DaigleHomework8.R

daiglechris

Mon Dec 3 13:53:19 2018

Chris Daigle

Homework 8

```r
#

# Exercise: We want to predict the number of applications using the other
# variables in the College data set. You can find this data set in ISLR library.
# Try subset selection, shrinkage methods, and dimension reduction methods and
# examine which method is working best based on training set and test set split.
rm(list = ls())
library(ISLR)
df <- College

set.seed(1)
train <- sample(1:nrow(df), round(nrow(df)) / 2)
dfTrain <- df[train, ]
dfTest <- df[-train, ]
xTest <- dfTest[, -2]
yTest <- dfTest[, 2]
xTrain <- dfTrain[, -2]
yTrain <- dfTrain[, 2]

### Subset Selection
library(leaps)
regFitFull <- regsubsets(Apps ~ ., data = df)
summary(regFitFull)
```

```
## Subset selection object
## Call: regsubsets.formula(Apps ~ ., data = df)
## 17 Variables  (and intercept)
##             Forced in Forced out
## PrivateYes      FALSE      FALSE
## Accept          FALSE      FALSE
## Enroll          FALSE      FALSE
## Top10perc       FALSE      FALSE
## Top25perc       FALSE      FALSE
## F.Undergrad     FALSE      FALSE
## P.Undergrad     FALSE      FALSE
## Outstate        FALSE      FALSE
## Room.Board      FALSE      FALSE
## Books           FALSE      FALSE
## Personal        FALSE      FALSE
## PhD             FALSE      FALSE
## Terminal        FALSE      FALSE
## S.F.Ratio       FALSE      FALSE
```

```
## perc.alumni     FALSE      FALSE
## Expend          FALSE      FALSE
## Grad.Rate       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 ) " "        "*"    " "    " "       " "       " "
## 2  ( 1 ) " "        "*"    " "    "*"       " "       " "
## 3  ( 1 ) " "        "*"    " "    "*"       " "       " "
## 4  ( 1 ) " "        "*"    " "    "*"       " "       " "
## 5  ( 1 ) " "        "*"    "*"    "*"       " "       " "
## 6  ( 1 ) " "        "*"    "*"    "*"       " "       " "
## 7  ( 1 ) " "        "*"    "*"    "*"       "*"       " "
## 8  ( 1 ) "*"        "*"    "*"    "*"       " "       " "
##          P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1  ( 1 ) " "         " "      " "        " "   " "      " " " "
## 2  ( 1 ) " "         " "      " "        " "   " "      " " " "
## 3  ( 1 ) " "         " "      " "        " "   " "      " " " "
## 4  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 5  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 6  ( 1 ) " "         "*"      "*"        " "   " "      " " " "
## 7  ( 1 ) " "         "*"      "*"        " "   " "      " " " "
## 8  ( 1 ) " "         "*"      "*"        " "   " "      "*" " "
##          S.F.Ratio perc.alumni Expend Grad.Rate
## 1  ( 1 ) " "       " "         " "    " "
## 2  ( 1 ) " "       " "         " "    " "
## 3  ( 1 ) " "       " "         "*"    " "
## 4  ( 1 ) " "       " "         "*"    " "
## 5  ( 1 ) " "       " "         "*"    " "
## 6  ( 1 ) " "       " "         "*"    " "
## 7  ( 1 ) " "       " "         "*"    " "
## 8  ( 1 ) " "       " "         "*"    " "
```

```r
regFitFull <- regsubsets(Apps ~ ., data = df, nvmax = 19)
regFitFullSummary <- summary(regFitFull)
names(regFitFullSummary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```r
regFitFullSummary$rsq
```

```
##  [1] 0.8900990 0.9157839 0.9183356 0.9212640 0.9237599 0.9247464 0.9257649
##  [8] 0.9268725 0.9276780 0.9283103 0.9288011 0.9289945 0.9291223 0.9291632
## [15] 0.9291878 0.9291885 0.9291887
```

```r
regFitFullSummary$adjr2
```

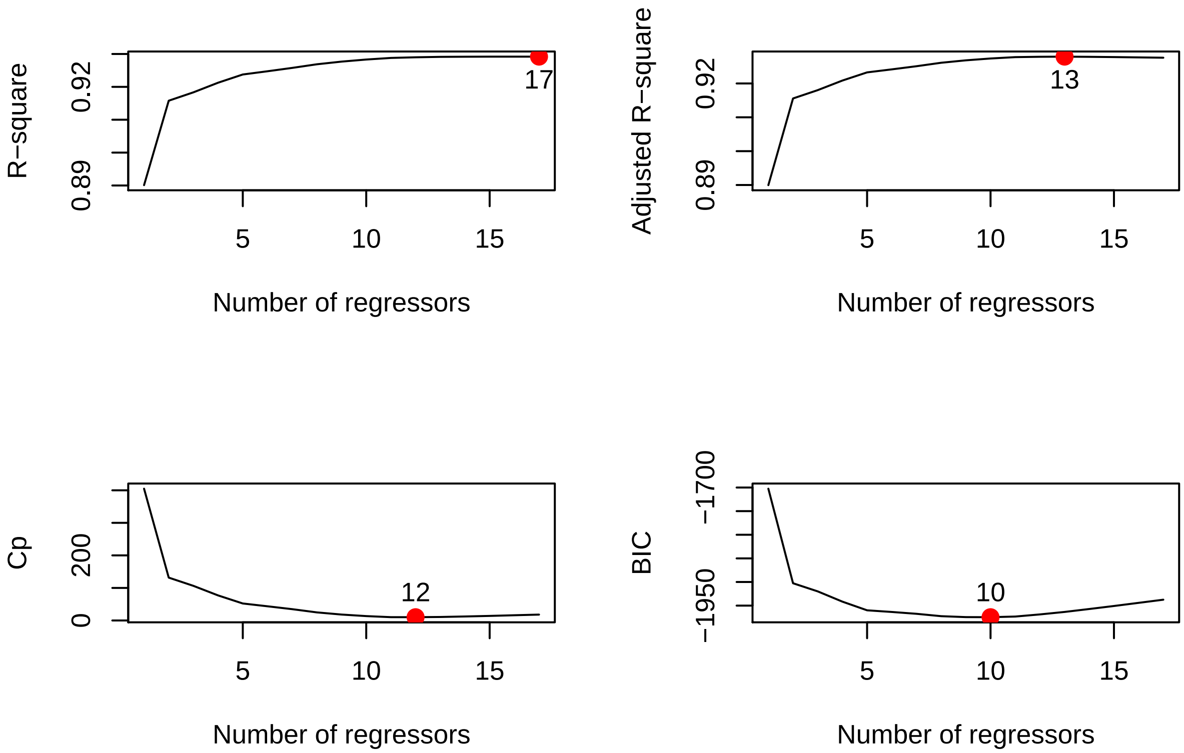
```
##  [1] 0.8899572 0.9155663 0.9180186 0.9208560 0.9232655 0.9241600 0.9250892
##  [8] 0.9261108 0.9268294 0.9273744 0.9277773 0.9278792 0.9279147 0.9278617
## [15] 0.9277921 0.9276978 0.9276027
```

```r
regFitFullSummary$rss
```

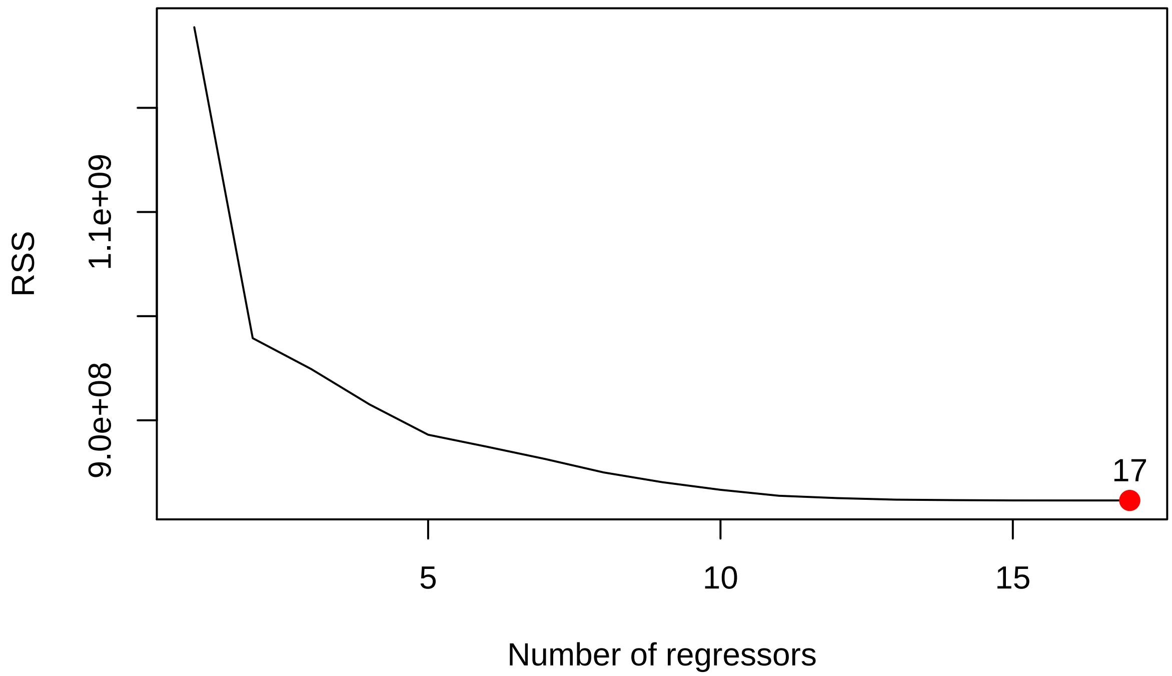
```
##  [1] 1277410811  978867162  949208869  915171254  886160591  874694084
##  [7]  862855633  849981358  840619277  833270183  827565469  825317242
## [13]  823831288  823356478  823069994  823062005  823059948
```

```r
par(mfrow = c(2, 2))
plot(
  regFitFullSummary$rsq,
  xlab = "Number of regressors",
  ylab = "R-square",
  type = "l"
)
a <- which.max(regFitFullSummary$rsq)
points(
  a,
  regFitFullSummary$rsq[a],
  col = "red",
  cex = 2,
  pch = 20
)
text(a, regFitFullSummary$rsq[a], labels = a, pos = 1)

plot(
  regFitFullSummary$adjr2,
  xlab = "Number of regressors",
  ylab = "Adjusted R-square",
  type = "l"
)
a1 <- which.max(regFitFullSummary$adjr2)
points(
  a1,
  regFitFullSummary$adjr2[a1],
  col = "red",
  cex = 2,
  pch = 20
)
text(a1,
     regFitFullSummary$adjr2[a1],
     labels = a1,
     pos = 1)

plot(regFitFullSummary$cp,
     xlab = "Number of regressors",
     ylab = "Cp",
     type = "l")
a2 <- which.min(regFitFullSummary$cp)
points(
  a2,
  regFitFullSummary$cp[a2],
  col = "red",
  cex = 2,
  pch = 20
)
text(a2, regFitFullSummary$cp[a2], labels = a2, pos = 3)

plot(
  regFitFullSummary$bic,
  xlab = "Number of regressors",
```

```
  ylab = "BIC",
  type = "l"
)
a3 <- which.min(regFitFullSummary$bic)
points(
  a3,
  regFitFullSummary$bic[a3],
  col = "red",
  cex = 2,
  pch = 20
)
text(a3, regFitFullSummary$bic[a3], labels = a3, pos = 3)
```




```
par(mfrow = c(1, 1))
plot(
  regFitFullSummary$rss,
  xlab = "Number of regressors",
  ylab = "RSS",
  type = "l"
)
a4 <- which.min(regFitFullSummary$rss)
points(
  a4,
  regFitFullSummary$rss[a4],
  col = "red",
  cex = 2,
  pch = 20
)
text(a4, regFitFullSummary$rss[a4], labels = a4, pos = 3)
```




```r
par(mfrow = c(2, 2))
plot(regFitFull, scale = "r2")
plot(regFitFull, scale = "adjr2")
plot(regFitFull, scale = "Cp")
plot(regFitFull, scale = "bic")

coef(regFitFull, 12)
```

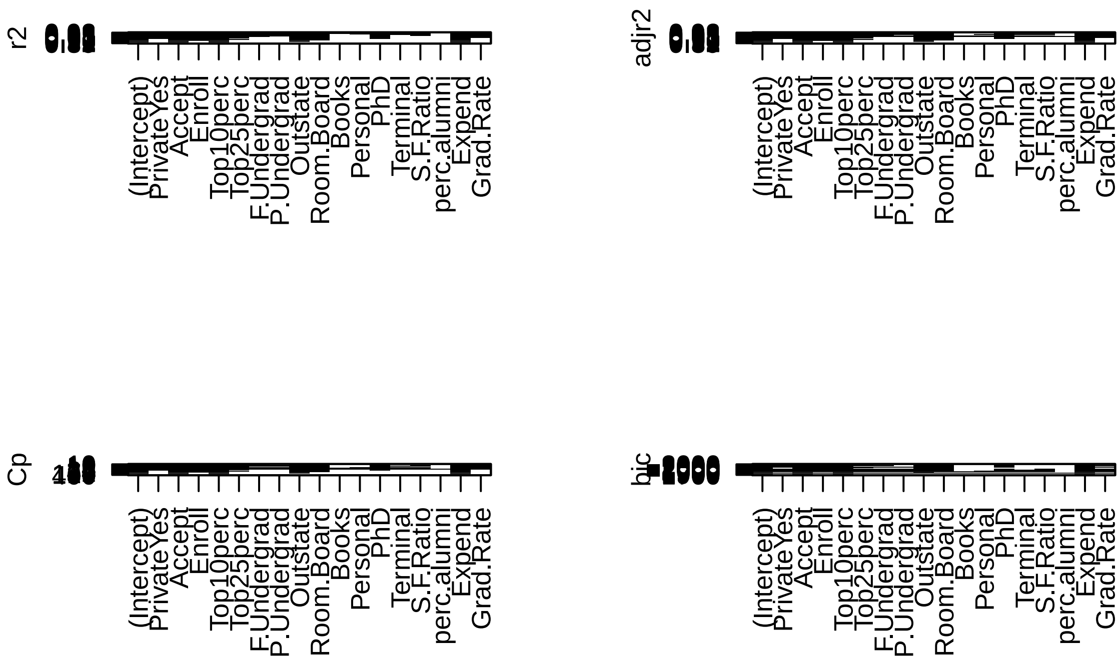
```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc
## -157.28685883 -511.78760196    1.58691470   -0.88265385   50.41131660
##     Top25perc    F.Undergrad   P.Undergrad      Outstate    Room.Board
##  -14.74735373     0.05945481    0.04593068   -0.09017643    0.14776586
##           PhD        Expend     Grad.Rate
##  -10.70502848     0.07246655    8.63961002
```

```r
#paste(names(coef(regFitFull, 12))[2:length(coef(regFitFull, 12))], collapse='+')

# Selecting the info from the 12th model
regFit <-
  lm(
    Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad +
      P.Undergrad + Outstate + Room.Board + PhD + Expend + Grad.Rate,
    data = dfTrain
  )
regPred <- predict(regFit, xTest)

subMSEP <- mean((regPred - yTest) ^ 2)

### Shrinkage Method: Ridge
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```




```r
xTemp <- model.matrix(Apps ~ ., df)

head(xTemp)
```

```
##                              (Intercept) PrivateYes Accept Enroll
## Abilene Christian University           1          1   1232    721
## Adelphi University                     1          1   1924    512
## Adrian College                         1          1   1097    336
## Agnes Scott College                    1          1    349    137
## Alaska Pacific University              1          1    146     55
## Albertson College                      1          1    479    158
##                              Top10perc Top25perc F.Undergrad P.Undergrad
## Abilene Christian University        23        52        2885         537
## Adelphi University                  16        29        2683        1227
## Adrian College                      22        50        1036          99
## Agnes Scott College                 60        89         510          63
## Alaska Pacific University           16        44         249         869
## Albertson College                   38        62         678          41
##                              Outstate Room.Board Books Personal PhD
## Abilene Christian University     7440       3300   450     2200  70
## Adelphi University              12280       6450   750     1500  29
## Adrian College                  11250       3750   400     1165  53
## Agnes Scott College             12960       5450   450      875  92
## Alaska Pacific University        7560       4120   800     1500  76
## Albertson College               13500       3335   500      675  67
##                              Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University       78      18.1          12   7041
## Adelphi University                 30      12.2          16  10527
## Adrian College                     66      12.9          30   8735
## Agnes Scott College                97       7.7          37  19016
## Alaska Pacific University          72      11.9           2  10922
## Albertson College                  73       9.4          11   9727
```

```
##                                 Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

```r
x <- xTemp[, -2]
y <- df$Apps

grid <- 10 ^ seq(10,-2, length = 100)
ridgeMod <- glmnet(x, y, alpha = 0, lambda = grid)

dim(coef(ridgeMod))
```

```
## [1]  18 100
```

```r
# Cross validation to choose lambda
train <- sample(1:nrow(x), round(nrow(x) / 2))
yTrain1 <- y[train]
xTrain1 <- x[train, ]
yTest1 <- y[-train]
xTest1 <- x[-train, ]

ridgeMod <-
  glmnet(xTrain1,
         yTrain1,
         alpha = 0,
         lambda = grid,
         thresh = 1e-12)
ridgePred <- predict(ridgeMod, s = 4, newx = xTest1)
ridgeMSEP <- mean((ridgePred - yTest) ^ 2)

ridgePred <-
  predict(
    ridgeMod,
    s = 0,
    newx = xTest1,
    exact = TRUE,
    x = xTrain1,
    y = yTrain1
  )
mean((ridgePred - yTest1) ^ 2)
```

```
## [1] 1733471
```

```r
cvOut <- cv.glmnet(xTrain1, yTrain1, alpha = 0)
plot(cvOut)
bestLam <- cvOut$lambda.min
bestLam
```

```
## [1] 380.8738
```

```r
ridgePred <- predict(ridgeMod, s = bestLam, newx = xTest1)
ridgeMSEP <- mean((ridgePred - yTest1) ^ 2)
```

```r
### Dimensional Reduction
library(pls)
```

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

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

```r
pcrFit <- pcr(Apps ~ .,
              data = df,
              scale = TRUE,
              validation = "CV")
summary(pcrFit)
```

```
## Data:    X dimension: 777 17
##  Y dimension: 777 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            3873     3840     2024     2036     1707     1583     1581
## adjCV         3873     3840     2022     2038     1623     1577     1578
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1569     1543     1496      1493      1496      1497      1503
## adjCV     1570     1539     1493      1490      1494      1494      1501
##        14 comps  15 comps  16 comps  17 comps
## CV         1504      1443      1159      1125
## adjCV      1501      1425      1153      1119
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      31.670    57.30    64.30    69.90    75.39    80.38    83.99
## Apps    2.316    73.06    73.07    82.08    84.08    84.11    84.32
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.40    90.50     92.91     95.01     96.81      97.9     98.75
## Apps    85.18    85.88     86.06     86.06     86.10      86.1     86.13
##       15 comps  16 comps  17 comps
## X        99.36     99.84    100.00
## Apps     90.32     92.52     92.92
```

```r
validationplot(pcrFit, val.type = "MSEP")
pcrFit <- pcr(Apps ~ .,
              data = dfTrain,
              scale = TRUE,
              ncomp = 5)
pcrPred <- predict(pcrFit, xTest, ncomps = 5)
pcrMSEP <- mean((pcrPred - yTest) ^ 2)

pcrFit <- pcr(Apps ~ ., data = df, scale = TRUE)
summary(pcrFit)
```

```
## Data:    X dimension: 777 17
```

```
##  Y dimension: 777 1
## Fit method: svdpc
## Number of components considered: 17
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      31.670    57.30    64.30    69.90    75.39    80.38    83.99
## Apps    2.316    73.06    73.07    82.08    84.08    84.11    84.32
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.40    90.50     92.91     95.01     96.81      97.9     98.75
## Apps    85.18    85.88     86.06     86.06     86.10      86.1     86.13
##       15 comps  16 comps  17 comps
## X        99.36     99.84    100.00
## Apps     90.32     92.52     92.92
```

```r
plsFit <- plsr(Apps ~ .,
               data = df,
               scale = TRUE,
               validation = "CV")
summary(plsFit)
```

```
## Data:    X dimension: 777 17
##  Y dimension: 777 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            3873     1838     1533     1421     1305     1154     1139
## adjCV         3873     1837     1531     1418     1295     1139     1132
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1133     1127     1127      1126      1126      1126      1125
## adjCV     1127     1122     1122      1121      1121      1121      1120
##        14 comps  15 comps  16 comps  17 comps
## CV         1125      1125      1125      1125
## adjCV      1119      1119      1119      1119
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       25.76    40.33    62.59    64.97    66.87    71.33    75.39
## Apps    78.01    85.14    87.67    90.73    92.63    92.72    92.77
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       79.37    82.36     85.04     87.92     90.65     92.69     95.50
## Apps    92.82    92.87     92.89     92.90     92.91     92.92     92.92
##       15 comps  16 comps  17 comps
## X        96.87     98.65    100.00
## Apps     92.92     92.92     92.92
```

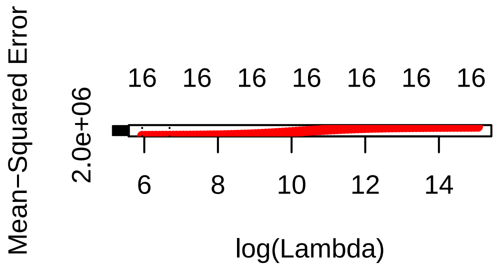
```r
validationplot(plsFit, val.type = "MSEP")

plsFit <- plsr(Apps ~ .,
               data = dfTrain,
               scale = TRUE,
               ncomp = 5)
plsPred <- predict(plsFit, xTest, ncomp = 5)
```

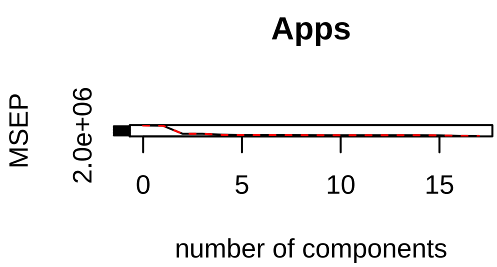
```r
plsMSEP <- mean((plsPred - yTest) ^ 2)

plsFit <- plsr(Apps ~ .,
               data = df,
               scale = TRUE,
               ncomp = 5)
summary(plsFit)
```

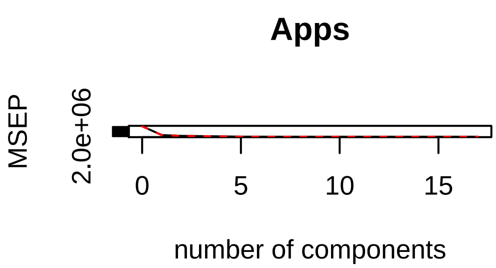
```
## Data:    X dimension: 777 17
##  Y dimension: 777 1
## Fit method: kernelpls
## Number of components considered: 5
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps
## X       25.76    40.33    62.59    64.97    66.87
## Apps    78.01    85.14    87.67    90.73    92.63
```

```r
msep <- list(subMSEP, ridgeMSEP, plsMSEP, pcrMSEP)
bestMethod <- which.min(msep)
bestMethod
```

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







Partial Least Squares and Principal Component Regression, 3, corresponds to best subset selection method