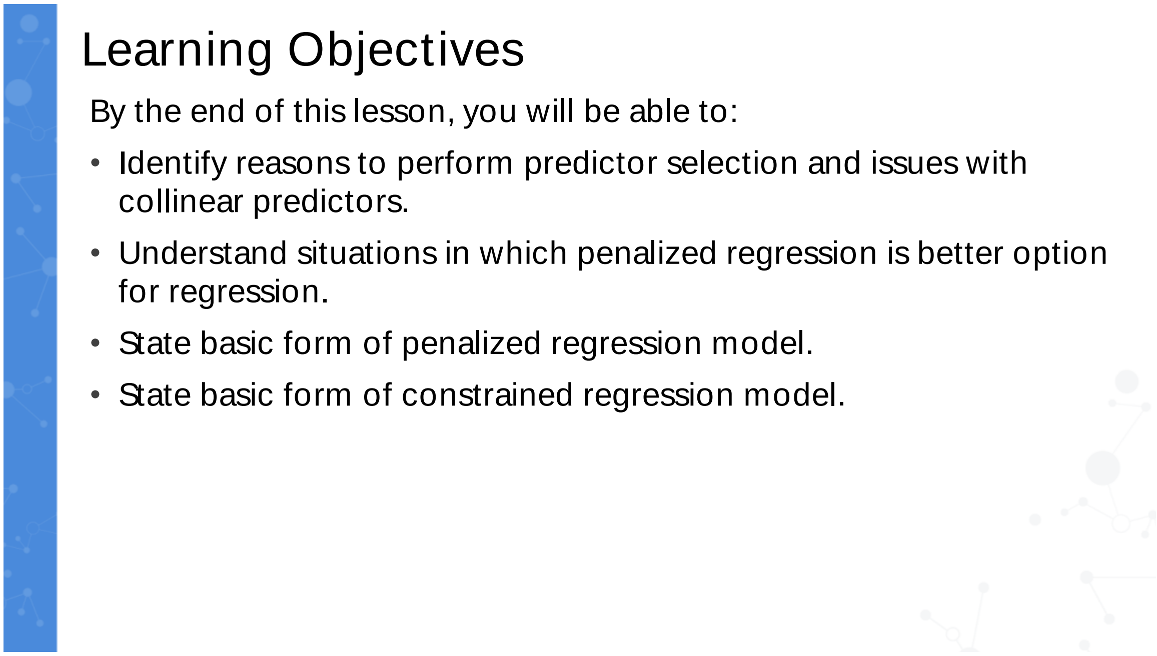
UNIVERSITY OF WISCONSIN
DATA SCIENCE

DS 740

Data Mining

Predictor Selection and Penalties
Background for Penalized Regression

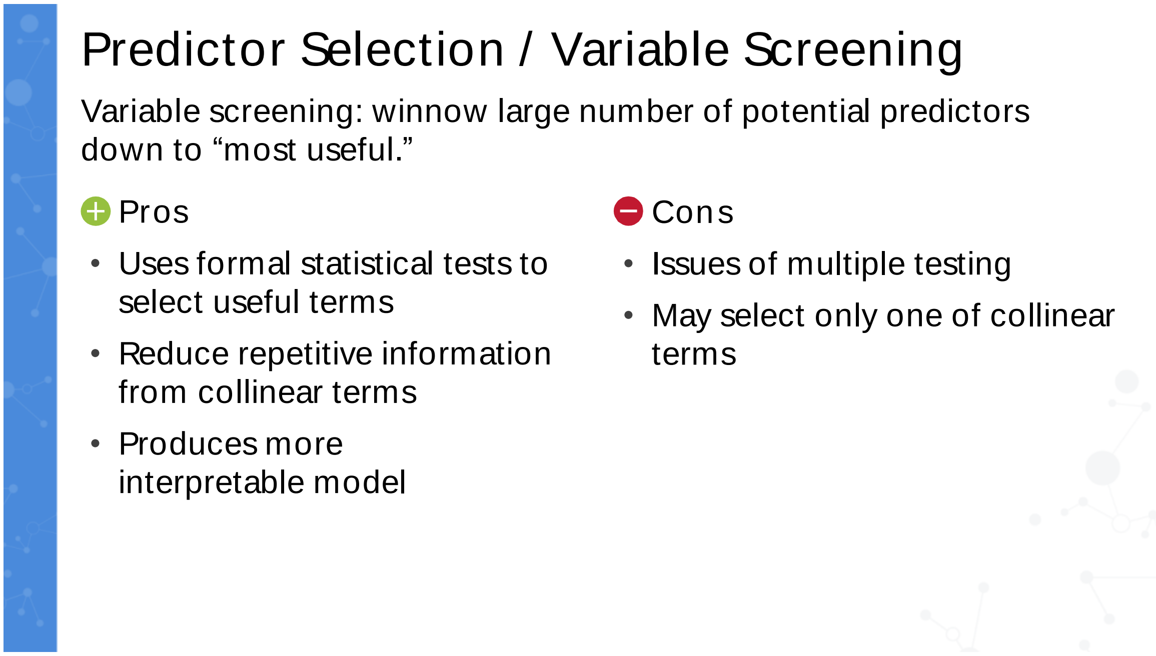
Important note: Transcripts are **not** substitutes for textbook assignments.



Learning Objectives

By the end of this lesson, you will be able to:

- Identify reasons to perform predictor selection and issues with collinear predictors.
- Understand situations in which penalized regression is better option for regression.
- State basic form of penalized regression model.
- State basic form of constrained regression model.



Predictor Selection / Variable Screening

Variable screening: winnow large number of potential predictors down to "most useful."

⊕ Pros

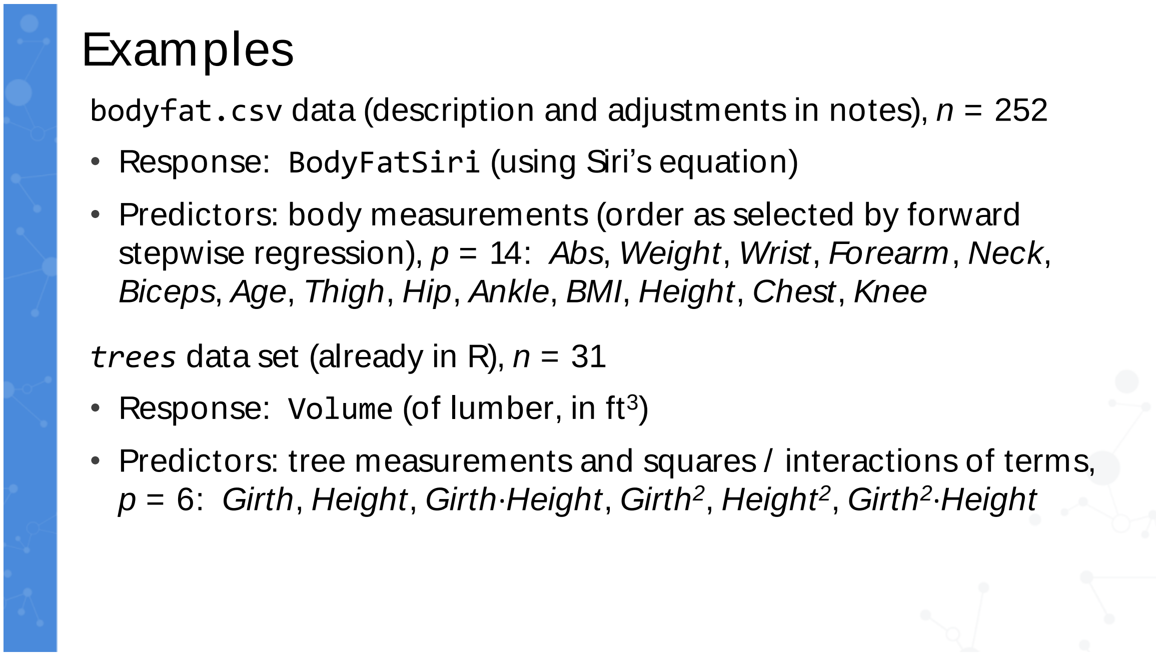
- Uses formal statistical tests to select useful terms
- Reduce repetitive information from collinear terms
- Produces more interpretable model

⊖ Cons

- Issues of multiple testing
- May select only one of collinear terms

As part of multiple linear regression review, you saw predictor thinning be a stepwise and best subset selection methods. Such methods are aimed to thin the original number of predictors to a manageable size. We get a set of most useful predictors for the response, which is particularly helpful in achieving an accurate estimation with collinear or correlated predictors.

Some issues arise when using these methods, however, such as Type I and Type II errors resulting from multiple testing, as well as selection of possibly only one of collinear terms.



Examples

`bodyfat.csv` data (description and adjustments in notes), n = 252

- Response: `BodyFatSiri` (using Siri's equation)
- Predictors: body measurements (order as selected by forward stepwise regression), p = 14: *Abs*, *Weight*, *Wrist*, *Forearm*, *Neck*, *Biceps*, *Age*, *Thigh*, *Hip*, *Ankle*, *BMI*, *Height*, *Chest*, *Knee*

`trees` data set (already in R), n = 31

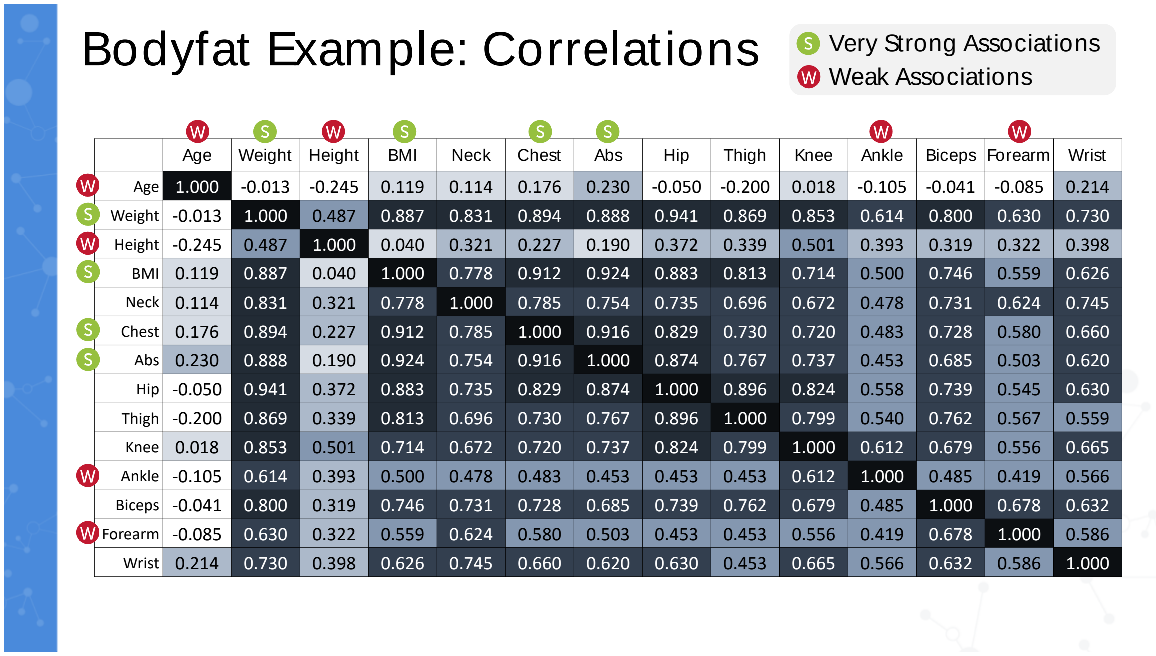
- Response: `Volume` (of lumber, in ft^3)
- Predictors: tree measurements and squares / interactions of terms, p = 6: *Girth*, *Height*, *Girth·Height*, *Girth*^{2}, *Height*^{2}, *Girth*^{2}·*Height*

We will examine two sets of collinear data in our discussions of penalized regression. The first set is the body fat data seen in previous lectures. The 14 predictors are all measures of body size in some way, so we anticipate similar information from at least some of them, meaning that there will be a correlation, some of them large.

In the second example, we use the trees data, which includes the original two predictors as well as three transformations of those original predictors. So strong correlations do exist.

Notes:

Find the bodyfat.csv file in the online course.
Descriptions and adjustments of the data set:
Fitting Percentage of Body Fat to Simple Body Measurements - Roger W. Johnson



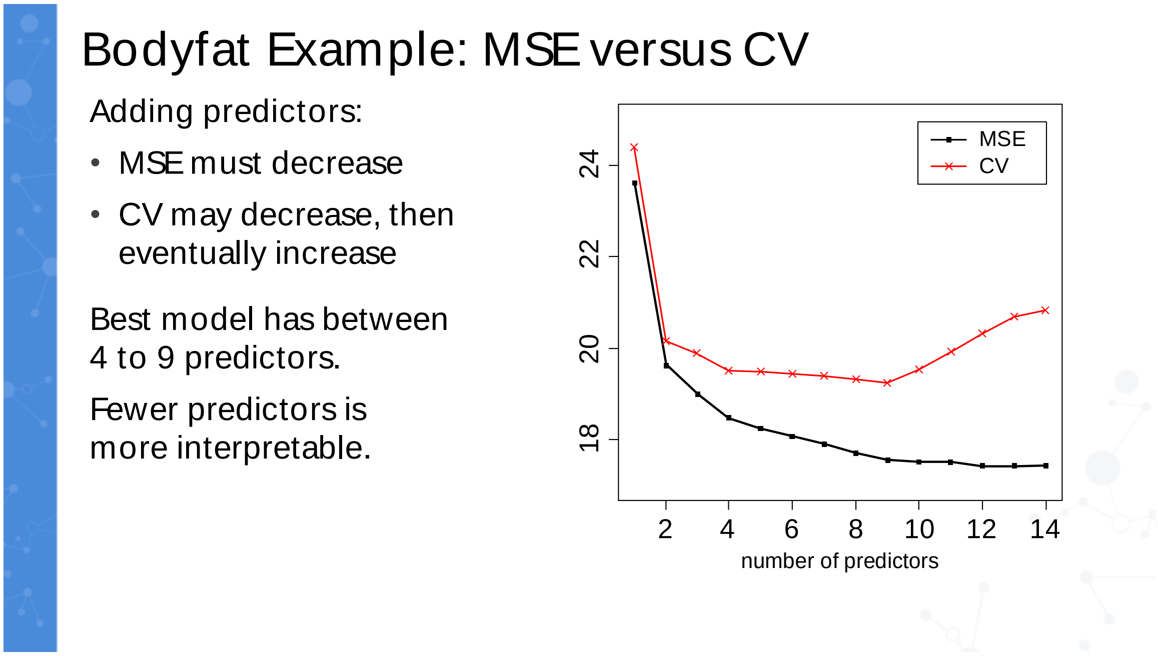
Bodyfat Example: Correlations

S Very Strong Associations
W Weak Associations

	W Age	S Weight	W Height	S BMI	Neck	S Chest	S Abs	Hip	Thigh	Knee	W Ankle	Biceps	W Forearm	Wrist
W Age	1.000	-0.013	-0.245	0.119	0.114	0.176	0.230	-0.050	-0.200	0.018	-0.105	-0.041	-0.085	0.214
S Weight	-0.013	1.000	0.487	0.887	0.831	0.894	0.888	0.941	0.869	0.853	0.614	0.800	0.630	0.730
W Height	-0.245	0.487	1.000	0.040	0.321	0.227	0.190	0.372	0.339	0.501	0.393	0.319	0.322	0.398
S BMI	0.119	0.887	0.040	1.000	0.778	0.912	0.924	0.883	0.813	0.714	0.500	0.746	0.559	0.626
Neck	0.114	0.831	0.321	0.778	1.000	0.785	0.754	0.735	0.696	0.672	0.478	0.731	0.624	0.745
S Chest	0.176	0.894	0.227	0.912	0.785	1.000	0.916	0.829	0.730	0.720	0.483	0.728	0.580	0.660
S Abs	0.230	0.888	0.190	0.924	0.754	0.916	1.000	0.874	0.767	0.737	0.453	0.685	0.503	0.620
Hip	-0.050	0.941	0.372	0.883	0.735	0.829	0.874	1.000	0.896	0.824	0.558	0.739	0.545	0.630
Thigh	-0.200	0.869	0.339	0.813	0.696	0.730	0.767	0.896	1.000	0.799	0.540	0.762	0.567	0.559
Knee	0.018	0.853	0.501	0.714	0.672	0.720	0.737	0.824	0.799	1.000	0.612	0.679	0.556	0.665
W Ankle	-0.105	0.614	0.393	0.500	0.478	0.483	0.453	0.453	0.453	0.612	1.000	0.485	0.419	0.566
Biceps	-0.041	0.800	0.319	0.746	0.731	0.728	0.685	0.739	0.762	0.679	0.485	1.000	0.678	0.632
W Forearm	-0.085	0.630	0.322	0.559	0.624	0.580	0.503	0.453	0.453	0.556	0.419	0.678	1.000	0.586
Wrist	0.214	0.730	0.398	0.626	0.745	0.660	0.620	0.630	0.453	0.665	0.566	0.632	0.586	1.000

For the body fat data, we examine the correlation matrix between the 14 predictors. Note that darker cells correspond to stronger positive correlations. Age, height, ankle and forearm are four predictors that do not show particularly strong associations with other predictors.

But the remaining 10 predictors are indeed strongly positively associated with each other. In particular, weight, BMI, chest and abs show a correlation greater than 0.9 with at least a couple predictors. So the question then becomes, do we really need to use all of these as predictors?



Bodyfat Example: MSE versus CV

Adding predictors:

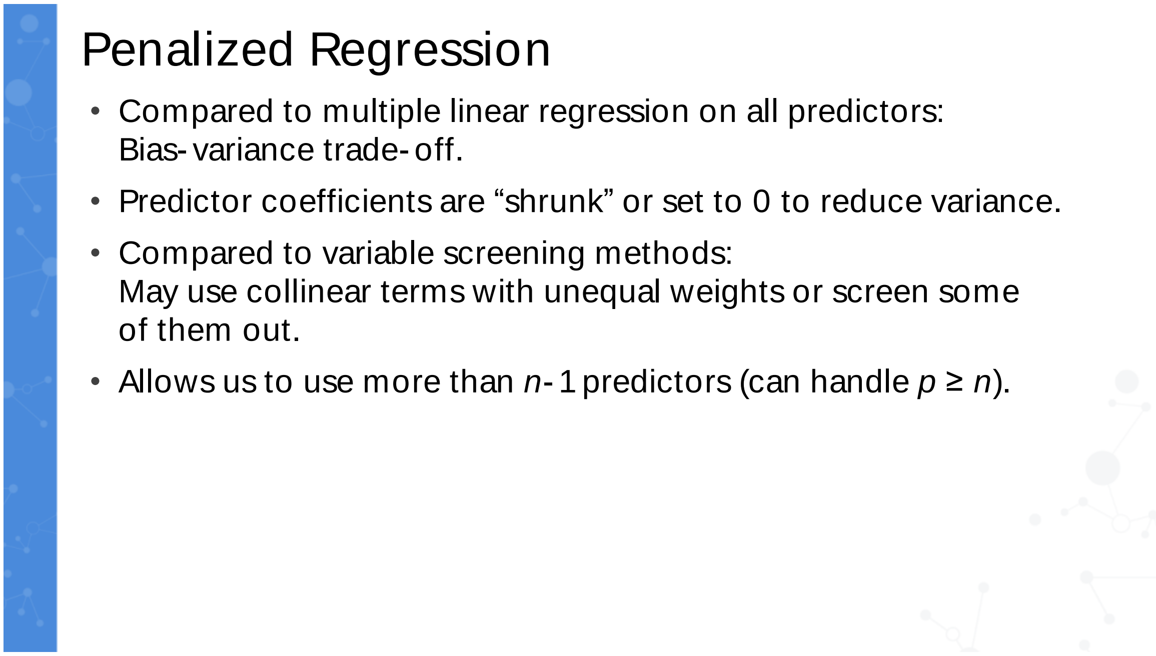
- MSE must decrease
- CV may decrease, then eventually increase

Best model has between 4 to 9 predictors.

Fewer predictors is more interpretable.

As seen in the discussion of cross-validation for model selection, while unadjusted mean square error, the MSE, must always decrease when we add more predictors, the amount of decrease levels off after about 9 predictors as we take a look at the plot of the MSE across the number of predictors included.

More tellingly, CV as a true measure of assessment of model predictive ability, is relatively flat from 4 to 9, which suggests 4 predictors will be essentially as good as up to 9 predictors. And then adding more than 9 predictors actually increases CV. This points to a less stable model, and not as good of a predictive model as the number of predictors increases above 9.

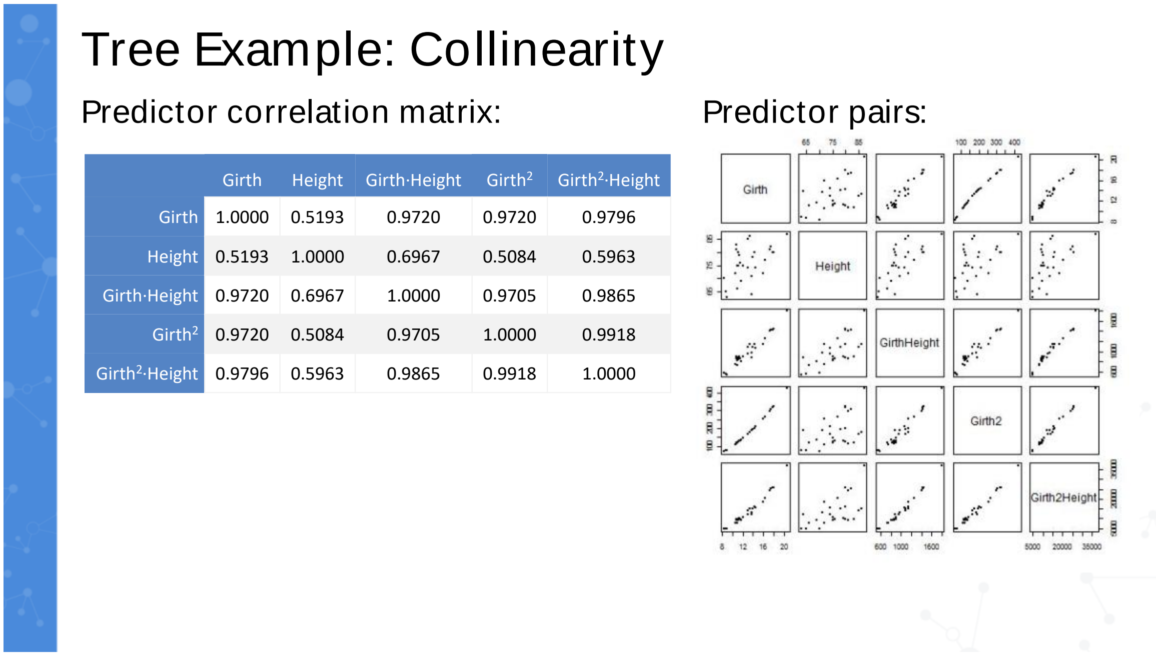


Penalized Regression

- Compared to multiple linear regression on all predictors:
  Bias-variance trade-off.
- Predictor coefficients are “shrunk” or set to 0 to reduce variance.
- Compared to variable screening methods:
  May use collinear terms with unequal weights or screen some of them out.
- Allows us to use more than $n-1$ predictors (can handle $p \geq n$).

This so-called instability when dealing with collinear predictors is based on the variability of the predictor estimates. This, in turn, gives a motivation for why we use what is known as penalized regression. While these methods of penalized regression add small amounts of bias to coefficient estimates, they can strongly reduce variance of the estimates by capping the overall values of the predictors.

This so to speak shrinks the coefficients toward 0, and for some methods, actually sets some of the coefficients to be 0, effectively removing the predictors from the model. This allows the method to handle collinear terms and even situations where we have more predictors than observations.



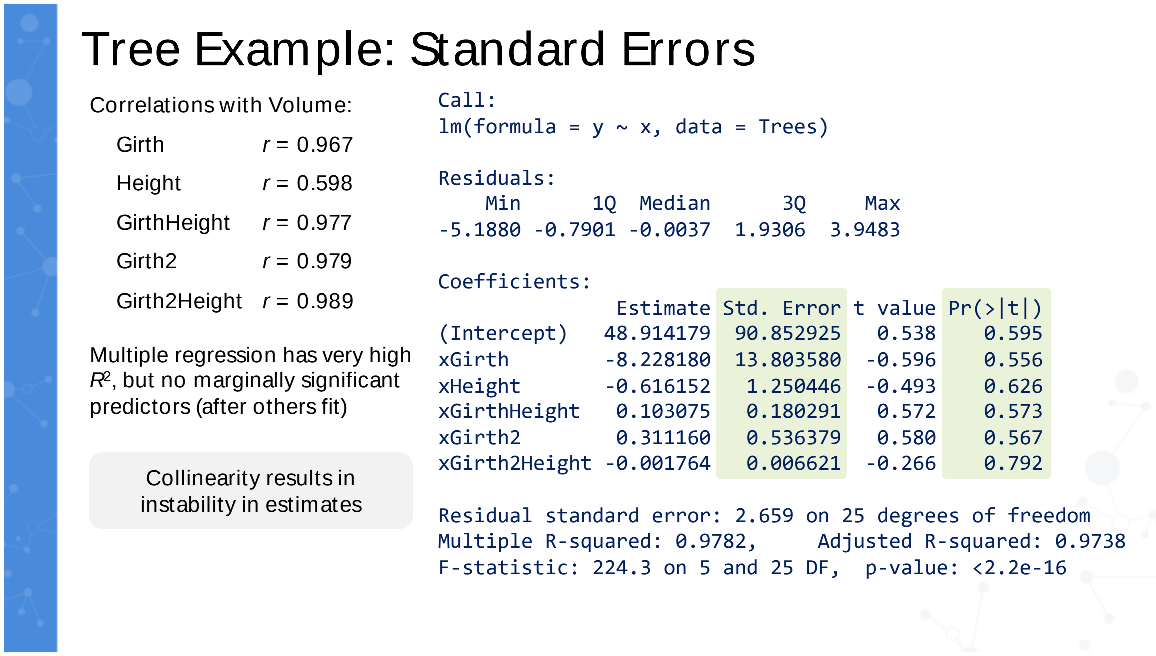
Tree Example: Collinearity

Predictor correlation matrix:

	Girth	Height	Girth·Height	Girth^2	Girth^2·Height
Girth	1.0000	0.5193	0.9720	0.9720	0.9796
Height	0.5193	1.0000	0.6967	0.5084	0.5963
Girth·Height	0.9720	0.6967	1.0000	0.9705	0.9865
Girth^2	0.9720	0.5084	0.9705	1.0000	0.9918
Girth^2·Height	0.9796	0.5963	0.9865	0.9918	1.0000

Predictor pairs:

As another example, going back to the trees data, we see some predictors that are not just strongly positively correlated, but extremely correlated, with correlations between 0.97 and 0.99. This occurs simply because some of the variables are transformations of others with large values.



Tree Example: Standard Errors

Correlations with Volume:

Girth	r = 0.967
Height	r = 0.598
GirthHeight	r = 0.977
Girth2	r = 0.979
Girth2Height	r = 0.989

Multiple regression has very high R^2, but no marginally significant predictors (after others fit)

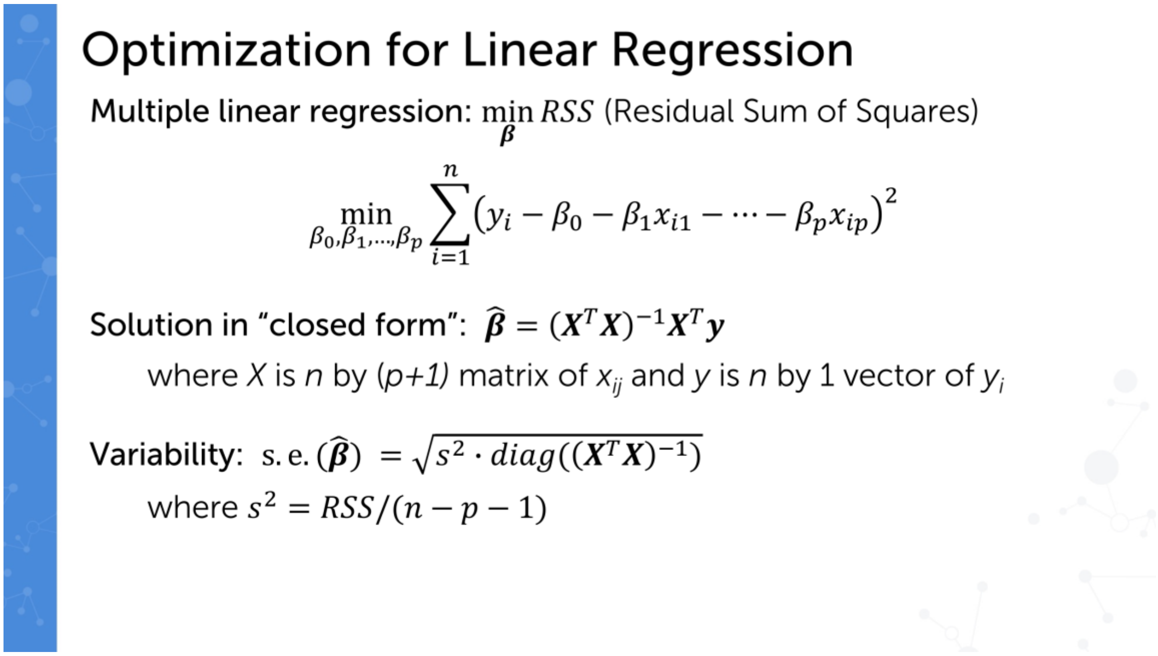
Collinearity results in instability in estimates

```
Call:
lm(formula = y ~ x, data = Trees)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.1880 -0.7901 -0.0037  1.9306  3.9483 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   48.914179  90.852925   0.538    0.595
xGirth        -8.228180  13.803580  -0.596    0.556
xHeight       -0.616152   1.250446  -0.493    0.626
xGirthHeight   0.103075   0.180291   0.572    0.573
xGirth2        0.311160   0.536379   0.580    0.567
xGirth2Height -0.001764   0.006621  -0.266    0.792

Residual standard error: 2.659 on 25 degrees of freedom
Multiple R-squared: 0.9782,     Adjusted R-squared: 0.9738 
F-statistic: 224.3 on 5 and 25 DF,  p-value: <2.2e-16
```

We note that for the trees data, in turn, the response y equals volume is also strongly correlated to most of the predictors. This results in a multiple linear regression model with a very high r squared value, theoretically meaning the model should predict the data very well. But at least it fits the data very well.

The collinearity of the predictors, however, means that some predictors can essentially replace others. The end result is when we take a look at the marginal significance of the predictors, that is, we take a look at the significance of each predictor after the others have been fit, none of them are significant.

Note that the marginal t-test of each coefficient uses the standard error of the coefficients in the denominator. And we are seeing very large standard errors relative to the size of the coefficient estimates. The reason why is that collinearity means instability in our estimates.



Optimization for Linear Regression

Multiple linear regression: $\min_{\boldsymbol{\beta}} RSS$ (Residual Sum of Squares)

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}\right)^2$$

Solution in "closed form": $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

where *X* is *n* by *(p+1)* matrix of x_{ij} and *y* is *n* by 1 vector of y_i

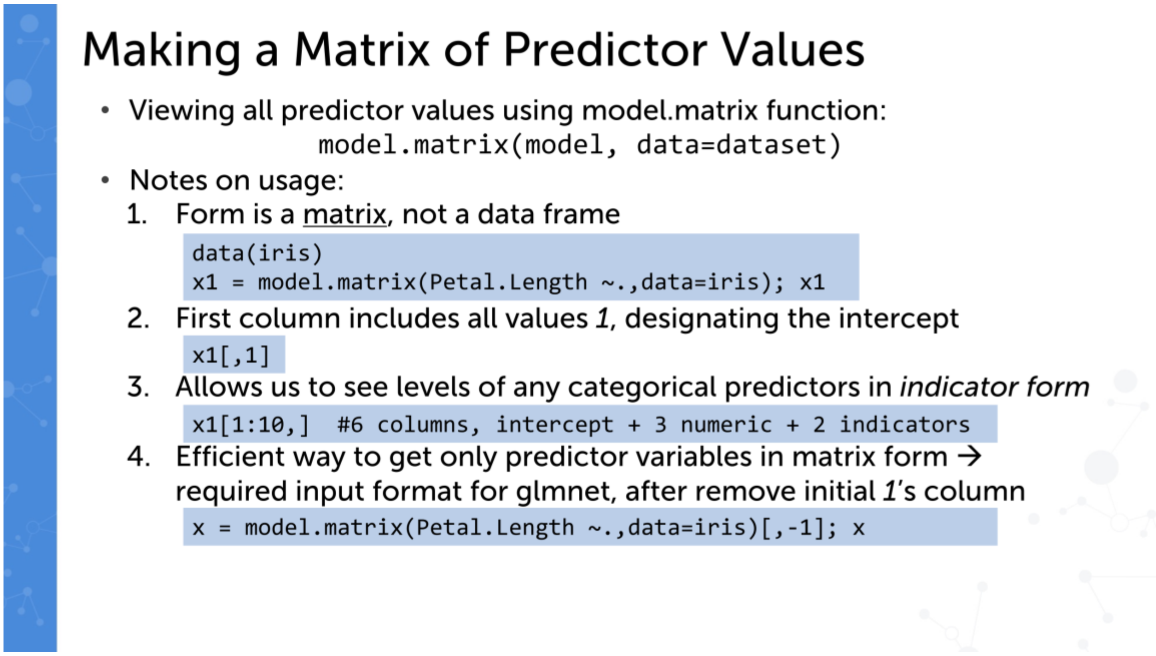
Variability: $\mathrm{s.e.}(\widehat{\boldsymbol{\beta}}) = \sqrt{s^2 \cdot diag((\boldsymbol{X}^T\boldsymbol{X})^{-1})}$

where $s^2 = RSS/(n-p-1)$

Coefficient estimates for multiple linear regression when residuals have constant variance is simply the result of a minimization of the sum of squared residuals. Mathematically, this produces a beautifully simple answer. The beta hats-- that is, the coefficient estimates-- are just a matrix calculation from the predictor matrix x and the response vector y. We'll take a little bit more of a look at the predictor matrix x on the next page.

With the additional assumption of normality of errors, we can estimate the variance of the errors by using s squared as the adjusted mean square error, and then the standard errors of the coefficient estimates-- that is, se of the beta hats are equal to the square root of s squared times the diagonal of the inverse of a matrix x transpose x. It is not an essential part of this course to understand the matrix setup, but we will need to produce a matrix as part of our application of penalized regression.

And we'll see how to do that in the next couple slides. We will also see how this affects our estimation of the coefficients. That is, what is the variability of our estimates in future examples?



Making a Matrix of Predictor Values

- Viewing all predictor values using model.matrix function:

```
model.matrix(model, data=dataset)
```

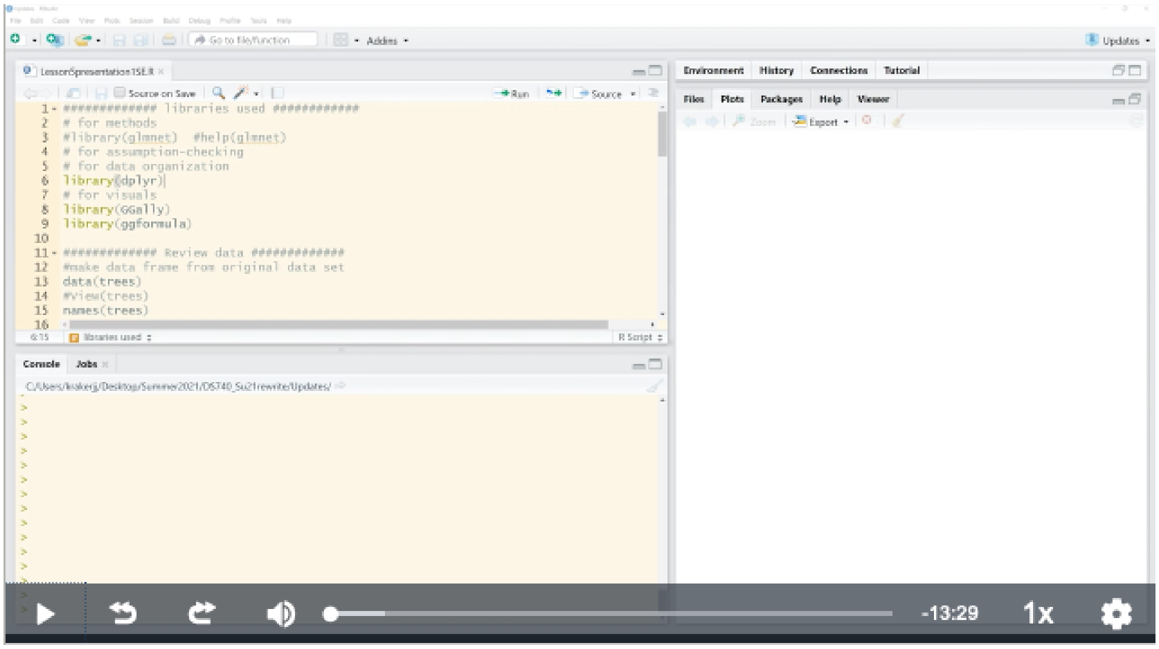
- Notes on usage:
  1. Form is a matrix, not a data frame
     ```
     data(iris)
     x1 = model.matrix(Petal.Length ~.,data=iris); x1
     ```
  2. First column includes all values *1*, designating the intercept
     ```
     x1[,1]
     ```
  3. Allows us to see levels of any categorical predictors in *indicator form*
     ```
     x1[1:10,]  #6 columns, intercept + 3 numeric + 2 indicators
     ```
  4. Efficient way to get only predictor variables in matrix form → required input format for glmnet, after remove initial *1*'s column
     ```
     x = model.matrix(Petal.Length ~.,data=iris)[,-1]; x
     ```

Producing a matrix of the predictor or x values is actually quite simple. There is a function in R that allows us to do that called model dot matrix. And we implement this by setting up a model as we normally would in, for example, linear regression, a model specification, and then identifying our data set.

What results from this, for example, if we run the lines you see in item one is actually a matrix, not a data frame. So we're not able to call on the variable names, but we are produced an array with rows and columns that we can reference. And we see that in note two where we pull the first column of our matrix, which we stored in x1, and that first column contains only the value 1 all the way down the entire column. This designates the intercept.

In note three, we take a look at the first 10 rows of this matrix to get an idea of the content. So we have the intercept, the three numeric columns, three columns representing the numeric values of the numeric variables, plus indicators for two of the three levels of the categorical predictor species. So this is under the model that we're fitting petal length on all the remaining variables.

The final note is what we'll actually-- is how we'll actually have to implement this for use in penalized regression. That is, we'll have to take off the first column by using the minus 1 designation of the column to get an x matrix without that initial intercept column.



```
############## libraries used ############
# for methods
#library(glmnet)  #help(glmnet)
# for assumption-checking
# for data organization
library(dplyr)
# for visuals
library(GGally)
library(ggformula)

############## Review data #############
#make data frame from original data set
data(trees)
#View(trees)
names(trees)
```

```
C:/Users/krakerj/Desktop/Summer2021/DS740_Su21rewrite/Updates/
>
>
>
>
>
>
>
>
>
>
>
>
```

This slide represents a video/screencast in the lecture. The transcript does not substitute video content.

Today, we're going to further examine the relationships among predictors by looking at the correlations between them. An effect that will also be taking a look at is the behavior of the standard errors of the estimated coefficients when using those predictors in a linear model for a numeric response. To start, we're going to be using a data organization library and a couple plotting libraries, as well as a data set that we've looked at before with three variables. Volume will be our response. And then we have two possible predictors, height and girth.

And when we take a look at these variables, we notice that the relationship between our potential response variable and our two possible predictor variables, so volume against girth. Volume against height. Both look positive and linear. But girth appears to be much more useful for predicting volume than does height. And so simply put, we're going to see a larger correlation if we take a look at the correlation between volume and girth, a correlation of about 0.967. Between volume and height is only about 0.598.

However, when we take a closer look at some of these plots, it appears that there might be a bit of a curve in terms of how volume relates to either height or girth. And so even though when we do an original linear model fit, girth is very clearly a highly significant predictor, and height is only marginally significant, we might want to take a look at additional ways that we could bring those predictors into a model. And so we're going to take trees transformed as our new data frame, simply pulling y to the front and then adding in a couple different variables.

So we're going to use the mutate function to define girth height, which is an interaction term between girth and height. Meaning that the relationship of volume to girth may differ for different values of height, girth squared, and girth squared times height. The girth squared is perhaps evident because there's just a little bit of curvature here. The girth squared times height is something to put a little bit more thought into. But effectively, it's a geometric interpretation. And so we're using practical physical characteristics of girth and height to suggest what ways that could be best used to predict volume.

So we define the data frame. Now, we fit on all five of those predictors. Very oddly, none of the predictors are now significant, at least not significant after having fit the remaining predictors. So the marginal effects of any one predictor aren't significant, but some subsets of the predictors are clearly very significant. We have a very high r-squared value, a very significant overall test. So clearly, the model's working, but maybe we don't need all the predictors.

Now, why is it that this comes about? Why is it that the predictor terms are not significant? Well, we're going to take a look at standard errors.

When we talk about standard errors, the standard errors are actually already given as part of the summary output. Specifically, when we take a look at something like girth squared, we see the estimated coefficient, as well as the standard error of that coefficient. And it's the standard errors of these coefficients that really lead to a result in the non-significance because the t-test statistic is the ratio of the coefficient divided by its standard error or estimated standard deviation.

And this test statistic is small enough so that it's not at all significant. And that's because if the standard error is large relative to the point estimate, we're going to get a small t. And that results in a non-significant result.

It would be helpful just to take a closer look at the definition of standard errors to get a better feel for this. And to do this, we're going to take a look at our model matrix. And the model matrix is the mathematical underlying construct that's being used to fit the model. That is, our program, specifically through the LM function, is using matrix manipulations on this predictor matrix.

Now, if I take a look at the predictor matrix, the function needed is called model.matrix-- I'm going to put this definition first-- model.matrix, the model and appropriate data set. And so if I were to run that, it's going to give me the full predictor matrix. And this is for the original data set, trees without any of the transformations.

You'll note that we've got a column for girth, a column for height, and then this column at the start, that includes an intercept. Again, this has to do with the mathematical way that this model is fit. What's important is that most model fitting methods already account for that intercept. And so, oftentimes, instead of wanting to take a look at the model matrix, which here we're going to store in x1.orig, we might want to take a look at the model matrix without that first column of one. So now, if I take a look at x.orig, it's just the columns with the different predictors of interest.

I also define my y response. And take a look at some of the correlations among the predictors, as well as with the response. And these are the same numbers that we see over here at the right.

As we said, we can see standard errors. So if I go back to summary of orig fit, I'll be able to see the standard errors of the different predictor variables. And we also note that the estimate, at least the estimated coefficient for girth, relative to standard error, is quite large, meaning we get a large test statistic and thus, a very significant result.

Now, where are these standard errors coming from? Again they're simply a mathematical transformation. And if you are comfortable with matrices, you can visualize this by first computing s squared, which is the estimated variance of the residuals. And then multiplying that by the diagonal of a matrix transformation of the x with the column of ones at the front.

Again, that's only if you're comfortable with matrices. But if you want to take a closer look at that, you actually get the matrix formulation. You take the square root of s squared times the diagonal of that matrix transformation. And if we were to go back and take a look at those standard errors, those standard errors precisely match this direct computation because that is, effectively, what the LM function is doing.

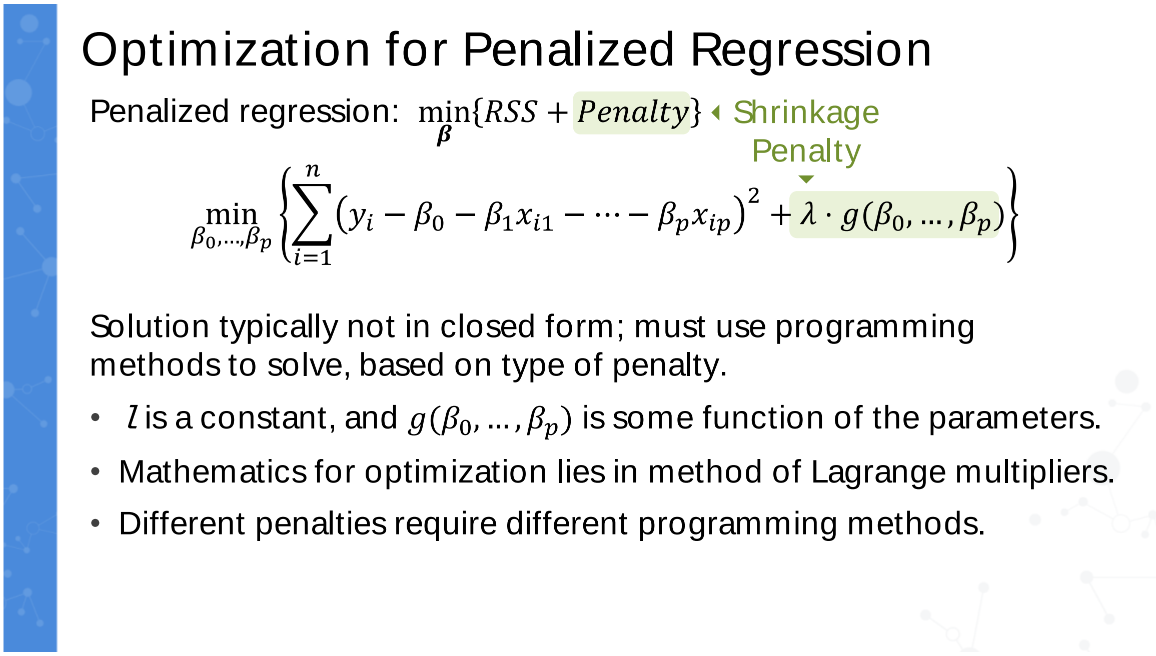
So the final step is to take a look at the application when we have the five predictors including our transformed predictor variables. So I similarly define some model matrices with x full and x1 full, without and with the column of ones, respectively. I may wish to take a look at correlations again, or I can just display all of that at once. Rather than writing out these matrices separately, I can display all of this at once in a ggpairs visualization. And it now displays with my five predictors and my response variable at the end.

OK, so what are we seeing here? Well, the major observation that I would point out is the predictors, visually speaking, are highly correlated with some of the other predictors. And the obvious reason for that is there are direct transformations.

So bearing that in mind, we're going to get very strong multicollinearity among the predictors here. And that means we don't need all of them. It also means that this will come out in the effect on the standard errors. That is, we'll amplify the standard errors because of this high multicollinearity among the predictor variables. And we'll see, in the next presentation, how we address that through penalized regression.

Notes:

See the online course for a downloadable R file containing the set of commands used in this demonstration.



Optimization for Penalized Regression

Penalized regression: $\min_{\boldsymbol{\beta}}\{RSS + Penalty\}$ Shrinkage Penalty

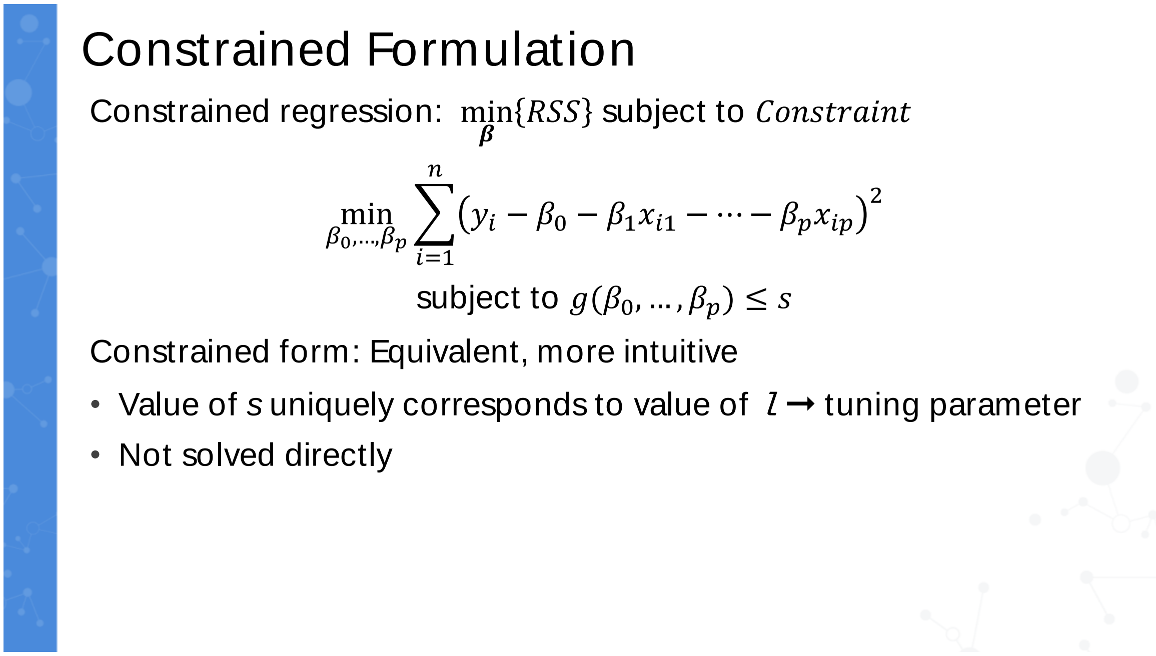
$$\min_{\beta_0,\ldots,\beta_p}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\beta_1x_{i1}-\cdots-\beta_px_{ip}\right)^2+\lambda\cdot g(\beta_0,\ldots,\beta_p)\right\}$$

Solution typically not in closed form; must use programming methods to solve, based on type of penalty.

- l is a constant, and $g(\beta_0,\ldots,\beta_p)$ is some function of the parameters.
- Mathematics for optimization lies in method of Lagrange multipliers.
- Different penalties require different programming methods.

Penalized regression coefficient estimates are mathematically obtained in the same way as multiple regression estimates, by minimizing a goal function involving the residual sum of squares. However, there is now an addition to that goal function, a penalty on the coefficients. This is known as a shrinkage penalty, since the net result is generally to shrink coefficients towards the origin. That is, to overall make their size smaller.

The generic form of the shrinkage penalty is the constant lambda times some function of the coefficients. This function g as denoted above will help define what kinds of values parameter estimates can take. The originating mathematics is based on Lagrange multipliers. And the computation of this minimization is achieved with programming methods appropriate to the form, and is quite a bit more complicated for some methods than for others.



Constrained Formulation

Constrained regression: $\min_{\boldsymbol{\beta}}\{RSS\}$ subject to $Constraint$

$$\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip}\right)^2$$

subject to $g(\beta_0, \ldots, \beta_p) \leq s$

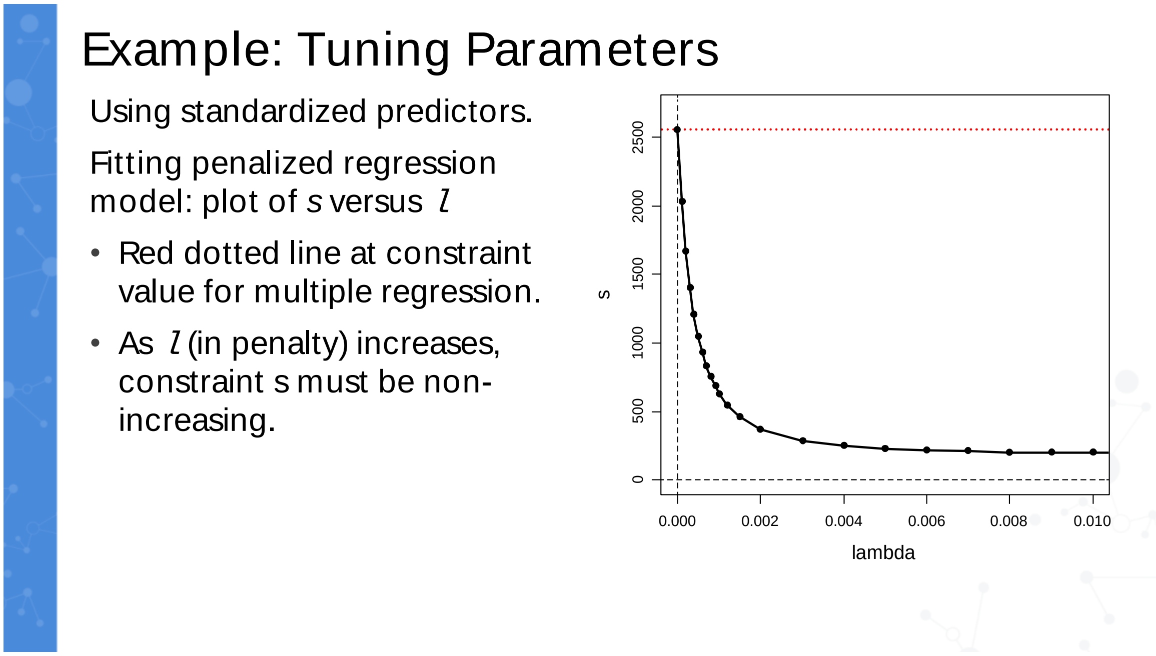
Constrained form: Equivalent, more intuitive

- Value of s uniquely corresponds to value of l → tuning parameter
- Not solved directly

A more intuitive but equivalent way to view this process for estimating coefficients is via the constrained formulation of the optimization from the previous page. Here, we select the betas, the beta j's, to minimize the residual sum of squares, but constrained by the function g of the betas being less than or equal to some value s.

In other words, rather than including a penalty in the minimization, we have the minimization as with multiple linear regression, but constrained by the function g of the betas being less than or equal to s. This value s that is used is known as a tuning parameter for the constrained regression. And it is uniquely corresponding to a value of the tuning parameter lambda in the penalized form.

While this is the more understandable version, the penalized form is the more useful one for computation purposes.



Example: Tuning Parameters

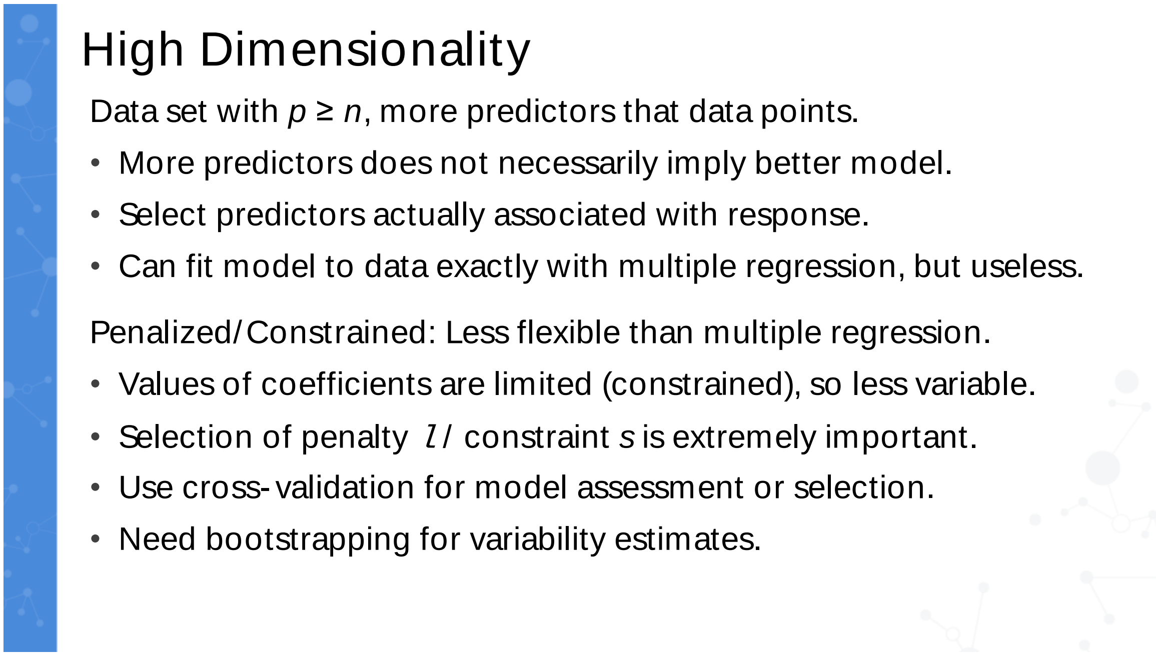
Using standardized predictors.

Fitting penalized regression model: plot of *s* versus *l*

- Red dotted line at constraint value for multiple regression.
- As *l* (in penalty) increases, constraint s must be non-increasing.

Any constraint tuning parameter value s must be less than the constraint function computed at the multiple regression solution. Therefore, restricting the coefficients more requires that we pick a smaller constraint value s. And this, in turn, corresponds to a larger penalty, tuning parameter lambda. This is visualized in the plot at the right.

So for a larger lambda and a smaller s, these correspond to a smaller value of the constraint function calculated for the coefficients. These relationships between the tuning parameters lambda and s apply to the various penalized methods we will discuss in the next lecture.



High Dimensionality

Data set with $p \geq n$, more predictors that data points.

- More predictors does not necessarily imply better model.
- Select predictors actually associated with response.
- Can fit model to data exactly with multiple regression, but useless.

Penalized/Constrained: Less flexible than multiple regression.

- Values of coefficients are limited (constrained), so less variable.
- Selection of penalty l/ constraint s is extremely important.
- Use cross-validation for model assessment or selection.
- Need bootstrapping for variability estimates.

Penalized and constrained regression models will become particularly important with high dimensionality. In particular, if we happen to have data with more predictors than data points, multiple linear regression could fit the model exactly to the data, but would be completely useless in making predictions.

So this implies that we want to enough predictors to do a good job predicting the response, and we want them put together in the model in the appropriate way with the appropriate coefficients. But ideally, we'd also like to select only the predictors that are actually associated with the response. And some penalized or constrained methods can do that automatically.

Penalized or constrained methods work very well with high dimensionality, because they're actually less flexible than multiple regression. And because there is a constraint upon the values of the coefficients, there is less variability in the estimates. But when fitting such a penalized model, the selection of the tuning parameter is extremely important, and is part of the model selection.

Because there is less assumptions when working with penalized models, regression models, there are also fewer theoretical results. And thus, we have to use the data properly through cross-validation for model assessment or selection. And we'll have to estimate the variability of the coefficients through bootstrapping.

Summary

We have seen issues with multiple linear regression with collinear predictors and high numbers of predictors.

An alternative to classical subset selection methods is to use a penalized regression method. Such methods add a constraint on the predictor values, trading some bias for potentially large reductions in variance.

Computationally intensive methods are required for selection of tuning parameter and for estimation of variability of coefficients in penalized regression model fits.