Homework Assignment #5

Cadee Pikerton, Nicholas Noel, Lisette Villa

Due April 28, 2023

Instructions

You should submit either two or three files:

1. You should write your solutions to the Applied Problems and Conceptual Problem 3 in this R Markdown file and submit the (.Rmd) file.
2. You should knit the final solution file to pdf and submit the pdf. If you are having trouble getting code chunks to run, add `eval = FALSE` to the chunks that do not run. If you are having trouble getting R Studio to play nice with your LaTeX distribution, I will begrudgingly accept an HTML file instead.
3. Solutions to the Key Terms and the other Conceptual Problems can be submitted in a separate Word or pdf file or included in the same files as your solutions to Conceptual Problem 3 and the Applied Problems.

This homework assignment is worth a total of **40 points**.

Key Terms (5 pts)

Read Sections 6.1, 6.2, and 6.4 of Introduction to Statistical Learning, Second Edition. Based on your reading, answer the following questions.

1. Briefly explain what is meant by the term *feature selection* or *variable selection*.

Answer: Feature or variable selection is the process of picking the best predictors for a model. It is simply the process of choosing only the most important variables based on some objective measure of fit.

2. The book claims that we can use *deviance* instead of RSS when selecting among logistic regression models. Write the formula for deviance in terms of the maximized log-likelihood. Do smaller or larger values of deviance indicate a better fit?

Answer: Smaller deviance indicates a better fit and the formula for it is :

$-2L(\beta, \sigma^2)$

3. Explain why it is a bad idea to select the model with the lowest RSS on the training set.

Answer: It is a bad idea to select the model with the lowest RSS on the training set because we could be overfitting the data and choosing lowest RSS does not take into account bias-variance trade-off.

4. Does the AIC or BIC statistic tend to place a heavier penalty on bigger models? Why?

Answer: Since $\log n > 2$ for any $n > 7$, the BIC statistic generally places a heavier penalty on models with many variables than the AIC statistic.

5. In *ridge regression* and *LASSO*, what is the model when $\lambda = 0$? What about when $\lambda \to \infty$?

Answer: In ridge regression and lasso, when $\lambda = 0$ the model is the least squares estimates. As $\lambda \to \infty$, the ridge regression and lasso coefficient estimates will approach zero leaving us with the penalty term.

6. Suppose that we have *centered* the predictors (i.e., set all predictors to have mean 0). What is the intercept estimate $\hat{\beta}_0$ in this case?

Answer: By centering the data such that each predictor variable has mean 0, we ensure that $\hat{\beta}_0 = \bar{y}$ will be the same for all λ and not shrunk.

7. Why should we also *scale* the predictors (i.e., set all predictors to have standard deviation 1) when doing penalized regression?

Answer: By scaling the data so that each variable has standard deviation 1, we ensure that $\hat{\beta}_{j,\lambda}$ depends on the relative importance of each predictors.

8. When would we expect ridge regression to outperform the lasso? When would we expect the lasso to outperform ridge regression? In the real world, do we know which situation we are in?

Answer: We would expect lasso to outperform ridge regression when we want to perform both shrinkage and variable selection. We would expect ridge regression to outperform lasso when the predictors there are many correlated predictors and all of them are important in predicting the response. No, because we don't know when predictors are important.

9. Explain why each of the following approaches to adjusting the training set RSS are *inappropriate* in high-dimensional ($p > n$ or $p \approx n$) settings: (a) traditional R^2 and adjusted R^2; (b) AIC, BIC, and Cp.

Answer:

(a) Problems arise in the application of adjusted R^2 and R^2 in the high-dimensional setting, since one can easily obtain a model with an adjusted R^2 and an R^2 value of 1.

(b) The C_p, AIC, and BIC approaches are not appropriate in the high-dimensional setting, because estimating $\hat{\sigma}^2$ is problematic.

10. Explain the *curse of dimensionality* in two ways: first, in terms of overfitting, and second, in terms of multicollinearity issues.

Answer: With a large number of predictors p, the model becomes more complex and flexible which can result in overfitting the data. This means we are capturing a lot of noise rather than capturing the overall pattern within the data. In the high-dimensional setting, the multicollinearity problem is extreme: any variable in the model can be written as a linear combination of all of the other variables in the model. Essentially, this means that we can never know exactly which variables (if any) truly are predictive of the outcome, and we can never identify the best coefficients for use in the model.

Conceptual Problems

Conceptual Problem 1 (3 pts)

Textbook Exercise 6.6.2.

(iii) Less flexible and will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

As lambda increases, the estimated coefficients decrease. This leads to a decrease in the overall variance in exchange for some bias (The decrease in variance is greater than our increase in bias).

(b)

Lasso and Ridge both work similar in terms of theory so for the reasons that iii was chosen in part a is the same here in part b. The primary difference between the two methods is their shape with lasso using the absolute value and ridge using squares.

(c) Repeat (a) for non-linear methods relative to least squares.

In this example, compared to ridge and lasso, non-linear methods are impacted differently. Here, ii, is correct as non linear methods are more flexible having less bias at the cost of higher variance.

Conceptual Problem 2 (8 pts total)

In the textbook, it is claimed that if we define our prior distribution of the β_j slopes to be $p(\beta) = \prod_{j=1}^{p} g(\beta_j)$, with $g(\beta_j)$ a Gaussian distribution with mean 0 and standard deviation a function of λ, then the ridge regression solution yields the posterior mode.

Let's investigate this in the simplest case. Suppose that we standardize both a single predictor X and the response Y such that the population model passes through (0, 0), i.e., $\beta_0 = 0$ and so $Y = \beta_1 X + \epsilon$ with $\epsilon \sim N(0, \sigma)$.

Part a (2 pts)

Find the ridge regression solution, i.e., the value of β_1 that minimizes

$$\sum_{i=1}^{n} (y_i - \beta_1 x_i)^2 + \lambda \beta_1^2$$

, in terms of the y_i's, x_i's, and λ.

Part b (1 pt)

Write out the likelihood function $L(\beta_1 | x, y)$, where x is the vector of x_i's and y is the vector of y_i's.

Part c (1 pt)

Suppose we define the prior distribution of β_1 as $\beta_1 \sim N(0, f(\lambda))$ as suggested in the textbook, i.e.,

$$g(\beta_1) = \frac{1}{f(\lambda)\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\beta_1}{f(\lambda)})^2}$$

Under this assumption, find the posterior distribution of β_1 given x and y. You may rewrite the integral in the denominator as "C", an arbitrary constant.

Part d (2 pts)

Find the posterior mode, i.e., the value of β_1 that maximizes the posterior pdf of β_1 from part (c), in terms of the x_i's, y_i's, σ, and $f(\lambda)$. (HINT: It is much easier to maximize the logarithm of the posterior pdf instead.)

Part e (2 pts)

Find a function $f(\lambda)$ for which the ridge regression solution equals the posterior mode. The function should be independent of the data (x_i's and y_i's) but may depend on σ. You may assume that $\lambda > 0$.

Simulation Problems

```
library(tidyverse)
```

Simulation Problem 1 (9 pts)

Part a (Code: 0.5 pts)

Textbook Exercise 6.6.8 part (a). When generating X, use `mean = 0` and `sd = 1` so that we don't need to worry about normalizing anything.

```
n <- 100
X <- rnorm(n, 0,1)

eps <- rnorm(n, 0, 0.25)
```

Part b (Code: 0.5 pts)

Textbook Exercise 6.6.8 part (b).

```
b0 <- 1
b1 <- 2
b2 <- 3
b3 <- 4

Y = b0 + b1*X + b2*X^2 + b3*X^3 + eps

df = data.frame(X,X^2,X^3,X^4,X^5,X^6,X^7,X^8,X^9,X^10,Y)
```

Part c (Code: 2 pts)

Textbook Exercise 6.6.8 part (c).

```
library(leaps)

regsubsets(Y~. , data = df)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = df)
## 10 Variables  (and intercept)
##      Forced in Forced out
## X        FALSE      FALSE
## X.2      FALSE      FALSE
## X.3      FALSE      FALSE
## X.4      FALSE      FALSE
## X.5      FALSE      FALSE
## X.6      FALSE      FALSE
## X.7      FALSE      FALSE
## X.8      FALSE      FALSE
## X.9      FALSE      FALSE
## X.10     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
```

```
regfit = regsubsets(Y~.,data=df,nvmax=10)
reg.summary = summary(regfit)
reg.summary$cp
```

```
##  [1] 3.646905e+04 2.866242e+03 1.475500e+00 9.240518e-01 2.034509e+00
##  [6] 3.954444e+00 5.719080e+00 7.483417e+00 9.014045e+00 1.100000e+01
```

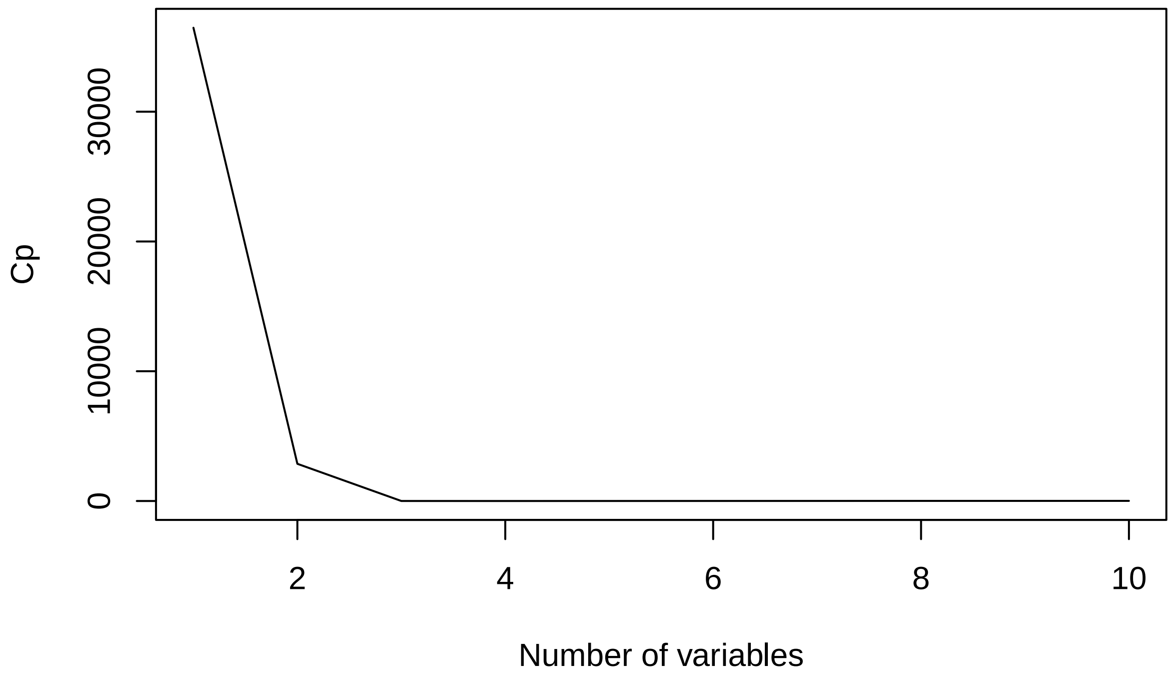
```
reg.summary$bic
```

```
##  [1] -274.1423 -520.9193 -861.8468 -860.0091 -856.3871 -851.8709 -847.5277
##  [8] -843.1856 -839.1063 -834.5169
```

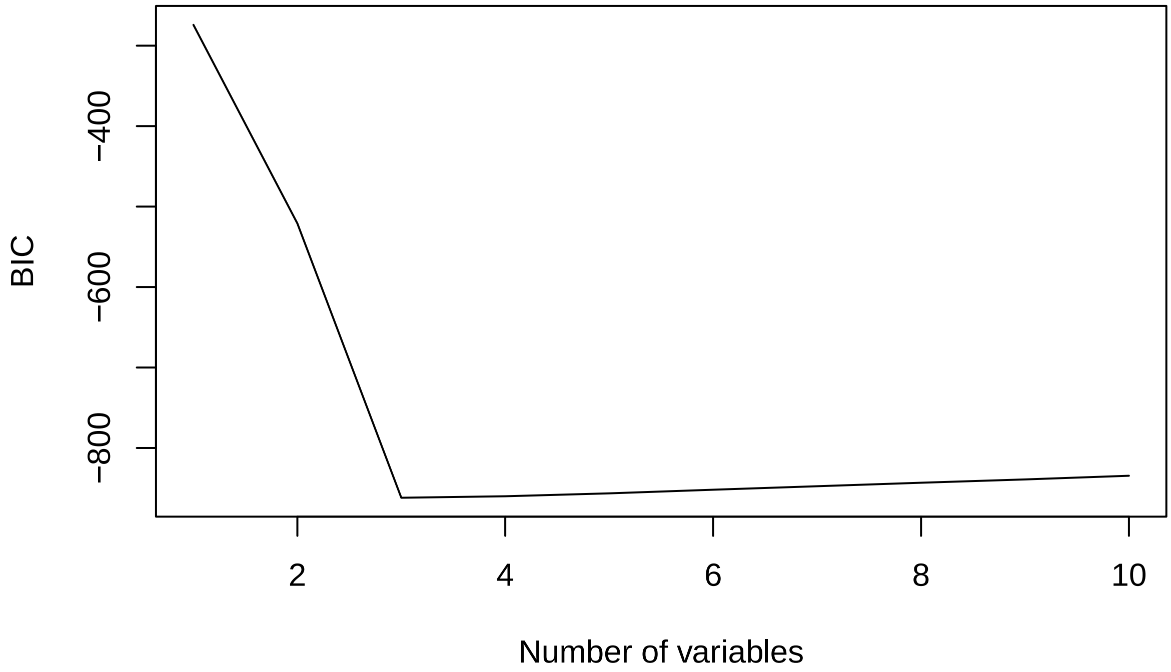
```
reg.summary$adjr2
```

```
##  [1] 0.9405948 0.9951411 0.9998450 0.9998476 0.9998475 0.9998460 0.9998447
##  [8] 0.9998434 0.9998425 0.9998408
```

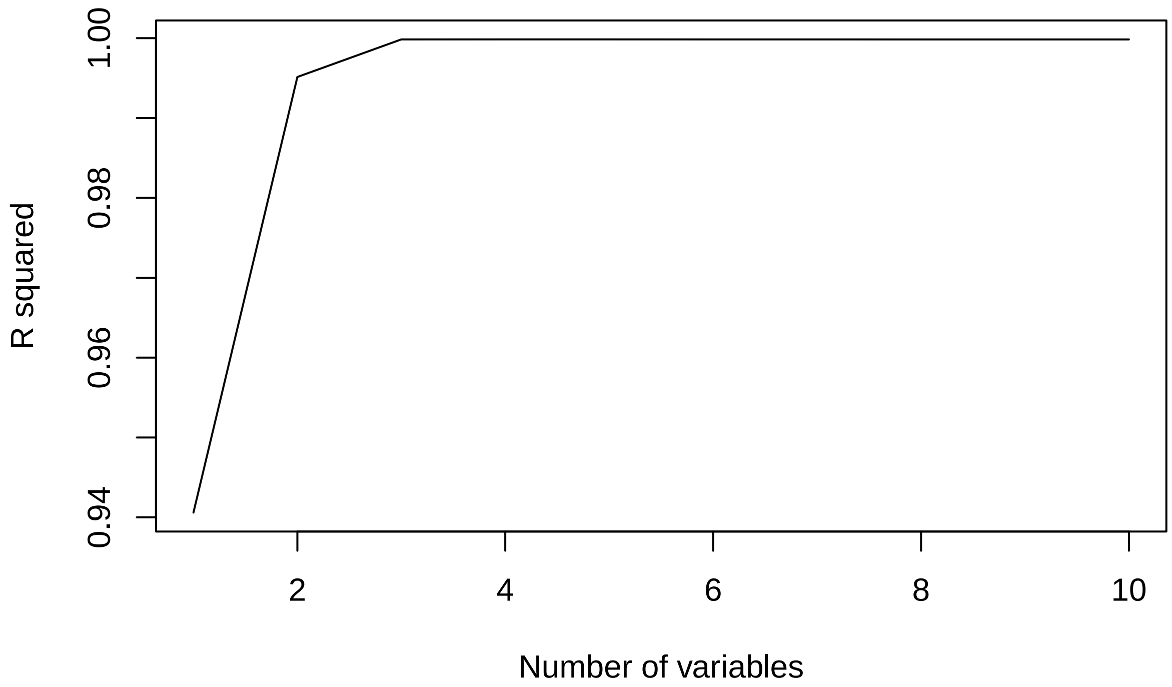
```
plot(reg.summary$cp,xlab="Number of variables", ylab="Cp",type="l")
```



```
plot(reg.summary$bic,xlab="Number of variables", ylab="BIC",type="l")
```



```
plot(reg.summary$adjr2,xlab="Number of variables", ylab="R squared",type="l")
```



Part d (Explanation: 2 pts)

```
coef(regfit, id = 3)
```

```
## (Intercept)           X         X.2         X.3
##    1.015257    1.990498    2.986506    3.991032
```

Choose one model from part (c) as your final "best model" and justify your answer. Compare the coefficient estimates for that model to the known coefficients in the population model.

The model that we are choosing is our model with four variables. Based on the three plots it is clear that we are optimized at that point. The coefficients of the model has an intercept of 0.9550098 and coefficients, 1.9701017, 3.0313854, and 4.0048057. All of which are not too far off from their respective 0,1,2 and 3. In total it is relatively clear that three variables is ideal.

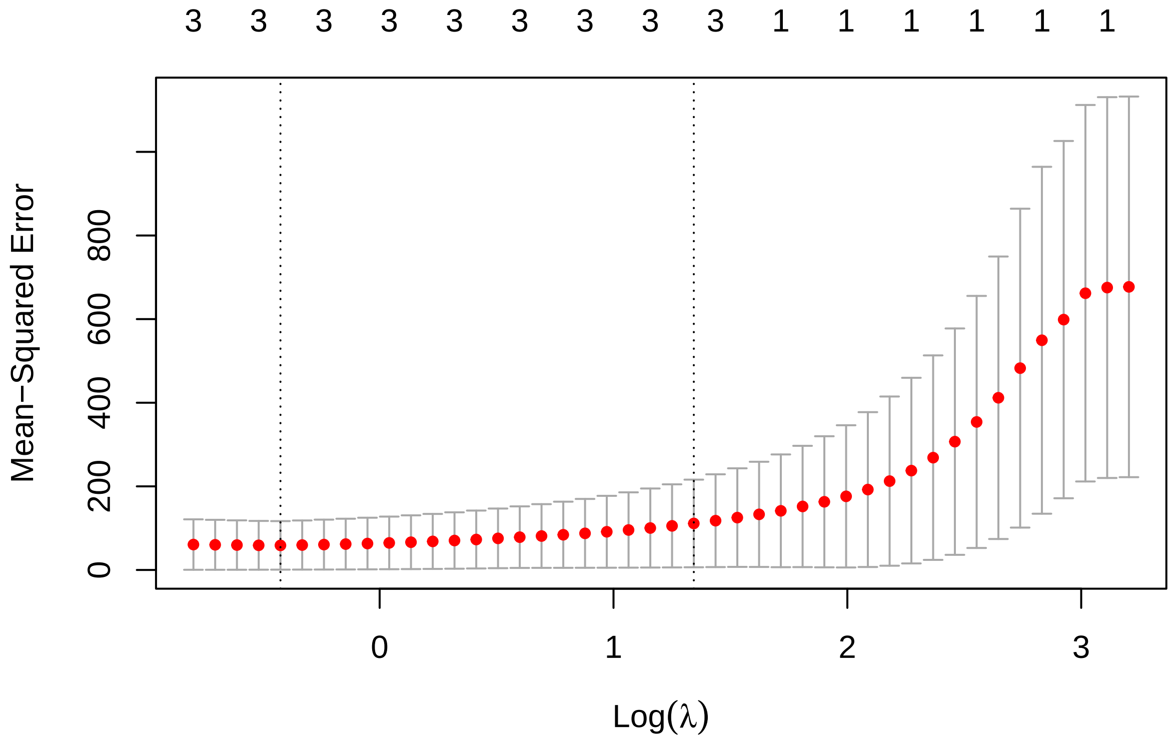
Part e (Code: 2 pts)

Fit a LASSO model on the simulated data (again using $X, X^2, \ldots, X^{10}$ as predictors). Perform cross-validation using either `cv.glmnet` or `tune_grid` to determine the optimal value of λ, then fit the final model using that value of λ on the entire dataset.

```
library(glmnet)

x = df %>%
  select(-Y)

y = df$Y
```

```
# Training and test sets.
train = sample(1:nrow(x), nrow(x)/2)

test = (-train)

y.test = y[test]

lasso_model <-  glmnet(x[train,], y[train], alpha=1)
```

```
x <- as.matrix(x)
cv = cv.glmnet(x[train,],y[train], alpha=1)
plot(cv)
```



```
cv$lambda.min
```

```
## [1] 0.6542988
```

```
final = glmnet(x,y,alpha=1)
```

Our optimal value for lamda is 0.306283.

Part f (Explanation: 2 pts)

How does shrinkage affect the coefficient estimates? Compare the coefficient estimates for your final LASSO model to *both* the coefficient estimates from the "best" model using subset selection *and* the known coefficients in the population model.

```
lasso.coef = predict(final, type="coefficients",s=0.306283)[1:11,]
```

The lasso model has four variables. The coefficients for X, X^2 and X^3 closely match the ones chosen earlier implying the goodness of the model, however out intercept of the population is 0 but here it is 1.258984, an increase I am not entirely sure of.

Applied Problems

Applied Problem 1 (15 pts total)

This exercise is *strongly* modeled on ISLR Exercise 6.6.9, in that we want to fit several models that predict the number of applications a college receives, using the `College` dataset.

```
library(ISLR2) # for college dataset
library(leaps)
```

Part a (Explanation: 1 pt)

Look up the documentation for the `College` dataset (i.e., `?College`). There is at least one variable, and perhaps as many as four variables, that *should not* be used when fitting a model to predict the number of applications received. Which variable(s) are you going to not even consider including in the model? Why?

We are not going to consider `Accept`, number of application accepted, and `Enroll`, number of new students enrolled, since those both depend on the number of applications received. We should also probably not include `F.Undergrad` and `P.Undergrad` since they also depend on the number of applications received.

Part b (Code: 1 pt)

Subset the College dataset to remove the offending variable(s). Then, randomly split the new dataset into a training set (containing approximately 75-80% of the data, your choice exactly how many rows) and a validation set (containing the remaining 20-25%).

```
subColl <- College %>%
  select(-Accept, -Enroll, -F.Undergrad, -P.Undergrad)

set.seed(777)
n <- nrow(subColl)
test.rows <- sample(n, size = floor(0.20*n), replace = FALSE)
coll_train <- subColl[-test.rows,]
coll_test <- subColl[test.rows,]
```

Part c (Code: 2 pts; Explanation: 1 pt)

Use best subset selection to obtain an optimal least-squares linear regression model on the training set. If you use the `regsubsets` function, make sure that you set `nvmax` to the number of remaining predictors in your dataset. Justify your choice of model, then fit that final model on the training set.

```
regfit.full <- regsubsets(Apps ~ ., data = coll_train, nvmax = 13)
reg.summary <- summary(regfit.full)
```

```
which.max(reg.summary$adjr2)
```

```
## [1] 10
```

```
coef(regfit.full, 10)
```

```
##   (Intercept)    PrivateYes     Top10perc     Top25perc    Room.Board
## -5052.6416311 -4070.2604611    29.1339852    20.8626908     0.3991009
##      Personal      Terminal     S.F.Ratio   perc.alumni        Expend
##     0.4433217    15.2121707   143.5710430   -45.7643752     0.1900287
##     Grad.Rate
##    41.0593087
```

Based on adjusted Rˆ2 we will be using a model with the 10 predictors printed above.

```
rgsb.fit <- lm(Apps ~ Private + Top10perc + Room.Board + Personal + Terminal + S.F.Ratio + perc.alumni
```

Part d (Code: 2 pts; Explanation: 1 pt)

Use cross-validation on the training set to find an optimal value of λ for a ridge regression model. Justify your choice of λ, then fit a ridge regression model on the training set using that value of λ.

```
library(glmnet)
library(tidymodels)
```

```
ridge_model <- linear_reg(mode = "regression", engine = "glmnet",
                          penalty = tune(),
                          mixture = 0)

ridge_wflow <- workflow() %>%
  add_model(ridge_model)
```

```
ridge_recipe <- recipe(Apps ~ ., data = coll_train) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_dummy(all_nominal_predictors())

ridge_wflow <- ridge_wflow %>%
  add_recipe(ridge_recipe)
```

```
set.seed(437)
coll_kfold <- vfold_cv(coll_train, v = 5, repeats = 3)

ridge_tune1 <- tune_grid(ridge_model,
                      ridge_recipe,
                      resamples = coll_kfold,
                      grid = grid_regular(penalty(range = c(-3, 4)), levels = 50))
```

```
ridge_best <- select_by_one_std_err(
  ridge_tune1,
  metric = "rmse",
  desc(penalty)
)

best_lam_rr <- ridge_best$penalty
```

The best value of lambda is 3727.5937203 since it was associated with the lowest rmse.

```
ridge_wflow_final <- finalize_workflow(ridge_wflow, parameters = ridge_best)

ridge_fit <- fit(ridge_wflow_final, data = coll_train)
```

Part e (Code: 2 pts; Explanation: 1 pt)

Use cross-validation on the training set to find an optimal value of λ for a LASSO model. Justify your choice of λ, then fit a LASSO model on the training set using that value of λ.

```
lasso_model <- linear_reg(mode = "regression", engine = "glmnet",
                          penalty = tune(),
                          mixture = 1)

lasso_wflow <- workflow() %>%
  add_model(lasso_model) %>%
  add_recipe(ridge_recipe)
```

We do need to re-tune the model:

```
lasso_tune1 <- tune_grid(lasso_model,
                      ridge_recipe,
                      resamples = coll_kfold,
                      grid = grid_regular(penalty(range = c(-3, 4)), levels = 50))
```

```
lasso_best <- select_by_one_std_err(
  lasso_tune1,
  metric = "rmse",
  desc(penalty)
)

best_lam_lass <- lasso_best$penalty
```

The best value of lambda is 372.759372 since it was associated with the lowest cross-validated RMSE.

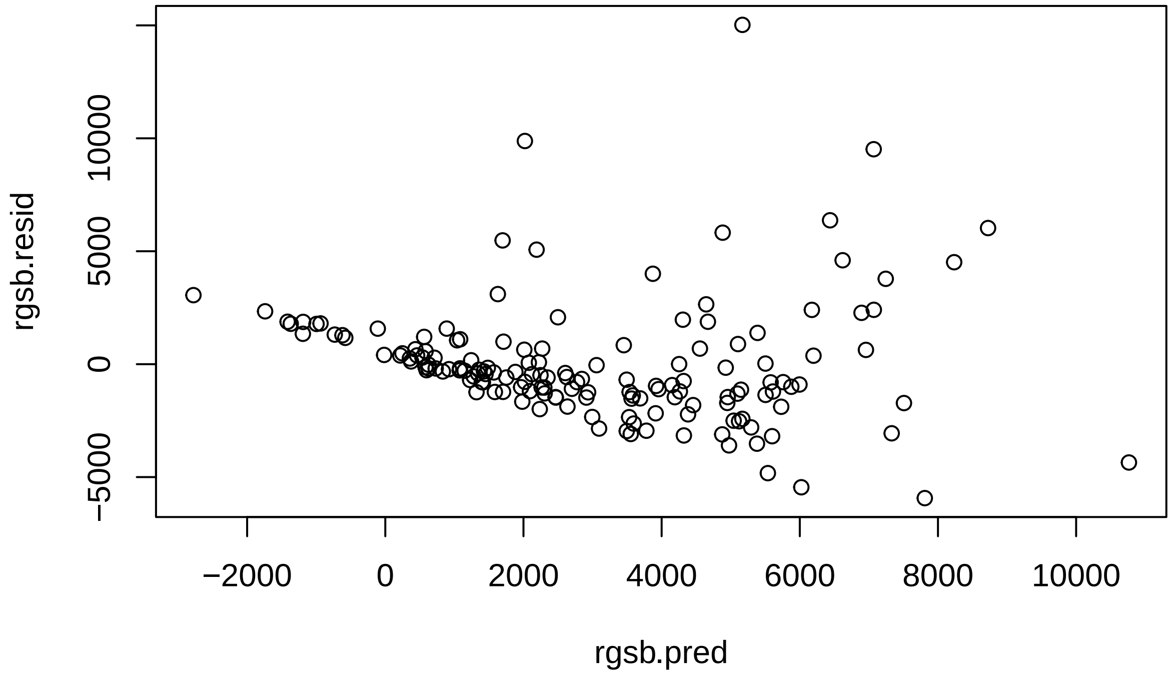
```
lasso_wflow_final <- finalize_workflow(lasso_wflow, parameters = lasso_best)

lasso_fit <- fit(lasso_wflow_final, data = coll_train)
```

Part f (Code: 2 pts)

Predict the number of applications received for colleges in the validation set, using each of the three models from parts (c)-(e).

For each model, create a plot of the model residuals (y-axis) against the predicted values (x-axis), and report the estimated test MSE or RMSE obtained for each model.

```
rgsb.pred <- predict(rgsb.fit, newdata = coll_test)
rgsb.resid <- coll_test$Apps - rgsb.pred
plot(rgsb.pred, rgsb.resid)
```



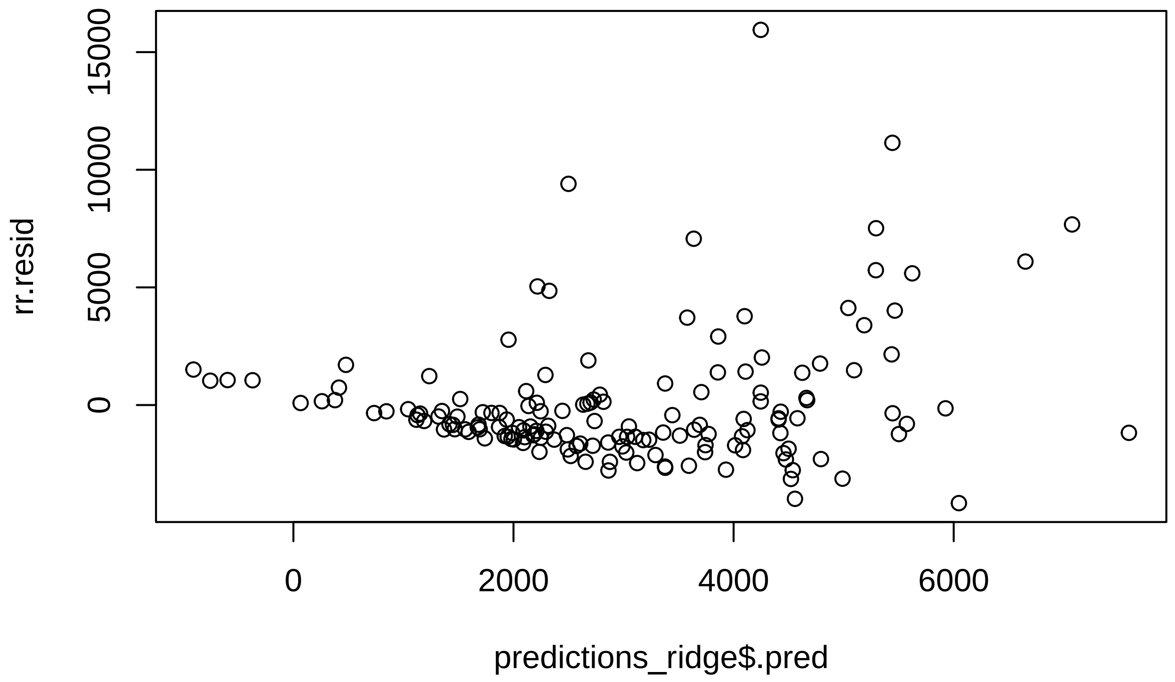
```
(RMSE.rgsb <- sqrt(sum((rgsb.resid)^2)/length(rgsb.resid)))
```

```
## [1] 2651.553
```

```
predictions_ridge <- broom::augment(ridge_fit, new_data = coll_test)

rmse(predictions_ridge, truth = Apps, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2736.
```

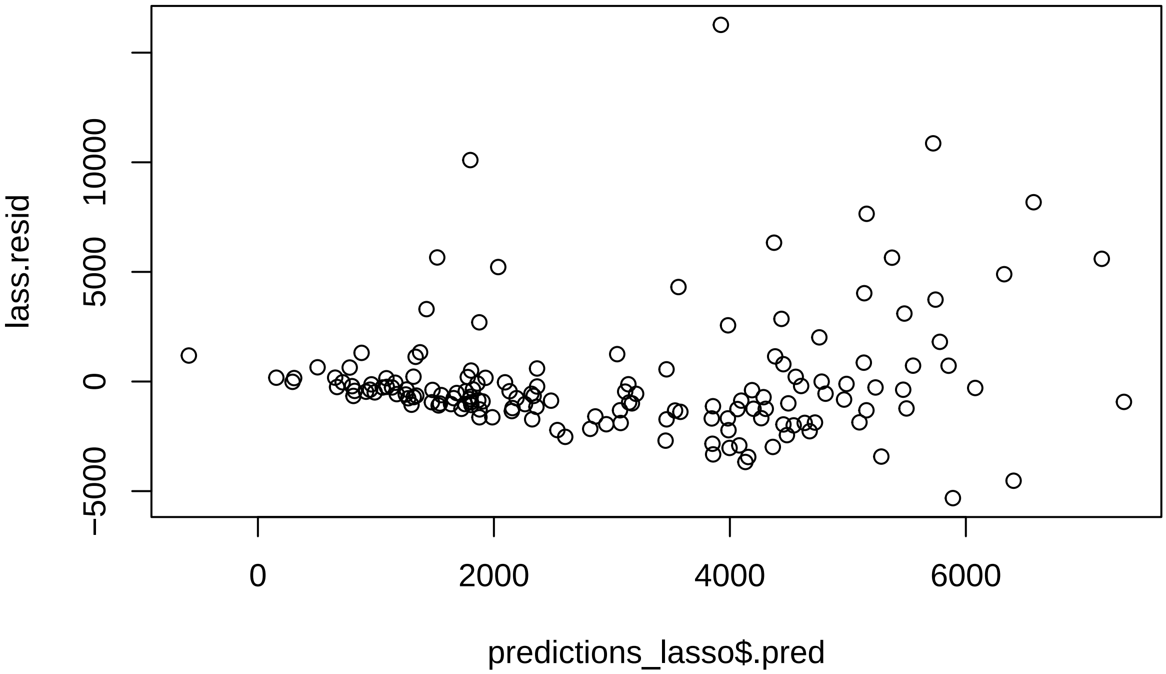
```
rr.resid <- coll_test$Apps - predictions_ridge$.pred

plot(predictions_ridge$.pred, rr.resid)
```



```
predictions_lasso <- broom::augment(lasso_fit, new_data = coll_test)

rmse(predictions_lasso, truth = Apps, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2756.
```

```
lass.resid <- coll_test$Apps - predictions_lasso$.pred

plot(predictions_lasso$.pred, lass.resid)
```



Part g (Explanation: 2 pts)

How accurately can we predict the number of college applications received? Looking at the three residual plots you created in part (f), do you notice any differences in the pattern of errors made by the three models? Any “obviously wrong” predictions? Explain your reasoning.

```
summary(coll_test$Apps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    81.0   817.5  1768.0  3010.9  3931.5 20192.0
```

```
sd(coll_test$Apps)
```

```
## [1] 3417.623
```

We are predicting the number of college applications with the lowest RMSE using the least-squares regression built after applying best subset selection. We are getting a better RMSE when predicting than indicated by our fitted models which indicates they are not overfit. When looking at the spread of our test response, that has a minimum of 81, a maximum of 20,192, and a standard deviation of 3418, an RMSE between 2600 - 2800 does not seem too extreme.

```
coll_test[which.max(rgsb.resid),]
```

```
##                   Private  Apps Top10perc Top25perc Outstate Room.Board Books
## Boston University     Yes 20192        45        80    18420       6810   475
##                   Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
## Boston University     1025  80       81      11.9          16  16836        72
```

```
rgsb.pred[which.max(rgsb.resid)]
```

```
## Boston University
##          5169.373
```

```
coll_test[which.max(rr.resid),]
```

```
##                   Private  Apps Top10perc Top25perc Outstate Room.Board Books
## Boston University     Yes 20192        45        80    18420       6810   475
##                   Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
## Boston University     1025  80       81      11.9          16  16836        72
```

```
coll_test[which.max(lass.resid),]
```

```
##                   Private  Apps Top10perc Top25perc Outstate Room.Board Books
## Boston University     Yes 20192        45        80    18420       6810   475
##                   Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
## Boston University     1025  80       81      11.9          16  16836        72
```

There doesn't seem to be any obvious difference between the residual plots, the best subset selection model seems to have more negative residuals, indicating that it was consistently predicting higher values of number of applications when compared to the ridge and LASSO regressions. There is also one obviously wrong prediction for all three models which is the Boston University, which was predicted to have much a much lower number of applications than it actually had.