Bruce Campbell ST-617 Homework 2

Tue Jul 12 16:01:39 2016

Chapter 6

Problem 9

In this exercise, we will predict the number of applications received using the other variables in the College data set.

a)

Split the data set into a training set and a test set.

```
rm(list = ls())
library(ISLR)
DF = College
train = sample(nrow(DF), floor(nrow(DF) * 2/3))
DFTrain <- DF[train, ]
DFTest <- DF[-train, ]
```

b)

Fit a linear model using least squares on the training set, and report the test error obtained.

```
names(DF)
```

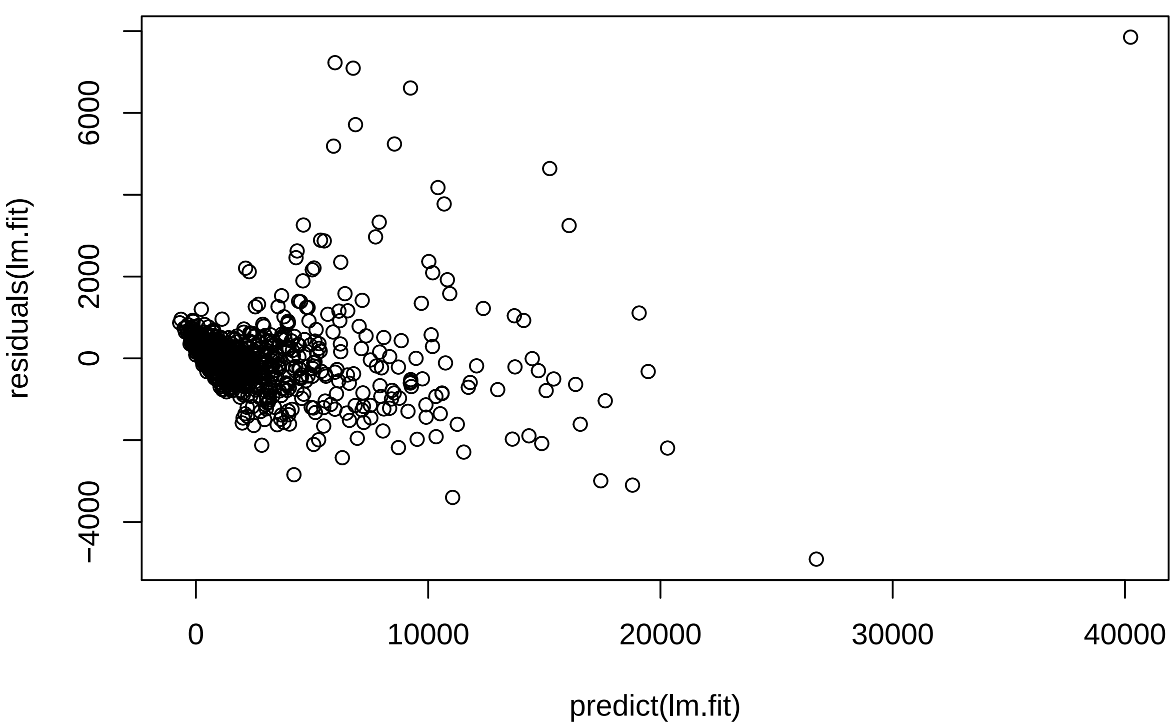
```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "Top10perc"
##  [6] "Top25perc"   "F.Undergrad" "P.Undergrad" "Outstate"    "Room.Board"
## [11] "Books"       "Personal"    "PhD"         "Terminal"    "S.F.Ratio"
## [16] "perc.alumni" "Expend"      "Grad.Rate"
```

```
lm.fit <- lm(Apps ~ ., data = DF)
summary(lm.fit)
```

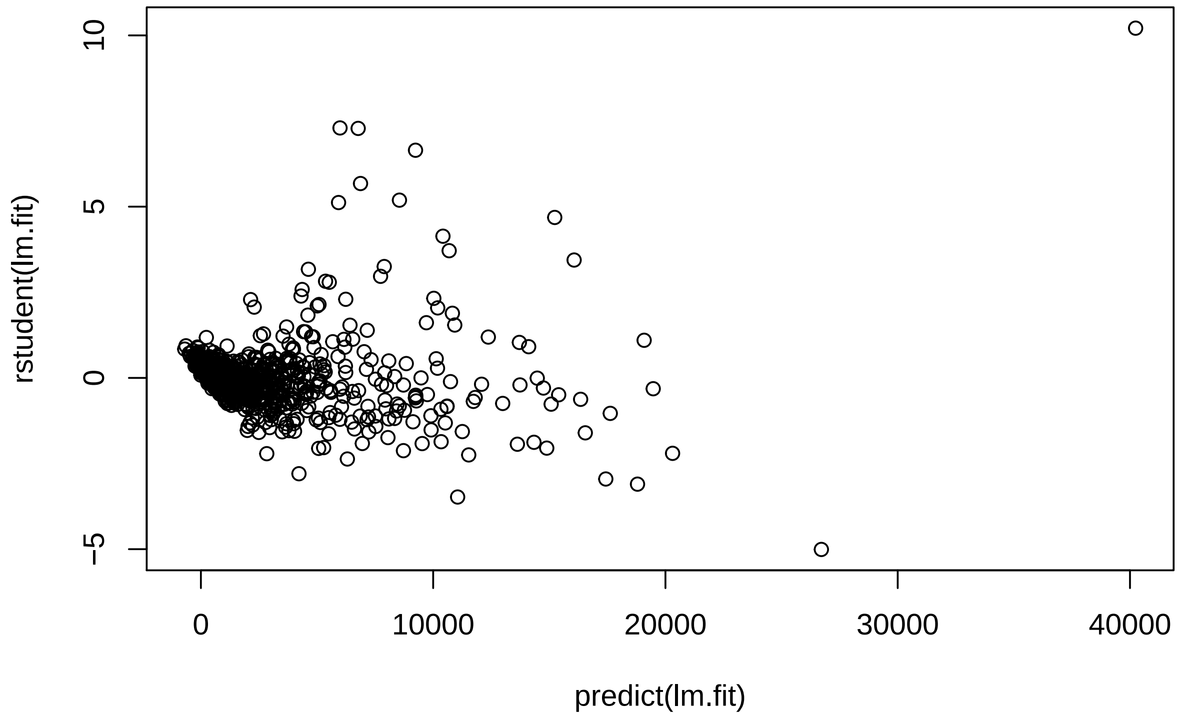
```
##
## Call:
## lm(formula = Apps ~ ., data = DF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4908.8  -430.2   -29.5   322.3  7852.5
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept) -445.08413  408.32855  -1.090 0.276053
## PrivateYes  -494.14897  137.81191  -3.586 0.000358 ***
```

```
## Accept         1.58581    0.04074  38.924  < 2e-16 ***
## Enroll        -0.88069    0.18596  -4.736 2.60e-06 ***
## Top10perc     49.92628    5.57824   8.950  < 2e-16 ***
## Top25perc    -14.23448    4.47914  -3.178 0.001543 **
## F.Undergrad    0.05739    0.03271   1.754 0.079785 .
## P.Undergrad    0.04445    0.03214   1.383 0.167114
## Outstate      -0.08587    0.01906  -4.506 7.64e-06 ***
## Room.Board     0.15103    0.04829   3.127 0.001832 **
## Books          0.02090    0.23841   0.088 0.930175
## Personal       0.03110    0.06308   0.493 0.622060
## PhD           -8.67850    4.63814  -1.871 0.061714 .
## Terminal      -3.33066    5.09494  -0.654 0.513492
## S.F.Ratio     15.38961   13.00622   1.183 0.237081
## perc.alumni    0.17867    4.10230   0.044 0.965273
## Expend         0.07790    0.01235   6.308 4.79e-10 ***
## Grad.Rate      8.66763    2.94893   2.939 0.003390 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1041 on 759 degrees of freedom
## Multiple R-squared:  0.9292, Adjusted R-squared:  0.9276
## F-statistic: 585.9 on 17 and 759 DF,  p-value: < 2.2e-16
```

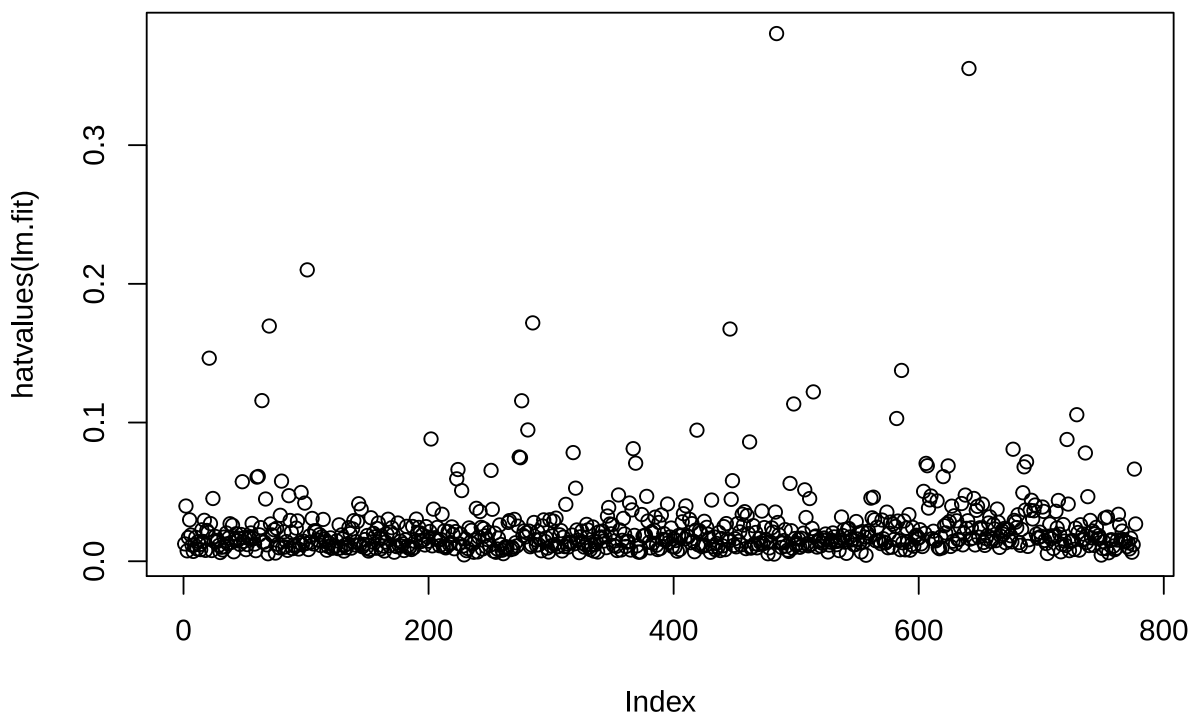
```r
plot(predict(lm.fit), residuals(lm.fit))
```



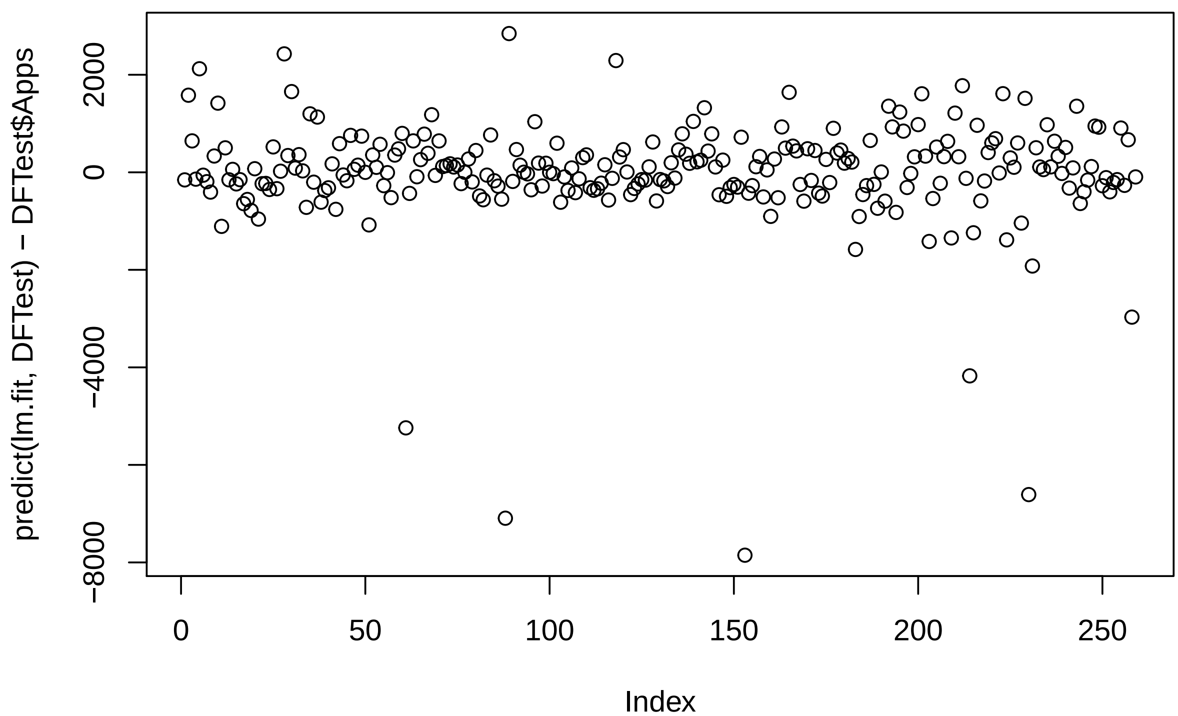
```r
plot(predict(lm.fit), rstudent(lm.fit))
```




```r
plot(hatvalues(lm.fit))
```




```
plot(predict(lm.fit, DFTest) - DFTest$Apps)
```




```
lm.test_mse <- mean((predict(lm.fit, DFTest) - DFTest$Apps)^2)

mse_summary <- data.frame(method = "lm", MSE = lm.test_mse)
```

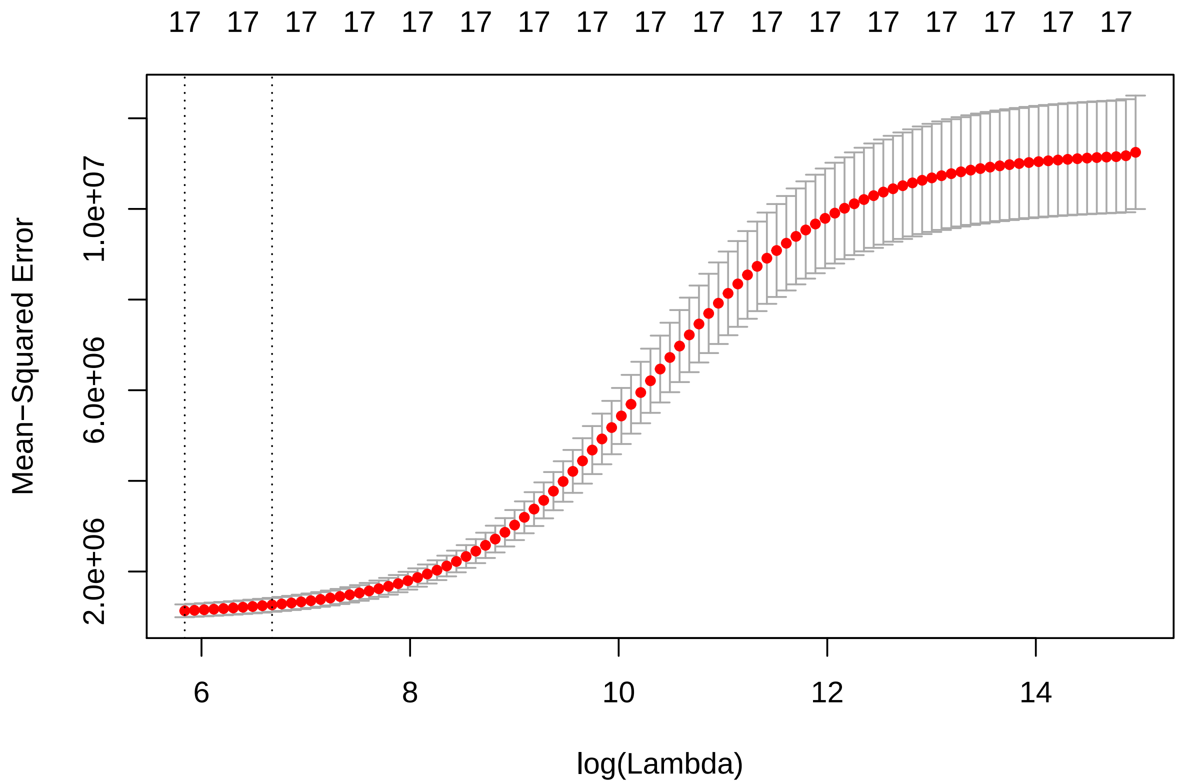
The test set for a linear model is

MSE = `1.2680767\times 10^{6}`

c)

Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

```
library(glmnet)
x_ridge = model.matrix(Apps ~ ., DFTrain)[, -1]
y_ridge = DFTrain$Apps
cv.out = cv.glmnet(x_ridge, y_ridge, alpha = 0)
plot(cv.out)
```




```r
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 343.5104
```

```r
best_ridge = glmnet(x_ridge, y_ridge, alpha = 0, lambda = bestlam)
predict(best_ridge, type = "coefficients", s = bestlam)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -1.420780e+03
## PrivateYes  -5.326368e+02
## Accept       7.428839e-01
## Enroll       6.917269e-01
## Top10perc    2.870298e+01
## Top25perc    1.386333e+00
## F.Undergrad  1.245860e-01
## P.Undergrad  3.684094e-02
## Outstate    -3.543716e-03
## Room.Board   2.335635e-01
## Books        1.471950e-01
## Personal    -3.601055e-02
## PhD         -1.863818e+00
## Terminal    -4.932057e+00
```

```
## S.F.Ratio    1.026248e+01
## perc.alumni -1.057626e+01
## Expend       5.480836e-02
## Grad.Rate    7.628455e+00
```

```r
x_ridge_test = model.matrix(Apps ~ ., DFTest)[, -1]
y_ridge_test = DFTest$Apps

ridge.pred = predict(best_ridge, newx = x_ridge_test)
ridge.test_mse <- mean((ridge.pred - y_ridge_test)^2)



mse_summary <- rbind(mse_summary, data.frame(method = "ridge", MSE = ridge.test_mse))
```

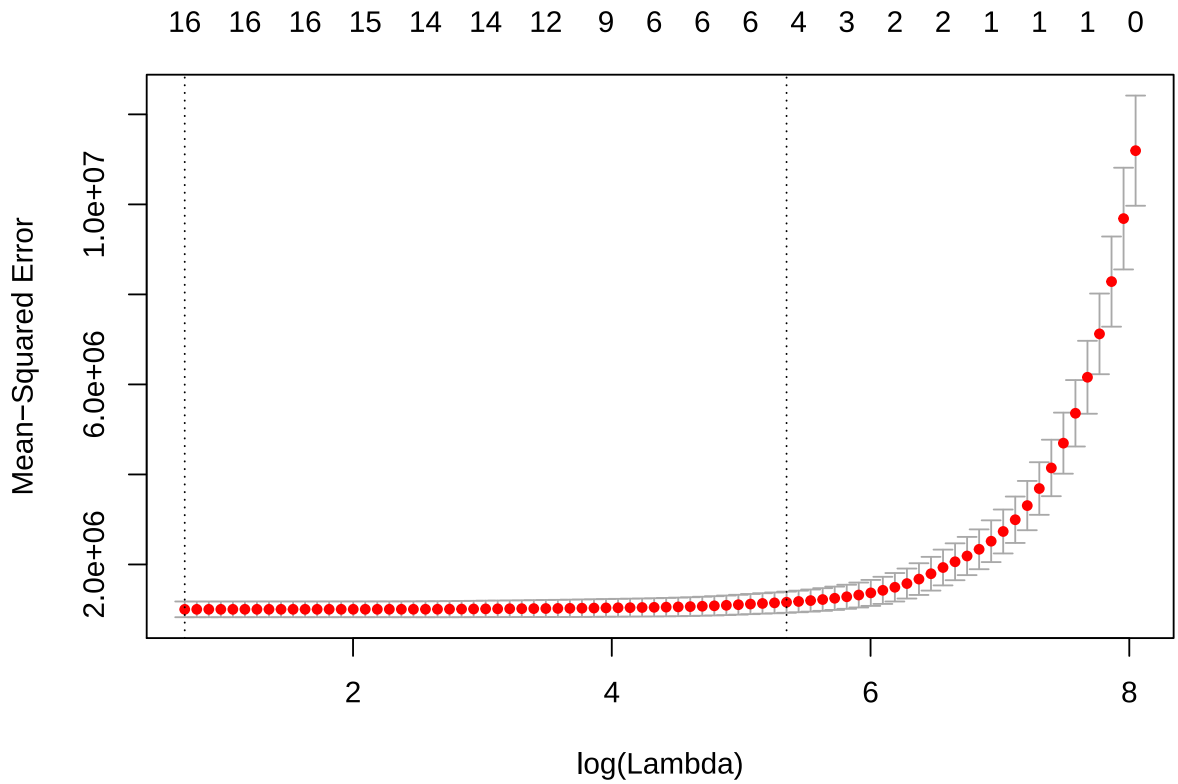
The test set MSE for a ridge regression model where the regularization parameter is set by cross validation is

MSE = `3.0560717\times 10^{6}`

d) Fit a lasso model on the training set, with λ chosen by crossvalidation.

Report the test error obtained, along with the number of non-zero coefficient estimates.

```r
x_lasso = model.matrix(Apps ~ ., DFTrain)[, -1]
y_lasso = DFTrain$Apps
cv.out = cv.glmnet(x_lasso, y_lasso, alpha = 1)
plot(cv.out)
```




```
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 2.011948
```

```
best_lasso = glmnet(x_lasso, y_lasso, alpha = 1, lambda = bestlam)
predict(best_lasso, type = "coefficients", s = bestlam)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept) -422.16694227
## PrivateYes  -621.27194997
## Accept         1.27442885
## Enroll        -0.41570343
## Top10perc     50.30290244
## Top25perc    -12.11420026
## F.Undergrad    0.08665575
## P.Undergrad    0.04289514
## Outstate      -0.04425771
## Room.Board     0.21395925
## Books          0.01996283
## Personal       .
## PhD           -6.28349446
## Terminal      -4.13328862
```

```
## S.F.Ratio      9.97848435
## perc.alumni   -4.63662826
## Expend         0.04614050
## Grad.Rate      5.81481538
```

```
x_lasso_test = model.matrix(Apps ~ ., DFTest)[, -1]
y_lasso_test = DFTest$Apps

lasso.pred = predict(best_lasso, newx = x_lasso_test)
lasso.test_mse <- mean((lasso.pred - y_lasso_test)^2)

coeff_lasso <- predict(best_lasso, type = "coefficients", s = bestlam)[1:18,
    ]
library(pander)
pander(coeff_lasso)
```

Table 1: Table continues below

(Intercept)	PrivateYes	Accept	Enroll	Top10perc	Top25perc
-422.2	-621.3	1.274	-0.4157	50.3	-12.11

Table 2: Table continues below

F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD
0.08666	0.0429	-0.04426	0.214	0.01996	0	-6.283

Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
-4.133	9.978	-4.637	0.04614	5.815

```
mse_summary <- rbind(mse_summary, data.frame(method = "lasso", MSE = lasso.test_mse))
```

The test set MSE for a lasso regression model where the regularization parameter is set by cross validation is

MSE = `1.7529978\times 10^{6}`

All but one of the predictors was incuded in the lasso model with the best lambda selected by cross validation. A more parsimonious model may help with inference so using the cross validation MSE chart we below we bump up lambda to $e^4.5$ to get a model with fewer predictors

```
bestlam = exp(4.2)

best_lasso = glmnet(x_lasso, y_lasso, alpha = 1, lambda = bestlam)
predict(best_lasso, type = "coefficients", s = bestlam)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                          1
```

```
## (Intercept) -796.75905060
## PrivateYes  -404.55485351
## Accept         1.19260455
## Enroll         .
## Top10perc     30.10557021
## Top25perc      .
## F.Undergrad    0.05566246
## P.Undergrad    .
## Outstate       .
## Room.Board     0.10175944
## Books          .
## Personal       .
## PhD            .
## Terminal       .
## S.F.Ratio      .
## perc.alumni   -1.29990394
## Expend         0.02395187
## Grad.Rate      .
```

```r
x_lasso_test = model.matrix(Apps ~ ., DFTest)[, -1]
y_lasso_test = DFTest$Apps

lasso.pred = predict(best_lasso, newx = x_lasso_test)
lasso.test_mse <- mean((lasso.pred - y_lasso_test)^2)

mse_summary <- rbind(mse_summary, data.frame(method = "lasso-reduced", MSE = lasso.test_mse))

coeff_lasso <- predict(best_lasso, type = "coefficients", s = bestlam)[1:18,
    ]
library(pander)
pander(coeff_lasso)
```

Table 4: Table continues below

(Intercept)	PrivateYes	Accept	Enroll	Top10perc	Top25perc
-796.8	-404.6	1.193	0	30.11	0

Table 5: Table continues below

F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD
0.05566	0	0	0.1018	0	0	0

Terminal	S.F.Ratio	perc.alumni	Expend	Grad.Rate
0	0	-1.3	0.02395	0

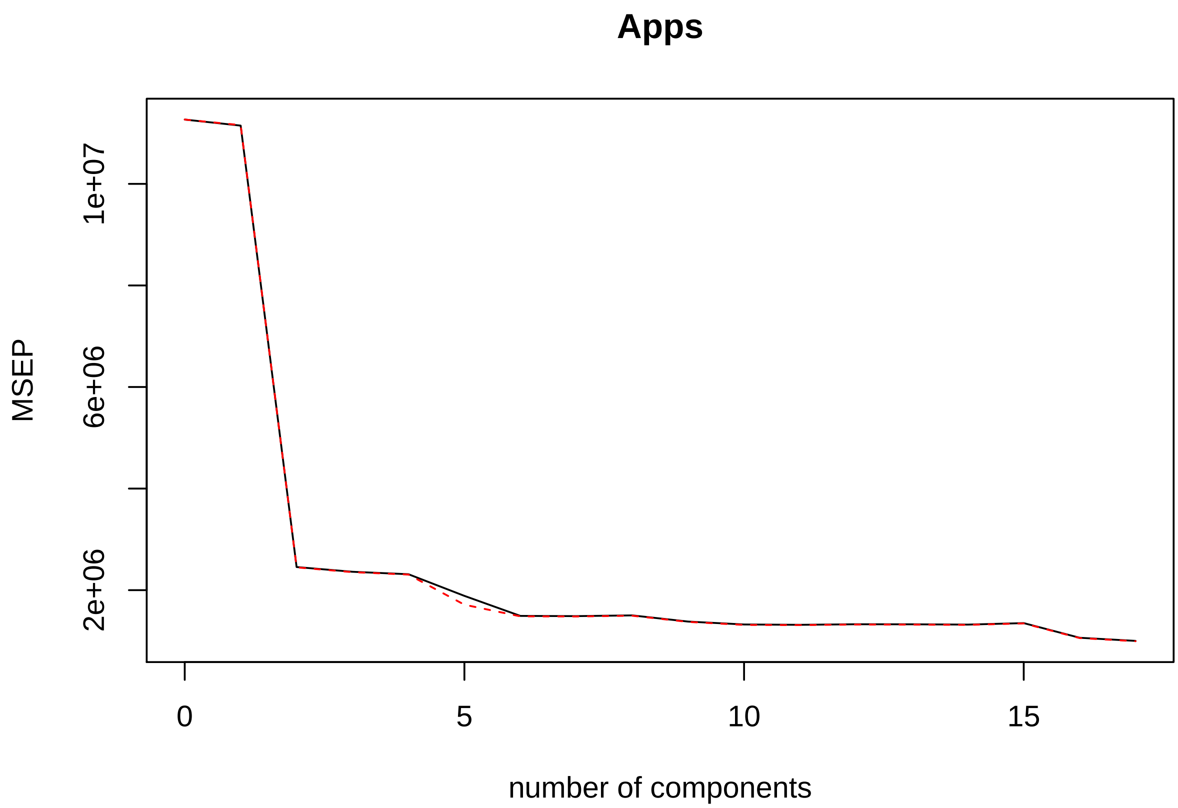
e)

Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.

```r
pcr.test_mse = 1
library(pls)
pcr.fit = pcr(Apps ~ ., data = DFTrain, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 518 17
##  Y dimension: 518 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            3357     3339     1567     1537     1521     1374     1222
## adjCV         3357     3340     1565     1534     1519     1309     1218
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1220     1226     1175      1151      1148      1153      1152
## adjCV     1217     1223     1173      1147      1146      1150      1150
##        14 comps  15 comps  16 comps  17 comps
## CV         1150      1162      1031    1000.5
## adjCV      1147      1160      1028     997.3
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      32.408    57.95    65.52    71.30    76.23    81.08    84.78
## Apps    2.267    79.23    80.18    80.82    86.60    87.70    87.73
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.91    90.71     93.01     95.13     96.90     97.93     98.89
## Apps    87.77    88.59     89.06     89.10     89.13     89.23     89.25
##       15 comps  16 comps  17 comps
## X        99.41     99.81    100.00
## Apps     89.25     91.41     92.13
```

```r
validationplot(pcr.fit, val.type = "MSEP")
```




```
pcr.pred = predict(pcr.fit, DFTest, ncomp = 8)
pcr.test_mse <- mean((pcr.pred - DFTest$Apps)^2)

mse_summary <- rbind(mse_summary, data.frame(method = "pcr", MSE = pcr.test_mse))
```

The test set MSE for a principal components regression is

MSE = `4.5624813\times 10^{6}`

f)

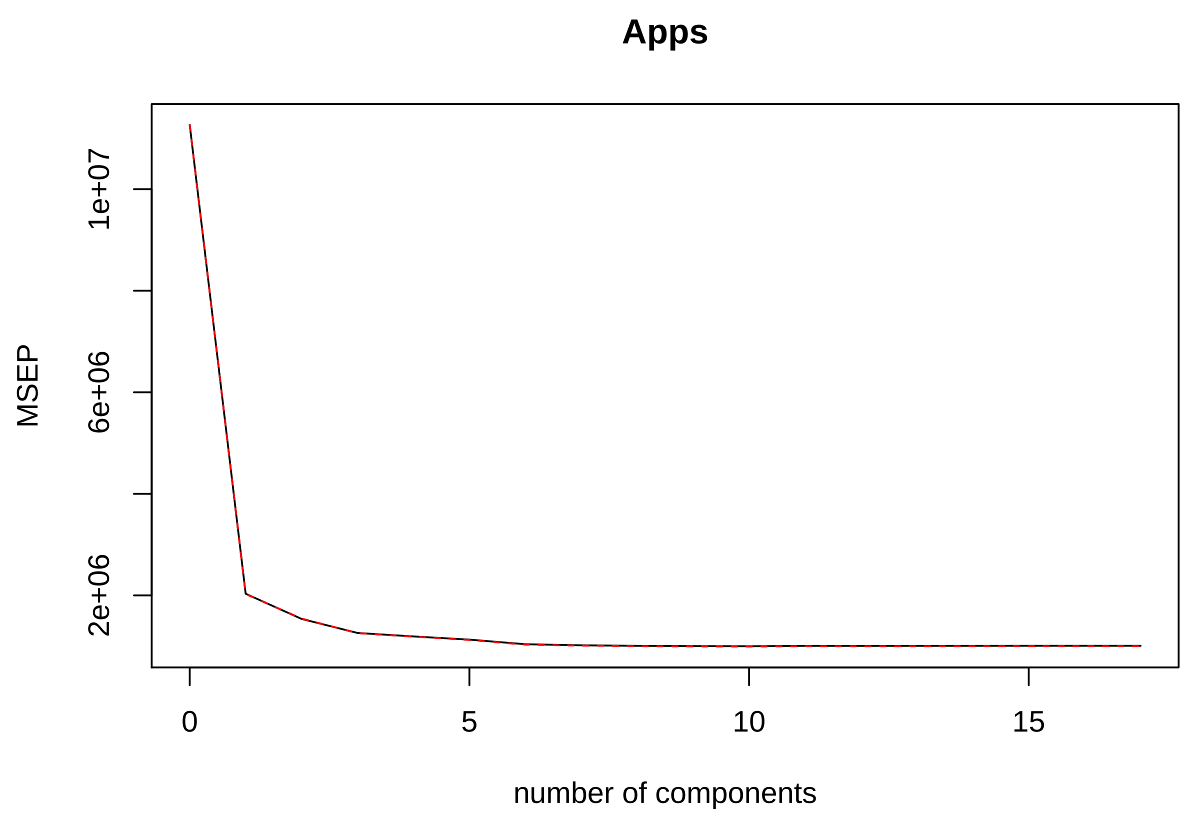
`Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.`

```
plsr.fit = plsr(Apps ~ ., data = DFTrain, scale = TRUE, validation = "CV")
summary(plsr.fit)
```

```
## Data:    X dimension: 518 17
##  Y dimension: 518 1
## Fit method: kernelpls
## 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            3357     1425     1239     1122     1091     1062     1018
## adjCV         3357     1423     1242     1121     1090     1059     1014
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1008   1002.9   1000.5       999    1002.8      1002    1002.7
## adjCV     1005    999.8    997.4       996     999.5       999     999.3
##        14 comps  15 comps  16 comps  17 comps
## CV       1003.0    1003.3    1003.3    1003.3
## adjCV     999.6     999.9     999.9     999.9
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       25.76    44.75    63.11    68.68    72.23    74.37    78.96
## Apps    82.57    86.87    89.38    90.10    90.96    91.81    91.97
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       81.21    83.23     86.38     88.50     91.31     93.52     96.29
## Apps    92.04    92.08     92.10     92.12     92.13     92.13     92.13
##       15 comps  16 comps  17 comps
## X        97.68     99.46    100.00
## Apps     92.13     92.13     92.13
```

```r
validationplot(plsr.fit, val.type = "MSEP")
```



```r
plsr.pred = predict(plsr.fit, DFTest, ncomp = 8)
plsr.test_mse <- mean((plsr.pred - DFTest$Apps)^2)


mse_summary <- rbind(mse_summary, data.frame(method = "plsr", MSE = plsr.test_mse))
```

The test set MSE for a partial least quares regression is

MSE = `1.7637545\times 10^{6}`

g)

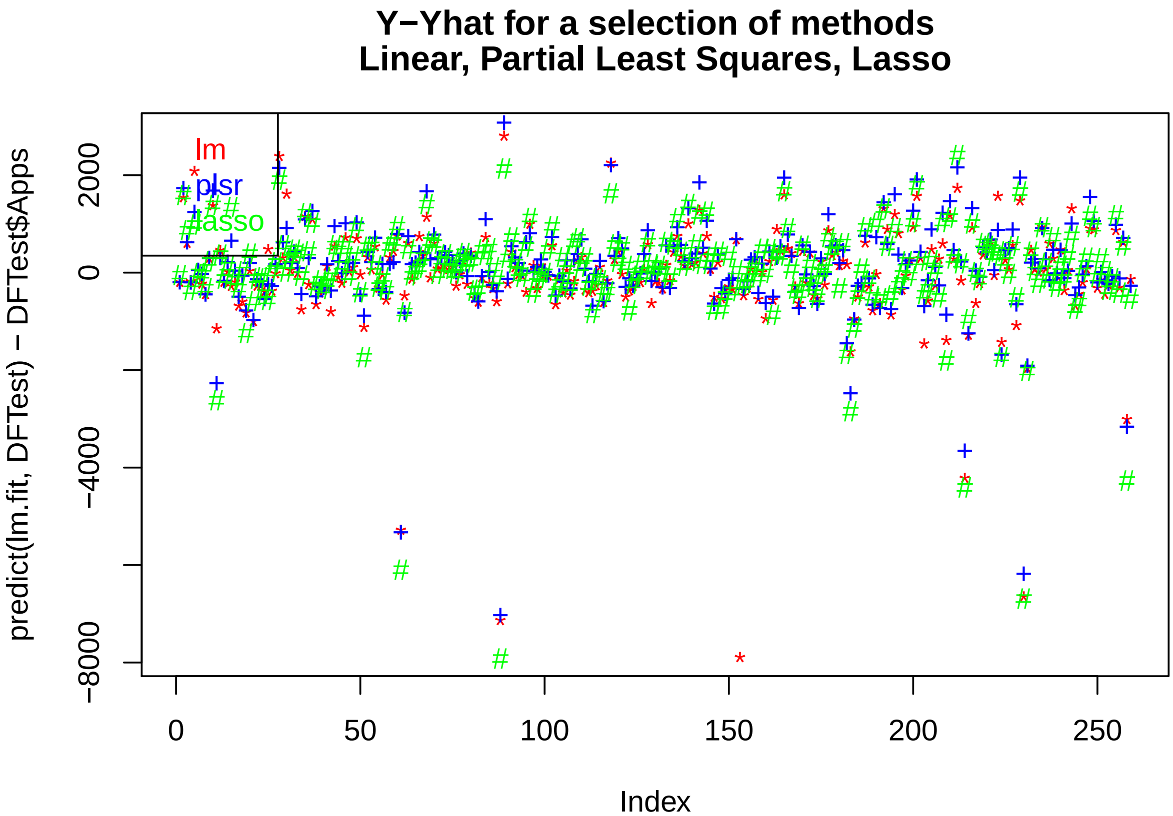
`Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?`

```r
pander(mse_summary)
```

method	MSE
lm	1268077
ridge	3056072
lasso	1752998
lasso-reduced	2006021
pcr	4562481
plsr	1763755

We see the linear model is the best but that the lasso is competitive

```r
plot(predict(lm.fit, DFTest) - DFTest$Apps, pch = "*", col = "red")
points(plsr.pred - DFTest$Apps, pch = "+", col = "blue")
points(lasso.pred - y_lasso_test, pch = "#", col = "green")
legend("topleft", title.col = "black", c("lm", "plsr", "lasso"), text.col = c("red",
    "blue", "green"), text.font = 1, cex = 1)
title(c("Y-Yhat for a selection of methods", "Linear, Partial Least Squares, Lasso"))
```


Y−Yhat for a selection of methods
Linear, Partial Least Squares, Lasso
lm
plsr
lasso
predict(lm.fit, DFTest) − DFTest$Apps
2000
0
−4000
−8000
0
50
100
150
200
250
Index