

4주차 과제

11기 명재성

September 3, 2020

```
rm(list=ls())
```

6.5 Lab 1: Subset Selection Methods

6.5.1 Best Subset Selection

```
library(ISLR)
fix(Hitters)
names(Hitters)
```

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

```
dim(Hitters)
```

```
## [1] 322 20
```

```
sum(is.na(Hitters$Salary))
```

```
## [1] 59
```

```
Hitters = na.omit(Hitters)
dim(Hitters)
```

```
## [1] 263 20
```

```
sum(is.na(Hitters))
```

```
## [1] 0
```

```
library(leaps)
regfit.full = regsubsets(Salary~., Hitters)
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., Hitters)
## 19 Variables (and intercept)
##              Forced in Forced out
## AtBat          FALSE      FALSE
## Hits           FALSE      FALSE
## HmRun          FALSE      FALSE
## Runs           FALSE      FALSE
## RBI            FALSE      FALSE
## Walks          FALSE      FALSE
## Years          FALSE      FALSE
## CAtBat         FALSE      FALSE
## CHits          FALSE      FALSE
## CHmRun         FALSE      FALSE
## CRuns          FALSE      FALSE
## CRBI           FALSE      FALSE
## CWalks         FALSE      FALSE
## LeagueN        FALSE      FALSE
## DivisionW      FALSE      FALSE
## PutOuts        FALSE      FALSE
## Assists        FALSE      FALSE
## Errors         FALSE      FALSE
## NewLeagueN     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##              AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " "
## 4 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " "
## 5 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " " "
## 7 ( 1 ) " " "*" " " " " " " "*" " " " "*" "*" " " " " " "
## 8 ( 1 ) "*" "*" " " " " " " "*" " " " " "*" "*" " " " " " "
##              CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) " " " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " "*" " " " " "
```

```
## 4 ( 1 ) " " " " "*" "*" " " " " " "
## 5 ( 1 ) " " " " "*" "*" " " " " " "
## 6 ( 1 ) " " " " "*" "*" " " " " " "
## 7 ( 1 ) " " " " "*" "*" " " " " " "
## 8 ( 1 ) "*" " " "*" "*" " " " " " "
```

```
regfit.full = regsubsets(Salary~., data=Hitters, nvmax=19)
reg.summary = summary(regfit.full)
```

```
names(reg.summary)
```

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

```
reg.summary$rsq
```

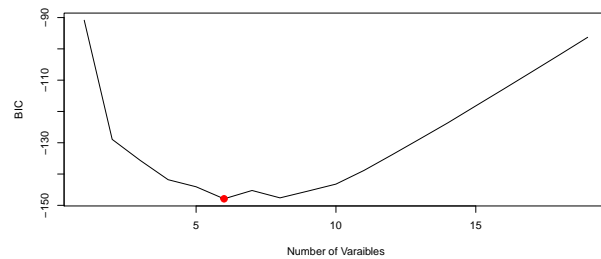
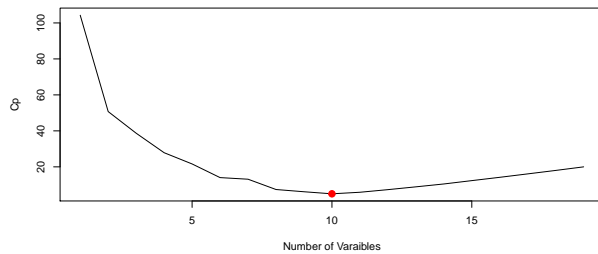
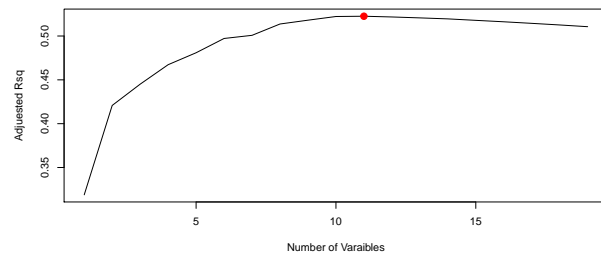
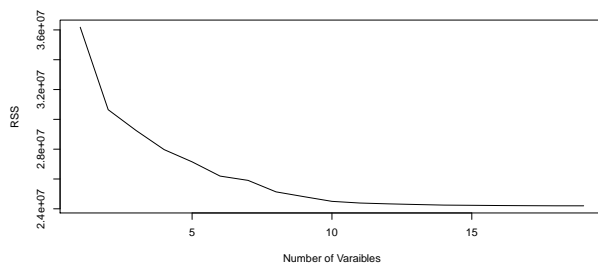
```
## [1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227
## [8] 0.5285569 0.5346124 0.5404950 0.5426153 0.5436302 0.5444570 0.5452164
## [15] 0.5454692 0.5457656 0.5459518 0.5460945 0.5461159
```

```
par(mfrow=c(2,2))
plot(reg.summary$rss, xlab="Number of Variables", ylab="RSS", type="l")

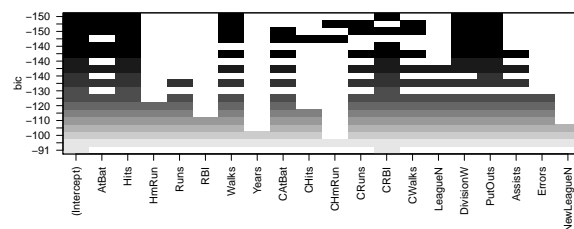
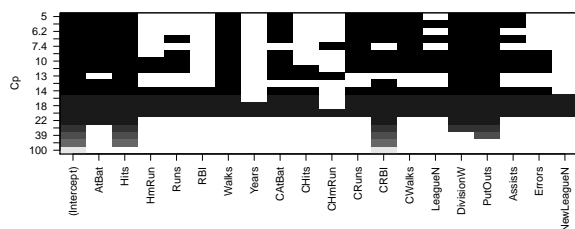
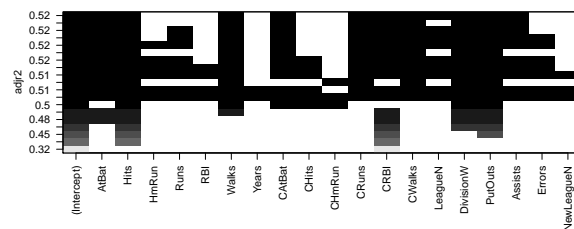
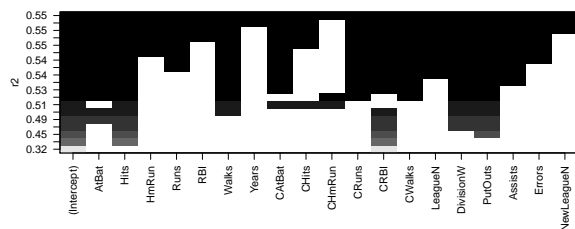
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="Adjusted Rsq", type="l")
# which.max(reg.summary$adjr2)
# [1] 11
points(11, reg.summary$adjr2[11], col="red", cex=2, pch=20)

plot(reg.summary$cp, xlab="Number of Variables", ylab="Cp", type="l")
# which.min(reg.summary$cp)
# [1] 10
points(10, reg.summary$cp[10], col="red", cex=2, pch=20)

plot(reg.summary$bic, xlab="Number of Variables", ylab="BIC", type="l")
# which.min(reg.summary$bic)
# [1] 6
points(6, reg.summary$bic[6], col="red", cex=2, pch=20)
```



```
par(mfrow=c(2,2))
plot(regfit.full, scale="r2")
plot(regfit.full, scale="adjr2")
plot(regfit.full, scale="Cp")
plot(regfit.full, scale="bic")
```



```
coef(regfit.full, 6)
```

```
## (Intercept)      AtBat      Hits      Walks      CRBI      DivisionW
```

```
## 91.5117981 -1.8685892 7.6043976 3.6976468 0.6430169 -122.9515338
## PutOuts
## 0.2643076
```

6.5.2 Forward and Backward Stepwise Selection

```
regfit.fwd = regsubsets(Salary~., data=Hitters, nvmax=19, method="forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "forward")
## 19 Variables (and intercept)
##           Forced in Forced out
## AtBat      FALSE      FALSE
## Hits       FALSE      FALSE
## HmRun       FALSE      FALSE
## Runs        FALSE      FALSE
## RBI         FALSE      FALSE
## Walks       FALSE      FALSE
## Years       FALSE      FALSE
## CAtBat      FALSE      FALSE
## CHits       FALSE      FALSE
## CHmRun      FALSE      FALSE
## CRuns       FALSE      FALSE
## CRBI        FALSE      FALSE
## CWalks      FALSE      FALSE
## LeagueN     FALSE      FALSE
## DivisionW   FALSE      FALSE
## PutOuts     FALSE      FALSE
## Assists     FALSE      FALSE
## Errors      FALSE      FALSE
## NewLeagueN  FALSE      FALSE
## 1 subsets of each size up to 19
## Selection Algorithm: forward
##           AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 4 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 5 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 7 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 8 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 9 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 10 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 11 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 12 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 13 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 14 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 15 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 16 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 17 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 18 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 19 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
```

```

## 11 ( 1 ) "*" "*" " " " " " " "*" " " "*" " " " " "*" "*"
## 12 ( 1 ) "*" "*" " " "*" " " "*" " " "*" " " " " "*" "*"
## 13 ( 1 ) "*" "*" " " "*" " " "*" " " "*" " " " " "*" "*"
## 14 ( 1 ) "*" "*" "*" "*" " " "*" " " "*" " " " " "*" "*"
## 15 ( 1 ) "*" "*" "*" "*" " " "*" " " "*" "*" " " "*" "*"
## 16 ( 1 ) "*" "*" "*" "*" "*" "*" " " "*" "*" " " "*" "*"
## 17 ( 1 ) "*" "*" "*" "*" "*" "*" " " "*" "*" " " "*" "*"
## 18 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" " " "*" "*"
## 19 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
##           CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " "*" " " " " "
## 4 ( 1 ) " " " " "*" "*" " " " " " "
## 5 ( 1 ) " " " " "*" "*" " " " " " "
## 6 ( 1 ) " " " " "*" "*" " " " " " "
## 7 ( 1 ) "*" " " "*" "*" " " " " " "
## 8 ( 1 ) "*" " " "*" "*" " " " " " "
## 9 ( 1 ) "*" " " "*" "*" " " " " " "
## 10 ( 1 ) "*" " " "*" "*" "*" " " " " "
## 11 ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 12 ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 13 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 14 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 15 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 16 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 17 ( 1 ) "*" "*" "*" "*" "*" "*" "*" " " "
## 18 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" " "
## 19 ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" " "

```

```

regfit.bwd = regsubsets(Salary~., data=Hitters, nvmax=19, method="backward")
summary(regfit.bwd)

```

```

## Subset selection object
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "backward")
## 19 Variables (and intercept)
##           Forced in Forced out
## AtBat      FALSE      FALSE
## Hits       FALSE      FALSE
## HmRun       FALSE      FALSE
## Runs       FALSE      FALSE
## RBI        FALSE      FALSE
## Walks      FALSE      FALSE
## Years      FALSE      FALSE
## CAtBat     FALSE      FALSE
## CHits      FALSE      FALSE
## CHmRun     FALSE      FALSE

```

```

## CRuns          FALSE      FALSE
## CRBI           FALSE      FALSE
## CWalks         FALSE      FALSE
## LeagueN        FALSE      FALSE
## DivisionW      FALSE      FALSE
## PutOuts        FALSE      FALSE
## Assists        FALSE      FALSE
## Errors         FALSE      FALSE
## NewLeagueN     FALSE      FALSE
## 1 subsets of each size up to 19
## Selection Algorithm: backward
##
##      AtBat Hits HmRun Runs RBI Walks Years CatBat CHits CHmRun CRuns CRBI
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " "
## 3 ( 1 ) " " "*" " " " " " " " " " " " " " " " " " " " " "
## 4 ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " "
## 5 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " " "
## 7 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " " "
## 8 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " "*"
## 9 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " "*"
## 10 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " "*"
## 11 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " "*"
## 12 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " "*"
## 13 ( 1 ) "*" "*" " " " " " " "*" " " " " " " " " " " " " "*"
## 14 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " " " " " " "*"
## 15 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " " " " " " "*"
## 16 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " " " " " " "*"
## 17 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " " " " " " "*"
## 18 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " " " " " " "*"
## 19 ( 1 ) "*" "*" "*" " " " " " "*" " " " " " " " " " " " " "*"
##
##      CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
## 1 ( 1 ) " " " " " " " " " " " " " "
## 2 ( 1 ) " " " " " " " " " " " " " "
## 3 ( 1 ) " " " " " " "*" " " " " " "
## 4 ( 1 ) " " " " " " "*" " " " " " "
## 5 ( 1 ) " " " " " " "*" " " " " " "
## 6 ( 1 ) " " " " "*" "*" " " " " " "
## 7 ( 1 ) "*" " " "*" "*" " " " " " "
## 8 ( 1 ) "*" " " "*" "*" " " " " " "
## 9 ( 1 ) "*" " " "*" "*" " " " " " "
## 10 ( 1 ) "*" " " "*" "*" "*" " " " " "
## 11 ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 12 ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 13 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 14 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 15 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "
## 16 ( 1 ) "*" "*" "*" "*" "*" "*" " " " "

```

```
## 17 ( 1 ) "*"      "*"      "*"      "*"      "*"      "*"      "*"
## 18 ( 1 ) "*"      "*"      "*"      "*"      "*"      "*"      "*"
## 19 ( 1 ) "*"      "*"      "*"      "*"      "*"      "*"      "*"

```

6.5.3 Choosing Among Models Using the Validation Set Approach and Cross-Validation

```
set.seed(1)
train = sample(c(TRUE,FALSE), nrow(Hitters), rep=TRUE)
test=(!train)

```

```
regfit.best = regsubsets(Salary~., data=Hitters[train, ], nvmax=19)
test.mat = model.matrix(Salary~., data=Hitters[test, ])

```

```
val.errors = rep(NA, 19)
for(i in 1:19){
  coefi = coef(regfit.best, id=i)
  pred = test.mat[, names(coefi)] %*% coefi
  val.errors[i] = mean((Hitters$Salary[test] - pred)^2)
}

```

```
val.errors

```

```
## [1] 164377.3 144405.5 152175.7 145198.4 137902.1 139175.7 126849.0 136191.4
## [9] 132889.6 135434.9 136963.3 140694.9 140690.9 141951.2 141508.2 142164.4
## [17] 141767.4 142339.6 142238.2

```

```
which.min(val.errors)

```

```
## [1] 7

```

```
coef(regfit.best, 7)

```

```
## (Intercept)      AtBat      Hits      Walks      CRuns      CWalks
## 67.1085369 -2.1462987 7.0149547 8.0716640 1.2425113 -0.8337844
## DivisionW      PutOuts
## -118.4364998 0.2526925

```

```
predict.regsubsets = function(object, newdata, id, ...){
  form = as.formula(object$call[[2]])
  mat = model.matrix(form, newdata)
  coefi = coef(object, id=id)
  xvars = names(coefi)
  mat[, xvars] %*% coefi
}

```



```
regfit.best = regsubsets(Salary~., data=Hitters, nvmax=19)
coef(regfit.best, 7)
```

```
## (Intercept)      Hits      Walks      CAtBat      CHits      CHmRun
## 79.4509472    1.2833513    3.2274264   -0.3752350    1.4957073    1.4420538
## DivisionW      PutOuts
## -129.9866432    0.2366813
```

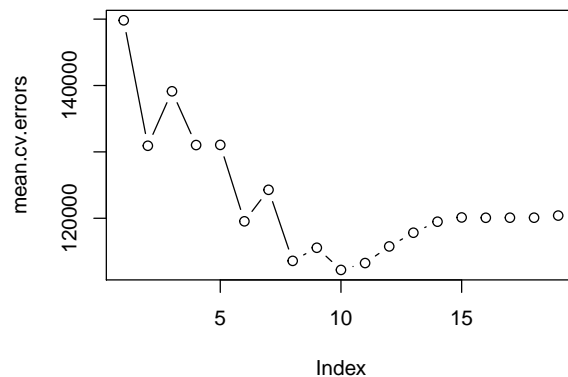
```
k = 10
set.seed(1)
folds = sample(1:k, nrow(Hitters), rep=TRUE)
cv.errors = matrix(NA, k, 19, dimnames=list(NULL, paste(1:19)))
```

```
for(j in 1:k){
  best.fit = regsubsets(Salary~., data=Hitters[folds != j, ], nvmax=19)
  for (i in 1:19){
    pred = predict.regsubsets(best.fit, Hitters[folds == j, ], id=i)
    # pred = predict(best.fit, Hitters[folds == j, ], id=i) ???
    cv.errors[j,i] = mean((Hitters$Salary[folds == j] - pred)^2)
  }
}
```

```
mean.cv.errors = apply(cv.errors, 2, mean)
mean.cv.errors
```

```
##      1      2      3      4      5      6      7      8
## 149821.1 130922.0 139127.0 131028.8 131050.2 119538.6 124286.1 113580.0
##      9     10     11     12     13     14     15     16
## 115556.5 112216.7 113251.2 115755.9 117820.8 119481.2 120121.6 120074.3
##     17     18     19
## 120084.8 120085.8 120403.5
```

```
plot(mean.cv.errors, type="b")
```



```
reg.best = regsubsets(Salary~., data=Hitters, nvmax=19)
coef(reg.best, 10)
```

```
## (Intercept)      AtBat      Hits      Walks      CAtBat      CRuns
## 162.5354420   -2.1686501   6.9180175   5.7732246  -0.1300798   1.4082490
##          CRBI      CWalks  DivisionW      PutOuts      Assists
##    0.7743122   -0.8308264 -112.3800575   0.2973726   0.2831680
```

6.6 Lab 2: Ridge Regression and the Lasso

```
x = model.matrix(Salary~., Hitters)[, -1]
y = Hitters$Salary
```

6.6.1 Ridge Regression

```
library(glmnet)
grid = 10^seq(10, -2, length=100)
ridge.mod = glmnet(x, y, alpha=0, lambda=grid)
```

```
dim(coef(ridge.mod))
```

```
## [1] 20 100
```

```
ridge.mod$lambda[50]
```

```
## [1] 11497.57
```

```
coef(ridge.mod)[, 50]
```

```
##      (Intercept)      AtBat      Hits      HmRun      Runs
## 407.356050200    0.036957182    0.138180344    0.524629976    0.230701523
##      RBI      Walks      Years      CAtBat      CHits
## 0.239841459    0.289618741    1.107702929    0.003131815    0.011653637
##      CHmRun      CRuns      CRBI      CWalks      LeagueN
## 0.087545670    0.023379882    0.024138320    0.025015421    0.085028114
##      DivisionW      PutOuts      Assists      Errors      NewLeagueN
## -6.215440973    0.016482577    0.002612988    -0.020502690    0.301433531
```

```
sqrt(sum(coef(ridge.mod)[-1, 50]^2))
```

```
## [1] 6.360612
```

```
ridge.mod$lambda[60]
```

```
## [1] 705.4802
```

```
coef(ridge.mod)[, 60]
```

```
##      (Intercept)      AtBat      Hits      HmRun      Runs      RBI
## 54.32519950    0.11211115    0.65622409    1.17980910    0.93769713    0.84718546
##      Walks      Years      CAtBat      CHits      CHmRun      CRuns
## 1.31987948    2.59640425    0.01083413    0.04674557    0.33777318    0.09355528
##      CRBI      CWalks      LeagueN      DivisionW      PutOuts      Assists
## 0.09780402    0.07189612    13.68370191    -54.65877750    0.11852289    0.01606037
##      Errors      NewLeagueN
## -0.70358655    8.61181213
```

```
sqrt(sum(coef(ridge.mod)[-1, 60]^2))
```

```
## [1] 57.11001
```

```
predict(ridge.mod, s=50, type="coefficients")[1:20, ]
```

```
##      (Intercept)      AtBat      Hits      HmRun      Runs
## 4.876610e+01 -3.580999e-01    1.969359e+00 -1.278248e+00    1.145892e+00
##      RBI      Walks      Years      CAtBat      CHits
## 8.038292e-01    2.716186e+00 -6.218319e+00    5.447837e-03    1.064895e-01
##      CHmRun      CRuns      CRBI      CWalks      LeagueN
## 6.244860e-01    2.214985e-01    2.186914e-01 -1.500245e-01    4.592589e+01
##      DivisionW      PutOuts      Assists      Errors      NewLeagueN
## -1.182011e+02    2.502322e-01    1.215665e-01 -3.278600e+00 -9.496680e+00
```

```

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

ridge.mod = glmnet(x[train, ], y[train], alpha=0, lambda=grid, thres=1e-12)
ridge.pred = predict(ridge.mod, s=4, newx=x[test, ])
mean((ridge.pred - y.test)^2)

```

```
## [1] 142199.2
```

```
mean((mean(y[train]) - y.test)^2)
```

```
## [1] 224669.9
```

```

ridge.pred = predict(ridge.mod, s=1e10, newx=x[test, ])
mean((ridge.pred - y.test)^2)

```

```
## [1] 224669.8
```

```

ridge.pred = predict(ridge.mod, s=0, newx=x[test, ])
mean((ridge.pred - y.test)^2)

```

```
## [1] 167789.8
```

```
lm(y~x, subset=train)
```

```

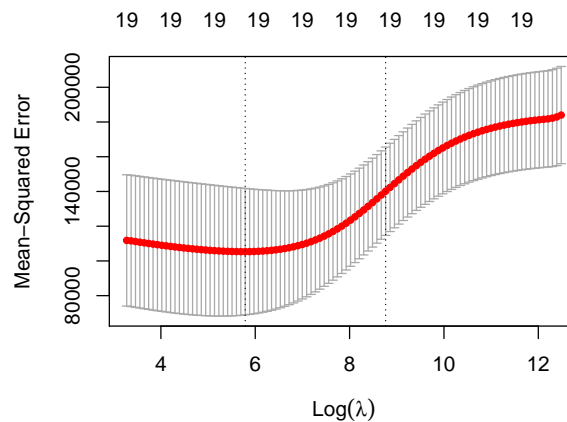
##
## Call:
## lm(formula = y ~ x, subset = train)
##
## Coefficients:
## (Intercept)      xAtBat      xHits      xHmRun      xRuns      xRBI
##    274.0145    -0.3521    -1.6377     5.8145     1.5424     1.1243
##      xWalks      xYears    xCAtBat    xCHits    xCHmRun    xCRuns
##     3.7287   -16.3773    -0.6412     3.1632     3.4008    -0.9739
##      xCRBI      xCWalks    xLeagueN  xDivisionW    xPutOuts    xAssists
##    -0.6005     0.3379    119.1486   -144.0831     0.1976     0.6804
##      xErrors  xNewLeagueN
##    -4.7128   -71.0951

```

```
predict(ridge.mod, s=0, type="coefficients")[1:20, ]
```

```
## (Intercept)      AtBat      Hits      HmRun      Runs      RBI
## 274.2089049 -0.3699455 -1.5370022  5.9129307  1.4811980  1.0772844
##      Walks      Years      CAtBat      CHits      CHmRun      CRuns
##  3.7577989 -16.5600387 -0.6313336  3.1115575  3.3297885 -0.9496641
##      CRBI      CWalks      LeagueN      DivisionW      PutOuts      Assists
## -0.5694414  0.3300136 118.4000592 -144.2867510  0.1971770  0.6775088
##      Errors      NewLeagueN
## -4.6833775 -70.1616132
```

```
set.seed(1)
cv.out = cv.glmnet(x[train, ], y[train], alpha=0)
plot(cv.out)
```



```
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 326.0828
```

```
ridge.pred = predict(ridge.mod, s=bestlam, newx=x[test, ])
mean((ridge.pred - y.test)^2)
```

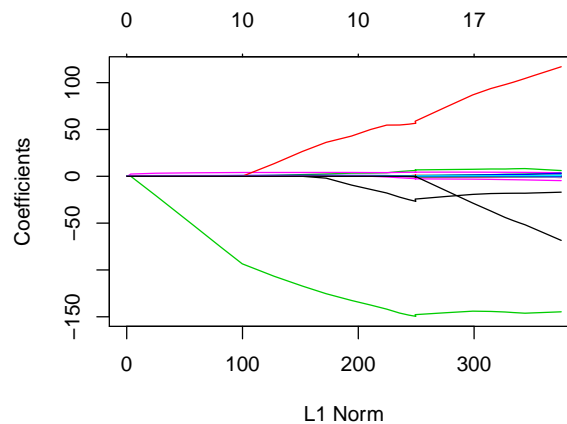
```
## [1] 139856.6
```

```
out = glmnet(x, y, alpha=0)
predict(out, type="coefficients", s=bestlam)[1:20, ]
```

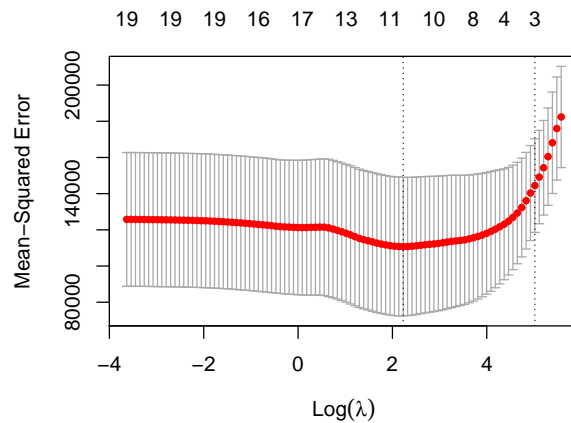
```
## (Intercept)      AtBat      Hits      HmRun      Runs      RBI
## 15.44383135  0.07715547  0.85911581  0.60103107  1.06369007  0.87936105
##      Walks      Years      CAtBat      CHits      CHmRun      CRuns
##  1.62444616  1.35254780  0.01134999  0.05746654  0.40680157  0.11456224
##      CRBI      CWalks      LeagueN      DivisionW      PutOuts      Assists
##  0.12116504  0.05299202  22.09143189 -79.04032637  0.16619903  0.02941950
##      Errors      NewLeagueN
## -1.36092945  9.12487767
```

6.6.2 The Lasso

```
lasso.mod = glmnet(x[train, ], y[train], alpha=1, lambda=grid)
plot(lasso.mod)
```



```
set.seed(1)
cv.out = cv.glmnet(x[train, ], y[train], alpha=1)
plot(cv.out)
```



```
bestlam = cv.out$lambda.min
lasso.pred = predict(lasso.mod, s=bestlam, newx=x[test, ])
mean((lasso.pred - y.test)^2)
```

```
## [1] 143673.6
```

```
out = glmnet(x, y, alpha=1, lambda=grid)
lasso.coef = predict(out, type="coefficients", s=bestlam)[1:20, ]
lasso.coef
```

```
##      (Intercept)      AtBat      Hits      HmRun      Runs
##      1.27479059    -0.05497143    2.18034583    0.00000000    0.00000000
##           RBI      Walks      Years      CAtBat      CHits
##      0.00000000    2.29192406   -0.33806109    0.00000000    0.00000000
##      CHmRun      CRuns      CRBI      CWalks      LeagueN
##      0.02825013    0.21628385    0.41712537    0.00000000    20.28615023
##      DivisionW      PutOuts      Assists      Errors      NewLeagueN
##     -116.16755870    0.23752385    0.00000000   -0.85629148    0.00000000
```

```
lasso.coef[lasso.coef != 0]
```

```
##      (Intercept)      AtBat      Hits      Walks      Years
##      1.27479059    -0.05497143    2.18034583    2.29192406   -0.33806109
##      CHmRun      CRuns      CRBI      LeagueN      DivisionW
##      0.02825013    0.21628385    0.41712537    20.28615023   -116.16755870
##      PutOuts      Errors
##      0.23752385   -0.85629148
```