# AARMS Statistical Learning
# Assignment 3 Solutions-Part II

**3. Problem 5, page 261** It is well known that ridge regresson tends to give similar coefficient values to correlated variables, whereas the lasso may give quite different coefficien values to correlated variables. We will now explore this property in a very simple setting.

Suppose that $n = 2$, $p = 2$, $x_{11} = x_{12}$, $x_{21} = x_{22}$. Furthermore, suppose that $y_1 + y_2 = 0$ and $x_{11} + x21 = 0$ and $x_{12} + x_{22} = 0$, so that the estimate for the intercept in a least squares, ridge regression, or lasso model is zero: $\hat{beta}_0 = 0$.

(a) (2 points) Write out the ridge regression optimization problem in this setting.

*Answer:*

A general form of Ridge regression optimization looks like

$$Minimize : \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \sum_{j=1}^{p}\hat{\beta}_j x_j)^2 + \lambda \sum_{i=1}^{p}\hat{\beta}_j^2 \tag{1}$$

In this case, $\hat{\beta}_0 = 0$ and $n = p = 2$. So, the optimization looks like

$$Minimize : (y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2). \tag{2}$$

(b) (2 points) Argue that in this setting, the ridge coefficient estimates satisfy $\hat{\beta}_1 = \hat{\beta}_2$.

*Answer:*

Given the situations that $x_{11} = x_{12} = x_1$, $x_{21} = x_{22} = x2$, take the derivatives of the expression in (a) with respect to both $\hat{\beta}_1$ and $\hat{\beta}_2$ and setting them equal zero, then we get

$$\hat{\beta}_1^* = \frac{x_1 y_1 + x_2 y_2 - \hat{\beta}_2^*(x_1^2 + x_2^2)}{\lambda + x_1^2 + x_2^2} \tag{3}$$

$$\hat{\beta}_2^* = \frac{x_1 y_1 + x_2 y_2 - \hat{\beta}_1^*(x_1^2 + x_2^2)}{\lambda + x_1^2 + x_2^2} \tag{4}$$

The symmetry form in the above formulae suggests that $\hat{\beta}_1 = \hat{\beta}_2$.

(c) (2 points) Write down the lasso optimization problem in this setting.

*Answer:*

The optimization looks like

$$Minimize : (y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|) \tag{5}$$

(d) (5 points) Argue that in this setting, the lasso coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ are not unique - in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions.

*Answer:* The Lasso contraint takes the form $|\hat{\beta}_1| + |\hat{\beta}_2| < s$, which plotted takes the shape of a diamond centered at origin $(0, 0)$. Next consider the squared optimization constrain

$(y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2$. We use the facts $x_{11} = x_{12}$, $x_{21} = x_{22}$, $x_{11} + x_{21} = 0$, $x_{12} + x_{22} = 0$, and $y_1 + y_2 = 0$ to simlify it to minimize: $2(y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_{11})^2$.

This optimization problem has a simple solution: $\hat{\beta}_1 + \hat{\beta}_2 = \frac{y_1}{x_{11}}$. This is a line parallel to the edge of Lasso-diamond $\hat{\beta}_1 + \hat{\beta}_2 = s$. Now the soluitons to the original Lasso optimization problem are contours of the function $(y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_{11})^2$ that touch the Lasso-diamond $\hat{\beta}_1 + \hat{\beta}_2 = s$. Finally, as $\hat{\beta}_1$ and $\hat{\beta}_2$ vary along the line $\hat{\beta}_1 + \hat{\beta}_2 = \frac{y_1}{x_{11}}$, these contours touch the Lasso-diamond edge $\hat{\beta}_1 + \hat{\beta}_2 = s$ at different points. As a result, the enrire edge $\hat{\beta}_1 + \hat{\beta}_2 = s$ is a potential solution to the Lasso optimization problem!

Similar argument can be made for the opposite Lasso-diamond edge: $\hat{\beta}_1 + \hat{\beta}_2 = -s$.

Thus, the Lasso problem does not have a unique solution. The general form of solution is

$$\hat{\beta}_1 + \hat{\beta}_2 = s; \hat{\beta}_1 \geq 0; \hat{\beta}_2 \geq 0; \text{ and } \hat{\beta}_1 + \hat{\beta}_2 = -s; \hat{\beta}_1 \leq 0; \hat{\beta}_2 \leq 0. \quad (6)$$

**4. Problem 8, page 262-263, "In this exercise, we will generate simulated data, and will"**

(a) (2 points) Use the rnorm function to generate a predictor $X$ of length $n = 100$, as wells as noise vector $\epsilon$ of length $n = 100$

*Answer:*

```
set.seed(100)
x = rnorm(100, 2, 2)
epsilon = rnorm(100, 0, 1)
```

(b) (2 points) Generate a response vector $Y$ of length $n = 100$ according to the model $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$, where $\beta_0, \beta_1, \beta_2$, and $\beta_3$ are constants of your choice.

```
beta = sample(1:100, 4, replace=TRUE)
y= beta[1]+beta[2]*x+beta[3]*x^2+beta[4]*x^3+epsilon
```

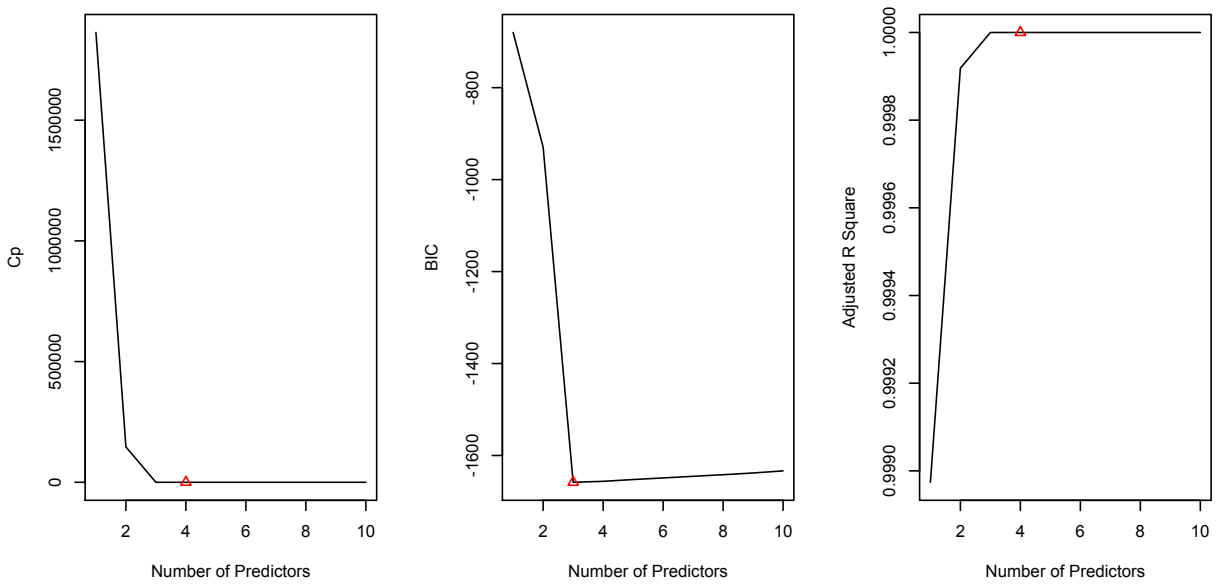
(c) (5 points) Use the regsubsets() function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model obtained accordng to Cp, BIC and adjusted $R^2$? Show some plotsto provide evidence for your answer. Note you will need to use the data.frame() function to create a single data set containing both $X$ and $Y$.

```
library(leaps)
best.full=regsubsets(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10)
                     data=data.frame(x=x,y=y),nvmax=10)
best.summary=summary(best.full)
par(mfrow=c(1,3))
plot(1:10, best.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")
cp.min=min(best.summary$cp)
points(c(1:10)[best.summary$cp==cp.min], cp.min, pch=2, col="red")

plot(1:10, best.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")
bic.min=min(best.summary$bic)
```

```
points(c(1:10)[best.summary$bic==bic.min], bic.min, pch=2, col="red")

plot(1:10, best.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square",
type="l")
adjr2.max=max(best.summary$adjr2)
points(c(1:10)[best.summary$adjr2==adjr2.max], adjr2.max, pch=2, col="red")
```




The best model selected by Cp has four predictors: $X$, $X^2$, $X^3$ and $X^6$. The best model selected by BIC has three predictors: $X$, $X^2$ and $X^3$. The best model selected by adjusted $R^2$ is the same as the one selected by Cp, i.e. a model with predictors $X$, $X^2$, $X^3$ and $X^6$.

(d). (5 points) Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c).

*Answer:* Run the R codes below. We got the same results as those in (c).

```
##### Stepwise Forward Selection #####
best.frd=regsubsets(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),
                    data=data.frame(x=x,y=y),nvmax=10, method="forward")
frd.summary=summary(best.frd)
par(mfrow=c(1,3))
plot(1:10, frd.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")
cp.min=min(frd.summary$cp)
points(c(1:10)[frd.summary$cp==cp.min], cp.min, pch=2, col="red")

plot(1:10, frd.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")
bic.min=min(frd.summary$bic)
points(c(1:10)[frd.summary$bic==bic.min], bic.min, pch=2, col="red")

plot(1:10, frd.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square", type
```

```
adjr2.max=max(frd.summary$adjr2)
points(c(1:10)[frd.summary$adjr2==adjr2.max], adjr2.max, pch=2, col="red")

### Stepwise Backward Selection ###
best.bkd=regsubsets(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),
                    data=data.frame(x=x,y=y),nvmax=10, method="backward")
bkd.summary=summary(best.bkd)
par(mfrow=c(1,3))
plot(1:10, bkd.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")
cp.min=min(bkd.summary$cp)
points(c(1:10)[bkd.summary$cp==cp.min], cp.min, pch=2, col="red")

plot(1:10, bkd.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")
bic.min=min(bkd.summary$bic)
points(c(1:10)[bkd.summary$bic==bic.min], bic.min, pch=2, col="red")

plot(1:10, bkd.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square", type
adjr2.max=max(bkd.summary$adjr2)
points(c(1:10)[bkd.summary$adjr2==adjr2.max], adjr2.max, pch=2, col="red")
```

(e) (5 points) Now fit a lasso model to the simulated data, again using $X$, $X^2, \ldots X^{10}$ as predictors. Use cross-validation to select the optimal value of $\lambda$. Create plots of the cross-validation error as a function of $\lambda$. Report the resulting coefficient estimates, and discuss the results obtained.

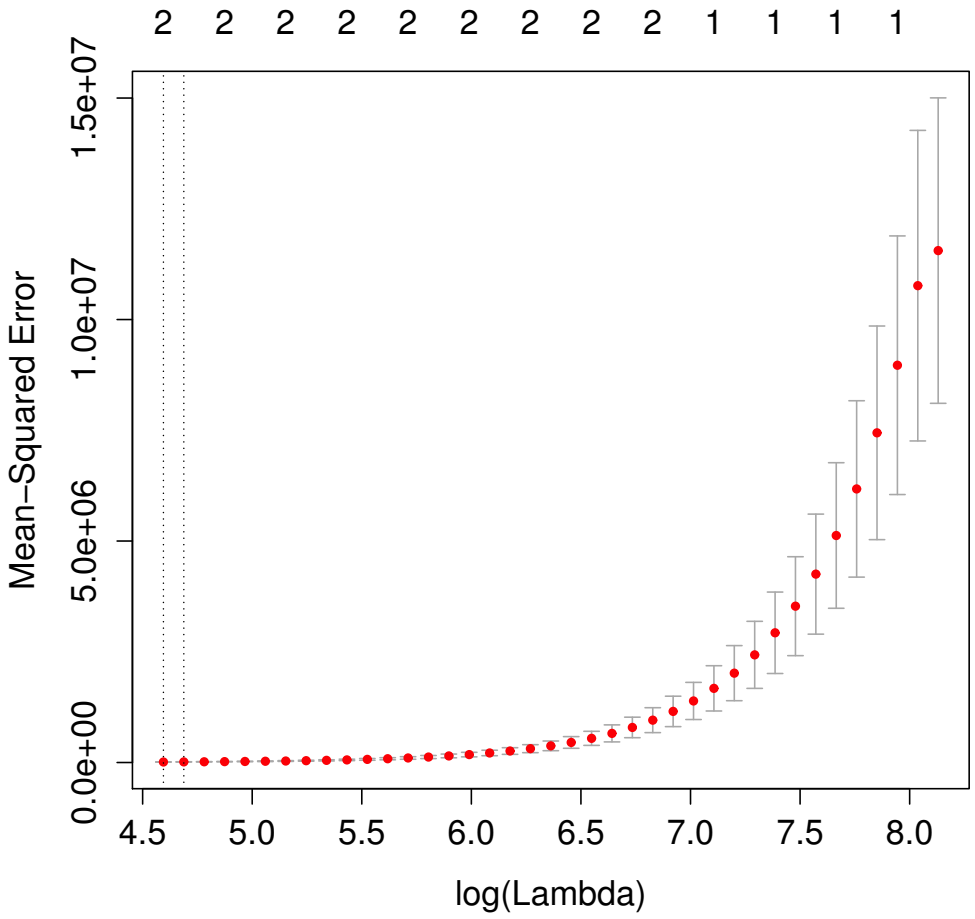
*Answer:* Use the below R codes.

```
library(glmnet)
set.seed(100)
x=cbind(x,x^2,x^3,x^4,x^5,x^6,x^7,x^8,x^9,x^10)
y=y

### Cross-validation to choose lambda ###
lasso.cv = cv.glmnet(x,y, alpha=1)
lasso.cv$lambda.min
lasso.cv$lambda.1se
plot(lasso.cv)

### Refit the model using the chosen lambda ###
lasso.mod=glmnet(x,y,alpha=1, lambda=lasso.cv$lambda.min)
coef(lasso.mod)[,1]
```

```
> lasso.cv$lambda.min
[1] 98.97694
> lasso.cv$lambda.1se
[1] 108.6271
```




```
> coef(lasso.mod)[,1]
(Intercept)           x
  124.03894     0.00000    40.92403    42.60483     0.00000     0.00000     0.00000

    0.00000     0.00000
```

The plot includes the cross-validation curve (red dotted line), and upper and lower standard deviation curves along the sequence of $\lambda$ values. Two selected $\lambda$ values are indicated by the vertical dotted line: lambda giving the minimum cv error and the lambda within one standard devation of the minimum cv error: in this example, they are 98.97694 and 108.6271 respectively.

With the value of $\lambda$ giving the minimum cv error, the Lasso shrinks the majority predictors to zero, and only leaves $X^2$ and $X^3$ nozero.

(f) (5 points) Now generate a response vector $Y$ according to the model $Y = \beta_0 + beta_7 X^7 + \epsilon$, and perform best subset selection an the lasso. Discuss the results obtained.

*Answer:*

```
# The results from the best subset selection
> best.summary
Subset selection object
```

```
Call: regsubsets.formula(Y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +
    I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data.frame(x = x,
    Y = Y), nvmax = 10)
..........
1 subsets of each size up to 10
Selection Algorithm: exhaustive
          x   I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9) I(x^10)
1  ( 1 )  " " " "    " "    " "    " "    " "    "*"    " "    " "    " "
2  ( 1 )  "*" " "    " "    " "    " "    " "    "*"    " "    " "    " "
3  ( 1 )  " " " "    " "    " "    " "    " "    "*"    "*"    "*"    " "
4  ( 1 )  " " " "    " "    "*"    "*"    " "    "*"    " "    " "    "*"
5  ( 1 )  " " " "    "*"    "*"    " "    "*"    "*"    "*"    " "    " "
6  ( 1 )  " " "*"    "*"    "*"    " "    "*"    "*"    "*"    " "    " "
7  ( 1 )  " " "*"    "*"    "*"    " "    "*"    "*"    " "    "*"    "*"
8  ( 1 )  " " " "    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
9  ( 1 )  "*" " "    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
10  ( 1 ) "*" "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"    "*"
# Coefficients from Lasso
> coef(lasso.mod)[,1]
 (Intercept)           x
4.706655e+04 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 2.247459e+
```

The best subset selected using Cp and BIC is the set with two predictors: $X$ and $X^7$. The Lasso regression zeros $X_1, \ldots, X_6$.

**5. Problem 9, page 263, "In this exercise, we will predict the number of applications"**

(a) (2 points) Split the data set into a training set and a test set.

*Answer:*

```
# load and split the College data
library(ISLR)
set.seed(11)
sum(is.na(College))

train.size = dim(College)[1] / 2
train = sample(1:dim(College)[1], train.size)
test = -train
College.train = College[train, ]
College.test = College[test, ]
```

(b) (3 points) Fit a linear model using least squares on the training set, and report the test erro obtained.

*Answer:*

```
lm.fit = lm(Apps~., data=College.train)
lm.pred = predict(lm.fit, College.test)
mean((College.test[, "Apps"] - lm.pred)^2)
```

The result is

```
> mean((College.test[, "Apps"] - lm.pred)^2)
[1] 1538442
```

(c) (3 points) Fit a ridge regression model on the triaing set, with $\lambda$ chosen by cross-validation. Report the test error obtained.

*Answer:*

```
train.mat = model.matrix(Apps~., data=College.train)
test.mat = model.matrix(Apps~., data=College.test)
grid = 10 ^ seq(4, -2, length=100)
mod.ridge = cv.glmnet(train.mat, College.train[, "Apps"],
alpha=0, lambda=grid, thresh=1e-12)
lambda.best = mod.ridge$lambda.min

ridge.pred = predict(mod.ridge, newx=test.mat, s=lambda.best)
mean((College.test[, "Apps"] - ridge.pred)^2)
```

The results are

```
> lambda.best
[1] 18.73817

> mean((College.test[, "Apps"] - ridge.pred)^2)
[1] 1608859
```

(d) (3 points) Fit a lasso model on the training set, with $\lambda$ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates.

*Answer:*

```
mod.lasso = cv.glmnet(train.mat, College.train[, "Apps"],
alpha=1, lambda=grid, thresh=1e-12)
lambda.best = mod.lasso$lambda.min
lambda.best

lasso.pred = predict(mod.lasso, newx=test.mat, s=lambda.best)
mean((College.test[, "Apps"] - lasso.pred)^2)

mod.lasso = glmnet(model.matrix(Apps~., data=College),
College[, "Apps"], alpha=1)
predict(mod.lasso, s=lambda.best, type="coefficients")
```

The results are

```
> lambda.best
[1] 21.54435

> mean((College.test[, "Apps"] - lasso.pred)^2)
[1] 1635280
```

```
> predict(mod.lasso, s=lambda.best, type="coefficients")
19 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept) -6.038452e+02
(Intercept)  .
PrivateYes  -4.235413e+02
Accept       1.455236e+00
Enroll      -2.003696e-01
Top10perc    3.367640e+01
Top25perc   -2.403036e+00
F.Undergrad  .
P.Undergrad  2.086035e-02
Outstate    -5.781855e-02
Room.Board   1.246462e-01
Books        .
Personal     1.832912e-05
PhD         -5.601313e+00
Terminal    -3.313824e+00
S.F.Ratio    4.478684e+00
perc.alumni -9.796600e-01
Expend       6.967693e-02
Grad.Rate    5.159652e+00
```

(e) (3 points) Fit a PCR model on the training set, with $M$ chosen by cross-validation. Report the test error obtained, along with the value of $M$ selected by cross-validation.

*Answer:*

```
library(pls)
pcr.fit = pcr(Apps~., data=College.train, scale=T, validation="CV")
validationplot(pcr.fit, val.type="MSEP")

pcr.pred = predict(pcr.fit, College.test, ncomp=10)
mean((College.test[, "Apps"] - data.frame(pcr.pred))^2)
```

The results are:

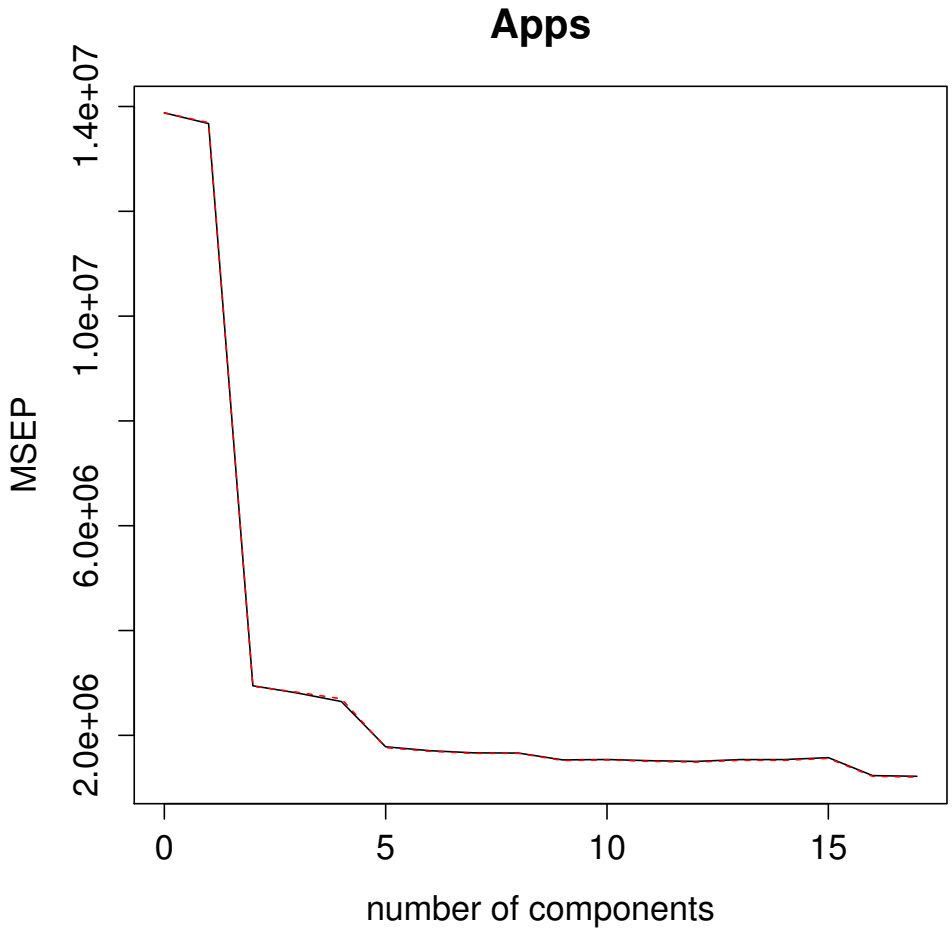
```
> mean((College.test[, "Apps"] - data.frame(pcr.pred))^2)
[1] 3014496
```

(f) (3 points) Fit a PLS model on the training set, with $M$ chosen by cross-validation. Report the test error obtained, along with the value of $M$ selected by cross-validation.

*Answer:*

```
pls.fit = plsr(Apps~., data=College.train, scale=T,
validation="CV")
validationplot(pls.fit, val.type="MSEP")

pls.pred = predict(pls.fit, College.test, ncomp=10)
mean((College.test[, "Apps"] - data.frame(pls.pred))^2)
```




The results are:

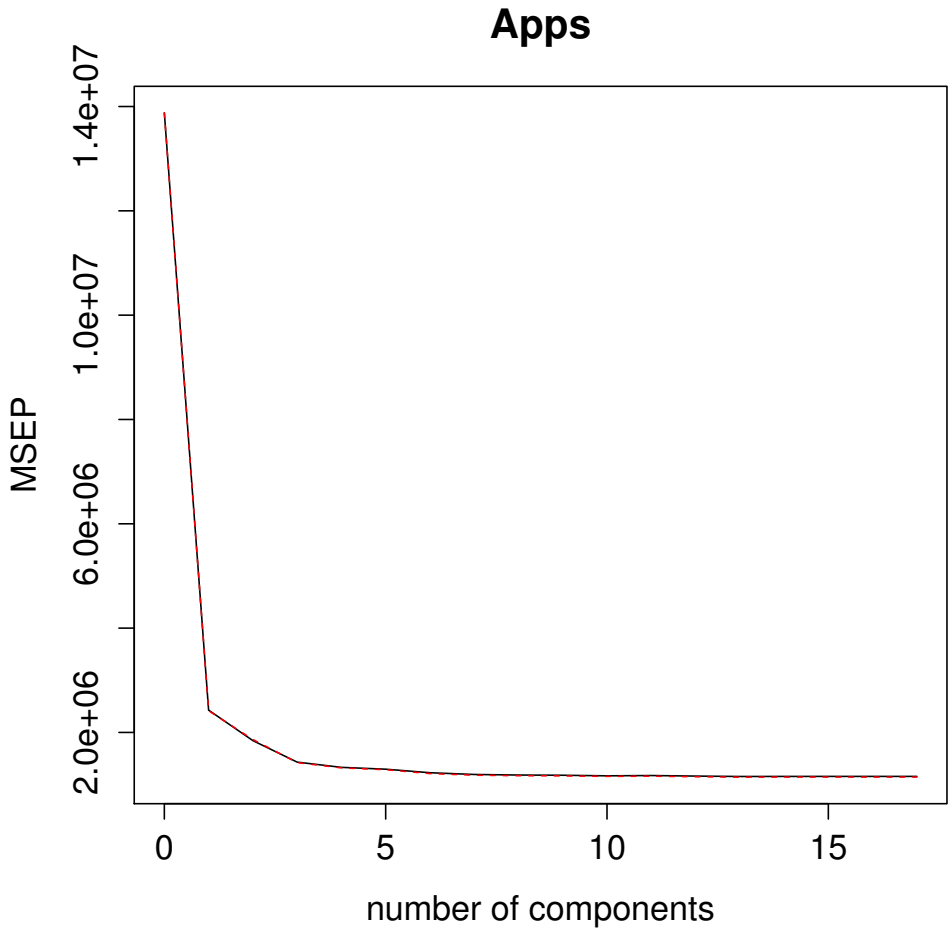
```
> mean((College.test[, "Apps"] - data.frame(pls.pred))^2)
[1] 1508987
```

(g) (3 points) Comment on the results obtained. How accurately can we predict the number of college aplications received? Is there much difference among the test errors resulting from these five approaches?

*Answer:*

```
test.avg = mean(College.test[, "Apps"])
lm.test.r2 = 1 - mean((College.test[, "Apps"] - lm.pred)^2)/mean((College.test[, "Apps"]
ridge.test.r2 = 1 - mean((College.test[, "Apps"] -ridge.pred)^2) /mean((College.test[, "
lasso.test.r2 = 1 - mean((College.test[, "Apps"] -lasso.pred)^2) /mean((College.test[, "
pcr.test.r2 = 1 - mean((College.test[, "Apps"] -data.frame(pcr.pred))^2) /mean((College.
pls.test.r2 = 1 - mean((College.test[, "Apps"] -data.frame(pls.pred))^2) /mean((College.
barplot(c(lm.test.r2, ridge.test.r2, lasso.test.r2, pcr.test.r2, pls.test.r2), col="red"
```

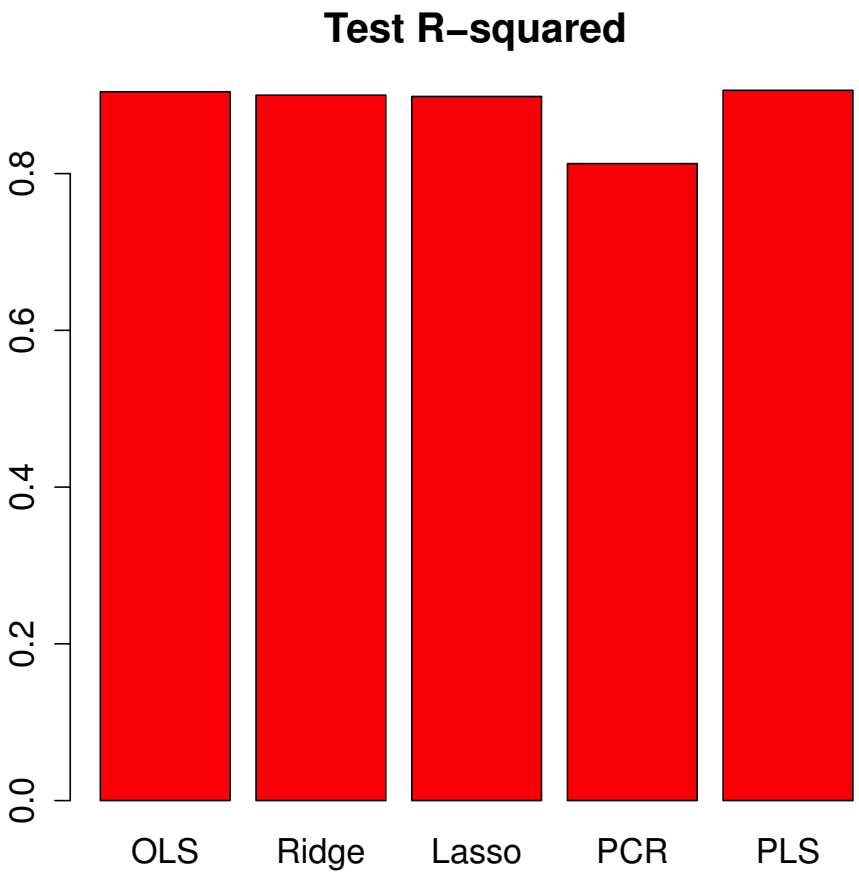
The results for LS, Lasso, Ridge are comparable. Lasso reduces the variables "F. Undergrade" and "Books" variables to zero and shrinks coefficients of other variables. The plot shows the test $R^2$ for all the models. PCR has a smallest test $R^2$. Except PCR, all models predict college applications with high accuracy.




6. **Problem 6, page 299, "In this exercise, you will further analyze the Wage data"**
(a) (5 points) Perform polynomial regression to predict "wage" using "age". Use cross-validation to select the optimal degree $d$ for the polynomial. What degree was chosen, and how does this compare to the results of hypothesis testing using ANOVA? Make a plot of the resulting polynomial fit to the data.

*Answer:*

```
set.seed(100)
library(ISLR)
library(boot)
all.deltas = rep(NA, 10)
for (i in 1:10) {
  glm.fit = glm(wage~poly(age, i), data=Wage)
  all.deltas[i] = cv.glm(Wage, glm.fit, K=10)$delta[2]
}
plot(1:10, all.deltas, xlab="Degree", ylab="CV error", type="l", pch=20, lwd=2, ylim=c(1
min.point = min(all.deltas)
sd.points = sd(all.deltas)
abline(h=min.point + 0.2 * sd.points, col="red", lty="dashed")
```

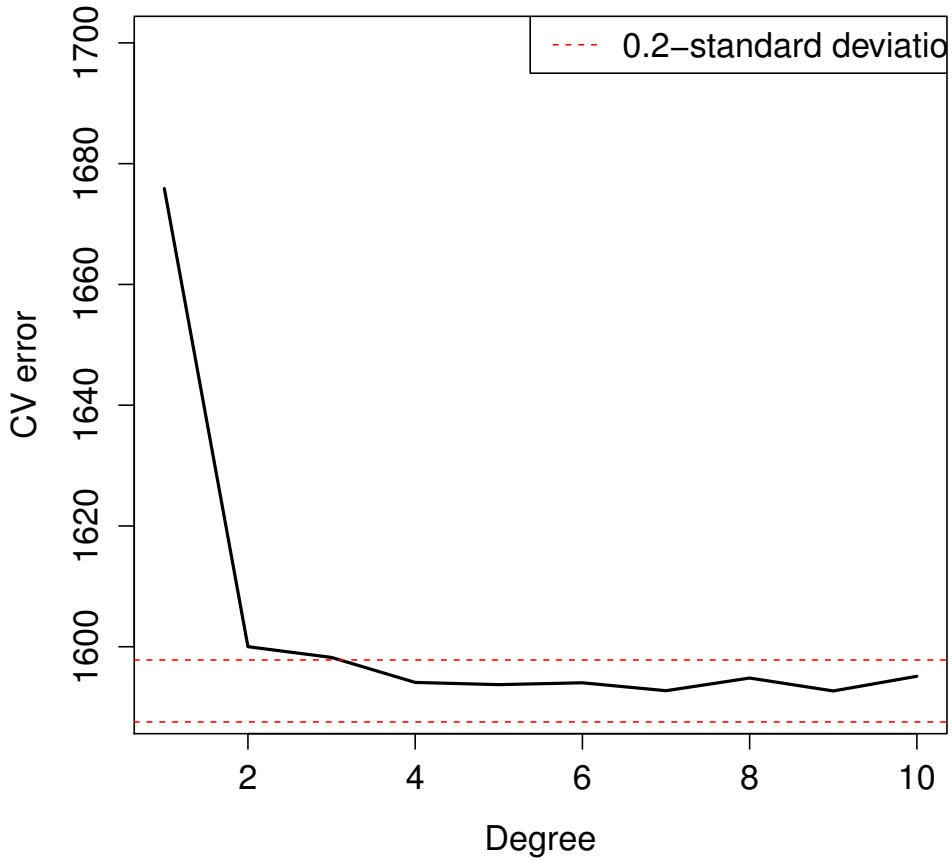



```
abline(h=min.point - 0.2 * sd.points, col="red", lty="dashed")
legend("topright", "0.2-standard deviation lines",
lty="dashed", col="red")
```

The CV plot with standard devation lines show that $d = 3$ is the smallest degree giving a reasonable small cross-validation error.

Then we find best degree using ANOVA.

```
fit.1 = lm(wage~poly(age, 1), data=Wage)
fit.2 = lm(wage~poly(age, 2), data=Wage)
fit.3 = lm(wage~poly(age, 3), data=Wage)
fit.4 = lm(wage~poly(age, 4), data=Wage)
fit.5 = lm(wage~poly(age, 5), data=Wage)
fit.6 = lm(wage~poly(age, 6), data=Wage)
fit.7 = lm(wage~poly(age, 7), data=Wage)
fit.8 = lm(wage~poly(age, 8), data=Wage)
fit.9 = lm(wage~poly(age, 9), data=Wage)
fit.10 = lm(wage~poly(age, 10), data=Wage)
anova(fit.1, fit.2, fit.3, fit.4, fit.5, fit.6, fit.7, fit.8,
```




```
fit.9, fit.10)

> anova(fit.1, fit.2, fit.3, fit.4, fit.5, fit.6, fit.7, fit.8,
+ fit.9, fit.10)
Analysis of Variance Table

Model  1: wage ~ poly(age, 1)
Model  2: wage ~ poly(age, 2)
Model  3: wage ~ poly(age, 3)
Model  4: wage ~ poly(age, 4)
Model  5: wage ~ poly(age, 5)
Model  6: wage ~ poly(age, 6)
Model  7: wage ~ poly(age, 7)
Model  8: wage ~ poly(age, 8)
Model  9: wage ~ poly(age, 9)
Model 10: wage ~ poly(age, 10)
   Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1    2998 5022216
2    2997 4793430  1    228786 143.7638 < 2.2e-16 ***
```

```
3    2996 4777674  1     15756   9.9005  0.001669 **
4    2995 4771604  1      6070   3.8143  0.050909 .
5    2994 4770322  1      1283   0.8059  0.369398
6    2993 4766389  1      3932   2.4709  0.116074
7    2992 4763834  1      2555   1.6057  0.205199
8    2991 4763707  1       127   0.0796  0.777865
9    2990 4756703  1      7004   4.4014  0.035994 *
10   2989 4756701  1         3   0.0017  0.967529
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

ANOVA shows that we should choose the polynomials with degree=3 and degree 9. Others are not significant at the significance level $\alpha = 0.05$.

```
plot(wage~age, data=Wage, col="darkgrey")
agelims = range(Wage$age)
age.grid = seq(from=agelims[1], to=agelims[2])
lm.fit = lm(wage~poly(age, 3), data=Wage)
lm.pred = predict(lm.fit, data.frame(age=age.grid))
lines(age.grid, lm.pred, col="blue", lwd=2)
```

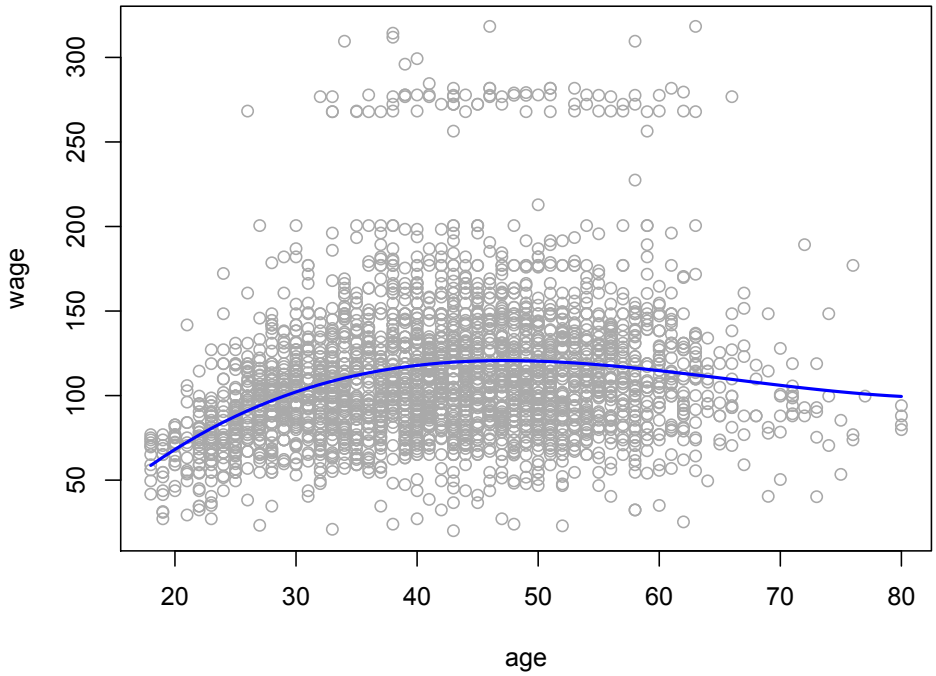
(b) (4 points) Fit a step function to predict "wage" using "age", and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.

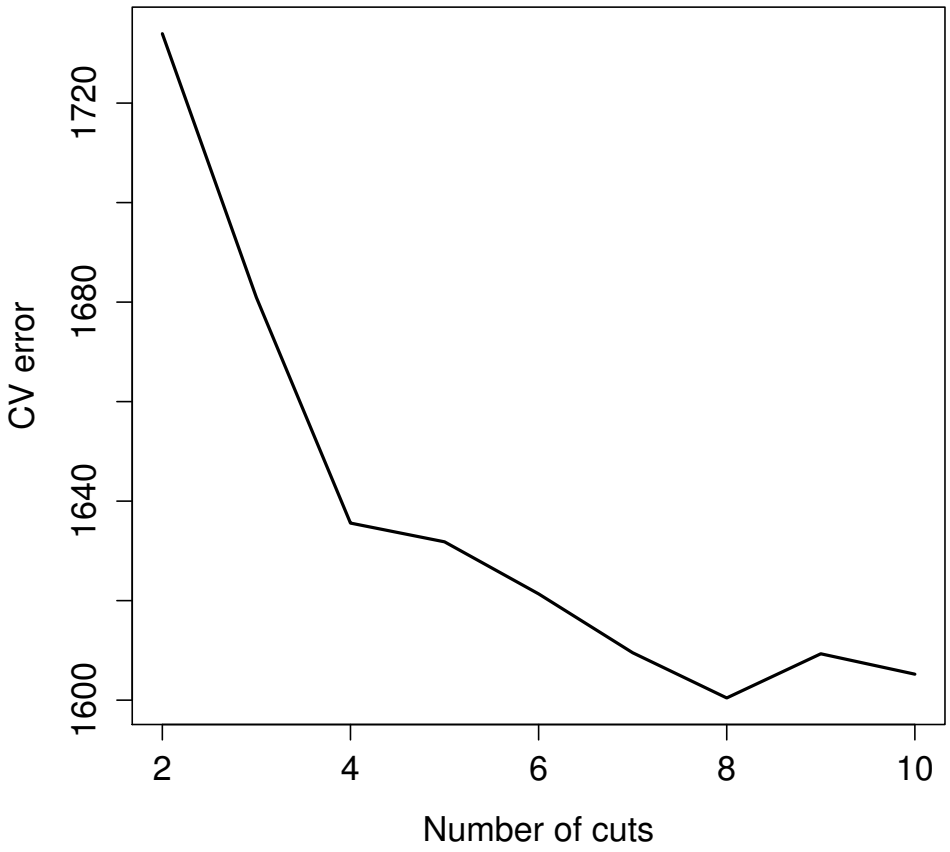
*Answer:*

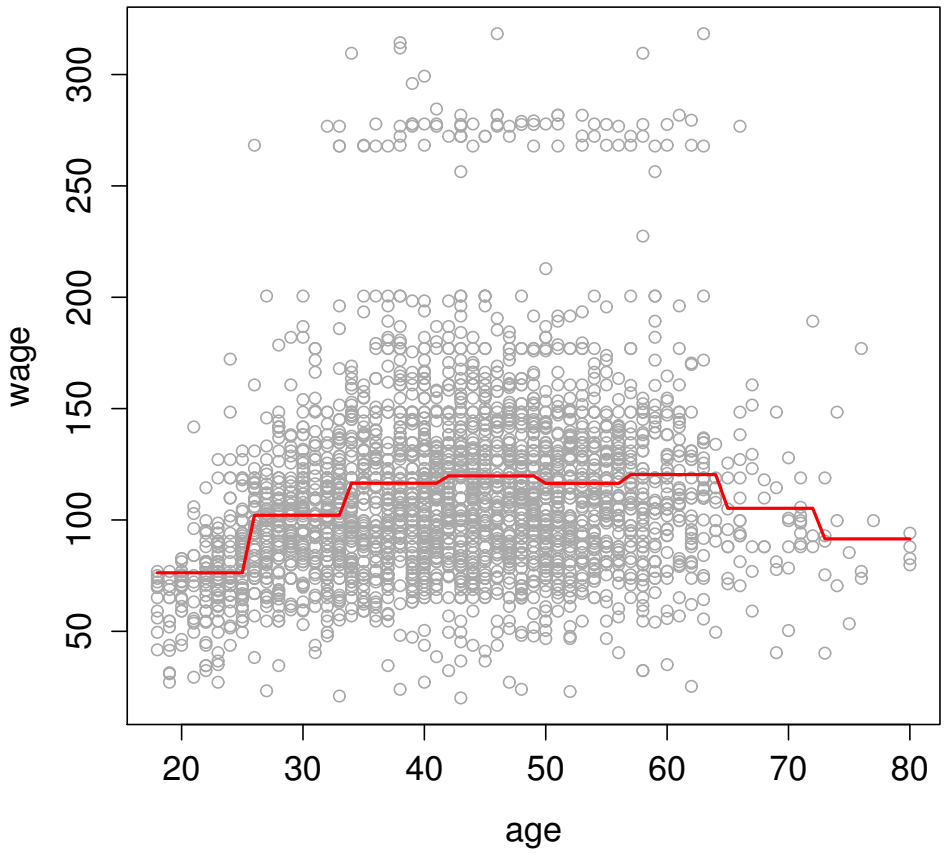
```
all.cvs = rep(NA, 10)
for (i in 2:10) {
  Wage$age.cut = cut(Wage$age, i)
  lm.fit = glm(wage~age.cut, data=Wage)
  all.cvs[i] = cv.glm(Wage, lm.fit, K=10)$delta[2]
}
plot(2:10, all.cvs[-1], xlab="Number of cuts", ylab="CV error", type="l", pch=20, lwd=2)
```

The optimal number of cut is $K = 8$.

```
lm.fit = glm(wage~cut(age, 8), data=Wage)
agelims = range(Wage$age)
age.grid = seq(from=agelims[1], to=agelims[2])
lm.pred = predict(lm.fit, data.frame(age=age.grid))
plot(wage~age, data=Wage, col="darkgrey")
lines(age.grid, lm.pred, col="red", lwd=2)
```


wage
50
100
150
200
250
300
20
30
40
50
60
70
80
age


CV error
1600
1640
1680
1720
2
4
6
8
10
Number of cuts


wage
50
100
150
200
250
300
20
30
40
50
60
70
80
age