Homework 3 - Stat 495

Cassandra Jin

Due Monday, Oct. 2 by midnight

Practicing Academic Integrity

If you worked with others or used resources outside of provided course material (anything besides our textbook(s), course materials in Moodle/Git repo, R help menu) to complete this assignment, please acknowledge them below using a bulleted list.

I acknowledge the following individuals with whom I worked on this assignment:

Name(s) and corresponding problem(s)

-

I used the following sources to help complete this assignment:

Source(s) and corresponding problem(s)

-

PROBLEMS TO TURN IN: Additional 1-3

Additional 1 - Applications to College

Adapted from ISLR

```
data(College)
```

This data set contains information from 1995 on US Colleges from US News and World Report (see help file for details). Our goal is to predict the number of applications received (Apps) using the other variables as potential predictors.

> part a: Split the data into training and test data sets. (2/3 - 1/3 split is fine.)

SOLUTION:

```
College <- na.omit(College)

set.seed(495)
n <- nrow(College)
train_index <- sample(1:n, (2/3) * n)
test_index <- setdiff(1:n, train_index)

train <- College[train_index, ]
test <- College[test_index, ]

#further set up for using glmnet
x_train <- model.matrix(Apps ~ ., train)[, -1]
x_test <- model.matrix(Apps ~ ., test)[, -1]

y_train <- train %>%
  select(Apps) %>%
  unlist() %>%
  as.numeric()

y_test <- test %>%
  select(Apps) %>%
  unlist() %>%
  as.numeric()
```

> part b: Fit a "kitchen sink" linear regression model on the training set. Report the test error obtained, and comment on any issues with the model (such as conditions, etc.).

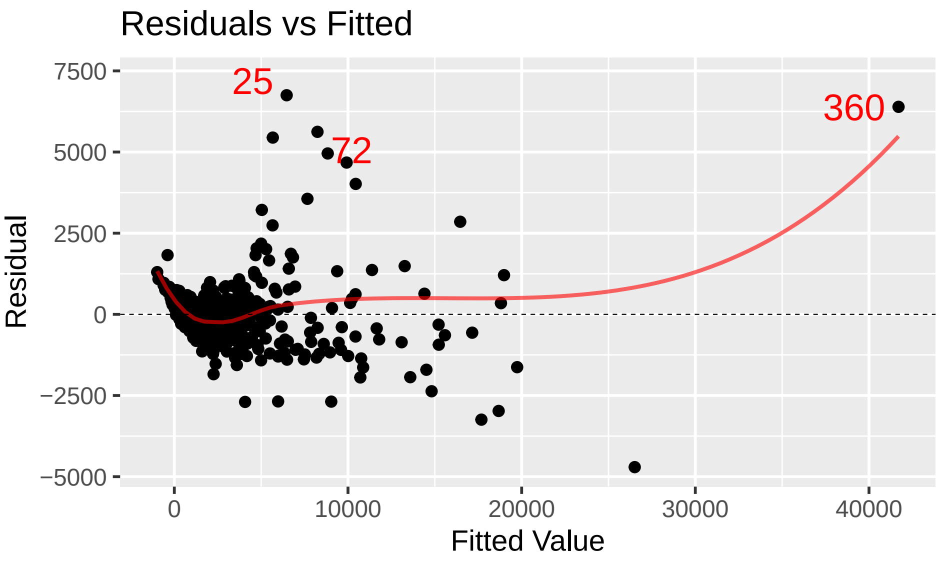
SOLUTION:

```
all_mod <- lm(Apps ~ ., data = train)
msummary(all_mod)
```

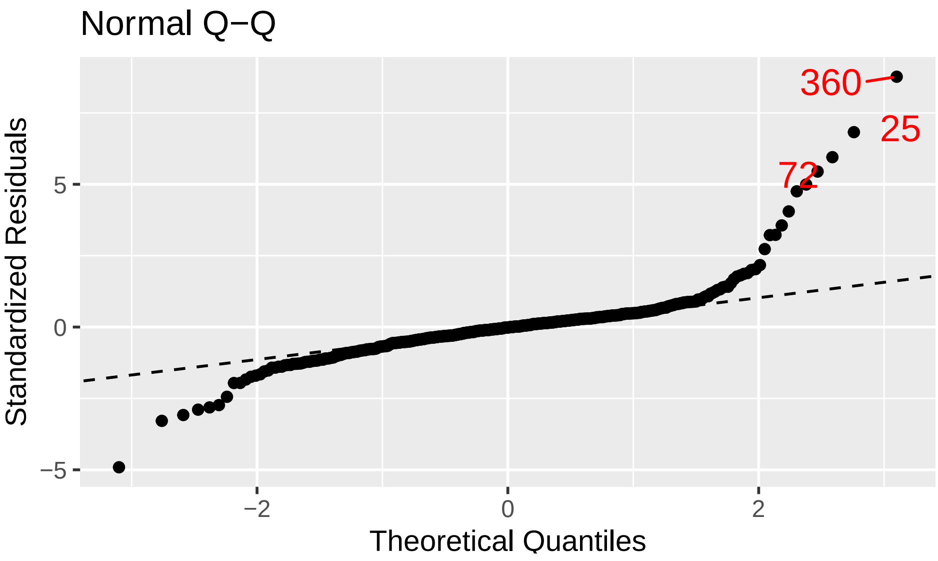
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00e+03   4.97e+02   -2.01    0.045 *
## PrivateYes  -4.12e+02   1.68e+02   -2.45    0.015 *
## Accept       1.68e+00   4.54e-02   36.97  < 2e-16 ***
## Enroll      -1.46e+00   2.26e-01   -6.44  2.8e-10 ***
## Top10perc    2.85e+01   7.09e+00    4.03  6.6e-05 ***
## Top25perc   -6.10e+00   5.54e+00   -1.10    0.271
## F.Undergrad  1.09e-01   3.88e-02    2.82    0.005 **
## P.Undergrad  6.83e-02   3.90e-02    1.75    0.081 .
## Outstate    -1.47e-01   2.37e-02   -6.21  1.1e-09 ***
## Room.Board   8.71e-02   6.04e-02    1.44    0.150
## Books        7.06e-03   2.69e-01    0.03    0.979
## Personal     7.61e-02   7.47e-02    1.02    0.309
## PhD         -1.20e+01   5.52e+00   -2.18    0.030 *
## Terminal    -4.92e+00   6.07e+00   -0.81    0.418
## S.F.Ratio    3.61e+01   1.70e+01    2.12    0.034 *
## perc.alumni  4.01e+00   4.76e+00    0.84    0.400
## Expend       1.93e-01   1.98e-02    9.71  < 2e-16 ***
## Grad.Rate    1.56e+01   3.68e+00    4.22  2.9e-05 ***
##
## Residual standard error: 1020 on 500 degrees of freedom
## Multiple R-squared:  0.94,   Adjusted R-squared:  0.938
## F-statistic:  460 on 17 and 500 DF,  p-value: <2e-16
```

```r
mplot(all_mod, which = 1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
mplot(all_mod, which = 2)
```



```r
all_pred <- predict(all_mod, newx = x_test, exact = T, x = x_train, y = y_train)
mean((all_pred - y_test)^2)
```

```
## [1] 26751721
```

The test MSE is 26751721. The kitchen-sink model does not seem to satisfy the conditions well. We see extreme heteroscedasticity on the residuals vs. fitted plot, as well as a downward then plateauing trend from left to right. For the Q-Q plot, although many points lie solidly on the line for a center section, they begin to stray downward on the left and also upward on the right. These indicate that the model fails to satisfy the conditions of equal variances and normality of errors.

> part c: Fit a linear regression model using an automated variable selection method of your choice (from Stat 230) on the training set. Report the test error obtained, and comment on any issues with the model (such as conditions, etc.).

SOLUTION:

```r
best <- regsubsets(Apps ~ ., data = train, nbest = 1)
with(summary(best), data.frame(rsq, adjr2, cp, rss, outmat))
```

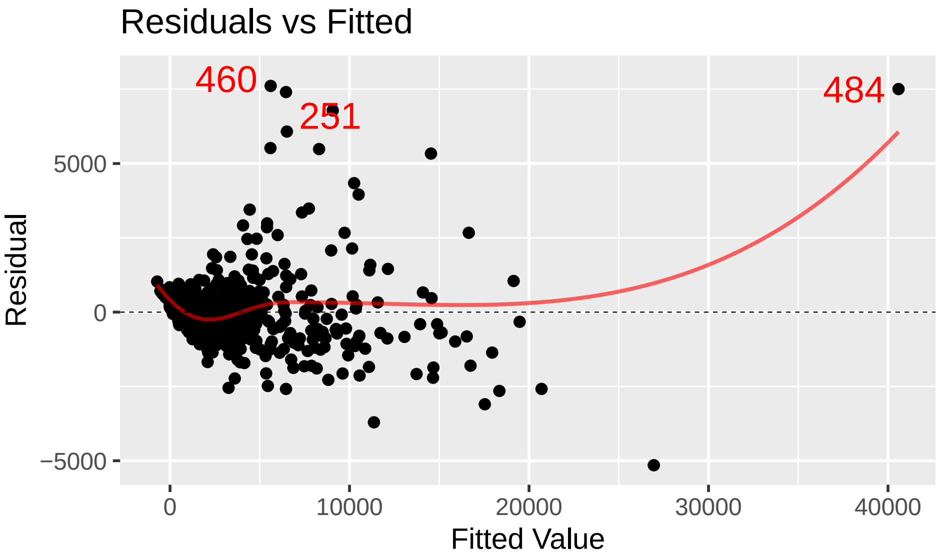
```
##               rsq    adjr2       cp       rss PrivateYes Accept Enroll
## 1  ( 1 ) 0.895322 0.895119 356.4781 899688671                 *
## 2  ( 1 ) 0.920601 0.920293 148.2644 682420832                 *
## 3  ( 1 ) 0.925081 0.924644 113.0116 643917874                 *
## 4  ( 1 ) 0.929142 0.928589  81.2424 609015492                 *      *
## 5  ( 1 ) 0.933090 0.932437  50.4061 575077324                 *      *
## 6  ( 1 ) 0.934601 0.933834  39.8405 562090134                 *      *
## 7  ( 1 ) 0.936073 0.935195  29.6060 549445045                 *      *
## 8  ( 1 ) 0.937185 0.936198  22.3566 539885259                 *      *
##          Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )                                                    *
## 4  ( 1 )                                                    *
## 5  ( 1 )         *                                          *
## 6  ( 1 )         *                                          *
## 7  ( 1 )         *                     *                    *
## 8  ( 1 )         *                     *                    *
##          Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
## 1  ( 1 )
## 2  ( 1 )                                                  *
## 3  ( 1 )                                                  *
## 4  ( 1 )                                                  *
## 5  ( 1 )                                                  *
## 6  ( 1 )                                                  *         *
## 7  ( 1 )                                                  *         *
## 8  ( 1 )            *                                     *         *
```

```r
best_mod <- lm(Apps ~ Accept + Enroll + Top10perc + F.Undergrad + Outstate + PhD + Expend + Grad.Rate, 
msummary(best_mod)
```

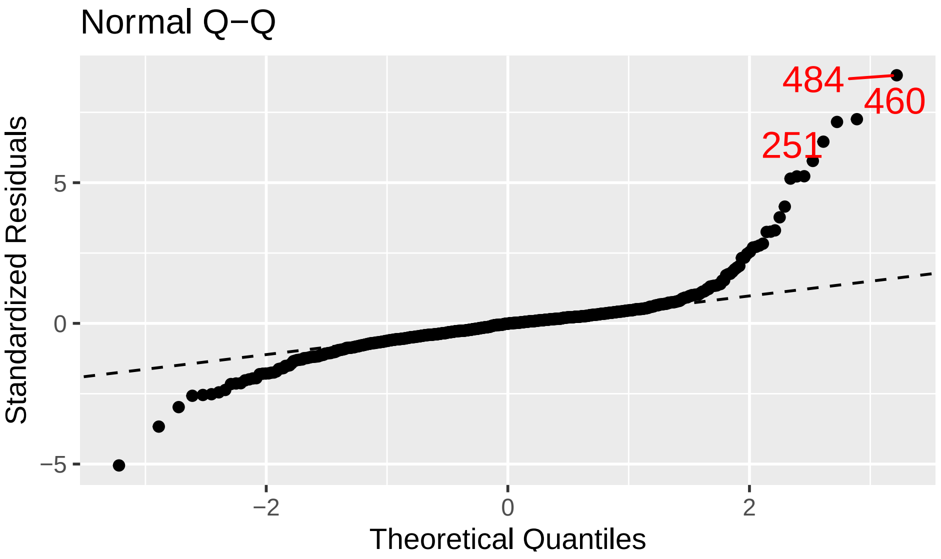
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -445.4969   222.6461   -2.00   0.0458 *
## Accept         1.6005     0.0402   39.84  < 2e-16 ***
## Enroll        -0.9449     0.1860   -5.08 4.7e-07 ***
## Top10perc     33.2431     3.3281    9.99  < 2e-16 ***
## F.Undergrad    0.0907     0.0313    2.90   0.0038 **
## Outstate       -0.0999     0.0158   -6.31  4.6e-10 ***
## PhD            -7.5321     2.9865   -2.52   0.0119 *
## Expend          0.0868     0.0113    7.69  4.6e-14 ***
## Grad.Rate       6.9207     2.8522    2.43   0.0155 *
##
## Residual standard error: 1060 on 768 degrees of freedom
## Multiple R-squared:  0.925,  Adjusted R-squared:  0.925
## F-statistic: 1.19e+03 on 8 and 768 DF,  p-value: <2e-16
```

```
mplot(best_mod, which = 1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
mplot(best_mod, which = 2)
```



```
best_pred <- predict(best_mod, newx = x_test, exact = T, x = x_train, y = y_train)
mean((best_pred - y_test)^2)
```

```
## [1] 25527700
```

To maximize Adjusted R^2 and minimize Cp value, I picked the model with 8 predictors. The test MSE for this model is 25527700, slightly smaller than that of the kitchen-sink model. The best subsets model still does not seem to satisfy the conditions well. There is less of a shaped trend in the points on the residuals vs. fitted plot, but we still see extreme heteroscedasticity. For the Q-Q plot, the points deviate from the line further toward the ends compared to how they behave on the kitchen-sink Q-Q plot, but the points still stray considerably from the line. These indicate that the best subsets model also fails to satisfy the conditions of equal variances and normality of errors.

> part d: Fit a ridge regression model on the training set, with lambda chosen by cross-validation. Report the lambda chosen and the test error obtained.

SOLUTION:

```
set.seed(495)
x <- model.matrix(Apps ~ ., College)[, -1] # trim off the first column
y <- College %>% select(Apps) %>%  unlist() %>%  as.numeric()

grid <- 10^seq(10, -2, length = 100)
ridge_mod <- glmnet(x, y, alpha = 0, lambda = grid)

dim(coef(ridge_mod))
```

```
## [1]  18 100
```

```
# plot(ridge_mod, label = TRUE)

cvridge.out <- cv.glmnet(x_train, y_train, alpha = 0) # Fit ridge regression model on training data
bestlam_ridge <- cvridge.out$lambda.min  # Select lambda that minimizes training MSE
bestlam_ridge
```

```
## [1] 385.428
```

```
# fit model on training set
ridge_mod <- glmnet(x_train, y_train, alpha = 0, lambda = grid)
#get predictions on test set using model
ridge_pred <- predict(ridge_mod, s = bestlam_ridge, newx = x_test)
# compute MSE on test set
mean((ridge_pred - y_test)^2)
```

```
## [1] 1238299
```

```
# view coefficients
predict(ridge_mod, type = "coefficients", s = bestlam_ridge)[1:18,]
```

```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc     Top25perc
## -1.76911e+03 -5.42798e+02  1.05130e+00  3.09005e-01  1.57179e+01  3.80014e+00
##   F.Undergrad   P.Undergrad      Outstate    Room.Board         Books      Personal
##   6.84966e-02  3.77103e-02 -4.81964e-02  1.76365e-01  8.28837e-02  4.96908e-02
##           PhD      Terminal     S.F.Ratio   perc.alumni        Expend     Grad.Rate
## -6.81271e+00 -6.98860e+00  2.94891e+01 -4.73107e+00  1.37072e-01  1.42885e+01
```

The value of λ that results in the smallest cross-validation error is 385.428. The test MSE for this model is 139216, which is much smaller than those of the kitchen-sink model and the best subsets model.

> part e: Fit a lasso model on the training set, with lambda chosen by cross-validation. Report the lambda chosen and the test error obtained.

SOLUTION:

```
lasso_mod <- glmnet(x_train, y_train, alpha = 1, lambda = grid) #make sure alpha = 1 for Lasso
set.seed(495)
```

```r
cvlasso.out <- cv.glmnet(x_train, y_train, alpha = 1)
# plot(cvlasso.out)
bestlam_lasso <- cvlasso.out$lambda.min
bestlam_lasso
```

```
## [1] 1.87418
```

```r
lasso.pred <- predict(lasso_mod, s = bestlam_lasso, newx = x_test)
mean((lasso.pred - y_test)^2)
```

```
## [1] 1605308
```

```r
# plot(lasso_mod)
predict(lasso_mod, type = "coefficients", s = bestlam_lasso)[1:18,]
```

```
##  (Intercept)   PrivateYes       Accept       Enroll    Top10perc    Top25perc
## -1.02270e+03 -4.07693e+02  1.66482e+00 -1.33398e+00  2.70312e+01 -4.74753e+00
##  F.Undergrad  P.Undergrad     Outstate   Room.Board        Books     Personal
##  9.28605e-02  6.72591e-02 -1.43518e-01  8.57916e-02  4.76283e-03  7.35831e-02
##          PhD     Terminal    S.F.Ratio  perc.alumni       Expend    Grad.Rate
## -1.16922e+01 -4.92282e+00  3.53653e+01  3.35754e+00  1.90823e-01  1.50663e+01
```

The value of λ that results in the smallest cross-validation error is 1.87418, and the test MSE for this model is 1605308, which is larger than the test MSE for the ridge regression model.

> part f: Comment on the results obtained across the four models. Address the following questions as part of your response. Do the test errors differ much? Do the coefficients differ greatly? In particular, if any variables were left out of the model in (c) or (e), is there any insight that they might have been removed based on the models in (b) or (d)? Which final model would you select here? Why?

SOLUTION:

The kitchen-sink linear regression model and the model picked by the best subsets method had similarly huge test MSEs, within the same order of magnitude. The models obtained through ridge and lasso were also similar in test MSE, but the MSE values were much smaller than those of the kitchen-sink and best subsets linear regression models.

From the kitchen-sink model in (b) to the best subsets model in (c), the variables PrivateYes, Top25perc, P.Undergrad, Room.Board, Books, Personal, Terminal, S.F.Ratio, and perc.alumni were dropped, and all of these variables demonstrated either insignificant or nearly insignificant fitted coefficients in the summary output of the kitchen-sink model. With the best subsets method applied, some of the leftover variables (Accept, Enroll, F.Undergrad, Outstate, Expend, Grad.Rate) had coefficient values of smaller magnitude, while others (Top10perc, PhD) had larger magnitude in the model for (c).

Between the ridge and lasso models, no coefficients are eliminated. Both models don't set any coefficients to 0, but observing the variables which were determined to have significant predictability in the best subsets model, all of these variables had larger magnitudes in the lasso model than in the ridge. The only notable coefficient was the Enroll variable, which has a positive coefficient for ridge but a negative one for lasso.

Additional 2 - Soil predictions

The data comes from a Kaggle competition: https://www.kaggle.com/c/afsis-soil-properties. The original data set contained 3600 variables, 3599 possible predictors (really, 3578 and some other variables) and a response, Sand. The 3599 predictors were reduced to 106 (methods to be taught later this semester) that can "best" distinguish between the two levels of Depth (another variable in the data set). The resulting data set of 107 variables (106 predictors and the response variable, Sand) was saved in the data set "newsoil". The row numbers should be removed as demonstrated below.

```
newsoil <- read.csv("https://awagaman.people.amherst.edu/stat495/newsoil.csv",
                    header = T)
newsoil <- select(newsoil, -X)
```

Our focus is on predicting the response variable Sand, using the selected variables from previous work.

> part a: Split the data set into a training and test set (75/25), with a seed of your choice. You may also wish to create appropriate x and y matrices for future function inputs at the same time.

SOLUTION:

```
set.seed(495)
n <- nrow(newsoil)
train_index <- sample(1:n, 0.75 * n)
test_index <- setdiff(1:n, train_index)

train <- newsoil[train_index, ]
test <- newsoil[test_index, ]

#further set up for using glmnet
x_train <- model.matrix(Sand ~ ., train)[, -1]
x_test <- model.matrix(Sand ~ ., test)[, -1]

y_train <- train %>%
  select(Sand) %>%
  unlist() %>%
  as.numeric()

y_test <- test %>%
  select(Sand) %>%
  unlist() %>%
  as.numeric()
```

> part b: Fit lasso models to predict Sand using all the possible predictors. Choose two lasso models - one that has a "best" lambda determined in some appropriate way, and another model with a different non-zero lambda of your choice. How many slope coefficients are set to 0 in each of your chosen lasso models?

SOLUTION:

```
lasso_mod <- glmnet(x_train, y_train, alpha = 1, lambda = grid) #make sure alpha = 1 for Lasso
set.seed(495)
cvlasso.out <- cv.glmnet(x_train, y_train, alpha = 1)
# plot(cvlasso.out)
bestlam_lasso <- cvlasso.out$lambda.min
bestlam_lasso
```

```
## [1] 5.99889e-05
```

```r
predict(lasso_mod, type = "coefficients", s = 6)[1:18,]
```

```
##  (Intercept)     m3748.98     m3747.05     m3745.13      m3743.2     m3741.27 
## -0.000524917  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000 
##     m3739.34     m3737.41     m3735.48     m3733.55     m3731.63      m3729.7 
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000 
##     m3727.77     m3725.84     m3723.91     m3721.98     m3720.05     m3718.13 
##  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000  0.000000000
```

```r
predict(lasso_mod, type = "coefficients", s = bestlam_lasso)[1:18,]
```

```
## (Intercept)    m3748.98    m3747.05    m3745.13     m3743.2    m3741.27 
##   -0.857255   -2.246324    0.000000    0.000000    0.000000    0.000000 
##    m3739.34    m3737.41    m3735.48    m3733.55    m3731.63     m3729.7 
##    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000 
##    m3727.77    m3725.84    m3723.91    m3721.98    m3720.05    m3718.13 
##    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
```

Using cross-validation to choose lambda, we find that 5.99889e-05 is the "best" lambda. In this model, all but one of the slope coefficients is set to 0, and in the model with my arbitrarily chosen lambda value of 6, all slope coefficients are 0.

> part c: Fit a ridge model to predict Sand using all the possible predictors. How many slope coefficients are set to 0 in your ridge model?

SOLUTION:

```r
set.seed(495)
x <- model.matrix(Sand ~ ., newsoil)[, -1] # trim off the first column
y <- newsoil %>% select(Sand) %>%  unlist() %>%  as.numeric()

grid <- 10^seq(10, -2, length = 100)
ridge_mod <- glmnet(x, y, alpha = 0, lambda = grid)

dim(coef(ridge_mod))
```

```
## [1] 107 100
```

```r
# plot(ridge_mod, label = TRUE)

cvridge.out <- cv.glmnet(x_train, y_train, alpha = 0) # Fit ridge regression model on training data
bestlam_ridge <- cvridge.out$lambda.min  # Select lambda that minimizes training MSE
bestlam_ridge
```

```
## [1] 0.0599889
```

```r
# fit model on training set
ridge_mod <- glmnet(x_train, y_train, alpha = 0, lambda = grid)
# view coefficients
predict(ridge_mod, type = "coefficients", s = bestlam_ridge)[1:18,]
```

```
## (Intercept)    m3748.98    m3747.05    m3745.13     m3743.2    m3741.27 
## -0.68437069 -0.49804163 -0.49940757 -0.48084484 -0.44127940 -0.39706509 
##    m3739.34    m3737.41    m3735.48    m3733.55    m3731.63     m3729.7 
## -0.35027421 -0.30313974 -0.26198377 -0.21087509 -0.14517597 -0.09279505 
##    m3727.77    m3725.84    m3723.91    m3721.98    m3720.05    m3718.13 
## -0.06001678 -0.00160409  0.07951157  0.16863490  0.25544545  0.31035715
```

Fitting a ridge model to predict Sand using all the possible predictors, we find a best lambda value of 0.0599889, and none of the slope coefficients are set to 0 in this ridge model.

> part d: Compute test MSEs for both of your lasso models and your ridge model.

SOLUTION:

```r
# arbitrary lasso MSE
lasso.pred1 <- predict(lasso_mod, s = 6, newx = x_test)
mean((lasso.pred1 - y_test)^2)
```

```
## [1] 0.966146
```

```r
# selected lasso MSE
lasso.pred2 <- predict(lasso_mod, s = bestlam_lasso, newx = x_test)
mean((lasso.pred2 - y_test)^2)
```

```
## [1] 0.28289
```

```r
# ridge MSE
ridge_pred <- predict(ridge_mod, s = bestlam_ridge, newx = x_test)
mean((ridge_pred - y_test)^2)
```

```
## [1] 0.278501
```

The test MSEs for the lasso model and the ridge model are 0.28289 and 0.278501, respectively.

> part e: Write a few sentences to address the following questions. Does the test MSE from the model with the "best" lambda suggest it is in fact a better predictive model than your other lasso model? Is ridge better than the lasso models? Which final model would you choose here from these three models? Why?

SOLUTION:

The test MSE from the model with the "best" lambda does suggest it is in fact a better predictive model than the other lasso model. I'm not sure how effectively large the magnitudes of the test MSEs for each lasso model are in the context of the data, but comparing the test MSE for the lasso model with arbitrary lambda (0.966146) and the test MSE for the lasso model with the cross-validated lambda (0.28289), we see that the latter is multiple times smaller than the former. However, the test MSE for the lasso model with the cross-validated lambda is still just slightly larger than that of the ridge model. Even so, the final model I would you choose here from these three is the lasso model with the cross-validated lambda, because the ridge model assigns non-zero slope coefficients to so many variables and thus retains a massive number of predictors, whereas the the lasso model yields many zero-value coefficients. For an increase of less than 0.005 of test MSE, I think it is worth it to use a much simpler model with fewer predictors allowed.

> part f: What is the default setting for the normalize option in lars and the standardize option in glmnet? Why is this setting important to the model fit?

SOLUTION:

The default setting for both the normalize option in lars and the standardize option in glmnet is TRUE. This is important to the model fit, because both normalization and standardization ensure that the regularization penalties are applied fairly and effectively to all predictors, regardless of their scales or magnitudes. Thus, they help maintain the interpretability and stability of the resulting models.

> part g: Explain what option you would change in order to fit an elastic net penalty (not OLS or ridge) using glmnet.

SOLUTION:

In order to fit an elastic net penalty (not OLS or ridge) using glmnet, I would adjust the alpha parameter to a value that represents the mixing parameter between lasso and ridge regression penalties. Since the function applies the lasso penalty when alpha = 1 and the ridge penalty when alpha = 0, I would choose an alpha value in between, such as 0.5, to balance the two penalties and achieve an elastic net one.

Additional 3

After reading through your portfolio reflections, I decided to try to adapt some class activities and assignments to better align with your goals. This is a little bit challenging because they are quite varied. However, the *College* data set that we used for Additional 1 does permit a host of different analyses, so we're going to try using it to help with this.

For this problem, your assignment is to tackle some aspect of your goals for class using the *College* data set. Include your work here, in the outline below. I'll give you feedback and work to correct any statistical issues, and note that while assessing this, I'm not looking for any one specific thing from any of you. I'm including some examples of what you might do below, based on some of the goals I read.

Examples, if you said . . .

- I want to demonstrate my understanding of method X. Can you apply method X to this data set? (You may have to do some work like create a variable for example to run an ANOVA; take one of the quantitative variables and cut it into 4 groups.) Try it. Write a summary of your findings.
- I want to work on my EDA. We know the (overall) goal here is to predict Apps. What EDA would you do (we skipped it above!)? Describe it, do some of it, etc.
- I want to practice commenting my code and making visuals extremely clear. Pick a simple analysis (predict Apps with like 2-3 other predictors), make your code awesome and visuals really good.
- I want to practice writing more precisely / shorter paragraphs for my results. Pick a simple analysis (predict Apps with like 2-3 other predictors), and write your results the way you typically would. Then, leave that there, and try a revision, working to improve the writing (this way, you see the original and the revision).

Other guidance:

- Spend at most 2 hours on this question. This is designed to let you practice something you wanted to work on, not take up all your time.
- If none of your goals seem to align with this, you can pick one based on the examples I listed above, or something similar that you want to work on.
- Use your best judgement for any models you need to fit here. If you just need a model to practice writing, it is okay if that's not the overall best model you might pick. On the other hand, if you want to practice model fitting, you should be spending time discussing that and showing your work for it. We will practice the entire data analysis process in future work.

part a: What aspect of your goals for class are you going to tackle for this assignment? How?

SOLUTION:

I'm going to work on my EDA and make sure to annotate my code clearly throughout.

part b: Include your work here!

SOLUTION: