432 Class 11

https://thomaselove.github.io/432-2024

2024-02-20

## Today’s Agenda

* Variable (Feature) Selection in Linear and Logistic Regression Models
* New Packages to install: bestglm and glmnet
* Linear Regression and Prostate Cancer Data
  + Stepwise Regression and its (many) problems
  + Feature Selection via “Best Subsets” and bestglm()
* Logistic Regression and Pima Indians Diabetes Data
* Ridge Regression, the Lasso and the Elastic Net via glmnet

## Today’s R Setup

knitr::opts\_chunk$set(comment = NA)  
  
library(janitor)  
library(broom)  
library(gt)  
library(car) ## for vif  
  
library(bestglm) ## new package - identifies "best subsets"  
library(glmnet) ## new package - ridge regression, elastic net, lasso  
  
library(tidyverse)  
  
theme\_set(theme\_bw())

# Linear Regression Work

## The prost data

The prost.csv file I will work with today is explained and discussed at length in Chapters 15 and 16 of our Course Notes.

prost <- read\_csv("c11/data/prost.csv", show\_col\_types = FALSE) |>   
 clean\_names() |>  
 mutate(across(where(is\_character), as\_factor)) |>  
 mutate(subject = as.character(subject))  
  
dim(prost)

[1] 97 10

Our outcome of interest is lpsa. The other eight variables (besides subject) are candidate predictors, with bph and gleason being three-level factors, and svi being either 0 or 1. The other candidate predictors are quantitative.

## The prost data

There are no missing data in prost.

prost

# A tibble: 97 × 10  
 subject lpsa lcavol lweight age bph svi lcp gleason pgg45  
 <chr> <dbl> <dbl> <dbl> <dbl> <fct> <dbl> <dbl> <fct> <dbl>  
 1 1 -0.431 -0.580 2.77 50 Low 0 -1.39 6 0  
 2 2 -0.163 -0.994 3.32 58 Low 0 -1.39 6 0  
 3 3 -0.163 -0.511 2.69 74 Low 0 -1.39 7 20  
 4 4 -0.163 -1.20 3.28 58 Low 0 -1.39 6 0  
 5 5 0.372 0.751 3.43 62 Low 0 -1.39 6 0  
 6 6 0.765 -1.05 3.23 50 Low 0 -1.39 6 0  
 7 7 0.765 0.737 3.47 64 Medium 0 -1.39 6 0  
 8 8 0.854 0.693 3.54 58 High 0 -1.39 6 0  
 9 9 1.05 -0.777 3.54 47 Low 0 -1.39 6 0  
10 10 1.05 0.223 3.24 63 Low 0 -1.39 6 0  
# ℹ 87 more rows

## Checking for Collinearity

Are the candidate predictors strongly correlated?

* Check the linear regression using main effects for all eight candidate predictors (“kitchen sink” model.)

model\_ks <- lm(lpsa ~ lcavol + lweight + age + bph +   
 svi + lcp + gleason + pgg45, data = prost)  
  
vif(model\_ks) ## from the car package

GVIF Df GVIF^(1/(2\*Df))  
lcavol 2.162180 1 1.470435  
lweight 1.521074 1 1.233318  
age 1.368718 1 1.169922  
bph 1.524371 2 1.111150  
svi 2.018209 1 1.420637  
lcp 3.146281 1 1.773776  
gleason 2.657470 2 1.276783  
pgg45 2.750207 1 1.658375

## A Kitchen-Sink Model

model\_ks <- lm(lpsa ~ lcavol + lweight + age + bph +   
 svi + lcp + gleason + pgg45, data = prost)  
model\_ks

Call:  
lm(formula = lpsa ~ lcavol + lweight + age + bph + svi + lcp +   
 gleason + pgg45, data = prost)  
  
Coefficients:  
(Intercept) lcavol lweight age bphMedium bphHigh   
 0.116841 0.544314 0.702238 -0.023858 0.364036 0.248790   
 svi lcp gleason7 gleason> 7 pgg45   
 0.710949 -0.119312 0.273843 0.053097 0.003985

* Could also fit with ols() and/or include non-linear terms.
* There are 11 coefficients here, including the intercept.

## model\_ks uses 10 degrees of freedom

A rule some use for a minimum sample size () to fit a single linear model with coefficients besides the intercept (i.e. degrees of freedom) is .

* By that rule, with 10 df in our kitchen sink model, we’d need at least observations, just to fit that one model with some accuracy.
* The prost data set has available data points to both fit and evaluate potential models.

## So, the model is too big?



## Could we use stepwise regression?

step(model\_ks)   
 ## output omitted here

* Here, step() fits **31** models of various sizes, landing on a 5-predictor model omitting lcp, gleason and pgg45.

How many observations did we really need to do this well?

* If the models we fit averaged df (probably an underestimate with backwards stepwise regression), our “rule” suggests at least per model, times 31 models, so a minimum .

## Why Not Use Stepwise Procedures?

From Frank Harrell: (google for many, many more)

1. The for a model selected via step() is biased, high.
2. The coefficient estimates (too far from 0) and standard errors (too small) are biased.
3. In simulated stepwise analyses of prediction models, the final model represented noise 20-74% of the time.
4. In simulations, the final stepwise model usually contained less than half of the actual number of real predictors.

## Most Devastating Criticisms

1. Stepwise variable selection encourages the analyst not to think.
   * This is also mostly true for the other methods we’ll discuss today, to be fair.
2. All of the problems specified on the previous slide have no reasonable solutions which work across a broad range of modeling scenarios.

Key Takeaway for Today: Automated feature selection is a troubling enterprise.

# Using the bestglm() package (and function) to select “Best Subsets” from candidate predictors

## Setting Up Our Candidates

To use the bestglm package as it’s designed, we need to set up our data so that:

* we present all of our candidate predictors first,
  + all of which must be either *numeric* or *factor* variables,
* then our outcome variable last,
  + which must be *numeric* for a linear regression
* and then convert this from a tibble to a data frame.

## Our Candidates for a Linear Model

In our case, we’ll do the following.

prost\_set <- prost |>   
 select(-subject) |> ## include only candidate predictors and outcome  
 relocate(lpsa, .after = pgg45) |> ## place outcome last  
 data.frame() ## convert from tibble to data frame, only

Have we included all eight predictors, then the outcome?

names(prost\_set)

[1] "lcavol" "lweight" "age" "bph" "svi" "lcp" "gleason"  
[8] "pgg45" "lpsa"

dim(prost\_set)

[1] 97 9

## Selecting an Information Criterion

Then, we’ll need to decide which of several available information criteria (IC) we might use.

* We’ll choose between AIC and BIC in 432.
* Other options include BIC and BIC, which we won’t use in 432.

We have other options we can tweak as well, but we’ll follow the approach on the next slide pretty closely for doing “best subsets” selection for a linear model.

## Best Subsets Selection: AIC or BIC

Here, we’ll run the bestglm() function on our prostate data, to identify the top 3 fitting linear models (with 0 (intercept only) to all (8 in this case) predictors), using AIC.

best\_lin\_AIC <-   
 bestglm(Xy = prost\_set, family = gaussian, IC = "AIC",   
 method = "exhaustive", TopModels = 3, nvmax = "default")

Morgan-Tatar search since factors present with more than 2 levels.

Next, we’ll do the same thing, but using BIC as our criterion.

best\_lin\_BIC <-   
 bestglm(Xy = prost\_set, family = gaussian, IC = "BIC",  
 method = "exhaustive", TopModels = 3, nvmax = "default")

Morgan-Tatar search since factors present with more than 2 levels.

## Which model does AIC like best?

Which of the potential subsets of our 8 predictors does bestglm() package identify as having the best (lowest) AIC?

best\_lin\_AIC

AIC  
Best Model:  
 Df Sum Sq Mean Sq F value Pr(>F)   
lcavol 1 69.00 69.00 142.198 < 2e-16 \*\*\*  
lweight 1 7.17 7.17 14.781 0.000225 \*\*\*  
age 1 0.65 0.65 1.331 0.251770   
bph 2 2.15 1.07 2.214 0.115220   
svi 1 5.27 5.27 10.870 0.001400 \*\*   
Residuals 90 43.67 0.49   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

* Suggests a 5-predictor model, leaving out lcp, gleason and pgg45. (Note: same as step() result.)

## Top Model at each size, via AIC?

What are the best subsets at each predictor count?

cbind(preds = rownames(best\_lin\_AIC$Subsets), best\_lin\_AIC$Subsets) |>  
 gt() |> tab\_options(table.font.size = 20)

| preds | Intercept | lcavol | lweight | age | bph | svi | lcp | gleason | pgg45 | logLikelihood | AIC |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | -13.41878 | 26.83755 |
| 1 | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | 24.18302 | -46.36603 |
| 2 | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | 30.47923 | -56.95846 |
| 3 | TRUE | TRUE | TRUE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | 35.58872 | -65.17744 |
| 4 | TRUE | TRUE | TRUE | FALSE | FALSE | TRUE | FALSE | TRUE | FALSE | 37.28942 | -64.57884 |
| 5\* | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | 38.70157 | -65.40314 |
| 6 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | FALSE | 40.52069 | -65.04139 |
| 7 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | 40.31968 | -64.63936 |
| 8 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | 41.69765 | -63.39529 |

## Best 3 Models, according to AIC?

best\_lin\_AIC$BestModels |> gt() |> tab\_options(table.font.size = 24)

| lcavol | lweight | age | bph | svi | lcp | gleason | pgg45 | Criterion |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | -65.40314 |
| TRUE | TRUE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | -65.17744 |
| TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | FALSE | -65.04139 |

1. Use lcavol, lweight, svi, plus age and bph (5 predictors)
2. Use lcavol, lweight, svi. (3 predictors)
3. Use lcavol, lweight, svi, plus age, bph and gleason (6 predictors)

## Which model does BIC think is best?

best\_lin\_BIC

BIC  
Best Model:  
 Df Sum Sq Mean Sq F value Pr(>F)   
lcavol 1 69.00 69.00 137.80 < 2e-16 \*\*\*  
lweight 1 7.17 7.17 14.32 0.000272 \*\*\*  
svi 1 5.17 5.17 10.33 0.001798 \*\*   
Residuals 93 46.57 0.50   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

* BIC suggests a 3-predictor model, leaving out age and bph in addition to lcp, gleason and pgg45.
* Note: This was the #2 choice via AIC, and was the best-fitting model by AIC among models with three predictors.

## Top Model at each size, via BIC?

Best subsets of predictors via BIC for each model size?

cbind(preds = rownames(best\_lin\_BIC$Subsets), best\_lin\_BIC$Subsets) |>  
 gt() |> tab\_options(table.font.size = 20)

| preds | Intercept | lcavol | lweight | age | bph | svi | lcp | gleason | pgg45 | logLikelihood | BIC |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | -13.41878 | 26.83755 |
| 1 | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | 24.18302 | -43.79132 |
| 2 | TRUE | TRUE | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | 30.47923 | -51.80904 |
| 3\* | TRUE | TRUE | TRUE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | 35.58872 | -57.45330 |
| 4 | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | FALSE | FALSE | FALSE | 36.24195 | -54.18505 |
| 5 | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | FALSE | FALSE | TRUE | 37.13312 | -51.39268 |
| 6 | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | TRUE | 37.80106 | -48.15385 |
| 7 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | 40.31968 | -44.04167 |
| 8 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | 41.69765 | -37.64818 |

## Best 3 Models, according to BIC?

best\_lin\_BIC$BestModels |> gt() |> tab\_options(table.font.size = 24)

| lcavol | lweight | age | bph | svi | lcp | gleason | pgg45 | Criterion |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRUE | TRUE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | -57.45330 |
| TRUE | TRUE | TRUE | FALSE | TRUE | FALSE | FALSE | FALSE | -54.18505 |
| TRUE | TRUE | FALSE | FALSE | TRUE | FALSE | FALSE | TRUE | -53.92715 |

1. Use lcavol, lweight, svi. (3 predictors)
2. Use lcavol, lweight, svi, plus age (4 predictors)
3. Use lcavol, lweight, svi, plus pgg45 (4 predictors)

# “Best Subsets” in logistic regression

## The Pima Indians Diabetes Data

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. All patients here are females at least 21 years old of Pima Indian heritage.

The data includes 8 medical predictors (next slide) for one outcome variable, diabetes, which is either 1 (subject has diabetes) or 0 (subject does not).

## Pima Diabetes Predictors

| Variable | Description |
| --- | --- |
| age | Age (years) |
| bmi | Body Mass Index () |
| dbp | Diastolic Blood Pressure (mm Hg) |
| glucose | Plasma glucose concentration at 2 hours in an oral glucose tolerance test |
| insulin | 2-hour serum insulin (mu U/ml) |
| pedig | Diabetes Pedigree Function |
| preg | Number of times pregnant |
| triceps | Triceps Skin fold thickness (mm) |

## Ingesting the data

pimadm <- read\_csv("c11/data/pima\_diabetes.csv",   
 show\_col\_types = FALSE) |>  
 clean\_names() |>  
 rename(preg = pregnancies, dbp = blood\_pressure,   
 triceps = skin\_thickness,   
 pedig = diabetes\_pedigree\_function)  
  
dim(pimadm)

[1] 768 9

## Setting up for bestglm(), 1

We need a data frame, showing the candidate predictors (numerical or factor), then the outcome (as a factor).

pimadm

# A tibble: 768 × 9  
 age bmi dbp glucose insulin pedig preg triceps diabetes  
 <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 50 33.6 72 148 0 0.627 6 35 1  
 2 31 26.6 66 85 0 0.351 1 29 0  
 3 32 23.3 64 183 0 0.672 8 0 1  
 4 21 28.1 66 89 94 0.167 1 23 0  
 5 33 43.1 40 137 168 2.29 0 35 1  
 6 30 25.6 74 116 0 0.201 5 0 0  
 7 26 31 50 78 88 0.248 3 32 1  
 8 29 35.3 0 115 0 0.134 10 0 0  
 9 53 30.5 70 197 543 0.158 2 45 1  
10 54 0 96 125 0 0.232 8 0 1  
# ℹ 758 more rows

## Setting up for bestglm(), 2

Here, we just need to convert diabetes to a factor, and then convert the tibble to a data frame.

pimadm\_df <- pimadm |>  
 mutate(diabetes =   
 fct\_recode(factor(diabetes), "Yes" = "1", "No" = "0")) |>  
 as.data.frame()

## bestglm() for Logistic Models

We’ll run the bestglm() function on our pimadm\_df data, to identify the top 3 fitting logistic models (with 0 (intercept only) to all (8 here) predictors), first using AIC, and then using BIC.

best\_pima\_AIC <-  
 bestglm(Xy = pimadm\_df, family = binomial, IC = "AIC",  
 method = "exhaustive", TopModels = 3, nvmax = "default")

Morgan-Tatar search since family is non-gaussian.

best\_pima\_BIC <-  
 bestglm(Xy = pimadm\_df, family = binomial, IC = "BIC",  
 method = "exhaustive", TopModels = 3, nvmax = "default")

Morgan-Tatar search since family is non-gaussian.

## AIC’s favorite model

best\_pima\_AIC

AIC  
BICq equivalent for q in (0.910337349179392, 0.965036759857444)  
Best Model:  
 Estimate Std. Error z value Pr(>|z|)  
(Intercept) -8.405136208 0.7167032628 -11.727498 9.214195e-32  
age 0.014788838 0.0092896771 1.591965 1.113926e-01  
bmi 0.090088589 0.0144619078 6.229371 4.683116e-10  
dbp -0.013213574 0.0051536754 -2.563913 1.034996e-02  
glucose 0.035112252 0.0036624713 9.587038 9.064975e-22  
insulin -0.001157035 0.0008141589 -1.421142 1.552755e-01  
pedig 0.947595358 0.2980062755 3.179783 1.473853e-03  
preg 0.123172450 0.0320687734 3.840884 1.225919e-04

* Seven predictors (all but triceps)

## BIC’s favorite model

best\_pima\_BIC

BIC  
BICq equivalent for q in (0.169717182537119, 0.610343314895634)  
Best Model:  
 Estimate Std. Error z value Pr(>|z|)  
(Intercept) -8.41585098 0.656907771 -12.811313 1.417163e-37  
bmi 0.07809694 0.013770941 5.671140 1.418501e-08  
glucose 0.03382636 0.003345272 10.111690 4.903090e-24  
pedig 0.90129355 0.291696408 3.089834 2.002682e-03  
preg 0.14192631 0.027105325 5.236104 1.640012e-07

* Four predictors (drop age, dbp, insulin, triceps)

## AIC: Top 3 Models

best\_pima\_AIC$BestModels |> gt() |> tab\_options(table.font.size = 24)

| age | bmi | dbp | glucose | insulin | pedig | preg | triceps | Criterion |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | 737.4534 |
| TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | 737.4617 |
| FALSE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | 737.9706 |

1. bmi, dbp, glucose, pedig, preg, age and insulin.
2. bmi, dbp, glucose, pedig, preg, and age.
3. bmi, dbp, glucose, pedig, preg, and insulin.

## BIC: Top 3 Models

best\_pima\_BIC$BestModels |> gt() |> tab\_options(table.font.size = 24)

| age | bmi | dbp | glucose | insulin | pedig | preg | triceps | Criterion |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FALSE | TRUE | FALSE | TRUE | FALSE | TRUE | TRUE | FALSE | 760.8810 |
| FALSE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | 761.7786 |
| FALSE | TRUE | FALSE | TRUE | FALSE | FALSE | TRUE | FALSE | 764.0563 |

1. bmi, glucose, preg, and pedig.
2. bmi, glucose, preg, pedig and dbp
3. bmi, glucose, and preg.

## AIC: Top Models by Size

cbind(preds = rownames(best\_pima\_AIC$Subsets), best\_pima\_AIC$Subsets) |>  
 gt() |> tab\_options(table.font.size = 20)

| preds | Intercept | age | bmi | dbp | glucose | insulin | pedig | preg | triceps | logLikelihood | AIC |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | -496.7420 | 993.4839 |
| 1 | TRUE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | -404.3598 | 810.7196 |
| 2 | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | -385.7015 | 775.4030 |
| 3 | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | FALSE | TRUE | FALSE | -372.0625 | 750.1249 |
| 4 | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | TRUE | TRUE | FALSE | -367.1529 | 742.3059 |
| 5 | TRUE | FALSE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | -364.2798 | 738.5596 |
| 6 | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | -362.7308 | 737.4617 |
| 7\* | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | -361.7267 | 737.4534 |
| 8 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | -361.7227 | 739.4454 |

## BIC: Top Models by Size

cbind(preds = rownames(best\_pima\_BIC$Subsets), best\_pima\_BIC$Subsets) |>  
 gt() |> tab\_options(table.font.size = 20)

| preds | Intercept | age | bmi | dbp | glucose | insulin | pedig | preg | triceps | logLikelihood | BIC |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | TRUE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | FALSE | -496.7420 | 993.4839 |
| 1 | TRUE | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | -404.3598 | 815.3634 |
| 2 | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | FALSE | FALSE | FALSE | -385.7015 | 784.6906 |
| 3 | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | FALSE | TRUE | FALSE | -372.0625 | 764.0563 |
| 4\* | TRUE | FALSE | TRUE | FALSE | TRUE | FALSE | TRUE | TRUE | FALSE | -367.1529 | 760.8810 |
| 5 | TRUE | FALSE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | -364.2798 | 761.7786 |
| 6 | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | TRUE | TRUE | FALSE | -362.7308 | 765.3244 |
| 7 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | FALSE | -361.7267 | 769.9600 |
| 8 | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | TRUE | -361.7227 | 776.5957 |

# Is this really a solution? (Introduction to glmnet package)

## Feature Selection as commonly done

All subsets / best subsets / stepwise methods either include a variable or drop it from the model. Often, this choice is based on only a tiny difference in fit quality.

* Harrell: not reasonable to assume that a population regression coefficient would be exactly zero just because it failed to meet a criterion for significance.
* Efron: this approach is “overly greedy, impulsively eliminating covariates which are correlated with other covariates.”

## Feature Selection as commonly done

* Greenland: Variable selection does more damage to confidence interval widths than to point estimates.
* Greenland: Stepwise variable selection on confounders leaves important confounders uncontrolled.
* Greenland: Shrinkage approaches (like ridge regression and the lasso) are far superior to variable selection.

So, what’s the alternative?

## Lasso, Ridge Regression, Elastic Net

These methods are particularly useful when we believe the effects are sparse, in the sense that we believe that few of the many predictors we are evaluating have a meaningful effect.

Consider, for instance, the analysis of gene expression data, where we have good reason to believe that only a small number of genes have an influence on our response of interest.

Or, in medical claims data, where we can have thousands of available codes to search through that may apply to some of the people included in a large analysis relating health care costs to outcomes.

## The glmnet package

We’ll use the glmnet package to fit models via penalized maximum likelihood, which incorporate some coefficient shrinkage, using

* the *lasso* ( parameter = 1)
* *ridge regression* ( parameter = 0)
* the *elastic net* ( = 0.5), combining the lasso and ridge regression

to minimize mean-squared error of estimation (in linear regression) and deviance (in logistic).

## Role of the parameter

We’ll use K-fold cross-validation to determine the value of , which is the penalty used in building these parsimonious models.

* When , we have ordinary least squares.
* When is large, all coefficients will shrink towards zero.

## Set up our prost data for glmnet

names(prost)

[1] "subject" "lpsa" "lcavol" "lweight" "age" "bph" "svi"   
 [8] "lcp" "gleason" "pgg45"

Create a data matrix of the predictors (I’ll call it pred\_x here), and then a data matrix of the outcome (which I’ll call out\_y.)

pred\_x <- prost |> select(lcavol:pgg45) |> as.matrix()  
out\_y <- prost |> select(lpsa) |> as.matrix()

In the next three slides, we’ll fit three different models (changing ), and we’ll use 10-fold cross-validation to select the parameter based on minimizing mean squared error.

## Fit a lasso model for lpsa

set.seed(4321)  
pros\_cv1 <- cv.glmnet(pred\_x, out\_y, type.measure = "mse", nfolds = 10)

Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion

Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion

pros\_lasso <- glmnet(pred\_x, out\_y, alpha = 1, lambda = pros\_cv1$lambda.min)

Warning in storage.mode(xd) <- "double": NAs introduced by coercion

tidy(pros\_lasso) |> gt() |> tab\_options(table.font.size = 20) |>  
 fmt\_number(columns = estimate:dev.ratio, decimals = 3)

| term | step | estimate | lambda | dev.ratio |
| --- | --- | --- | --- | --- |
| (Intercept) | 1 | -0.278 | 0.027 | 0.645 |
| lcavol | 1 | 0.504 | 0.027 | 0.645 |
| lweight | 1 | 0.658 | 0.027 | 0.645 |
| age | 1 | -0.008 | 0.027 | 0.645 |
| svi | 1 | 0.559 | 0.027 | 0.645 |
| pgg45 | 1 | 0.003 | 0.027 | 0.645 |

* This model includes 5 predictors (omitting bph, lcp and gleason) and also shrinks the coefficients towards zero.

## Fit an elastic net model for lpsa

set.seed(4322)  
pros\_cv2 <- cv.glmnet(pred\_x, out\_y, type.measure = "mse", nfolds = 10)

Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion

Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion

pros\_elnet <- glmnet(pred\_x, out\_y, alpha = 0.5,   
 lambda = pros\_cv2$lambda.min)

Warning in storage.mode(xd) <- "double": NAs introduced by coercion

tidy(pros\_elnet) |> gt() |> tab\_options(table.font.size = 20) |>  
 fmt\_number(columns = estimate:dev.ratio, decimals = 3)

| term | step | estimate | lambda | dev.ratio |
| --- | --- | --- | --- | --- |
| (Intercept) | 1 | -0.235 | 0.001 | 0.652 |
| lcavol | 1 | 0.556 | 0.001 | 0.652 |
| lweight | 1 | 0.745 | 0.001 | 0.652 |
| age | 1 | -0.016 | 0.001 | 0.652 |
| svi | 1 | 0.685 | 0.001 | 0.652 |
| lcp | 1 | -0.097 | 0.001 | 0.652 |
| pgg45 | 1 | 0.006 | 0.001 | 0.652 |

* 6 predictor model (omitting bph and gleason)

## Fit a ridge regression model for lpsa

set.seed(4323)  
pros\_cv3 <- cv.glmnet(pred\_x, out\_y, type.measure = "mse", nfolds = 10)

Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion  
  
Warning in storage.mode(xd) <- "double": NAs introduced by coercion

Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion  
  
Warning in cbind2(1, newx) %\*% nbeta: NAs introduced by coercion

pros\_ridge <- glmnet(pred\_x, out\_y, alpha = 0,   
 lambda = pros\_cv3$lambda.min)

Warning in storage.mode(xd) <- "double": NAs introduced by coercion

tidy(pros\_ridge) |> gt() |> tab\_options(table.font.size = 20) |>  
 fmt\_number(columns = estimate:dev.ratio, decimals = 3)

| term | step | estimate | lambda | dev.ratio |
| --- | --- | --- | --- | --- |
| (Intercept) | 1 | -0.224 | 0.030 | 0.652 |
| lcavol | 1 | 0.531 | 0.030 | 0.652 |
| lweight | 1 | 0.732 | 0.030 | 0.652 |
| age | 1 | -0.015 | 0.030 | 0.652 |
| svi | 1 | 0.663 | 0.030 | 0.652 |
| lcp | 1 | -0.073 | 0.030 | 0.652 |
| pgg45 | 1 | 0.005 | 0.030 | 0.652 |

* Same terms here (different shrinkage) as elastic net.

## Setting up for glmnet in a logistic regression setting

pimadm

# A tibble: 768 × 9  
 age bmi dbp glucose insulin pedig preg triceps diabetes  
 <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 50 33.6 72 148 0 0.627 6 35 1  
 2 31 26.6 66 85 0 0.351 1 29 0  
 3 32 23.3 64 183 0 0.672 8 0 1  
 4 21 28.1 66 89 94 0.167 1 23 0  
 5 33 43.1 40 137 168 2.29 0 35 1  
 6 30 25.6 74 116 0 0.201 5 0 0  
 7 26 31 50 78 88 0.248 3 32 1  
 8 29 35.3 0 115 0 0.134 10 0 0  
 9 53 30.5 70 197 543 0.158 2 45 1  
10 54 0 96 125 0 0.232 8 0 1  
# ℹ 758 more rows

pima\_x <- pimadm |> select(-diabetes) |> as.matrix()  
pima\_y <- pimadm |> select(diabetes) |> as.matrix() # number, not factor

## Lasso results under logistic fit

Cross-validate so as to minimize residual deviance…

set.seed(1234)  
pima\_cv1 <- cv.glmnet(pima\_x, pima\_y, type.measure = "deviance", nfolds = 25)  
pima\_lasso <- glmnet(pima\_x, pima\_y, family = binomial, alpha = 1,   
 lambda = pima\_cv1$lambda.min)  
tidy(pima\_lasso)

# A tibble: 8 × 5  
 term step estimate lambda dev.ratio  
 <chr> <dbl> <dbl> <dbl> <dbl>  
1 (Intercept) 1 -8.08 0.00338 0.271  
2 age 1 0.0136 0.00338 0.271  
3 bmi 1 0.0839 0.00338 0.271  
4 dbp 1 -0.0111 0.00338 0.271  
5 glucose 1 0.0337 0.00338 0.271  
6 insulin 1 -0.000866 0.00338 0.271  
7 pedig 1 0.856 0.00338 0.271  
8 preg 1 0.116 0.00338 0.271

* Omits triceps and shrinks coefficients

## Elastic Net results under logistic fit

set.seed(4325)  
pima\_cv2 <- cv.glmnet(pima\_x, pima\_y, type.measure = "deviance", nfolds = 10)  
pima\_elnet <- glmnet(pima\_x, pima\_y, alpha = 0.5,   
 lambda = pima\_cv2$lambda.min)  
tidy(pima\_elnet)

# A tibble: 8 × 5  
 term step estimate lambda dev.ratio  
 <chr> <dbl> <dbl> <dbl> <dbl>  
1 (Intercept) 1 -0.836 0.00447 0.303  
2 age 1 0.00251 0.00447 0.303  
3 bmi 1 0.0129 0.00447 0.303  
4 dbp 1 -0.00210 0.00447 0.303  
5 glucose 1 0.00580 0.00447 0.303  
6 insulin 1 -0.000137 0.00447 0.303  
7 pedig 1 0.141 0.00447 0.303  
8 preg 1 0.0201 0.00447 0.303

## Ridge regression for logistic fit

set.seed(4312354)  
pima\_cv3 <- cv.glmnet(pima\_x, pima\_y, type.measure = "deviance", nfolds = 15)  
pima\_ridge <- glmnet(pima\_x, pima\_y, alpha = 0,   
 lambda = pima\_cv3$lambda.min)  
tidy(pima\_ridge)

# A tibble: 9 × 5  
 term step estimate lambda dev.ratio  
 <chr> <dbl> <dbl> <dbl> <dbl>  
1 (Intercept) 1 -0.847 0.00447 0.303  
2 age 1 0.00266 0.00447 0.303  
3 bmi 1 0.0131 0.00447 0.303  
4 dbp 1 -0.00228 0.00447 0.303  
5 glucose 1 0.00586 0.00447 0.303  
6 insulin 1 -0.000172 0.00447 0.303  
7 pedig 1 0.146 0.00447 0.303  
8 preg 1 0.0204 0.00447 0.303  
9 triceps 1 0.000145 0.00447 0.303

* Here, no coefficients dropped, just shrunk.

# Conclusions and advice

## Minimizing the chance of overfitting

So, what **should** we be thinking about when confronted with a situation where a new model is under development, and we have some data and a lot of predictors to consider?

1. Pre-specify well-motivated predictors and how to model them.
2. Eliminate predictors without using the outcome.

## Minimizing the chance of overfitting

1. Use the outcome, but cross-validate the target measure of prediction error.
2. Use the outcome, and **shrink** the coefficient estimates.

This last comment applies to things like our “best subsets” approach as well as standard stepwise procedures.

## Next Time

* More on K-Fold Cross Validation in linear regression
* Creating Table 1 with the tableone package
* Quiz 1 coming your way at 5 PM Thursday.