Building Regression Models for Prediction

Book Chapter 6.

Jared S. Murray
The University of Texas McCombs School of Business

1. Variable Selection and Regularization
2. Dimension Reduction Methods

Model Building Process

When building a regression model remember that simplicity is your friend... smaller models are easier to interpret and have fewer unknown parameters to be estimated.

Keep in mind that every additional parameter represents a cost!!

The first step of every model building exercise is the selection of the the universe of variables to be potentially used. This task is entirely solved through you experience and context specific knowledge...

- Think carefully about the problem
- Consult subject matter research and experts
- Avoid the mistake of selecting too many variables

Model Building Process

With a universe of variables in hand, the goal now is to select the model. Why not include all the variables in?

Big models tend to over-fit and find features that are specific to the data in hand... ie, not generalizable relationships.
The results are bad predictions and bad science!

In addition, bigger models have more parameters and potentially more uncertainty about everything we are trying to learn...

We need a strategy to build a model in ways that accounts for the trade-off between fitting the data and the uncertainty associated with the model

1. Variable Selection and Regularization

When working with linear regression models where the number of X variables is large, we need to think about strategies to select what variables to use...

We will focus on 3 ideas:

- Subset Selection
- Shrinkage
- Dimension Reduction

Subset Selection

The idea here is very simple: fit as many models as you can and compare their performance based on some criteria!

Issues:

- How many possible models? Total number of models $= 2^p$
  Is this large?
- What criteria to use?
  Just as before, if prediction is what we have in mind, out-of-sample predictive ability should be the criteria

Information Criteria

Another way to evaluate a model is to use Information Criteria metrics which attempt to quantify how well our model would have predicted the data (regardless of what you've estimated for the β_j's).

A good alternative is the BIC: Bayes Information Criterion, which is based on a "Bayesian" philosophy of statistics.

$$BIC = n \log(s^2) + p \log(n)$$

You want to choose the model that leads to minimum BIC.

Information Criteria

One nice thing about the BIC is that you can interpret it in terms of model probabilities. Given a list of possible models $\{M_1, M_2, \ldots, M_R\}$, the probability that model i is correct is

$$P(M_i) \approx \frac{e^{-\frac{1}{2}BIC(M_i)}}{\sum_{r=1}^{R} e^{-\frac{1}{2}BIC(M_r)}} = \frac{e^{-\frac{1}{2}[BIC(M_i)-BIC_{min}]}}{\sum_{r=1}^{R} e^{-\frac{1}{2}[BIC(M_r)-BIC_{min}]}}$$

(Subtract $BIC_{min} = \min\{BIC(M_1) \ldots BIC(M_R)\}$ for numerical stability.)

Similar, alternative criteria include AIC, C_p, adjusted R^2...
bottom line: these are only useful if we lack the ability to compare models based on their out-of-sample predictive ability!!!

Search Strategies: Stepwise Regression

One computational approach to build a regression model step-by-step is "stepwise regression" There are 3 options:

- Forward: adds one variable at the time until no remaining variable makes a significant contribution (or meet a certain criteria... could be out of sample prediction)
- Backwards: starts will all possible variables and removes one at the time until further deletions would do more harm them good
- Stepwise: just like the forward procedure but allows for deletions at each step

Shrinkage Methods

An alternative way to deal with selection is to work with all p predictors at once while placing a constraint on the size of the estimated coefficients

This idea is a regularization technique that reduces the variability of the estimates and tend to lead to better predictions.

The hope is that by having the constraint in place, the estimation procedure will be able to focus on "the important β's"

Ridge Regression

Ridge Regression is a modification of the least squares criteria that minimizes (as a function of β's)

$$\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

for some value of $\lambda > 0$

- The "blue" part of the equation is the traditional objective function of LS
- The "red" part is the shrinkage penalty, ie, something that makes costly to have big values for β

Ridge Regression

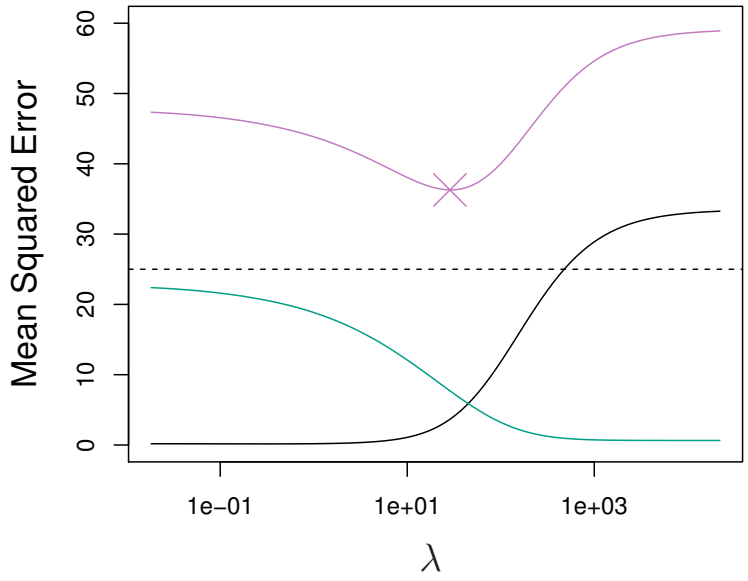
$$\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

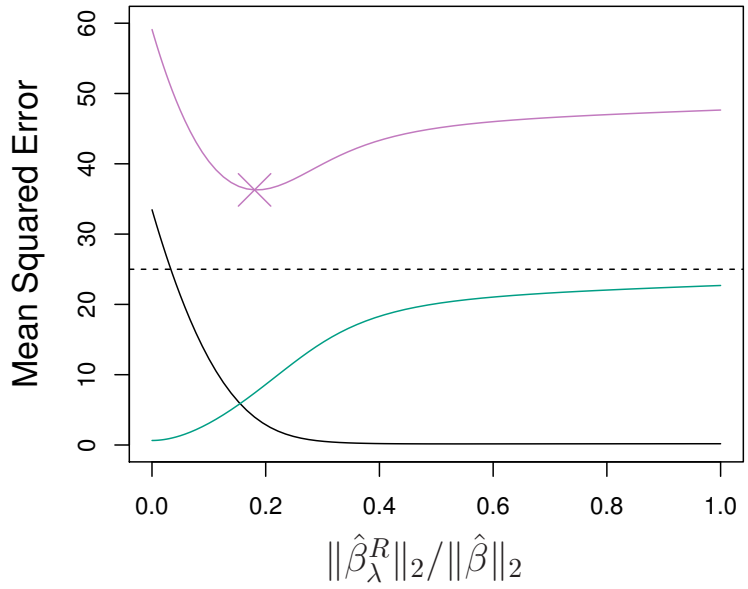
- if $\lambda = 0$ we are back to least squares
- when $\lambda \to \infty$, it is "too expensive" to allow for any β to be different than 0...
- So, for different values of λ we get a different solution to the problem

Ridge Regression

- What ridge regression is doing is exploring the bias-variance trade-off! The larger the λ the more bias (towards zero) is being introduced in the solution, ie, the less flexible the model becomes... at the same time, the solution has less variance
- As always, the trick to find the "right" value of λ that makes the model not too simple but not too complex!
- Whenever possible, we will choose λ by comparing the out-of-sample performance (usually via cross-validation)

Ridge Regression





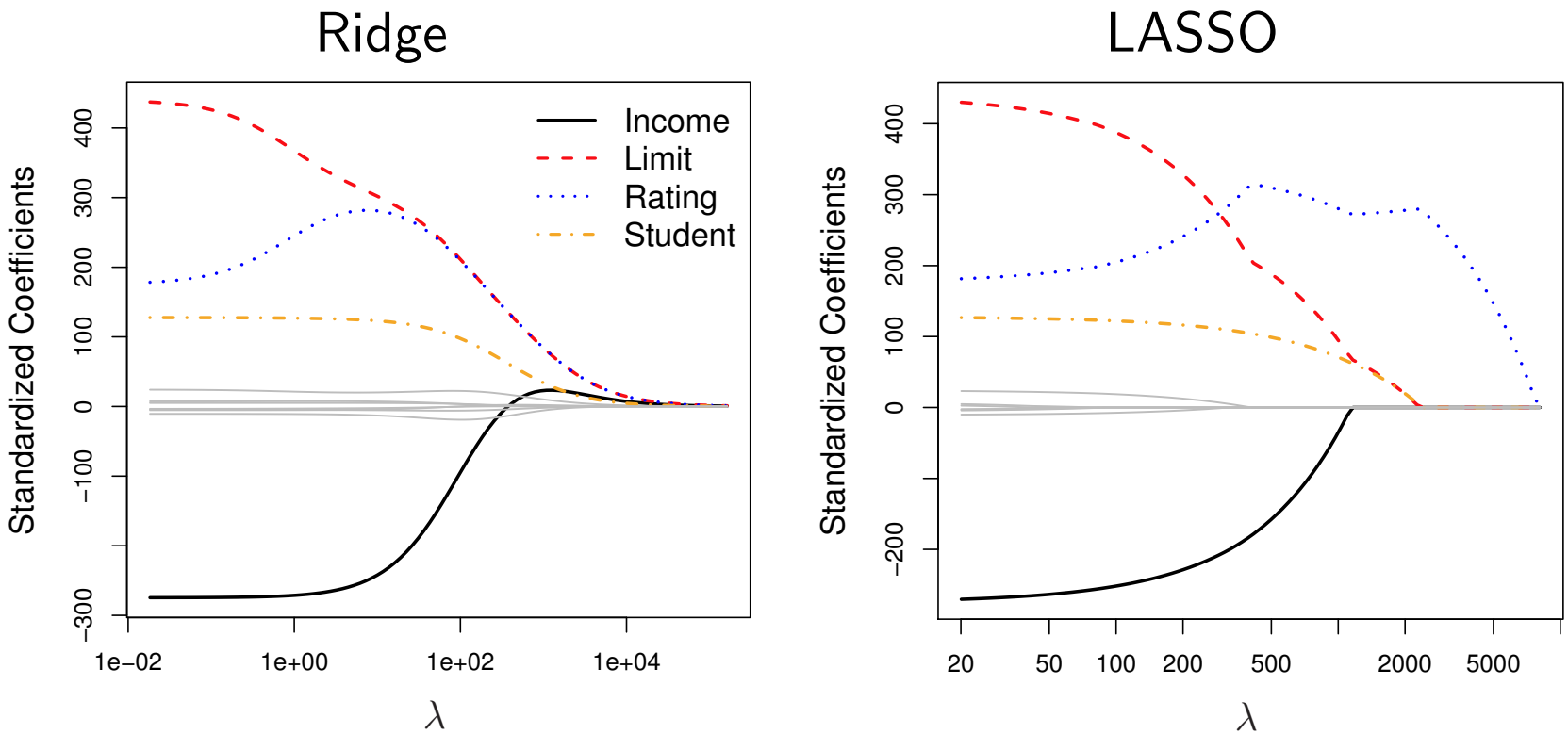
$bias^2$ (black), var(green), test MSE (purple)

Comments:

- Ridge is computationally very attractive as the "computing cost" is almost the same of least squares (contrast that with subset selection!)
- It's a good practice to always center and scale the X's before running ridge

LASSO

The LASSO is a shrinkage method that performs automatic selection. It is similar to ridge but it will provide solutions that are sparse, ie, some β's exactly equal to 0! This facilitates interpretation of the results...



LASSO

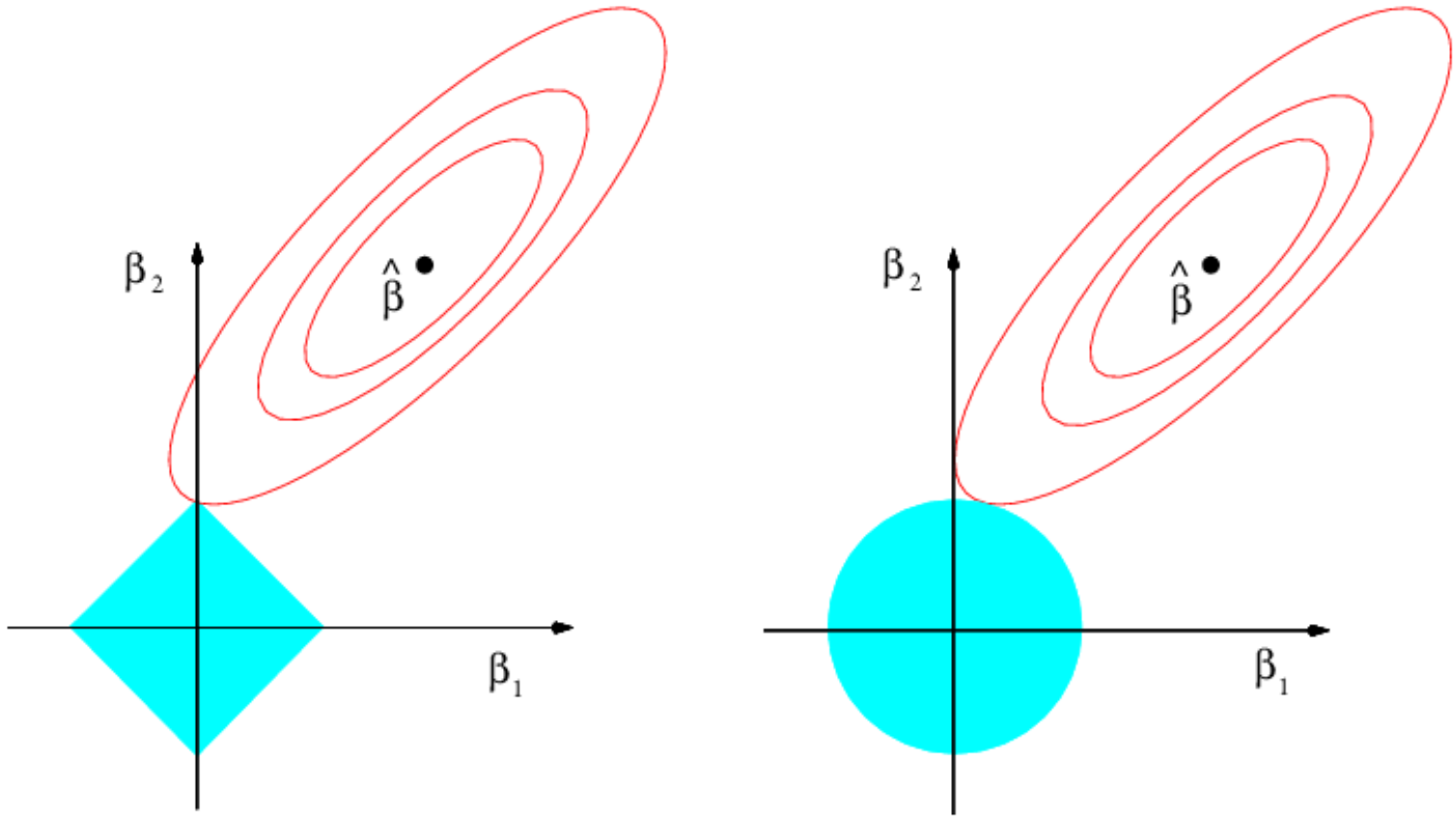
The LASSO solves the following problem:

$$\arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

- Once again, λ controls how flexible the model gets to be
- Still a very efficient computational strategy
- Whenever possible, we will choose λ by comparing the out-of-sample performance (usually via cross-validation)

Ridge vs. LASSO

Why does the LASSO output zeros?



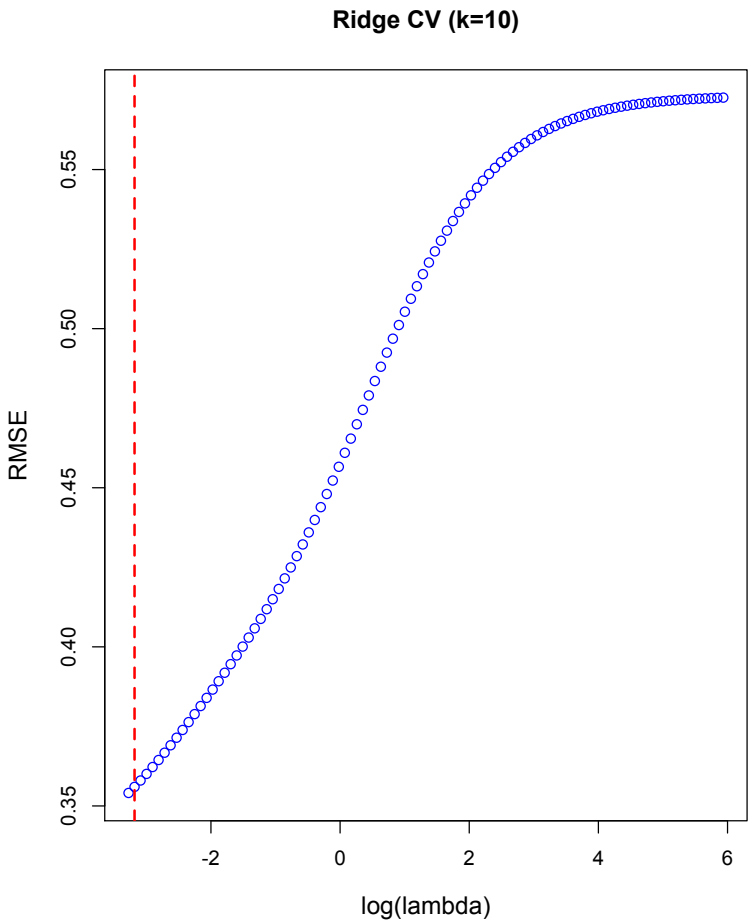
Ridge vs. LASSO

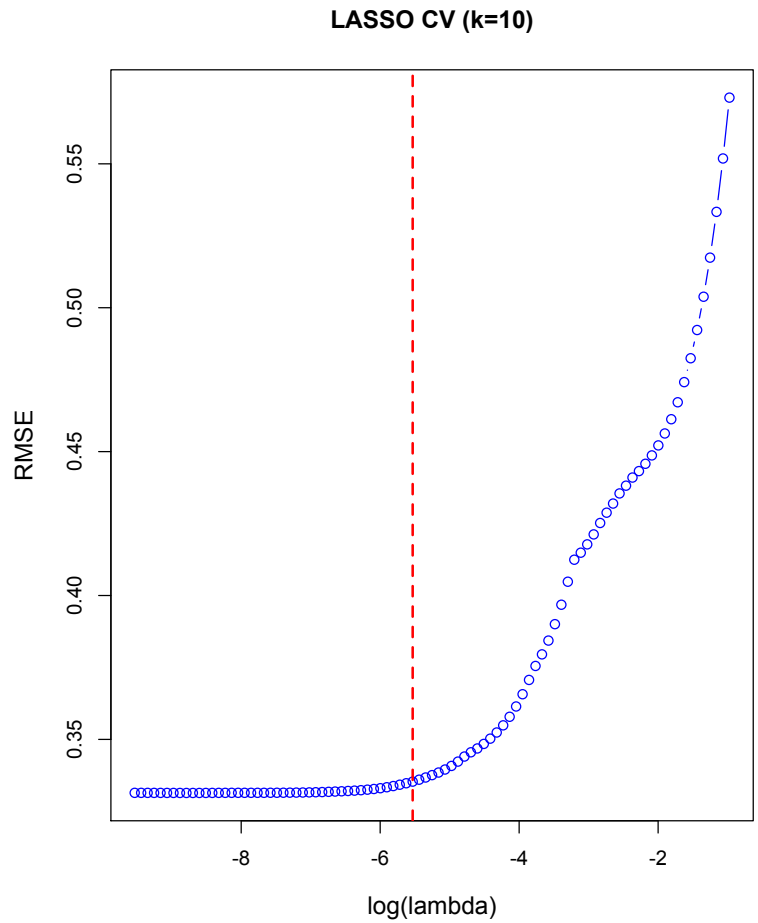
Which one is better?

- It depends...
- In general LASSO will perform better than Ridge when a relative small number of predictors have a strong effect in Y while Ridge will do better when Y is a function of many of the X's and the coefficients are of moderate size
- LASSO can be easier to interpret (the zeros help!)
- But, if prediction is what we care about the only way to decide which method is better is comparing their out-of-sample performance

Choosing λ: California Housing Data

The idea is to solve the ridge or LASSO objective function over a grid of possible values for λ...





2. Dimension Reduction Methods

Sometimes, the number (p) of X variables available is too large for us to work with the methods presented above.

Perhaps, we could first *summarize* the information in the predictors into a smaller set of variables ($m << p$) and then try to predict Y.

In general, these summaries are often linear combinations of the original variables.

Principal Components Regression

A very popular way to summarize multivariate data is Principal Components Analysis (PCA).

PCA is a **dimensionality reduction** technique that tries to represent p variables with a $k < p$ "new" variables.

These "new" variables are create by linear combinations of the original variables and the hope is that a small number of them are able to effectively represent what is going on in the original data.

Principal Components Analysis

Assume we have a dataset where p variables are observed. Let X_i be the i^{th} observation of the p-dimensional vector X. PCA writes:

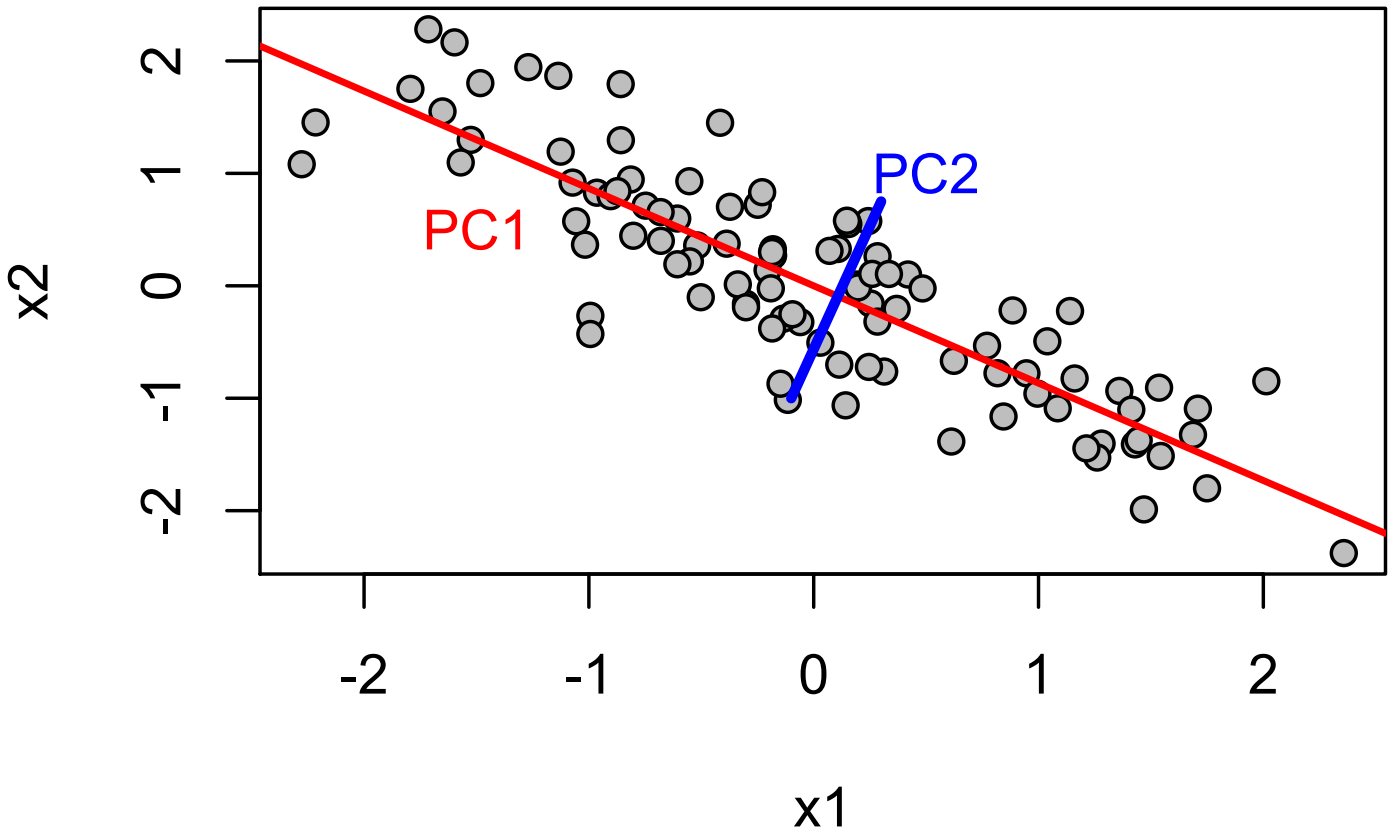
$$x_{ij} = b_{1j}z_{i1} + b_{2j}z_{i2} + \cdots + b_{kj}z_{ik} + e_{ij}$$

where z_{ij} is the i^{th} observation of the j^{th} principal component.

You can think about these z variables as the "essential variables" responsible for all the action in X.

Principal Components Analysis

Here's a picture...

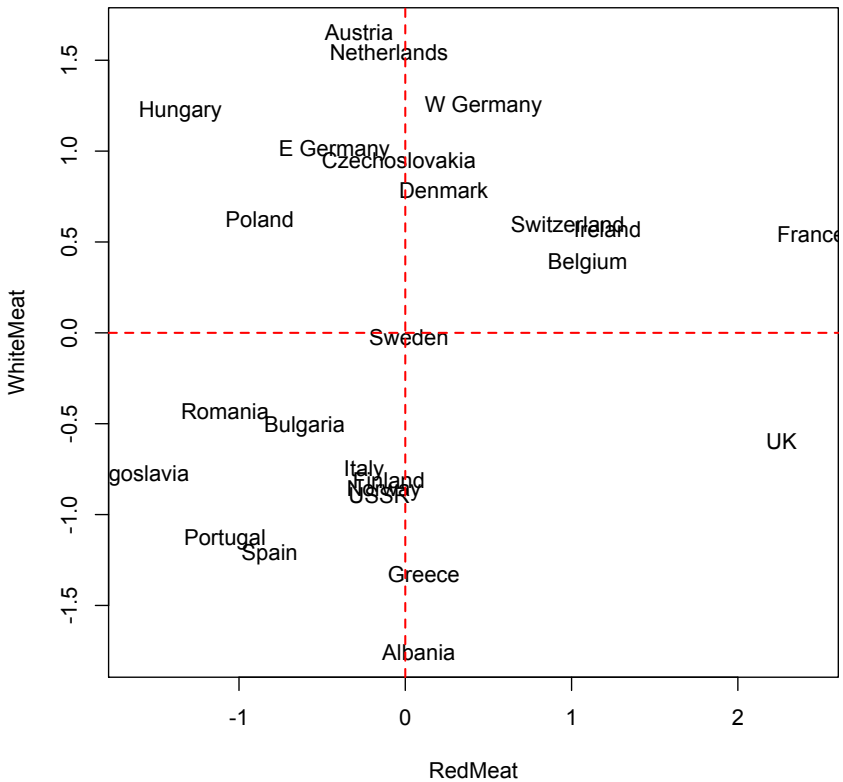


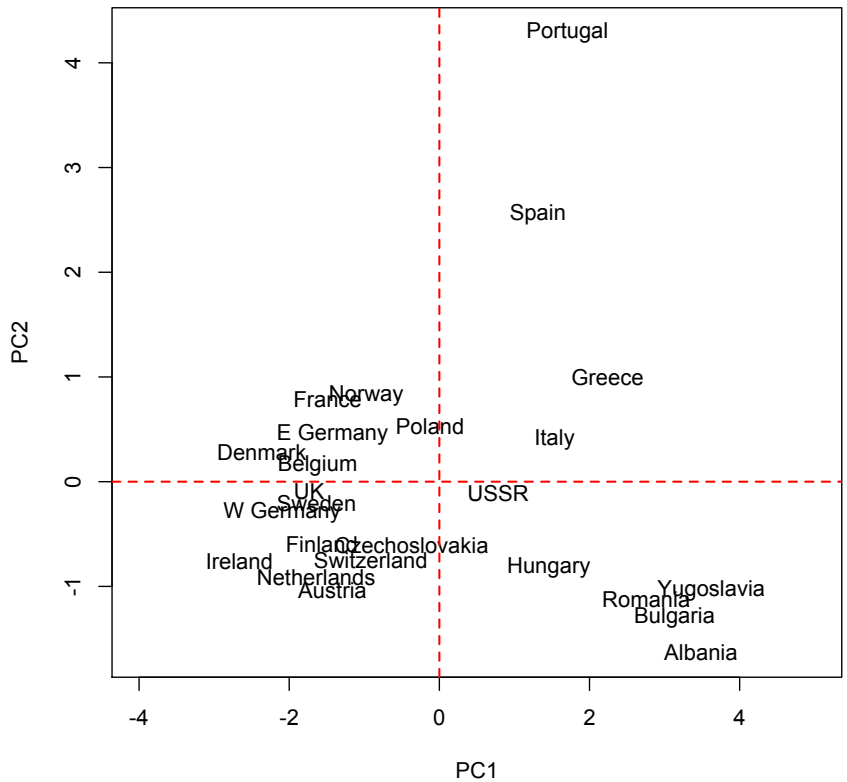
These two variables x_1 and x_2 are very correlated. $PC1$ tells you almost everything that is going on in this dataset!

PCA will look for linear combinations of the original variables that account for most of their variability!

Principal Components Analysis

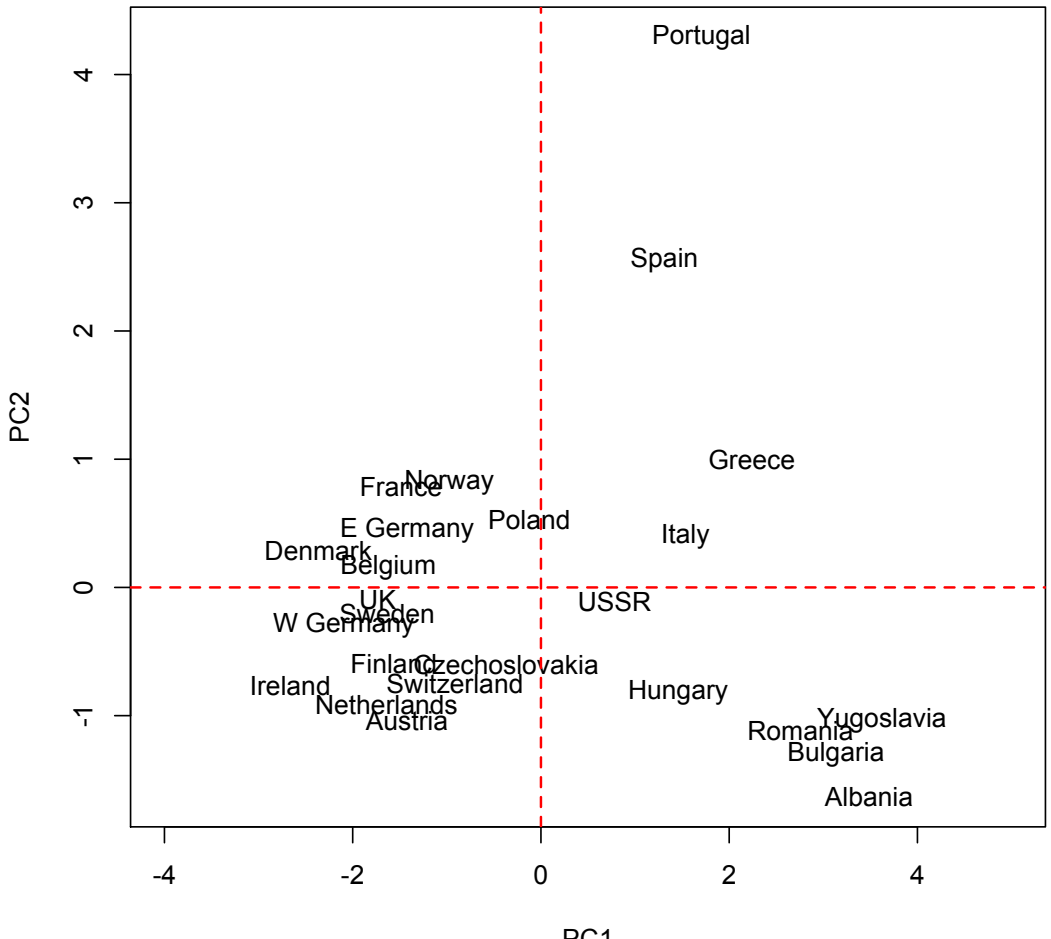
Let's look at a simple example... Data: Protein consumption by person by country for 7 variables: red meat, white meat, eggs, milk, fish, cereals, starch, nuts, vegetables.





Looks to me that PC1 measures how rich you are and PC2 something to do with the Mediterranean diet!

Principal Components Analysis

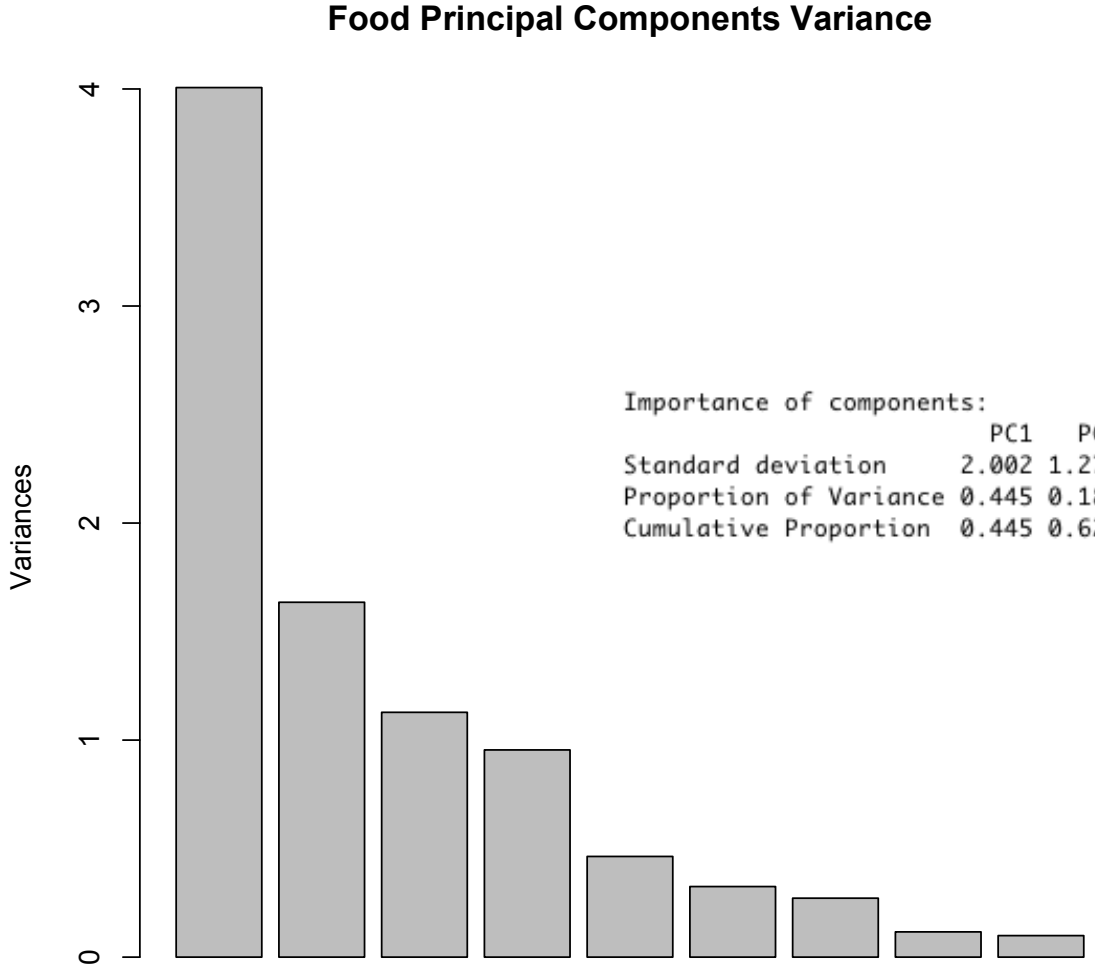


	RedMeat	WhiteMeat	Eggs	Milk	Fish	Cereals	Starch	Nuts	Fr.Veg
PC1	-0.30	-0.31	-0.43	-0.38	-0.14	0.44	-0.30	0.42	0.11
PC2	-0.06	-0.24	-0.04	-0.18	0.65	-0.23	0.35	0.14	0.54
PC3	-0.30	0.62	0.18	-0.39	-0.32	0.10	0.24	-0.05	0.41

These are the weights defining the principal components...

Principal Components Analysis

3 variables might be enough to represent this data... Most of the variability (75%) is explained with PC1, PC2 and PC3.



Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Standard deviation	2.002	1.279	1.062	0.977	0.6811	0.5702	0.5212	0.3410	0.315
Proportion of Variance	0.445	0.182	0.125	0.106	0.0515	0.0361	0.0302	0.0129	0.011
Cumulative Proportion	0.445	0.627	0.752	0.858	0.9098	0.9459	0.9761	0.9890	1.000

Principal Components Analysis: Comments

- PCA is a great way to summarize data
- It "clusters" both variables and observations simultaneously!
- The choice of k can be evaluated as a function of the interpretation of the results or via the fit (% of the variation explained)
- The units of each PC is not interpretable in an absolute sense. However, relative to each other it is... see example above.
- Always a good idea to center the data before running PCA.

Principal Components Regression (PCR)

Let's go back to and think of predicting Y with a potentially large number of X variables...

PCA is sometimes used as a way to reduce the dimensionality of X... if only a small number of PC's are enough to represent X, I don't need to use all the X's, right? Remember, smaller models tend to do better in predictive terms!

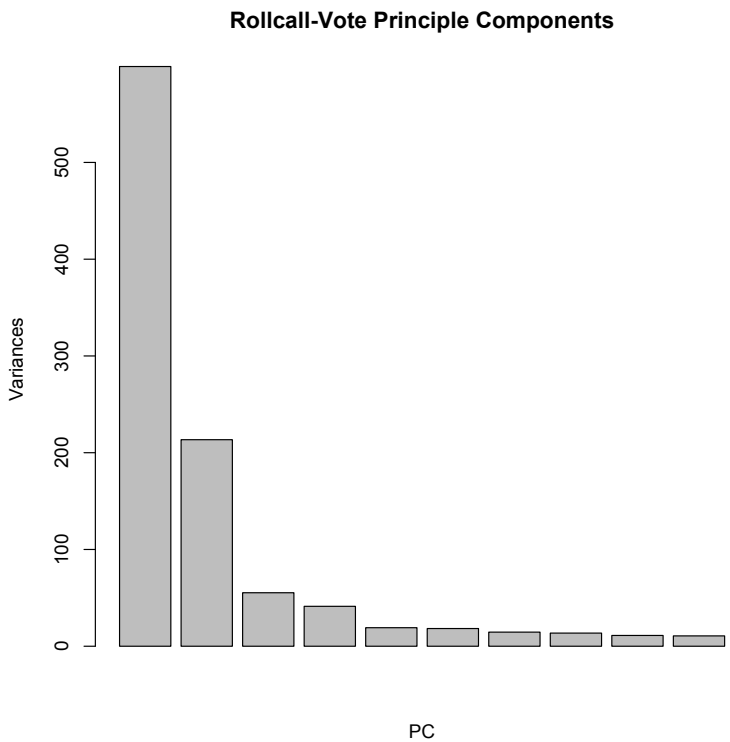
This is called Principal Component Regression. First represent X via k principal components (Z) and then run a regression of Y onto Z. PCR assumes that the *directions in which shows the most variation (the PCs), are the directions associated with Y*.

The choice of k can be done by comparing the out-of-sample predictive performance.

Principal Components Regression (PCR)

Example: Roll Call Votes in Congress... all votes in the 111^{th} Congress (2009-2011); $p = 1647$, $n = 445$.
Goal: Predict party how "liberal" a district is a function of the votes by their representative

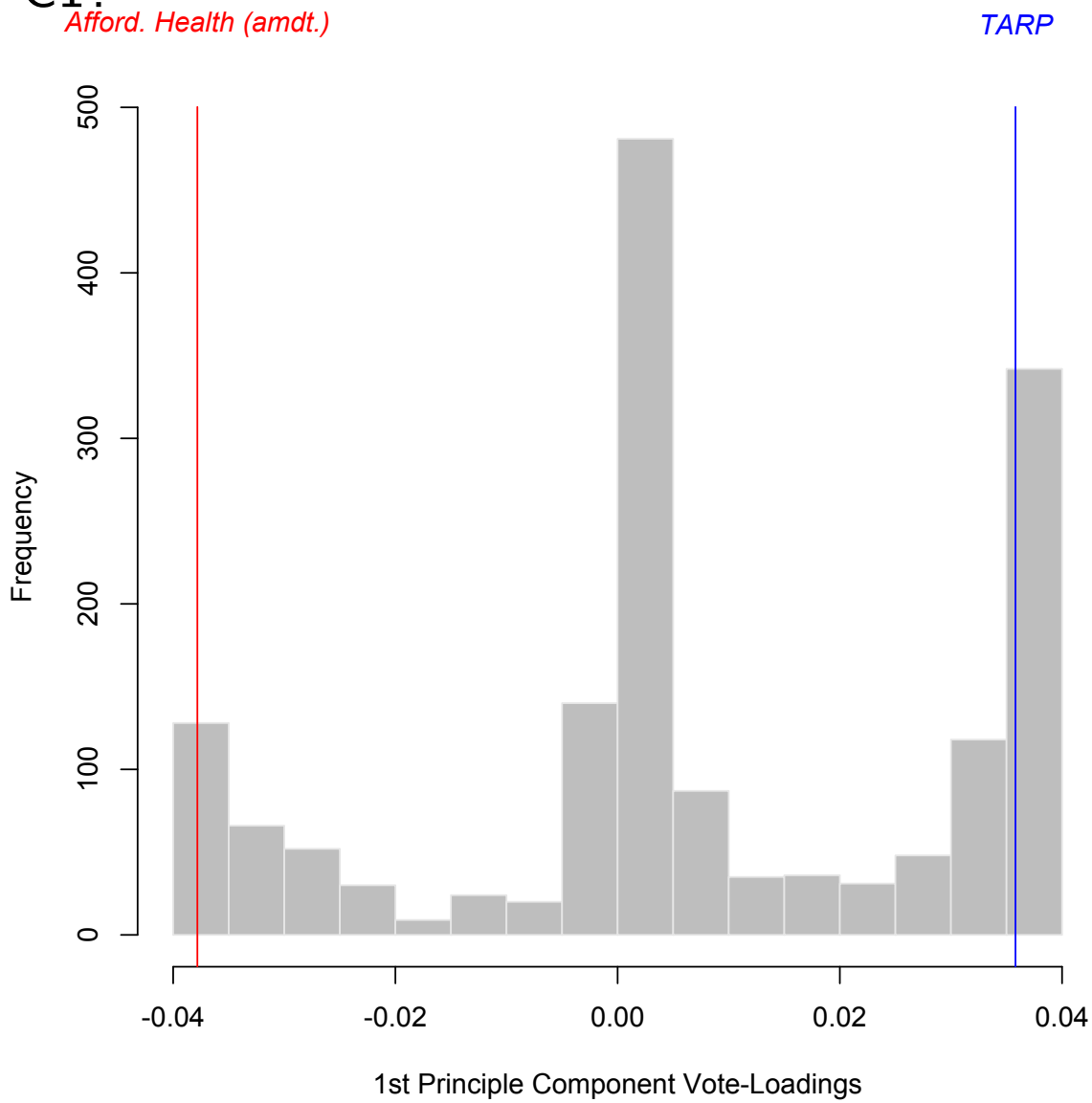
Let's first take the principal component decomposition of the votes...



It looks like 2 PC capture much of what is going on...

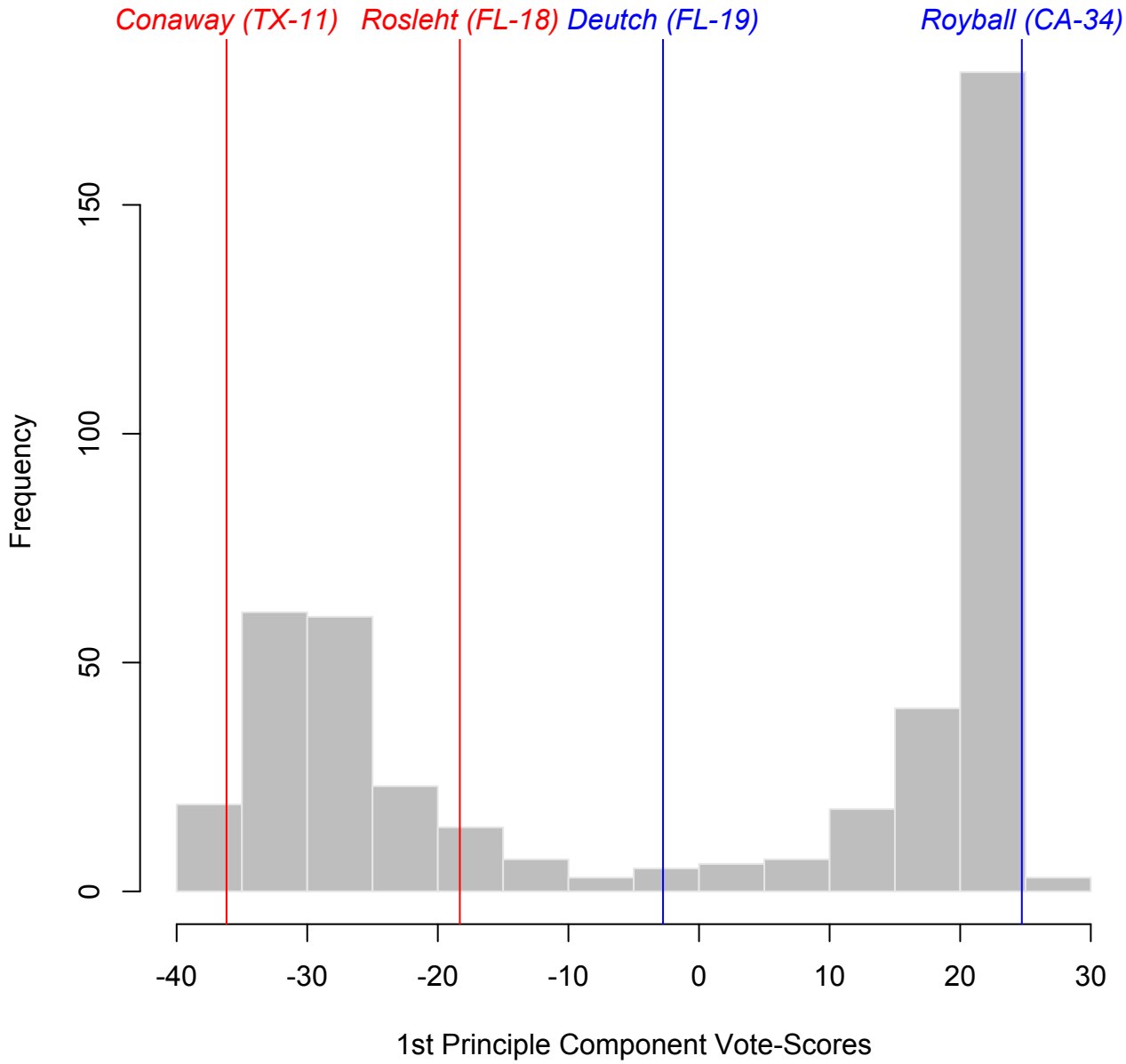
Principal Components Regression (PCR)

Histogram of loadings on PC1... What bills are important in defining PC1?



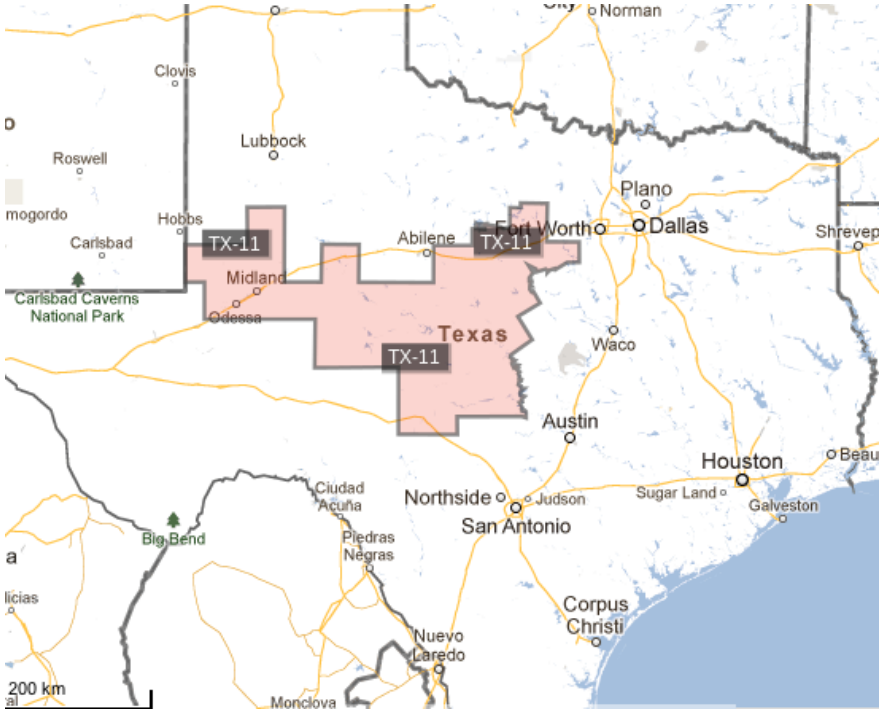
Principal Components Regression (PCR)

Histogram of PC1... "Ideology Score"

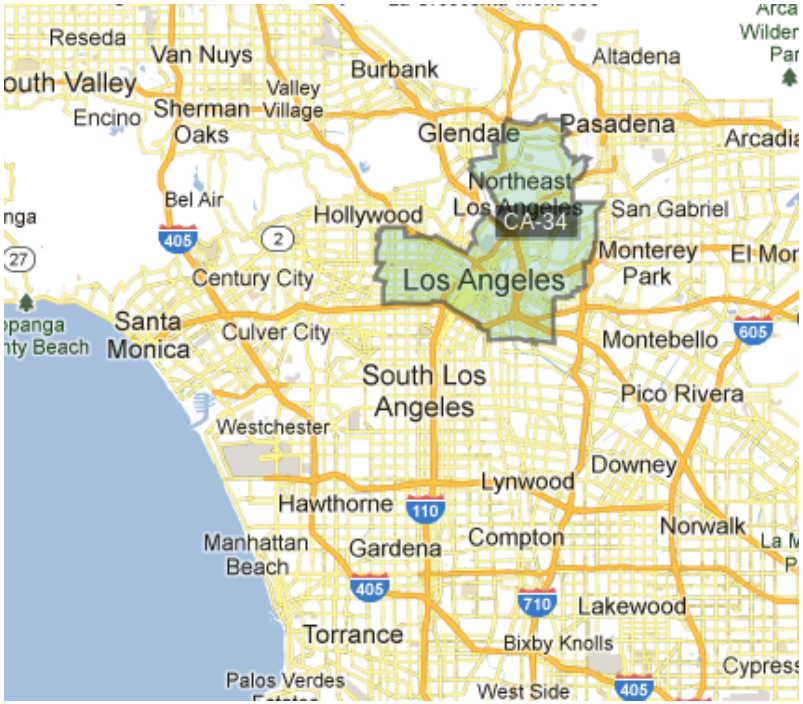


Principal Components Regression (PCR)

TX-11



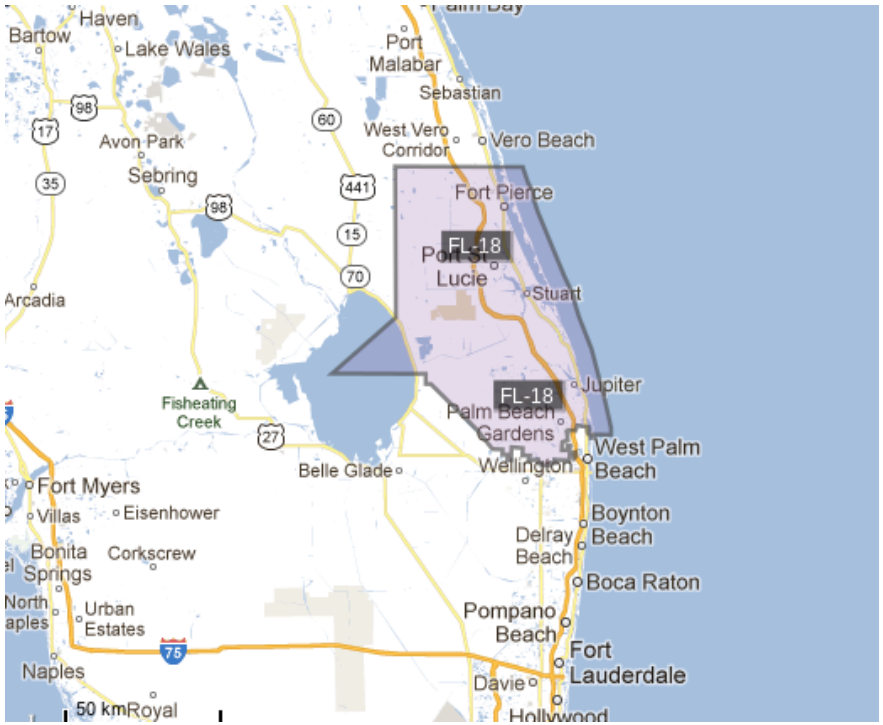
CA-34



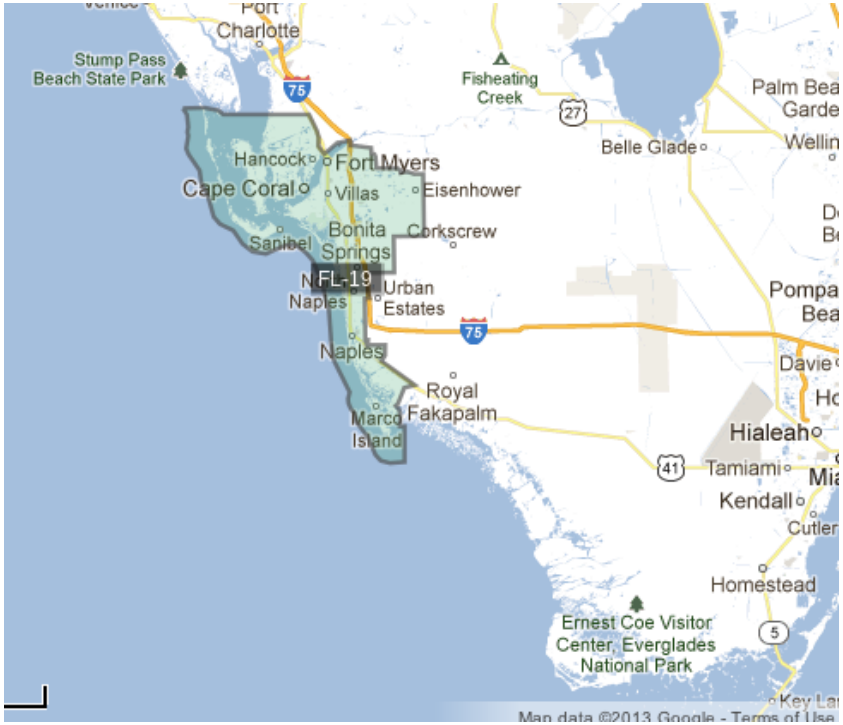
The two extremes...

Principal Components Regression (PCR)

FL-18

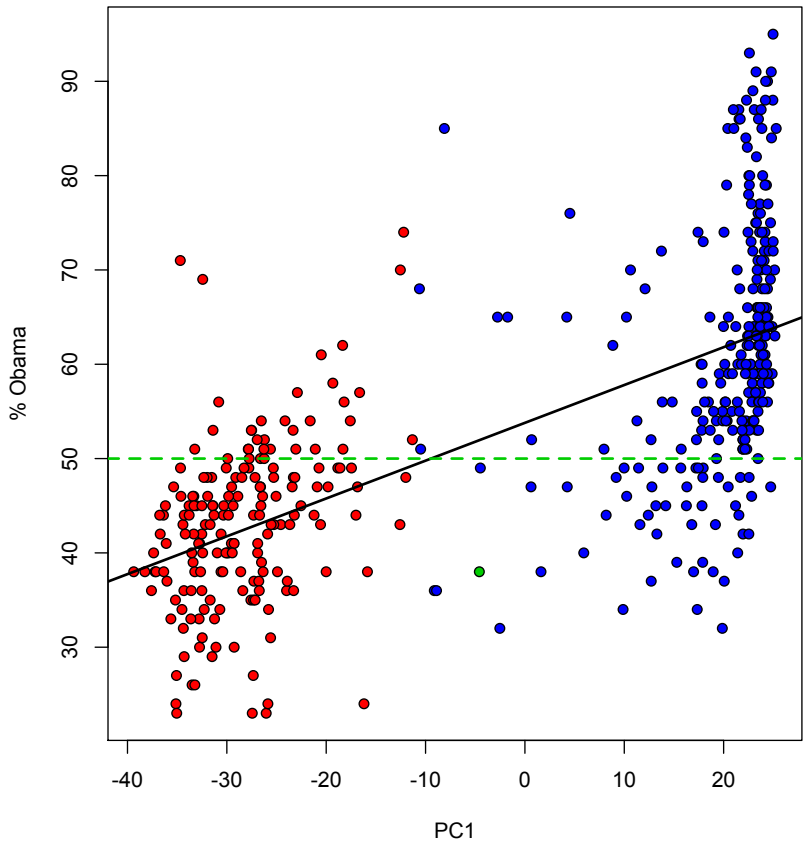


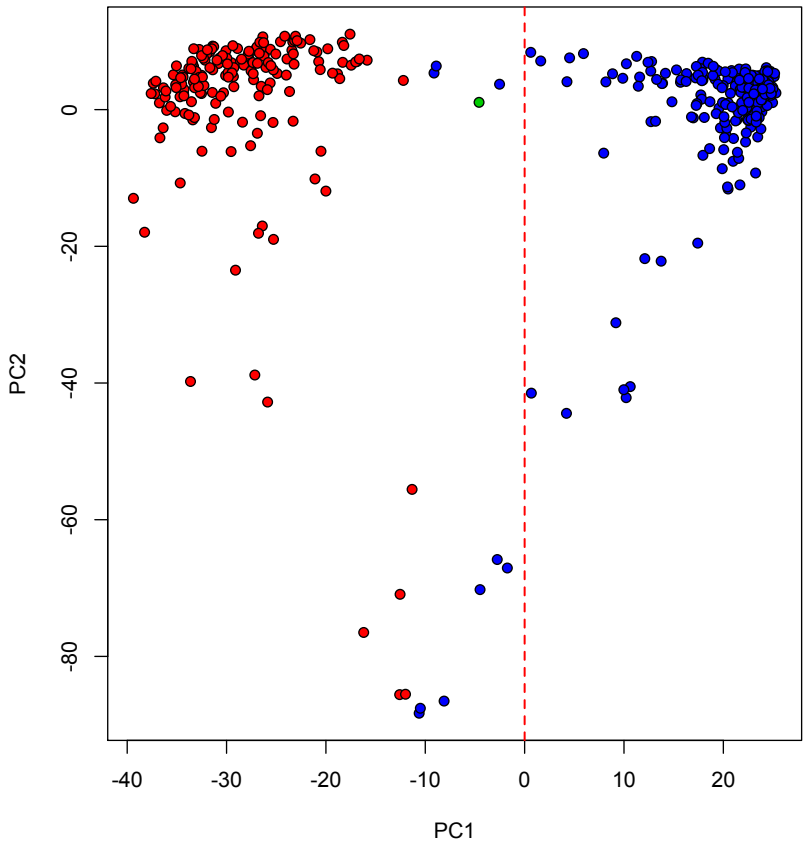
FL-19



The swing state!

Principal Components Regression (PCR)

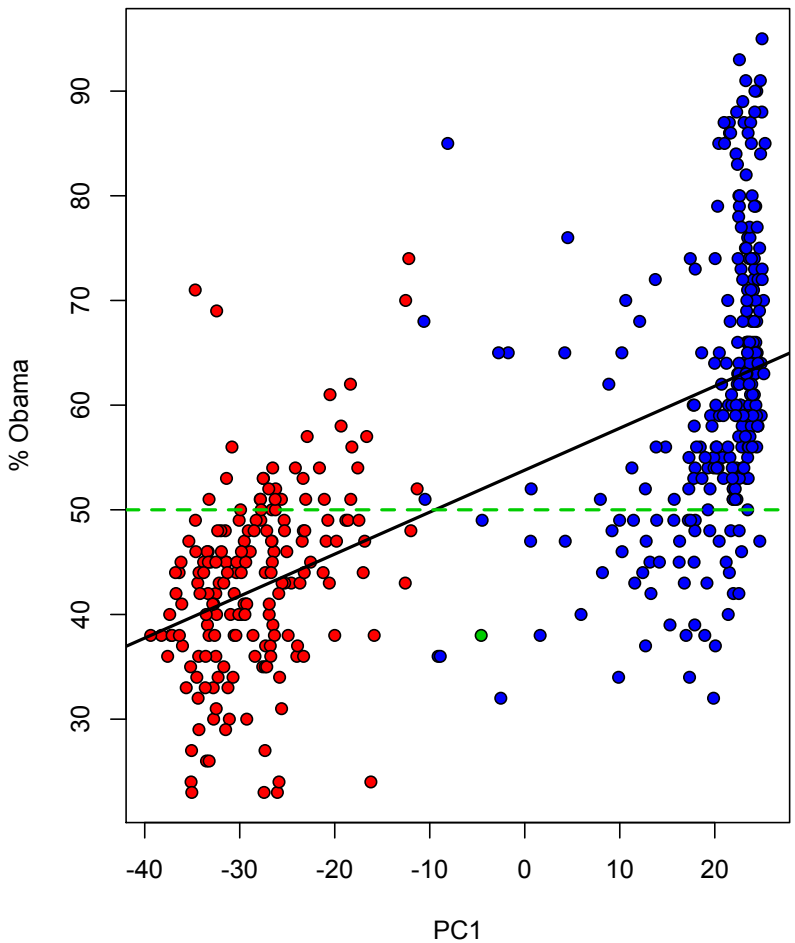


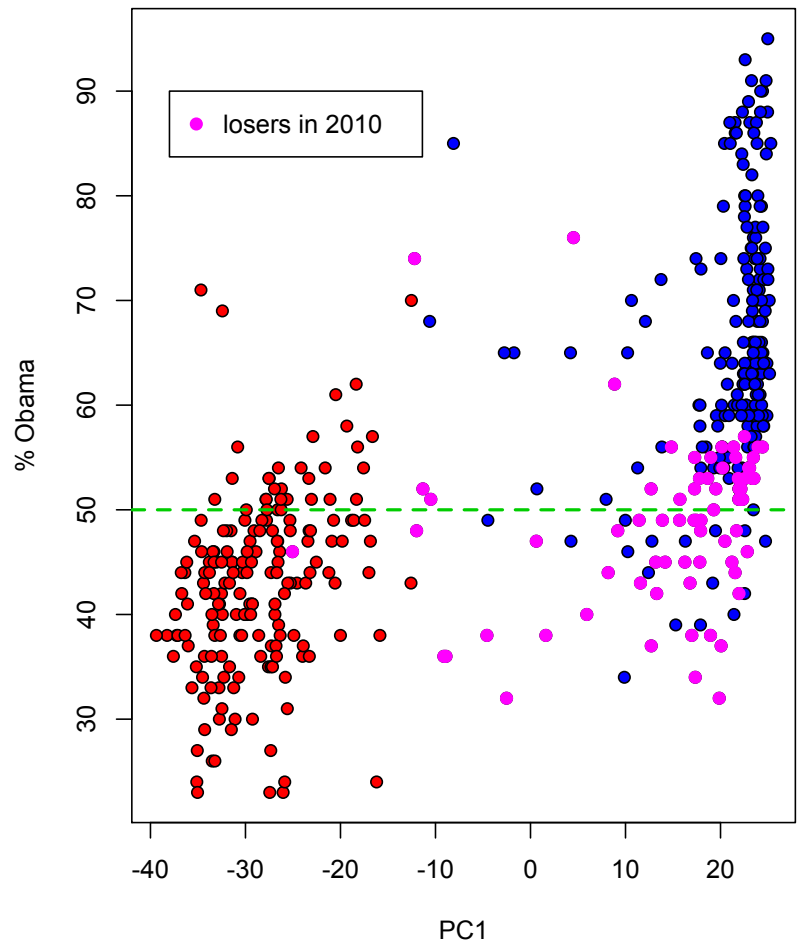


All we need is PC1 to predict party affiliation!

How can this picture help you understand what happened in the 2010 election?

Principal Components Regression (PCR)





Partial Least Squares (PLS)

PLS works very similarly to PCR as it will create "new" variables (Z) by taking linear combinations of the original variables (X).

The different is that PLS *attempts to find the directions of variation in X that help explain BOTH X and Y.*

It is a *supervised learning* alternative to PCR...

Partial Least Squares (PLS)

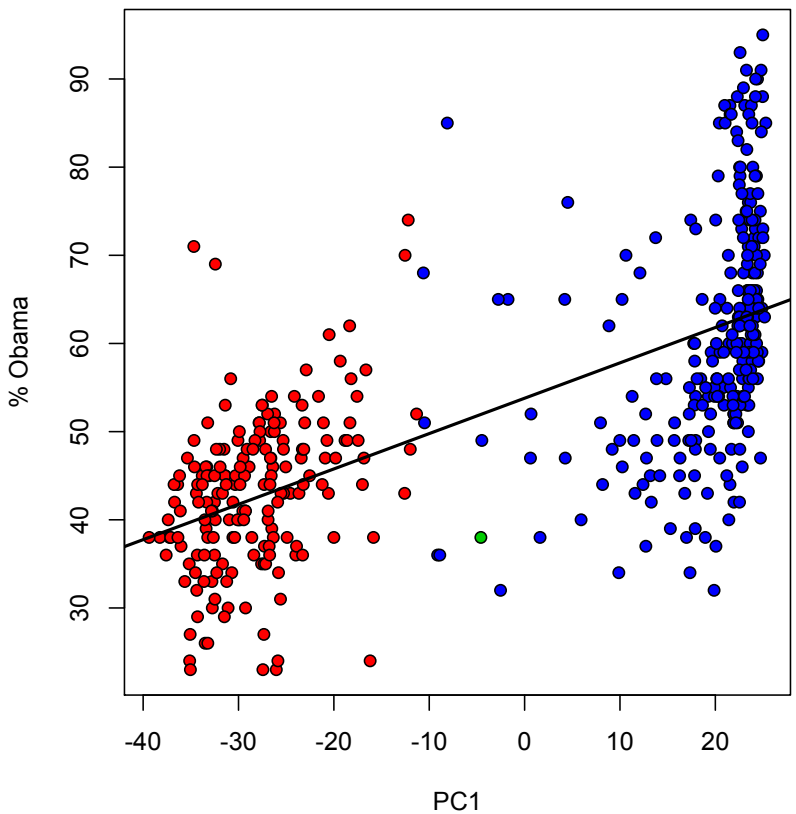
PLS works as follows:

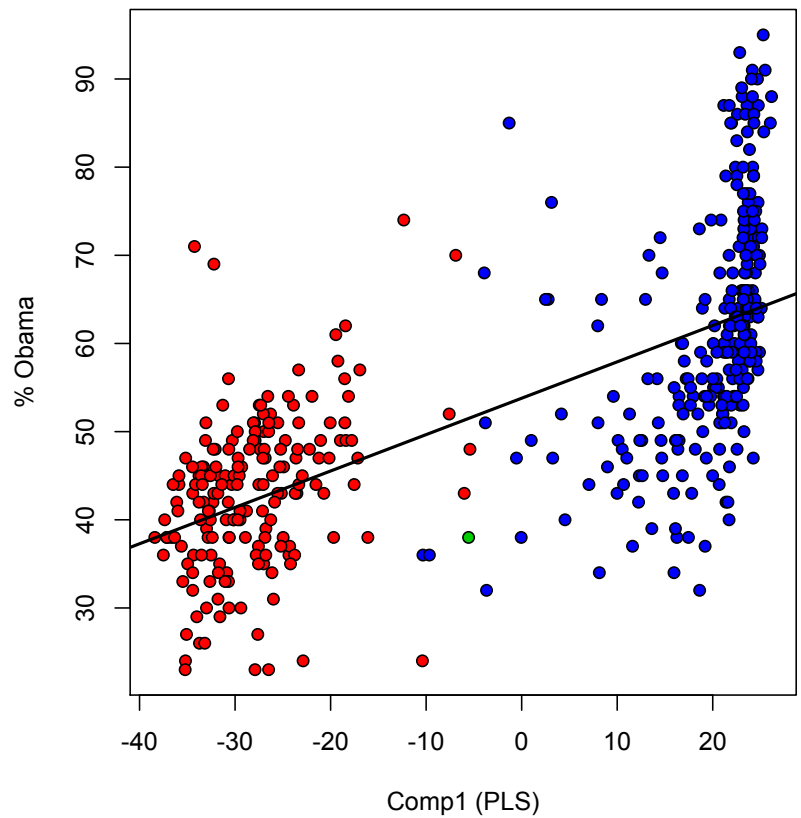
1. The weights of the first linear combination (Z_1) is defined by the regression of Y onto each of the X's... i.e., large weights are going to placed on the X variables most related to Y in a univariate sense
2. Regress each X variable onto Z_1 and compute the residuals
3. Repeat step (1) using the residuals from (2) in place of X
4. iterate

As always, the choice of where to stop, i.e., how many Z variables to use should be done by comparing the out-of-sample predictive performance.

Partial Least Squares (PLS)

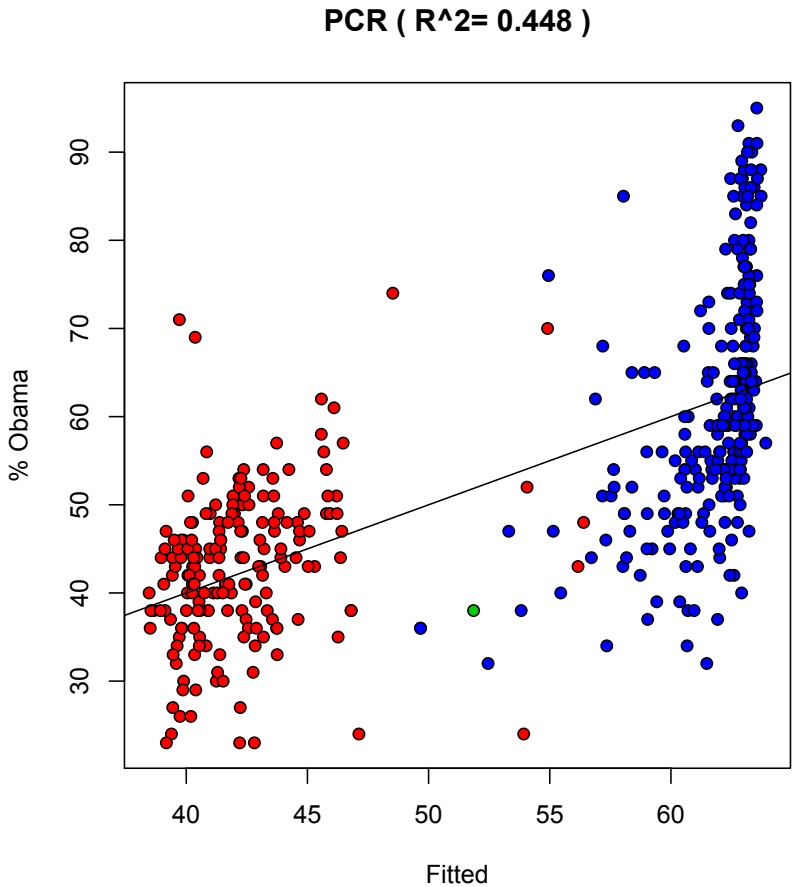
Roll Call Data again... it looks like the first component from PLS is the same as the first principal component!

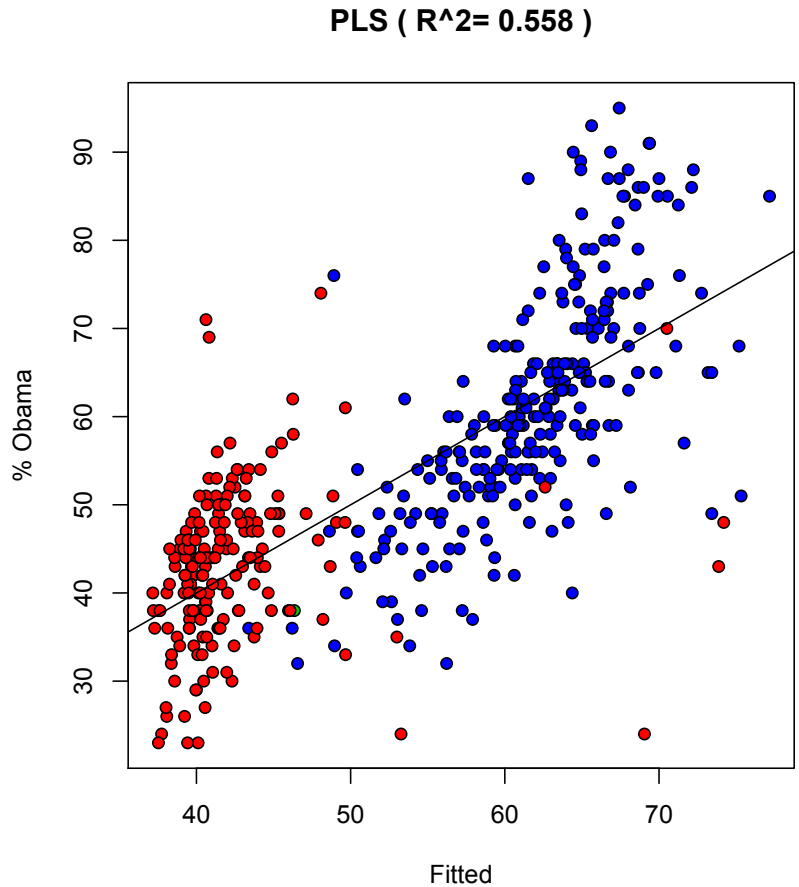




Partial Least Squares (PLS)

But, using two components PLS does better than PCR!!





Partial Least Squares (PLS)

Not easy to understand the difference between the second component in each method (how is that for a homework!)... the bottom line is that by using the information from Y in summarizing the X variables, PLS find a second component that has the ability to explain part of Y.