Homework #8: Variable Selection and Shrinkage

1. **(Real data analysis - Chap. 6, # 9, p.263)**
   Predict the number of applications received based on the other variables in the **College** data set. This data set is from our textbook. To access it, you can type `library(ISLR)`; `attach(College)`. Fit

   (a) Least squares regression, selecting the best model;

```
set.seed(666)
n <- nrow(College)
z <- sample(n,n/2)
train <- College[z,]
test <- College[-z,]
## (a) LSE
reg.fit <- regsubsets(Apps ~ ., data = College) # leaps
summary(reg.fit)

## Subset selection object
## Call: regsubsets.formula(Apps ~ ., data = College)
## 17 Variables  (and intercept)
##             Forced in Forced out
## PrivateYes      FALSE      FALSE
## Accept          FALSE      FALSE
## Enroll          FALSE      FALSE
## Top10perc       FALSE      FALSE
## Top25perc       FALSE      FALSE
## F.Undergrad     FALSE      FALSE
## P.Undergrad     FALSE      FALSE
## Outstate        FALSE      FALSE
## Room.Board      FALSE      FALSE
## Books           FALSE      FALSE
## Personal        FALSE      FALSE
## PhD             FALSE      FALSE
## Terminal        FALSE      FALSE
## S.F.Ratio       FALSE      FALSE
## perc.alumni     FALSE      FALSE
## Expend          FALSE      FALSE
## Grad.Rate       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 ) " "        "*"    " "    " "        " "        " "
## 2  ( 1 ) " "        "*"    " "    "*"        " "        " "
## 3  ( 1 ) " "        "*"    " "    "*"        " "        " "
## 4  ( 1 ) " "        "*"    " "    "*"        " "        " "
```

```
## 5  ( 1 ) " "         "*"     "*"     "*"        " "        " "
## 6  ( 1 ) " "         "*"     "*"     "*"        " "        " "
## 7  ( 1 ) " "         "*"     "*"     "*"        "*"        " "
## 8  ( 1 ) "*"         "*"     "*"     "*"        " "        " "
##          P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 2  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 3  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 4  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 5  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 6  ( 1 ) " "          "*"      "*"        " "   " "      " " " "
## 7  ( 1 ) " "          "*"      "*"        " "   " "      " " " "
## 8  ( 1 ) " "          "*"      "*"        " "   " "      "*" " "
##          S.F.Ratio perc.alumni Expend Grad.Rate
## 1  ( 1 ) " "       " "          " "    " "
## 2  ( 1 ) " "       " "          " "    " "
## 3  ( 1 ) " "       " "          "*"    " "
## 4  ( 1 ) " "       " "          "*"    " "
## 5  ( 1 ) " "       " "          "*"    " "
## 6  ( 1 ) " "       " "          "*"    " "
## 7  ( 1 ) " "       " "          "*"    " "
## 8  ( 1 ) " "       " "          "*"    " "
```

```
which.max(summary(reg.fit)$adjr2) # Adjusted R2
```

```
## [1] 8
```

```
which.min(summary(reg.fit)$cp)  # Mallows Cp
```

```
## [1] 8
```

```
which.min(summary(reg.fit)$bic)  # BIC = Bayesian information criterion
```

```
## [1] 8
```

```
    # 8 predictors
summary(reg.fit)$which[8,] # predictors
```

```
## (Intercept)  PrivateYes      Accept      Enroll   Top10perc   Top25perc
##        TRUE        TRUE        TRUE        TRUE        TRUE       FALSE
## F.Undergrad P.Undergrad    Outstate  Room.Board       Books    Personal
##       FALSE       FALSE        TRUE        TRUE       FALSE       FALSE
##         PhD    Terminal   S.F.Ratio perc.alumni      Expend   Grad.Rate
##        TRUE       FALSE       FALSE       FALSE        TRUE       FALSE
```

```
reg <- lm(Apps ~ Private + Accept + Enroll + Top10perc + Outstate + Room.Board + P
summary(reg)
```

```
##
## Call:
## lm(formula = Apps ~ Private + Accept + Enroll + Top10perc + Outstate +
##     Room.Board + PhD + Expend, data = College)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5014.0  -440.9   -16.1   323.0  7822.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -161.21352  233.71986  -0.690 0.490546
## PrivateYes  -536.94435  132.84967  -4.042 5.84e-05 ***
## Accept         1.58311    0.04016  39.421  < 2e-16 ***
## Enroll        -0.56700    0.11165  -5.079 4.78e-07 ***
## Top10perc     37.29291    3.19325  11.679  < 2e-16 ***
## Outstate      -0.08659    0.01797  -4.817 1.75e-06 ***
## Room.Board     0.17216    0.04694   3.668 0.000262 ***
## PhD          -11.22089    3.10707  -3.611 0.000324 ***
## Expend         0.07670    0.01117   6.868 1.34e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1052 on 768 degrees of freedom
## Multiple R-squared:  0.9269,Adjusted R-squared:  0.9261
## F-statistic:  1217 on 8 and 768 DF,  p-value: < 2.2e-16
```

```
    # Validation set approach
ols.reg <- glm(Apps ~ Private + Accept + Enroll + Top10perc + Outstate + Room.Boar
y.hat <- predict(ols.reg, test)
(err.lm <- mean((test$Apps - y.hat)^2))  # test error
```

```
## [1] 1559802
```

(b) Ridge regression, with λ chosen by cross-validation;

```
reg <- lm(Apps ~Private + Accept + Enroll + Top10perc + Outstate + Room.Board + Ph
X <- model.matrix(reg)
Y <- Apps[z]
cv.ridge <- cv.glmnet(X,Y,alpha=0, nfolds = 10, lambda = seq(0,40,0.01)) # library
(lambda <- cv.ridge$lambda.min) # cv least lambda
```

```
## [1] 15.53
```

```
X.test <- model.matrix(Apps ~Private + Accept + Enroll + Top10perc + Outstate + Ro
y.hat.ridge <- predict(cv.ridge, s=lambda, newx=X.test) # s: Value(s) of the penal
```

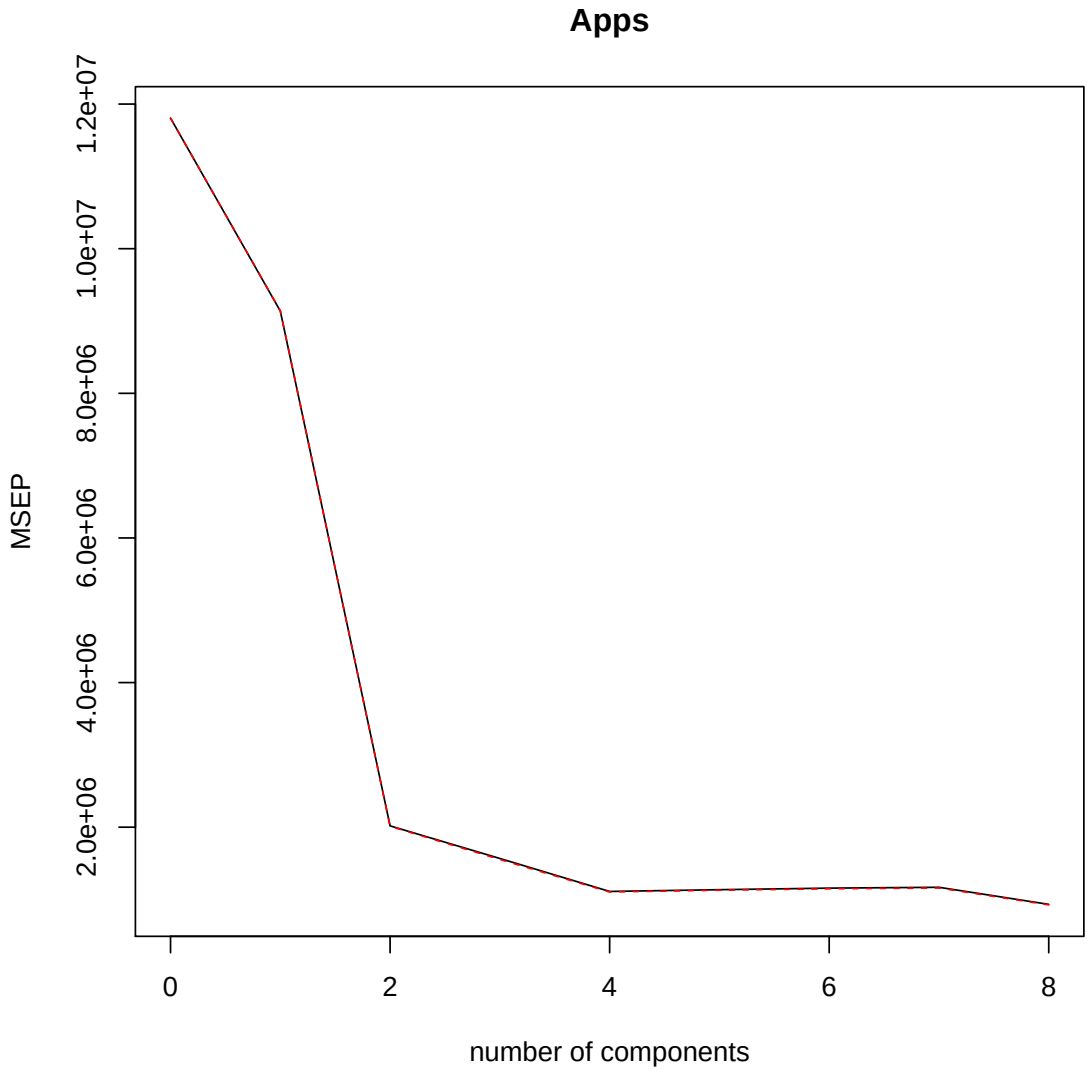
```
(err.ridge <- mean((test$Apps - y.hat.ridge)^2))  # test error

## [1] 1618175
```

(c) Lasso, with λ chosen by cross-validation;

```
cv.lasso <- cv.glmnet(X,Y,alpha=1,nfolds = 10, lambda = seq(0,40,0.01)) # library(
(lambda <-cv.lasso$lambda.min) # cv least lambda

## [1] 8.52

y.hat.lasso <- predict(cv.lasso, s=lambda, newx=X.test)
(err.lasso <- mean((test$Apps - y.hat.lasso)^2))  # test error

## [1] 1574253
```

(d) PCR model, with M, the number of principal components, chosen by cross-validation;

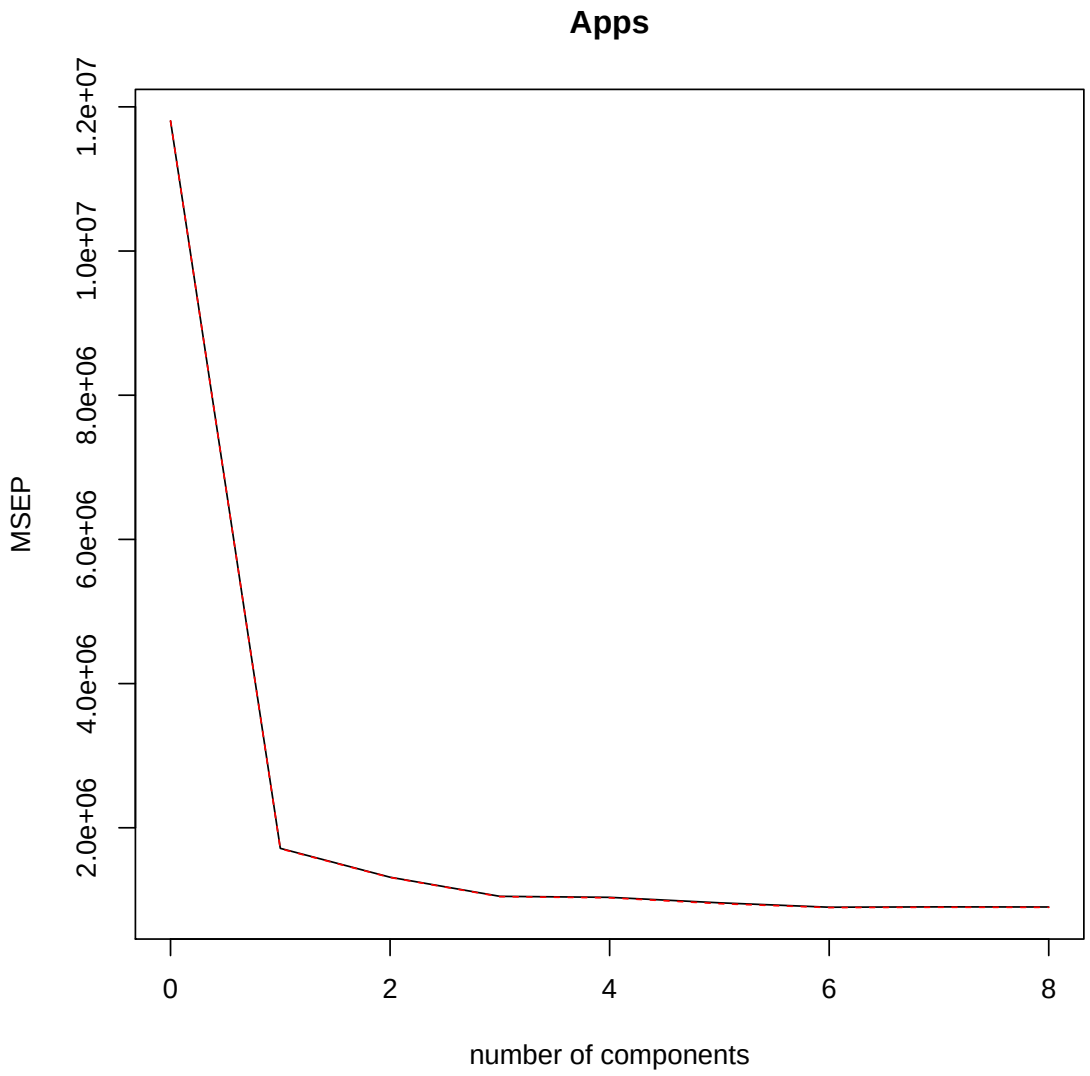
```
cv.prin <- pcr(Apps ~Private + Accept + Enroll + Top10perc + Outstate + Room.Board
               data = train, scale = TRUE, validation = "CV") # library(pls)
validationplot(cv.prin, val.type="MSEP")
```




```
MSEP(cv.prin) # MSE

##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           11805563  9137916  2018144  1567400  1111387  1135906  1157303
## adjCV        11805563  9142203  2012224  1553499  1104288  1128459  1148248
##        7 comps  8 comps
## CV     1168872   933138
## adjCV  1159142   926117

y.hat.pcr <- predict(cv.prin, test, ncomp=8)
(err.pcr <- mean((test$Apps - y.hat.pcr)^2))  # test error

## [1] 1559802
```

(e) PLS model, with M chosen by cross-validation.

```
cv.plsr <- plsr(Apps ~Private + Accept + Enroll + Top10perc + Outstate + Room.Boar
                data = train, scale = TRUE, validation = "CV") # library(pls)
validationplot(cv.plsr, val.type="MSEP")
```




```
MSEP(cv.plsr) # MSE

##         (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV        11805563  1714639  1314103  1049161  1034921   959266   897986
## adjCV     11805563  1709056  1311067  1044698  1028442   947591   892962
##         7 comps  8 comps
## CV       902867   901293
## adjCV    897535   896123

y.hat.plsr <- predict(cv.plsr, test, ncomp=8)
```

```
(err.plsr <- mean((test$Apps - y.hat.plsr)^2))  # test error

## [1] 1559802
```

Evaluate performance of each method in terms of prediction accuracy. Report prediction mean squared errors obtained by cross-validation.

```
##       err.lm err.ridge err.lasso err.pcr err.plsr
## [1,] 1559802   1618175   1574253 1559802  1559802
```

Comment on the results obtained. How accurately can we predict the number of college applications? Is there much difference among the test errors resulting from these five approaches? Which method appears most accurate?

The OLS regression, PCR, PLS yields the lowest testing MSE. Since PCR and PLS do not reduce the dimension the predictors, in other words, they use the model of OLS. So OLS regression appears to be most accurate.