.



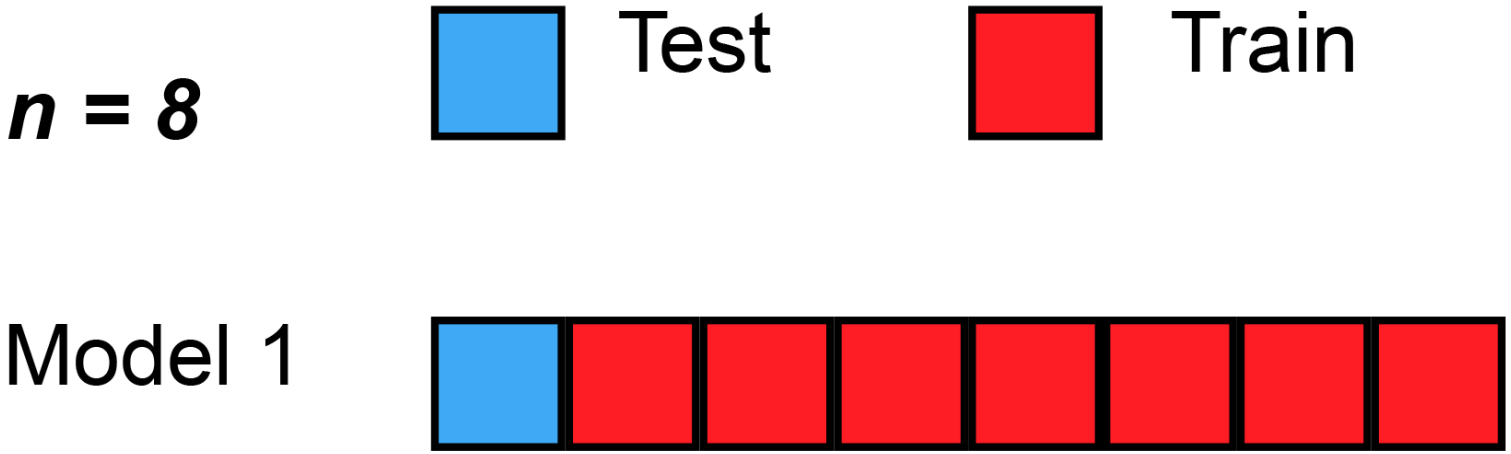
If we can't use training data to estimate test error, what can we do?

- Collect a second data set to use for testing… but we already discussed that this rarely happens
- Use resampling methods to estimate test error
  - Hold out a subset of data from the model fitting process
  - Apply the model to the held-out observations to get an estimate of the test error
  - The key is that the test and the training data must be non-overlapping!

K-fold cross validation



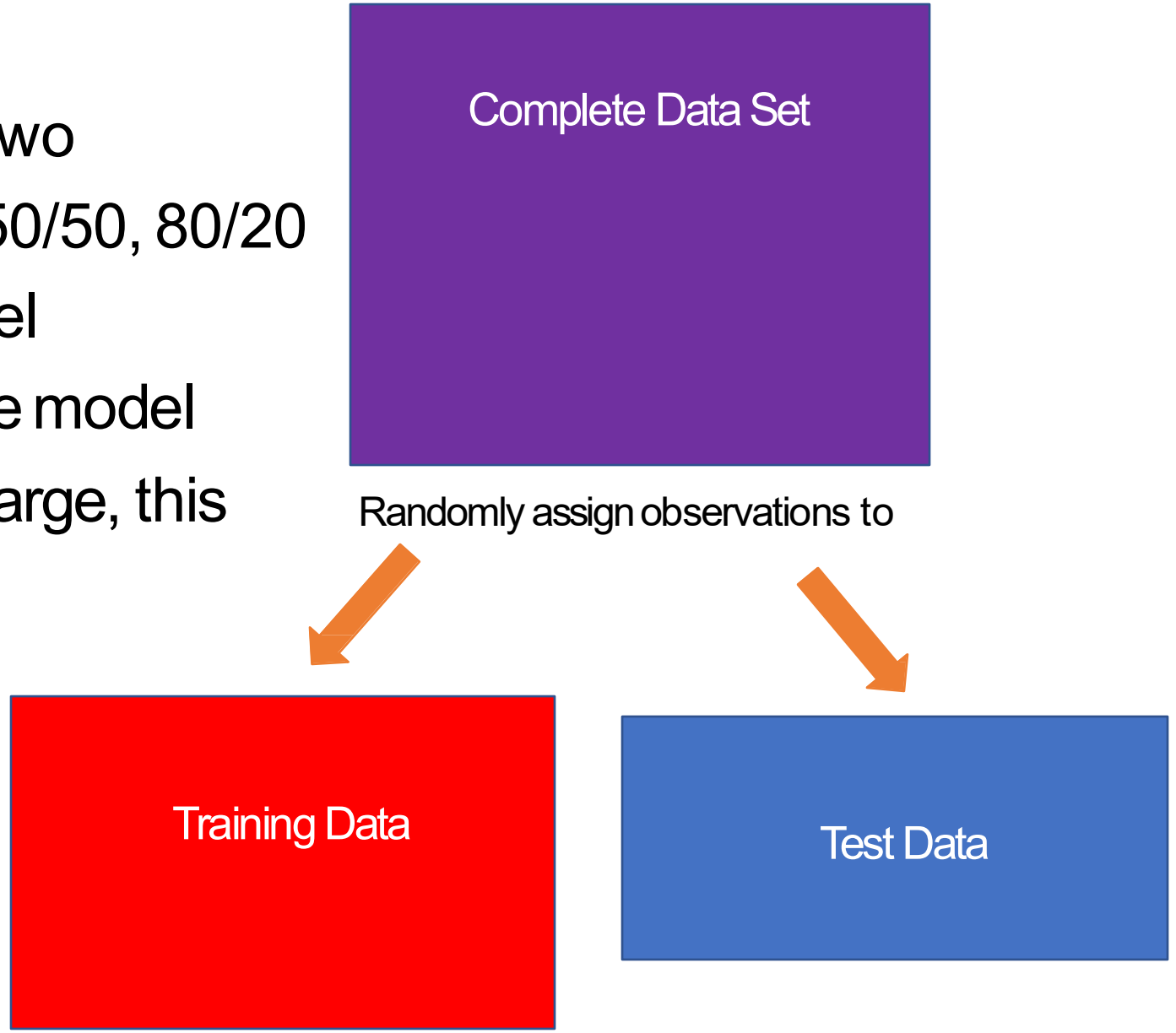
Leave one out cross validation (LOOCV)



MBanuelos22 (WIKIPEDIA)

Validation Approach

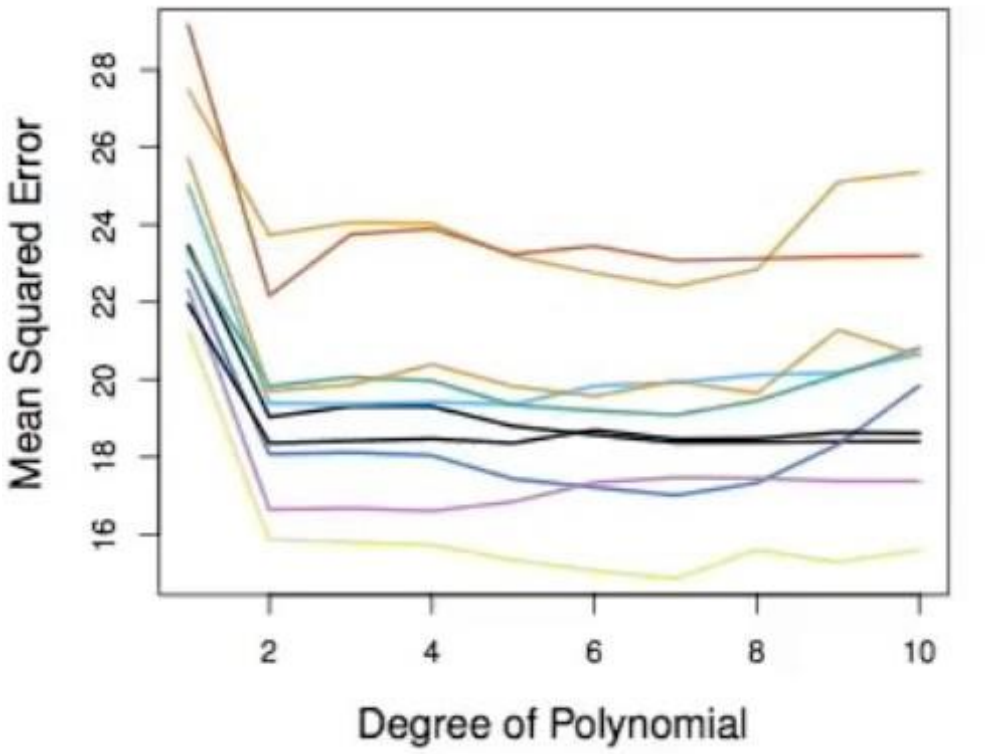
- Randomly parse your data into two
- Can split using different ratios: 50/50, 80/20
- Use one chunk to train the model
- Use the second chunk to test the model
- Unless your sample size is very large, this usually isn't the best approach



Validation Approach

- We could end up with a weird split where the training data is not representative of the distribution of values in entire data
- Training will work best when the data set is as large as it can be. The validation set approach reduces the power of the training data
- It often results in highly variable error estimates if you perform repeated simulations

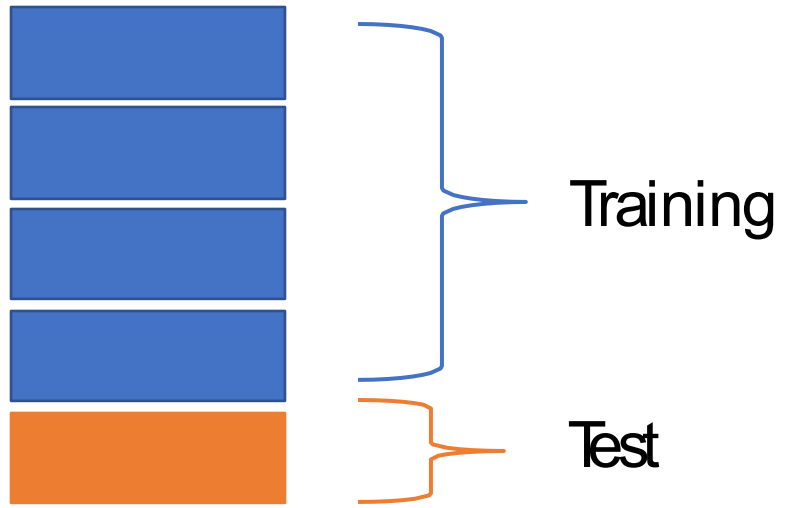
From Introduction to Statistical Learning textbook



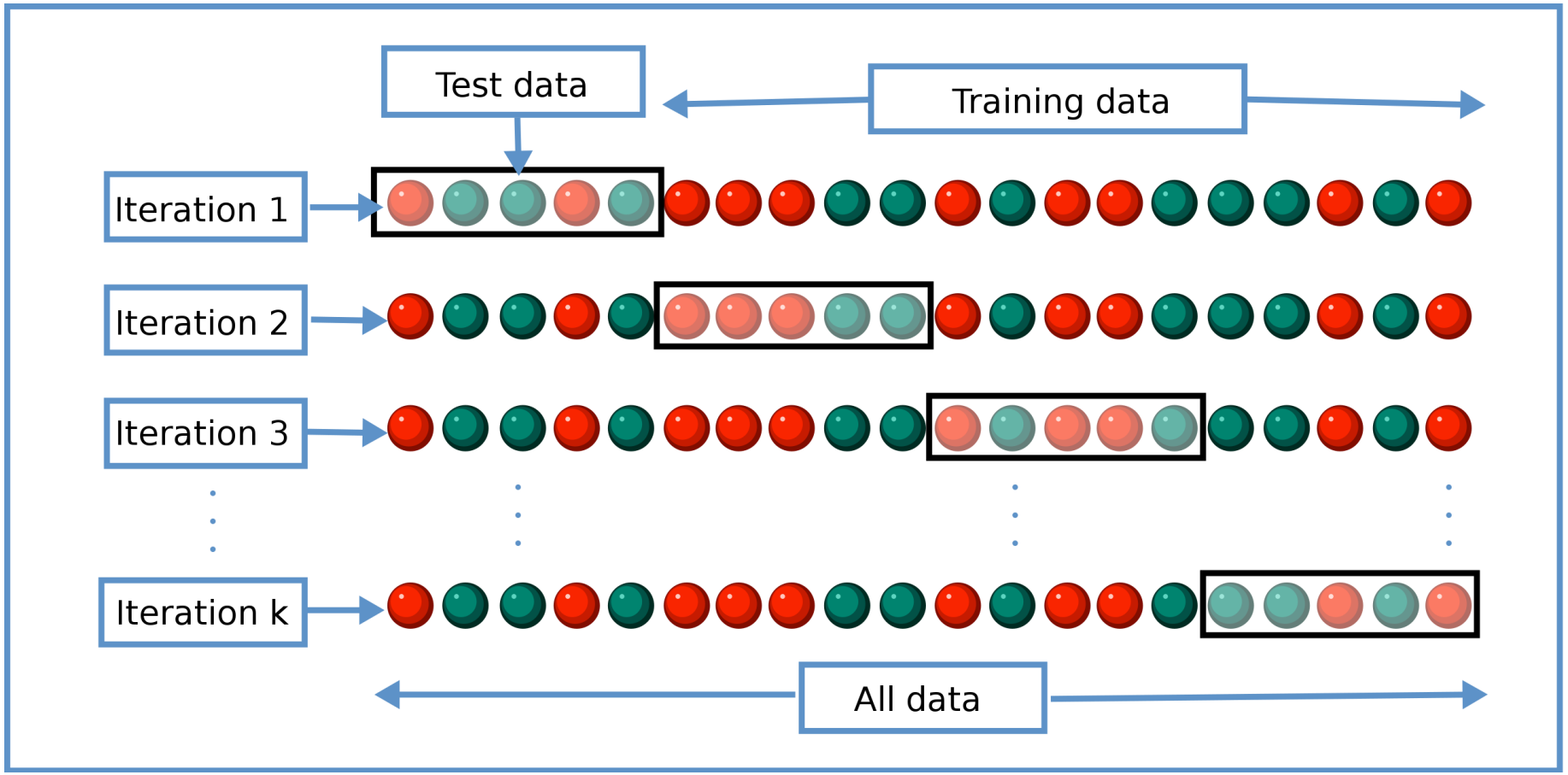
In this simulation, each color shows the performance of different splits. … we see high variability in the error estimates produced by each split

K-fold Cross Validation

- Randomly split data into Kfolds that each contain a similar number of samples
- Use K-1 folds for training the model
- Use the remaining fold for testing the model
- Repeat process using a different fold each time to test the model
- Take the average of the error estimates from each phase
- K is often set at 5 or 10
- A lot of methods in R have this process built in



K-fold cross validation

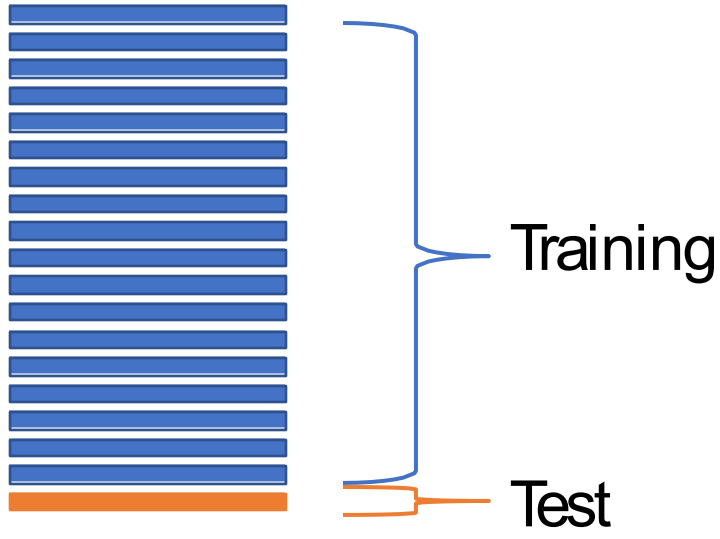


Gufosowa (WIKIPEDIA)

K-fold cross validation

Leave-one-out cross validation

- The number of folds is equal to the number of observations in the data
- n-1 observations make up each training set and a single observation acts as the test set
- The process repeats until every observation has been used as the test set
- More computationally demanding than K-fold CV
- LOOCV often doesn't shake up the data enough and most agree that K-fold is better
- Each training set is very similar b/c they only differ by 1 observation. The error estimates from each phase are highly correlated and their average can have high variance



Repeated K-Cross Validation

- Same idea as K-fold but you repeat the entire process multiple times
- For each repeat, the folds are partitioned differently
- Example: 10-fold CV with 4 repeats
- With regular k-fold CV: the folds contains the same data across each phase
- In repeated k-fold CV: the folds used for the phase of repeat 1 will contain different observations than folds used for the phases in repeat 2
- Gives a more accurate estimate of error than K-fold CV
- Which should you use? Perform regular k-fold CV. Look and see how much variability exists in the test error estimate across the folds. If the error estimates vary a lot between folds, consider using repeated K-fold CV

CV Pitfalls to Avoid

Do not do this:

- Step 1: Use data to find a set of best predictors
  - e.g. reduce from 20 predictors down to 6 predictors
- Step 2: Split your dataset into testing and training and perform CV using model with 6 best predictors
- Your model has already look at the full set of observations during step 1 to filter the predictors, which is a form of model training. You will underestimate test error.
- Imagine you have 5000 predictors. At least some of them will have correlations with the response data, even if there is no real relationship. If you cherry pick variables to find the best set and then use CV in step 2, you will estimate a very low error rate and think your model is great. Yikes!

CV Pitfalls to Avoid

Do this instead:

- Step 1: Define the folds. Remove one of the folds and set it aside
- Step 2: Create a model with best set of predictors on other folds
- Step 3: Use CV to test the model using the left-out fold

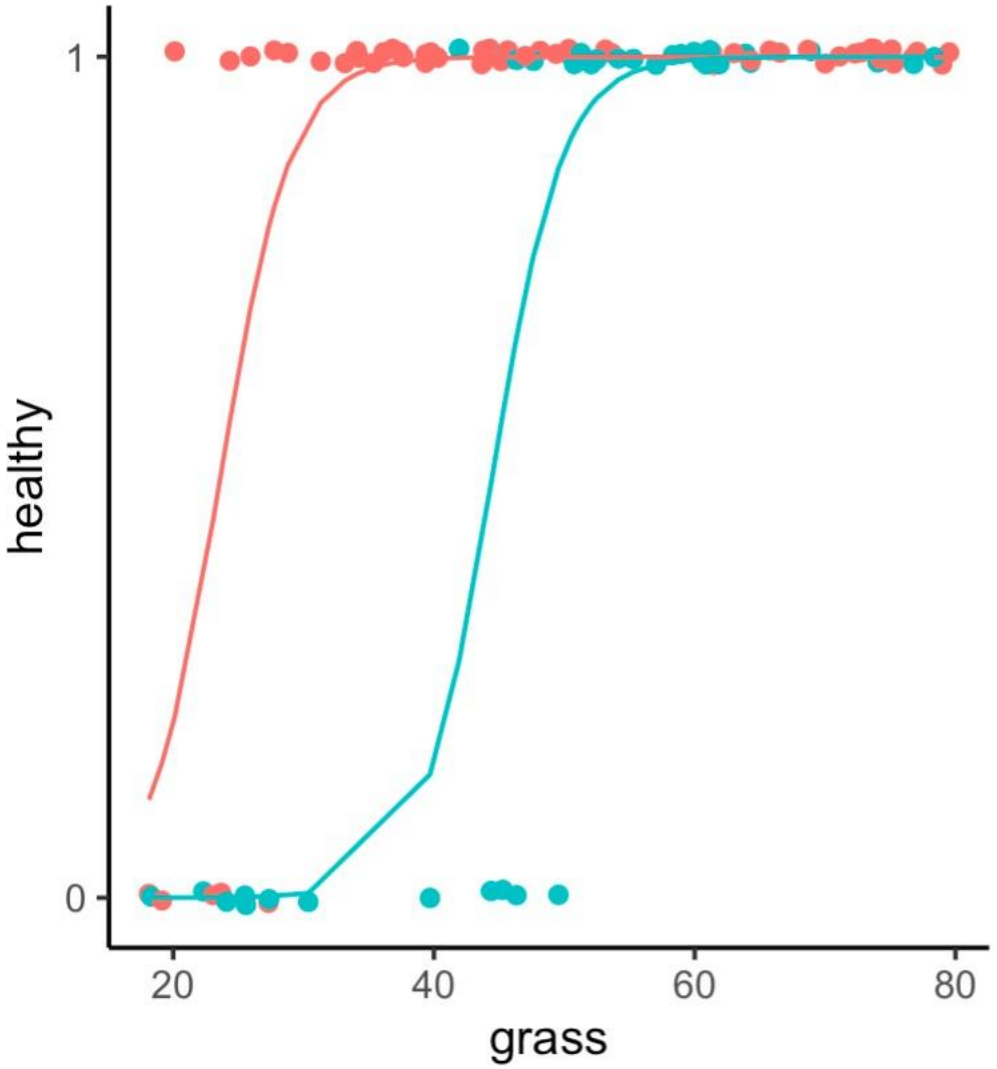
Ideally, you would repeat across multiple folds, in each phase you need to find the best set of predictors

Estimating accuracy: CV for Classification Problems

- We can use cross validation for classification problems where we are assigned our observations to different groups
- We have primarily discussed logistic regression models with a binomial response
- Evaluate the model by measuring the accuracy (if you're an optimist) or misclassification rate (if you're a pessimist)
- How many of the test data samples were correctly (or incorrectly) classified into their actual group using the model?

How well does our model predict ? (validation)

** This is totally made up data!! **



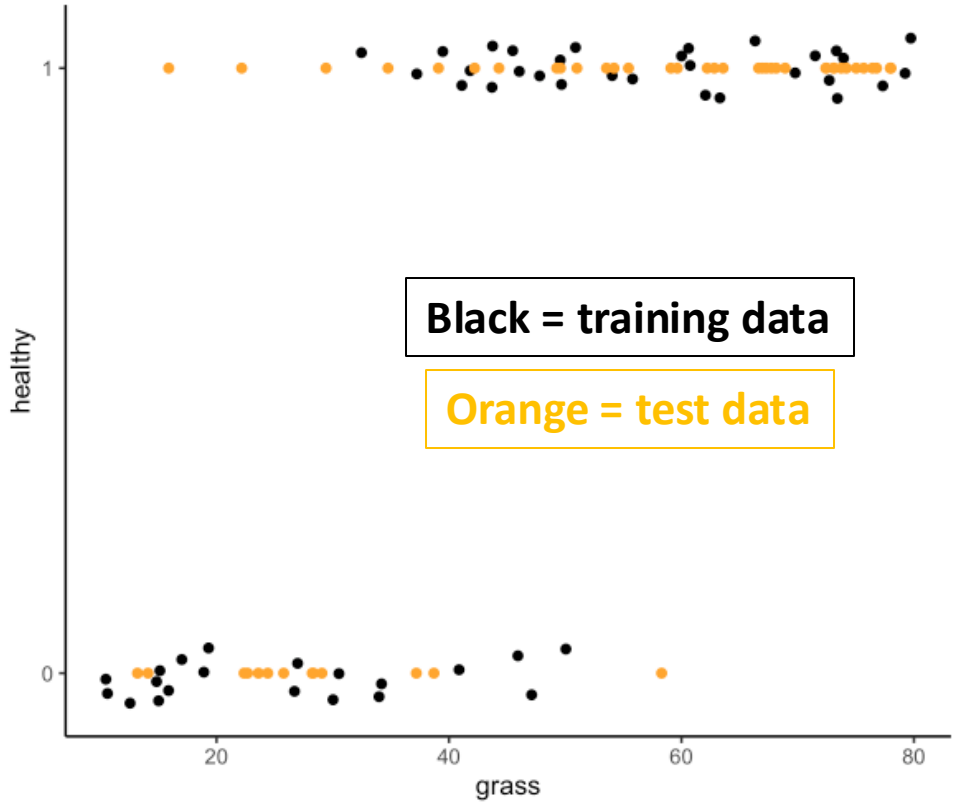
Log odds of Angus being healthy with 0% grass diet

Increase in log odds of being healthy with each %increase in grass in diet

Difference in Log odds of Hereford being healthy

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-3.81666	1.29913	-2.938	0.003305	**
grass	0.16624	0.03786	4.391	1.13e-05	***
breedHereford	-3.46904	0.95774	-3.622	0.000292	***

Example with simple validation (splitting the data in half into test and training set)



How well do we predict the test data?

Confusion matrix

```
          Reference
Prediction  0  1
         0 12  1
         1  4 33
```

45 correct and 5 incorrect = accuracy of 0.9

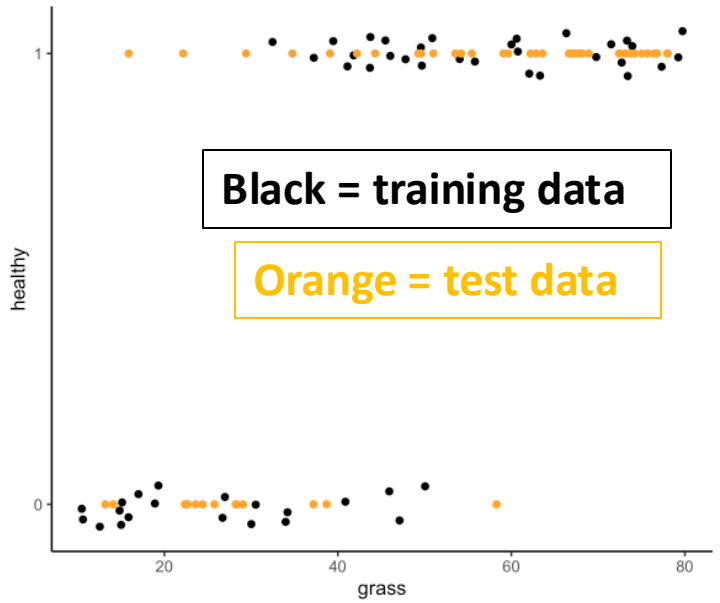
```
               Accuracy : 0.9
                 95% CI : (0.7819, 0.9667)
    No Information Rate : 0.68
    P-Value [Acc > NIR] : 0.000264

                  Kappa : 0.7582
```

Kappa compares accuracy to the accuracy expected by chance

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Reliability

	Sick	Healthy	
	TRUE POSITIVES (TP)	FALSE POSITIVES (FP)	$PVV = \frac{TP}{TP + FP}$
	FALSE NEGATIVES (FN)	TRUE NEGATIVES (TN)	$NPV = \frac{TN}{TN + FN}$
	Sensitivity $\frac{TP}{TP + FN}$	Specificity $\frac{TN}{TN + FP}$	



Black = training data

Orange = test data

How well do we predict the test data?

```
Sensitivity : 0.7500
Specificity : 0.9706
```

Model performance metrics: R^2, RMSE, and MAE

If we have a continuous response variable, we can evaluate the model using:
R2

Measures the proportion of the variance in the response variable which is explained by the model.

Ranges from 0 to 1.

The higher R2, the more closely a model predicts the actual observations.

R2 value always increases as the number of predictors increases, so we often look at Adjusted R2 which accounts for the number of predictor variables in the model.

While easy to interpret, RMSE and MAE are usually preferred over R2

Model performance metrics: R^2, RMSE, and MAE

If we have a continuous response variable, we can evaluate the model using:

Root Mean Square Error (RMSE)

Measures the average difference between the predictions made by the model and the actual observations.

The lower the RMSE, the more closely a model predicts the actual observations.

Due to the squaring operation, very small values (between 0 and 1) become even smaller, and larger values become even larger. This means that big error values are magnified, whereas small ones are ignored. RMSE is sensitive to outliers.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Model performance metrics: R^2, RMSE, and MAE

If we have a continuous response variable, we can evaluate the model using:

Mean Absolute Error (MAE)

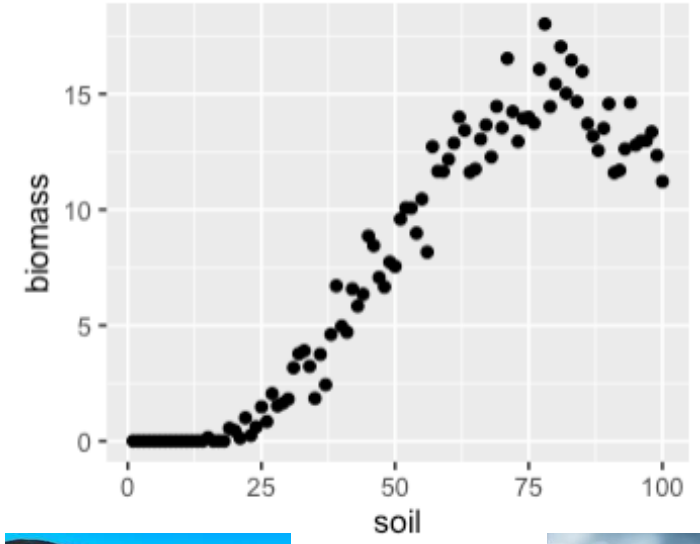
Measures the average absolute difference between the predictions made by the model and the actual observations.

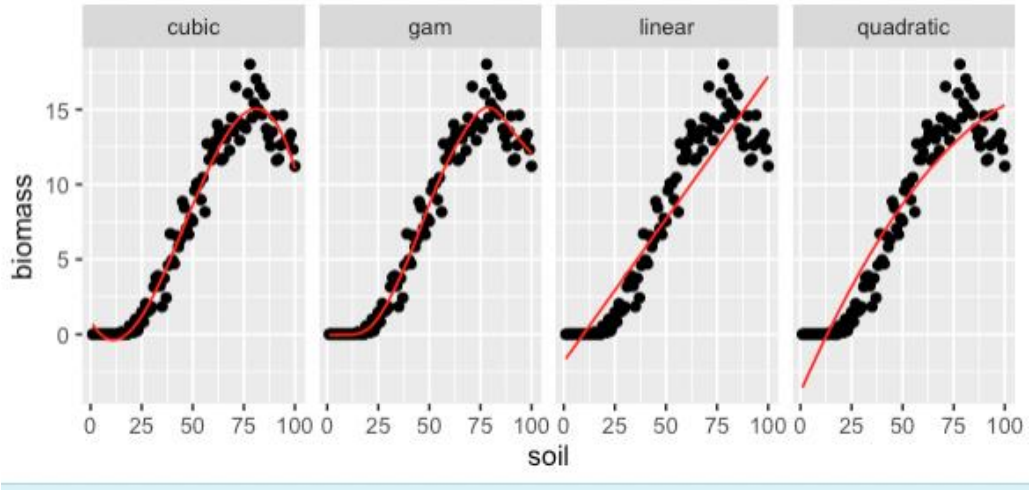
The lower the MAE, the more closely a model can predict the actual observations.

The contribution of individual data points to MAE follows a linear behavior. This means that an error of 10 contributes twice as much as an error of 5. An error of 1000 contributes 10 times as much as an error of 100.

MAE is more robust to data with outliers than RMSE

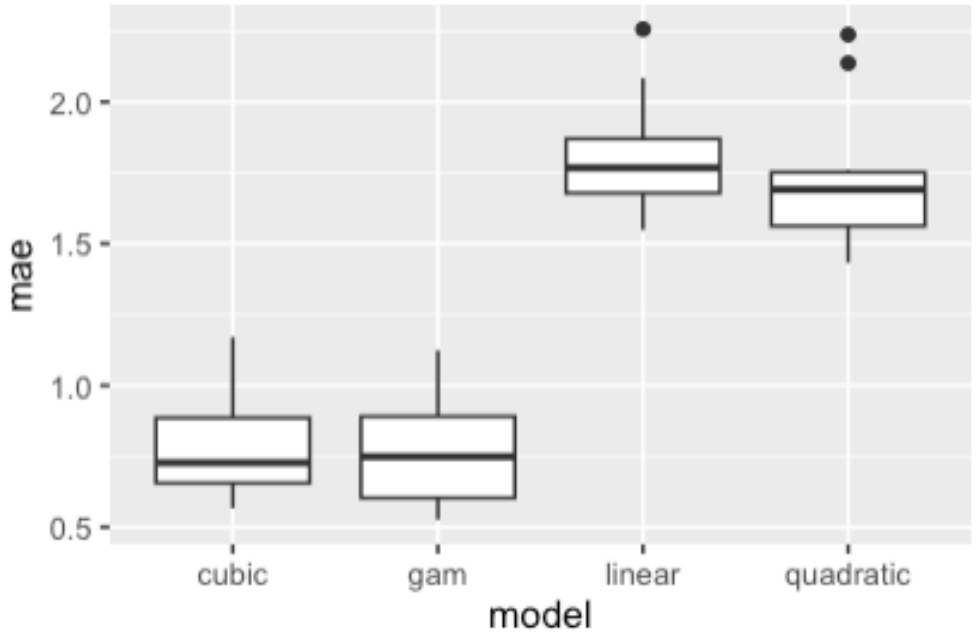
$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$





5-fold CV

Results from each iteration summarized



CV for Model Selection

- Will the model I have produced give accurate predictions on future data?
- Even if your primary goal is hypothesis testing, you should still care if your model is useful for prediction. If your model has high prediction error, is it adequately explaining any biological phenomenon?
- Use cross validation to find the RMSE, MAE or accuracy for several models. Pick the model with best value.
  - lowest RMSE or MAE
  - highest accuracy (or lowest misclassification rate)
- Cross validation is general. We can compare many different models this way.

PLS206 Applied Multivariate Modeling in Agricultural and Environmental Sciences

Regularization:
- Ridge regression
- The Lasso

Cross validation:
- "Validation"
- K-fold CV
- Leave one out (LOOCV)
- Measuring Accuracy
  - Classification
  - Continuous response
- Selecting models