SML HW#6

2022-10-30

```r
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-4
```

#Problem 1 a) The best subset model with k predictors has the smallest of the training RSS. This is because it evaluates every possible model for the set of predictors. Therefore for a very large amount of k predictors the best subset model has a better chance of finding the model that fits the training data the best.

b) Any of the three methods could have the lowest test RSS. This is because even though best subset considers many more models than forward and backward selection, forward and backward selection could still find the model that fits the test set data better than the best subset method. This would produce a lower RSS.

c)

d) True. Forward selection methods forms the model with k+1 predictors by adding an addition predictor to the model with k predictors.

ii) True. Backward selection methods forms the model with k predictors by subtracting a predictor from the model with k + 1 predictors.
iii) False, forward and backward selection can select different predictors.
iv) False, backward and forward selection can select different predictors.
v) False, the model with k+1 predictors does not necessarily contain the same predictors as the model with k predictors for the best subset method.

#Problem 2 a) LASSO RELATIVE TO LEAST SQUARES : iii) Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. There is a large decrease in the variance for a small increase in the bias. This is when lambda increase, thus causing the flexibility of the fit to decrease and estimated coefficients decrease (some to zero).

b) RIDGE REGRESSION RELATIVE TO LEAST SQUARES : iii) Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. There is a large decrease in the variance for a small increase in the bias. This is when lambda increase, thus causing the flexibility of the fit to decrease and estimated coefficients decrease.

c) NON-LINEAR METHOD RELATIVE TO LEAST SQUARES : ii) More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias. Predictions improve when the increase in variance is less than the decrease in the bias.

#Problem 4 a) As we increase λ from 0, the training RSS will: iii) Steadily increase. As λ increases the estimates of β becomes smaller, therefore the RSS will increase.

b) As we increase λ from 0, the test RSS will: ii) Decrease initially, and then eventually start increasing in an inverted U shape. As we increase λ the flexibility of the fit decreases therefore reducing the variance of predictions for a small increase in bias. We will start to see more accurate predictions which will initially decrease the test RSS. When λ is too large we will see a large increase in bias in comparison to the decrease in variance, thus making the predictions biased and we see a rise in the test RSS.

c) As we increase λ from 0, the variance will: iv) Steadily decrease. The decrease in variance is due to flexibility decrease that occurs when λ increases.

d) As we increase λ from 0, the (squared) bias will: iii) Steadily increase. The increase in (squared) bias is due to flexibility decrease that occurs when λ increases.

e) As we increase λ from 0, the irreducible error will: v) Remains constant. The irreducible error will never change.

#Problem 8 a), b)

```
set.seed(0)
X <- rnorm(100)
eps <- rnorm(100, 0, 0.05)
beta0 <- 6
beta1 <- 4
beta2 <- 0.5
beta3 <- 1.5


Y <- beta0 + beta1*X + beta2*(X^2) +  beta3*(X^3) + eps
```

#c)

```
data <- data.frame(Y, X, X^2, X^3, X^4, X^5, X^6, X^7, X^8, X^9, X^10)

fit <- regsubsets(Y ~ ., data = data)
summary(fit)$cp
```

```
## [1] 1.993131e+05 1.031527e+04 2.241973e+00 2.918931e+00 3.080799e+00
## [6] 4.596512e+00 5.878131e+00 7.847182e+00
```

```
summary(fit)$bic
```

```
## [1] -228.5848 -519.2458 -985.0993 -981.9079 -979.3008 -975.2288 -971.4197
## [8] -966.8490
```

```
which.min(summary(fit)$bic)
```

```
## [1] 3
```

```
summary(fit)$adjr2
```

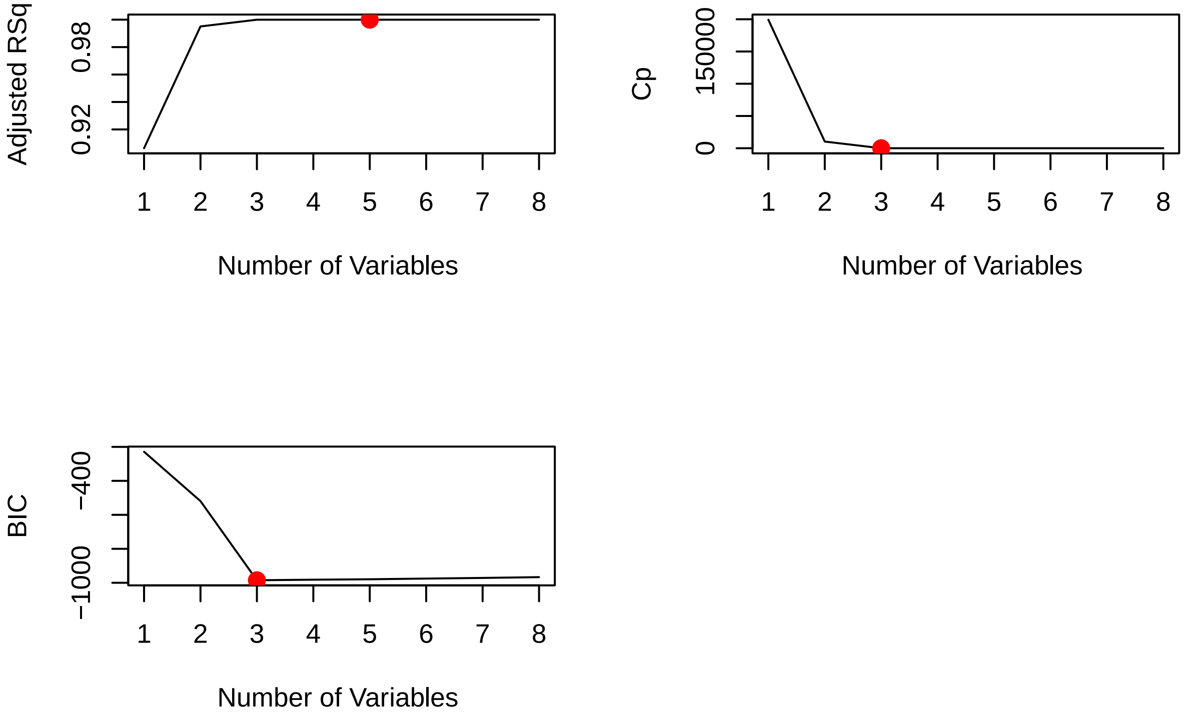
```
## [1] 0.9063133 0.9950591 0.9999548 0.9999550 0.9999554 0.9999551 0.9999550
## [8] 0.9999545
```

#The cp decreases for the models with one, two, and three then it increases in small amounts after that. #The bic is the smallest for the model with 3 variables. #The adjusted R^2 increases to 0.9999 from 0.9944 in the model from two to three variables. Does not change much for the models with more variables after it rises to 0.9999. #The overall best model is the model including X, X^2, X^3. We can see this in the plots below as well.

```
par(mfrow=c(2,2))
plot(summary(fit)$adjr2,xlab="Number of Variables ",
ylab="Adjusted RSq",type="l")
points(which.max(summary(fit)$adjr2), summary(fit)$adjr2[which.max(summary(fit)$adjr2)], col="red",cex=

plot(summary(fit)$cp ,xlab="Number of Variables ",ylab="Cp", type='l')
points(which.min(summary(fit)$cp), summary(fit)$cp[which.min(summary(fit)$cp)],col="red",cex=2,pch=20)
```

```
plot(summary(fit)$bic ,xlab="Number of Variables ",ylab="BIC", type='l')
points(which.min(summary(fit)$bic),summary(fit)$bic[which.min(summary(fit)$bic)],col="red",cex=2,pch=20
```



#d) #BACKWARD SELECTION

```
set.seed(0)
fitback <- regsubsets(Y ~ ., data = data, method = "backward")
summary(fitback)$cp
```

```
## [1] 1.993131e+05 1.031527e+04 2.241973e+00 3.278334e+00 3.080799e+00
## [6] 4.613489e+00 5.878131e+00 7.847182e+00
```

```
summary(fitback)$bic
```

```
## [1] -228.5848 -519.2458 -985.0993 -981.5219 -979.3008 -975.2100 -971.4197
## [8] -966.8490
```

```
which.min(summary(fitback)$bic)
```

```
## [1] 3
```

```
summary(fitback)$adjr2
```

```
## [1] 0.9063133 0.9950591 0.9999548 0.9999548 0.9999554 0.9999551 0.9999550
## [8] 0.9999545
```

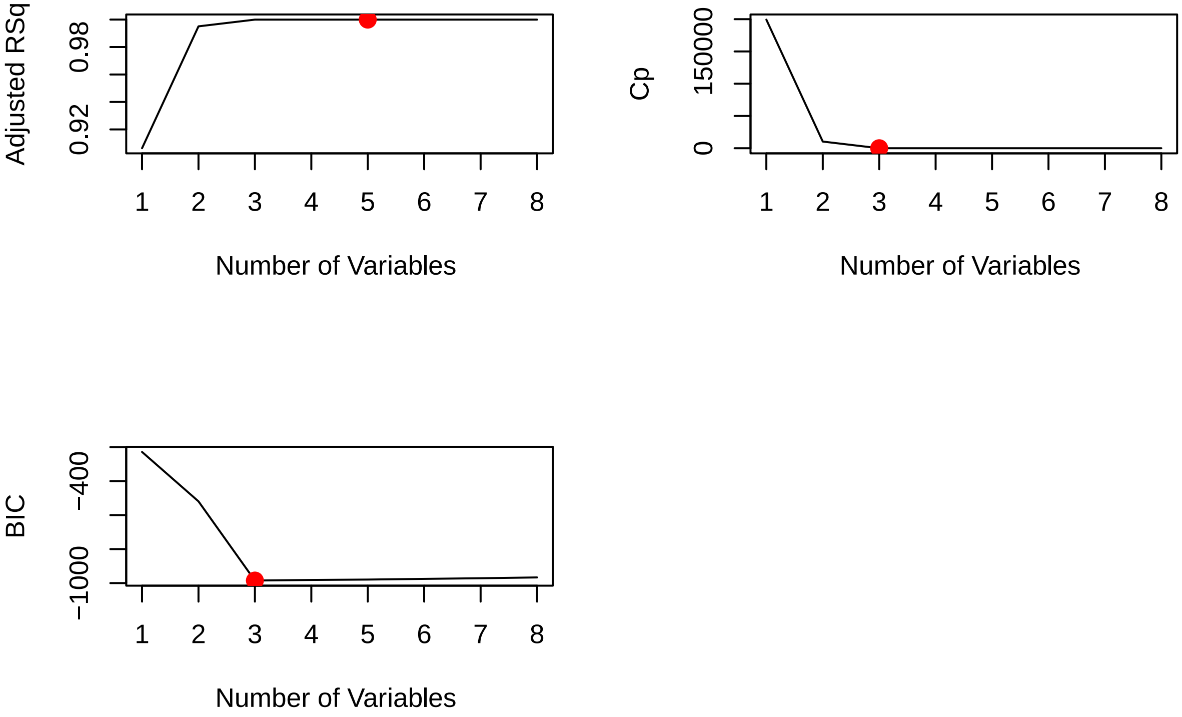
#The cp decreases for the models with one, two, and three then it increases in small amounts after that. #The bic is the smallest for the model with 3 variables. #The adjusted R^2 increases to 0.9999 from 0.9944 in the model from two to three variables. Does not change much for the models with more variables after it rises to 0.9999. #The overall best model is the model including X, X^2, X^3. We can see this in the plots below as well.

```
par(mfrow=c(2,2))
plot(summary(fitback)$adjr2,xlab="Number of Variables ",
ylab="Adjusted RSq",type="l")
```

```
points(which.max(summary(fitback)$adjr2), summary(fitback)$adjr2[which.max(summary(fitback)$adjr2)], co

plot(summary(fitback)$cp ,xlab="Number of Variables ",ylab="Cp", type='l')
points(which.min(summary(fitback)$cp), summary(fitback)$cp[which.min(summary(fitback)$cp)],col="red",ce

plot(summary(fitback)$bic ,xlab="Number of Variables ",ylab="BIC", type='l')
points(which.min(summary(fitback)$bic),summary(fitback)$bic[which.min(summary(fitback)$bic)],col="red",
```



#FORWARD SELECTION

```
set.seed(0)
fitfor <- regsubsets(Y ~ ., data = data, method = "forward")
summary(fitfor)$cp
```

```
## [1] 1.993131e+05 1.031527e+04 2.241973e+00 2.918931e+00 4.245930e+00
## [6] 5.060332e+00 6.725838e+00 8.077667e+00
```

```
summary(fitfor)$bic
```

```
## [1] -228.5848 -519.2458 -985.0993 -981.9079 -978.0297 -974.7181 -970.4809
## [8] -966.5928
```

```
which.min(summary(fitfor)$bic)
```

```
## [1] 3
```

```
summary(fitfor)$adjr2
```

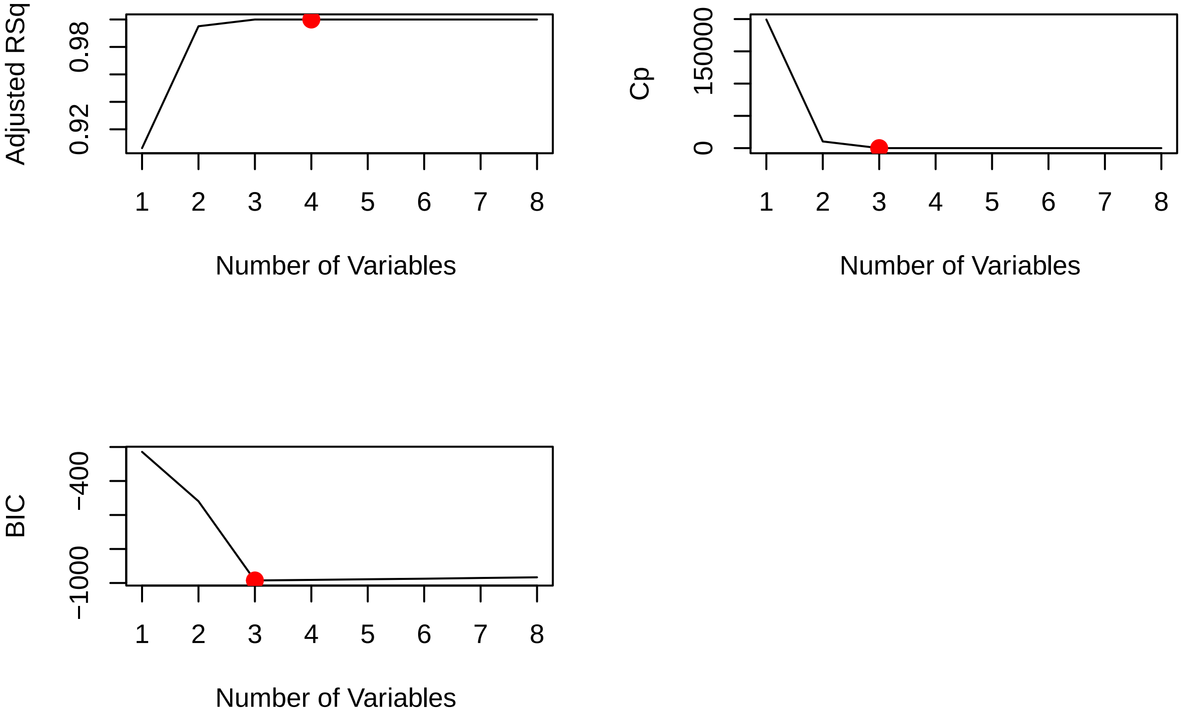
```
## [1] 0.9063133 0.9950591 0.9999548 0.9999550 0.9999548 0.9999549 0.9999546
## [8] 0.9999544
```

#The cp decreases for the models with one, two, and three then it increases in small amounts after that (with the exception of the 5 variable model which decreases again). #The bic is the smallest for the model with 3 variables. #The adjusted Rˆ2 increases to 0.9999 from 0.9944 in the model from two to three variables. Does not change much for the models with more variables after it rises to 0.9999. #The overall best model is the model including X, X^2, X^3. We can see this in the plots below as well.

```
par(mfrow=c(2,2))
plot(summary(fitfor)$adjr2,xlab="Number of Variables ",
ylab="Adjusted RSq",type="l")
points(which.max(summary(fitfor)$adjr2), summary(fitfor)$adjr2[which.max(summary(fitfor)$adjr2)], col="

plot(summary(fitfor)$cp ,xlab="Number of Variables ",ylab="Cp", type='l')
points(which.min(summary(fitfor)$cp), summary(fitfor)$cp[which.min(summary(fitfor)$cp)],col="red",cex=2

plot(summary(fitfor)$bic ,xlab="Number of Variables ",ylab="BIC", type='l')
points(which.min(summary(fitfor)$bic),summary(fitfor)$bic[which.min(summary(fitfor)$bic)],col="red",cex
```

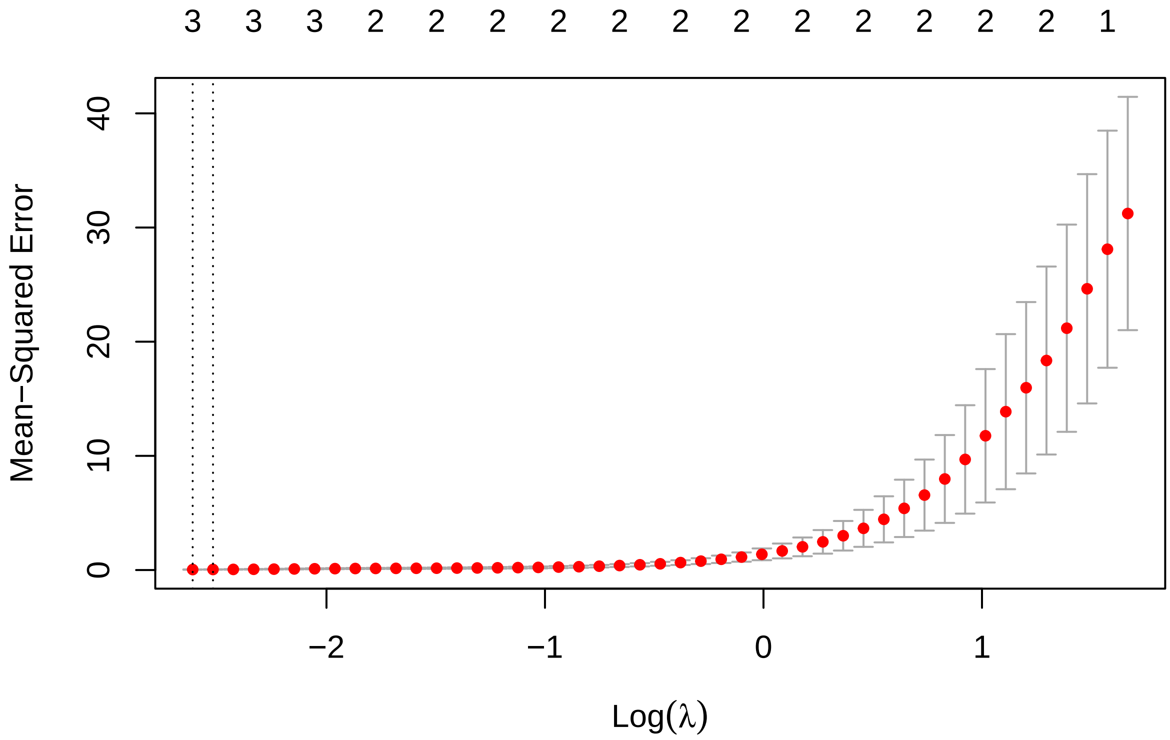


#e)

```
set.seed(0)
x <- model.matrix(Y ~. , data)[,-1]
y <- data$Y

train <- sample(1:nrow(x), nrow(x)/2)
test <- (-train)
y.test <- y[test]

lasso <- glmnet(x[train,], y[train], alpha = 1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
plot(cv.out)
```



```
lam <- cv.out$lambda.min
lam
```

```
## [1] 0.07336514
```

```
coef <- glmnet(x, y, alpha = 1)
lasso.coef <- predict(coef, type = "coefficient", s = 0.20)[1:11,]
lasso.coef
```

```
## (Intercept)           X         X.2         X.3         X.4         X.5
##   6.1190028   3.8295961   0.3494854   1.4939401   0.0000000   0.0000000
##         X.6         X.7         X.8         X.9        X.10
##   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
```

#From the plot above we can see that for higher values of lambda there will be an increase in the MSE. The optimal value of lambda is 0.2.

#The lasso model is optimal with four variables. This will include the intercepet, X, Xˆ2, Xˆ3.

#f) BEST SUBSET

```
set.seed(0)
beta7 <- 7

Y1 <- beta0 + beta7*(X^7) + eps
data1 <- data.frame(Y1, X, X^2, X^3, X^4, X^5, X^6, X^7, X^8, X^9, X^10)

fit1 <- regsubsets(Y1 ~ ., data = data1)
summary(fit1)$cp
```

```
## [1] -0.9408820  0.2576803  0.6955842  0.1994045  1.9345782  3.9318120  5.8472253
## [8]  7.2620084
```

```
summary(fit1)$bic
```

```
## [1] -1850.679 -1846.920 -1843.986 -1842.111 -1837.800 -1833.198 -1828.686
## [8] -1824.735
```

```
which.min(summary(fit1)$bic)
```

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

```
summary(fit1)$adjr2
```

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

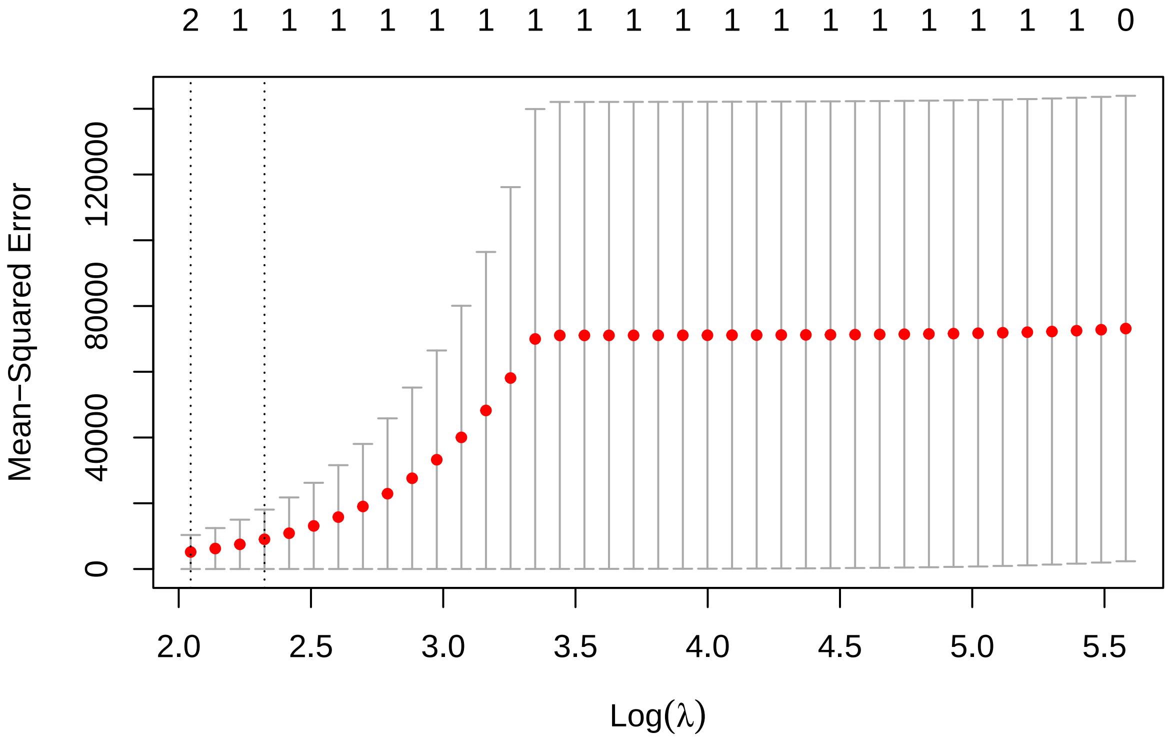
#The lowest Cp value is the one variable model. #The lowest BIC value is the one variable model. #The adjusted R2 is 0.99999 for the one variable model.

LASSO

```
set.seed(0)
x1 <- model.matrix(Y1 ~. , data1)[,-1]
y1 <- data1$Y1

train1 <- sample(1:nrow(x1), nrow(x1)/2)
test1 <- (-train1)
y.test1 <- y[test1]

lasso1 <- glmnet(x1[train1,], y1[train1], alpha = 1)
cv.out1 <- cv.glmnet(x1[train1,], y1[train1], alpha = 1)
plot(cv.out1)
```



```
lam1 <- cv.out1$lambda.min
lam1
```

```
## [1] 7.730763
```

```
coef1 <- glmnet(x1, y1, alpha = 1)
lasso.coef1 <- predict(coef1, type = "coefficient", s = 2.2)[1:11,]
lasso.coef1
```

```
## (Intercept)           X         X.2         X.3         X.4         X.5 
##   7.5073454   0.0000000   0.0000000   0.0000000   0.0000000   0.1112426 
##         X.6         X.7         X.8         X.9        X.10 
##   0.0000000   6.7764350   0.0000000   0.0000000   0.0000000
```

#Lasso assigns non-zero coefficients to Xˆ5 and Xˆ7. The coefficient on Xˆ5 is very small. So we can conclude the optimal model is the model with one variable of Xˆ7