Homework 3

Xiao-Qun Wang, Zili Ma

Q1. We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p+1 models containing 0,1,2,⋯,p predictors. Explain your answers :

a. Which of the three models with *k* predictors has the smallest training RSS ?

Best subset selection: The model with k predictors is the model with the smallest RSS among all the C_p^k models with k predictors. Forward stepwise selection: The model with k predictors is the model with the smallest RSS among the p−k models which augment the predictors in M_{k-1} with one additional predictor. Backward stepwise selection: The model with k predictors is the model with the smallest RSS among the k models which contains all but one of the predictors in M_{k+1}.So, the model with k predictors which has the smallest training RSS is the one obtained from best subset selection as it is the one selected among all k predictors models.

b. Which of the three models with *k* predictors has the smallest test RSS ?

It is hard to say, the best subset selection may have the smallest test RSS because it takes into account more models than the other methods. However, the other methods might also pick a model with smaller test RSS by sheer luck.

c. True or False :

i. The predictors in the *k*-variable model identified by forward stepwise are a subset of the predictors in the (*k*+1)-variable model identified by forward stepwise selection.

True. The model with (k+1) predictors is obtained by augmenting the predictors in the model with k predictors with one additional predictor.

ii. The predictors in the *k*-variable model identified by backward stepwise are a subset of the predictors in the (*k*+1)-variable model identified by backward stepwise selection.

True. The model with k predictors is obtained by removing one predictor from the model with (k+1) predictors.

iii. The predictors in the *k*-variable model identified by backward stepwise are a subset of the predictors in the (*k*+1)-variable model identified by forward stepwise selection.

False. There is no direct link between the models obtained from forward and backward selection.

iv. The predictors in the *k*-variable model identified by forward stepwise are a subset of the predictors in the (*k*+1)-variable model identified by backward stepwise selection.

False. There is no direct link between the models obtained from forward and backward selection.

v. The predictors in the *k*-variable model identified by best subset are a subset of the predictors in the (*k*+1)-variable model identified by best subset selection.

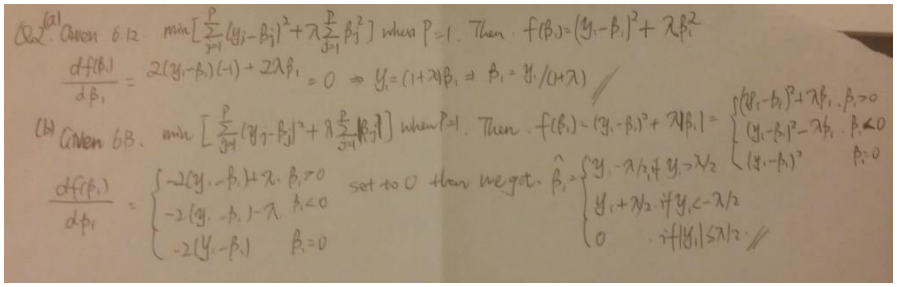
False. The model with (k+1) predictors is obtained by selecting among all possible models with (k+1) predictors, and so does not necessarily contain all the predictors selected for the k-variable model.

Q2. We will now explore (6.12) and (6.13) further.

(a) Consider (6.12) with $p = 1$. For some choice of $y1$ and $\lambda > 0$, plot (6.12) as a function of $\beta 1$. Your plot should confirm that (6.12) is solved by (6.14).

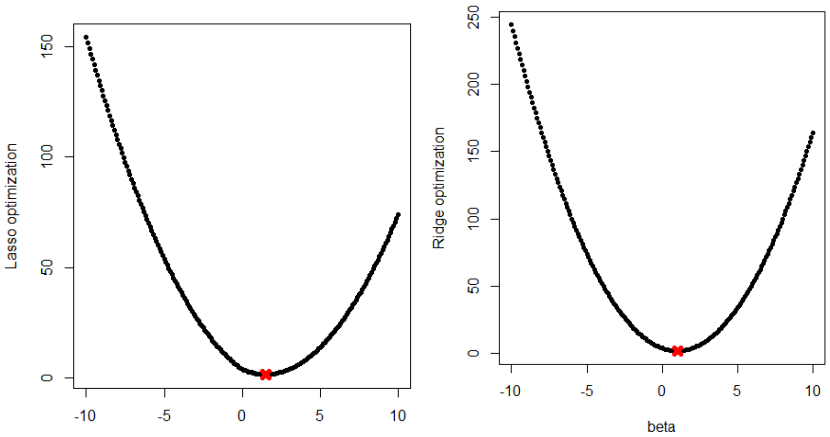
(b) Consider (6.13) with $p = 1$. For some choice of $y1$ and $\lambda > 0$, plot (6.13) as a function of $\beta 1$. Your plot should confirm that (6.13) is solved by (6.15).

Ans:



By following codes, we can verify the formulas give us the right $\beta 1$ values to optimize the method of Lasso and Ridge, respectively.

```
#####Q2#####
#(a)
y=2
lambda=1
beta=seq(-10, 10, 0.1)
plot(beta, (y-beta)^2+lambda*abs(beta), pch=20, xlab="beta", ylab ="Lasso optimization")
beta.min=y-lambda/2
points(beta.min,(y-beta.min)^2+lambda*abs(beta.min), col="red", pch=4, lwd=5)
#(b)
y=2
lambda=1
beta=seq(-10, 10, 0.1)
plot(beta,(y-beta)^2+lambda*beta^2, pch=20, xlab="beta", ylab="Ridge optimization")
beta.min=y/(1+lambda)
points(beta.min,(y-beta.min)^2+lambda*abs(beta.min), col="red", pch=4, lwd=5)
```



Q3. In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Use the *rnorm()* function to generate a predictor X of length n = 100,

as well as a noise vector of length n = 100.

```
#(a)
set.seed(1)
x=rnorm(100)
noise=rnorm(100)
```

(b) Generate a response vector Y of length n = 100 according to the model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$$

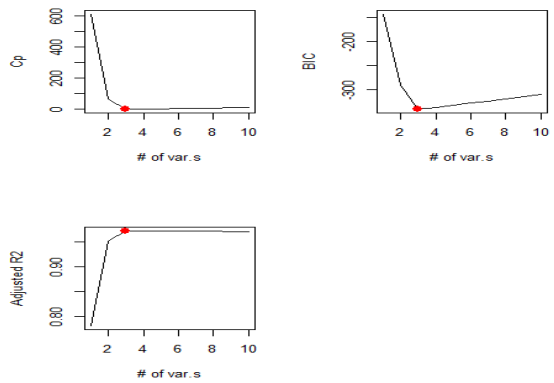
where β0, β1, β2, and β3 are constants of your choice.

```
#(b)
beta0=3
beta1=4
beta2=2
beta3=0.5
y=beta0+beta1*x+beta2*x^2+beta3*X^3+noise
```

Ans: we set beta0 = 3, beta1 = 4, beta2 = 2, beta3 = 0.5.

(c) Use the *regsubsets()* function to perform best subset selection in order to choose the best model containing the predictors X, X_2, . . . , X_{10}. What is the best model obtained according to C*p*, BIC, and adjusted R^2? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained. Note you will need to use the *data.frame()* function to create a single data set containing both X and Y .

```
library(leaps)
data.full=data.frame(y=y,x=x)
regfit.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.full,
                       nvmax=10)
reg.summary=summary(regfit.full)
par(mfrow=c(2,2))
#Cp
plot(reg.summary$cp,xlab="# of var.s", ylab="Cp",type="l")
points(which.min(reg.summary$cp),reg.summary$cp[which.min(reg.summary$cp)],
       col="red",cex=2,pch=20)
#BIC
plot(reg.summary$bic,xlab="# of var.s", ylab="BIC",type="l")
points(which.min(reg.summary$bic),reg.summary$bic[which.min(reg.summary$bic)],
       col="red",cex=2,pch=20)
#adj,R2
plot(reg.summary$adjr2,xlab="# of var.s",ylab="Adjusted R2",type="l")
points(which.max(reg.summary$adjr2),reg.summary$adjr2[which.max(reg.summary$adjr2)],
       col="red",cex=2,pch=20)
```



Ans: All the three index indicate that 3 variables (x, x^2, x^5) are the best fitting model here. The chosen best variables and their coefficients are shown as follows.

```
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.full, which.min(reg.summary$cp))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.full, which.min(reg.summary$bic))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.full, which.max(reg.summary$adjr2))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
```

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

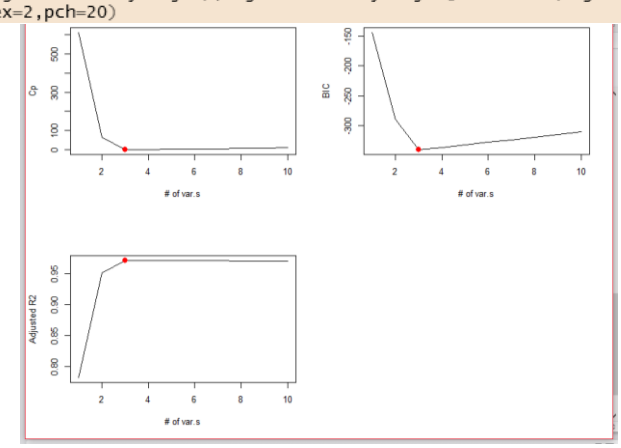
Ans:

Codes for forward and backward selection.

```
#(d)
data.full=data.frame(y=y,x=x)
regfit.fwd=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.full,method="forward",
                       nvmax=10)
regfit.bwd=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.full,method="backward",
                      nvmax=10)
reg.fwd.summary=summary(regfit.fwd)
reg.bwd.summary=summary(regfit.bwd)
```

Forward method summary:

```
#Cp
plot(reg.fwd.summary$cp,xlab="# of var.s", ylab="Cp",type="l")
points(which.min(reg.fwd.summary$cp),reg.fwd.summary$cp[which.min(reg.fwd.summary$cp)],
       col="red",cex=2,pch=20)
#BIC
plot(reg.fwd.summary$bic,xlab="# of var.s", ylab="BIC",type="l")
points(which.min(reg.fwd.summary$bic),reg.fwd.summary$bic[which.min(reg.fwd.summary$bic)],
       col="red",cex=2,pch=20)
#adj,R2
plot(reg.fwd.summary$adjr2,xlab="# of var.s",ylab="Adjusted R2",type="l")
points(which.max(reg.fwd.summary$adjr2),reg.fwd.summary$adjr2[which.max(reg.fwd.summary$adjr2)],
       col="red",cex=2,pch=20)
```

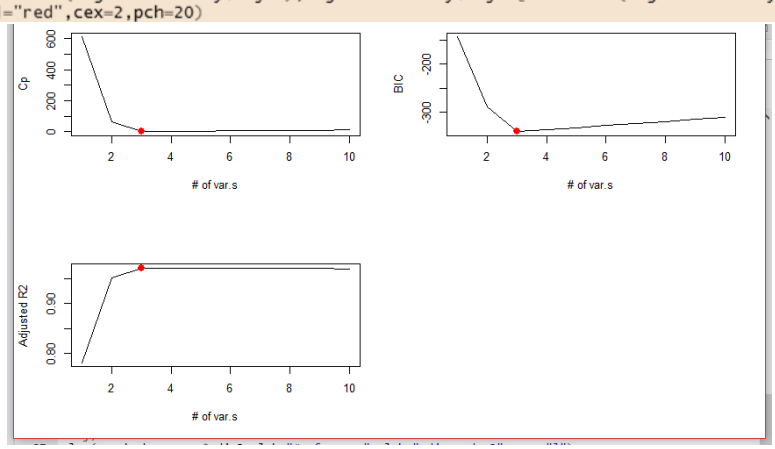


All the three index indicate that 3 variables (x, x^2, x^5) are the best fitting model here. The chosen best variables and their coefficients are shown as follows.

```
> coef(regfit.fwd, which.min(reg.fwd.summary$cp))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.fwd, which.min(reg.fwd.summary$bic))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.fwd, which.max(reg.fwd.summary$adjr2))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
```

Backward method summary:

```
par(mfrow=c(2,2))
#Cp
plot(reg.bwd.summary$cp,xlab="# of var.s", ylab="Cp",type="l")
points(which.min(reg.bwd.summary$cp),reg.bwd.summary$cp[which.min(reg.bwd.summary$cp)],
       col="red",cex=2,pch=20)
#BIC
plot(reg.bwd.summary$bic,xlab="# of var.s", ylab="BIC",type="l")
points(which.min(reg.bwd.summary$bic),reg.bwd.summary$bic[which.min(reg.bwd.summary$bic)],
       col="red",cex=2,pch=20)
#adj,R2
plot(reg.bwd.summary$adjr2,xlab="# of var.s",ylab="Adjusted R2",type="l")
points(which.max(reg.bwd.summary$adjr2),reg.bwd.summary$adjr2[which.max(reg.bwd.summary$adjr2)],
       col="red",cex=2,pch=20)
```



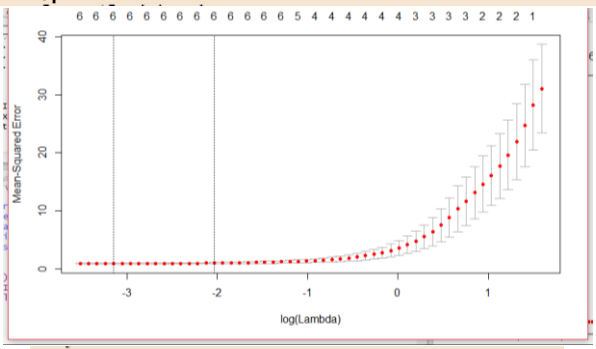
All the three index indicate that 3 variables (x, x^2, x^5) are the best fitting model here. The chosen best variables and their coefficients are shown as follows.

```
> coef(regfit.bwd, which.min(reg.bwd.summary$cp))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.bwd, which.min(reg.bwd.summary$bic))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
> coef(regfit.bwd, which.max(reg.bwd.summary$adjr2))
(Intercept)           x      I(x^2)      I(x^5)
 3.07219472  4.44514720  1.84323764  0.09022577
```

In these variable selection methods, all three methods select same variables (x, x^2, x^5).

(e) Now fit a lasso model to the simulated data, again using X, X_2, . . . , X_{10} as predictors. Use cross-validation to select the optimal value of λ. Create plots of the cross-validation error as a function of λ. Report the resulting coefficient estimates, and discuss the results obtained.

```
library(glmnet)
xmat=model.matrix(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.full)[,-1]
cv.lasso=cv.glmnet(xmat,y,alpha=1)
plot(cv.lasso)
```




```
> bestlamda=cv.lasso$lambda.min
> bestlamda
[1] 0.04285772
```

Ans: Our model's best lamda value is 0.0429 according to the cv value. By this value we can have the model with coefficients as follows. LASSO selects 6 variables ($x, x^2, x^3, x^4, x^5, x^7$).

```
> fit.lasso=glmnet(xmat,y,alpha=1)
> predict(fit.lasso,s=bestlamda,type="coefficients")[1:11,]
(Intercept)           x      I(x^2)      I(x^3)      I(x^4)      I(x^5)      I(x^6)      I(x^7)
3.154924881 4.204158225 1.660789201 0.289047849 0.037992565 0.007095140 0.000000000 0.005782064
     I(x^8)      I(x^9)     I(x^10)
0.000000000 0.000000000 0.000000000
```

(f) Now generate a response vector *Y* according to the model

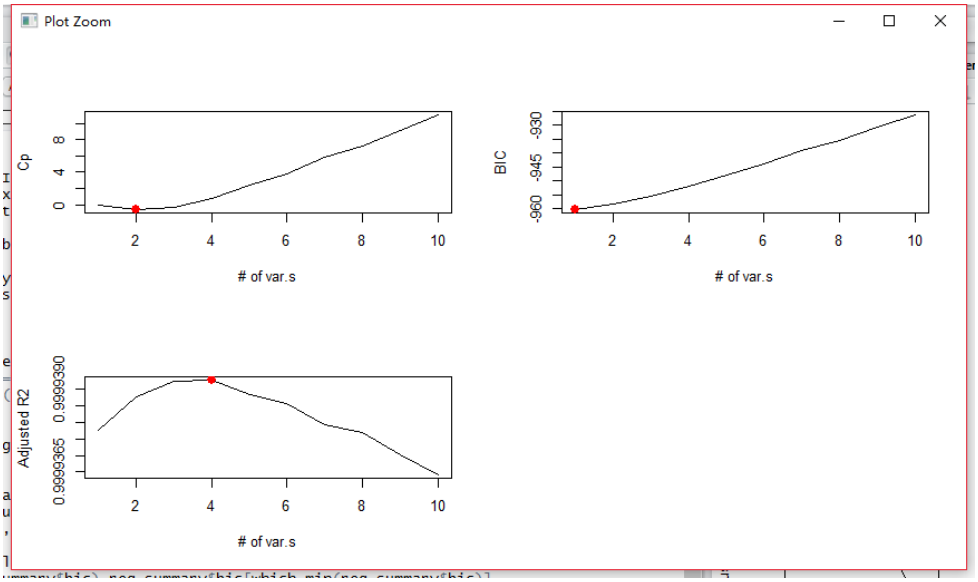
$$Y = \beta_0 + \beta_7 X^7 + \varepsilon,$$

and perform best subset selection and the lasso. Discuss the results obtained.

Ans:

Best subset:

```
#(f)
beta7 = 2
y=beta0+beta7*x^7+noise
data.full=data.frame(y=y,x=x)
regfit.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.full,
                       nvmax=10)
reg.summary=summary(regfit.full)
par(mfrow=c(2,2))
#Cp
plot(reg.summary$cp,xlab="# of var.s", ylab="Cp",type="l")
points(which.min(reg.summary$cp),reg.summary$cp[which.min(reg.summary$cp)],
       col="red",cex=2,pch=20)
#BIC
plot(reg.summary$bic,xlab="# of var.s", ylab="BIC",type="l")
points(which.min(reg.summary$bic),reg.summary$bic[which.min(reg.summary$bic)],
       col="red",cex=2,pch=20)
#adj,R2
plot(reg.summary$adjr2,xlab="# of var.s",ylab="Adjusted R2",type="l")
points(which.max(reg.summary$adjr2),reg.summary$adjr2[which.max(reg.summary$adjr2)],
       col="red",cex=2,pch=20)
```



From best subsets method, three index do not give us the same number of variables. Cp implies 2 variables; BIC gives 1 variable, adjusted R^2 uses 4 variables. The detail is shown below.

```
> coef(regfit.full, 1)
(Intercept)      I(x^7)
    2.95894     2.00077
> coef(regfit.full, 2)
(Intercept)      I(x^2)      I(x^7)
  3.0704904  -0.1417084   2.0015552
> coef(regfit.full, 4)
(Intercept)           x      I(x^2)      I(x^3)      I(x^7)
  3.0762524   0.2914016  -0.1617671  -0.2526527   2.0091338
```

LASSO:

```
xmat=model.matrix(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.full)[,-1]
cv.lasso=cv.glmnet(xmat,y,alpha=1)
bestlambda=cv.lasso$lambda.min
bestlambda
```

```
> bestlambda
[1] 3.879577
> fit.lasso=glmnet(xmat,y,alpha=1)
> predict(fit.lasso,s=bestlambda,type="coefficients")[1:11,]
(Intercept)           x      I(x^2)      I(x^3)      I(x^4)      I(x^5)      I(x^6)      I(x^7)
   3.229085    0.000000    0.000000    0.000000    0.000000    0.000000    0.000000    1.936760
     I(x^8)      I(x^9)     I(x^10)
   0.000000    0.000000    0.000000
```

By the method of LASSO, we achieve the best model matching to our simulated question. In this case, LASSO performed better than best subset method.

Q4 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 and a test set.

```
> library(ISLR)
> data(College)
> set.seed(11)
> train = sample(1:dim(College)[1], dim(College)[1] / 2)
> test <- -train
> College.train <- College[train, ]
> College.test <- College[test, ]
```

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

```
> fit.lm <- lm(Apps ~ ., data = College.train)
> pred.lm <- predict(fit.lm, College.test)
> mean((pred.lm - College.test$Apps)^2)
[1] 1538442
```

The test MSE is 1.538442*10^6

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

```
> train.mat <- model.matrix(Apps ~ ., data = College.train)
> test.mat <- model.matrix(Apps ~ ., data = College.test)
> grid <- 10 ^ seq(4, -2, length = 100)
> fit.ridge <- glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)
> cv.ridge <- cv.glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)
> bestlam.ridge <- cv.ridge$lambda.min
> bestlam.ridge
[1] 18.73817
> pred.ridge <- predict(fit.ridge, s = bestlam.ridge, newx = test.mat)
> mean((pred.ridge - College.test$Apps)^2)
[1] 1608859
```

The test MSE is 1.608859*10^6

d. Fit a lasso model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates.

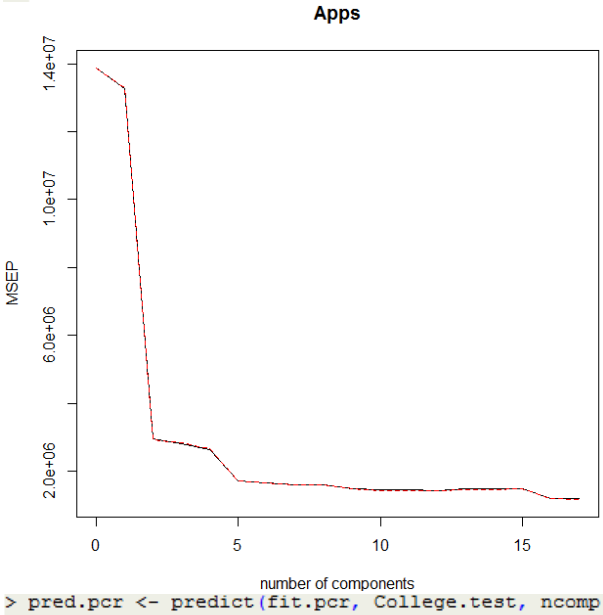
```
> fit.lasso <- glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
> cv.lasso <- cv.glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
> bestlam.lasso <- cv.lasso$lambda.min
> bestlam.lasso
[1] 21.54435
> pred.lasso <- predict(fit.lasso, s = bestlam.lasso, newx = test.mat)
> mean((pred.lasso - College.test$Apps)^2)
[1] 1635280
```

The test MSE is also higher for ridge regression than for least squares.

```
> predict(fit.lasso, s = bestlam.lasso, type = "coefficients")
19 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept) -836.50402310
(Intercept)    .
PrivateYes  -385.73749394
Accept         1.17935134
Enroll         .
Top10perc     22.70211938
Top25perc      .
F.Undergrad    0.07062149
P.Undergrad    0.01366763
Outstate      -0.03424677
Room.Board     0.01281659
Books         -0.02167770
Personal       .
PhD           -1.46396964
Terminal      -5.17281004
S.F.Ratio      5.70969524
perc.alumni   -9.95007567
Expend         0.14852541
Grad.Rate      5.79789861
```

e. 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.

```
> fit.pcr <- pcr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV")
> validationplot(fit.pcr, val.type = "MSEP")
```

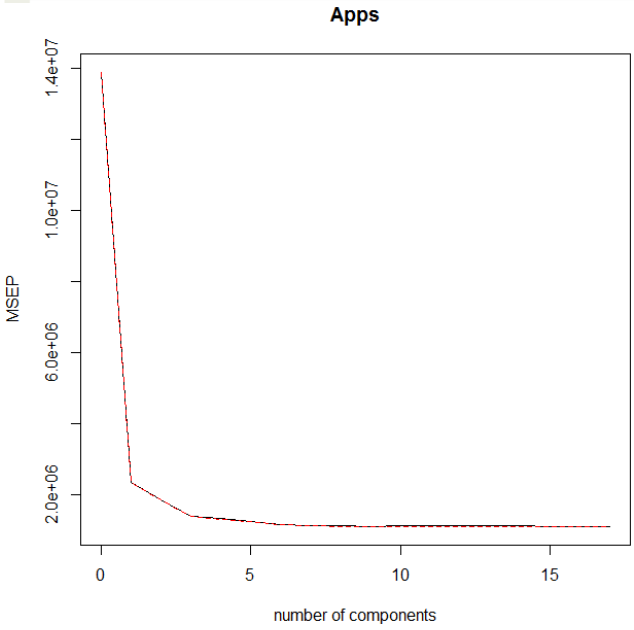



```
> pred.pcr <- predict(fit.pcr, College.test, ncomp = 10)
> mean((pred.pcr - College.test$Apps)^2)
[1] 3014496
```

The test MSE is also higher for PCR than for least squares.

f. 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.

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



```
> pred.pls <- predict(fit.pls, College.test, ncomp = 10)
> mean((pred.pls - College.test$Apps)^2)
[1] 1508987
```

The test MSE is lower for PLS than for least squares.

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 ?

We compute the test *R^2* for all models.

```
> test.avg <- mean(College.test$Apps)
> lm.r2 <- 1 - mean((pred.lm - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
> lm.r2
[1] 0.9044281
> ridge.r2 <- 1 - mean((pred.ridge - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
> ridge.r2
[1] 0.900641
> lasso.r2 <- 1 - mean((pred.lasso - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
> lasso.r2
[1] 0.8989591
> pcr.r2 <- 1 - mean((pred.pcr - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
> pcr.r2
[1] 0.8127319
> pls.r2 <- 1 - mean((pred.pls - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
> pls.r2
[1] 0.9062579
```

As shown above, the test R2 for least squares is 0.9044281, the test R2 for ridge is 0.9000641, the test R2 for lasso is 0.8989591, the test R2 for pcr is 0.8127319 and the test R2 for pls is 0.9062579. All models, except PCR, predict college applications with high accuracy.