Big Data and Economics

Linear Model Selection and Regularization

Kyle Coombs
Bates College | ECON/DCS 368

Table of contents

Prologue

Regressions

- What do we typically do when we run OLS?
- We run a regression with all the variables we think are important
- But what happens when we have more variables than observations?

Too many variables

- Most of the analysis we have done in this class has focused on the case where we have a small number of variables relative to the number of observations.
- But sometimes you have WIDE data
- In this case, you have a large number of variables J relative to the number of observations n.
- If you try to use OLS with all the variables, you will run into problems. Why?

Too many variables

- Most of the analysis we have done in this class has focused on the case where we have a small number of variables relative to the number of observations.
- But sometimes you have WIDE data
- In this case, you have a large number of variables J relative to the number of observations n.
- If you try to use OLS with all the variables, you will run into problems. Why?
- The number of variables is larger than the number of observations!
- Uh oh

Example of wide data

```
## Warning: The x argument of as_tibble.matrix() must have unique column names if
## .name_repair is omitted as of tibble 2.0.0.
## ℹ Using compatibility .name_repair.
## This warning is displayed once every 8 hours.
## Call lifecycle::last_lifecycle_warnings() to see where this warning was
## generated.

## # A tibble: 6 × 1,001
##        y     P_1     P_2    P_3    P_4     P_5     P_6     P_7    P_8    P_9
##    <dbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>
## 1   8.33 -0.560  -0.710   2.20  -0.715 -0.0736 -0.602   1.07   -0.728  0.356
## 2 -63.4  -0.230   0.257   1.31  -0.753 -1.17   -0.994  -0.0273 -1.54  -0.658
## 3  -8.21  1.56   -0.247  -0.265 -0.939 -0.635   1.03   -0.0333 -0.693  0.855
## 4  11.7   0.0705 -0.348   0.543 -1.05  -0.0288  0.751  -1.52    0.119  1.15
## 5  35.9   0.129  -0.952  -0.414 -0.437  0.671  -1.51    0.790  -1.36   0.276
## 6 -41.4   1.72   -0.0450 -0.476  0.331 -1.65   -0.0951 -0.211   0.590  0.144
## # ℹ 991 more variables: P_10 <dbl>, P_11 <dbl>, P_12 <dbl>, P_13 <dbl>,
## #   P_14 <dbl>, P_15 <dbl>, P_16 <dbl>, P_17 <dbl>, P_18 <dbl>, P_19 <dbl>,
## #   P_20 <dbl>, P_21 <dbl>, P_22 <dbl>, P_23 <dbl>, P_24 <dbl>, P_25 <dbl>,
## #   P_26 <dbl>, P_27 <dbl>, P_28 <dbl>, P_29 <dbl>, P_30 <dbl>, P_31 <dbl>,
## #   P_32 <dbl>, P_33 <dbl>, P_34 <dbl>, P_35 <dbl>, P_36 <dbl>, P_37 <dbl>,
## #   P_38 <dbl>, P_39 <dbl>, P_40 <dbl>, P_41 <dbl>, P_42 <dbl>, P_43 <dbl>,
## #   P_44 <dbl>, P_45 <dbl>, P_46 <dbl>, P_47 <dbl>, P_48 <dbl>, P_49 <dbl>, …
```

What if I run a regression?

A mess to include all variables

```
etable(feols(y ~ ..('^P'), data = wide_df))
```

```
## The variables 'P_100', 'P_101' and 899 others have been removed because of collinearity (see $collin.var)

##                   feols(y ~ ..
## Dependent Var.:               y
##
## Constant           357.7 (NaN)
## P_1                246.0 (NaN)
## P_2               -58.65 (NaN)
## P_3               -21.77 (NaN)
## P_4               -78.74 (NaN)
## P_5                14.93 (NaN)
## P_6                280.5 (NaN)
## P_7               -109.2 (NaN)
## P_8               -181.9 (NaN)
## P_9               -393.5 (NaN)
## P_10               265.5 (NaN)
## P_11              -3.331 (NaN)
## P_12              -6.579 (NaN)
## P_13               202.2 (NaN)
## P_14               88.59 (NaN)
## P_15               355.9 (NaN)
## P_16              -134.5 (NaN)
## P_17               161.8 (NaN)
## P_18              -60.64 (NaN)
```

How can we cut down on variables?

- How can we cut down on the number of variables?
- What would be the regression tree approach?

How can we cut down on variables?

- How can we cut down on the number of variables?
- What would be the regression tree approach?
- Iteratively split training data using variables that minimize residual sum of squares and use test data to determine the optimal number of leaves
- This is a form of **variable selection**
- But it forces us to turn continuous data binary ($X > c$ vs. $X\backslash\mathrm{geqc}$)
- But what other ways are available?

Linear Model Selection

Typical OLS

- Good old-fashioned regression minimizes the residual sum of squares (RSS)

$$\min_{\beta} \underbrace{\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2}_{\text{RSS}}$$

- What does that mean?

Typical OLS

- Good old-fashioned regression minimizes the residual sum of squares (RSS)

$$\min_{\beta} \underbrace{\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2}_{\text{RSS}}$$

- What does that mean?
- We are trying to find the βs that predict a dependent variable y as a linear combination of the independent variables x.

Adding dimensions with OLS

- Each additional variable x_j adds a new dimension to the problem
    - As in each additional variable is a new axis in J-dimensional space where J is the number of variables
    - (You've likely never thought about it that way before, but any regression is a multi-dimensional problem)
- If you have more variables than observations, you have more dimensions than observations
- Why? Well solve this equation:

$$x + y = 5$$

- How many solutions are there? Infinite
- Now solve this system of equations:

$$x + y = 5$$

$$x + 2y = 10$$

- The same logic applies to regression (though it is a bit more complicated)

Ridge Regression

Shrinkage

- In OLS, we are trying to minimize the residual sum of squares (RSS)
- In machine learning, there are shrinkage methods that add a penalty term to the RSS
    - These penalize coefficients that are too large

$$\min_{\beta} \sum_{i=1}^{n} \underbrace{\text{model fit}}_{\text{RSS}} + \text{penalty on size of coefficients}$$

- Why penalize large coefficients?
- Large coefficients are more likely to be overfitting the data since they are more sensitive to small changes in the data
    - By penalizing large coefficients, we are reducing the variance of the model and thus complexity
    - Intuitively, a larger β the further your model is from a null hypothesis of $\beta = 0$, which is the simplest model
- What happens if we reduce bias in the data?

Shrinkage

- In OLS, we are trying to minimize the residual sum of squares (RSS)
- In machine learning, there are shrinkage methods that add a penalty term to the RSS
    - These penalize coefficients that are too large

$$\min_{\beta} \sum_{i=1}^{n} \underbrace{\text{model fit}}_{\text{RSS}} + \text{penalty on size of coefficients}$$

- Why penalize large coefficients?
- Large coefficients are more likely to be overfitting the data since they are more sensitive to small changes in the data
    - By penalizing large coefficients, we are reducing the variance of the model and thus complexity
    - Intuitively, a larger β the further your model is from a null hypothesis of $\beta = 0$, which is the simplest model
- What happens if we reduce bias in the data?
- We increase variance!

Ridge Regression

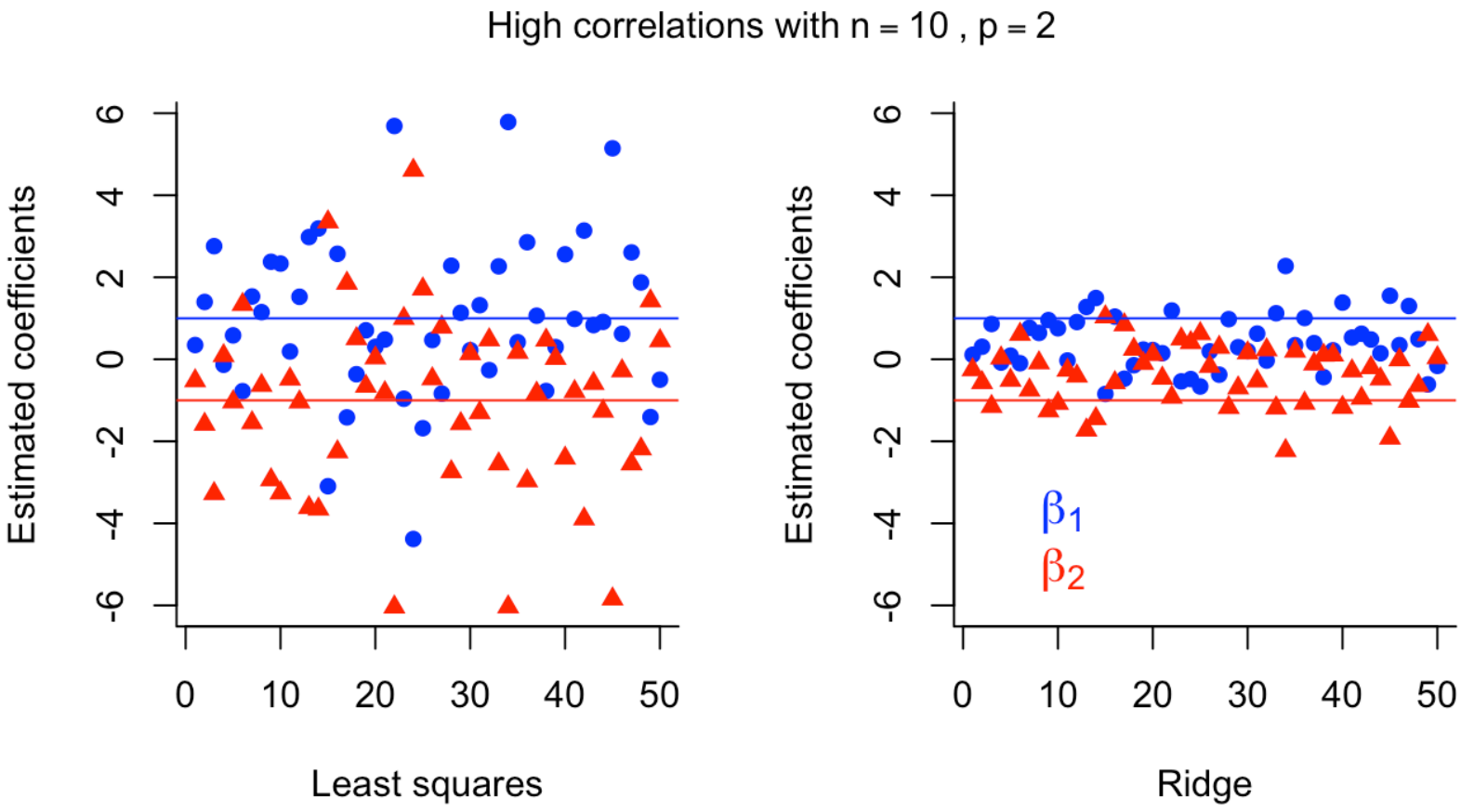
- So what form do these penalties take?
- Well Ridge Regression is one such example
- Ridge regression adds a penalty term to the RSS that is proportional to the sum of the squared coefficients
- Essentially, it adds a constraint to the optimization problem

$$\min \underbrace{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{J}\beta_j x_{ij})}_{\text{model fit}} + \underbrace{\lambda \sum_{j=1}^{J}\beta_j^2}_{\text{penalty}} = RSS + \lambda \sum_{j=1}^{J}\beta_j^2$$

λ is the "tuning parameter" that controls the strength of the penalty

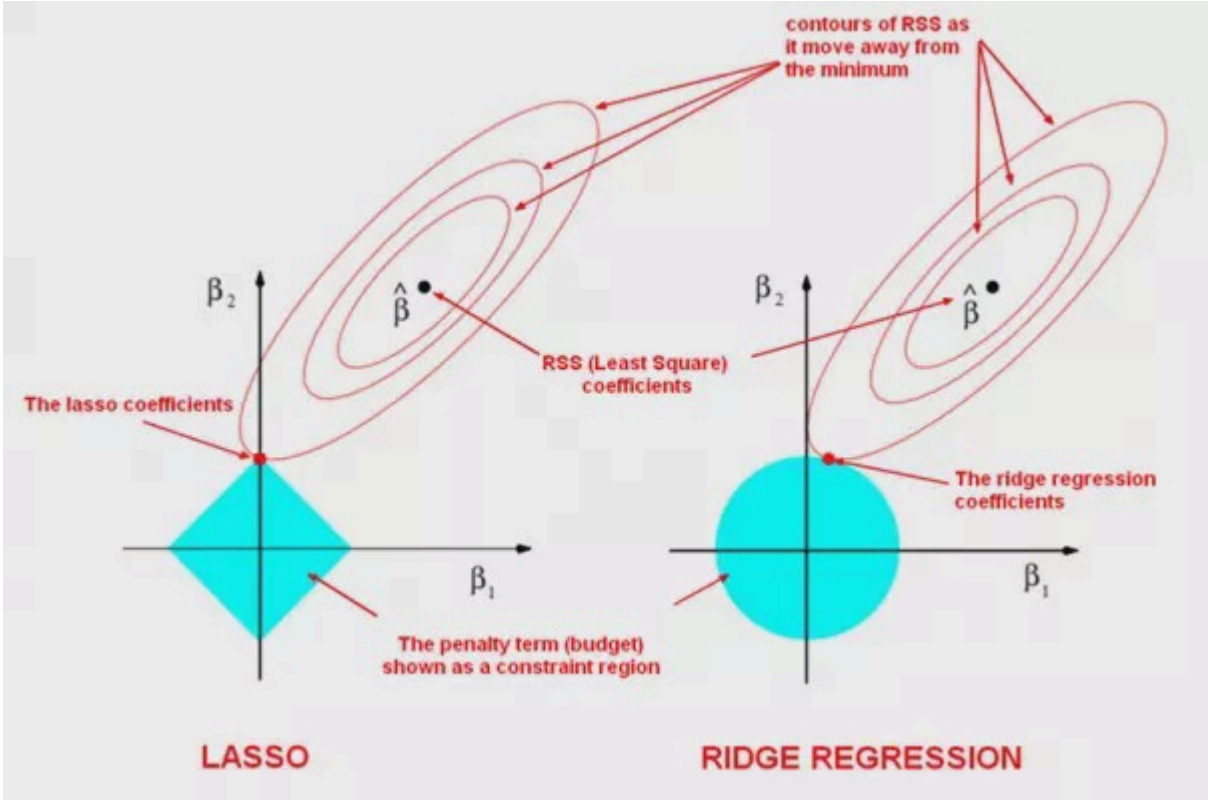
- In order to minimize, we need to find the βs that minimize the RSS and the penalty
- That means we need smaller βs -- and necessarily a simpler, less variable model
- Literally, we shrink the βs towards zero

Ridge Regression



Example taken from Dr. Samuel E. Jackson online textbook

Ridge Regression coefficients



Ridge Regression flaws

- Ridge regression keeps all the variables in the model -- it just shrinks the coefficients
- But what if some variables are just truly noise
    - i.e. they are not correlated with the dependent variable
- Sure, we can check by hand, but shouldn't we just toss them?

LASSO

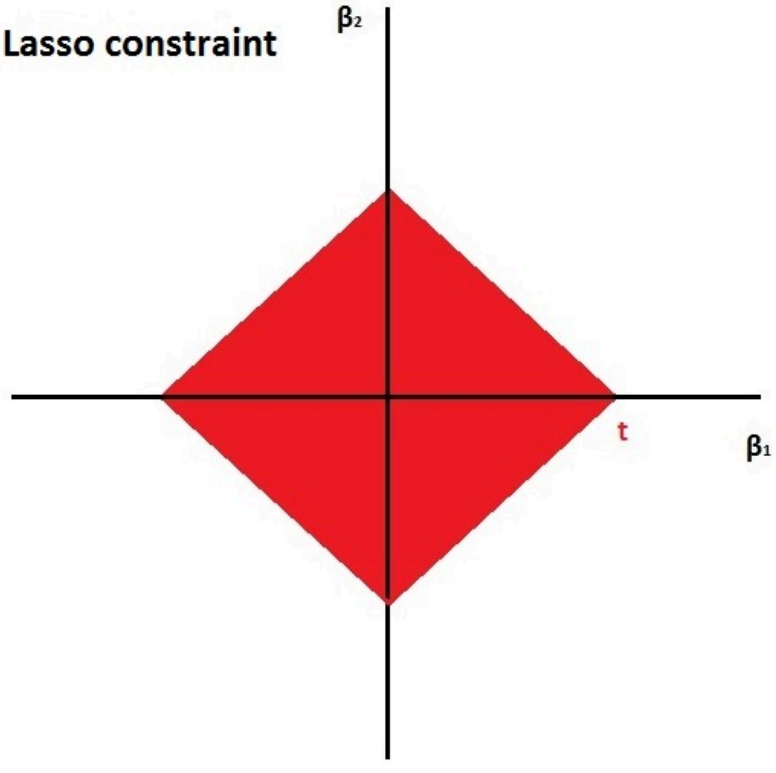
LASSO

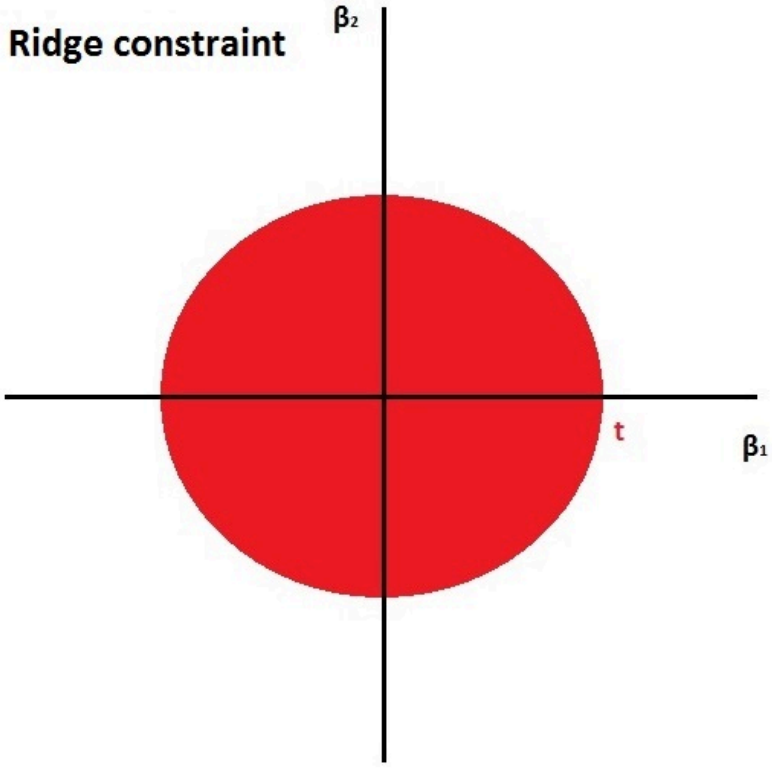
- LASSO stands for Least Absolute Shrinkage and Selection Operator
- It is another shrinkage method that adds a penalty term to the RSS
- But now the penalty term is proportional to the sum of the absolute value of the coefficients

$$\min \underbrace{\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{J}\beta_j x_{ij})}_{\text{model fit}} + \underbrace{\lambda \sum_{j=1}^{J}|\beta_j|}_{\text{penalty}} = RSS + \lambda \sum_{j=1}^{J}|\beta_j|$$

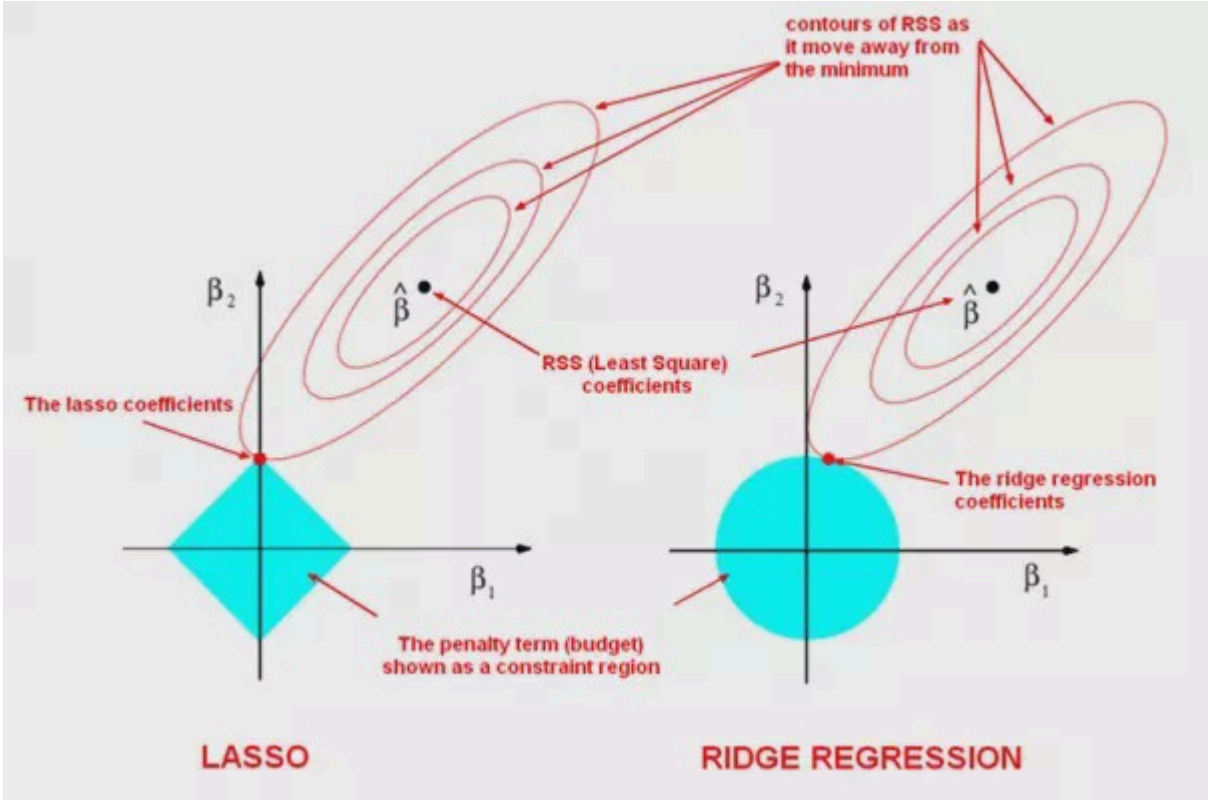
- Instead of the squared penalty on coefficient size, you have absolute value
- The magic of the absolute value is that it can shrink coefficients to zero with a sufficiently large λ
  - This means that LASSO can select variables: $\beta_j = 0$ means that x_j is not in the model
  - **Intuition**: The absolute value has a "sharp" corner at zero, so it can "cut" coefficients to zero, Ridge is a circle, so it can only shrink coefficients to the edge of the circle
- Selection is a big advantage over Ridge Regression
  - Of course, that can also be a disadvantage if you want to keep all the variables in the model
  - It leads to more bias

LASSO visualization





Ridge Regression coefficients



Other details on Regularization

K-fold cross-validation: How to pick λ

- The λ in is a "tuning parameter," which controls the strength of the penalty
- You need to do K-fold cross-validation:

1. Choose the number of "folds" or groups, K (usually 5 or 10)
2. Randomly split the data into K folds
3. Create a grid of feasible λ values to check
4. For each value of λ:
    - Run Ridge or LASSO on the $K-1$ folds
    - Calculate the MSE_k on the remaining k-fold
5. Calculate the average MSE_k for each λ

$$MSE_{CV}(\lambda) = \frac{1}{K}\sum_{k=1}^{K} MSE_k(\lambda)$$

6. Pick the λ with the lowest MSE

- You know what's neat? You can do this in R with the **glmnet** library!
- It will even plot the results for you, so you can see the optimal λ

Drawbacks of LASSO and Ridge

- Regularization/coefficient shinkage are useful for reducing variance and overfitting
- But they can also lead to bias
- The more you shrink the coefficients, the more bias you introduce
- You are no longer finding the best linear unbiased estimator (BLUE) that you find with OLS
- Instead, you get the best linear biased estimator (BLBE) because you trade some bias for less variance
- Sometimes you're okay with that!

Why are you okay with bias?

- Sometimes you don't mind being a little off in your predictions
- For example, if you are predicting the number of people who will show up to a party, you don't care if you are a little off
- Imagine someone tells you there's a 50% chance 0 people come and a 50% chance 100 people come
    - That's not very helpful

Why are you okay with bias?

- Sometimes you don't mind being a little off in your predictions
- For example, if you are predicting the number of people who will show up to a party, you don't care if you are a little off
- Imagine someone tells you there's a 50% chance 0 people come and a 50% chance 100 people come
    - That's not very helpful
- But what if they predict 45-55 people will show up and then 40 people showed up
    - That's wrong, but not so wrong to cause problems
- It is even less helpful if they tell you that to make an accurate prediction they need to know:
    - The number of invites
    - The weather
    - The day of the week
    - The time of day
    - The number of people who have already RSVP'd
    - The variety of chips you're serving
    - What is on TV that night
    - etc.

Warning

- Regularization is a useful tool for reducing variance and overfitting
- But just cause you can run a regression techniques doesn't mean you should
- You should always think about the problem you are trying to solve and the data you have
- Is it worth trying a technique?
- Will this technique help you solve your problem?
- Will it help you understand your data?
- Or are you just trying to seem flashy?

Conclusion

- Regularization is a useful tool for reducing variance and overfitting
- It recognizes that sometimes you are okay with a little bias if it means you get less variance
- It relies on a tuning parameter λ that controls the strength of the penalty from adding more complexity to a regression model
- LASSO can be used to select variables, while Ridge just reduces the magnitude of the coefficients

What next?

- Try an activity: ISLR lab using tidymodels
- Before class: work through the lab sections on Ridge and LASSO in a .Rmd file that you create
- Write up short answers to the following questions:

    1. What are the coefficients in the Ridge and LASSO regressions when the penalty is zero? Why?
    2. How does tidymodels pick the optimal λ in each method?
    3. What is the optimal λ in Ridge and LASSO?

Next lecture: Regular expressions and word clouds