Stat 432 Homework 4

Sharvi Tomar (stomar2)

30/08/21

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

```
sepsis = read.csv("Sepsis2.csv", row.names = 1)
```

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?

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

```
## [1] 53
```

```
as.matrix(colSums(is.na(sepsis)))
```

```
##          [,1]
## Health      0
## THERAPY     0
## PRAPACHE    0
## AGE         0
## BLGCS       3
## ORGANNUM    0
## BLIL6       0
## BLLPLAT     0
## BLLBILI    50
## BLLCREAT    0
## TIMFIRST    0
## BLADL       0
## blSOFA      0
```

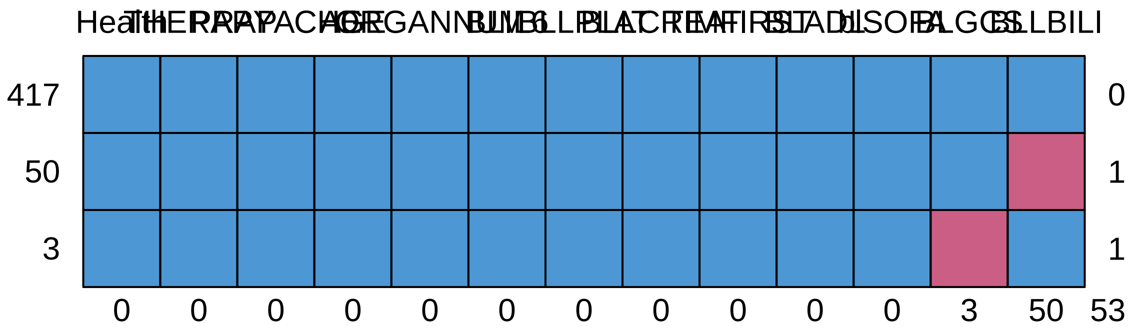
```
library(mice)
```

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

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

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

```
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
```

There are 417 observations with no missing values. 53 observations have missing values. The variable “BLGCS” has 3 missing values and “BLLBILI” has 50 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
# Method-1 Stochastic regression imputation
set.seed(667346304)
imp1 <- mice(sepsis[-c(1)],
            method = "norm.nob",
            m = 1,
            maxit = 1)
```

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

```
# Method-2 Imputation with mean
imp2 <- mice(sepsis[-c(1)],
            method = "mean",
            m = 1,
            maxit = 1)
```

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

```
# Completing the data with imputed values from the 2 methods
sepsis_imp1<-complete(imp1)
sepsis_imp2<-complete(imp2)
```

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

```
# Combining Health column from spesis data with imputed data1
sepsis_data1<-cbind(sepsis_imp1,sepsis$Health)
# Renaming 'sepsis$Health' column to 'Health'
names(sepsis_data1)[names(sepsis_data1) == "sepsis$Health"] <- "Health"

# Combining Health column from spesis data with imputed data2
sepsis_data2<-cbind(sepsis_imp2,sepsis$Health)
# Renaming 'sepsis$Health' column to 'Health'
names(sepsis_data2)[names(sepsis_data2) == "sepsis$Health"] <- "Health"

# Linear regression on sepsis_data1
model1<-lm(Health~., data=sepsis_data1)
# Linear regression on sepsis_data2
model2<-lm(Health~., data=sepsis_data2)
```

```
# Comparing model fitting results
summary(model1)
```

```
##
## Call:
## lm(formula = Health ~ ., data = sepsis_data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0348 -1.4190 -0.0262  1.2678  6.4603
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.250e-01  7.199e-01   0.590  0.55524
## THERAPY     -6.329e-02  1.913e-01  -0.331  0.74091
## PRAPACHE    -3.070e-02  1.764e-02  -1.741  0.08243 .
## AGE          6.438e-03  6.107e-03   1.054  0.29240
## BLGCS       -2.430e-02  3.119e-02  -0.779  0.43621
## ORGANNUM    -2.555e-02  1.199e-01  -0.213  0.83135
## BLIL6        9.838e-07  1.382e-06   0.712  0.47689
## BLLPLAT     -2.107e-04  7.121e-04  -0.296  0.76741
## BLLBILI     -6.199e-03  2.248e-02  -0.276  0.78285
## BLLCREAT    -8.689e-03  2.280e-02  -0.381  0.70333
## TIMFIRST    -3.265e-04  1.026e-04  -3.183  0.00156 **
## BLADL        1.474e-02  2.434e-02   0.606  0.54508
## blSOFA       3.525e-02  4.517e-02   0.780  0.43557
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.047 on 457 degrees of freedom
## Multiple R-squared:  0.03194,    Adjusted R-squared:  0.006522
## F-statistic: 1.257 on 12 and 457 DF,  p-value: 0.2416
```

```
RSS1 <- c(crossprod(model1$residuals))
MSE1 <- RSS1 / length(model1$residuals)
RMSE1 <- (MSE1)^0.5

summary(model2)
```

```
##
## Call:
## lm(formula = Health ~ ., data = sepsis_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9934 -1.3978 -0.0509  1.2717  6.4832
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.004e-01  7.195e-01   0.557  0.57813
## THERAPY     -5.960e-02  1.915e-01  -0.311  0.75578
## PRAPACHE    -2.987e-02  1.761e-02  -1.696  0.09055 .
## AGE          6.495e-03  6.109e-03   1.063  0.28832
## BLGCS       -2.342e-02  3.119e-02  -0.751  0.45309
## ORGANNUM    -1.600e-02  1.189e-01  -0.135  0.89305
## BLIL6        9.945e-07  1.383e-06   0.719  0.47244
## BLLPLAT     -2.150e-04  7.125e-04  -0.302  0.76299
## BLLBILI      1.942e-03  2.338e-02   0.083  0.93382
## BLLCREAT    -7.837e-03  2.279e-02  -0.344  0.73105
## TIMFIRST    -3.262e-04  1.026e-04  -3.180  0.00157 **
## BLADL        1.378e-02  2.430e-02   0.567  0.57103
## blSOFA       2.865e-02  4.473e-02   0.640  0.52219
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.047 on 457 degrees of freedom
## Multiple R-squared:  0.03175,    Adjusted R-squared:  0.006328
## F-statistic: 1.249 on 12 and 457 DF,  p-value: 0.2466
```

```
RSS2 <- c(crossprod(model2$residuals))
MSE2 <- RSS2 / length(model2$residuals)
RMSE2 <- (MSE2)^0.5
```

R-squared for model1(with imputed data from Stochastic Regression Imputation) is 0.03194 and for model2(with imputed data from mean) is 0.03175.

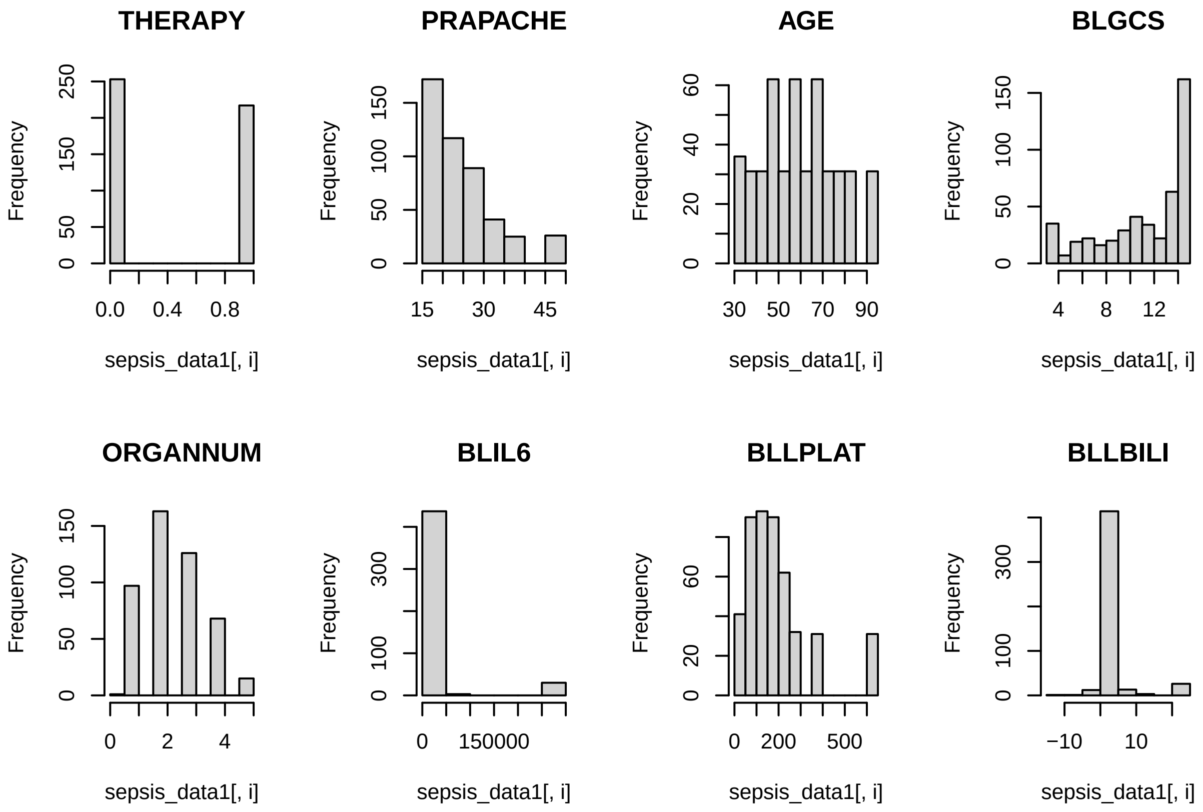
```
print(c(RMSE1, RMSE2))
```

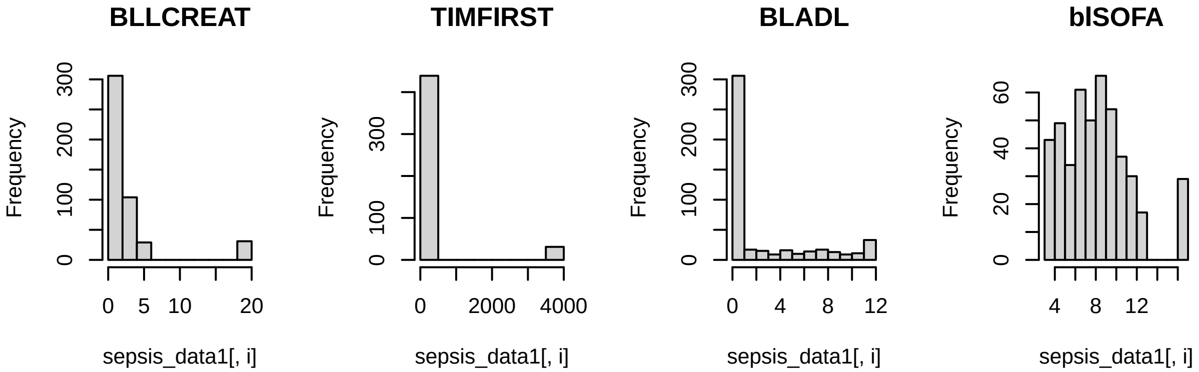
```
## [1] 2.018700 2.018896
```

RMSE for model1(with imputed data from Stochastic Regression Imputation) is 2.018700 and for model2(with imputed data from mean) is 2.018896.

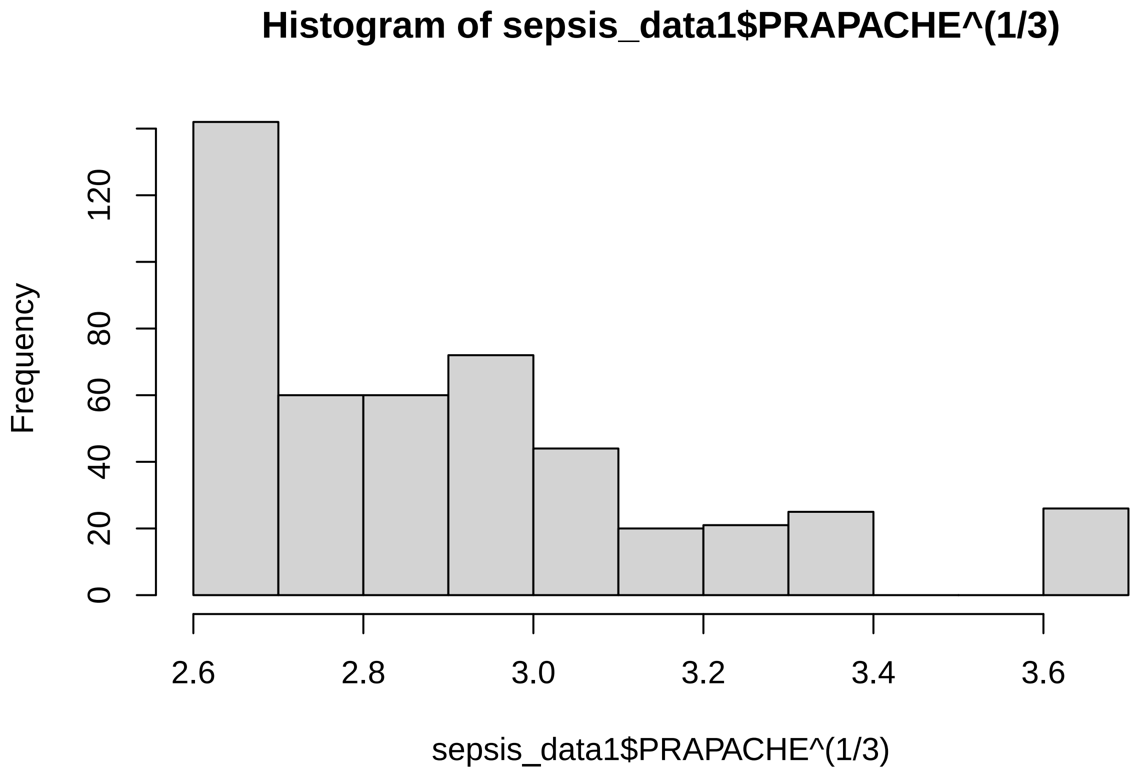
d. [20 Points] Investigate the marginal distribution of each variable (excluding the outcome `Health`) and decide whether the variable could benefit from any transformations. If so, then perform the transformation at your choice. **You need to provide clear evidence to reason your decision and also provide 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.

```
par(mfrow=c(2,4))
for(i in 1:ncol(sepsis_imp1))
  hist(sepsis_data1[,i],breaks=10,main=colnames(sepsis_imp1)[i])
```



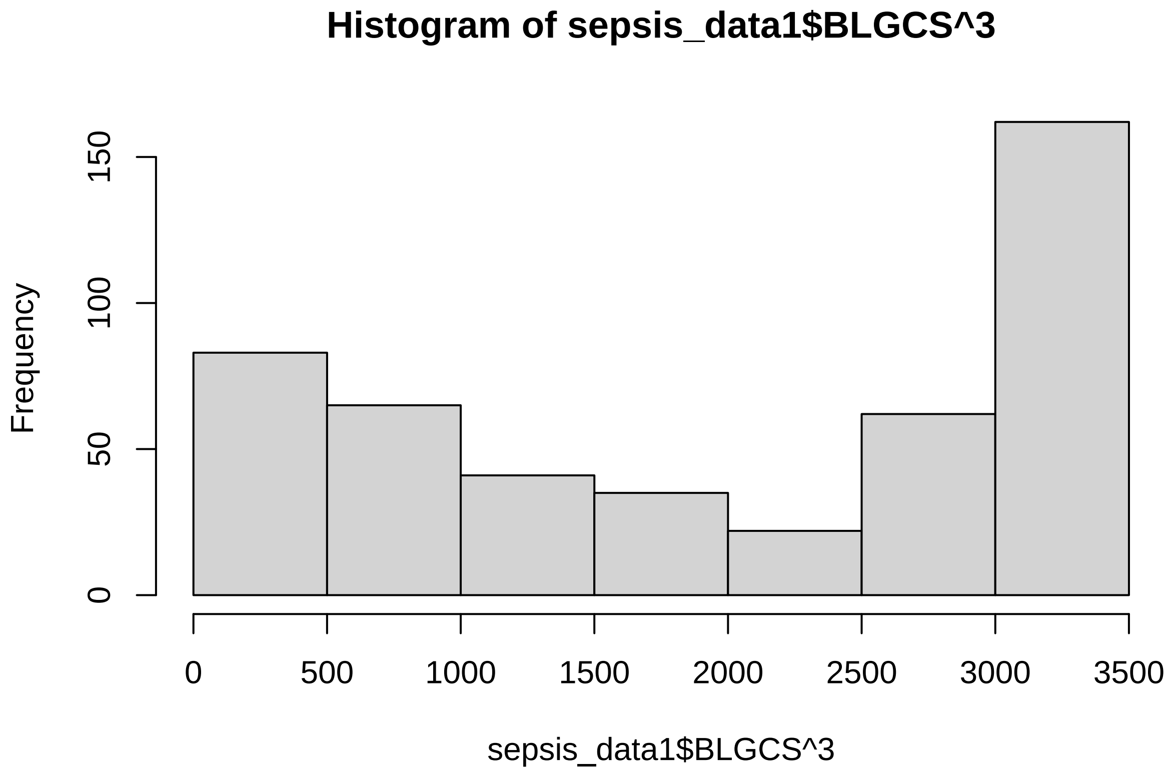


```r
# Since the data is right-skewed(clustered at lower values), deccreasing the power to 1/2, 1/3 etc.
hist(sepsis_data1$PRAPACHE^(1/3))
```



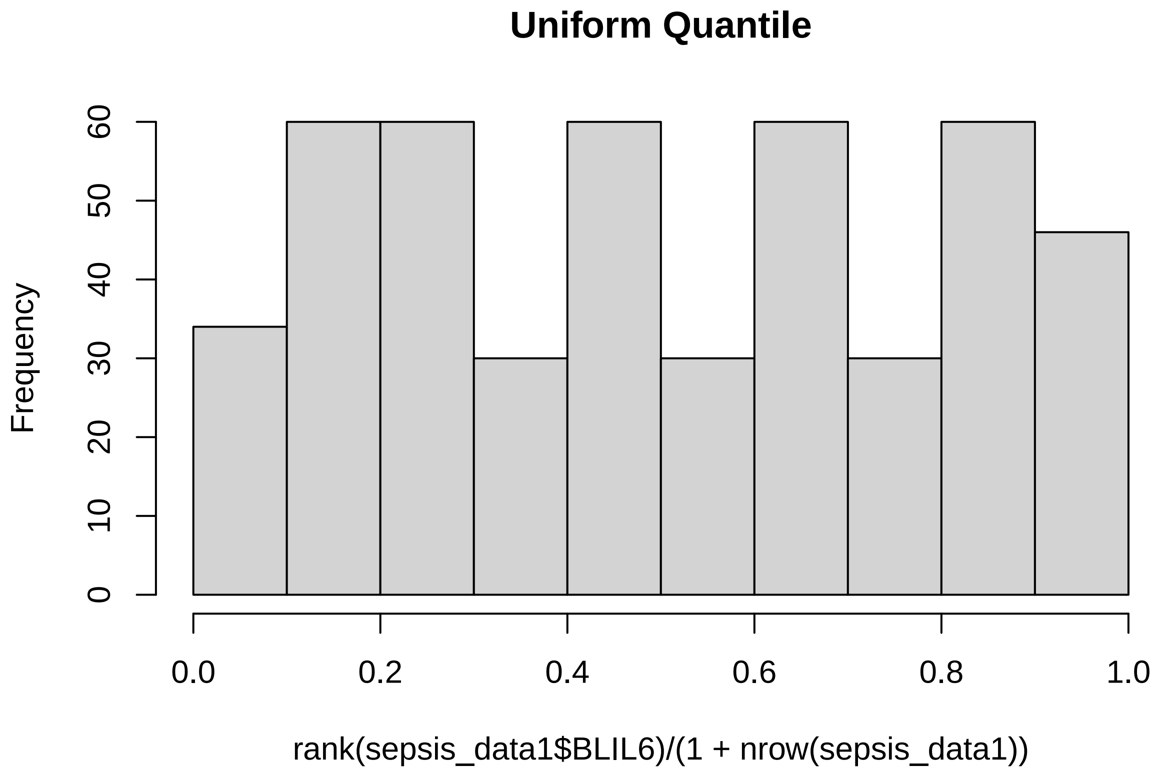
```
# Saving the transformed values of the variable PRAPACHE
transformed_PRAPACHE<-sepsis_data1$PRAPACHE^(1/3)
```

```
# Since the values for the variable BLGCS are left-skewed (clustered at higher values), increasing the
hist(sepsis_data1$BLGCS^3)
```

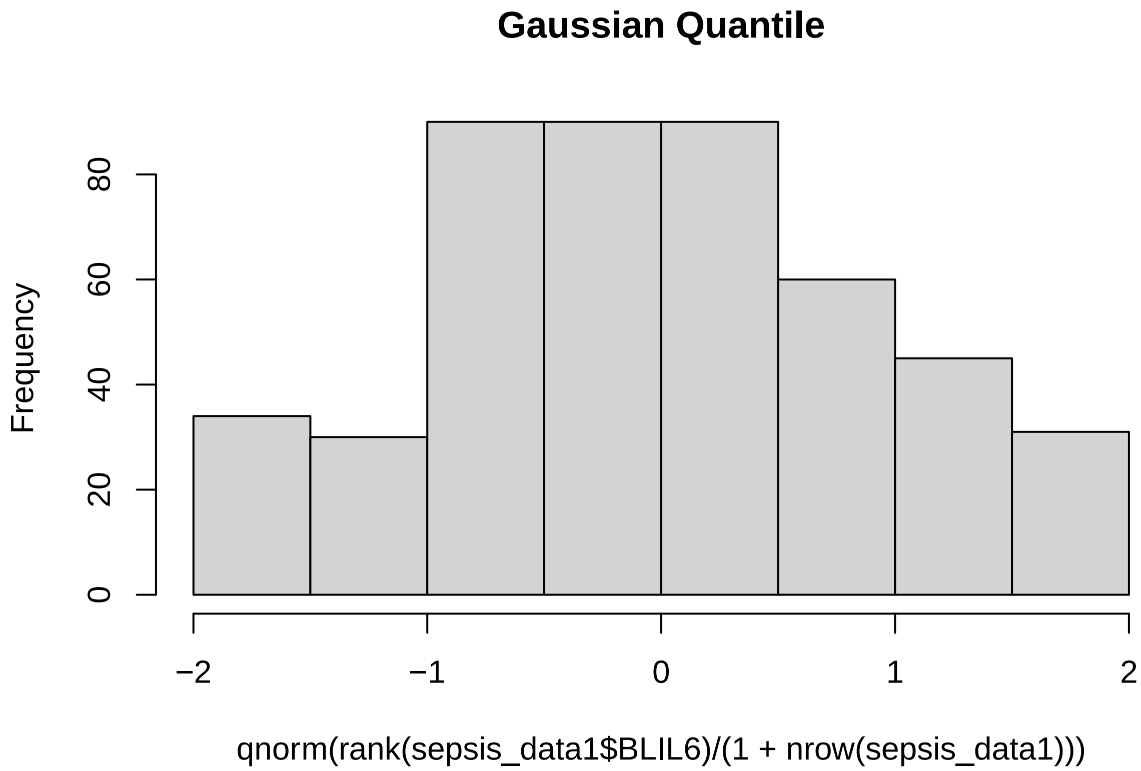


```
# Saving the transformed values of the variable BLGCS
transformed_BLGCS<-sepsis_data1$BLGCS^3
```

```
# Performing quantile transformation - as rank ranges from 1 to n, we can transform them to (0, 1)
hist(rank(sepsis_data1$BLIL6) / (1 + nrow(sepsis_data1)), main = "Uniform Quantile")
```

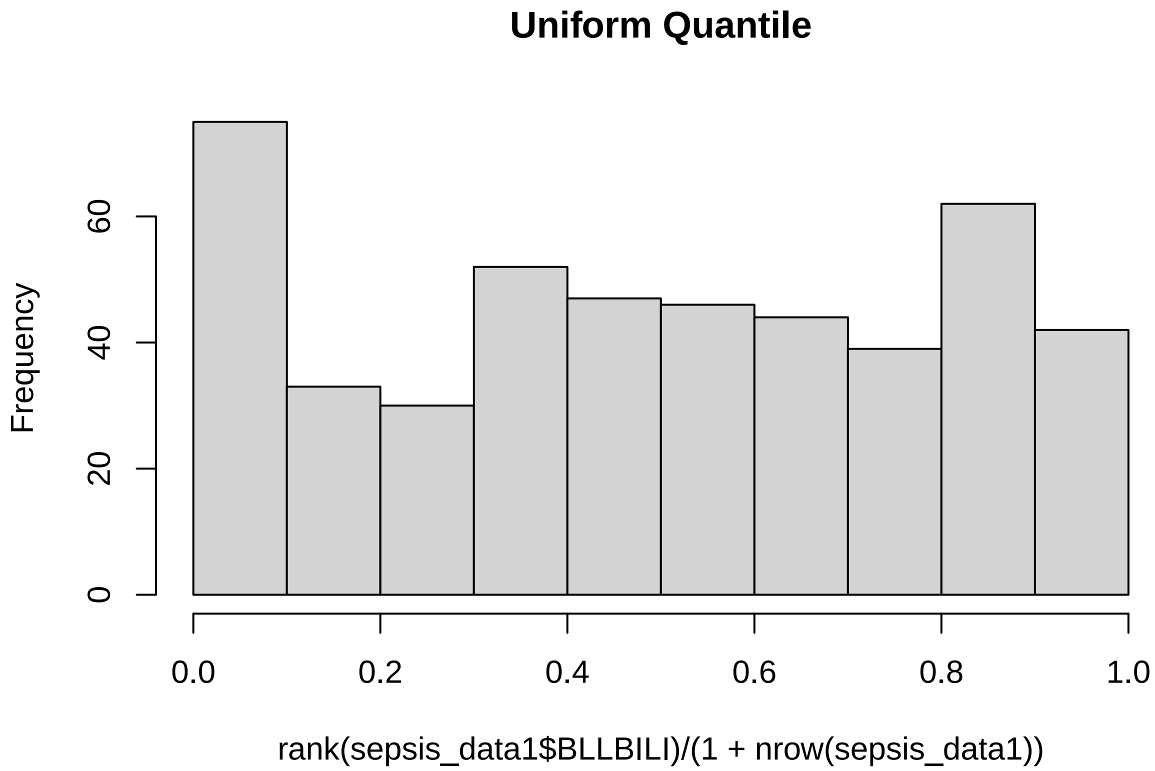


```
# Further transforming into Gaussian quantiles
hist(qnorm(rank(sepsis_data1$BLIL6) / (1 + nrow(sepsis_data1))),main = "Gaussian Quantile")
```

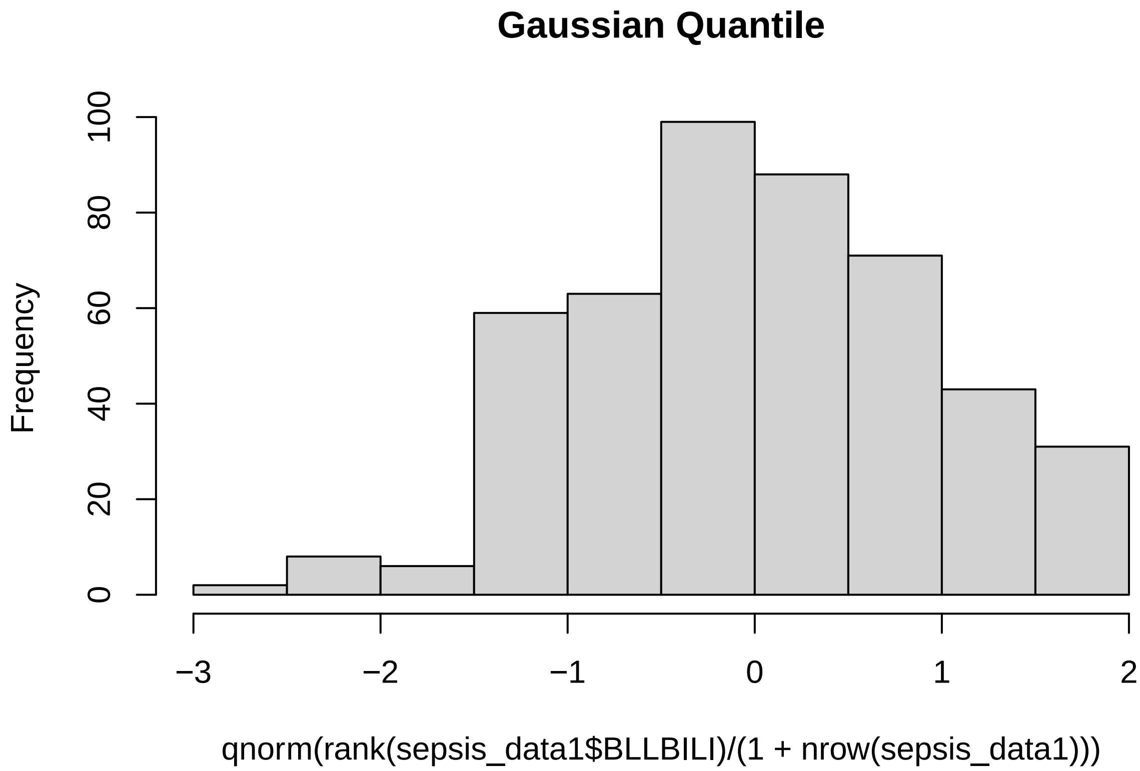


```r
# Saving the transformed values of the variable BLIL6
transformed_BLIL6<-qnorm(rank(sepsis_data1$BLIL6) / (1 + nrow(sepsis_data1)))
```

```r
# Performing quantile transformation - as rank ranges from 1 to n, we can transform them to (0, 1)
hist(rank(sepsis_data1$BLLBILI) / (1 + nrow(sepsis_data1)), main = "Uniform Quantile")
```

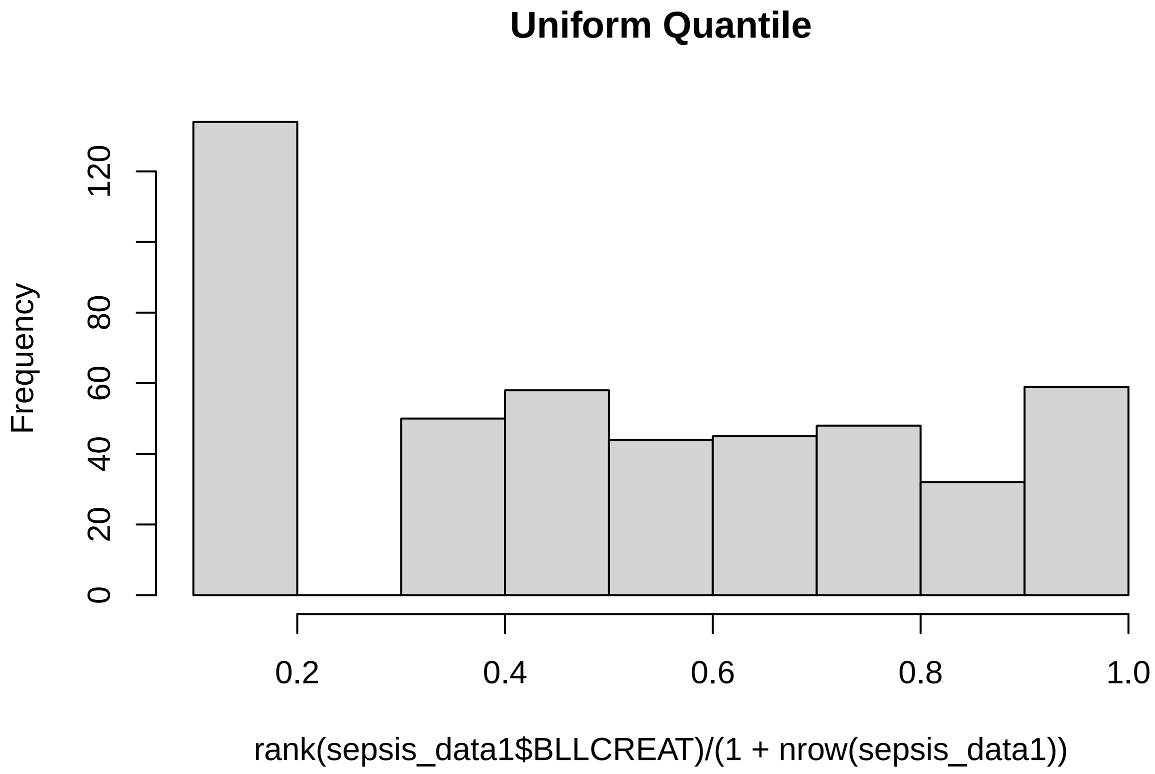


```
# Further transforming into Gaussian quantiles
hist(qnorm(rank(sepsis_data1$BLLBILI) / (1 + nrow(sepsis_data1))), main = "Gaussian Quantile")
```

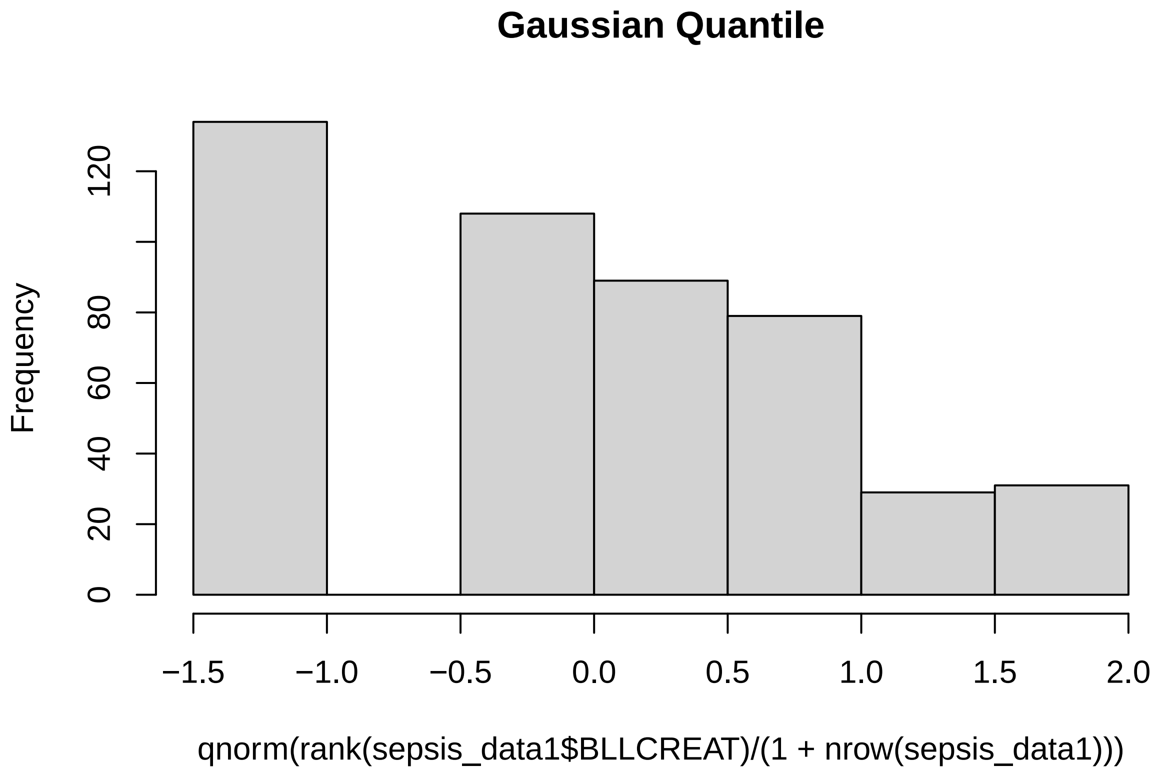


```
# Saving the transformed values of the variable BLLBILI
transformed_BLLBILI<-qnorm(rank(sepsis_data1$BLLBILI) / (1 + nrow(sepsis_data1)))
```

```
# Performing quantile transformation - as rank ranges from 1 to n, we can transform them to (0, 1)
hist(rank(sepsis_data1$BLLCREAT) / (1 + nrow(sepsis_data1)), main = "Uniform Quantile")
```

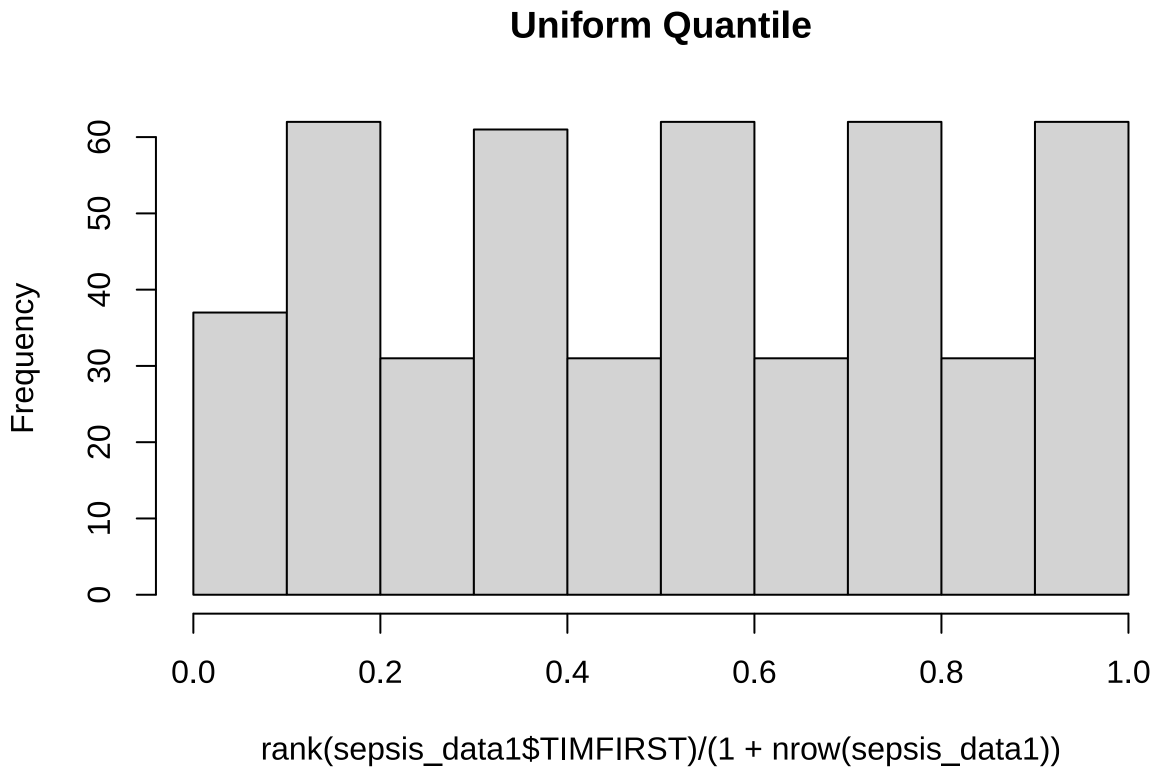


```
# Further transforming into Gaussian quantiles
hist(qnorm(rank(sepsis_data1$BLLCREAT) / (1 + nrow(sepsis_data1))), main = "Gaussian Quantile")
```

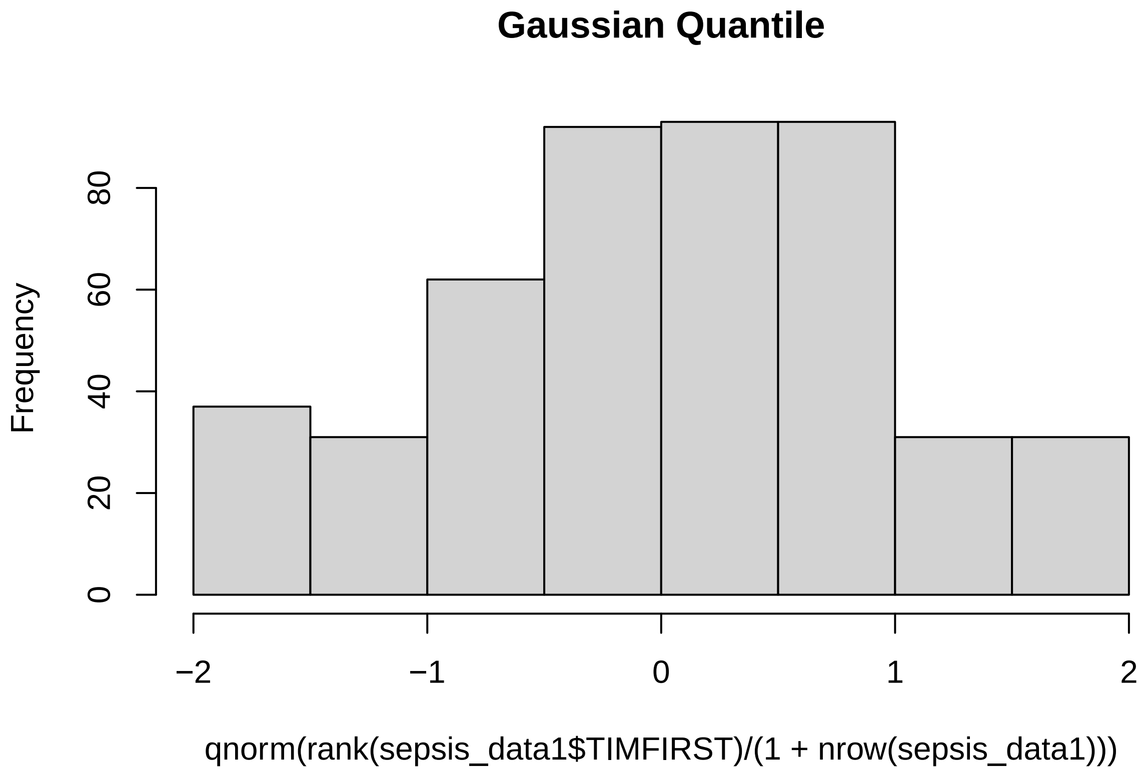


```
# Saving the transformed values of the variable BLLBILI
transformed_BLLCREAT<-qnorm(rank(sepsis_data1$BLLCREAT) / (1 + nrow(sepsis_data1)))
```

```
hist(rank(sepsis_data1$TIMFIRST) / (1 + nrow(sepsis_data1)), main = "Uniform Quantile")
```

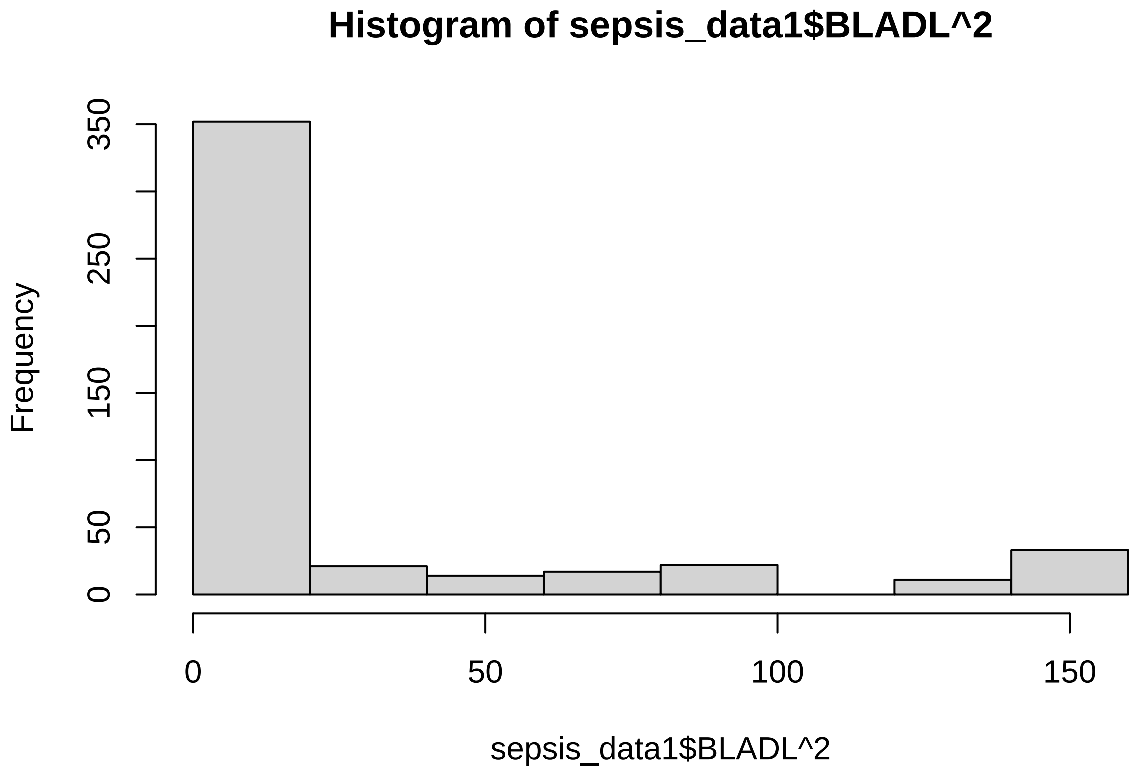


```
# this can be further transformed into Gaussian quantiles
hist(qnorm(rank(sepsis_data1$TIMFIRST) / (1 + nrow(sepsis_data1))),  main = "Gaussian Quantile")
```



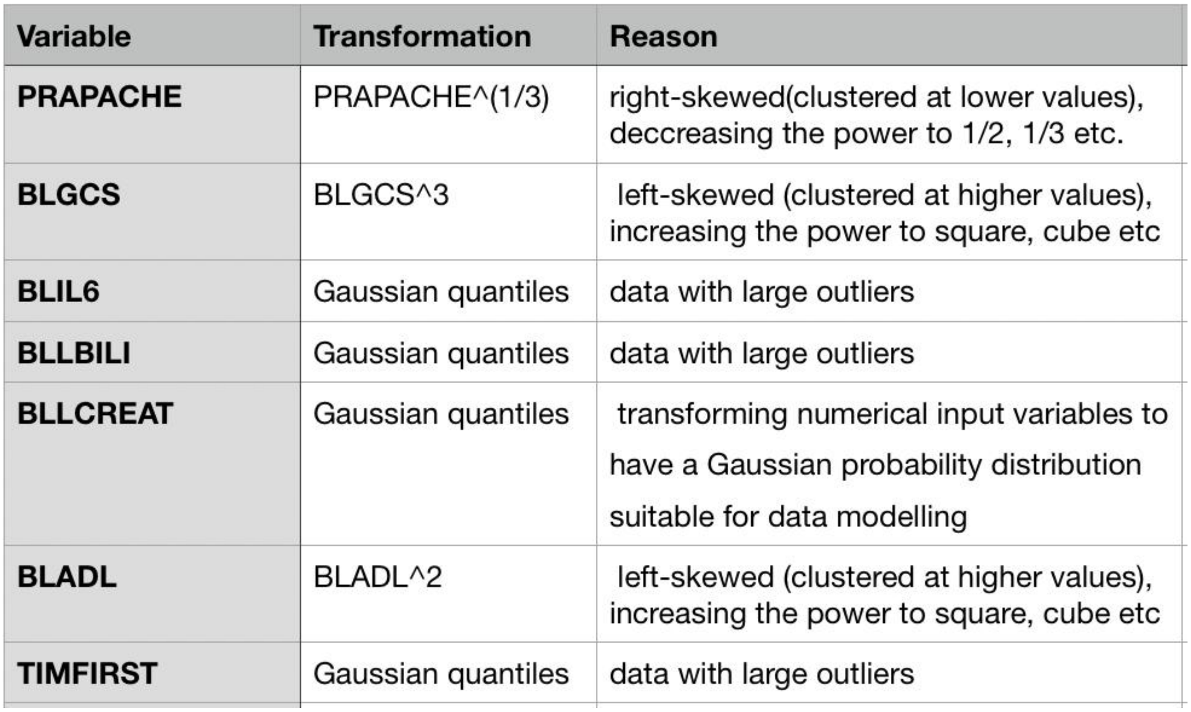
```r
transformed_TIMFIRST<-qnorm(rank(sepsis_data1$TIMFIRST) / (1 + nrow(sepsis_data1)))
```

```r
# Since the values for the variable BLGCS are left-skewed (clustered at higher values), increasing the
hist(sepsis_data1$BLADL^2)
```



```
# Saving the transformed values of the variable
transformed_BLADL<-sepsis_data1$BLADL^2
```

```
# Updating the values to the transformed values for the variable BLIL6 in the dataframe
sepsis_data1$PRAPACHE<-transformed_PRAPACHE
sepsis_data1$BLGCS<-transformed_BLGCS
sepsis_data1$BLIL6<-transformed_BLIL6
sepsis_data1$BLLBILI<-transformed_BLLBILI
sepsis_data1$BLLCREAT<-transformed_BLLCREAT
sepsis_data1$BLADL<-transformed_BLADL
sepsis_data1$TIMFIRST<-transformed_TIMFIRST
```



Variable	Transformation	Reason
PRAPACHE	PRAPACHE^(1/3)	right-skewed(clustered at lower values), deccreasing the power to 1/2, 1/3 etc.
BLGCS	BLGCS^3	left-skewed (clustered at higher values), increasing the power to square, cube etc
BLIL6	Gaussian quantiles	data with large outliers
BLLBILI	Gaussian quantiles	data with large outliers
BLLCREAT	Gaussian quantiles	transforming numerical input variables to have a Gaussian probability distribution suitable for data modelling
BLADL	BLADL^2	left-skewed (clustered at higher values), increasing the power to square, cube etc
TIMFIRST	Gaussian quantiles	data with large outliers

#

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`.

```
library(glmnet)
```

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

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

```
folds_grid<-seq(3,15,1)
folds_val=c()
min_lambda=c()
min_cv_error=c()
set.seed(1)
for(fold in folds_grid){
  lasso.fit = cv.glmnet(data.matrix(sepsis_data1[, 1:12]),
                    sepsis_data1$Health,
                    nfolds = fold,
                      alpha = 1)

  folds_val=append(folds_val,fold)
  min_lambda=append(min_lambda,lasso.fit$lambda.min)
  min_cv_error=append(min_cv_error,min(lasso.fit$cvm))
}
df_lasso<- data.frame(folds_val, min_lambda, min_cv_error)
best_lasso<-df_lasso[which.min(min_cv_error),]
```

- How many fold are you using in the cross-validation?

```
best_lasso$folds_val
```

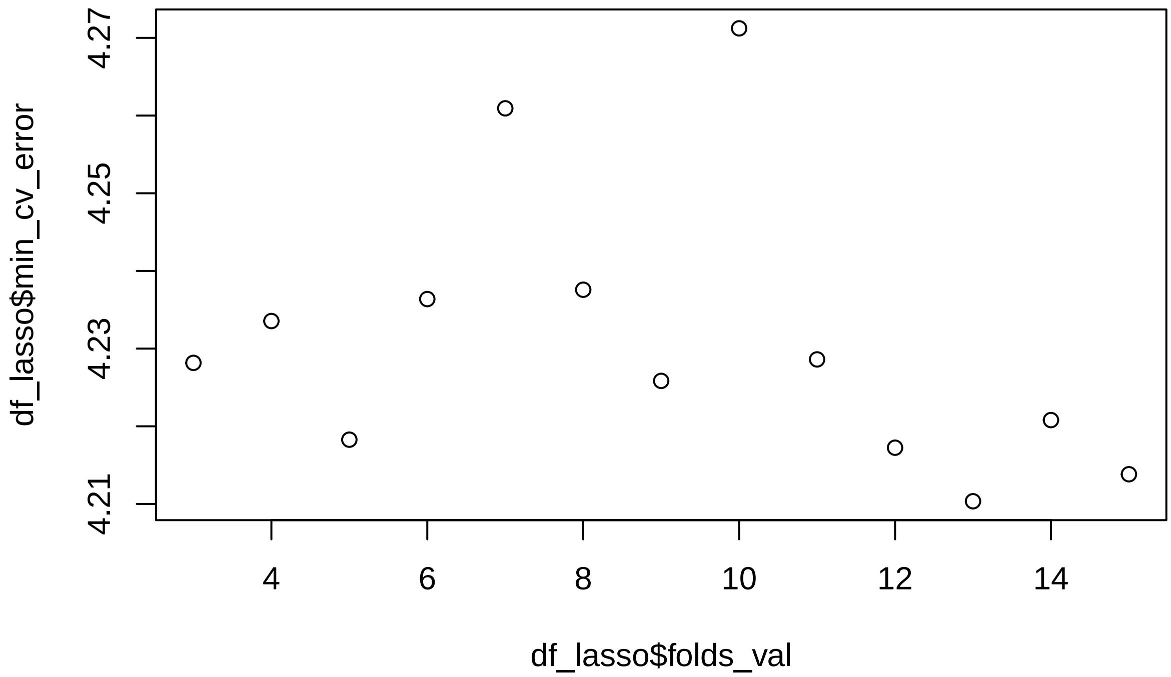
```
## [1] 13
```

I have used 13-fold cross-validation to perform Lasso.

- How did you decide which is the best tuning parameter? Please provide figures to support your answer.

Since the value of alpha is fixed for the Lasso α=1, we tune the parameter by creating a grid of fold_values for the parameter 'nfold' and calculate the cross-validation error for each value and select the one with least cross-validation error.

```
plot(x=df_lasso$folds_val,y=df_lasso$min_cv_error)
```



The plot demonstrates how “nfolds”(here, folds_val) changes the cross-validation error. The “nfolds”(here, folds_val) value corresponding to the minimum cross-validation error is selected as the best tuning parameter.

- What is the parameter estimates corresponding to this parameter? Is this solution sparse? Which variable is being excluded?

```r
print(c(best_lasso$min_lambda,best_lasso$folds_val,best_lasso$min_cv_error))
```

```
## [1]  0.1296639 13.0000000  4.2103495
```

The parameter estimate corresponding to the best tuning parameter are cross-validation error=4.2103495, lambda=0.1296639 and nfolds=13.

```r
lasso.best=cv.glmnet(data.matrix(sepsis_data1[, 1:12]),
                      sepsis_data1$Health,
                      nfolds = best_lasso$folds_val,
                      alpha = 1)
coef(lasso.best, s="lambda.min")
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -0.15493035
## THERAPY      .
## PRAPACHE     .
## AGE          .
## BLGCS        .
## ORGANNUM     .
## BLIL6        .
## BLLPLAT      .
## BLLBILI      .
## BLLCREAT     .
## TIMFIRST    -0.08863332
## BLADL        .
## blSOFA       .
```

Yes, the solution is sparse as it contains many nonzero parameters.

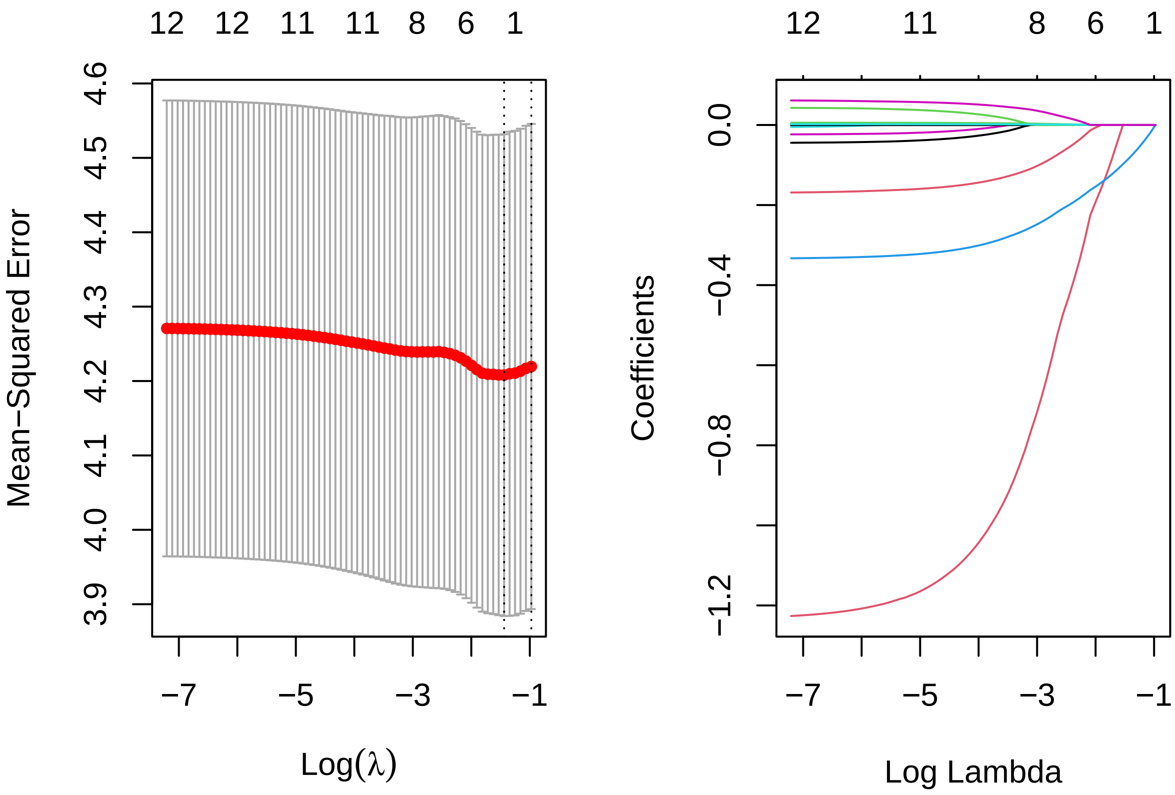
Variables being excluded are: THERAPY. PRAPACHE, AGE, BLGCS, ORGANNUM, BLIL6, BLLPLAT, BLLBILI, BLLCREAT, BLADL, blSOFA

b. [10 Points] Perform Elastic-Net model on this data. Report the following:

```r
# create a grid of   and calculate the cross-validation error for each combination and select the best c

alpha_grid=seq(0.1,1,by=0.1)
alpha_val=c()
min_lambda=c()
min_cv_error=c()


for(a in alpha_grid){
  enet.fit=cv.glmnet(data.matrix(sepsis_data1[, 1:12]),sepsis_data1$Health,nfolds=10,alpha=a)
  alpha_val=append(alpha_val,a)
  min_lambda=append(min_lambda,enet.fit$lambda.min)
  min_cv_error=append(min_cv_error,min(enet.fit$cvm))
}

df<- data.frame(alpha_val, min_lambda, min_cv_error)
best<-df[which.min(min_cv_error),]

enet.best=cv.glmnet(data.matrix(sepsis_data1[, 1:12]),sepsis_data1$Health,nfolds=10,alpha=best$alpha_val
```

- How did you choose the α parameter? Create a grid of combination of (λ,α) and calculated the cross-validation error for each combination and select the best one which had the least cross-validation error.

```
par(mar=c(4,4,4,2))
par(mfrow=c(1,2))
plot(enet.best)
plot(enet.best$glmnet.fit, "lambda")
```



- What is the parameter estimates corresponding to the minimum cross-validation error? Is it better than Lasso?

```
best$min_lambda
```

```
## [1] 0.2161064
```

```
best$alpha_val
```

```
## [1] 0.6
```

```
best$min_cv_error
```

```
## [1] 4.196866
```

```
best_lasso$min_cv_error
```

```
## [1] 4.210349
```

The parameter estimate corresponding to the minimum cross-validation error is 0.2953622(lambda), alpha_val=0.4, minimum cross-validation error=4.201605

The minimum cross-validation error obtained for Elastic net is 4.201605 and is slightly better than the Lasso solution for which the minimum cross-validation error is 4.210349

- Is this solution sparse? Any variable being excluded?

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

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) -0.15493533
## THERAPY      .
## PRAPACHE     .
## AGE          .
## BLGCS        .
## ORGANNUM     .
## BLIL6        .
## BLLPLAT      .
## BLLBILI      .
## BLLCREAT     .
## TIMFIRST    -0.08471611
## BLADL        .
## blSOFA       .
```

Yes, the solution is sparse as it contains many nonzero parameters.

Variables being excluded are: THERAPY. PRAPACHE, AGE, BLGCS, ORGANNUM, BLIL6, BLLPLAT, BLLBILI, BLLCREAT, BLADL, blSOFA,TIMFIRST

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.

All 3 penalized models- Elastic-Net, Lasso, Ridge Regularization favor simpler models to more complex models to prevent the model from overfitting to the data. They address the following concerns within a model: variance-bias tradeoff, multicollinearity, sparse data handling(i.e. the situation where there are more observations than features), feature selection, and an easier interpretation of the output.

Lasso:

Advantage: The model can be sparse, with some parameter(small ˆj values) estimates getting shrunk to exactly 0. This may prevents over-fitting and also improve the interpretability especially when the number of variables is large. This is called the variable selection property since only the nonzero parameters are selected be useful

Disadvantage: LASSO can not do group selection. If there is a group of variables among which the pairwise correlations are very high, then the LASSO tends to arbitrarily select only one variable from the group. Group selection is important, for example, in gene selection problems. Lasso does not work well with multicollinearity. Lasso might randomly choose one of the multicollinear variables without understanding the context. Such an action might eliminate relevant independent variables.

Ridge:

Advantage: Good at improving the least-squares estimate when there is multicollinearity.

Disadvantage: Does not reduce the parameter to zero. It includes all the predictors in the final model.

Elastic-Net

Advantage: Elastic Net combines feature elimination from Lasso and feature coefficient reduction from the Ridge model to improve model's predictions.

Disadvantage: One disadvantage is the computational cost. We need to cross-validate the relative weight of L1 vs. L2 penalty, α, and that increases the computational cost by the number of values in the αgrid.