Homework of Dataminning, CH6

Zexian Wang, Student ID 15420151152805

2017-03-31

Q8

(a)

```
set.seed(1234)
n <- 100
x <- rnorm(100)
e <- rnorm(100)
```

(b)

```
X <- as.matrix(cbind(rep(1,n), x, x^2, x^3))
colnames(X) <- c("intercept", "x1", "x2", "x3")
beta <- c(1,2,3,4)
Y <- X %*% beta + e
```

(c)

```
library(leaps)
d <- 10
data <- as.data.frame(cbind(Y, poly(x,degree = d,raw = T)))
names(data) <- c('Y',paste0("X",1:d))
```

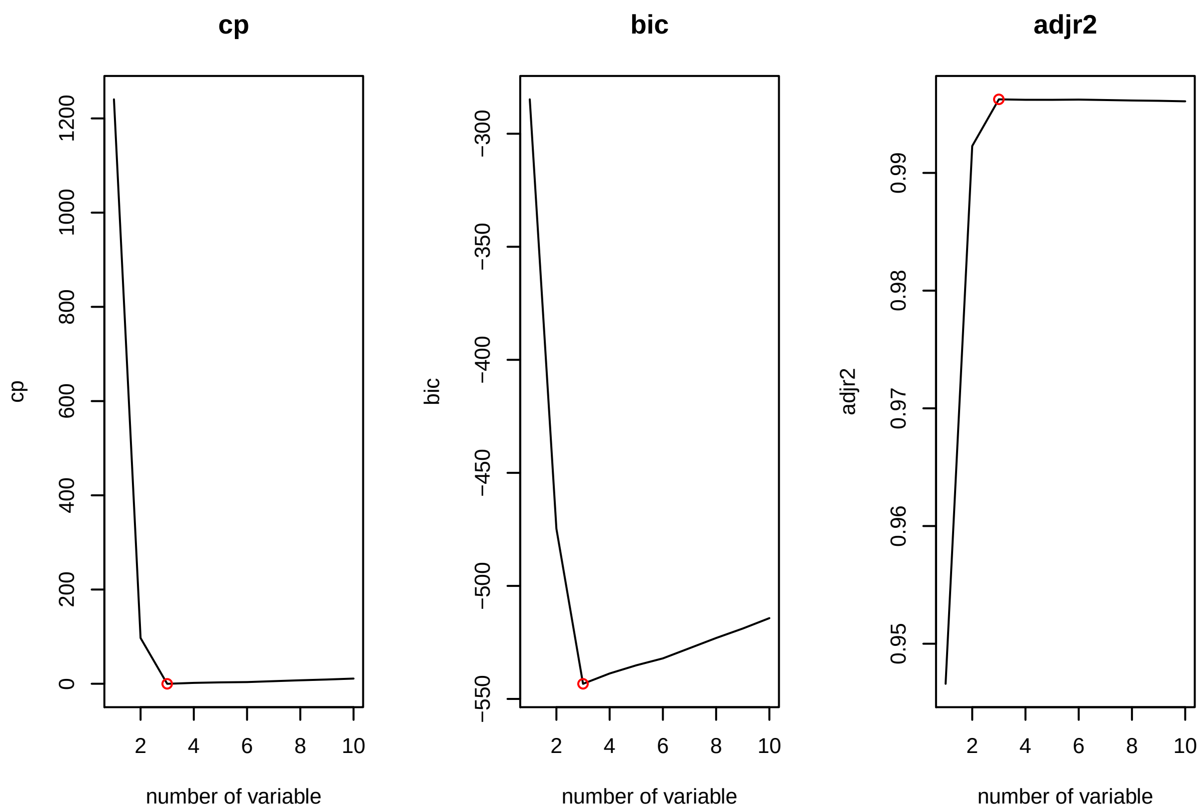
best subset

```
bestsubset <- regsubsets(Y~., data, nvmax = 10)
bestsubset_summary <- summary(bestsubset)

par(mfrow = c(1,3))
plot(bestsubset_summary$cp,type = "l",xlab = "number of variable", ylab = "cp", main = "cp")
whichcp <- which.min(bestsubset_summary$cp)
points(whichcp, bestsubset_summary$cp[whichcp], col = "red")

plot(bestsubset_summary$bic,type = "l",xlab = "number of variable", ylab = "bic", main = "bic")
whichbic <- which.min(bestsubset_summary$bic)
points(whichbic, bestsubset_summary$bic[whichbic], col = "red")

plot(bestsubset_summary$adjr2,type = "l",xlab = "number of variable", ylab = "adjr2", main = "adjr2")
whichadjr2 <- which.max(bestsubset_summary$adjr2)
points(whichadjr2, bestsubset_summary$adjr2[whichadjr2], col = "red")
```




```r
# based on Cp
whichcp
```

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

```r
coef(bestsubset,whichcp)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

```r
# based on BIC
whichbic
```

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

```r
coef(bestsubset,whichbic)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

```r
# based on adjR2
whichadjr2
```

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

```r
coef(bestsubset,whichadjr2)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

(d)

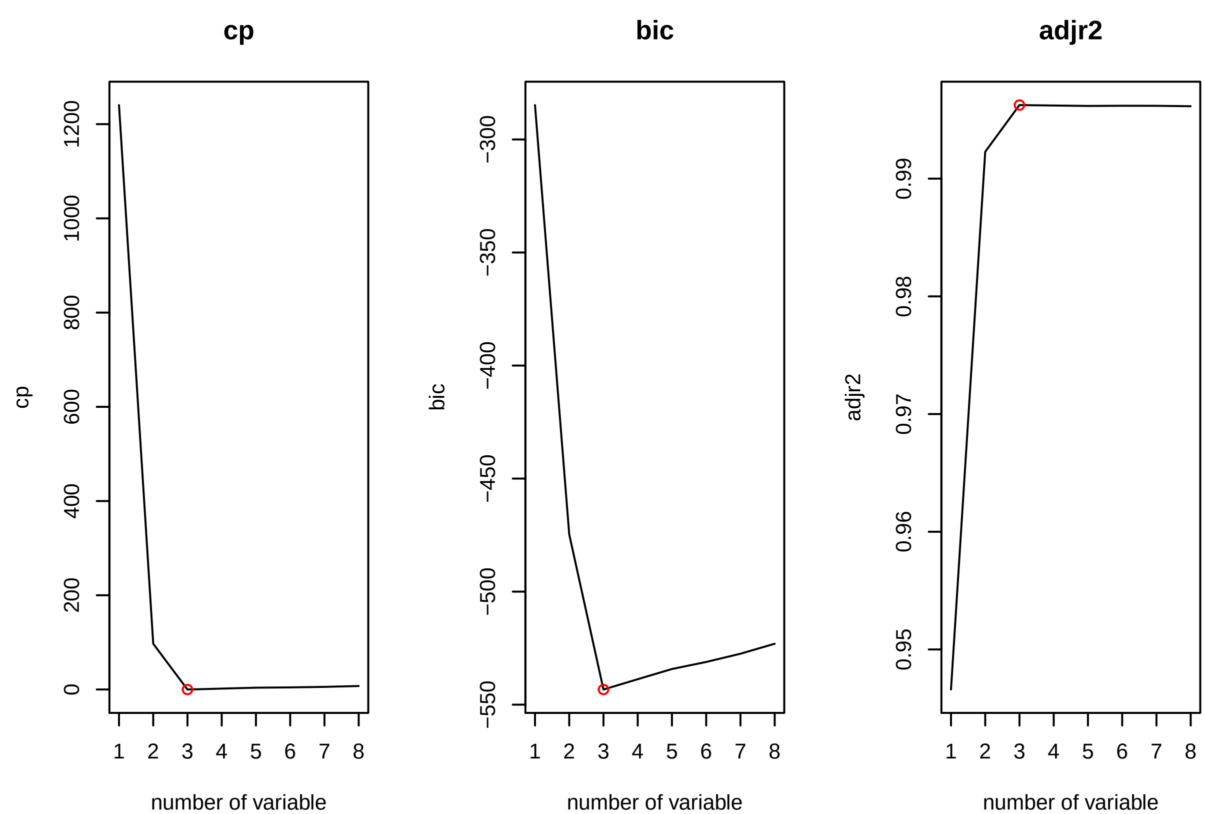
forward

```r
fw <- regsubsets(Y~., data, method = "forward")
summaryfw <- summary(fw)

par(mfrow = c(1,3))
plot(summaryfw$cp,type = "l",xlab = "number of variable", ylab = "cp", main = "cp")
whichcp <- which.min(summaryfw$cp)
points(whichcp, summaryfw$cp[whichcp], col = "red")

plot(summaryfw$bic,type = "l",xlab = "number of variable", ylab = "bic", main = "bic")
whichbic <- which.min(summaryfw$bic)
points(whichbic, summaryfw$bic[whichbic], col = "red")

plot(summaryfw$adjr2,type = "l",xlab = "number of variable", ylab = "adjr2", main = "adjr2")
whichadjr2 <- which.max(summaryfw$adjr2)
points(whichadjr2, summaryfw$adjr2[whichadjr2], col = "red")
```




```r
# based on Cp
whichcp
```

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

```r
coef(fw,whichcp)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

```r
# based on BIC
whichbic
```

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

```r
coef(fw,whichbic)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

```r
# based on adjR2
whichadjr2
```

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

```r
coef(fw,whichadjr2)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

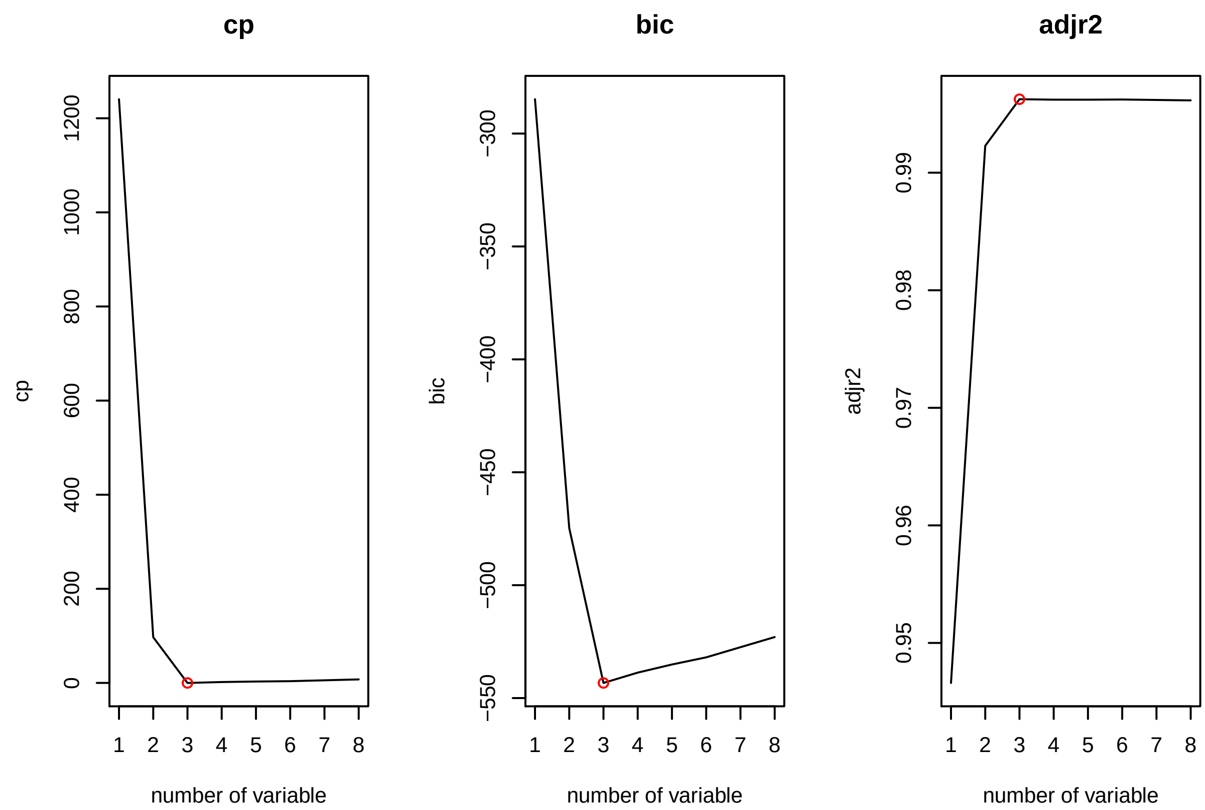
backward

```r
bw <- regsubsets(Y~., data, method = "backward")
summarybw <- summary(bw)

par(mfrow = c(1,3))
plot(summarybw$cp,type = "l",xlab = "number of variable", ylab = "cp", main = "cp")
whichcp <- which.min(summarybw$cp)
points(whichcp, summarybw$cp[whichcp], col = "red")

plot(summarybw$bic,type = "l",xlab = "number of variable", ylab = "bic", main = "bic")
whichbic <- which.min(summarybw$bic)
points(whichbic, summarybw$bic[whichbic], col = "red")

plot(summarybw$adjr2,type = "l",xlab = "number of variable", ylab = "adjr2", main = "adjr2")
whichadjr2 <- which.max(summarybw$adjr2)
points(whichadjr2, summarybw$adjr2[whichadjr2], col = "red")
```




```r
# based on Cp
whichcp
```

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

```r
coef(bw,whichcp)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

```r
# based on BIC
whichbic
```

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

```r
coef(bw,whichbic)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

```r
# based on adjR2
whichadjr2
```

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

```r
coef(bw,whichadjr2)
```

```
## (Intercept)          X1          X2          X3
##    1.132470    1.912586    2.893627    4.032305
```

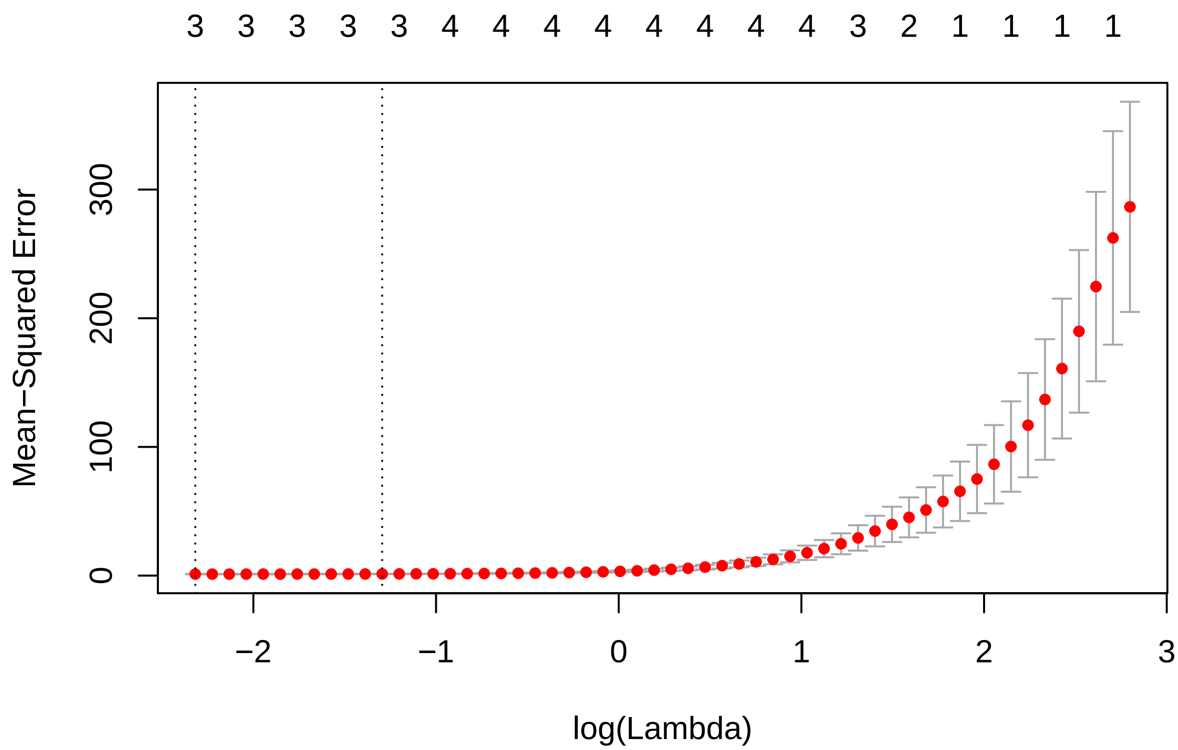
All the model choose the right model.

(e)

```r
par(mfrow = c(1,1))
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-5
```

```r
modelx <- model.matrix(Y~., data)[,-1]
modely <- data$Y
modellasso <- cv.glmnet(modelx, modely, alpha = 1)
plot(modellasso)
```




```r
coef(modellasso)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                    1
## (Intercept) 1.275528
## X1          1.730667
## X2          2.727971
## X3          4.015361
## X4          .
## X5          .
## X6          .
## X7          .
## X8          .
```

```
## X9             .
## X10            .
```

Lasso model also choose the right variables.

(f)

```
beta7 <- 5
Y = rep(beta[1],n) + beta7*x^7 + e
data$Y <- Y
```

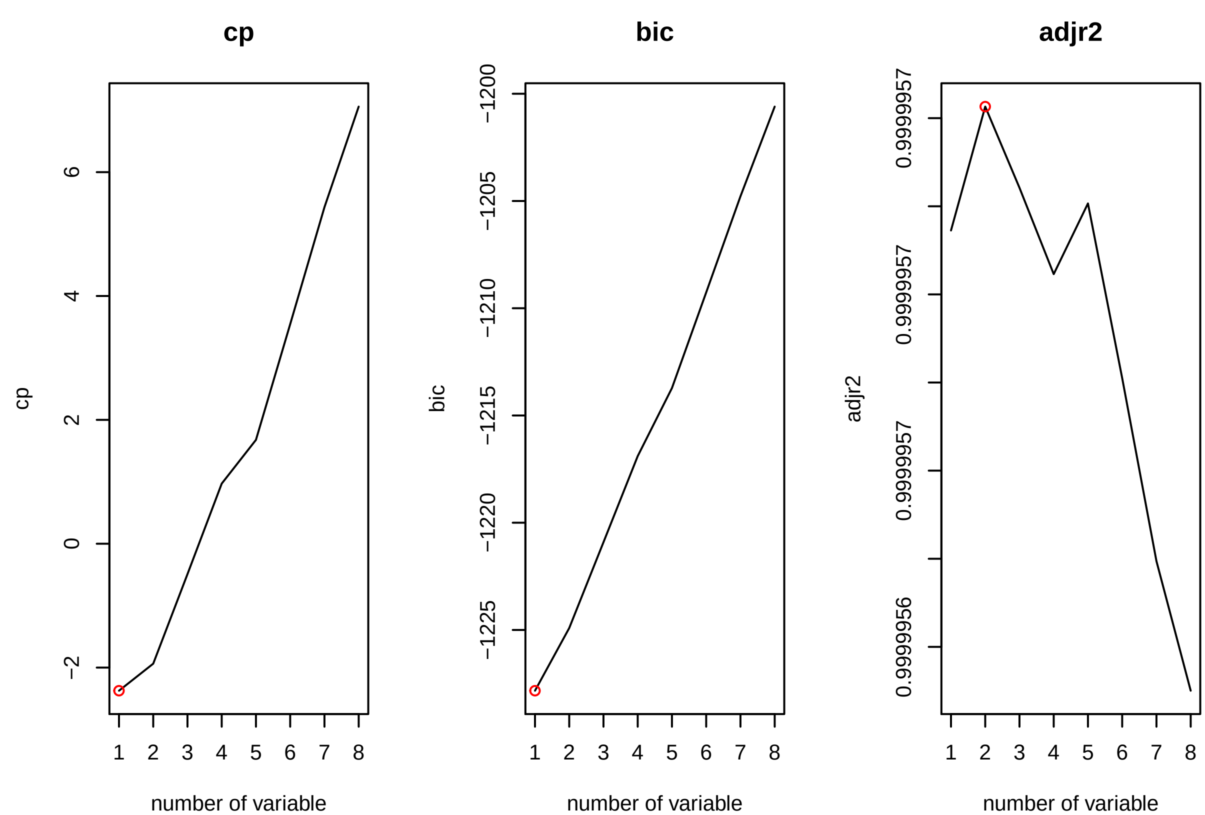
best subset

```
bestsubset <- regsubsets(Y~., data)
bestsubset_summary <- summary(bestsubset)

par(mfrow = c(1,3))
plot(bestsubset_summary$cp,type = "l",xlab = "number of variable", ylab = "cp", main = "cp")
whichcp <- which.min(bestsubset_summary$cp)
points(whichcp, bestsubset_summary$cp[whichcp], col = "red")

plot(bestsubset_summary$bic,type = "l",xlab = "number of variable", ylab = "bic", main = "bic")
whichbic <- which.min(bestsubset_summary$bic)
points(whichbic, bestsubset_summary$bic[whichbic], col = "red")

plot(bestsubset_summary$adjr2,type = "l",xlab = "number of variable", ylab = "adjr2", main = "adjr2")
whichadjr2 <- which.max(bestsubset_summary$adjr2)
points(whichadjr2, bestsubset_summary$adjr2[whichadjr2], col = "red")
```



```r
# based on Cp
whichcp
```

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

```r
coef(bestsubset,whichcp)
```

```
## (Intercept)          X7
##    1.042105    4.999908
```

```r
# based on BIC
whichbic
```

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

```r
coef(bestsubset,whichbic)
```

```
## (Intercept)          X7
##    1.042105    4.999908
```

```r
# based on adjR2
whichadjr2
```

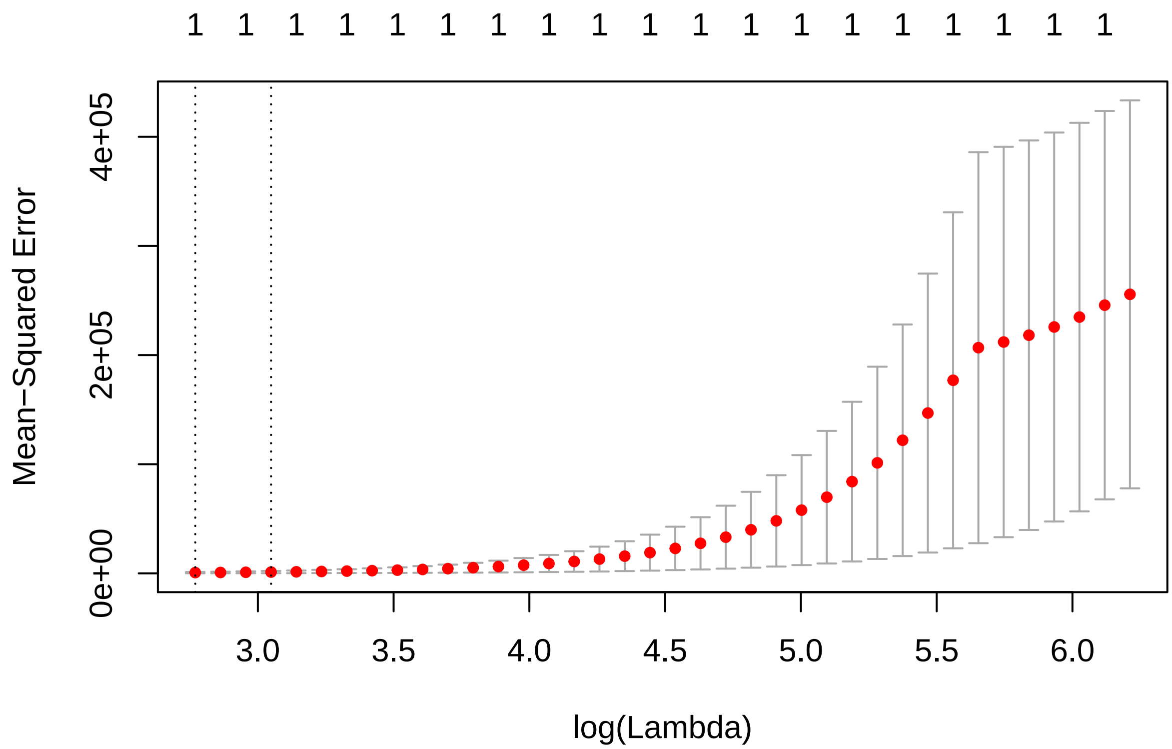
```
## [1] 2
```

```r
coef(bestsubset,whichadjr2)
```

```
## (Intercept)          X2          X7
##   1.1471879  -0.1074417   5.0004274
```

lasso

```r
par(mfrow = c(1,1))
library(glmnet)
modelx <- model.matrix(Y~., data)[,-1]
modely <- data$Y
modellasso <- cv.glmnet(modelx, modely, alpha = 1)
plot(modellasso)
```




```
coef(modellasso)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                    1
## (Intercept) 3.022355
## X1          .
## X2          .
## X3          .
## X4          .
## X5          .
## X6          .
## X7          4.788450
## X8          .
## X9          .
## X10         .
```

Only the best subset selection with adjR2 choose the wrong variable X2

Q9

(a)

```
library(ISLR)
n = nrow(College)
set.seed(1)
College <- College[,c("Apps",names(College)[-2])]
```

```r
trainindex <- sample(n, n/3*2,replace = F)
train <- College[trainindex,]
test <- College[-trainindex,]
```

(b)

LS

```r
modellm <- lm(Apps~.,train)
mean((predict(modellm,newdata = test[,-1])-test$Apps)^2)
```

```
## [1] 925316.1
```

(c)

Ridge

```r
library(glmnet)
trainx <- model.matrix(Apps~., train)[,-1]
testx <- model.matrix(Apps~.,test)[,-1]
trainy <- train$Apps
modelridge <- cv.glmnet(trainx, trainy, alpha = 0)
mean((predict(modelridge,newx = testx)-test$Apps)^2)
```

```
## [1] 1260720
```

(d)

Lasso

```r
trainx <- model.matrix(Apps~., train)[,-1]
testx <- model.matrix(Apps~.,test)[,-1]
trainy <- train$Apps
modellasso <- cv.glmnet(trainx, trainy, alpha = 1)
mean((predict(modellasso,newx = testx)-test$Apps)^2)
```

```
## [1] 1298099
```

```r
sum(coef(modellasso)!=0)
```

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

(e)

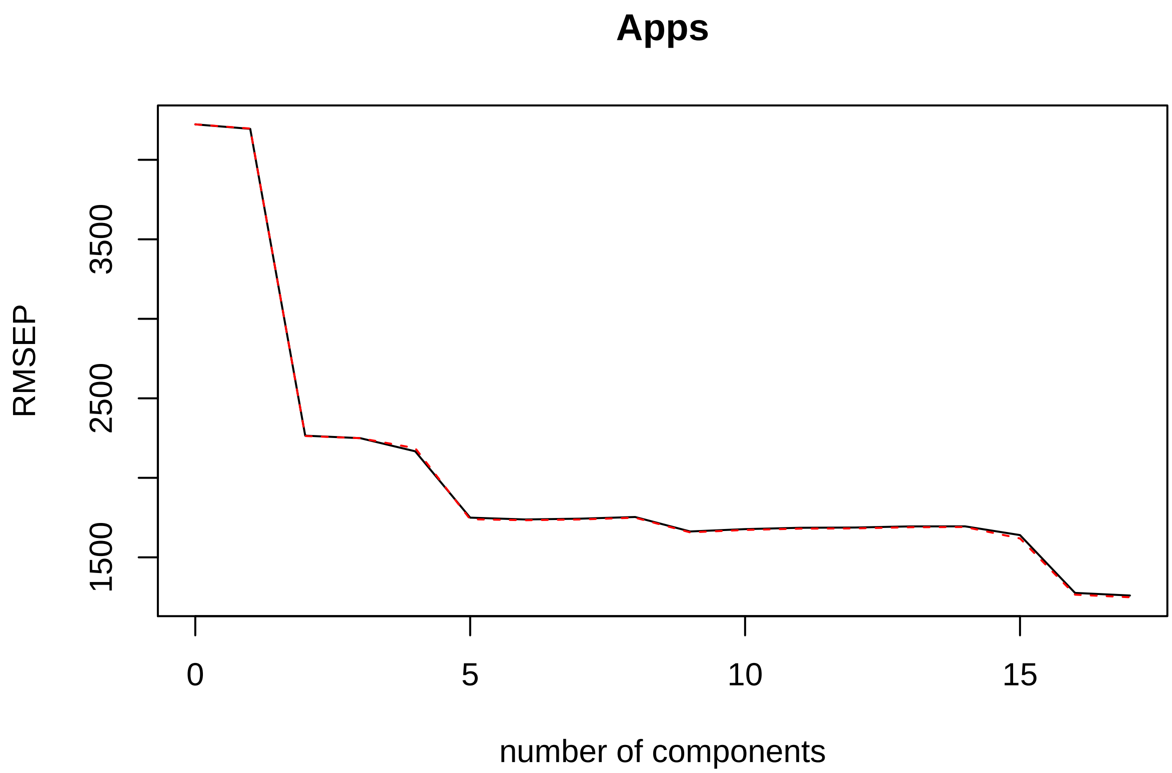
PCR

```r
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```r
set.seed(1)
pcr.fit <- pcr(Apps~., data=train, scale = T, validation = "CV")
validationplot(pcr.fit)
```




```r
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            4223     4195     2265     2249     2167     1750     1738
## adjCV         4223     4196     2263     2249     2187     1740     1733
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1743     1754     1663      1677      1686      1688      1695
## adjCV     1738     1749     1657      1672      1680      1682      1689
##        14 comps  15 comps  16 comps  17 comps
## CV         1695      1640      1276      1260
## adjCV      1691      1619      1265      1249
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      30.930    57.85    64.82    70.64    76.17    81.10    84.63
## Apps    2.145    71.92    72.40    74.19    83.98    84.09    84.17
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.99    90.77     93.06     95.10     96.79     97.93     98.74
## Apps    84.17    85.79     85.87     85.88     85.88     85.90     86.08
```

```
##       15 comps  16 comps  17 comps
## X        99.38     99.85    100.00
## Apps     91.01     93.30     93.55
```

The best M is 17.

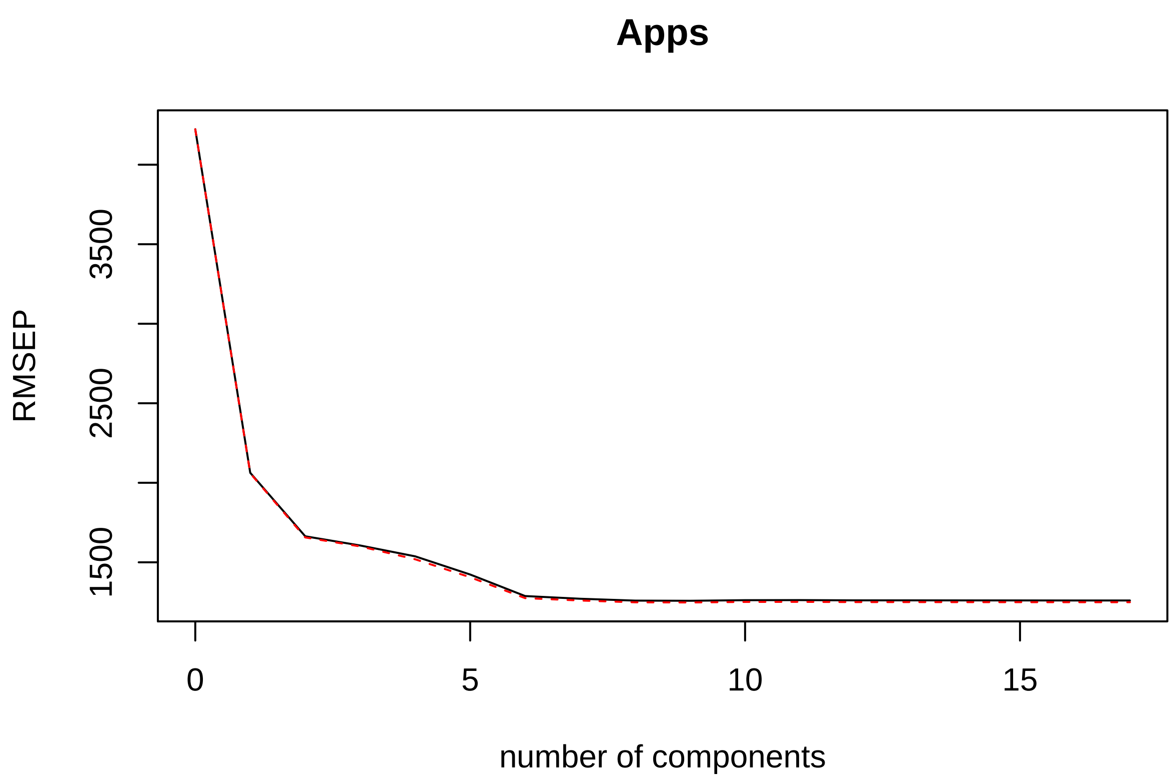
```r
predictpcr <- predict(pcr.fit, test[,-1], ncomp = 17)
mean((predictpcr - test$Apps)^2)
```

```
## [1] 925316.1
```

(f)

PLS

```r
set.seed(1)
pls.fit <- plsr(Apps~., data = train, scale = T, validation = "CV")
validationplot(pls.fit)
```




```r
summary(pls.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           4223     2063     1664     1606     1537     1423     1288
## adjCV        4223     2060     1656     1600     1519     1406     1275
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1271     1259     1258      1262      1262      1261      1261
## adjCV     1260     1248     1247      1251      1251      1250      1250
##        14 comps  15 comps  16 comps  17 comps
## CV         1260      1260      1260      1260
## adjCV      1249      1249      1249      1249
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       26.92    36.26    63.09    65.86    70.29    73.79    78.38
## Apps    77.16    86.34    87.72    91.18    92.67    93.37    93.41
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       80.76    83.65     86.95     89.54     91.09     92.23     94.41
## Apps    93.47    93.51     93.52     93.54     93.55     93.55     93.55
##       15 comps  16 comps  17 comps
## X        96.77     98.31    100.00
## Apps     93.55     93.55     93.55
```

The best M is 9.

```r
predictpls <- predict(pls.fit, test[,-1], ncomp = 9)
mean((predictpls - test$Apps)^2)
```

```
## [1] 931713.9
```

(g)

The MSE of these models are: LS

LS:925316.1

Ridge:1260720

Lasso:1298099

PCR:925316.1

PLS:931713.9

The best model is PCR with 17 compents and Least Square. The worest model is Lasso. However, there are not very obvious difference between these MSEs.