DNSC 6279 Section 11 Spring 2020 Homework 3

Ziwei Li

Remark. This homework aims to help you further understand the model selection techniques in linear model. Credits for **Theoretical Part** and **Computational Part** are in total 100 pt. For **Computational Part** , please complete your answer in the **RMarkdown** file and summit your printed PDF homework created by it.

Computational Part

Hint. Before starting your work, carefully read Textbook Chapter 6.5-6.7 (Lab 1-3). Mimic the related analyses you learn from it. Related packages have been loaded in setup.

1. (Model Selection, Textbook 6.8, *25 pt*) In this exercise, we will generate simulated data, and will then use this data to perform model selection.

    a. Use the `rnorm` function to generate a predictor \bm$X$ of length $n = 100$, as well as a noise vector \bm$\epsilon$ of length $n = 100$.

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

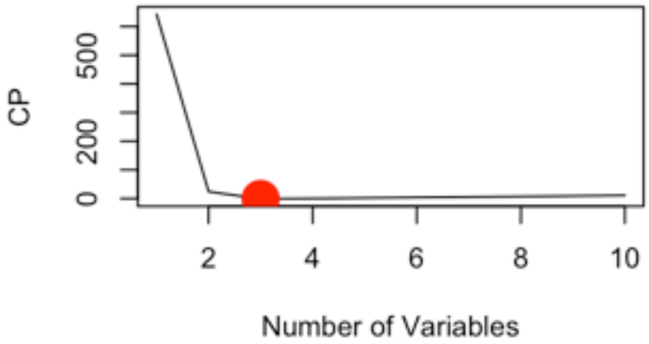
```
(b) Generate a response vector $\bm{Y}$ of length $n = 100$ according to the model $$
Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon, $$ where $\beta_0 = 3
$, $\beta_1 = 2$, $\beta_2 = -3$, $\beta_3 = 0.3$.
```

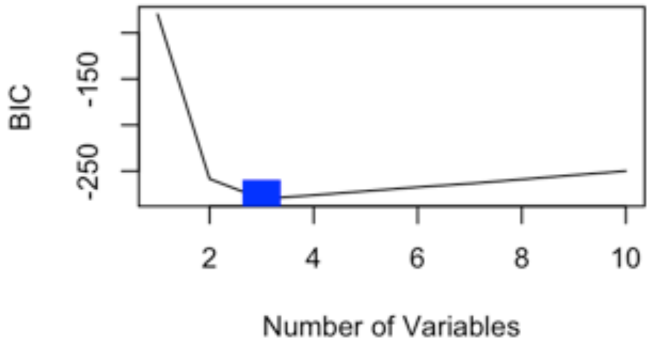
```
## set the coefficients and formulate the true model
b0 = 3
b1 = 2
b2 = -3
b3 = 0.3
y = b0 + b1 * x + b2 * x^2 + b3 * x^3 + epsilon
```

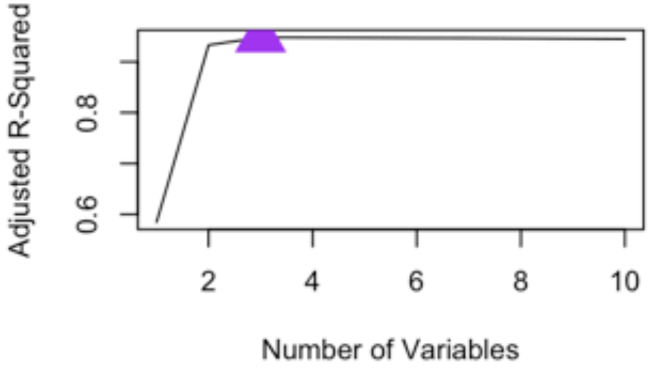
```
(c) Use the `regsubsets` function from `leaps` package to perform best subset selecti
on in order to choose the best model containing the predictors $(X, X^2, \cdots, X^{1
0})$. What is the best model obtained according to $C_p$, BIC, and adjusted $R^2$? Sh
ow some plots to provide evidence for your answer, and report the coefficients of the
best model obtained.
```

```r
## include variables from x to x^10, and save them into a dataframe
full = data.frame(y=y,x=x,x2=x^2,x3=x^3,x4=x^4,x5=x^5,x6=x^6,x7=x^7,x8=x^8,x9=x^9,x10=x^10)
## perform best subset selection using regsubsets()
regfit1 = regsubsets(y ~ .,data = full,nvmax=10)
regfit1.sum=summary(regfit1)
```

```r
par(mfrow=c(2,2))
## plot of cp
plot(regfit1.sum$cp,xlab='Number of Variables',ylab='CP',type="l")
points(which.min(regfit1.sum$cp),regfit1.sum$cp[which.min(regfit1.sum$cp)],col='red',cex=3,pch=16)
## plot of BIC
plot(regfit1.sum$bic,xlab='Number of Variables',ylab='BIC',type="l")
points(which.min(regfit1.sum$bic),regfit1.sum$bic[which.min(regfit1.sum$bic)],col='Blue',cex=3,pch=15)
## plot of Adjusted R-squared
plot(regfit1.sum$adjr2,xlab='Number of Variables',ylab='Adjusted R-Squared',type="l")
points(which.max(regfit1.sum$adjr2),regfit1.sum$adjr2[which.max(regfit1.sum$adjr2)],col='Purple',cex=3,pch=17)

## We may notice that when the number of variables = 3, the CP and BIC is the smallest and Adjusted R-squared is the biggest. Therefore, the best model is when number of variables = 3.
```







```
## Best model's coefficient
coef(regfit1,3)
```

```
## (Intercept)           x          x2          x7
##  3.07627412  2.35623596 -3.16514887  0.01046843
```

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

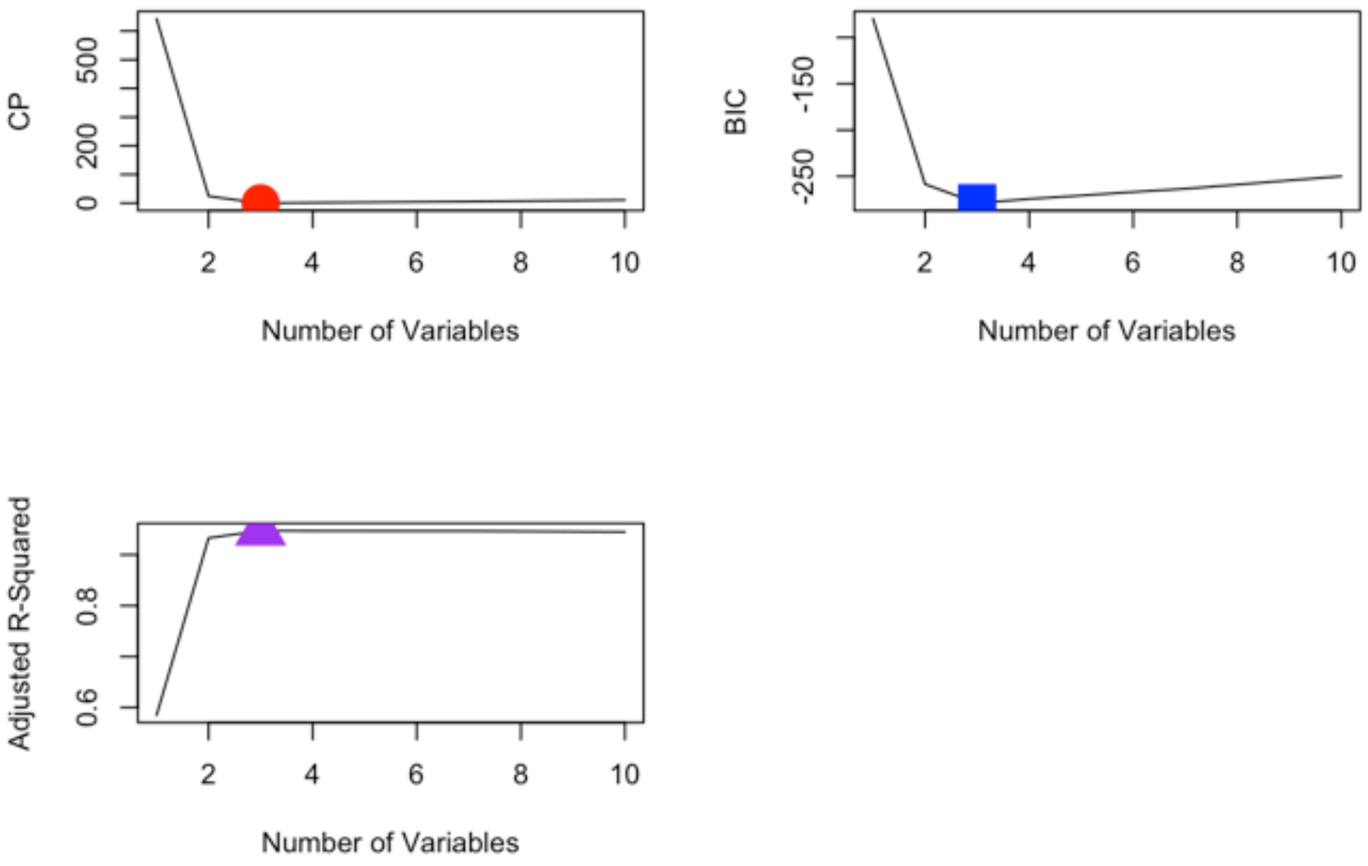
```
regfit.back = regsubsets(y~.,data=full,nvmax=10,method='backward')
back1 = summary(regfit.back)
coef(regfit.back,3)
```

```
##  (Intercept)            x           x2            x9
##  3.078881355  2.419817953 -3.177235617  0.001870457
```

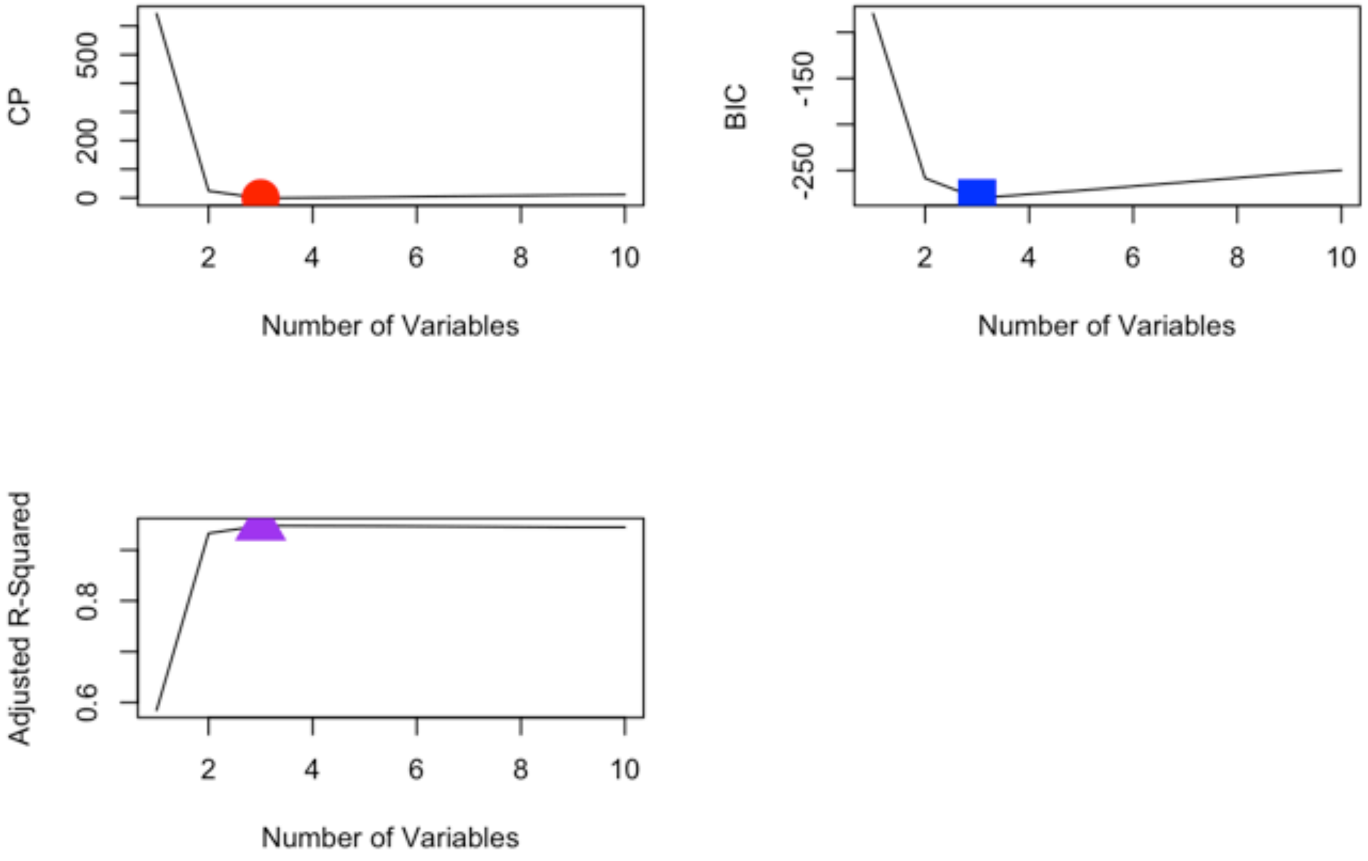
```
regfit.for = regsubsets(y~.,data=full,nvmax=10,method='forward')
for1 = summary(regfit.for)
coef(regfit.for,3)
```

```
## (Intercept)           x          x2          x7
##  3.07627412  2.35623596 -3.16514887  0.01046843
```

```
# plot for backward selection
par(mfrow=c(2,2))
## plot of cp
plot(back1$cp,xlab='Number of Variables',ylab='CP',type="l")
points(which.min(back1$cp),back1$cp[which.min(back1$cp)],col='red',cex=3,pch=16)
## plot of BIC
plot(back1$bic,xlab='Number of Variables',ylab='BIC',type="l")
points(which.min(back1$bic),back1$bic[which.min(back1$bic)],col='Blue',cex=3,pch=15)
## plot of Adjusted R-squared
plot(back1$adjr2,xlab='Number of Variables',ylab='Adjusted R-Squared',type="l")
points(which.max(back1$adjr2),back1$adjr2[which.max(back1$adjr2)],col='Purple',cex=3,
pch=17)
```



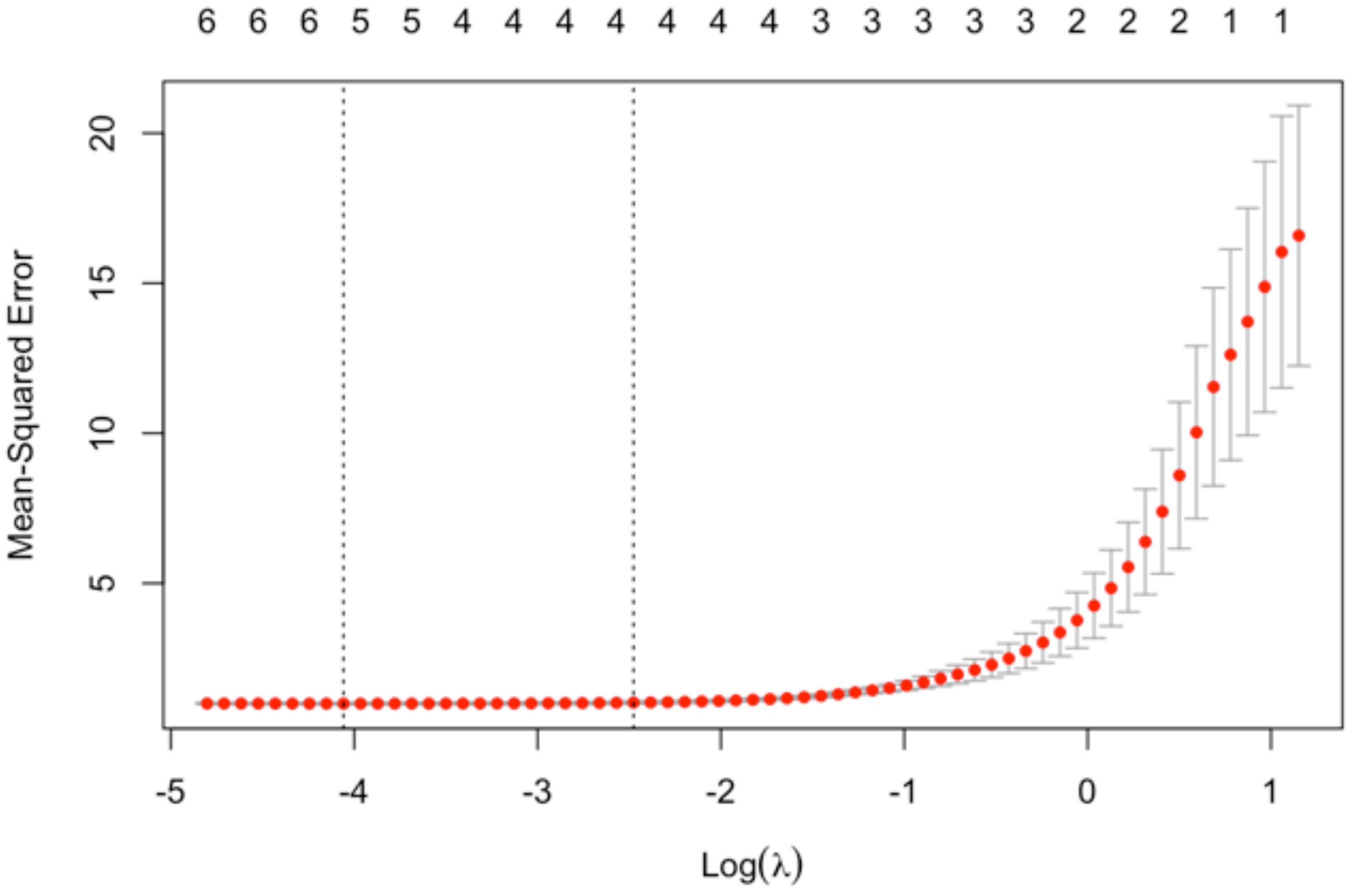
```
#plot of forward selection
par(mfrow=c(2,2))
## plot of cp
plot(for1$cp,xlab='Number of Variables',ylab='CP',type="l")
points(which.min(for1$cp),for1$cp[which.min(for1$cp)],col='red',cex=3,pch=16)
## plot of BIC
plot(for1$bic,xlab='Number of Variables',ylab='BIC',type="l")
points(which.min(for1$bic),for1$bic[which.min(for1$bic)],col='Blue',cex=3,pch=15)
## plot of Adjusted R-squared
plot(for1$adjr2,xlab='Number of Variables',ylab='Adjusted R-Squared',type="l")
points(which.max(for1$adjr2),for1$adjr2[which.max(for1$adjr2)],col='Purple',cex=3,pch
=17)
```



```
# When using the backward forward selection, the number of true variables are all 3,
which is equal to the best subset selection.
# While in best-subset-selection and forward-selection, the best variables are x1, x2
, and x7. However, the best variables using backward selection are x1,x2 and x9.
```

```
(e) Now fit a LASSO model with `glmnet` function from `glmnet` package to the simulat
ed data, again using $(X,X^2,\cdots,X^{10})$ as predictors. Use cross-validation to s
elect 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 re
sults obtained.
```

```
#set x and y
y.l = as.matrix(full[,1])
x.l = as.matrix(full[,2:11])

#use glmnet to fit the lasso regression and use 5-fold cross-validation
cv.l <- cv.glmnet(x.l,y.l,alpha=1,nfolds=5)

# Plot the result
plot(cv.l)
```



```
cat("Lambda with smallest CV Error", cv.l$lambda[which.min(cv.l$cvm)],fill=TRUE)
```

```
## Lambda with smallest CV Error 0.01727789
```

```
cat("Coefficients", as.numeric(coef(cv.l)),fill=TRUE)
```

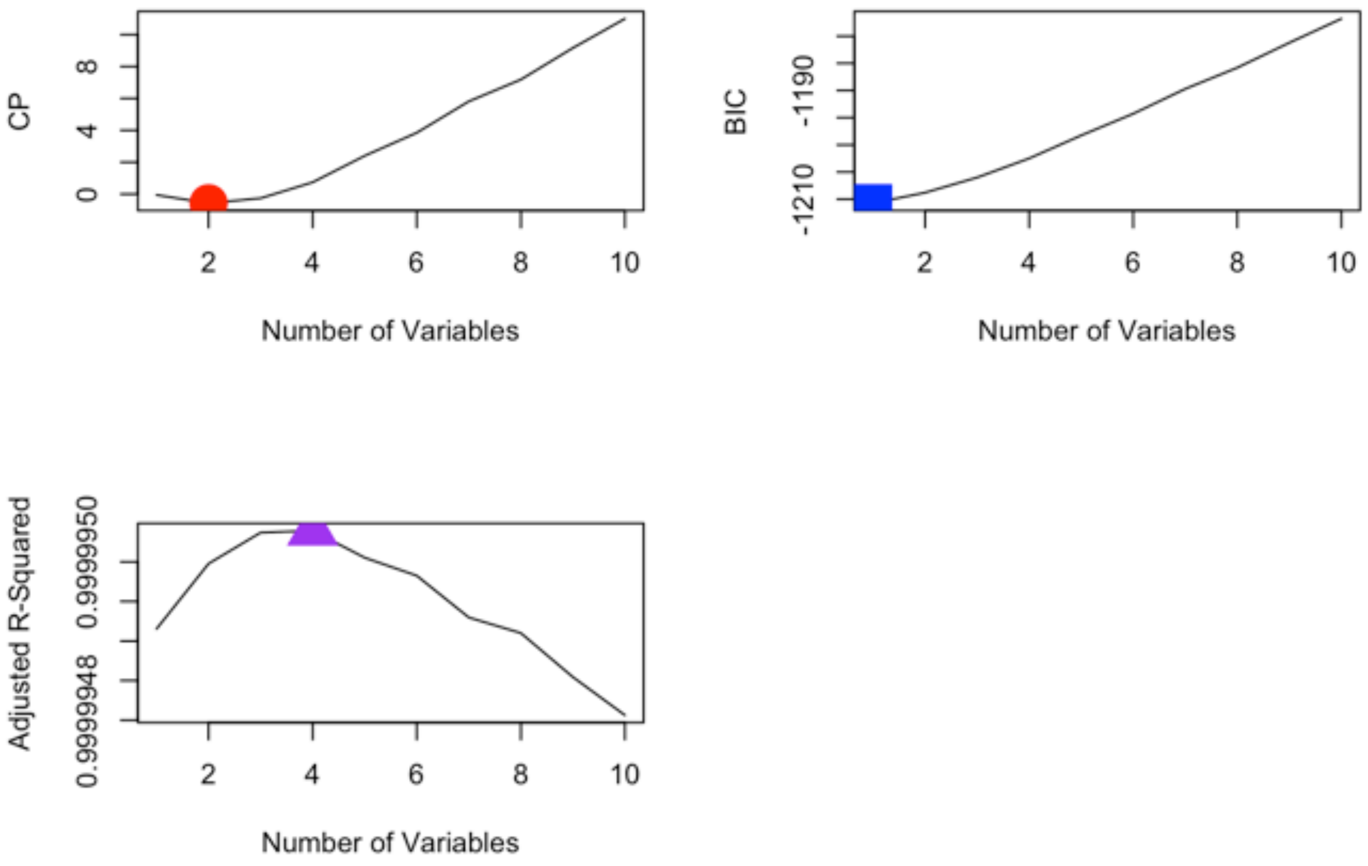
```
## Coefficients 3.00316 2.179634 -3.049368 0.01947017 0 0.04709426 0 0 0 0 0
```

```
# After using lasso regression, we can get X, X^2, X^3, and X^5 as variables for this
model.
```

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

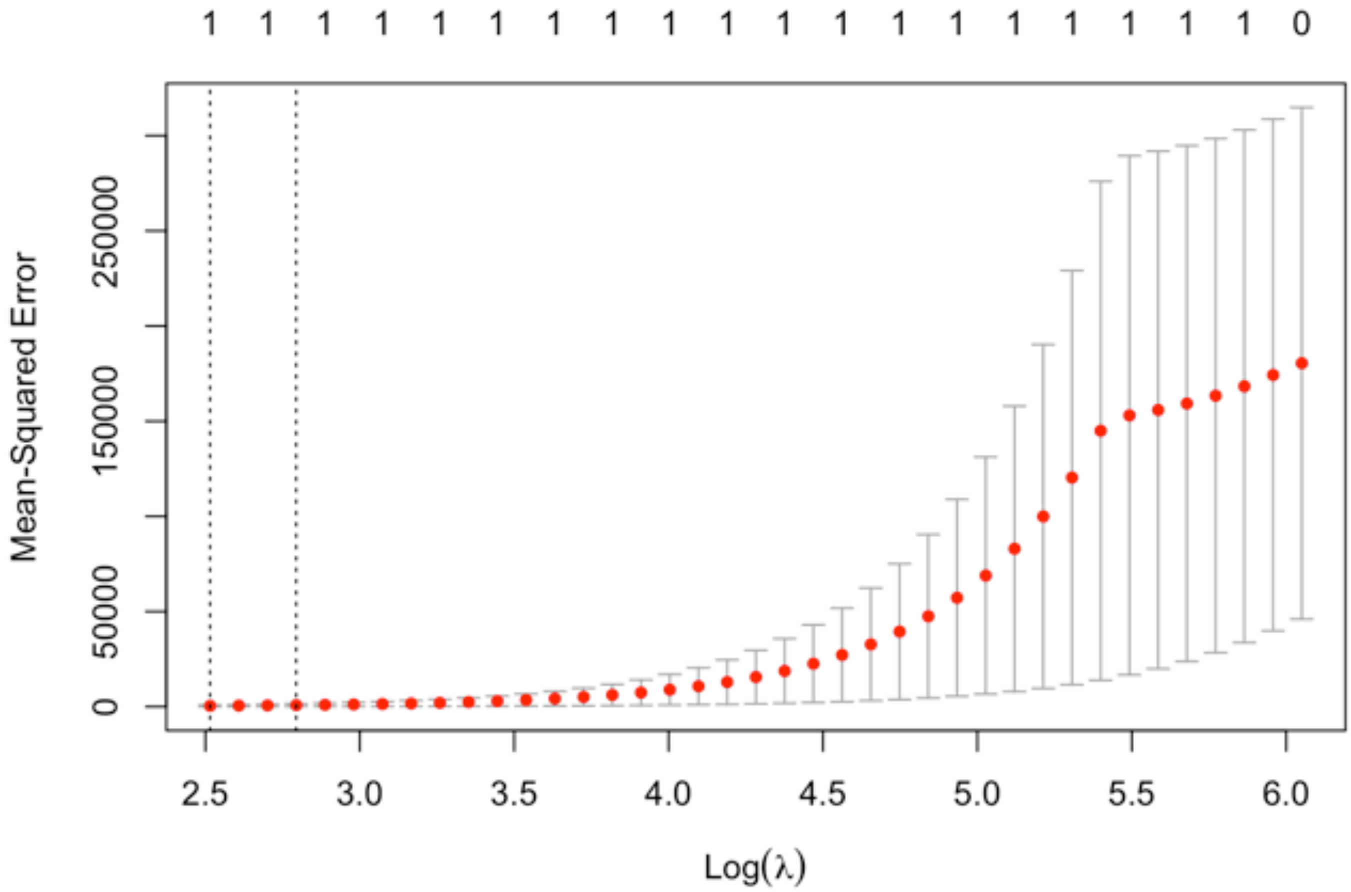
```
b7 = 7
y2 = b0 + b7 * x^7 + epsilon
full2 = data.frame(y2=y2,x=x,x2=x^2,x3=x^3,x4=x^4,x5=x^5,x6=x^6,x7=x^7,x8=x^8,x9=x^9,
x10=x^10)
```

```
#Best-subset-selection
regfit2 = regsubsets(y2 ~ .,data = full2,nvmax=10)
regfit2.sum=summary(regfit2)

# plot the best-subset-selection
par(mfrow=c(2,2))
## plot of cp
plot(regfit2.sum$cp,xlab='Number of Variables',ylab='CP',type="l")
points(which.min(regfit2.sum$cp),regfit2.sum$cp[which.min(regfit2.sum$cp)],col='red',
cex=3,pch=16)
## plot of BIC
plot(regfit2.sum$bic,xlab='Number of Variables',ylab='BIC',type="l")
points(which.min(regfit2.sum$bic),regfit2.sum$bic[which.min(regfit2.sum$bic)],col='Bl
ue',cex=3,pch=15)
## plot of Adjusted R-squared
plot(regfit2.sum$adjr2,xlab='Number of Variables',ylab='Adjusted R-Squared',type="l")
points(which.max(regfit2.sum$adjr2),regfit2.sum$adjr2[which.max(regfit2.sum$adjr2)],c
ol='Purple',cex=3,pch=17)
```



```
# When use cp to choose the best model,, the result is when number of variables are 2 .

# When use BIC to choose the best model, the result is when number of variables are 1 .

# When use Adjusted R_Squared to choose the best model, the result is when number of variables are 4.
```

```
#set x and y
y.l2 = as.matrix(full2[,1])
x.l2 = as.matrix(full2[,2:11])

#use glmnet to fit the lasso regression and use 5-fold cross-validation
cv.l2 <- cv.glmnet(x.l2,y.l2,alpha=1,nfolds=5)

# Plot the result
plot(cv.l2)
```



```
min(cv.l2$lambda)
```

```
## [1] 12.36884
```

```
print(cv.l2)
```

```
##
## Call:  cv.glmnet(x = x.l2, y = y.l2, nfolds = 5, alpha = 1)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure    SE Nonzero
## min  12.37   458.8 414.9       1
## 1se  16.35   798.5 722.8       1
```

```
#When using Lasso, we can get the smallest lambda is 12.37, and the number of non-zer
o parameters is 1.
```

2. (Prediction, Textbook 6.9, *25 pt*) In this exercise, we will predict the number of applications received using the other variables in the `College` data set from `ISLR` package.

    a. Randomly split the data set into a training set and a test set (1:1).

```
set.seed(1)
#College2 = College
train=sample(nrow(College),size=0.5*nrow(College)) #select training and test data
test=-(train) #select test data
collegetrain = College[train,]
collegetest = College[test,]
```

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

```
lm1 = lm(Apps~.,data=collegetrain)
value = collegetest[,-2]
pred = predict(lm1,value)
test_error = mean((pred-collegetest$Apps)^2)
test_error
```

```
## [1] 1135758
```

```
(c) Fit a ridge regression model on the training set, with $\lambda$ chosen by 5-fold cross-validation. Report the test error obtained.
```

```
# Transfer the dataset into matrix.
x.train=model.matrix(Apps~.,collegetrain)[,-2] #put regressors from training set into a matrix
y.train=collegetrain$Apps #label for training set
x.test=model.matrix(Apps~.,collegetest)[,-2] #put regressors from test set into a matrix
y.test=collegetest$Apps
```

```
# Perform ridge regression
ridge.mod = glmnet(x.train,y.train,alpha=0)
cv.ridge = cv.glmnet(x.train,y.train,alpha=0,nfolds = 5)
bestlam_r = cv.ridge$lambda.min

ridge.pred = predict(ridge.mod,s=bestlam_r,newx=x.test)
ridge.err = mean((ridge.pred-y.test)^2)
ridge.err
```

```
## [1] 1007688
```

```
bestlam_r
```

```
## [1] 405.8404
```

```
(d) Fit a LASSO model on the training set, with $\lambda$ chosen by 5-fold cross-vali
dation. Report the test error obtained, along with the number of non-zero coefficient
estimates.
```

```
lasso.mod = glmnet(x.train,y.train,alpha=1)
cv.lasso = cv.glmnet(x.train,y.train,alpha=1,nfolds = 5)
bestlam_l = cv.lasso$lambda.min

lasso.pred = predict(lasso.mod,s=bestlam_l,newx=x.test)
lasso.err = mean((lasso.pred-y.test)^2)
lasso.err
```

```
## [1] 1140473
```

```
bestlam_l
```

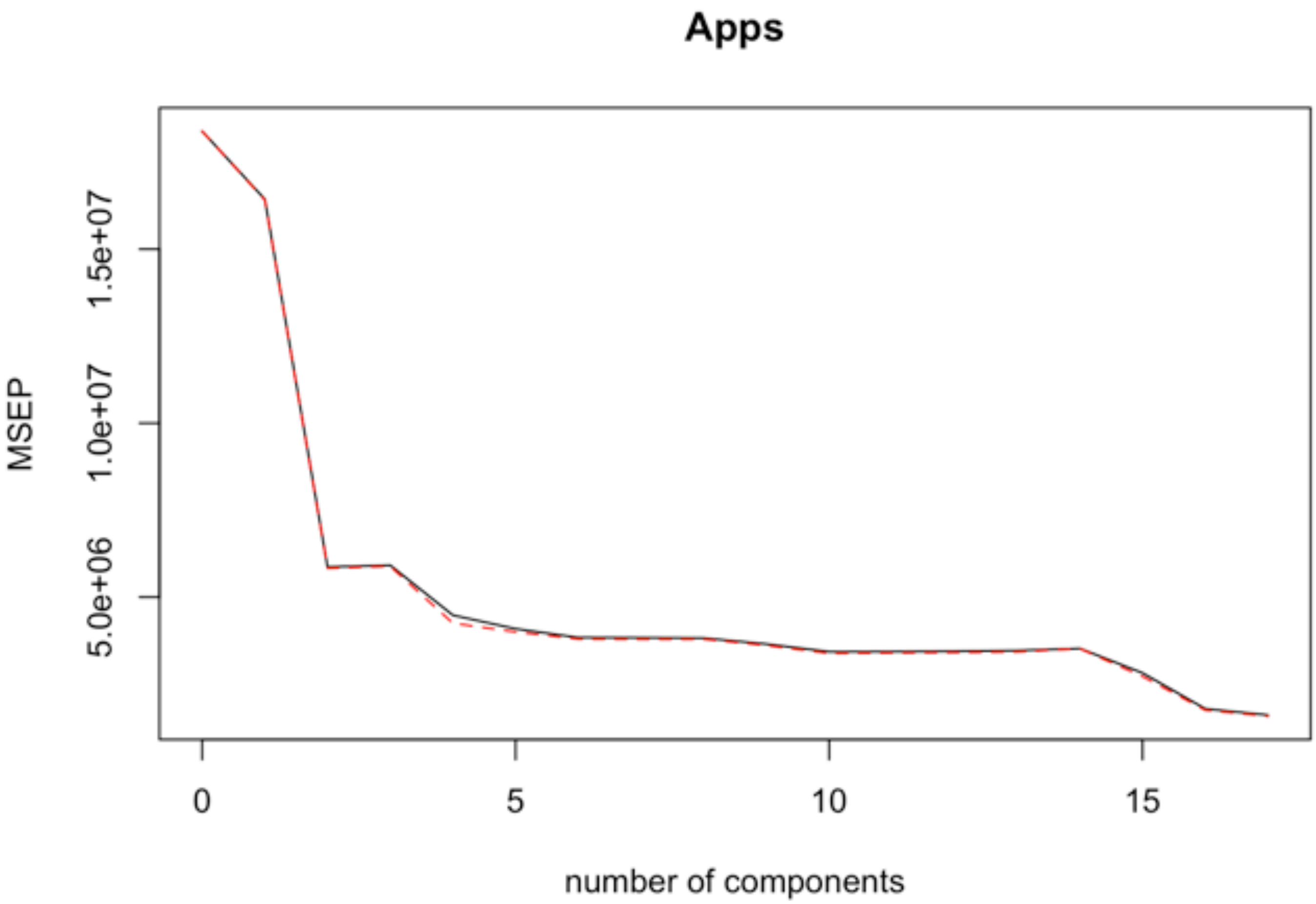
```
## [1] 2.165848
```

```
(e) Fit a PCR model on the training set, with $M$ chosen by 5-fold cross-validation.
Report the test error obtained, along with the value of $M$ selected by cross-validat
ion.
```

```
pcr.fit=pcr(Apps~.,data=collegetrain, scale=TRUE, validation="CV",nfold=5)
summary(pcr.fit)
```

```
## Data:    X dimension: 388 17
##  Y dimension: 388 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            4288     4054     2422     2432     2117     2022     1959
## adjCV         4288     4051     2415     2426     2061     1999     1948
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1957     1955     1911      1852      1853      1856      1861
## adjCV     1947     1947     1900      1840      1843      1846      1851
##        14 comps  15 comps  16 comps  17 comps
## CV         1877      1679      1338      1269
## adjCV      1876      1649      1323      1256
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       32.20    57.78    65.31    70.99    76.37    81.27     84.8
## Apps    13.44    70.93    71.07    79.87    81.15    82.25     82.3
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.85    90.62     92.91     94.98     96.74     97.79     98.72
## Apps    82.33    83.38     84.76     84.80     84.84     85.11     85.14
##       15 comps  16 comps  17 comps
## X        99.42     99.88    100.00
## Apps     90.55     93.42     93.89
```

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

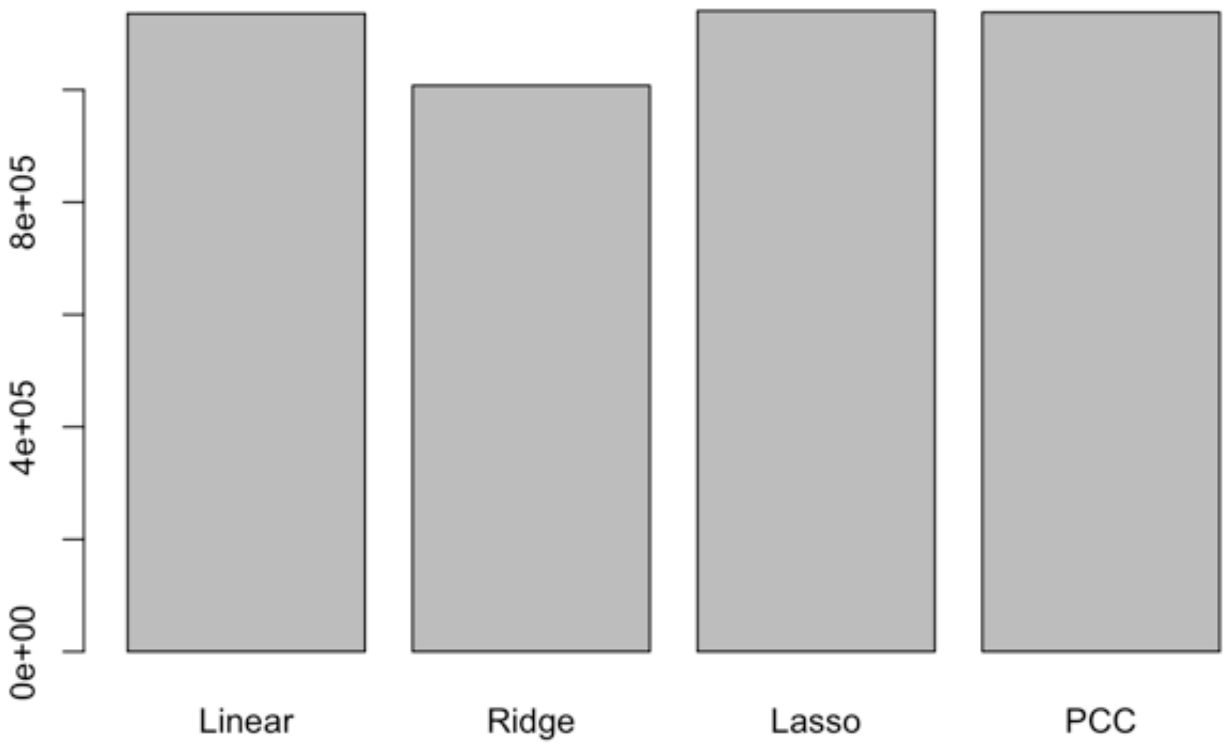


```
pcr.pred=predict(pcr.fit,collegetest,ncomp=16)
pcc.err = mean((collegetest$Apps-pcr.pred)^2)
pcc.err
```

```
## [1] 1137877
```

```
(f) Comment on the results obtained. How accurately can we predict the number of coll
ege applications received? Is there much difference among the test errors resulting f
rom these four approaches?
```

```
error = c(test_error,ridge.err,lasso.err,pcc.err)
names(error)=c("Linear","Ridge","Lasso","PCC")
barplot(error)
```



```
# We can see from the plot that all the method's tet error is almost the same. Howeve
r,the lasso and ridge may perform better than the other methods.
```

Theoretical Part

1.

(1) For Ridge Regression:

$$||y - X\beta||^2 + \lambda \sum_{k=1}^{p} {\beta_k}^2$$

When λ increases from 0:

(i) training RSS: steadily increase. Because as λ gets larger, the coefficients become shrunken towards zero. We will have less variables to fit the data. Therefore, the training RSS will steadily increase.

(ii): test RSS: Increase initially and then eventually start decreasing in an inverted U shape. At first, λ equals to 0. In this situation, we consider as using least square methods to fit the data using all the variables. Then, the RSS will increase. As λ is increasing, the number of β which will be zero will also increase. Thus, the over-fitting problem will be solved. Therefore, the RSS will decrease.

(iii): Variance: Steadily decrease. When λ equals to 0, we use least square method to fit the data, and we include all the variables. Therefore, the variance may be large. As λ increases, the coefficients will become shrunken and decrease to 0, and we have less variables, When λ is infinity, we will have no variables and the variance becomes approximately 0.

(iv): (squared) bias: Steadily increase. According to the formula,

Test Error = Bias2 + Variance

When λ is 0, we have all the variables including true model's variables. In this situation, the bias2 is the smallest. As λ starts to increase, β will start to decrease, therefore, we may have less variables to fit the data. When λ becomes infinity, we have no variables and bias2 will be the largest.

(2) for Lasso Regression

$$||y - X\beta||^2 + \lambda \sum_{j} |\beta_j|$$

Equivalently, find β that minimizes

$$||y - X\beta||^2$$

Subject to the constraint that

$$\sum_{j=1}^{p} |\beta_j| \leq s$$

(i): training RSS: Steadily decease. If s increases from 0, when s is extremely large, the results will be estimates of using least square method. Therefore, When s equals zero, there will be no betas, and training RSS is the biggest. When s starts to increase, the training RSS will decrease, and when s is close to infinity, it will reach to the OLS RSS

(ii): test RSS: Decrease initially, and then eventually start decreasing in an inverted U shape. When s=0, there will be no betas, and the test RSS is the biggest. As s starts to increase, the number of non-zero betas will start to decrease, and test RSS will begin to decrease. While s continues to increase, the over-fitting problems will appear, and RSS will begin to increase.

(iii): Variance: When s=0, the model has no betas and is the simplest, therefore the variance is the smallest. While s starts to increase, the number of non-zero betas will decrease, and the variance will start to increase.

(iv): (squared) bias: When s=0, the model has no betas, and is not a true model, therefore the bis is the biggest. While s starts to increase, the model is more close to true model, the bias will start to decrease.

2. (25 pt) This problem illustrates the estimator property in the shrinkage methods. Let Y be a single observation. Consider Y regressed on an intercept

$$Y = 1 \cdot \beta + \epsilon$$

(a) Using the formulation as shown in class, write down the optimization problem of general linear model, ridge regression and LASSO in estimating β respectively.

General linear model:

$$\begin{matrix} min \\ \beta \end{matrix} = (Y - \beta)^2$$

Ridge Regression:

$$\begin{matrix} min \\ \beta \end{matrix} = (Y - \beta)^2 + \lambda\beta^2$$

Lasso Regression:

$$\begin{matrix} min \\ \beta \end{matrix} = (Y - \beta)^2 + \lambda|\beta|$$

(b) For fixed tuning parameter λ, solve for β (general linear model), β_λ (ridge regression) and β_λ

(LASSO) respectively.

1. generalized linear model

$$\min_{\beta} = (Y-\beta)^2$$

$$\hat{\beta} = Y.$$

2. Ridge regression.

$$\min_{\beta} (Y-\beta)^2 + \lambda\beta^2 = Loss_1(\beta)$$

$$Loss_1(\beta) = Y^2 + \beta^2 - 2\beta Y + \lambda\beta^2$$

take the derivate of β

$$\frac{\partial L}{\partial \beta} = 2\beta - 2Y + 2\lambda\beta$$

$$\frac{\partial L}{\partial \beta} = -2(Y-\beta) + 2\lambda\beta$$

when $-2(Y-\beta) + 2\lambda\beta = 0$

$$\hat{\beta}^R_\lambda = \frac{Y}{1-\lambda}$$

3. Ridge Regression

$$\min_{\beta} (Y-\beta)^2 + \lambda|\beta|$$

$$L(\beta) \begin{cases} (Y-\beta)^2 + \lambda\beta & \beta \geq 0 \\ (Y-\beta)^2 - \lambda\beta & \beta < 0 \end{cases}$$

(1)

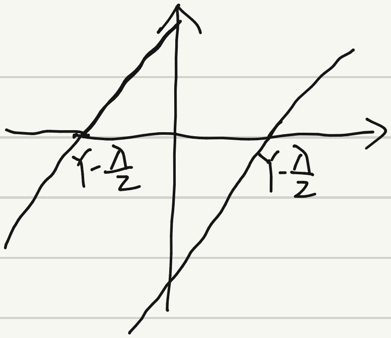
When $\beta \geq 0$

$$Y^2 + \beta^2 - 2Y\beta + \lambda\beta$$

Take the first derivate of β

$$\frac{\partial L}{\partial \beta} = 2\beta - 2Y + \lambda = 0$$

$$2\beta = 2Y - \lambda \Rightarrow \beta = Y - \frac{\lambda}{2}$$



$$\text{argmin } \beta = \begin{cases} Y-\frac{\lambda}{2} & \text{if } Y-\frac{\lambda}{2} \geqslant 0 \\ 0 & Y-\frac{\lambda}{2} < 0 \end{cases}$$

$$\min L(\beta) = \begin{cases} \lambda Y - \frac{1}{4}\lambda^2 & \text{if } Y-\frac{\lambda}{2} \geqslant 0 \\ Y^2 & Y-\frac{\lambda}{2} < 0 \end{cases}$$

(2) $\beta < 0$

$(Y-\beta)^2 - \lambda\beta$

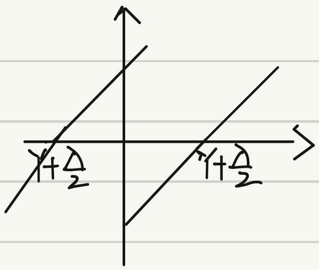
$Y^2+\beta^2-2Y\beta-\lambda\beta$

take the first derivate

$Loss_2(\beta) = Y^2+\beta^2-2Y\beta-\lambda\beta.$

$\frac{\partial L}{\partial \beta} = 2\beta - 2Y - \lambda.$

$\frac{\partial L}{\partial \beta} = 0 \Rightarrow \beta = Y + \frac{\lambda}{2}$



$$\text{argmin } \beta = \begin{cases} 0 & Y+\frac{\lambda}{2} > 0 \\ Y+\frac{\lambda}{2} & Y+\frac{\lambda}{2} \leq 0 \end{cases}$$

$$\min L(\beta) = \begin{cases} Y^2 & Y+\frac{\lambda}{2} > 0 \\ -\frac{1}{4}\lambda^2 - \lambda Y & Y+\frac{\lambda}{2} \leq 0 \end{cases}$$

Then, we need to compare different situations to get the smallest valllle.

When $Y < \frac{\lambda}{2}$

$\min L(\beta) = \lambda Y - \frac{1}{4}\lambda^2$

$\min L(\beta) = Y^2$

$\lambda Y - \frac{1}{4}\lambda^2 < Y^2.$

$\hat{\beta}^L_\lambda = Y - \frac{\lambda}{2}$

When $-\frac{\lambda}{2} < Y \leq \frac{\lambda}{2}$

$\min_{\beta\leq 0} L(\beta) = Y^2 \qquad \min_{\beta\geqslant 0} L(\beta) = Y^2$

when $Y \leq -\frac{\lambda}{2}$

$\min_{\beta\leq 0} L(\beta) = Y^2, \quad \min_{\beta\geqslant 0} L(\beta) = -\lambda Y - \frac{1}{4}\lambda^2$

Part C

$$\hat{\beta} = Y$$

$$\hat{\beta}_{\lambda}^{R} = \frac{Y}{1+\lambda}$$

$$\hat{\beta}_{\lambda}^{L} = \begin{cases} Y+\frac{\lambda}{2} & Y \leq -\frac{\lambda}{2} \\ 0 & -\frac{\lambda}{2} \leq Y \leq \frac{\lambda}{2} \\ Y-\frac{\lambda}{2} & Y > \frac{\lambda}{2} \end{cases}$$

3.

Lasso: $||y-x\beta||^2+\lambda\sum_j|\beta_j|$

(a) $\min_\beta (y_1-x_1\beta_1-x_1\beta_2)^2+(y_2-x_2\beta_1-x_2\beta_2)^2+\lambda(\beta_1^2+\beta_2)^2=\ell(\beta)$

b) $\frac{\partial\ell(\beta)}{\partial\beta_1}=-2x_1(y_1-x_1\beta_1-x_1\beta_2)-2x_2(y_2-x_2\beta_1-x_2\beta_2)+2\lambda\beta_1=0$

$\frac{\partial\ell(\beta)}{\partial\beta_2}=-2x_1(y_1-x_1\beta_1-x_1\beta_2)-2x_2(y_2-x_2\beta_1-x_2\beta_2)+2\lambda\beta_2=0$

$$\hat\beta^R_{\lambda,1}=\hat\beta^R_{\lambda,2}$$

c) $\min_\beta (y_1-x_1\beta_1-x_1\beta_2)^2+(y_2-x_2\beta_1-x_2\beta_2)^2+\lambda(|\beta_1|+|\beta_2|)$

$\beta_1\neq0,\ \beta_2\neq0 \qquad \lambda\to\infty$

d) $\min\ell(\beta)=2(y_1-x_1(\beta_1+\beta_2))^2$

$$\hat\beta_1+\hat\beta_2=\frac{y_1}{x_1}$$

$\min_\beta (y_1-x_1\beta_1-x_1\beta_2)^2+(y_2-x_2\beta_1-x_2\beta_2)^2$

$|\beta_1|+|\beta_2|\leq s$

when $\frac{y_1}{x_1}>0$,

$\hat\beta^L_{\lambda,1}+\hat\beta^L_{\lambda,2}=s$

$\hat\beta^L_{\lambda,1}\geq0 \qquad \hat\beta^L_{\lambda,2}\geq0$

when $\frac{y_1}{x_1}<0$,

$\hat\beta^L_{\lambda,1}+\hat\beta^L_{\lambda,2}=-s$

$\hat\beta^L_{\lambda,1}\leq0 \qquad \hat\beta^L_{\lambda,2}\leq0$

Solution is not unique