432 Class 13 Slides

github.com/THOMASELOVE/2019-432

2019-03-19

Setup

```
library(skimr); library(rms); library(broom)
library(MASS); library(GGally); library(janitor)
library(lars)
library(tidyverse)
```

Today's Materials

Beyond Feature Selection

- Shrinkage
- Ridge Regression
- The Lasso

The PTSD study (See Class 09 Slides)

The `maleptsd` data

The `maleptsd` file on our web site contains information on PTSD (post traumatic stress disorder) symptoms following childbirth for 64 fathers[1]. There are ten predictors and the response is a measure of PTSD symptoms. The raw, untransformed values (`ptsd_raw`) are right skewed and contain zeros, so we will work with a transformation, specifically, `ptsd = log(ptsd_raw + 1)` as our outcome, which also contains a lot of zeros.

```
maleptsd <- read_csv("data/maleptsd.csv") %>%
  clean_names() %>%
  mutate(ptsd = log(ptsd_raw + 1))
```

---

[1]Source: Ayers et al. 2007 *J Reproductive and Infant Psychology*. The data are described in more detail in Wright DB and London K (2009) *Modern Regression Techniques Using R* Sage Publications.

```
maleptsd %>% select(-id, -ptsd_raw) %>%
skim()
```

Scatterplot Matrix, part 1

Scatterplot Matrix, part 2
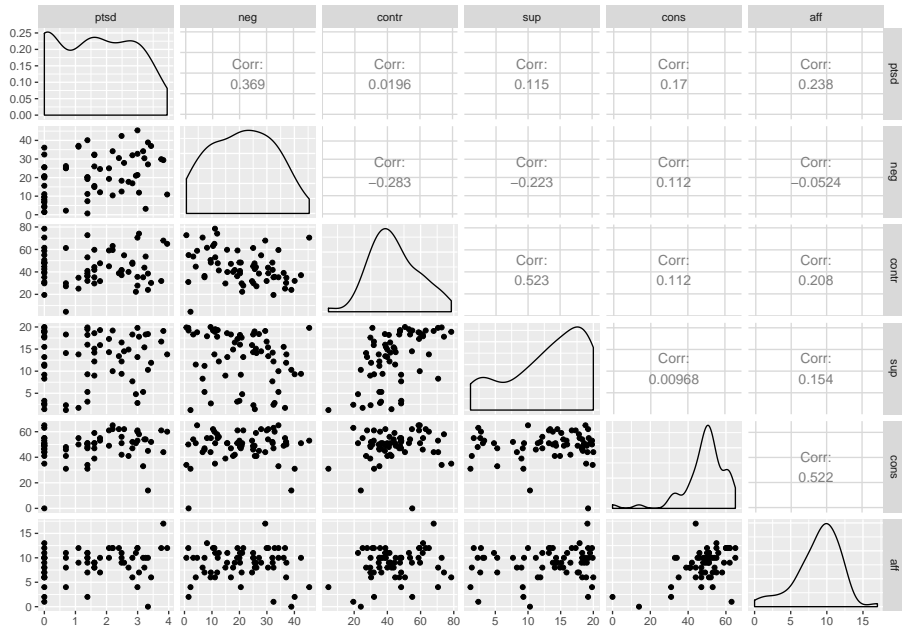
A Kitchen Sink Model

Kitchen Sink Model: PTSD

With 64 observations, a kitchen sink model with 10 predictors (not counting the intercept) is clearly overfit, but we'll take a look at collinearity and some related issues first using that model.

```
d <- datadist(maleptsd)
options(datadist = "d")
m_ks <- ols(ptsd ~ over2 + over3 + over5 + bond + posit +
                   neg + contr + sup + cons + aff,
            data=maleptsd, x=TRUE, y=TRUE)
```

`m_ks` output, part 1

```
> m_ks
Linear Regression Model

ols(formula = ptsd ~ over2 + over3 + over5 + bond + posit + neg +
    contr + sup + cons + aff, data = maleptsd, x = TRUE, y = TRUE)

                Model Likelihood      Discrimination
                  Ratio Test              Indexes
Obs        64   LR chi2      23.27   R2       0.305
sigma1.1553    d.f.            10    R2 adj   0.174
d.f.       53   Pr(> chi2) 0.0098    g        0.790

Residuals

     Min       1Q    Median       3Q       Max
-2.06186 -0.93217   0.08441  0.80816   2.80545
```

`m_ks` output, part 2

```
           Coef    S.E.    t    Pr(>|t|)
Intercept  1.7373  1.6058  1.08  0.2842
over2     -0.0713  0.0580 -1.23  0.2238
over3      0.1188  0.0645  1.84  0.0712
over5     -0.1235  0.1332 -0.93  0.3579
bond      -0.0880  0.0551 -1.60  0.1165
posit      0.0249  0.0205  1.21  0.2310
neg        0.0360  0.0168  2.14  0.0367
contr     -0.0086  0.0132 -0.65  0.5180
sup        0.0412  0.0327  1.26  0.2132
cons       0.0146  0.0171  0.85  0.3966
aff        0.0367  0.0639  0.57  0.5689
```

Assessing collinearity in an `ols` object

```
rms::vif(m_ks)
```

```
    over2    over3    over5     bond    posit      neg
1.767484 1.931723 1.501908 1.347394 2.409917 1.788657
    contr      sup     cons      aff
1.798061 1.740154 1.724590 1.826600
```

There are several VIF functions. With an `ols` object, you want the one in `rms`.

Conclusions?

So, the model is too big?

432 Class 13 Slides

Four strategies for 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.
3. Use the outcome, but cross-validate the target measure of prediction error.
4. Use the outcome, and **shrink** the coefficient estimates.

Stepwise Regression

```
mod_ks <- lm(ptsd ~ over2 + over3 + over5 + bond + posit +
                    neg + contr + sup + cons + aff,
             data=maleptsd)

step(mod_ks)
```

Stepwise Results (edited)

Backwards Elimination starting with the Kitchen Sink

1. AIC starts at 28.41 for Kitchen Sink
2. Drop aff to get to AIC = 26.80
3. Drop contr to move to AIC = 25.42
4. Drop over5 to move to AIC = 24.54
5. Drop posit to move to AIC = 23.87
6. Drop over2 to move to AIC = 23.75
7. No improvements available.

This yields:

```
lm(ptsd ~ over3 + bond + neg + sup + cons, data = maleptsd)
```

Why Not Use Stepwise Procedures?

From Harrell:

1. The R^2 for a model selected in a stepwise manner is biased, high.
2. The coefficient estimates and standard errors are biased.
3. The *p* values for the individual-variable t tests are too small.
4. In stepwise analyses of prediction models, the final model represented noise 20-74% of the time.
5. In stepwise analyses, the final model usually contained less than half of the actual number of real predictors.
6. It is not logical that a population regression coefficient would be exactly zero just because its estimate was not statistically significant.

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

Attacking a flaw of "Best Subsets" or Stepwise Approaches

The best subsets methods we have studied either include a variable or drop it from the model. Often, this choice is based on only a tiny difference in the quality of a fit to data.

- 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."
- 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.
- Greenland: Variable selection does more damage to confidence interval widths than to point estimates.

So, what's the alternative?

Ridge Regression

Ridge Regression

Ridge regression involves a more smooth transition between useful and not useful predictors which can be obtained by constraining the overall size of the regression coefficients.

Ridge regression assumes that the regression coefficients (after normalization) should not be very large. This is reasonable to assume when you have lots of predictors and you believe *many* of them have some effect on the outcome.

Pros:

1. Some nice statistical properties
2. Can be calculated using only standard least squares approaches, so it's been around for a while.
3. Available in the MASS package.

Ridge Regression

Ridge regression takes the sum of the squared estimated standardized regression coefficients and constrains that sum to only be as large as some value k.

$$\sum \hat{\beta_j}^2 \leq k.$$

The value k is one of several available measures of the amount of shrinkage, but the main one used in the MASS package is a value λ. As λ increases, the amount of shrinkage goes up, and k goes down.

Assessing a Ridge Regression Approach

We'll look at a plot produced by the `lm.ridge` function for a ridge regression for the Male PTSD study.

- Several (here 101) different values for λ, our shrinkage parameter, will be tested.
- Results are plotted so that we see the coefficients across the various (standardized) predictors.
  - Each selection of a λ value implies a different vector of covariate values across the predictors we are studying.
  - The idea is to pick a value of λ for which the coefficients seem relatively stable.

Code for our Ridge Regression

```
preds <- with(maleptsd, cbind(over2, over3, over5, bond,
                              posit, neg, contr, sup,
                              cons, aff))
### requires MASS package
x <- lm.ridge(maleptsd$ptsd~preds,
              lambda=seq(0, 100, by=1))
plot(x)
title("Ridge Regression")
abline(h=0)
```

Usually, you need to use trial and error to decide the range of λ to be tested. Here, seq(0, 100, by=1) means going from 0 (no shrinkage) to 100 in steps of 1.

Resulting Ridge Regression Plot

Ridge Regression for the Male PTSD Data
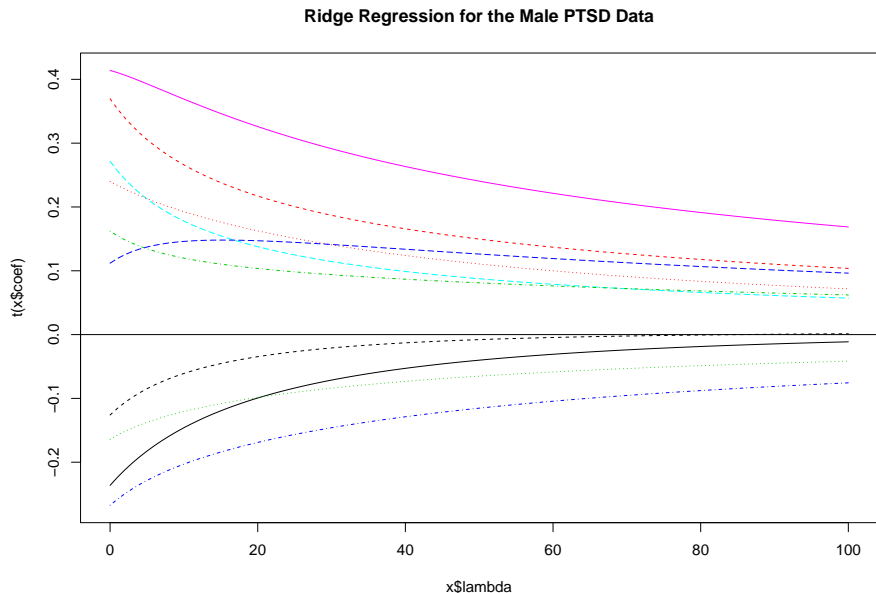
The `lm.ridge` plot - where do coefficients stabilize?

Does $\lambda = 50$ seem like a stable spot here?

```r
x <- lm.ridge(maleptsd$ptsd~preds, lambda=seq(0, 100, by=1))
plot(x)
title("Ridge Regression for the Male PTSD Data")
abline(h=0)
abline(v=50, lty=2, col="black")
```

Does $\lambda = 50$ seem like a stable spot here?

Ridge Regression for the Male PTSD Data
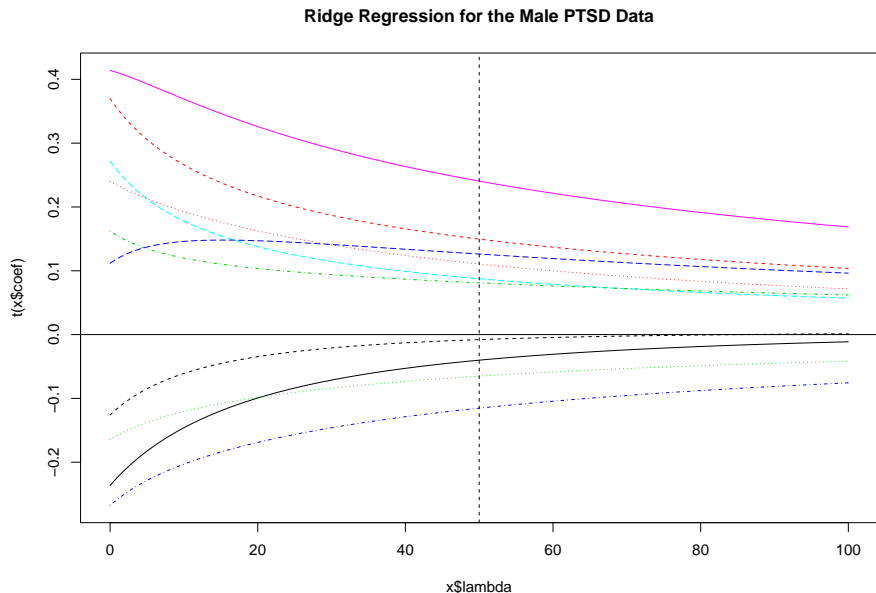
Coefficient values at $\lambda = 50$

The coefficients at $\lambda = 50$ can be determined from the `lm.ridge` output. These are fully standardized coefficients. The original predictors are centered by their means and then scaled by their standard deviations and the outcome has also been centered, in these models.

```
round(x$coef[,50],3)
```

```
predsover2 predsover3 predsover5  predsbond predsposit
    -0.041      0.151     -0.066     -0.117      0.089
  predsneg predscontr   predssup  predscons   predsaff
     0.243     -0.008      0.112      0.082      0.127
```

Was an intercept used?

```
x$Inter
```

```
[1] 1
```

Automated way to pick λ

Use the select function in the MASS package, and since select is used by dplyr, for example, you'll have to specifically tell R to use the MASS version.

```
MASS::select(x)
```

```
modified HKB estimator is 16.46749
modified L-W estimator is 22.03396
smallest value of GCV  at 40
```

I'll use the GCV estimate of $\lambda = 40$.

```r
x <- lm.ridge(maleptsd$ptsd~preds, lambda=seq(0, 100, by=1))
plot(x)
title("Ridge Regression for the Male PTSD Data")
abline(h=0)
abline(v=40, lty=2, col="black")
```

I'll use the GCV estimate of $\lambda = 40$.

Ridge Regression for the Male PTSD Data
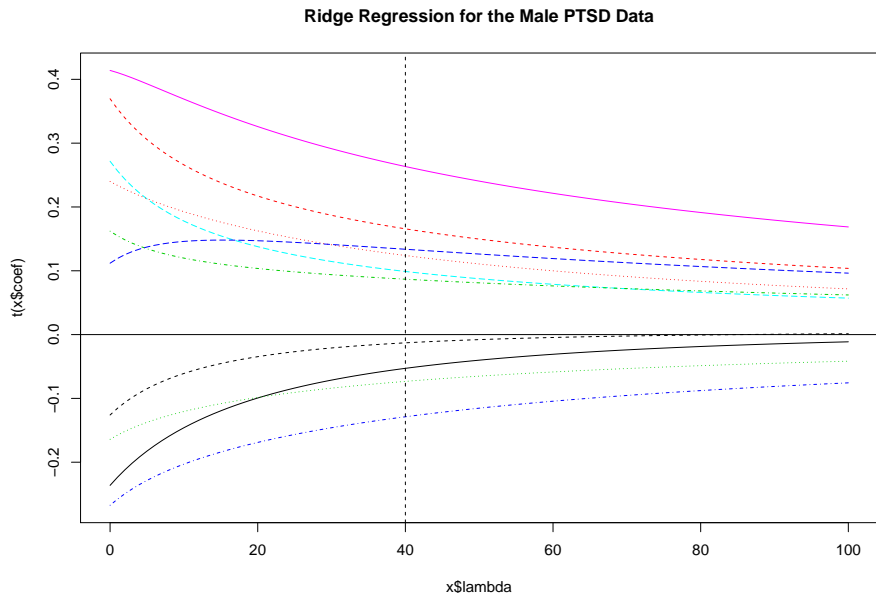
Coefficients at $\lambda = 40$

```
round(x$coef[,40],4) # Ridge Regression (standardized)
```

```
predsover2 predsover3 predsover5  predsbond predsposit
   -0.0544     0.1676    -0.0743    -0.1303     0.1003
  predsneg predscontr    predssup  predscons   predsaff
    0.2659    -0.0136      0.1254     0.0874     0.1345
```

A Scaled Linear Regression Model

```
st.ptsd <- maleptsd %>% select(-id, -ptsd_raw) %>%
    scale() %>% as.data.frame()
mod_ks_sc <- lm(ptsd ~ over2 + over3 + over5 + bond +
                    posit + neg + contr + sup + cons + aff,
                data=st.ptsd)
```

Coefficients at $\lambda = 40$

Ridge Regression at $\lambda = 40$

```r
round(x$coef[,40],4) # Ridge Regression (standardized)
```

```
predsover2 predsover3 predsover5  predsbond predsposit
  -0.0544     0.1676    -0.0743    -0.1303     0.1003
  predsneg predscontr   predssup  predscons   predsaff
   0.2659    -0.0136     0.1254     0.0874     0.1345
```

Linear Regression (standardized variables)

```r
round(mod_ks_sc$coef,4)[-1] # do not show intercept
```

```
  over2    over3    over5     bond    posit      neg    contr
-0.1874   0.2931  -0.1302  -0.2121   0.2155   0.3283  -0.1000
    sup     cons      aff
 0.1903   0.1285   0.0887
```

Ridge Regression Conclusions

The main problem with ridge regression is that all it does is shrink the coefficient estimates, but it's not so useful in practical settings because it still includes all variables.

1. It's been easy to do ridge regression for many years, so you see it occasionally in the literature.
2. It leads to the **lasso**, which incorporates the positive features of shrinking regression coefficients with the ability to wisely select some variables to be eliminated from the predictor pool.

The Lasso

The Lasso

The lasso works by constraining the sum of the **absolute values** of the estimated standardized regression coefficients to be no larger than some value k.

$$\sum |\hat{\beta}_j| \leq k.$$

This looks like a minor change from ridge regression's $\sum \hat{\beta}_j^2 \leq k$ constraint, but it's not.

The Name

The lasso is not an acronym, but rather refers to cowboys using a rope to pull cattle from the herd, much as we will pull predictors from a model.

Consequences of the Lasso Approach

1. In ridge regression, while the individual coefficients shrink and sometimes approach zero, they seldom reach zero and are thus excluded from the model. With the lasso, some coefficients do reach zero and thus, those predictors do drop out of the model.
   - So the lasso leads to more parsimonious models than does ridge regression.
   - Ridge regression is a method of shrinkage but not model selection. The lasso accomplishes both tasks.
2. If k is chosen to be too small, then the model may not capture important characteristics of the data. If k is too large, the model may over-fit the data in the sample and thus not represent the population of interest accurately.
3. The lasso is far more difficult computationally than ridge regression (the problem requires an algorithm called least angle regression, which was published in 2004), although R has a package (lars) which can do the calculations pretty efficiently.

How The Lasso Works

The `lars` package lets us compute the lasso coefficient estimates **and** do cross-validation to determine the appropriate amount of shrinkage. The main tool is a pair of graphs.

1. The first plot (below) shows what coefficients get selected as we move from constraining all of the coefficients to zero (complete shrinkage) towards fewer constraints all the way up to ordinary least squares, showing which variables are included in the model at each point.
2. The second plot (coming soon) suggests where on the first plot we should look for a good model choice, according to a cross-validation approach.

```
## requires lars package
lasso1 <- lars(preds, maleptsd$ptsd, type="lasso")
plot(lasso1)
```

Resulting Lasso Plot 1 (Coefficient Progress)
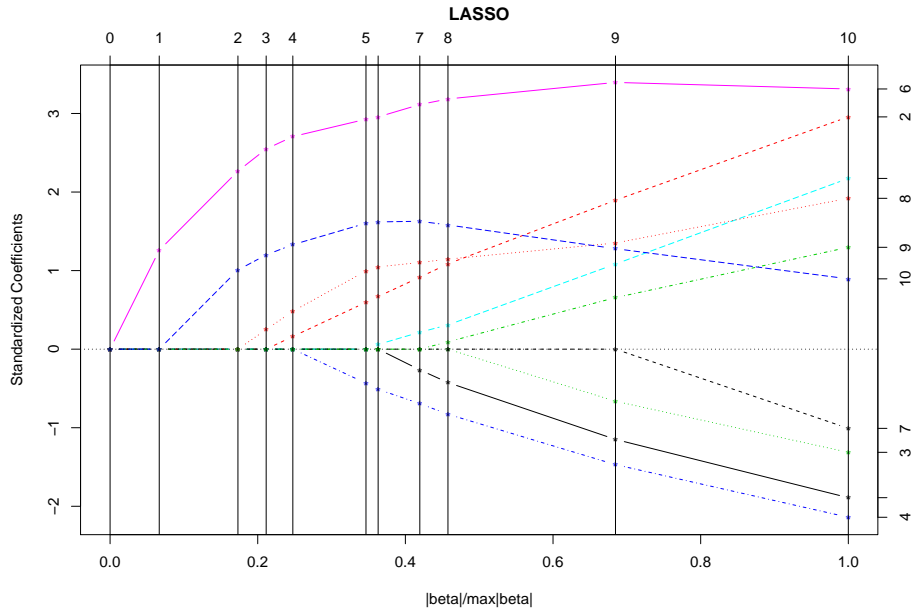
Description of Lasso Plot 1

- The y axis shows standardized regression coefficients.
    - The lars package standardizes all variables so the shrinkage doesn't penalize some coefficients because of their scale.
- The x-axis is labeled |beta|/max|beta|.
    - This ranges from 0 to 1.
    - 0 means that the sum of the $|\hat{\beta}_j|$ is zero (completely shrunk)
    - 1 means the ordinary least squares unbiased estimates.
- The lasso graph starts at constraining all of the coefficients to zero, and then moves toward ordinary least squares.

Identifiers for the predictors (numbers) are shown to the right of the graph.

- The vertical lines in the lasso plot show when a variable has been eliminated from the model, and in fact these are the only points that are actually shown in the default lasso graph.
- The labels on the top of the graph tell you how many predictors are in the model at that stage.

Summary for Lasso Graph 1

```r
summary(lasso1)
```

```
LARS/LASSO
Call: lars(x = preds, y = maleptsd$ptsd, type = "lasso")
   Df     Rss      Cp
0   1 101.751 14.2369
1   2  93.962 10.4010
2   3  85.955  6.4018
3   4  83.928  6.8827
4   5  82.277  7.6461
5   6  78.698  6.9645
6   7  78.223  8.6088
7   8  76.762  9.5140
8   9  75.901 10.8686
9  10  72.330 10.1934
10 11  70.738 11.0000
```
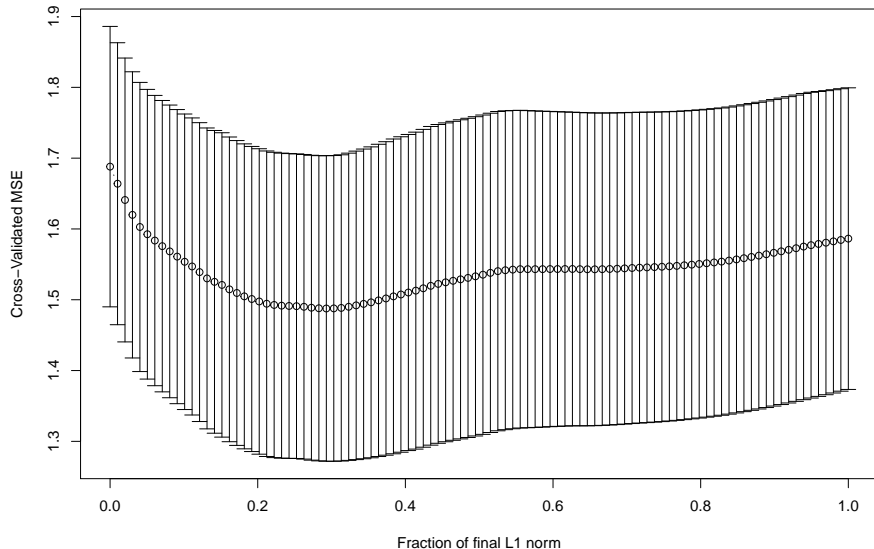
Cross-Validation with the Lasso

Normally, cross-validation methods are used to determine how much shrinkage should be used. We'll use the cv.lars function.

- 10-fold ($K = 10$) cross-validation
    - the data are randomly divided into 10 groups.
    - Nine groups are used to predict the remaining group for each group in turn.
    - Overall prediction performance is computed, and the machine calculates a cross-validation criterion (mean squared error) and standard error for that criterion.

The cross-validation plot is the second lasso plot. We're looking to minimize cross-validated mean squared error in this plot.

```
set.seed(432432)
lassocv <- cv.lars(preds, maleptsd$ptsd, K=10)
## default cv.lars K is 10
```
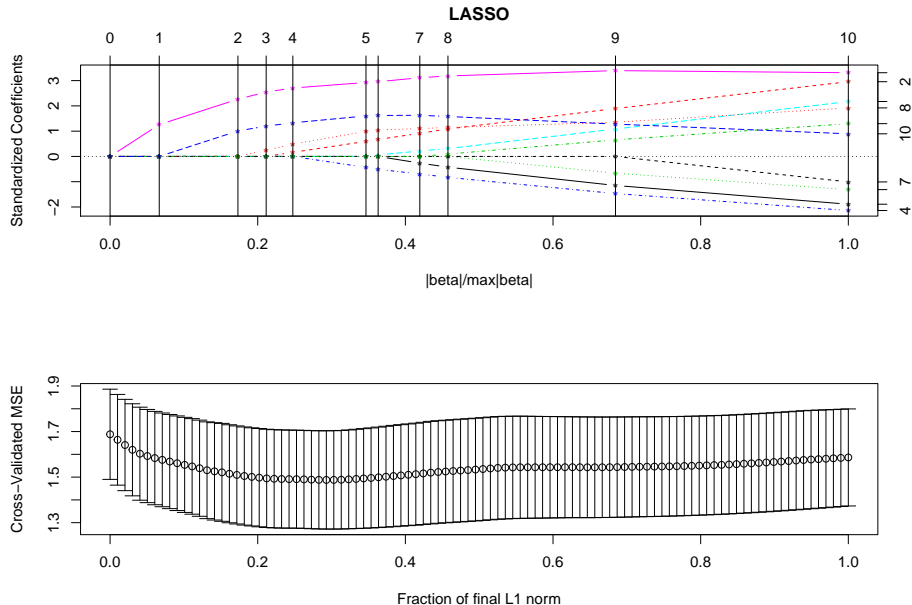
Lasso Graph 2

What value of the key fraction minimizes cross-validated MSE?

```
frac <- lassocv$index[which.min(lassocv$cv)]
frac
```

```
[1] 0.2929293
```

The cross-validation plot suggests we use a fraction of about 0.3, that's suggesting a model with 4-5 predictors, based on the top LASSO plot.

The Plots, Together

Coefficients for the Model via Lasso Cross-Validation

```
coef.cv <- coef(lasso1, s=frac, mode="fraction")
round(coef.cv,4)
```

```
  over2   over3   over5    bond   posit     neg   contr
 0.0000  0.0145  0.0000 -0.0082  0.0000  0.0305  0.0000
    sup    cons     aff
 0.0154  0.0000  0.0596
```

So the model suggested by the lasso includes over3, bond, neg, sup and aff. Note that our "best subsets" model with five predictors used the same five predictors.

Compare to original model (standardized)

```
  over2   over3   over5    bond   posit     neg   contr
-0.1874  0.2931 -0.1302 -0.2121  0.2155  0.3283 -0.1000
    sup    cons     aff
 0.1903  0.1285  0.0887
```

# Obtaining Fitted Values from Lasso

```
fits.cv <- predict.lars(lasso1, preds, s=frac,
                        type="fit", mode="fraction")
head(fits.cv$fit)

[1] 1.180148 1.441451 2.243622 1.704847 2.022155 1.763315
```
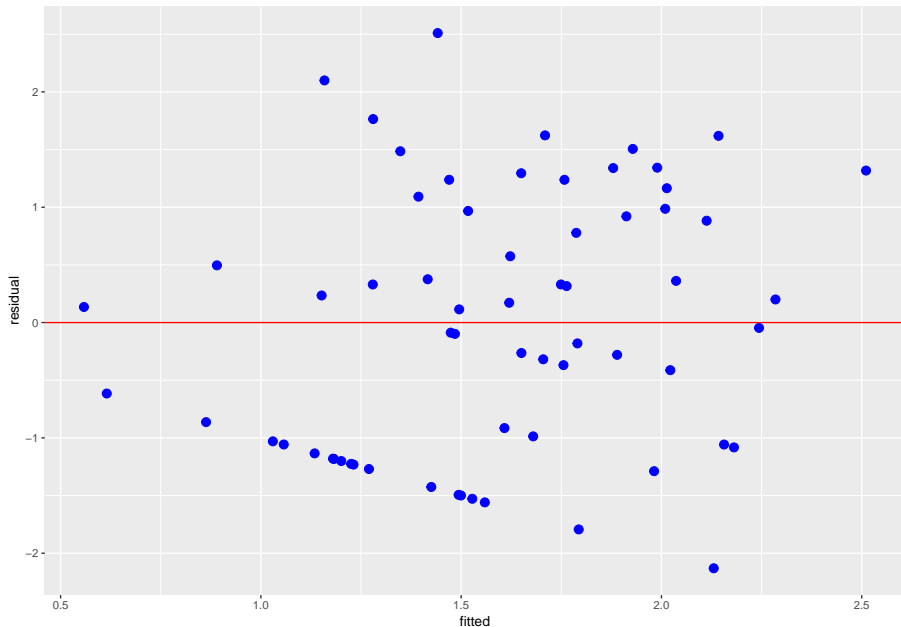
# Compare Observed and Fitted (Lasso) Values of `ptsd`

```
temp_res <- tibble(observed = maleptsd$ptsd,
                   fitted = fits.cv$fit,
                   residual = observed - fitted)
ggplot(temp_res, aes(x = fitted, y = observed)) +
    geom_point(size = 3) +
    geom_abline(intercept = 0, slope = 1, col = "red")
```

Plot Observed and Fitted (Lasso) Values of `ptsd`
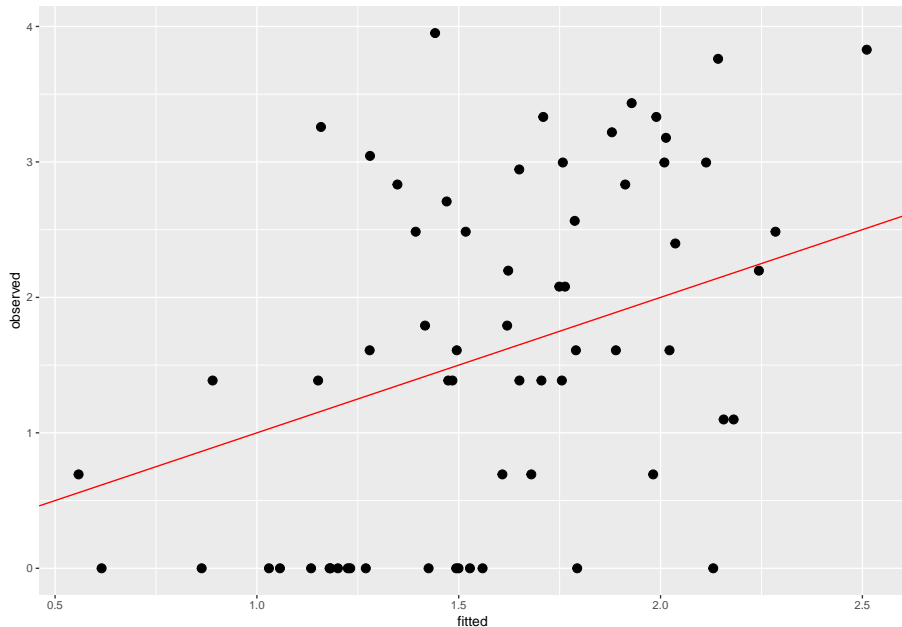
Plot Residuals vs. Fitted (Lasso) Values of `ptsd`

When is the Lasso Most Useful?

As Faraway suggests, the lasso is 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.

Are there other, even fancier, approaches?

Sure. The `glmnet` package is an interesting way to do several sets of model-building activities when the number of predictors is much larger than the sample size, especially if the predictors can be rescaled so as to avoid collinearity. An advantage of `glmnet` is that tidiers from `broom` are available.

Check out, for instance, the elastic net (which bridges the gap between the lasso and ridge regression), and its performance comparison for a simulated study at this link

Or take a look at the `glmnet` package vignette at this link

Next Time

- Logistic Regression on Aggregated Data
- "Best Subsets" for Logistic Regression?
- Probit Regression: A Useful Alternative Link