Homework 5, Question 9

Matthew Bradley, Ben Howell, Hayley Zorkic, Ayanna Fisher

3/24/2022

Question 9

a:

```r
set.seed(2)
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.0.5
```

```r
data(College)

sample_size <- floor(0.75 * nrow(College))

train_index <- sample(seq_len(nrow(College)), size = sample_size)
College_train <- College[train_index,]
College_test <- College[-train_index,]
```

b:

```r
model <- lm(Apps~., data = College_train)
mean((College_test$Apps - predict.lm(model,College_test))^2)
```

```
## [1] 1287764
```

Our test error for the linear model is 1,287,764, which is very high.

c:

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-3
```

```r
lamdas <- cv.glmnet(x = data.matrix(College_train[-2]), y = College_train$Apps, alpha = 0)
bestlam <- lamdas$lambda.min
bestlam
```

```
## [1] 389.2482
```

```r
ridge <- glmnet(x = data.matrix(College_train[-2]), y = College_train$Apps, nlambda = round(bestlam), al
ridge.pred <- predict(ridge, newx = data.matrix(College_test[-2]))
mean((College_test$Apps - ridge.pred)^2)
```

```
## [1] 5519040
```

Our test error for the ridge model is 5,519,040, which is even higher than the linear model.

d:

```
lamdas <- cv.glmnet(data.matrix(College_train[-2]), y = College_train$Apps, alpha = 1)
bestlam <- lamdas$lambda.min
bestlam
```

```
## [1] 2.077301
```

```
lasso <- glmnet(x = data.matrix(College_train[-2]), y = College_train$Apps, alpha = 1, nlambda = round(
lasso.pred <- predict(lasso, newx = data.matrix(College_test[-2]))
mean((College_test$Apps - lasso.pred)^2)
```

```
## [1] 5573200
```

```
coef(lasso, s =2)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -427.27491109
## Private     -392.61358627
## Accept         1.62193287
## Enroll        -1.15561058
## Top10perc     46.13903704
## Top25perc    -14.45489244
## F.Undergrad    0.09709869
## P.Undergrad    0.05795143
## Outstate      -0.07734922
## Room.Board     0.18625620
## Books          0.19960768
## Personal       0.05830401
## PhD           -6.42015481
## Terminal      -4.29365990
## S.F.Ratio     23.10272519
## perc.alumni    3.54220870
## Expend         0.07148452
## Grad.Rate      5.66653637
```

The test error for the lasso model is 5,573,200, which is similar to our error in the ridge model. All 17 of the coefficients are nonzero.

e:

```
set.seed(1)
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
pcr.fit <- pcr(Apps ~ ., data = College_train , scale = TRUE , validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 582 17
##  Y dimension: 582 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            4091     4041     2156     2156     1809     1720     1712
## adjCV         4091     4042     2152     2154     1762     1705     1704
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1705     1682     1623      1628      1638      1638      1656
## adjCV     1700     1673     1617      1622      1631      1632      1649
##        14 comps  15 comps  16 comps  17 comps
## CV         1658      1560      1225      1158
## adjCV      1652      1521      1215      1148
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X      31.692    57.03    64.32    69.88    75.13    80.01    83.85    87.41
## Apps    3.817    73.74    73.92    82.89    84.42    84.44    84.67    85.30
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       90.65     92.97     95.03     96.79     97.89     98.71     99.34
## Apps    86.04     86.17     86.17     86.23     86.24     86.26     91.95
##       16 comps  17 comps
## X        99.83    100.00
## Apps     93.31     93.98
```

```
pcr.pred <- predict(pcr.fit , College_test, ncomp = 17)
mean ((pcr.pred - College_test$Apps)^2)
```

```
## [1] 1287764
```

The test error in the pcr model is 1,287,764, which is similar to the linear regression model. The M selected was 17 (all components were considered).

g:

All of the test errors were large (in the millions). The linear model and PCR had the lowest errors while the ridge and lasso models had the highest errors. We cannot predict the number of apps received very accurately because the errors in the models are high.