154Lab7

Jiyoon Clover Jeong

10/16/2017

```
library(ISLR)

library(pls)
```

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

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

```
library(glmnet)
```

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

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

```
## Loaded glmnet 2.0-13
```

```
library(DAAG)
```

```
## Loading required package: lattice
```

```
library(caret)
```

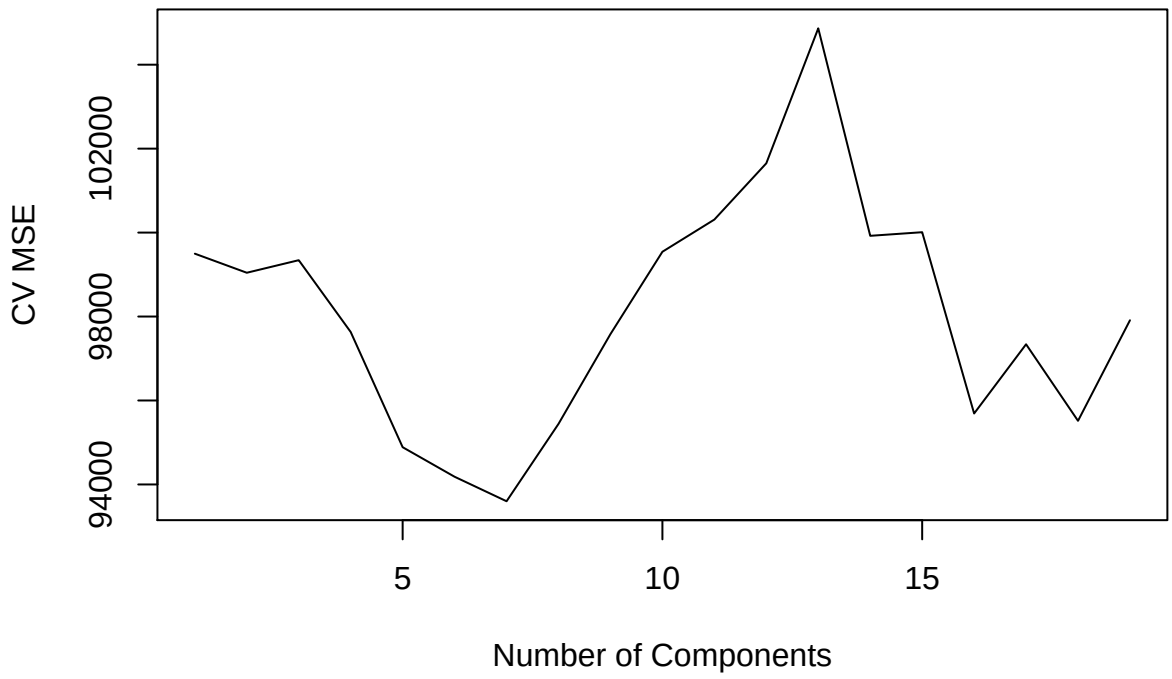
```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:pls':
##
##     R2
```

Cross-validation for pcr() and plsr()

```
n <- nrow(Hitters)
set.seed(100)
pcr_fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE,
validation = "CV", segments=10)


# Q pcr_fit$validation$PRESS[1, ]   vs pcr_fit$validation$PRESS

plot(pcr_fit$validation$PRESS[1, ] / n, type="l", main="PCR",
xlab="Number of Components", ylab="CV MSE")
```



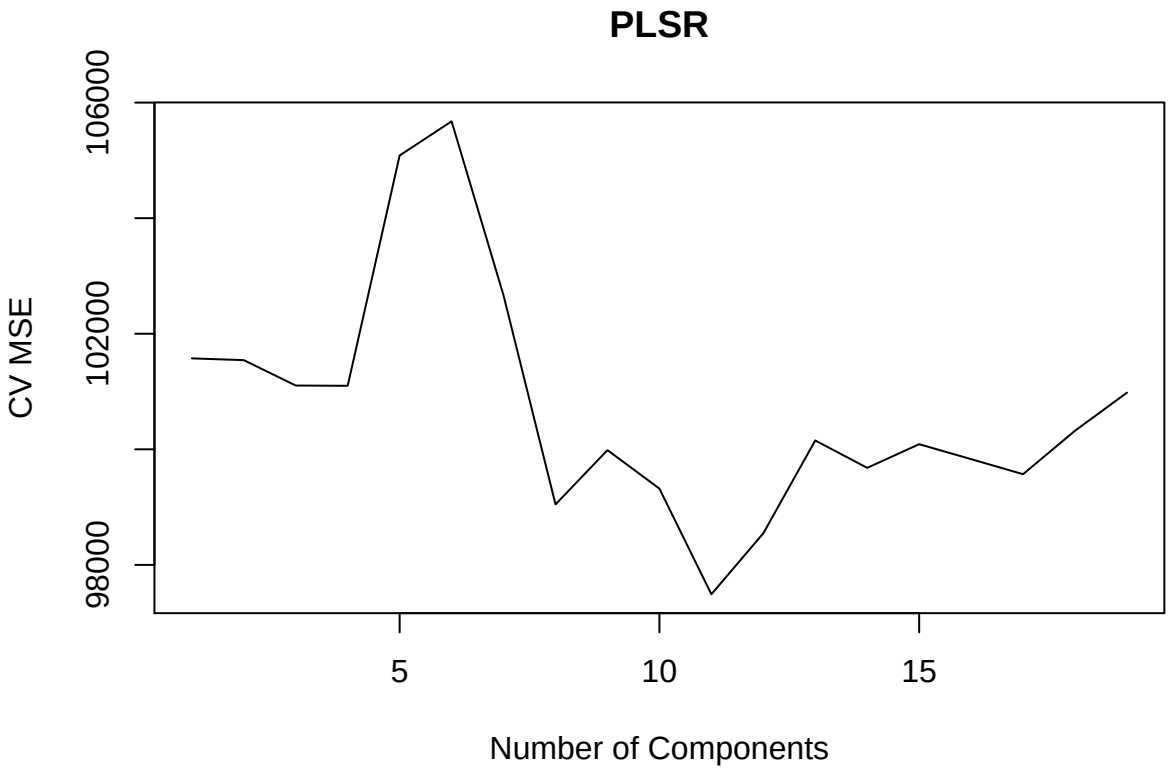

```
pcr_fit$validation$PRESS[1,]
```

```
##  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## 32038208 31891885 31988380 31437430 30553787 30327020 30139108 30732602
##  9 comps 10 comps 11 comps 12 comps 13 comps 14 comps 15 comps 16 comps
## 31421387 32052952 32299978 32731502 33766780 32175250 32202972 30812232
## 17 comps 18 comps 19 comps
## 31343373 30756073 31527451
```

```
# number of components
which.min(pcr_fit$validation$PRESS)
```

```
## [1] 7
```

```
set.seed(200)
plsr_fit <- plsr(Salary ~ ., data = Hitters, scale = TRUE,
validation = "CV", segments=10)
plot(plsr_fit$validation$PRESS[1, ] / n , type="l", main="PLSR",
xlab="Number of Components", ylab="CV MSE")
```




```
summary(plsr_fit)
```

```
## Data:    X dimension: 263 19
##  Y dimension: 263 1
## Fit method: kernelpls
## Number of components considered: 19
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV             452    352.6    352.6    351.8    351.8    358.7    359.7
## adjCV          452    352.1    351.4    350.5    350.5    356.4    356.9
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       354.5    348.2    349.9     348.7     345.5     347.4     350.2
## adjCV    352.1    346.0    347.6     346.6     343.5     345.2     347.7
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
## CV        349.3     350.1     349.6     349.2     350.5     351.6
## adjCV     346.9     347.5     347.1     346.7     347.9     349.0
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X         38.08    51.03    65.98    73.93    78.63    84.26    88.17
## Salary    43.05    46.40    47.72    48.71    50.53    51.66    52.34
##         8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X         90.12    92.92     95.00     96.68     97.68     98.22     98.55
## Salary    53.26    53.52     53.77     54.04     54.20     54.32     54.47
##         15 comps  16 comps  17 comps  18 comps  19 comps
## X          98.98     99.24     99.71     99.99    100.00
## Salary     54.54     54.59     54.61     54.61     54.61
```

```r
# number of components
which.min(plsr_fit$validation$PRESS)
```

```
## [1] 11
```

Cross-validation for ridge regression and lasso

Ridge : alpha = 0 Lasso : alpha = 1

```r
set.seed(300)
# code for ridge regression CV

names(Hitters)
```

```
##  [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"
##  [6] "Walks"     "Years"     "CAtBat"    "CHits"     "CHmRun"
## [11] "CRuns"     "CRBI"      "CWalks"    "League"    "Division"
## [16] "PutOuts"   "Assists"   "Errors"    "Salary"    "NewLeague"
```

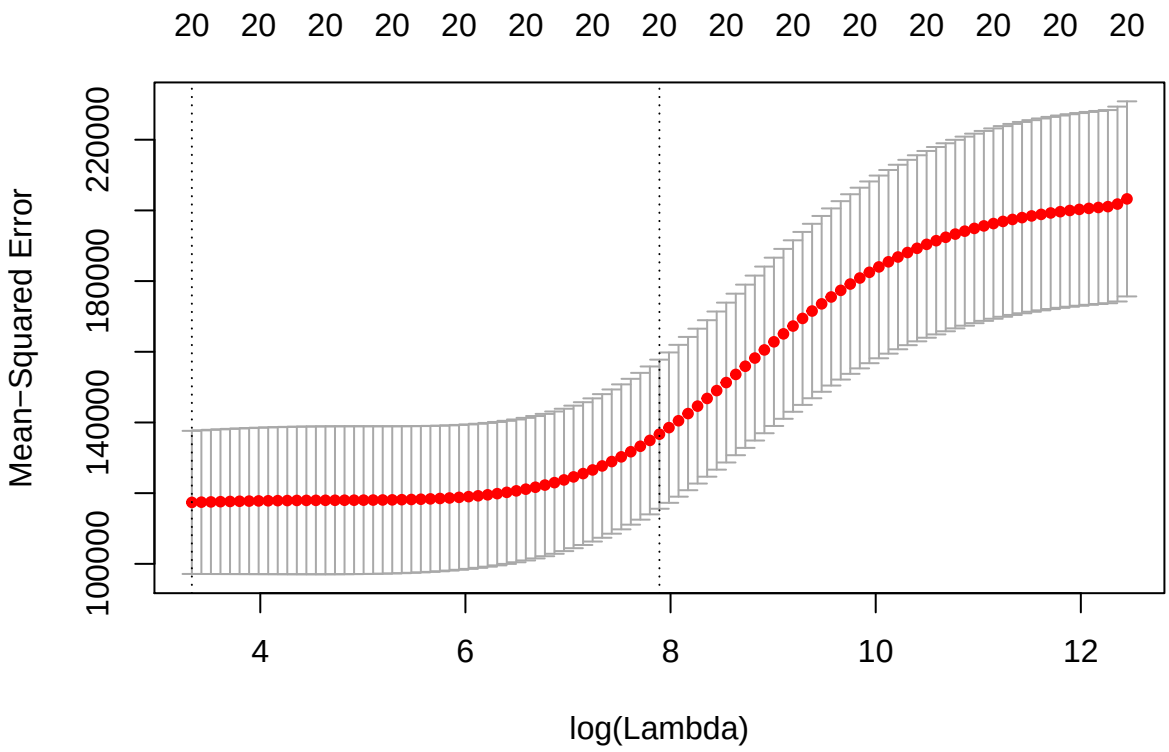
```r
data <- na.omit(Hitters)
class(data)
```

```
## [1] "data.frame"
```

```r
X <- model.matrix(Salary ~. -1,data)

X <- cbind(X, Salary = data$Salary)

ridgecv <- cv.glmnet(as.matrix(X[,-c(21)]), X[,21] , alpha = 0)

summary(ridgecv)
```

```
##            Length Class  Mode
## lambda     99     -none- numeric
## cvm        99     -none- numeric
## cvsd       99     -none- numeric
## cvup       99     -none- numeric
## cvlo       99     -none- numeric
## nzero      99     -none- numeric
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
```

```r
plot.cv.glmnet(ridgecv)
```




```
coef(ridgecv, s = "lambda.min")
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  1.041375e+02
## AtBat       -6.301512e-01
## Hits         2.642007e+00
## HmRun       -1.384250e+00
## Runs         1.049276e+00
## RBI          7.318299e-01
## Walks        3.276489e+00
## Years       -8.705697e+00
## CAtBat       1.136403e-04
## CHits        1.319492e-01
## CHmRun       6.898036e-01
## CRuns        2.831928e-01
## CRBI         2.512166e-01
## CWalks      -2.603598e-01
## LeagueA     -2.871264e+01
## LeagueN      2.874200e+01
## DivisionW   -1.223809e+02
## PutOuts      2.621883e-01
## Assists      1.628961e-01
## Errors      -3.669810e+00
## NewLeagueN  -2.108745e+01
```

```
summary(ridgecv)
```

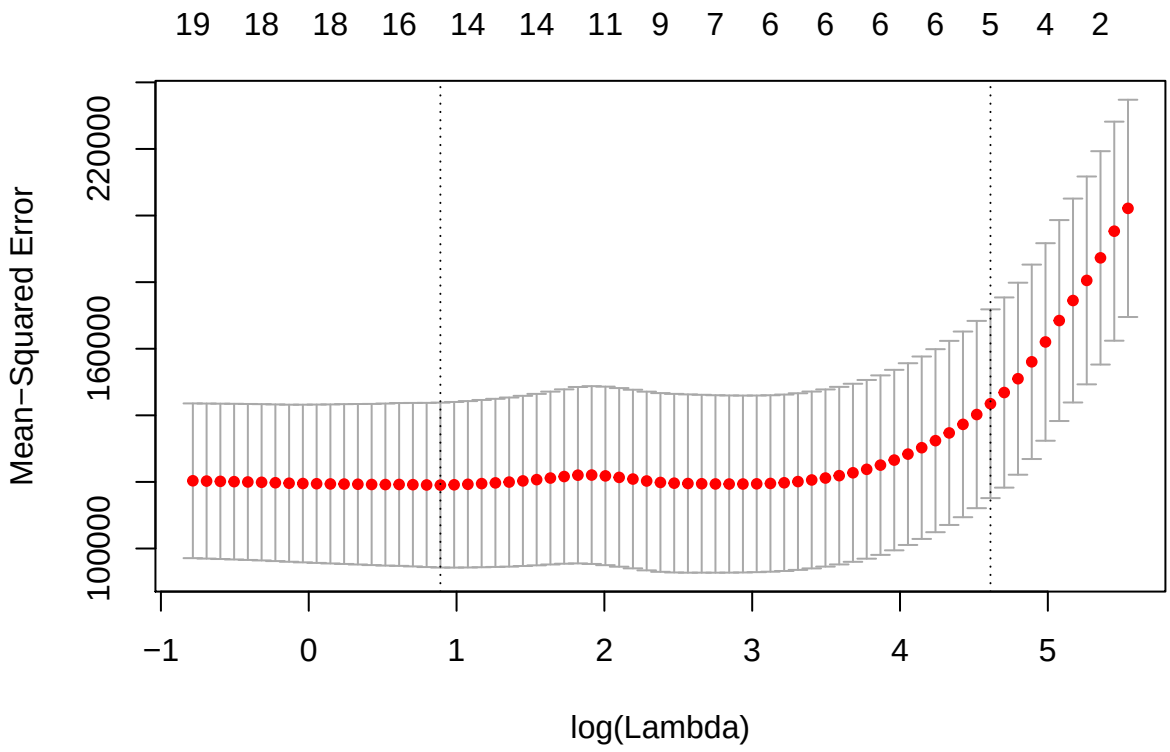
```
##            Length Class  Mode
## lambda     99     -none- numeric
## cvm        99     -none- numeric
```

```
## cvsd       99     -none- numeric
## cvup       99     -none- numeric
## cvlo       99     -none- numeric
## nzero      99     -none- numeric
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
```

```
set.seed(400)
# code for lasso CV

lassocv <- cv.glmnet(as.matrix(X[,-c(21)]), X[,21] , alpha = 1)


summary(lassocv)
```

```
##            Length Class  Mode
## lambda     69     -none- numeric
## cvm        69     -none- numeric
## cvsd       69     -none- numeric
## cvup       69     -none- numeric
## cvlo       69     -none- numeric
## nzero      69     -none- numeric
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
```

```
plot.cv.glmnet(lassocv)
```



```r
coef(lassocv, s = "lambda.min")
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept)  1.618802e+02
## AtBat       -1.613015e+00
## Hits         5.805892e+00
## HmRun        .
## Runs         .
## RBI          .
## Walks        4.846934e+00
## Years       -9.972405e+00
## CAtBat       .
## CHits        .
## CHmRun       5.374550e-01
## CRuns        6.811938e-01
## CRBI         3.903563e-01
## CWalks      -5.560144e-01
## LeagueA     -3.246461e+01
## LeagueN      4.381163e-14
## DivisionW   -1.193481e+02
## PutOuts      2.741895e-01
## Assists      1.855978e-01
## Errors      -2.165084e+00
## NewLeagueN   .
```

Nested Cross Validation

```r
set.seed(250)


# Q : why this is not MSE of pcr?
# msep_pcr <- MSEP(pcr_fit)
# msep_pcr$val[1,1,]



folds <- createFolds(data[,19], 10)   # return indices
head(folds)
```

```
## $Fold01
##  [1]   6  13  44  46  52  55  62  90  93  98 127 136 141 152 160 163 170
## [18] 176 185 211 226 239 240 242 248 251
##
## $Fold02
##  [1]  20  31  33  39  41  42  72  74  77  82  89  94 101 105 114 135 139
## [18] 142 144 159 194 203 229 256 260
##
## $Fold03
##  [1]  12  18  35  38  43  57  60  64  70  75 100 130 132 133 137 143 148
## [18] 149 162 165 166 195 200 210 220 227 247
##
```

```
## $Fold04
##  [1]   1  17  23  29  34  40  73  83  95  96 104 111 113 120 129 131 145
## [18] 151 158 174 180 199 225 233 243 246
##
## $Fold05
##  [1]  11  49  53  61  63  65  81  88  91  92 102 118 121 138 146 147 150
## [18] 156 171 173 175 179 181 187 201 224 231
##
## $Fold06
##  [1]   2  10  16  22  27  28  30  36  47  54  67  68  99 103 134 153 164
## [18] 172 206 216 222 228 235 238 253 263
```

```r
# 10 folds
# For hyperparameter tuning, use a 10-fold CV --> automatically taken care of by the function glmnet
ols_mse <- c(0)
pcr_mse <- c(0)
plsr_mse <- c(0)
ridge_mse <- c(0)
lasso_mse <- c(0)

for(i in 1:10){


# Q : why warning???

     # olsfit <- lm(Salary ~., data = as.data.frame(X[-folds[[i]],]))
     # ols_mse[i] <- mean((predict(olsfit, as.data.frame(X[folds[[i]],])) - X[folds[[i]], 21 ])^2)



    # ols predict --> data = data.frame
    olsfit <- lm(Salary ~., data = data[-folds[[i]],])
    ols_mse[i] <- mean((  predict(olsfit, data[folds[[i]], -19]) - data[folds[[i]], 19 ])^2)


    # pcr predict <- data = matrix
    pcrfit <- pcr(Salary~., data = as.data.frame(X[-folds[[i]],]), validation ="CV", segments = 10)
    pcr_mse[i] <-  mean((predict(pcrfit, X[folds[[i]], -21], s = "lambda.min") - X[folds[[i]], "Salary"]


    # plsr
     plsrfit <- plsr(Salary~., data = as.data.frame(X[-folds[[i]],]), validation ="CV", segments = 10)
    plsr_mse[i] <-  mean((predict(plsrfit, X[folds[[i]], -21], s = "lambda.min") - X[folds[[i]], "Salary


    # lasso

    lassofit <- cv.glmnet(X[-folds[[i]], -21], X[-folds[[i]], 21],
                          alpha = 1, nfolds = 10 )
    lasso_mse[i] <- mean( (predict( lassofit, X[folds[[i]], -21], s = "lambda.min") - X[folds[[i]], 21]


    # ridge
```

```
    ridgefit <- cv.glmnet(X[-folds[[i]], -21], X[-folds[[i]], 21],
                          alpha = 0, nfolds = 10)
    ridge_mse[i] <- mean( (predict( ridgefit, X[folds[[i]], -21], s = "lambda.min") - X[folds[[i]], 21])
}

table <- rbind(ols_mse, pcr_mse, plsr_mse, lasso_mse, ridge_mse)

print(table)
```

```
##               [,1]       [,2]     [,3]     [,4]     [,5]     [,6]
## ols_mse   111059.8   94356.39 83960.04 78616.09 335390.1 49638.02
## pcr_mse   134176.3 1926536.72 83565.63 74740.64 348338.8 59379.62
## plsr_mse  122718.9  105905.34 81821.97 76215.85 344374.9 51612.67
## lasso_mse 168794.4  143220.71 75549.40 69592.10 343334.7 46771.65
## ridge_mse 173601.9  161917.56 69749.45 65519.44 342240.5 49303.96
##               [,7]     [,8]     [,9]     [,10]
## ols_mse   120958.26 89276.17 156843.0 146479.25
## pcr_mse   122702.50 90898.47 169627.7 160125.59
## plsr_mse  117906.64 89366.81 147588.0 149954.04
## lasso_mse  92814.48 69860.71 147905.8 130720.01
## ridge_mse  86921.10 81100.58 134632.8  90822.83
```

```
MSE_means <- rowMeans(table)

MSE_means
```

```
##   ols_mse   pcr_mse  plsr_mse lasso_mse ridge_mse
##  126657.7  317009.2  128746.5  128856.4  125581.0
```

```
cat(names(which.min(MSE_means)), "is the best model\n")
```

```
## ridge_mse is the best model
```