MAST90083: Computational Statistics and Data Science
Assignment1

Zexi Liu 813212

September 15, 2021

Problem 1 Single and Multiple Linear Regression

This question helps you learn how to uncover the relationship between the predictors and a response variable. In case of linear regression, however, this relationship would be overfitted because all predictors will try to explain the response even those that have negligible effect on the response. For this question you need to call two libraries "MASS" and "ISLR". We are going to use Boston dataset for this question. You can familiarize yourself with this dataset using these commands "names(Boston)" and "?Boston" before attempting this question

1 (a)

Using "lm" function and Boston dataset in R fit a linear model for "medv" using "rm" as a predictor, for instance if medv is stored as a response in variable y and rm as a predictor in x then fitted data can be obtained as "y.fit=lm(y $\sim$ x,data=Boston)". Both "medv" and "rm" are variables from Boston dataset.

Ans:

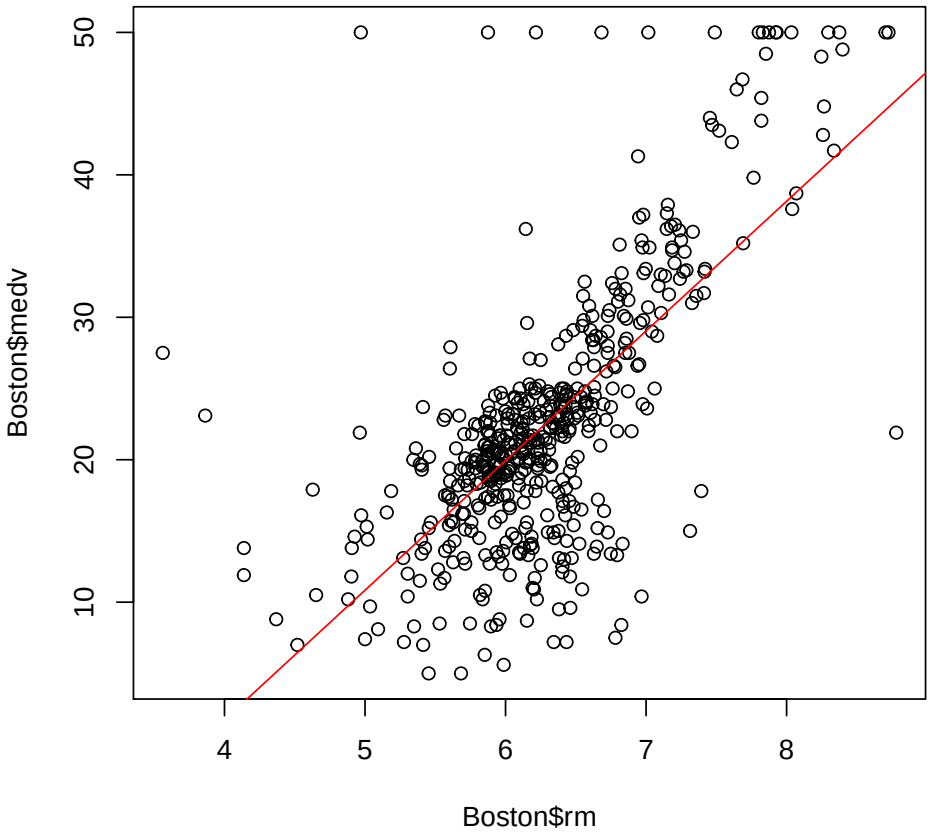
```
> # set up
> library(ISLR)
> library(MASS)
> library(corrplot)
> library(tidyverse)
> data("Boston")
> y.fit <- lm(medv~rm, data= Boston)
```

1 (b)

Use 'plot' command for plotting the response variable (medv) against the predictor (rm) and 'abline' for plotting the fitted data. Do you see the linear approximation by fitted data?

Ans: Accroding to the scatter plot and fitted line. The linear approximation by the fitted data appears to appropriated describe the relation between response and predictor.

```
> plot(Boston$rm, Boston$medv)
> abline(y.fit, col="red")
```



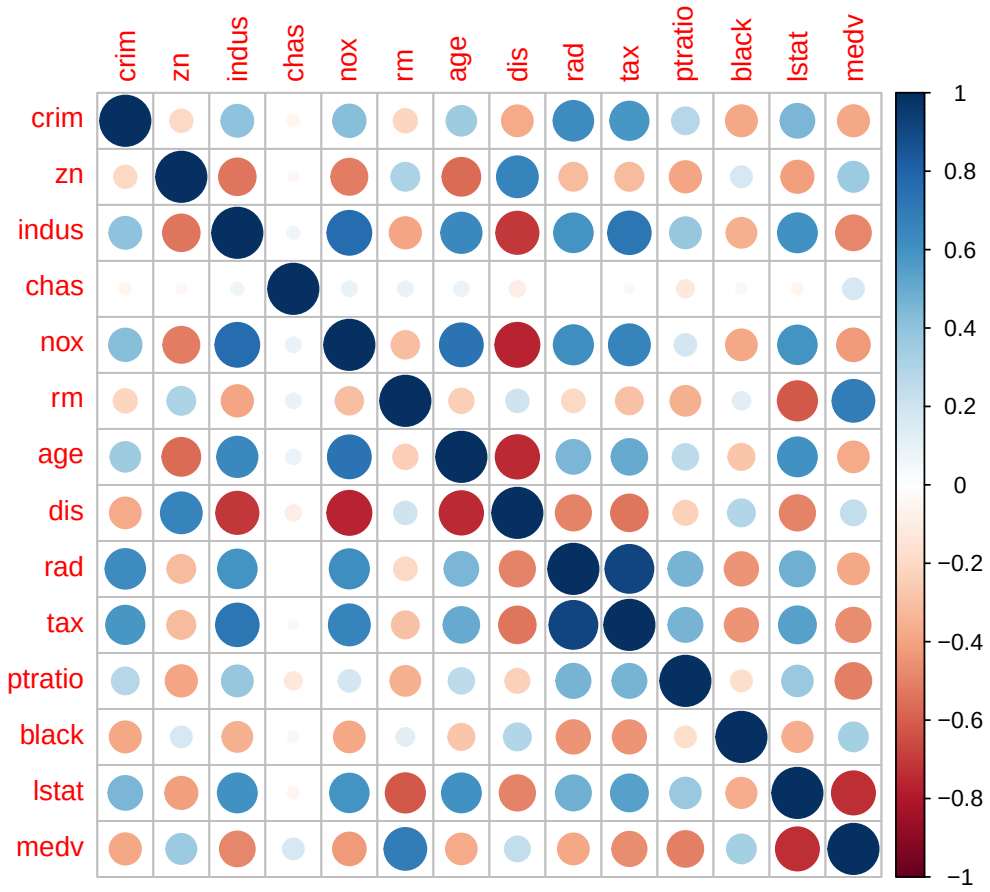
1 (c)

A summary command on variable "y.fit" will produce a "Multiple R-squared" value of 0.4835. Can you find any other variable in Boston dataset (other than "rm") that can produce higher "Multiple R-squared" than 0.4835. You must use correlation function on Boston dataset to suggest a solution. You may want to use library "corrplot" for this purpose. Now, confirm your result by using the selected variable for linear fitting and then taking the summary of this new fitted data, and what is its multiple R-squared value?

Ans The other variable is "lstat". This is because from the correlation plot, in the last row (medv), the "lstat" has the biggest red dot, meaning that lstat is strongly negatively correlated to medv. Therefore, "lstat" predictor may produce a better linear fit than "rm"

predictor. As verified in summary() command, the new multiple R-squared is 0.5441, which is higher than the original.

```
> corrplot(cor(Boston))
> summary(lm(medv~lstat, data= Boston))

Call:
lm(formula = medv ~ lstat, data = Boston)

Residuals:
    Min      1Q  Median      3Q     Max
-15.168  -3.990  -1.318   2.034  24.500

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.216 on 504 degrees of freedom
Multiple R-squared:  0.5441,        Adjusted R-squared:  0.5432
F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16

>
```



1 (d)

Repeat the linear fitting from part 1 for all variables in Boston dataset and store it in y.mfit and then take a summary. From the summary table you can find the p-value (Pr($< |t|$)) in right most column, using this can you suggest four variables that are enough to predict "medv"?

Ans: The four variables are "lstat", "rm ", "dist" and "ptratio". This is because they have the smallest numbers of 4 p-values. The null hypothesis tests(i.e Wald test) here is to set coefficient of the corresponding variable to be 0. Now the smaller p value, the less likely to reject null hypothesis. Therefore, it is reasonable to believe "lstat", "rm ", "dist" and "ptratio" with the smallest 4 p-values can fit the model well. The result can be further supporeted by AIC model selction criterion in step() function.

```
> y.mfit <- lm(medv~., data=Boston)
> summary(y.mfit)

Call:
lm(formula = medv ~ ., data = Boston)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-15.595  -2.730  -0.518   1.777  26.199

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,        Adjusted R-squared:  0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
> step(y.mfit,trace=0)

Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
    tax + ptratio + black + lstat, data = Boston)

Coefficients:
(Intercept)         crim           zn         chas          nox           rm
  36.341145    -0.108413     0.045845     2.718716   -17.376023     3.801579
       dis          rad          tax      ptratio        black        lstat
  -1.492711     0.299608    -0.011778    -0.946525     0.009291    -0.522553
```

1 (e)

Now plot Boston dataset using command ”plot(Boston)”, from this plot you must suggest which variable has minimal effect on ”medv”? Can you also suggest which variable has literally no role to play in this dataset? You can also see the value of the coefficients by using

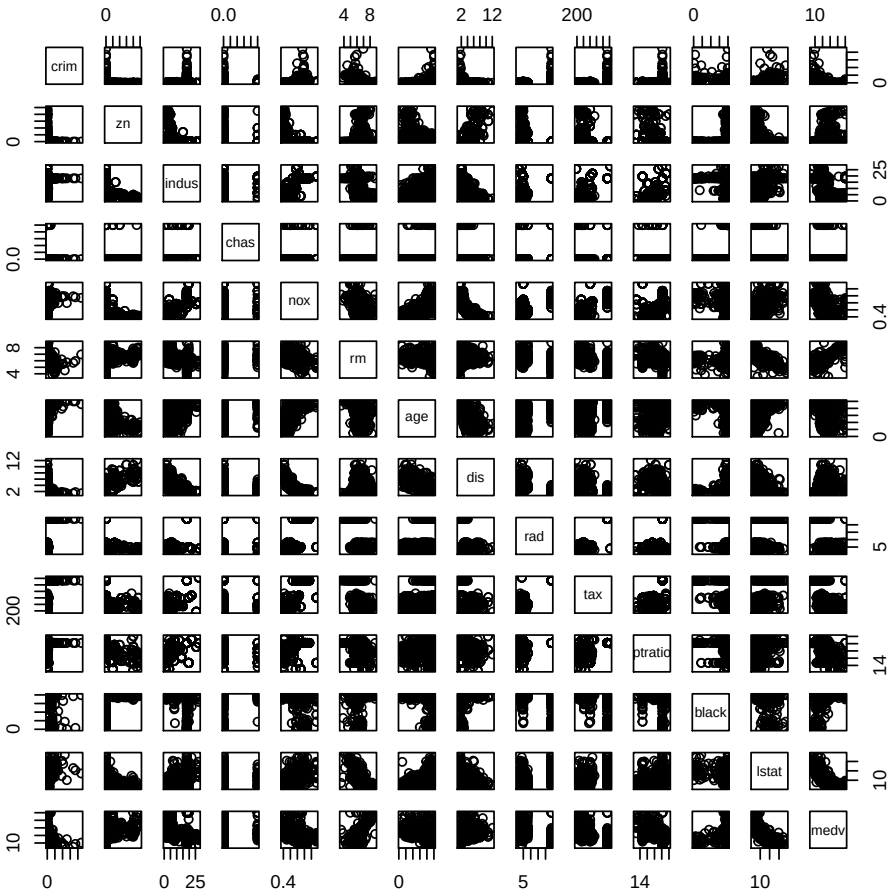
function "coef(y.mfit)", did you notice least square's weakness has clearly been signified as it has ended up selecting all variables and even that variable that has a very weak effect on "medv"? You may also want to use correlation matrix again on Boston data to get a clearer picture about this mystery variable.

Ans:

1. From the scatter plot, the variables "age" and "ptratio" have no clear linear/non-linear trends on "medv", it is not enough evidences to suggests only one variable(as asked in questions) that has minimal effect on "medv".

2. From the scatter plot, "black", "age" and "patratio" are likely to have no role to play in the whole dataset. This is because there are no clear trends of these three variables when we plot these three variables against any other variables. Similarly, it is not enough evidences just by eyeballing and pick the minimal effect variables out of these three.

   ```
   > plot(Boston)
   ```



3. FurtherMore, when we can see the value of the coefficients that "age" and "black" has the smallest two (close to 0). But "age" has p value greater than 0.5 in summary(), which suggests "age" is very insignificant.

```
> coef(y.mfit)

  (Intercept)          crim            zn         indus          chas
 3.645949e+01 -1.080114e-01  4.642046e-02  2.055863e-02  2.686734e+00
          nox            rm           age           dis           rad
-1.776661e+01  3.809865e+00  6.922246e-04 -1.475567e+00  3.060495e-01
          tax       ptratio         black         lstat
-1.233459e-02 -9.527472e-01  9.311683e-03 -5.247584e-01
```

4. Lastly, in the correlation matrix, "black" has the second least correlation with any other variables. "age" do not have strong correlations with any other variables but "nox" and "indus".

   "chas" is a dummy variable, it made sense for it to have the least correlations with any other variables.

5. Combined, it is very hard to answer which single variable is the only "mystery" variable in this dataset, because different metrics yield different results on each variable. There are a couple of candidate variables, e.g. "age" and "black".

   But if were to pick one variable, "age" is preferred over "black" because even if in correlation plot, it has strong correlations with some variables, it is insignificant when fitting lm() while black has significant p value. Furthermore, the value of "age" coefficient is 10 times smaller than "black".

Problem 2 Ridge Regression and Lasso Regression

This question presents a way to resolve the issues with linear regression by doing variable selection so that predictors that fail to significantly explain the response can be dropped. However you will find that ridge regression although penalizes the coefficients, still fails to do variable selection because it together shrinks all of the coefficients towards zero. Lasso on other hand resolves this issue and only shrinks the insignificant coefficients towards zero. For this question we are going to use Hitters dataset

2 (a)

Load the Hitters dataset. Remove all those rows from Hitters dataset that have entry NA in the "salary" column

Ans:

```
> data(Hitters)
> # this function takes a df and a column name
> # return a df that filtered out NA value of that column
> f1 <- function(data, desiredCols) {
+   completeVec <- complete.cases(data[, desiredCols])
+   return(data[completeVec,])
+ }
> cleanedHitter <-f1(Hitters,"Salary")
> dim(cleanedHitter)

[1] 263  20
```

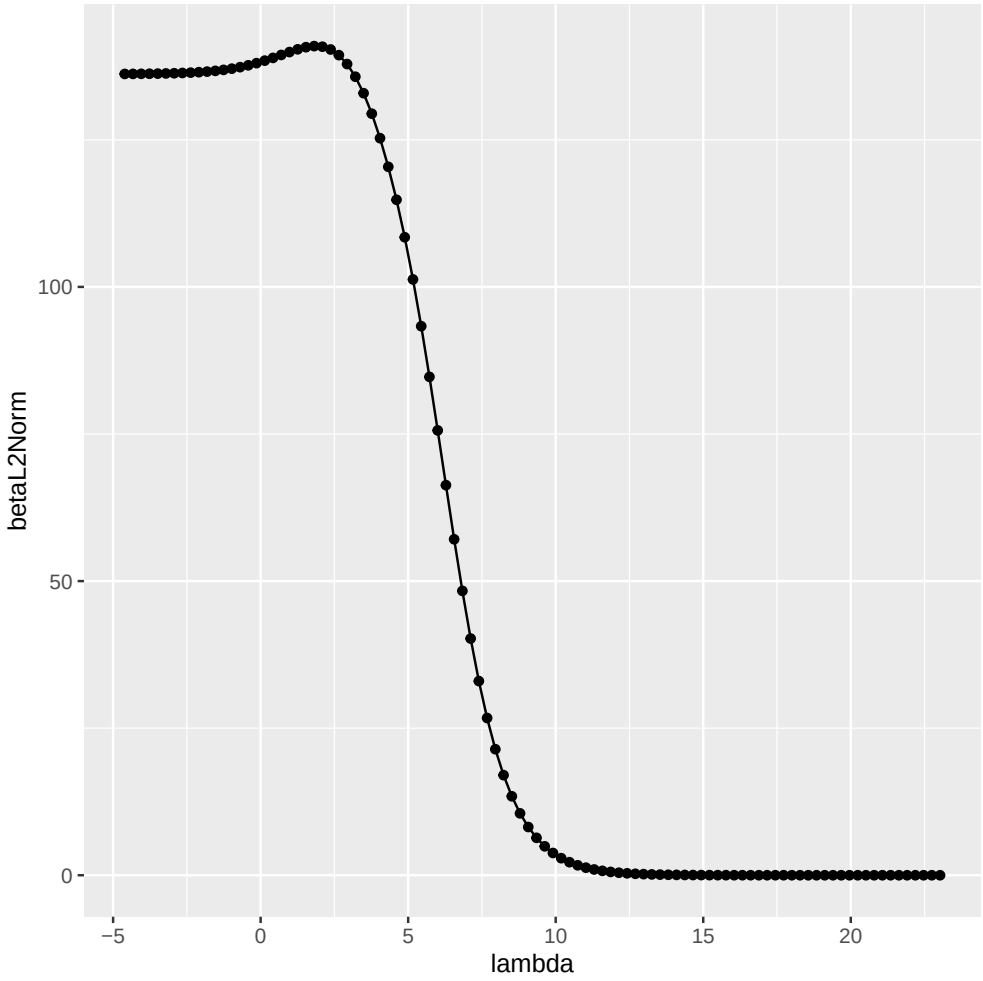
2 (b)

For a design matrix construction, use function "model.matrix" to read all variables in Hitters dataset excluding the salary and store them in variable x. Also, read the salary variable and store it in variable y. Generate a sequence of 100 values of λ between 10^{10} and 10^{-2} and call the function "glmnet" from glmnet library. You can generate the sequence as $10^{s}eq(10, -2, length = 100)$. For glmnet, set $\alpha = 0$, and estimate ridge coefficients for 100 λ values. Then, observe the set of coefficients for two extreme values of λ i.e. 10^{10} and also for 10^{-2}. For which value λ among these two, the coefficient values are more close to zero?

Ans:

1. based on the observation, for $\lambda = 10^{10}$, the coefficients are more closed to 0.

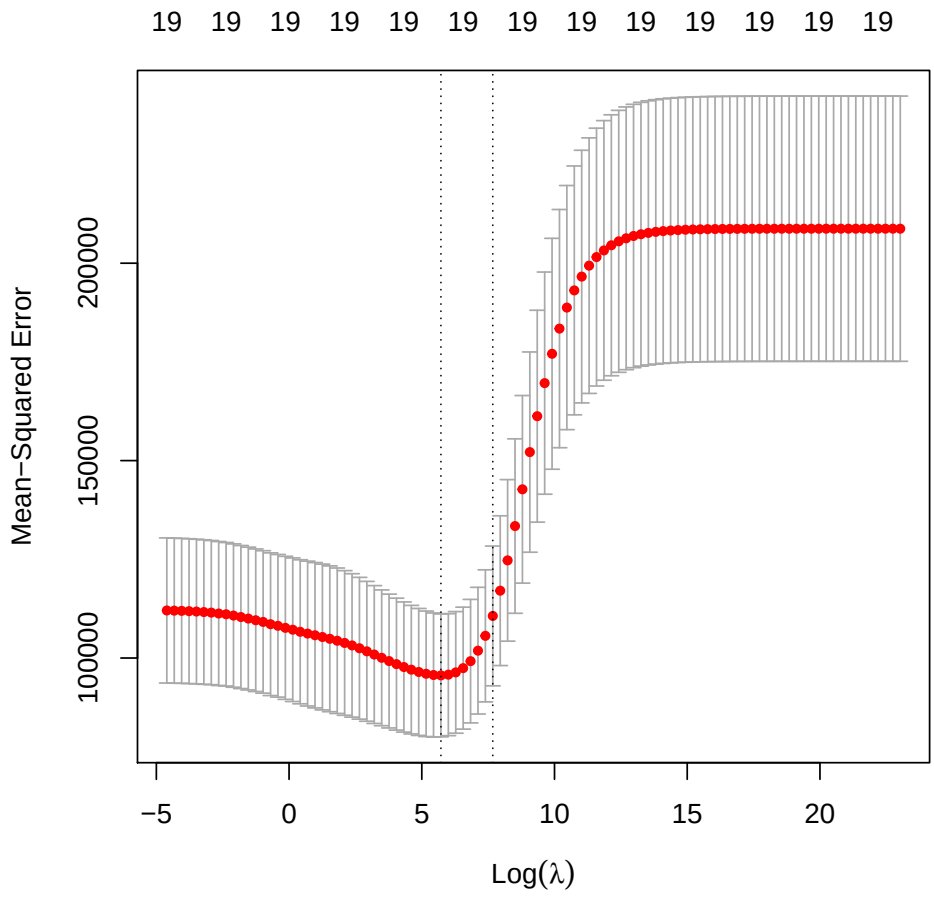
2. ```
   > library(glmnet)
   > x <- model.matrix(~.,
   +                   data = subset(cleanedHitter, select = -c(Salary)))
   > dim(x)

   [1] 263  20

   > y <- cleanedHitter %>% select(Salary) %>% unlist() %>% as.numeric()
   > lambdaSeq <- sort(10^seq(10,-2,length=100), decreasing=TRUE)
   > # trim first column for fitting
   > x<- x[,-1]
   > modQ20<- glmnet(x,y,family = "gaussian",
   +                 lambda = lambdaSeq, alpha = 0)
   > modQ20$beta[,1]

          AtBat           Hits         HmRun           Runs            RBI
   5.443467e-08  1.974589e-07  7.956523e-07  3.339178e-07  3.527222e-07
          Walks          Years        CAtBat         CHits        CHmRun
   4.151323e-07  1.697711e-06  4.673743e-09  1.720071e-08  1.297171e-07
          CRuns           CRBI        CWalks       LeagueN     DivisionW
   ```
   ```

```
 3.450846e-08  3.561348e-08  3.767877e-08 -5.800263e-07 -7.807263e-06
      PutOuts       Assists        Errors    NewLeagueN
 2.180288e-08  3.561198e-09 -1.660460e-08 -1.152288e-07
```

```
> modQ20$beta[,100]
```

```
        AtBat          Hits         HmRun          Runs           RBI
  -1.97386151    7.37772270    3.93660219   -2.19873625   -0.91623008
        Walks         Years        CAtBat         CHits        CHmRun
   6.20037718   -3.71403424   -0.17510063    0.21132772    0.05629004
        CRuns          CRBI        CWalks       LeagueN     DivisionW
   1.36605490    0.70965516   -0.79582173   63.40493257 -117.08243713
       PutOuts       Assists        Errors    NewLeagueN
   0.28202541    0.37318482   -3.42400281  -25.99081928
```

```
>
```

2 (c)

Now, draw a plot of l2 norm of coefficient values (excluding the intercept's coefficient value) against the logarithm of the λ values. Can you say from this plot that you cannot really decide the optimal λ value between 10^{10} and 10^{-2}, better is to use the mean square error (MSE) plot against the λ values? Explain how can you say that?

Ans:

1. No, you can not really decide the optimal value. This is because we are chosing lambda and estimator to minimize the [L2 norm of the errors + L2 norm of estimators] by the definition of ridge regression.Therefore, by only looking at the L2 norm of the estimators, it is simply not enough to say about the model fit, but the L2 norm of the errors in MSE can reflect. Our goal is to select lambda such that will yield adequate and robust model fit. Therefore, MSE would be a better metric for choosing lambda.

2. ```
   > betaL2Norm <- modQ20$beta[1:19,] %>%
   +   apply(. , MARGIN=2, FUN=function(x) sqrt(sum(x^2)))
   > df <- data.frame(betaL2Norm=betaL2Norm, lambda=log(lambdaSeq))
   > ggplot(df, aes(lambda, betaL2Norm)) + geom_line() + geom_point()
   ```
   ```



2 (d)

The glment library already has a function "cv.glmnet" that performs ten fold cross validation (CV). You are going to use this function to select an optimal λ. Now, first you need to set the seed equal to 10 for random number generator. Then randomly pick 131 samples from x for all variables and also the corresponding samples from y to construct a training dataset. The rest of the samples can be saved for testing dataset. Using this training dataset, plot the cross validation results, and find the best λ (the one that results in smallest CV error) value and its corresponding test MSE value (MSE value obtained using testing dataset and best λ), you may want to use "predict" function here. Now refit the ridge regression model on the full data set using the λ chosen by CV. Examine the coefficients are they all present, similar to the linear regression case?

Ans:

1. ```
   > set.seed(10)
   > sample <- sample.int(n = nrow(cleanedHitter),
   +                      size = 131, replace = F)
   ```
   ```

```
> Train <- cleanedHitter[sample,]
> Test <- cleanedHitter[-sample,]
> xTrain <- model.matrix(~.,
+                        data =
+                          subset(Train, select = -c(Salary)))[,-1]
> yTrain <- Train$Salary
> xTest <- model.matrix(~.,
+                       data =
+                         subset(Test, select = -c(Salary)))[,-1]
> yTest <- Test$Salary
> cv_fitRidge <- cv.glmnet(xTrain,
+                       yTrain,
+                       alpha=0,
+                       lambda = lambdaSeq,
+                       type.measure = "mse")
> plot(cv_fitRidge)
```



2. the optimal λ is 305.3856, with corresponding MSE value on test data is 143198.3.

```
> opt_lambda <- cv_fitRidge$lambda.min
> opt_lambda

[1] 305.3856

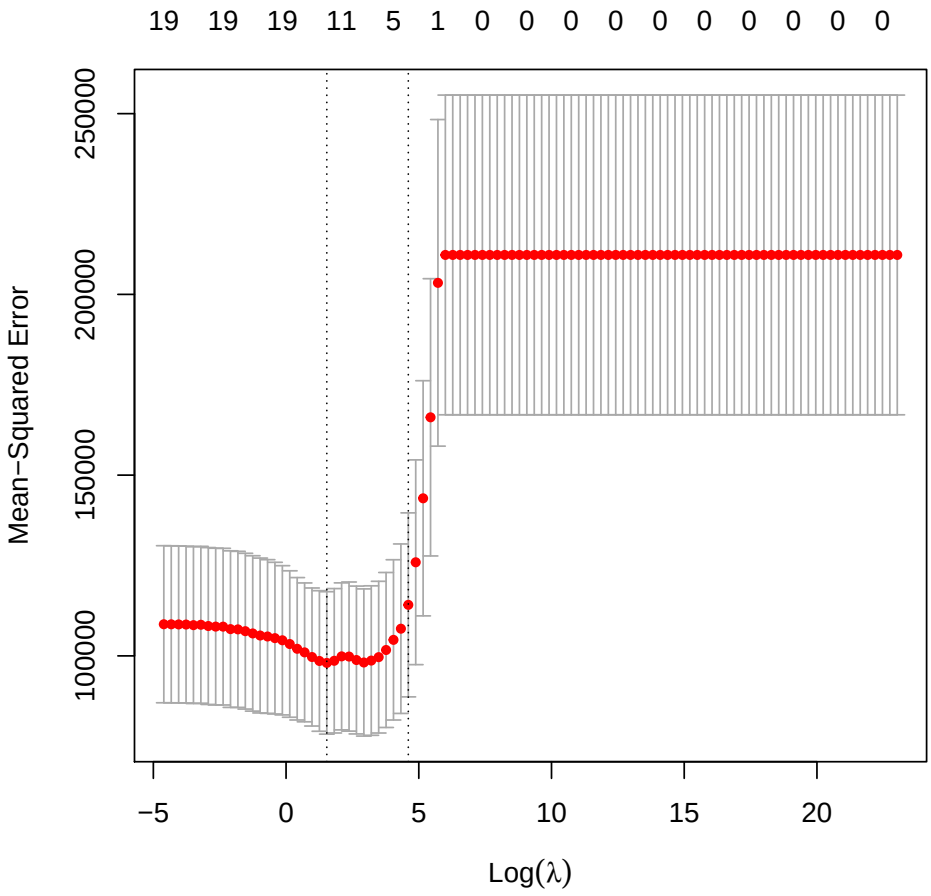
> fit <- cv_fitRidge$glmnet.fit
> y_predictedRidge <- predict(fit, s = opt_lambda, newx = xTest)
> mean((yTest - y_predictedRidge)^2)

[1] 143198.3
```

3. The coefficients present in ridge regression compared to linear regression are shrinked toward 0. They are smaller. Some even changed their signs due to the property of squaring e.g. AtBat, NewLeague.

```
> modQ2_4 <- glmnet(x,y,family = "gaussian", alpha = 0)
> # Display coefficients using lambda chosen by CV
> predict(modQ2_4, type = "coefficients", s = opt_lambda)

20 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept)  13.91429315
AtBat         0.07166979
Hits          0.87965227
HmRun         0.53587852
Runs          1.07207365
RBI           0.87917614
Walks         1.65070495
Years         1.19367801
CAtBat        0.01134204
CHits         0.05848894
CHmRun        0.41313344
CRuns         0.11655950
CRBI          0.12344347
CWalks        0.05009204
LeagueN      22.84559042
DivisionW   -81.04537252
PutOuts       0.17017769
Assists       0.03110668
Errors       -1.42665062
NewLeagueN    8.94720373

> lm(y~x)$coefficients

 (Intercept)       xAtBat        xHits       xHmRun        xRuns         xRBI
 163.1035878   -1.9798729    7.5007675    4.3308829   -2.3762100   -1.0449620
```

```
      xWalks       xYears      xCAtBat       xCHits      xCHmRun       xCRuns
   6.2312863   -3.4890543   -0.1713405    0.1339910   -0.1728611    1.4543049
      xCRBI      xCWalks     xLeagueN   xDivisionW     xPutOuts     xAssists
   0.8077088   -0.8115709   62.5994230 -116.8492456    0.2818925    0.3710692
     xErrors  xNewLeagueN
  -3.3607605  -24.7623251
```

2 (e)

This time we set $\alpha = 1$ (Lasso case) and again plot the cross validation results, and find the best λ value (using training set) and its corresponding MSE value (using testing set). Now predict the coefficients again using the best λ that we just selected. Were all coefficients selected again? Well most of them are zero, are they not?

Ans:

1. ```
   > cv_fitLasso <- cv.glmnet(xTrain,
   +                     yTrain,
   +                     alpha=1,
   +                     lambda = lambdaSeq,
   +                     type.measure = "mse")
   > plot(cv_fitLasso)
   ```
   ```



2. the optimal λ is 4.641589, with corresponding MSE value on test data is 141598.5

```
> opt_lambda1 <- cv_fitLasso$lambda.min
> opt_lambda1

[1] 4.641589

> y_predictedLasso <-
+   cv_fitLasso$glmnet.fit %>%
+   predict(s = opt_lambda1, newx = xTest)
> mean((yTest - y_predictedLasso)^2)

[1] 141598.5
```

3. No, not all the coefficients are slected. The number of selected coefficients are 14 out of 20 (including the intercept).

```
> modQ2_4 <- glmnet(x,y,family = "gaussian", alpha = 1)
>  # Display coefficients using lambda chosen by C
```

```
> lasso_coef <- predict(modQ2_4, type = "coefficients", s = opt_lambda1)
> lasso_coef

20 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept)   76.71631419
AtBat         -0.99747132
Hits           4.44648995
HmRun          .
Runs           .
RBI            .
Walks          3.74601242
Years         -6.52325967
CAtBat         .
CHits          .
CHmRun         0.28617764
CRuns          0.47879492
CRBI           0.41317832
CWalks        -0.30436214
LeagueN       28.75742837
DivisionW   -118.92356507
PutOuts        0.25754274
Assists        0.06992114
Errors        -1.18214997
NewLeagueN     .

> length(lasso_coef[lasso_coef!=0])

[1] 14
```

Problem 3 Model Selection

In this question we consider the analysis of three model selection criteria for selecting the order p of the following model

$$y_t = \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \eta_t \quad t = p+1, ..., n$$

where η_t are independent identically distributed (i.i.d) from $N(0, \sigma^2)$. The criteria we consider are

$$IC_1 = log(\hat{\sigma_p}^2) + \frac{2(p+1)}{T} \tag{1}$$

$$IC_2 = log(\hat{\sigma_p}^2) + \frac{T+p}{T-p-2} \tag{2}$$

$$IC_3 = log(\hat{\sigma_p}^2) + \frac{plog(T)}{T} \tag{3}$$

where $\hat{\sigma_p}^2 = \frac{RSS_p}{T} = \frac{\|y-\hat{y}\|^2}{T}$

3 (a)

In the IC's given above, T represents the number of effective samples. In the case of the model of order p above what is T?

Ans: T = n-p

3 (b)

Find the least square estimator of $\boldsymbol{\phi} = (\phi_1, ..., \phi_p)^T$

Ans: the model $y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \eta_t \quad t = p+1, ..., n$ can be written in a matrix form:

$$\begin{bmatrix} y_{p+1} \\ y_{p+2} \\ ... \\ y_n \end{bmatrix} = \begin{bmatrix} y_p & y_{p-1} & ... & y_1 \\ y_{p+1} & y_p & ... & y_2 \\ ... & ... & ... & ... \\ y_{n-1} & y_{n-2} & ... & y_{n-p} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ ... \\ \phi_p \end{bmatrix} + \begin{bmatrix} \eta_{p+1} \\ \eta_{p+2} \\ ... \\ \eta_{n-1} \end{bmatrix}$$

More compactly, $\boldsymbol{y} = \boldsymbol{Y}\boldsymbol{\phi} + \boldsymbol{\eta}$, where $\boldsymbol{y}$ and $\boldsymbol{\eta}$ are a $(n-p) \times 1$ column vector, $\boldsymbol{\phi}$ is a $p \times 1$ column vector and $\boldsymbol{Y}$ is a $(n-p) \times p$ matrix.
Writing in this matrix form, the least square of this model is $\boldsymbol{\eta}^T\boldsymbol{\eta} = (\boldsymbol{y} - \boldsymbol{Y}\boldsymbol{\phi})^T(\boldsymbol{y} - \boldsymbol{Y}\boldsymbol{\phi})$

Follows the same the least square estimator derivation for OLS, i.e. second derivative test, $\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \boldsymbol{\eta}^T\boldsymbol{\eta} = (\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{y}$

3 (c)

Provide the expression of $\hat{\sigma_p}^2$

Ans:

$$\begin{aligned}
\hat{\sigma}_p^2 &= \frac{RSS_p}{T} \\
&= \frac{\hat{\boldsymbol{\eta}}^T\hat{\boldsymbol{\eta}}}{T} \\
&= \frac{(\boldsymbol{y} - \boldsymbol{Y}\hat{\phi})^T(\boldsymbol{y} - \boldsymbol{Y}\hat{\phi})}{T} \\
&= \frac{(\boldsymbol{y} - \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{y})^T(\boldsymbol{y} - \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{y})}{T} \\
&= \frac{((\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y})^T((\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y})}{T} \\
&= \frac{\boldsymbol{y}^T(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y})}{n-p} \quad \text{where } \boldsymbol{H} \text{ is the hat matrix i.e. } \boldsymbol{H} = \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T\boldsymbol{y} \text{ and } \boldsymbol{I} - \boldsymbol{H} \text{ is idempotent}
\end{aligned}$$

3 (d)

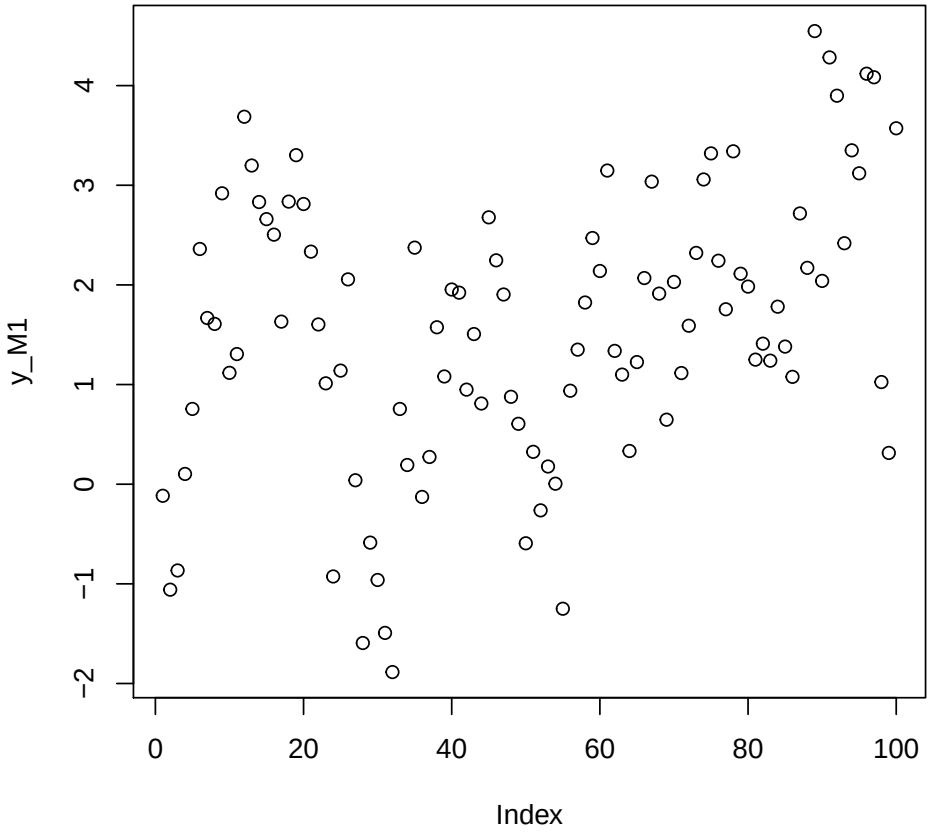
Generate two sets of 100 samples using the models

$$M1 : y_t = 0.434y_{t-1} + 0.217y_{t-2} + 0.145y_{t-3} + 0.108y_{t-4} + 0.087y_{t-5} + \eta_t \quad \eta_t \ N(0,1)$$

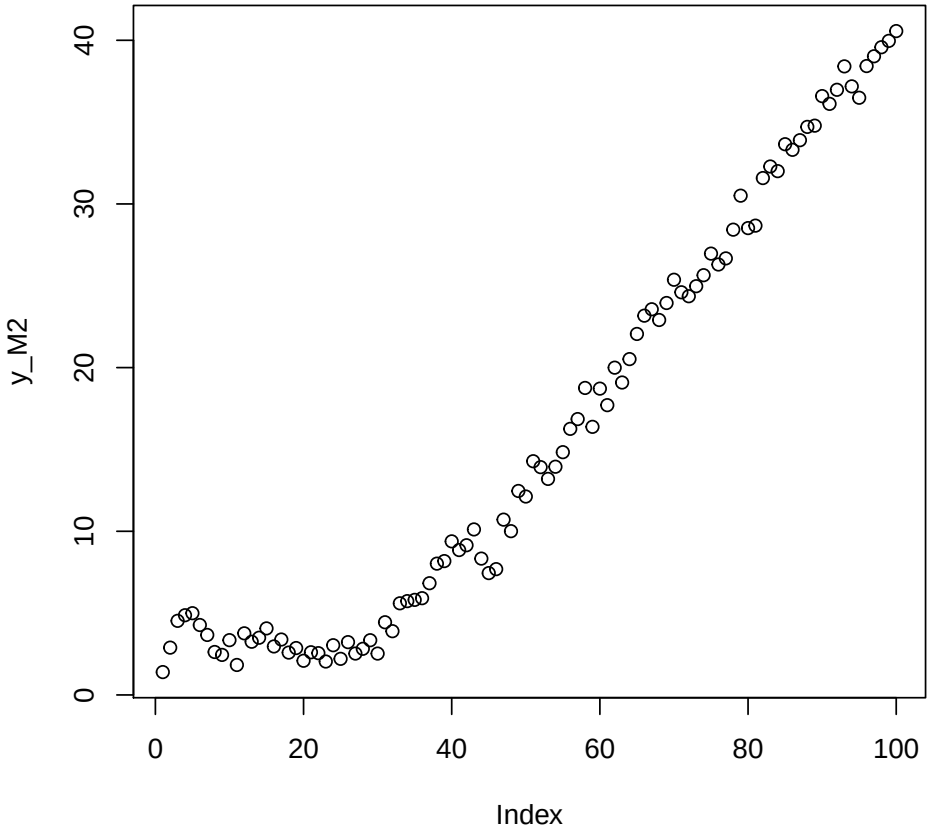
$$M2 : y_t = 0.682y_{t-1} + 0.346fy_{t-2} + \eta_t \quad \eta_t \ N(0,1)$$

Ans:

```
> set.seed(NULL)
> M1 <- function(y_t, i){
+   return(0.434*y_t[i-1]+0.217*y_t[i-2]+
+            0.145*y_t[i-3]+0.108*y_t[i-4]+0.087*y_t[i-5])
+ }
> simQ3_4 <- function(y_t, M, p){
+   y_t <- numeric()
+   y_t <-  c( rep(0,p))
+   eta <- rnorm(100+p, mean = 0, sd = 1)
+   for(i in (p+1):(100+p)) {
+     y_t <- c(y_t, M(y_t,i) + eta[i])
+   }
+   return(y_t[(p+1):(100+p)])
+ }
> y_M1 <- simQ3_4(y_M1, M1, 5)
> plot(y_M1)
```




```
> M2 <- function(y_t, i){
+   return(0.682*y_t[i-1]+0.346*y_t[i-2])
+ }
> y_M2 <- simQ3_4(y_M2, M2, 2)
> plot(y_M2)
```



3 (e)

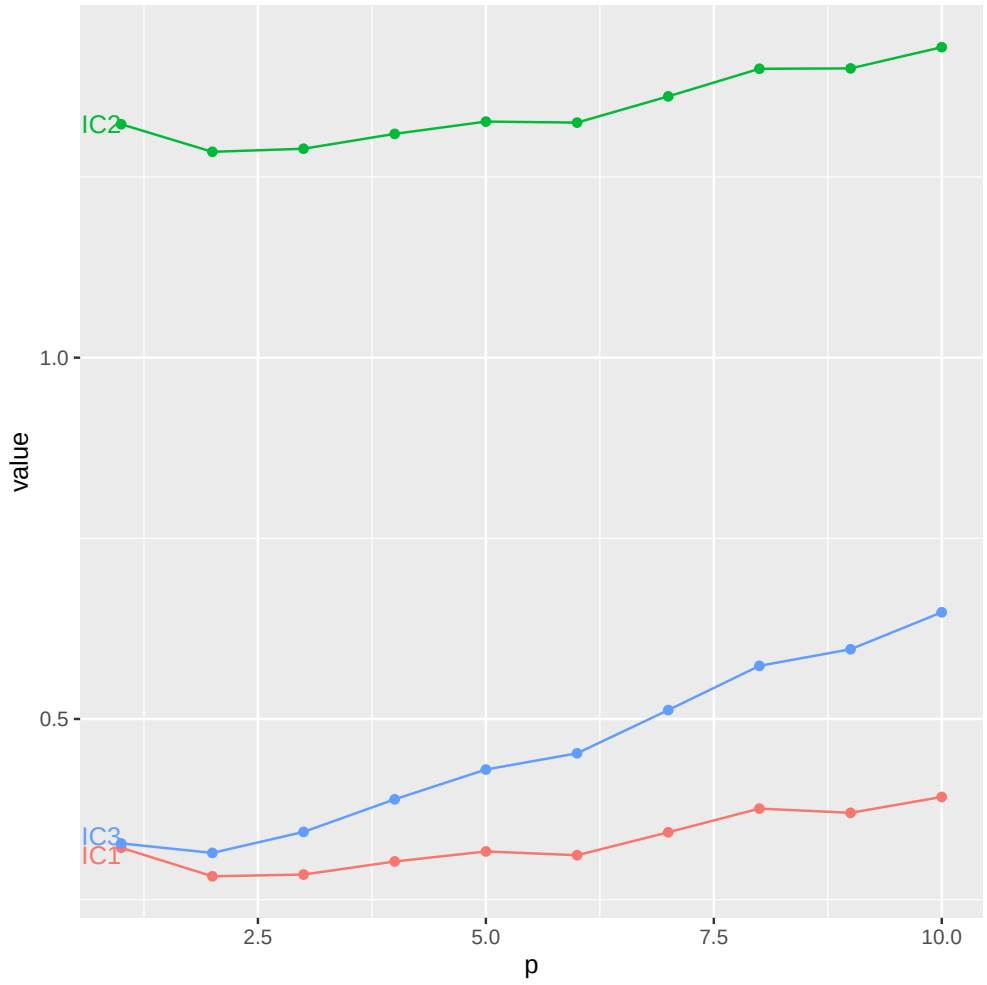
Using these two sets, compute the values of IC1, IC2 and IC3 for p = 1,...,10 for models M1 and M2. For each model provide a figure illustrating the variations of IC1, IC2 and IC3 (plot the three criteria in a single figure for each model).

Ans:

```
> library(ggplot2)
> require(reshape2)
> require(directlabels)
> ##############################################################
> # Functions
> ##############################################################
> # @ YDesign function takes a sequence of sample and generate
> # the design matrix for lm model, (n-p)*p
> YDesign <- function(y_t,p){
+   n <- length(y_t)
```

```
+   y <- y_t[p:(n-1)]
+   if(p ==1){
+     return(as.matrix(y, nrow = length(y)))
+   } else {
+     for(i in 2:p){
+       indexi <- p-i+1
+       y <- cbind(y,y_t[indexi:(n-i)])
+     }
+     return(as.matrix(y, nrow = n-p, ncol = p))
+   }
+ }
> # @ YResponse model takes a sample, generate (n-p) response
> YResponse <- function(y_t, p){
+   n <- length(y_t)
+   return(as.matrix(y_t[(p+1):n],
+                    nrow= length(y_t[(p+1):n])))
+ }
> # IC1,2,3 Criterions, takes a model fit, return ICi
> IC1 <- function(fit){
+   T <- length(fit$residuals)
+   p <- length(fit$coefficients)
+   RSS <- sum(resid(fit)^2)
+   sigma <- RSS/T
+   ic_1 <- log(sigma) + ((2*(p+1))/T)
+   return(ic_1)
+ }
> IC2 <- function(fit){
+   T <- length(fit$residuals)
+   p <- length(fit$coefficients)
+   RSS <- sum(resid(fit)^2)
+   sigma <- RSS/T
+   ic_2 <- log(sigma) + (T+p)/(T-p-2)
+   return(ic_2)
+ }
> IC3 <- function(fit){
+   T <- length(fit$residuals)
+   p <- length(fit$coefficients)
+   RSS <- sum(resid(fit)^2)
+   sigma <- RSS/T
+   ic_3 <- log(sigma) + ((p*log(T))/T)
+   return(ic_3)
+ }
> ###############################################################
> # Simulations
> ###############################################################
```

```
> IC1_M1q35 <- numeric()
> IC2_M1q35 <- numeric()
> IC3_M1q35 <- numeric()
> for(i in 1:10){
+   yDesign <- YDesign(y_M1,i)
+   yResponse <- YResponse(y_M1,i)
+   fit <- lm(yResponse~yDesign+0)
+   IC1_M1q35 <- append(IC1_M1q35,IC1(fit))
+   IC2_M1q35 <- append(IC2_M1q35,IC2(fit))
+   IC3_M1q35 <- append(IC3_M1q35,IC3(fit))
+ }
> #plot
> df <- data.frame(IC1=IC1_M1q35, IC2=IC2_M1q35,
+                  IC3=IC3_M1q35, p=1:10 )
> df.m <- melt(df,id.vars="p")
> p <- ggplot(df.m, aes(x=p, y=value, color=variable)) +
+   geom_line() +geom_point()
> direct.label(p)
```



3 (f)

Using model M1 generate 1000 sets (vectors) of size 100 and provide a table of counts of the selected model by IC1, IC2 and IC3

Ans:

```
> ################################################################
> # Simulations
> ################################################################
> simQ3_6 <- function(M, p, n){
+   y_t <- numeric()
+   y_t <-  c( rep(0,p))
+   eta <- rnorm(n+p, mean = 0, sd = 1)
+   for(i in (p+1):(n+p)) {
+     y_t <- c(y_t, M(y_t,i) + eta[i])
+   }
+   return(y_t[(p+1):(n+p)])
+ }
> ################################################################
> # Count_models
> ################################################################
> selectCount<-function(sim=simQ3_6,sampleSize, M, Mp){
+   result3.6 <- c()
+ # fixed IC1, 100 sample, underlying process M1
+   for(ic in c(IC1, IC2, IC3)){
+     dfQ3_6 <- data.frame(p1 = 0, p2 = 0, p3 = 0,
+                          p4 = 0, p5 = 0, p6 =0,
+                          p7 = 0, p8 = 0, p9 = 0,
+                          p10 =0)
+     for(i in 1:1000){
+       # simulate 100 M1 sample
+       yQ3_6 <- simQ3_6(M, Mp,sampleSize)
+       IC <- numeric()
+       for(p in 1:10){
+         yDesgin <-YDesign(yQ3_6,p)
+         yResponse <-YResponse(yQ3_6,p)
+         fit <- lm(yResponse~yDesgin+0)
+         IC <- append(IC, ic(fit))
+       }
+       # update
+       dfQ3_6[1,which(IC == min(IC))]<-
+         dfQ3_6[1,which(IC == min(IC))] + 1
+     }
+     result3.6<-c(result3.6, dfQ3_6)
+   }
```

```
+   return(result3.6)
+ }
> # make table
> makeTable <- function(result){
+   occur <- matrix(result,ncol=10,nrow=3, byrow = TRUE)
+   colnames(occur) <- c("p1","p2","p3","p4","p5",
+                        "p6","p7","p8","p9","p10")
+   rownames(occur) <- c("IC1","IC2","IC3")
+   return(occur)
+ }
> # M5,1000 of 100 sample size
> ##############################################################
> # Table 1
> ##############################################################
> selectCount(sampleSize = 100,M= M1, Mp=5)%>%makeTable()

    p1 p2  p3  p4  p5  p6 p7 p8 p9 p10
IC1 4  111 295 207 147 76 57 32 38 33
IC2 13 125 317 247 140 52 47 22 23 14
IC3 40 384 389 123 50  10 3  1  0  0
```

3 (g)

Using model M1 generate 1000 sets of size 15 and provide a table of counts of the selected model by IC1, IC2 and IC3

Ans:

```
> ##############################################################
> # Table 2
> ##############################################################
> selectCount(sampleSize = 15,M= M1, Mp=5)%>%makeTable()

    p1 p2 p3 p4 p5 p6 p7 p8   p9   p10
IC1 0  0  0  0  0  0  0  1000 1000 1000
IC2 0  0  0  0  0  0  0  1000 1000 1000
IC3 0  0  0  0  0  0  0  1000 1000 1000
```

3 (h)

Repeat questions 6 and 7 using model M2.

Ans:

```
> ##############################################################
> # Table 3
> ##############################################################
> selectCount(sampleSize = 100,M= M2, Mp=2)%>%makeTable()

    p1  p2  p3  p4 p5 p6 p7 p8 p9 p10
IC1 24  567 141 86 58 27 25 21 29 22
IC2 28  629 141 73 43 32 21 15 7  11
IC3 116 815 49  14 3  1  1  0  1  0

> ##############################################################
> # Table 4
> ##############################################################
> selectCount(sampleSize = 15,M= M2, Mp=2)%>%makeTable()

    p1 p2 p3 p4 p5 p6 p7 p8   p9   p10
IC1 0  0  0  0  0  0  0  1000 1000 1000
IC2 0  0  0  0  0  0  0  1000 1000 1000
IC3 0  0  0  0  0  0  0  1000 1000 1000
```

3 (i)

What do you observe from these tables?

Ans:

1. From Table 1 (M1, n=100), IC1, IC2, IC3 counts all seem to to underestimate the true $p_0 = 5$. IC1 and IC2 counts have similar distribution and $p = 3, p = 4$ tends to have more counts than others. IC3 has more skewed distribution, centered at p=2 and p=3

2. From Table 2 (M1, n=15) and Table 4 (M2, n=15), when p is greater than 8, the number of effective data is greater than the number of parameters when postuplate models. In this case, because there is not enough data for parameter fittings, the RSS is 0 in lm() function. As a result, the table produces negative infnity IC because σ^2 is 0, implies $\log(\sigma^2)$ is negative infinity. The counts takes no differences when choosing between negative inifinity at p=8,9,10. Consequently, this produces a table where each p =8,9,10 have the same number of model counts 1000.

3. From Table 3 (M2, n=100), we can observe the IC1,IC2,IC3 criteria all have decent amount of counts at p=2. IC1 and IC2 still have similar distribution and IC3 is still left skewed.

3 (j)

Derive expressions for the probabilities of overfitting for the model selection criteria IC1, IC2 and IC3. For the derivation you will assume the true model to be p0 and consider overfitting by L extra parameters.

Ans:

3 (k)

Provide tables of the calculated probabilities for M1 in the cases n = 25 and n = 100 with L = 1, ..., 8.

Ans:

```
> #IC, n=25, L =1,...8. p0=5
> # This function take IC type , N and true P, return the probability
> # of overfitting
> calculateProb <- function(IC, n, p){
+   result <- numeric()
+   for(L in 1:8){
+     if(IC == "IC1"){
+       criticalValue <- ((((n-2*p-2*L)*(n-p)/((2*L)*(n-p-L)))) *
+                           exp((2*p+2+2*L)/(n-p-L)- (2*p+2)/(n-p))
+                         - ((n-2*p-2*L)/(2*L)))
+     }else if(IC == "IC2"){
+       criticalValue <- ((((n-2*p-2*L)*(n-p)/((2*L)*(n-p-L)))) *
+                           exp((n)/(n-2*p-2*L-2)- (n)/(n-2*p-2))
+                         - ((n-2*p-2*L)/(2*L)))
+     }else if(IC == "IC3"){
+       criticalValue <- ((((n-2*p-2*L)*(n-p)/((2*L)*(n-p-L)))) *
+                           exp((p+L)*log(n-p-L)/(n-p-L)-
+                                 (p*log(n-p))/(n-p))
+                         - ((n-2*p-2*L)/(2*L)))
+     }else{
+       print("invalid input IC")
+     }
+     result <- c(result, 1-pf(criticalValue, 2*L, n-2*p-2*L))
+   }
+   return(result)
+ }
> ################################################################
> # Simulations
> ################################################################
> ICi <- c("IC1","IC2","IC3")
> N <- c(25, 100)
```

```
> output3_11 <- numeric()
> for(ic in ICi){
+   for(n in N){
+     output3_11<-c(output3_11,calculateProb(ic, n, 5))
+   }
+ }
> # plot
> output3_11 <- matrix(output3_11, nrow =6 ,
+                      ncol=8, byrow = TRUE)
> colnames(output3_11) <- c("L=1","L=2","L=3"
+                           ,"L=4","L=5","L=6"
+                      ,"L=7","L=8")
> rownames(output3_11) <- c("IC1 n=25","IC1 n=100",
+                           "IC2 n=25", "IC2 n=100",
+                      "IC3 n=25", "IC3 n=100")
> options(digits=2)
> ################################################################
> # Table 4
> ################################################################
> output3_11
```

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8
IC1 n=25	0.294	0.319	0.3500	0.40299	4.9e-01	6.5e-01	0.87562	NaN
IC1 n=100	0.232	0.212	0.1900	0.17034	1.5e-01	1.4e-01	0.12728	0.11700
IC2 n=25	0.074	0.022	0.0039	0.00024	1.2e-06	6.4e-15	1.00000	NaN
IC2 n=100	0.196	0.161	0.1287	0.10178	8.0e-02	6.2e-02	0.04798	0.03656
IC3 n=25	0.221	0.221	0.2358	0.27867	3.7e-01	5.4e-01	0.82778	NaN
IC3 n=100	0.069	0.030	0.0140	0.00668	3.3e-03	1.7e-03	0.00086	0.00046

3 (l)

What are the important remarks that can be made from these probability tables?

Ans:

1. When $n = 100$, $p_0 = 5$, The probability of overfitting decreases for all IC1, IC2 and IC3, as L increases
2. When $n = 25$, the effective sample size is $n - L - p_0 = 20 - L$, therefore, the probability of overfitting under 3different model selection criteria behave differently
   - For all 3 criteria, when $L = 7$, the overfitting probability suddenly blow up then at $L = 8$ the probability is not defined, meaning the critical value is negative for F distribution.
   - For IC1 and IC3, the probability is increasing as L goes to 6
   - For IC2 the probability is deccreasing as L goes to 6

3 (m)

The tables obtained from question 11 provide overfitting information as a function of the sample size. We are now interested in the case of large sample size or when n $\rightarrow \infty$ (p0 and L fixed). Derive the expressions of the probabilities of overfitting in this case.

Ans:

3 (n)

What is the important observation that you can make?

Ans:

1. the overfitting probability for IC1 and IC2 has the same limiting probability.
    - It follows a chi-square distribution with the same degree of freedom
    - This probability, either the distribution RHS or the critical value LHS is only dependent on L, but not p.
    - the limiting overfitting probability is 0 for IC3. Meaning that IC3 is a good choice for model selection when n goes large because the the probability of IC3 selecting a overfitting model is 0.