Stat 432 Homework 5

Giang Le (gianghl2)

Assigned: Sep 20, 2021; Due: 11:59 PM CT, Sep 28, 2021

Contents

Question 1: Data Preparation

We will use a modified data collected from sepsis patients. The data contains 470 observations and 13 variables, which are mainly clinical variables or blood measurements. Each patient went through an active treatment or no treatment, denoted by `THERAPY`, and outcome variable we want to predict is `Health`.

- Health: Health status, the higher the better
- THERAPY: 1 for active treatment, 0 for control treatment
- TIMFIRST: Time from first sepsis-organ fail to start drug
- AGE: Patient age in years
- BLLPLAT: Baseline local platelets
- blSOFA: Sum of baseline sofa score (cardiovascular, hematology, hepaticrenal, and respiration scores)
- BLLCREAT: Base creatinine
- ORGANNUM: Number of baseline organ failures
- PRAPACHE: Pre-infusion APACHE-II score
- BLGCS: Base GLASGOW coma scale score
- BLIL6: Baseline serum IL-6 concentration
- BLADL: Baseline activity of daily living score
- BLLBILI: Baseline local bilirubin

Complete the following steps for data preparation:

a. [5 Points] How many observations have missing values? Which variables have missing values and how many are missing?

First I check the names of columns of variables of the dataset and how many missing value there are in each variable.

```
sepsis = read.csv("Sepsis2.csv", row.names = 1)
sum(is.na(sepsis$Health))
```

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

```
sum(is.na(sepsis$THERAPY))
```

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

```
sum(is.na(sepsis$TIMFIRST))
```

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

```r
sum(is.na(sepsis$AGE))
```

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

```r
sum(is.na(sepsis$BLLPLAT))
```

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

```r
sum(is.na(sepsis$blSOFA))
```

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

```r
sum(is.na(sepsis$BLLCREAT))
```

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

```r
sum(is.na(sepsis$ORGANNUM))
```

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

```r
sum(is.na(sepsis$PRAPACHE))
```

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

```r
sum(is.na(sepsis$BLGCS))
```

```
## [1] 3
```

```r
sum(is.na(sepsis$BLIL6))
```

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

```r
sum(is.na(sepsis$BLADL))
```

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

```r
sum(is.na(sepsis$BLLBILI))
```

```
## [1] 50
```

According to this check, the variable "BLGCS" has 3 missing values and the variable "BLLBILI" has 50 missing values. I also check for missing values in the rows.

```r
rowSums(is.na(sepsis))[rowSums(is.na(sepsis))!=0]
```

```
## 392  41 347 130  83 350 447 344  28 361 139 241 233 124 367 446 309 399 270 279
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 437 394 266 211 268 388  94  14 105 216 443 389  20  15 262 358 119 282 162 166
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 422 457  67  99 204 426 192  40   1 286  11 258  16
##   1   1   1   1   1   1   1   1   1   1   1   1   1
```

From the result above, we can see that 53 observations have missing values.

b. [10 Points] Use two different approaches to address the missing value issue. One of the methods you use must be the stochastic regression imputation. Make sure that when you perform the imputation, do not involve the outcome variable. Make sure that you set random seeds using your UIN.

```r
install.packages("mice", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/3_mgdyf12z7dvb8rt4d60nt80000gn/T//Rtmp3uRMmY/downloaded_packages
```

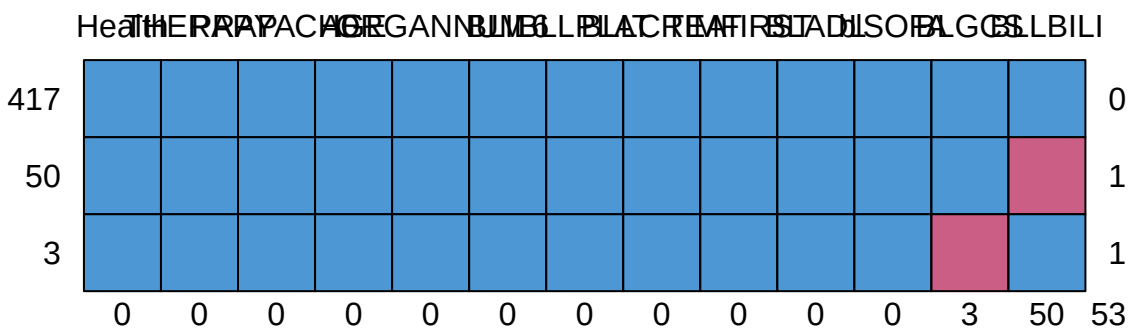
```r
library("mice")
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
# This functions shows the missing pattern
md.pattern(sepsis)
```




```
##     Health THERAPY PRAPACHE AGE ORGANNUM BLIL6 BLLPLAT BLLCREAT TIMFIRST BLADL
## 417      1       1        1   1        1     1       1        1        1     1
## 50       1       1        1   1        1     1       1        1        1     1
## 3        1       1        1   1        1     1       1        1        1     1
##          0       0        0   0        0     0       0        0        0     0
##     blSOFA BLGCS BLLBILI
## 417      1     1       1  0
## 50       1     1       0  1
## 3        1     0       1  1
##          0     3      50 53
```

Here I perform imputation with mean value. First I remove the outcome variable 'Heath'.

```r
sepsis_x = sepsis[, 2:13]
head(sepsis_x)
```

```
##     THERAPY PRAPACHE    AGE BLGCS ORGANNUM  BLIL6 BLLPLAT BLLBILI BLLCREAT
## 365       1       26 33.174     3        1 4952.0     137     0.4      1.0
## 467       1       25 33.174     3        1  212.3     137     0.4      1.0
## 433       0       24 42.921     3        3   60.3      92     0.5      2.2
## 425       0       26 59.871     3        4  723.0      92     0.5      1.2
## 239       0       33 42.921     3        2   37.1      62     0.6      5.0
## 210       1       31 46.532     3        4  406.6     359     0.7      1.1
##     TIMFIRST BLADL blSOFA
## 365    30.67     0      9
## 467  3775.90     0      7
## 433    59.17     0      9
## 425  3775.90     0     11
## 239    21.73     0     10
## 210    19.33     7      7
```

```r
set.seed(662095561)
# Imputation with mean value
imp_mean <- mice(sepsis_x, method = "mean", m = 1, maxit = 1)
```

```
##
##  iter imp variable
##   1   1  BLGCS  BLLBILI
```

Here I perform stochastic regression imputation.

```r
# Stochastic regression imputation.
imp_reg <- mice(sepsis_x, method = "norm.nob", m = 1, maxit = 1)
```

```
##
##  iter imp variable
##   1   1  BLGCS  BLLBILI
```

c. [10 Points] Perform a linear regression on each of your imputed data. Compare the model fitting results.

```r
 # after performing the imputation, I extract the imputed data
imp_data_mean <- complete(imp_mean)
imp_data_reg <- complete(imp_reg)
```

```r
# Regression using the data from imputation by means.
sepsis_lm_impbymean <- lm(sepsis$Health ~ imp_data_mean$THERAPY +
                            imp_data_mean$TIMFIRST +
                            imp_data_mean$AGE +
                            imp_data_mean$BLLPLAT +
                            imp_data_mean$blSOFA +
                            imp_data_mean$BLLCREAT +
                            imp_data_mean$ORGANNUM +
                            imp_data_mean$PRAPACHE +
                            imp_data_mean$BLGCS +
                            imp_data_mean$BLIL6 +
                            imp_data_mean$BLADL +
                            imp_data_mean$BLLBILI)
```

```r
# Regression using the data from imputation by stochastic regression.
sepsis_lm_impbyreg <- lm(sepsis$Health ~ imp_data_reg$THERAPY +
                            imp_data_reg$TIMFIRST +
                            imp_data_reg$AGE +
                            imp_data_reg$BLLPLAT +
                            imp_data_reg$blSOFA +
                            imp_data_reg$BLLCREAT +
                            imp_data_reg$ORGANNUM +
                            imp_data_reg$PRAPACHE +
                            imp_data_reg$BLGCS +
                            imp_data_reg$BLIL6 +
                            imp_data_reg$BLADL +
                            imp_data_reg$BLLBILI)
```

Let's compare the two models' MSE and R^2

```r
mean(sepsis_lm_impbymean$residuals^2)
```

```
## [1] 4.075943
```

```r
mean(sepsis_lm_impbyreg$residuals^2)
```

```
## [1] 4.075624
```

```r
summary(sepsis_lm_impbymean)$r.squared
```

```
## [1] 0.03175272
```

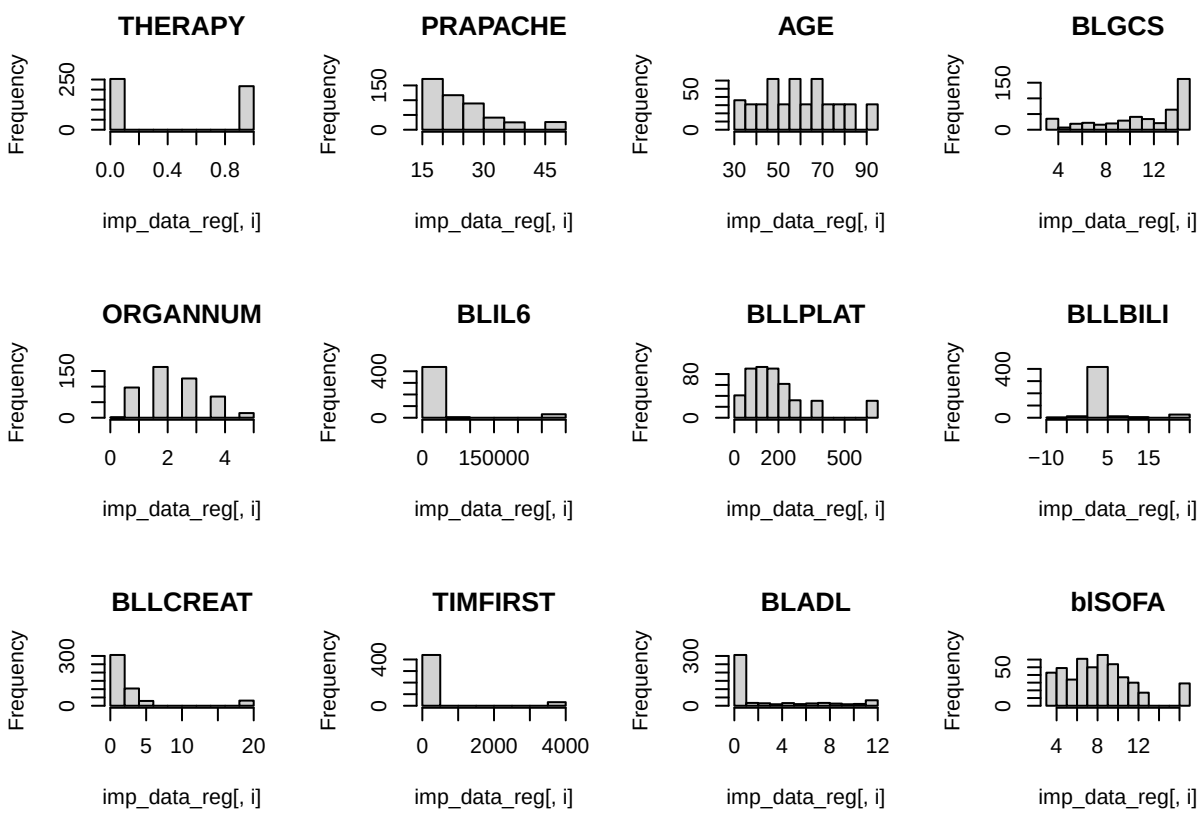
```r
summary(sepsis_lm_impbyreg)$r.squared
```

```
## [1] 0.03182855
```

MSE of the linear regression using the data imputed by stochastic regression is slightly smaller than the MSE of the linear regression using the data imputed by the mean. The R^2 of the linear regression using the data imputed by stochastic regression is slightly larger than the R^2 of the linear regression usng the the data imputed by the mean. So the model using data imputed by stochastic regression is slightly better.

d. [20 Points] Investigate the marginal distribution of each variable (excluding the outcome `Health`) and decide whether the variable could benefit from any transformations. If so, then perform the transformation at your choice. **You need to provide clear evidence to reason your decision and also provide a table that summarizes your decisions**. Save your final data for the next question. While performing these transformations, you do not need to worry about whether they will lead to a better model fitting. There may not be a best decision, or even correct decision. Simply use your best judgement based on the marginal distributions alone.

```r
par(mfrow = c(3, 4))
for (i in 1:ncol(imp_data_reg))
  hist(imp_data_reg[,i], breaks = 10, main = colnames(imp_data_reg)[i])
```



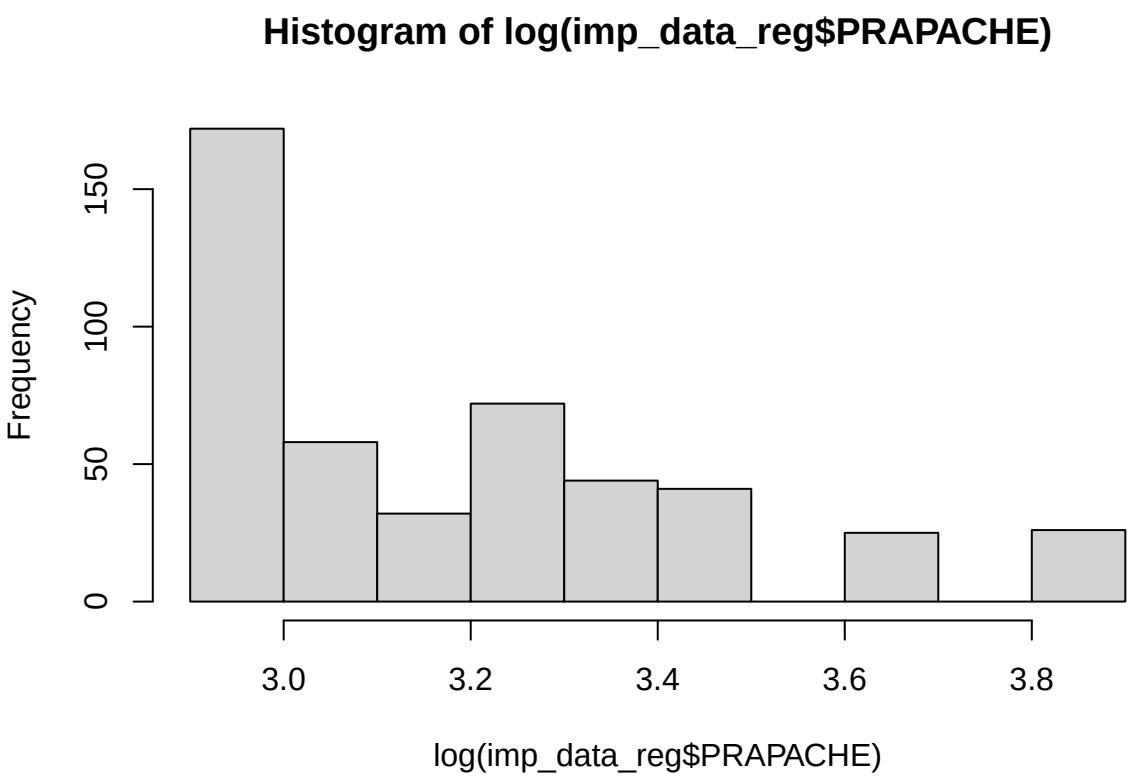
Observing the plots, I propose the following transformations to the data: - Convert THERAPY to a categorical variable because it shows two numeric values 0 and 1. - Perform log transformations to PRAPACHE, BLGCS,

BLIL6, BLLPLAT, BLLCREAT, TIMFIRST, and BLADL because these - variables have a long tail. For BLADL, 1 must be added before taking the log to avoid -Inf values. - Perform quantile transformation for BLLBILI because it has two heavy tails on both sides.

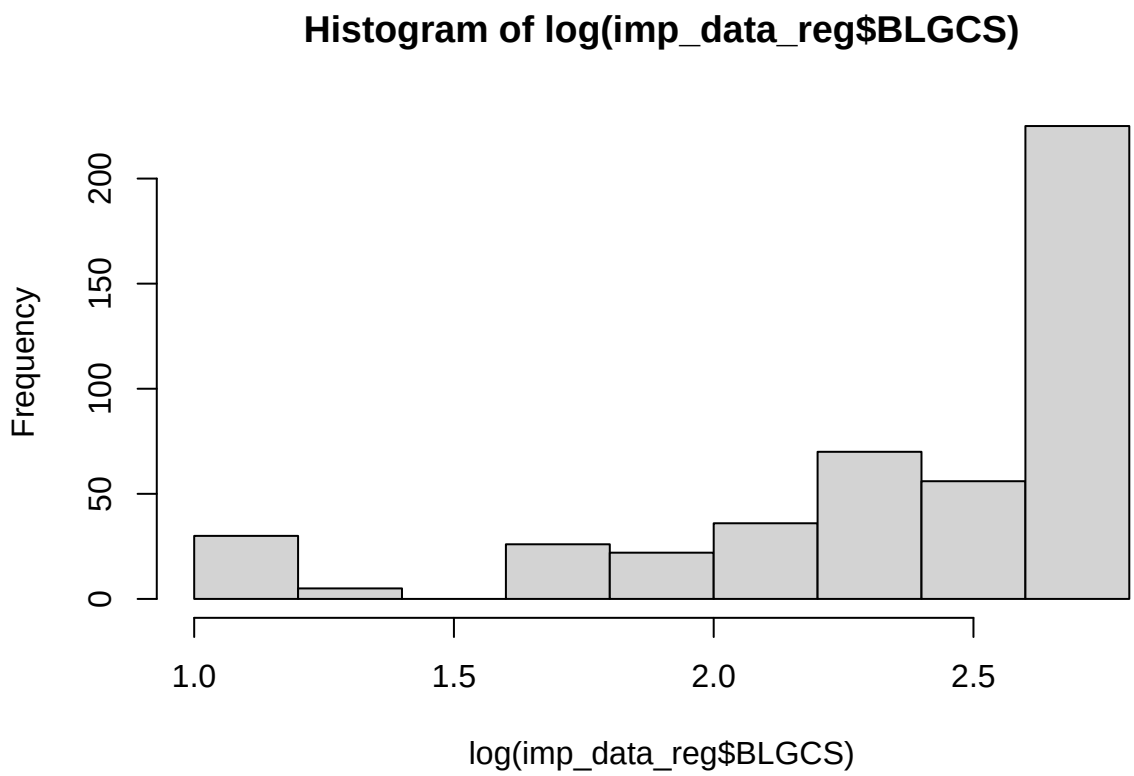
Variables	Transformation
`THERAPY`	Convert to a categorical variable
`PRAPACHE`	log transformation
`BLGCS`	log transformation
`BLIL6`	log transformation
`BLLPLAT`	log transformation
`BLLCREAT`	log transformation
`TIMFIRST`	log transformation
`BLADL`	log transformation with 1 added
`BLLBILI`	Quantile transformation

The histograms show that the transformations help alleviate the problem of skewness in the data.

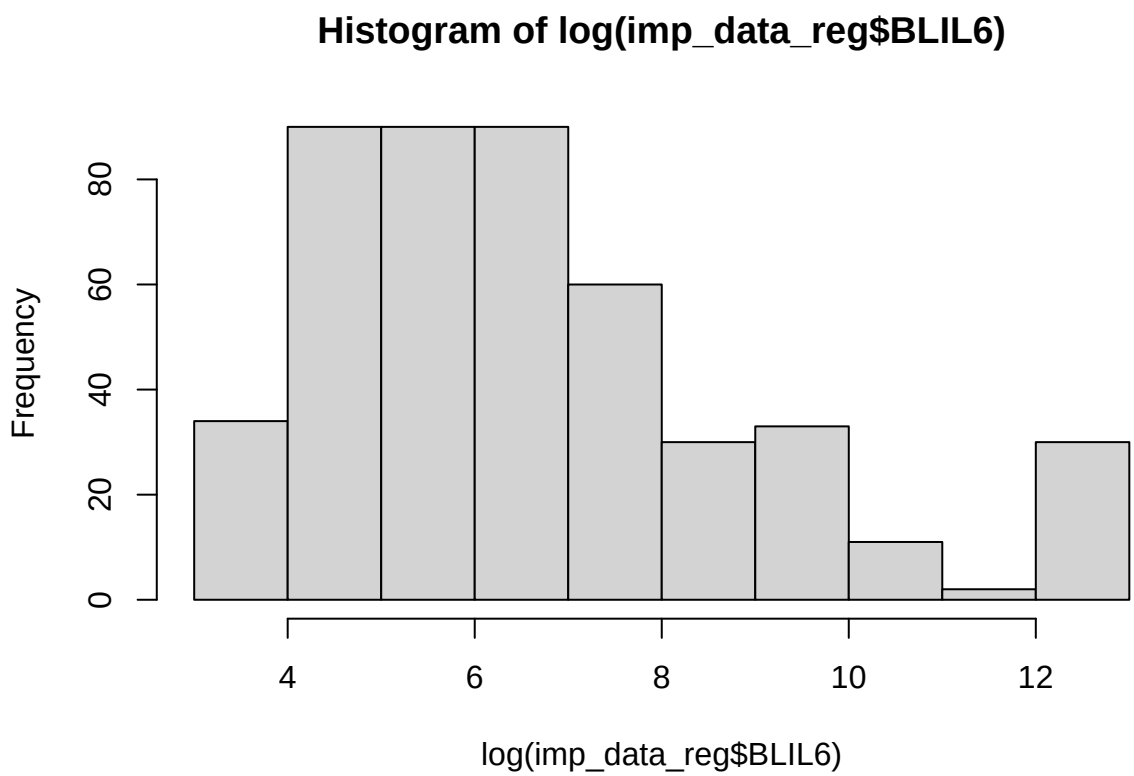
```
# Log transformations
hist(log(imp_data_reg$PRAPACHE))
```



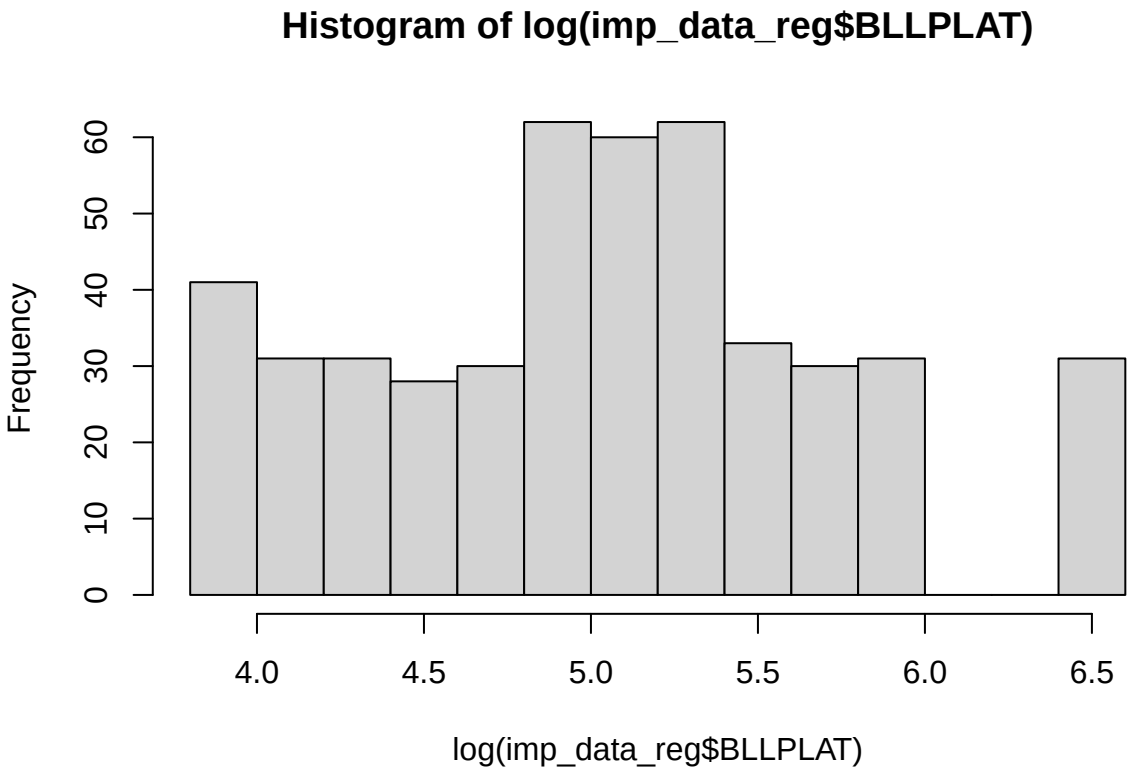

```
hist(log(imp_data_reg$BLGCS))
```



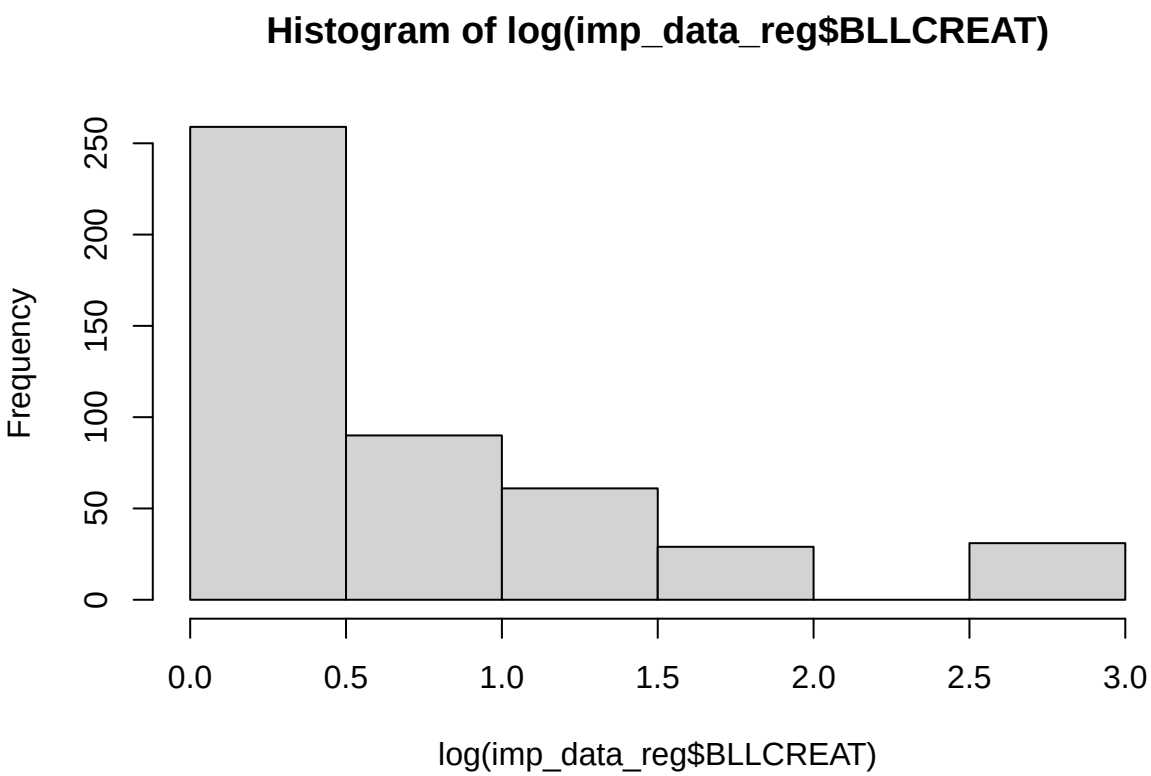

```
hist(log(imp_data_reg$BLIL6))
```



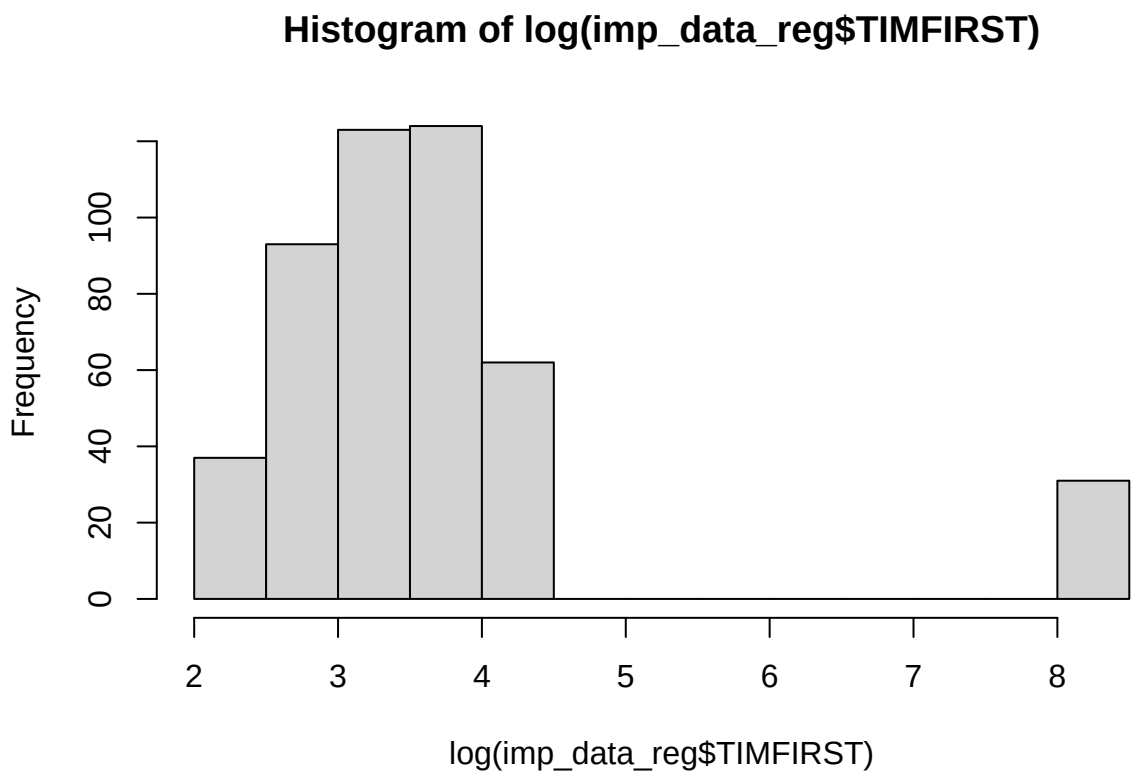

```
hist(log(imp_data_reg$BLLPLAT))
```



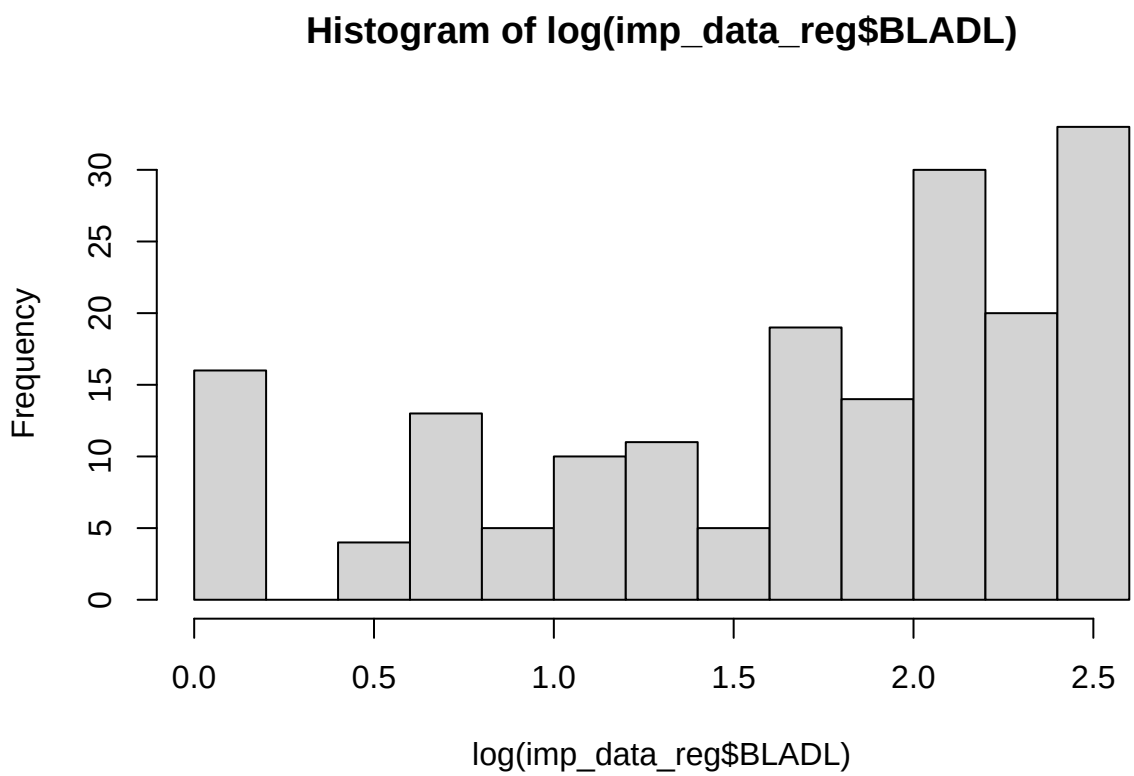

```
hist(log(imp_data_reg$BLLCREAT))
```




```
hist(log(imp_data_reg$TIMFIRST))
```



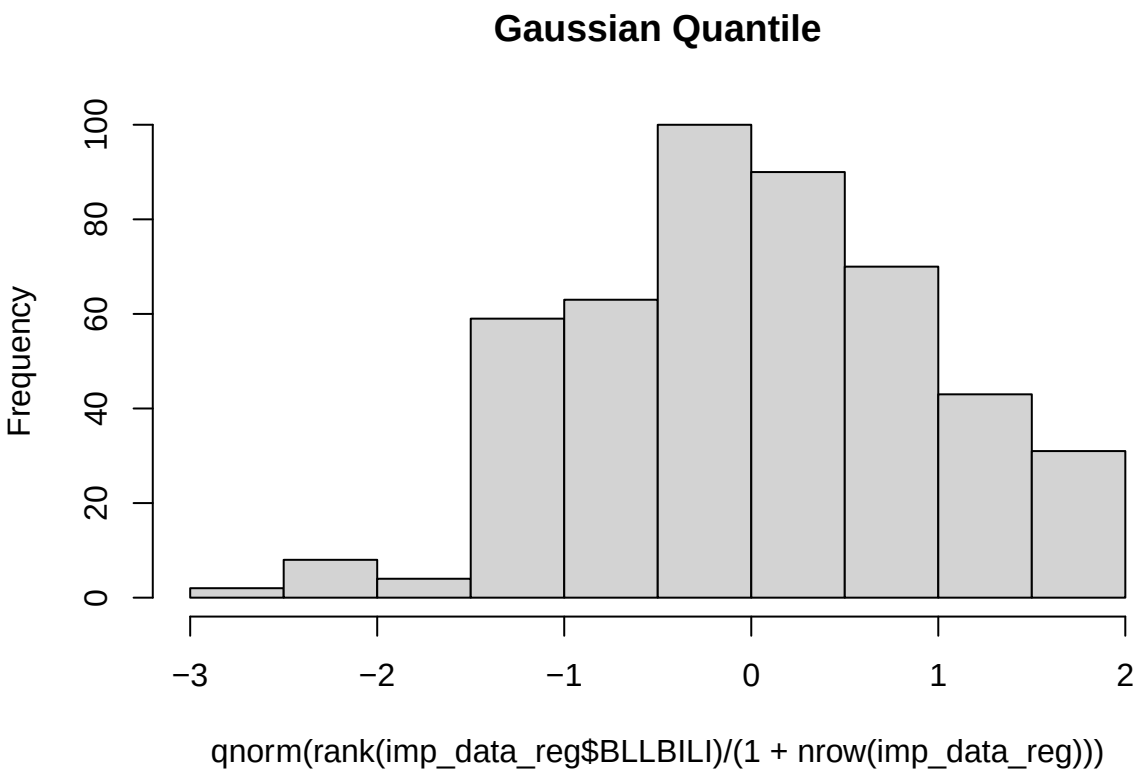

```
hist(log(imp_data_reg$BLADL))
```




```
imp_data_reg$LOG_PRAPACHE=log(imp_data_reg$PRAPACHE)
imp_data_reg$LOG_BLGCS=log(imp_data_reg$BLGCS)
imp_data_reg$LOG_BLIL6=log(imp_data_reg$BLIL6)
```

```r
imp_data_reg$LOG_BLLPLAT=log(imp_data_reg$BLLPLAT)
imp_data_reg$LOG_BLLCREAT=log(imp_data_reg$BLLCREAT)
imp_data_reg$LOG_TIMFIRST=log(imp_data_reg$TIMFIRST)
imp_data_reg$LOG_BLADL=log(1+imp_data_reg$BLADL)
```

```r
# Quantile transformation
 hist(qnorm(rank(imp_data_reg$BLLBILI) / (1 + nrow(imp_data_reg))),
       main = "Gaussian Quantile")
```




```r
imp_data_reg$BLLBILI_NEW=qnorm(rank(imp_data_reg$BLLBILI) / (1 + nrow(imp_data_reg)))
```

```r
imp_data_reg$THERAPY_NEW <- as.factor(imp_data_reg$THERAPY)
```

```r
summary(imp_data_reg)
```

```
##     THERAPY          PRAPACHE          AGE            BLGCS
##  Min.   :0.0000   Min.   :19.00   Min.   :33.17   Min.   : 3.00
##  1st Qu.:0.0000   1st Qu.:19.00   1st Qu.:46.53   1st Qu.: 9.00
##  Median :0.0000   Median :23.00   Median :59.87   Median :13.00
##  Mean   :0.4617   Mean   :25.33   Mean   :59.85   Mean   :11.63
##  3rd Qu.:1.0000   3rd Qu.:28.00   3rd Qu.:73.14   3rd Qu.:15.00
##  Max.   :1.0000   Max.   :48.00   Max.   :93.34   Max.   :15.00
##     ORGANNUM         BLIL6              BLLPLAT          BLLBILI
##  Min.   :0.000   Min.   :    37.1   Min.   : 45.0   Min.   :-9.695
##  1st Qu.:2.000   1st Qu.:   118.9   1st Qu.: 92.0   1st Qu.: 0.600
##  Median :2.000   Median :   406.6   Median :153.0   Median : 1.000
##  Mean   :2.443   Mean   : 21794.4   Mean   :192.3   Mean   : 2.663
##  3rd Qu.:3.000   3rd Qu.:  2568.0   3rd Qu.:244.0   3rd Qu.: 2.500
##  Max.   :5.000   Max.   :296550.0   Max.   :650.0   Max.   :20.400
##     BLLCREAT         TIMFIRST           BLADL            blSOFA
```

```
##  Min.   : 1.000   Min.   :  10.00   Min.   : 0.000   Min.   : 3.000
##  1st Qu.: 1.000   1st Qu.:  19.33   1st Qu.: 0.000   1st Qu.: 6.000
##  Median : 1.500   Median :  30.67   Median : 0.000   Median : 8.000
##  Mean   : 3.104   Mean   : 279.54   Mean   : 2.593   Mean   : 8.568
##  3rd Qu.: 3.000   3rd Qu.:  50.67   3rd Qu.: 4.441   3rd Qu.:10.000
##  Max.   :20.000   Max.   :3775.90   Max.   :12.000   Max.   :17.000
##   LOG_PRAPACHE     LOG_BLGCS       LOG_BLIL6       LOG_BLLPLAT
##  Min.   :2.944   Min.   :1.099   Min.   : 3.614   Min.   :3.807
##  1st Qu.:2.944   1st Qu.:2.197   1st Qu.: 4.778   1st Qu.:4.522
##  Median :3.135   Median :2.565   Median : 6.008   Median :5.030
##  Mean   :3.195   Mean   :2.375   Mean   : 6.615   Mean   :5.015
##  3rd Qu.:3.332   3rd Qu.:2.708   3rd Qu.: 7.851   3rd Qu.:5.497
##  Max.   :3.871   Max.   :2.708   Max.   :12.600   Max.   :6.477
##   LOG_BLLCREAT     LOG_TIMFIRST     LOG_BLADL       BLLBILI_NEW
##  Min.   :0.0000   Min.   :2.303   Min.   :0.0000   Min.   :-2.8592616
##  1st Qu.:0.0000   1st Qu.:2.962   1st Qu.:0.0000   1st Qu.:-0.6529315
##  Median :0.4055   Median :3.423   Median :0.0000   Median :-0.0345994
##  Mean   :0.6660   Mean   :3.665   Mean   :0.7249   Mean   :-0.0002376
##  3rd Qu.:1.0986   3rd Qu.:3.925   3rd Qu.:1.6939   3rd Qu.: 0.6139050
##  Max.   :2.9957   Max.   :8.236   Max.   :2.5649   Max.   : 1.9008257
##  THERAPY_NEW
##  0:253
##  1:217
##
##
##
##
```

Question 2: Lasso and Elatic-Net

Take the final data from your previous question, i.e., with missing data imputed and variable transformations addressed. You do not need to worry too much about whether these processes would improve the prediction error. Focus on fitting the regression models correctly for this question.

a. [20 Points] Perform Lasso on your data to predict `Health`. Report the following:
   - How many fold are you using in the cross-validation?
   - How did you decide which is the best tuning parameter? Please provide figures to support your answer.
   - What is the parameter estimates corresponding to this parameter? Is this solution sparse? Which variable is being excluded?
   - What is the mean cross-validation error corresponding to this?

```r
x_transformed <- imp_data_reg[,c(3, 5, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21)]
head(x_transformed)
```

```
##      AGE ORGANNUM blSOFA LOG_PRAPACHE LOG_BLGCS LOG_BLIL6 LOG_BLLPLAT
## 1 33.174        1      9     3.258097  1.098612  8.507547    4.919981
## 2 33.174        1      7     3.218876  1.098612  5.358000    4.919981
## 3 42.921        3      9     3.178054  1.098612  4.099332    4.521789
## 4 59.871        4     11     3.258097  1.098612  6.583409    4.521789
## 5 42.921        2     10     3.496508  1.098612  3.613617    4.127134
## 6 46.532        4      7     3.433987  1.098612  6.007830    5.883322
##   LOG_BLLCREAT LOG_TIMFIRST LOG_BLADL BLLBILI_NEW THERAPY_NEW
## 1   0.00000000     3.423285  0.000000  -1.3199955           1
## 2   0.00000000     8.236394  0.000000  -1.3199955           1
```

```
## 3   0.78845736     4.080415  0.000000  -0.8739131          0
## 4   0.18232156     8.236394  0.000000  -0.8739131          0
## 5   1.60943791     3.078694  0.000000  -0.6529315          0
## 6   0.09531018     2.961658  2.079442  -0.4512565          1
```

Performing Lasso regression. I used 10 folds for cross validation.

```
install.packages("glmnet", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/3_mgdyf12z7dvb8rt4d60nt80000gn/T//Rtmp3uRMmY/downloaded_packages
```

```
library("glmnet")
```

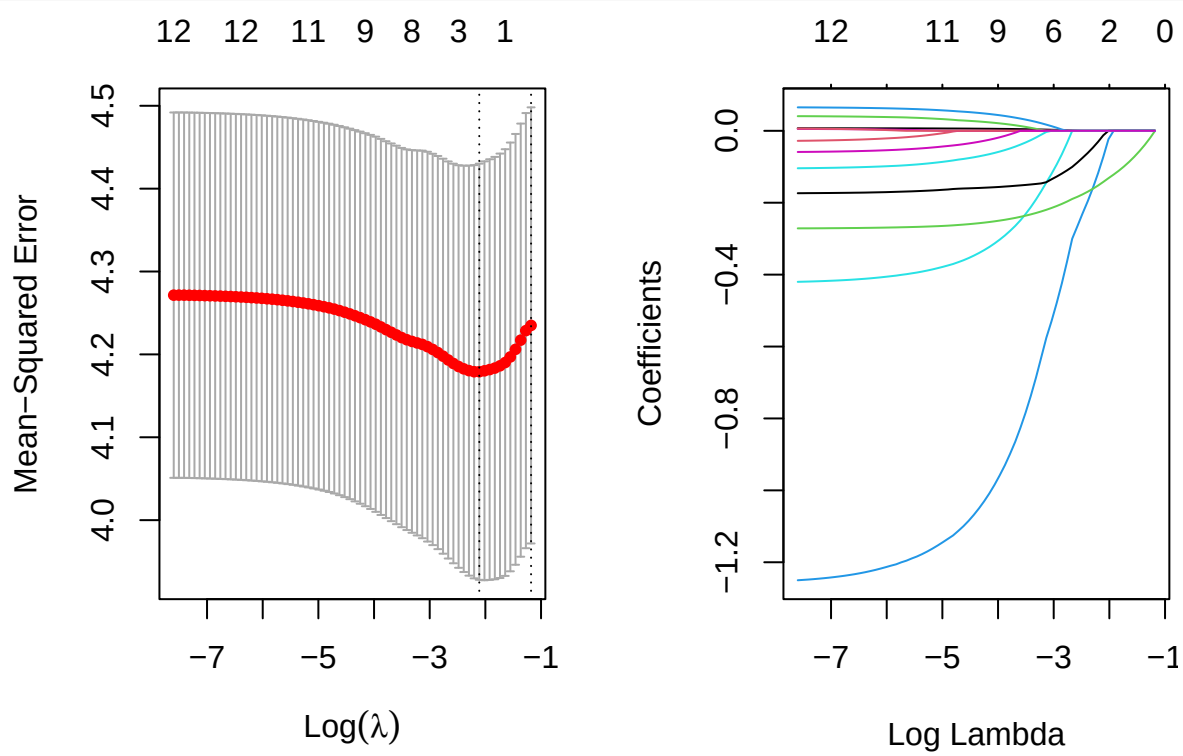
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```
lasso.fit = cv.glmnet(data.matrix(x_transformed[, 1:12]), sepsis$Health, nfolds = 10, alpha = 1)
```

If I choose minimzing the MSE as a tuning criteria then log(lambda) should be approximately -2. If I choose a more conservative larger penalty then log(lambda) should be approximately -1. We can see in the plots below, where the left line is the lambda.min and the right line is the lambda.1se, corresponding to the aforementioned method. I will choose minimizing MSE as a tuning criteria and go with lambda.min

```
par(mfrow = c(1, 2))
plot(lasso.fit)
plot(lasso.fit$glmnet.fit, "lambda")
```



Here are the coefficient estimates according to lambda.min

```
coef(lasso.fit, s = "lambda.min")
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                       s1
## (Intercept)   0.66122988
## AGE           .
## ORGANNUM      .
## blSOFA        .
## LOG_PRAPACHE -0.07262248
## LOG_BLGCS     .
## LOG_BLIL6     .
## LOG_BLLPLAT  -0.01256714
## LOG_BLLCREAT  .
## LOG_TIMFIRST -0.14221297
## LOG_BLADL     .
## BLLBILI_NEW   .
## THERAPY_NEW   .
```

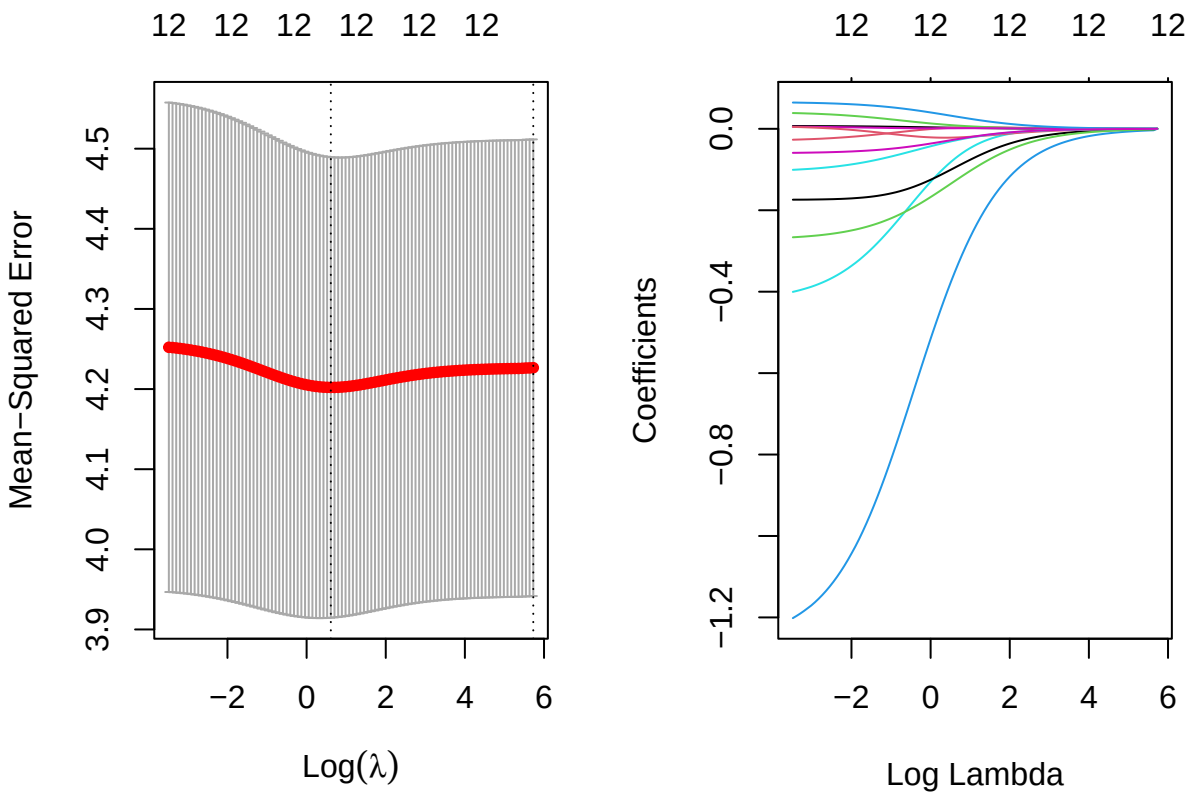
This solution is not as sparse as a solution where lambda.1se is used, because lambda.1se would force more coefficients to be 0. In the case of lambda.min, the variables being excluded are AGE, ORGANNUM, blSOFA, LOG_BLGCS, LOG_BLIL6, LOG_BLLCREAT, LOG_BLADL, BLLBILI_NEW, and THERAPY_NEW.

The MSE corresponding to this is about 4.2, according to the plot.

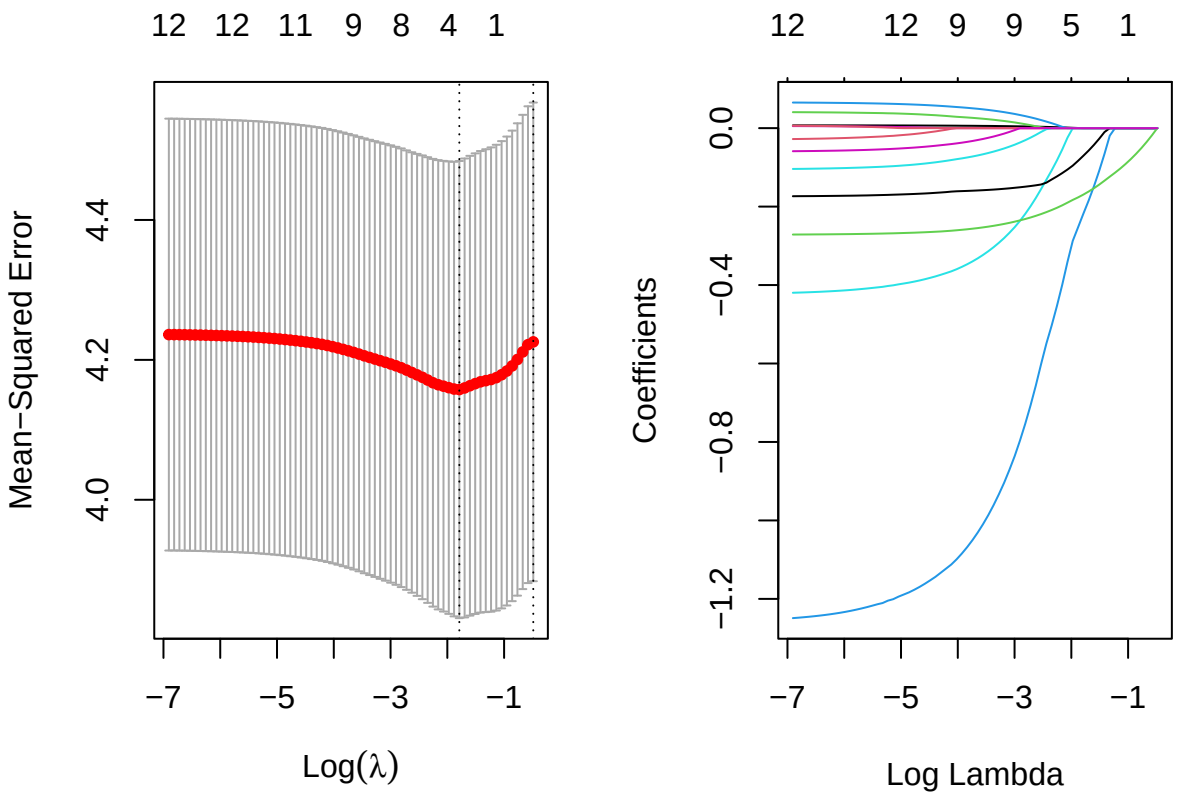
b. [10 Points] Perform Elastic-Net model on this data. Report the following:
   - How did you choose the α parameter?
   - What is the parameter estimates corresponding to the minimum cross-validation error? Is it better than Lasso?
   - Is this solution sparse? Any variable being excluded?

I chose the α parameter by plotting three different scenarios (when α is 0, 0.5, and 1). The MSE is the lowest when $\alpha = 1$. This MSE is better than Lasso, by looking at the two plots.

```r
par(mar=c(4,4,4,2))
  par(mfrow=c(1,2))
  enet.fit = cv.glmnet(data.matrix(x_transformed[, 1:12]), sepsis$Health, nfolds = 10, alpha = 0)
  plot(enet.fit)
  plot(enet.fit$glmnet.fit, "lambda")
```



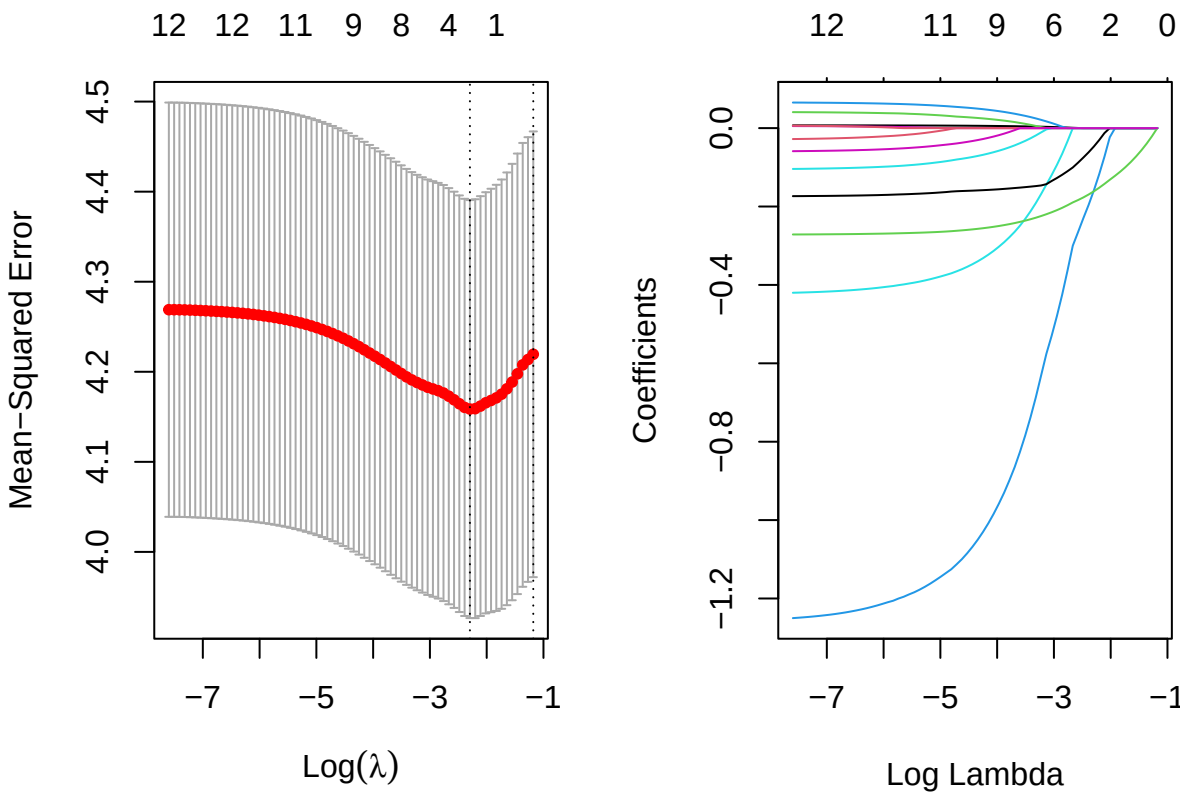

```
par(mar=c(4,4,4,2))
  par(mfrow=c(1,2))
  enet.fit = cv.glmnet(data.matrix(x_transformed[, 1:12]), sepsis$Health, nfolds = 10, alpha = 0.5)
  plot(enet.fit)
  plot(enet.fit$glmnet.fit, "lambda")
```




```
par(mar=c(4,4,4,2))
  par(mfrow=c(1,2))
  enet.fit = cv.glmnet(data.matrix(x_transformed[, 1:12]), sepsis$Health, nfolds = 10, alpha = 1)
  plot(enet.fit)
  summary(enet.fit)
```

```
##            Length Class  Mode
## lambda     70     -none- numeric
## cvm        70     -none- numeric
## cvsd       70     -none- numeric
## cvup       70     -none- numeric
## cvlo       70     -none- numeric
## nzero      70     -none- numeric
## call        5     -none- call
## name        1     -none- character
## glmnet.fit 12     elnet  list
## lambda.min  1     -none- numeric
## lambda.1se  1     -none- numeric
## index       2     -none- numeric
```

```
  plot(enet.fit$glmnet.fit, "lambda")
```



Here are the coefficient estimates according to lambda.min

```
coef(enet.fit, s = "lambda.min")
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)   1.17668679
## AGE           .
## ORGANNUM      .
## blSOFA        .
## LOG_PRAPACHE -0.15858427
## LOG_BLGCS     .
## LOG_BLIL6     .
## LOG_BLLPLAT  -0.04675446
## LOG_BLLCREAT  .
## LOG_TIMFIRST -0.16113372
## LOG_BLADL     .
## BLLBILI_NEW   .
## THERAPY_NEW   .
```

The same (number of) variables is excluded by using Elastic-Net as Lasso. This model is not as sparse as Lasso with lambda.1se as the tuning parameter. The variables being excluded are AGE, ORGANNUM, blSOFA, LOG_BLGCS, LOG_BLIL6, LOG_BLLCREAT, LOG_BLADL, BLLBILI_NEW, and THERAPY_NEW.

c. [15 Points] Provide a discussion of the three penalized models we have learned so far: Lasso, Ridge and Elastic-Net by giving at least one advantage and one disadvantage for each of them.

Model	Pro
Ridge	Ridge can deal with collinearity better than Lasso because it takes into account correlation structure

Model	Pro
Lasso	Can prevent overfitting by shrinkage (shrinking variables to 0)
Elastic-Net	Can also deal with collinearity better than Lasso, by combining both penalties

Model	Con
Ridge	No shrinkage, estimated parameters can never be 0
Lasso	The performance might suffer when two variables are highly correlated
Elastic-Net	Parameters might be trickier to tune because two hyperparameters are involved (computational cost increases in tuning)