DATA 303/473 Assignment 3 Solution

2022-03-29

Q1

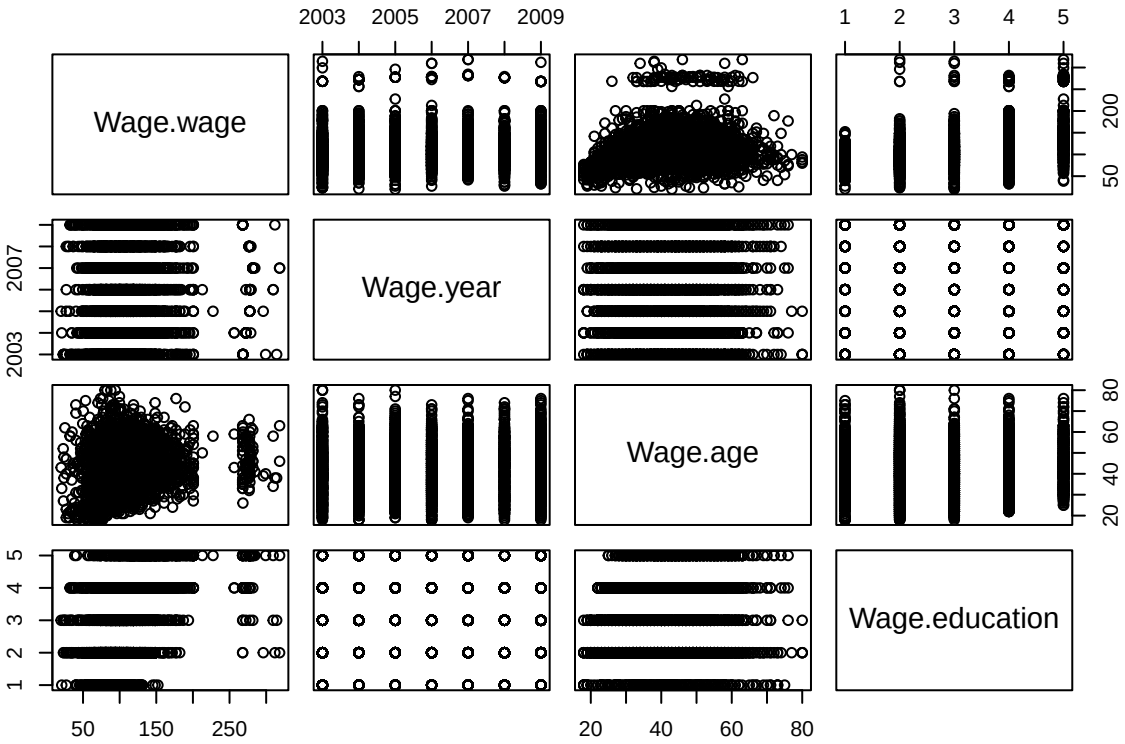
We use `Wage` data set which is in the library `ISLR2`. The `Wage` data set contains the following variables.

```
library(ISLR2)
#head(Wage)
summary(Wage)
```

```
##       year           age                      maritl           race
##  Min.   :2003   Min.   :18.00   1. Never Married: 648   1. White:2480
##  1st Qu.:2004   1st Qu.:33.75   2. Married      :2074   2. Black: 293
##  Median :2006   Median :42.00   3. Widowed      :  19   3. Asian: 190
##  Mean   :2006   Mean   :42.41   4. Divorced     : 204   4. Other:  37
##  3rd Qu.:2008   3rd Qu.:51.00   5. Separated    :  55
##  Max.   :2009   Max.   :80.00
##
##                education                     region               jobclass
##  1. < HS Grad      :268   2. Middle Atlantic   :3000   1. Industrial :1544
##  2. HS Grad        :971   1. New England       :   0   2. Information:1456
##  3. Some College   :650   3. East North Central:   0
##  4. College Grad   :685   4. West North Central:   0
##  5. Advanced Degree:426   5. South Atlantic    :   0
##                           6. East South Central:   0
##                           (Other)              :   0
##             health      health_ins      logwage           wage
##  1. <=Good     : 858   1. Yes:2083   Min.   :3.000   Min.   : 20.09
##  2. >=Very Good:2142   2. No : 917   1st Qu.:4.447   1st Qu.: 85.38
##                                      Median :4.653   Median :104.92
##                                      Mean   :4.654   Mean   :111.70
##                                      3rd Qu.:4.857   3rd Qu.:128.68
##                                      Max.   :5.763   Max.   :318.34
##
```

In the first part of the assignment. We are interested in `wage` in relation to `year`, `age` and `education`. This is a paired plot.

```
pairs(data.frame(Wage$wage, Wage$year, Wage$age, Wage$education))
```



It is known that `year` has approximately linear trend and the variable `education` is a categorical variable. We use the natural spline curve fitting for the trend of `age`. For this we use function `ns()` in the `splines` package and `lm()` function. We fit the following models

$$\begin{aligned}
\text{model1:} &\quad \text{waga} \sim \text{year} + ns(\text{age}, df = 1) + \text{education},\\
\text{model2:} &\quad \text{waga} \sim \text{year} + ns(\text{age}, df = 3) + \text{education},\\
\text{model3:} &\quad \text{waga} \sim \text{year} + ns(\text{age}, df = 5) + \text{education},\\
\text{model4:} &\quad \text{waga} \sim \text{year} + ns(\text{age}, df = 7) + \text{education},\\
\text{model5:} &\quad \text{waga} \sim \text{year} + ns(\text{age}, df = 9) + \text{education}.
\end{aligned}$$

(a) **(10 marks)** Fit the model and use `anova()` function to do the deviance test to compare the models. Choose the best model.

```
library(splines)
m1 <- lm(wage ~ year + ns(age, df=1) + education, data=Wage)
m2 <- lm(wage ~ year + ns(age, df=3) + education, data=Wage)
m3 <- lm(wage ~ year + ns(age, df=5) + education, data=Wage)
m4 <- lm(wage ~ year + ns(age, df=7) + education, data=Wage)
m5 <- lm(wage ~ year + ns(age, df=9) + education, data=Wage)
anova(m1,m2,m3,m4,m5)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ year + ns(age, df = 1) + education
## Model 2: wage ~ year + ns(age, df = 3) + education
## Model 3: wage ~ year + ns(age, df = 5) + education
## Model 4: wage ~ year + ns(age, df = 7) + education
## Model 5: wage ~ year + ns(age, df = 9) + education
##   Res.Df     RSS Df Sum of Sq       F Pr(>F)
## 1   2993 3854286
## 2   2991 3699770  2    154516 62.5205 <2e-16 ***
## 3   2989 3694885  2      4885  1.9765 0.1387
```

```
## 4   2987 3692452  2      2433  0.9845 0.3737
## 5   2985 3688635  2      3817  1.5443 0.2136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Increasing `df` for `age` from " 1 to 3" improved the fit significantly. However, the increase "3 to 5", "5 to 7" and "7 to 9" did not improve the fit significantly. From this the model with `df=3` (`model 2`) is the best model.

(b) **(5 marks)** Calculate `AIC` for each model fitted in (a). Choose the best model using the value of `AIC`.

```
AIC(m1,m2,m3,m4,m5)
```

```
##    df      AIC
## m1  8 30004.62
## m2 10 29885.87
## m3 12 29885.91
## m4 14 29887.93
## m5 16 29888.83
```

The `model 2` has the smallest `AIC`. `model 2` is the best model. Since `model 2` and `model 3` have similar `AIC` value, `model 3` is also a good model to consider.

(c) **(10 marks)** Split the data set (100%) into a training set (70%) and a test set (30%). Then fit model1–model5 on the training set, and calculate the test MSE for each model. Choose the best model.

```
set.seed(11)
train = sample(1:dim(Wage)[1], dim(Wage)[1]*0.7)
test <- -train
Wage.train <- Wage[train, ]
Wage.test <- Wage[test, ]

dim(Wage.train)
```

```
## [1] 2100   11
```

```
dim(Wage.test)
```

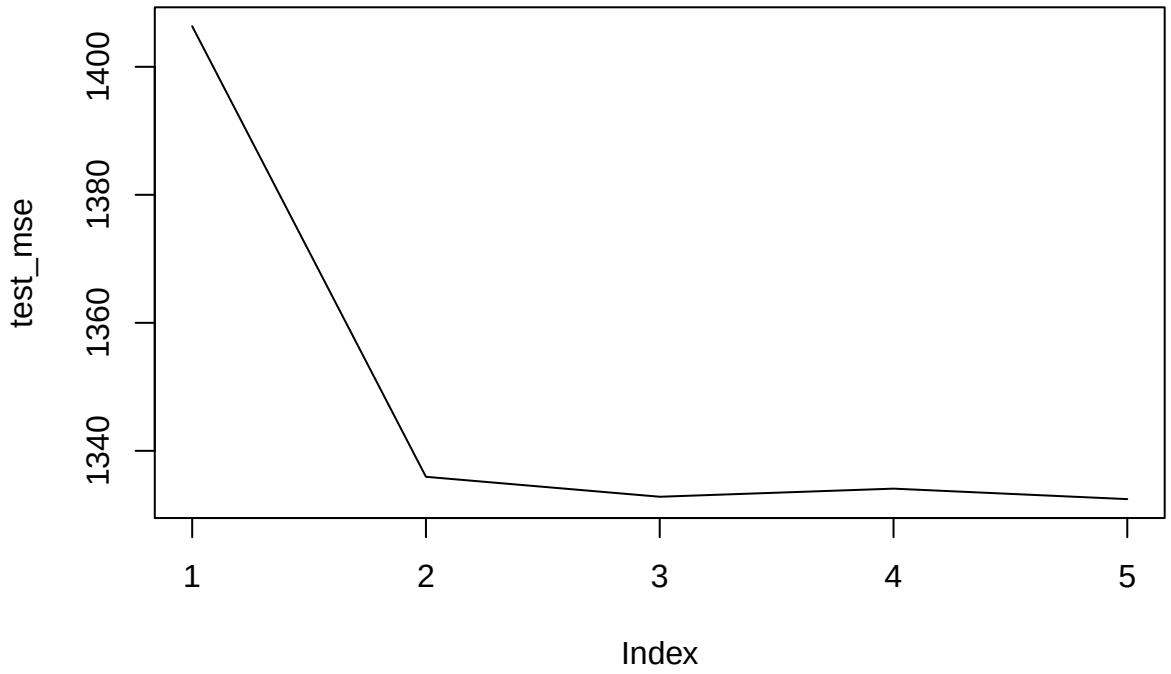
```
## [1] 900  11
```

```
m1 <- lm(wage ~ year + ns(age, df=1) + education, data=Wage.train)
m2 <- lm(wage ~ year + ns(age, df=3) + education, data=Wage.train)
m3 <- lm(wage ~ year + ns(age, df=5) + education, data=Wage.train)
m4 <- lm(wage ~ year + ns(age, df=7) + education, data=Wage.train)
m5 <- lm(wage ~ year + ns(age, df=9) + education, data=Wage.train)
yhat1 <- predict(m1, Wage.test)
yhat2 <- predict(m2, Wage.test)
yhat3 <- predict(m3, Wage.test)
yhat4 <- predict(m4, Wage.test)
yhat5 <- predict(m5, Wage.test)
mse1 <- mean((yhat1 - Wage.test$wage)^2)
mse2 <- mean((yhat2 - Wage.test$wage)^2)
mse3 <- mean((yhat3 - Wage.test$wage)^2)
mse4 <- mean((yhat4 - Wage.test$wage)^2)
mse5 <- mean((yhat5 - Wage.test$wage)^2)

test_mse <- c(mse1,mse2,mse3,mse4,mse5)
test_mse
```

```
## [1] 1406.339 1335.942 1332.825 1334.092 1332.456
```

```
plot(test_mse, type="l")
```



The `model5` has the smallest test MSE. Since `model2` and `model3` are simpler and have similar test MSE, we choose `model2` or `model3`. (We could apply `1se` rule here.)

(d) **(10 marks)** By combining the result from (a), (b) and (c), decide the best model. Refit the chosen model using all of the `Wage` data set. Interpret the out of the `summary()` function.

```
m2 <- lm(wage ~ year + ns(age, df=3) + education, data=Wage)
summary(m2)
```

```
##
## Call:
## lm(formula = wage ~ year + ns(age, df = 3) + education, data = Wage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.258  -19.694   -3.259   14.259  213.400
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2322.8434   637.0840  -3.646 0.000271 ***
## year                           1.1815     0.3176   3.720 0.000203 ***
## ns(age, df = 3)1              30.4808     2.9799  10.229  < 2e-16 ***
## ns(age, df = 3)2              74.5353     8.0524   9.256  < 2e-16 ***
## ns(age, df = 3)3               4.1361     6.3388   0.653 0.514124
## education2. HS Grad           10.9180     2.4282   4.496 7.18e-06 ***
## education3. Some College      23.4279     2.5550   9.170  < 2e-16 ***
## education4. College Grad      38.0297     2.5394  14.976  < 2e-16 ***
## education5. Advanced Degree   62.4889     2.7566  22.669  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 35.17 on 2991 degrees of freedom
## Multiple R-squared:  0.2915, Adjusted R-squared:  0.2896
## F-statistic: 153.8 on 8 and 2991 DF,  p-value: < 2.2e-16
```

```
m2a <- lm(wage ~ year + ns(age, df=2) + education, data=Wage)
AIC(m2a, m2)
```

```
##     df      AIC
## m2a  9 29890.47
## m2  10 29885.87
```

All of the variables `year`, `age` and `education` influence the variable `Wage` significantly. One of the coefficient of the 3rd degree natural spline for `age` is non-significant. We fit the model with the 2nd degree natural spline for `age` keeping other variables are the same. The AIC indicate the model with the 3rd degree natural spline for `age` is the better fit. We keep the `model 2` as the best model.

Q2

Here we will predict the number of applications received `Apps` using the other variables in the "College" data set.

The data set contains 777 observations on the following 18 variables.

```
# Private: A factor with levels No and Yes indicating private or public university
# Apps: Number of applications received
# Accept: Number of applications accepted
# Enroll: Number of new students enrolled
# Top10perc: Pct. new students from top 10% of H.S. class
# Top25perc: Pct. new students from top 25% of H.S. class
# F.Undergrad: Number of fulltime undergraduates
# P.Undergrad: Number of parttime undergraduates
# Outstate: Out-of-state tuition
# Room.Board: Room and board costs
# Books: Estimated book costs
# Personal: Estimated personal spending
# PhD: Pct. of faculty with Ph.D.'s
# Terminal: Pct. of faculty with terminal degree
# S.F.Ratio: Student/faculty ratio
# perc.alumni: Pct. alumni who donate
# Expend: Instructional expenditure per student
# Grad.Rate: Graduation rate
```

```
library(ISLR)
```

```
##
## Attaching package: 'ISLR'
```

```
## The following objects are masked from 'package:ISLR2':
##
##     Auto, Credit
```

```
data(College)
summary(College)
```

```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
```

```
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio      perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

(a) **(5 marks)** (Create trainig set and test set) Split the data set (100%) into a training set (70%) and a test set (30%).

```
set.seed(11)
train = sample(1:dim(College)[1], dim(College)[1]*0.7)
test <- -train
College.train <- College[train, ]
College.test <- College[test, ]

dim(College.train)
```

```
## [1] 543  18
```

```
dim(College.test)
```

```
## [1] 234  18
```

(b) **(10 marks)** (LASSO) Fit a lasso model on the training set, with λ chosen by cross-validation with the `1 se rule` . Report the test error obtained, along with the of non-zero coefficient estimates.

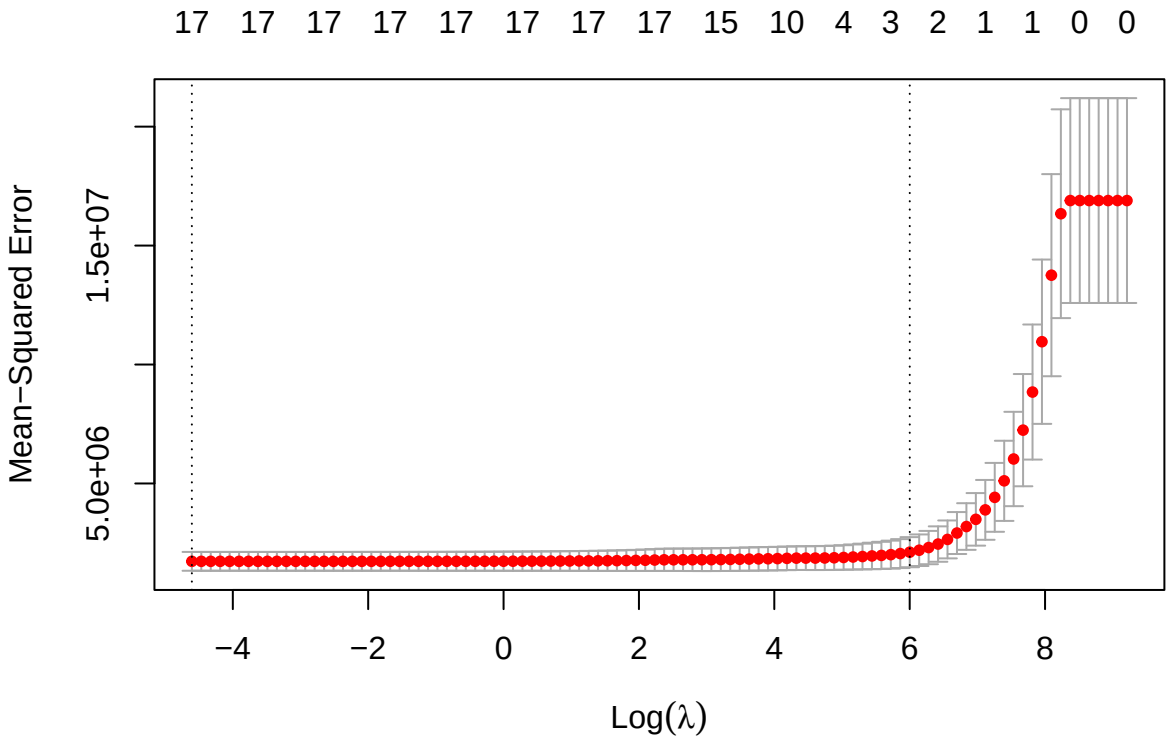
```
library(glmnet)
```

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

```
## Loaded glmnet 4.1-3
```

```
train.mat <- model.matrix(Apps ~ ., data = College.train)
test.mat <- model.matrix(Apps ~ ., data = College.test)
grid <- 10 ^ seq(4, -2, length = 100)

fit.lasso <- glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
cv.lasso <- cv.glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
plot(cv.lasso)
```



- lambda.1se

```
lam1se.lasso <- cv.lasso$lambda.1se
log(lam1se.lasso)
```

```
## [1] 6.000676
```

- Test MSE

```
pred.lasso <- predict(fit.lasso, s = lam1se.lasso, newx = test.mat)
mean((pred.lasso - College.test$Apps)^2)
```

```
## [1] 543036.7
```

- Non-zero coefficient estimates

```
predict(fit.lasso, s = lam1se.lasso, type = "coefficients")
```

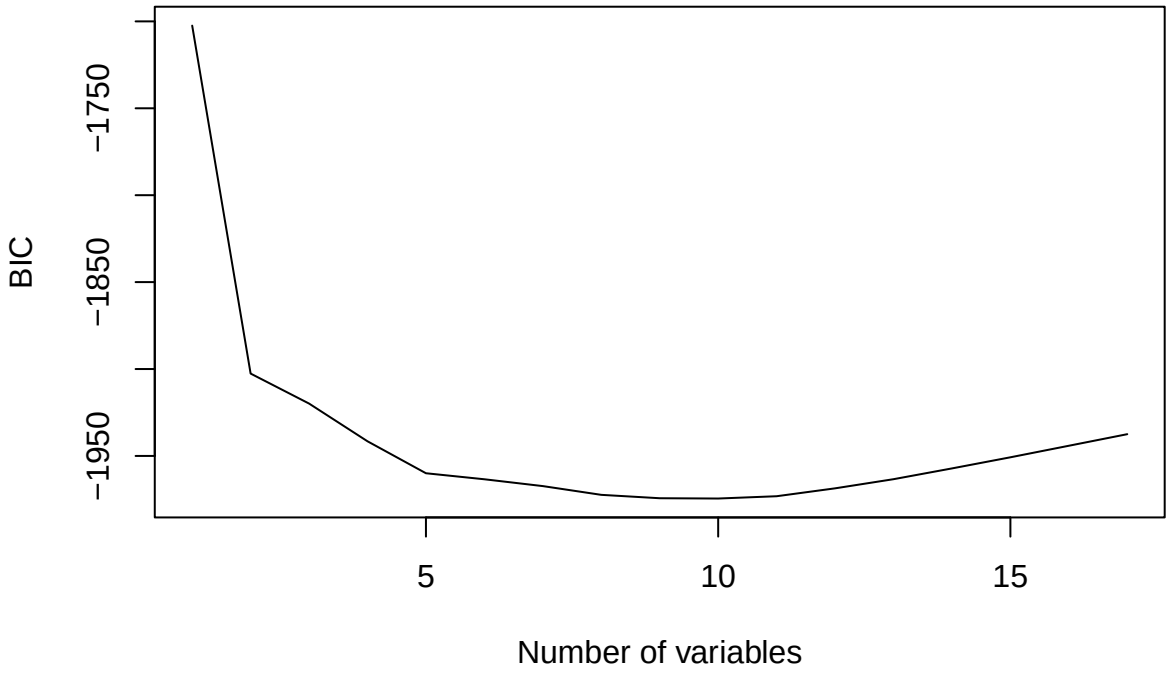
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -215.232306
## (Intercept)    .
## PrivateYes     .
## Accept         1.318401
## Enroll         .
```

```
## Top10perc     21.637700
## Top25perc      .
## F.Undergrad    .
## P.Undergrad    .
## Outstate       .
## Room.Board     .
## Books          .
## Personal       .
## PhD            .
## Terminal       .
## S.F.Ratio      .
## perc.alumni    .
## Expend         .
## Grad.Rate      .
```

(c) **(10 marks)** Do the best subset selection with `BIC` and choose the best model.

```
library(leaps)
m_bestsub <- regsubsets(Apps ~ ., College, nvmax=20)
m_bestsub_summary <- summary(m_bestsub)
```

```
plot(m_bestsub_summary$bic, xlab="Number of variables", ylab="BIC", type="l")
```




```
which.min(m_bestsub_summary$bic)
```

```
## [1] 10
```

```
coef(m_bestsub, 10)
```

```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc
## -100.51668243 -575.07060789    1.58421887   -0.56220848   49.13908916
##     Top25perc      Outstate    Room.Board           PhD        Expend
##  -13.86531103   -0.09466457    0.16373674  -10.01608705    0.07273776
##     Grad.Rate
##    7.33268904
```

(d) **(10 marks)** Use all of the `College` data set, refit the models chosen by LASSO in (b) and best subset selection in (c). Print output of the function `summary()` for these models. Then compute 'AIC' and 'BIC'. Between these 2 models, which model is the better model. Give reasons why.

```
m_lasso <- lm(Apps ~ Accept + Top10perc, data = College)
summary(m_lasso)
```

```
##
## Call:
## lm(formula = Apps ~ Accept + Top10perc, data = College)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5334.2  -513.9   -16.7   325.1  9780.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -892.97561   77.89816  -11.46   <2e-16 ***
## Accept         1.44004    0.01678   85.80   <2e-16 ***
## Top10perc     35.83112    2.33210   15.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1125 on 774 degrees of freedom
## Multiple R-squared:  0.9158, Adjusted R-squared:  0.9156
## F-statistic:  4208 on 2 and 774 DF,  p-value: < 2.2e-16
```

```
m_best <- lm(Apps ~ Private + Accept + Enroll+ Top10perc + Top25perc + Outstate
             + Room.Board + PhD + Expend + Grad.Rate, data = College)
summary(m_best)
```

```
##
## Call:
## lm(formula = Apps ~ Private + Accept + Enroll + Top10perc + Top25perc +
##     Outstate + Room.Board + PhD + Expend + Grad.Rate, data = College)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5085.2  -439.2   -27.4   315.6  7848.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -100.51668  265.47592  -0.379 0.705069
## PrivateYes  -575.07061  132.52820  -4.339 1.62e-05 ***
## Accept         1.58422    0.04011  39.500  < 2e-16 ***
## Enroll        -0.56221    0.11091  -5.069 5.02e-07 ***
## Top10perc     49.13909    5.51638   8.908  < 2e-16 ***
## Top25perc    -13.86531    4.41751  -3.139 0.001762 **
## Outstate      -0.09466    0.01829  -5.176 2.89e-07 ***
## Room.Board     0.16374    0.04668   3.508 0.000478 ***
## PhD          -10.01609    3.11921  -3.211 0.001378 **
## Expend         0.07274    0.01142   6.370 3.26e-10 ***
## Grad.Rate      7.33269    2.82114   2.599 0.009524 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1043 on 766 degrees of freedom
## Multiple R-squared:  0.9283, Adjusted R-squared:  0.9274
## F-statistic: 991.9 on 10 and 766 DF,  p-value: < 2.2e-16
```

```
AIC(m_lasso, m_best)
```

```
##         df      AIC
## m_lasso  4 13127.13
## m_best  12 13018.01
```

```
BIC(m_lasso, m_best)
```

```
##         df      BIC
## m_lasso  4 13145.76
## m_best  12 13073.87
```

The best subset selection choose the better model. Because

- The best subset selection compared all possible combinations of variable and chose the model with the minimum `BIC`.
- On the other hand LASSO choose the model by the LASSO penalty which does not guaranteed to minimize the BIC.

However when the number of covariates is very large, the best subset selection is constitutionally expensive. In this case, LASSO can be applied since the method is less expensive.

[**Total: 70 marks**]