Multiple Imputation and Cross-Validation for Classification of Survival Prediction

Robert Edwards

(2416963E)

MASTER THESIS

Biostatistics

Contents

List of Figures

List of Tables

1 Introduction

1.1 Aim of the Thesis

- paragraph about importance of in-sample vs. out-of-sample prediction accuracy
- Cross validation
- Over fitting / under fitting
- Paragraph about Missing data

1.2 The Clinical Study

1.3 Study Population & Data Description

1.4 The Statistical Challenege

2 Methodology

2.1 Basic Statistical Methods

2.1.1 Logistic Regression

Logistic regression is a widely used approach in binary classification. It is set up as a generalised linear model using a logit link that produces a probability.

2.1.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a widely used method classification method. generlization of Fisher's Linear Discriminant **(Fisher 1936)**. Ddiscriminant functions are created through a linear combination of the explanatory variables that characterize the classes.

$$\Pr(C_g \mid \mathbf{x}) = \frac{\pi_g \exp(-\frac{1}{2} d_g(\mathbf{x}))}{\sum_{i=1}^{2} \pi_g \exp(-\frac{1}{2} d_i(\mathbf{x}))} \qquad g = 1, 2$$

Assumptions of LDA:

- Explanatory variables are assumed to be normally distributed
- Homoskedasticity, equal class covariances
- No multicollinearity
- Independent observations

Drawbacks of LDA:

- Can only utilize continuous explanatory variables
- Cannot handle missing data

2.1.3 Quadratic Discriminant Analysis

(Cover 1965)

Quadratic Discriminant Analysis (QDA) is an even more generlized form of discriminant analysis than LDA. QDA has the same assumptions as LDA with the exception that the covariance of each class is not assumed to be identical.

Assumptions of QDA:

- Explanatory variables are assumed to be normally distributed
- No multicollinearity
- Independent observations

Drawbacks of QDA:

- Can only utilize continuous explanatory variables
- Cannot handle missing data

2.1.4 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a commonly used non-parametric classification method. To predict the class of a new observation, a distance matrix is constructed between all observations and the K nearest labelled observations to the new observation are considered. The new observation is then assigned the class label that the majority of its neighbors share. In case of only two classes, ties in class assignments are avoided by using odd values of K.

In the event of a tie, a class can be chosen at random. Various distance metrics may be used but it is common to use Euclidean distance to determine the closest training points, though it is advisable to scale variables so that one direction does not dominate the classification.

As K increases, the variability of the classification tends to decrease at the expense of increased bias.

2.1.5 Random Forests

2.2 Validation & Cross-validation

2.3 Accuracy Metrics

These are the default metrics used to evaluate algorithms on binary and multi-class classification datasets in caret.

2.3.1 Accuracy

Accuracy is the percentage of correctly classifies instances out of all instances. It is more useful on a binary classification than multi-class classification problems because it can be less clear exactly how the accuracy breaks down across those classes (e.g. you need to go deeper with a confusion matrix). Learn more about Accuracy here.

Don't use accuracy (or error rate) to evaluate your classifier! There are two significant problems with it. Accuracy applies a naive 0.50 threshold to decide between classes, and this is usually wrong when the classes are imbalanced. Second, classification accuracy is based on a simple count of the errors, and you should know more than this. You

should know which classes are being confused and where (top end of scores, bottom end, throughout?)

Table 1: Confusion matrix for two classes.

	Y	N
Y	a	b
N	c	d

For the two class confusion matrix in 1 accuracy is defined as:

$$\text{accuracy} = \frac{a+d}{a+b+c+d}$$

2.3.2 ROC

To mention or not to mention(??)

2.3.3 Cohen's Kappa

Kappa or Cohen's Kappa is like classification accuracy, except that it is normalized at the baseline of random chance on your dataset. It is a more useful measure to use on problems that have an imbalance in the classes On the ARDS datasets, for example, if `ECMO_Survival` is predicted to be "Y" for all cases, then the accuracy is 75% but the prediction is no better than the baseline likelihood of the class percentages.

Let p_o be the accuracy, the relative observed agreement between observed and predicted classes and let p_e be the probability of chance agreement based on the class probabilities. Cohen's Kappa is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

If all the observations are predicted correctly then $\kappa = 1$. **If the observations are predicted no better than expected by the class probabilities, p_e then $\kappa = 0$. If all the observations are predicted incorrectly, then $\kappa = -1$.** A positive κ indicates that the model predicts better than would be expected by chance whereas a negative κ indicates that the model predicts worse than would be expected by chance.

$$p_o = \frac{a+d}{a+b+c+d}$$

For class k, number of items N and n_{ki}, the number of times i is predicted as class k:

$$p_{o,Y} = \hat{p}_{k1} = \frac{a+d}{a+b+c+d} \cdot \frac{a+c}{a+b+c+d}$$

$$p_{o,N} = \hat{p}_{k2} = \frac{c+d}{a+b+c+d} \cdot \frac{b+d}{a+b+c+d}$$

$$p_e = p_{o,Y} + p_{o,N} = \sum_k \hat{p}_{k1}\hat{p}_{k2} = \sum_k \frac{n_{k1}}{N}\frac{n_{k2}}{N} = \frac{1}{N^2}\sum_k n_{k1}n_{k2}$$

2.3.4 Brier Score

2.3.5 F1 Score

3 Statistical Methods for the Analysis

Describe the methods step-by-step for the analysis

3.1 Missing Data

3.2 Multiple Imputation

3.3 Complete Case Analysis

3.4 Mean Imputation

3.5 Multiple Imputation

3.5.1 Joint-Model

3.5.2 Fully Conditional Specification

3.5.3 Predictive Mean Matching

Predictive Mean Matching (PMM) is a semi-parametric imputation approach. It is similar to the regression method except that for each missing value, it fills in a value randomly from among the a observed donor values from an observation whose regression-predicted values are closest to the regression-predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996). The PMM method ensures that imputed values are plausible; it might be more appropriate than the regression method (which assumes a joint multivariate normal distribution) if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Ensemble Multiple Imputation
The steps in the ensemble approach for multiply imputed data in k-fold cross-validation are as follows:

1. Randomly partition the training data into k folds
2. Define the k^{th} as the test set and the remaining $k-1$ folds as the training set
3. Impute the training set m times, with the response variable `ECMO_Survival` included, to create m imputed training sets
4. Concatenate the m imputed training sets into one extended training set
5. A model is fitted to the extended training set
6. The test set is concatenated with the extended training set

7. Impute the combined test and extended training set, with the response variable `ECMO_Survival` excluded, to create m imputed combined test and extended training sets
8. Extract the m test sets
9. Make m predictions on the m imputed test sets
10. Take the majority vote of the m predictions as the prediction for the fitted model
11. Validate the prediction against the test set by calculating Cohen's Kappa (note there are no missing values for the response variable in the data)
12. Repeat steps 2-11 k times and validate the fitted model on each training set against the test set for each fold
13. Average the k calculated Cohen's Kappas as the estimated in-sample accuracy metric

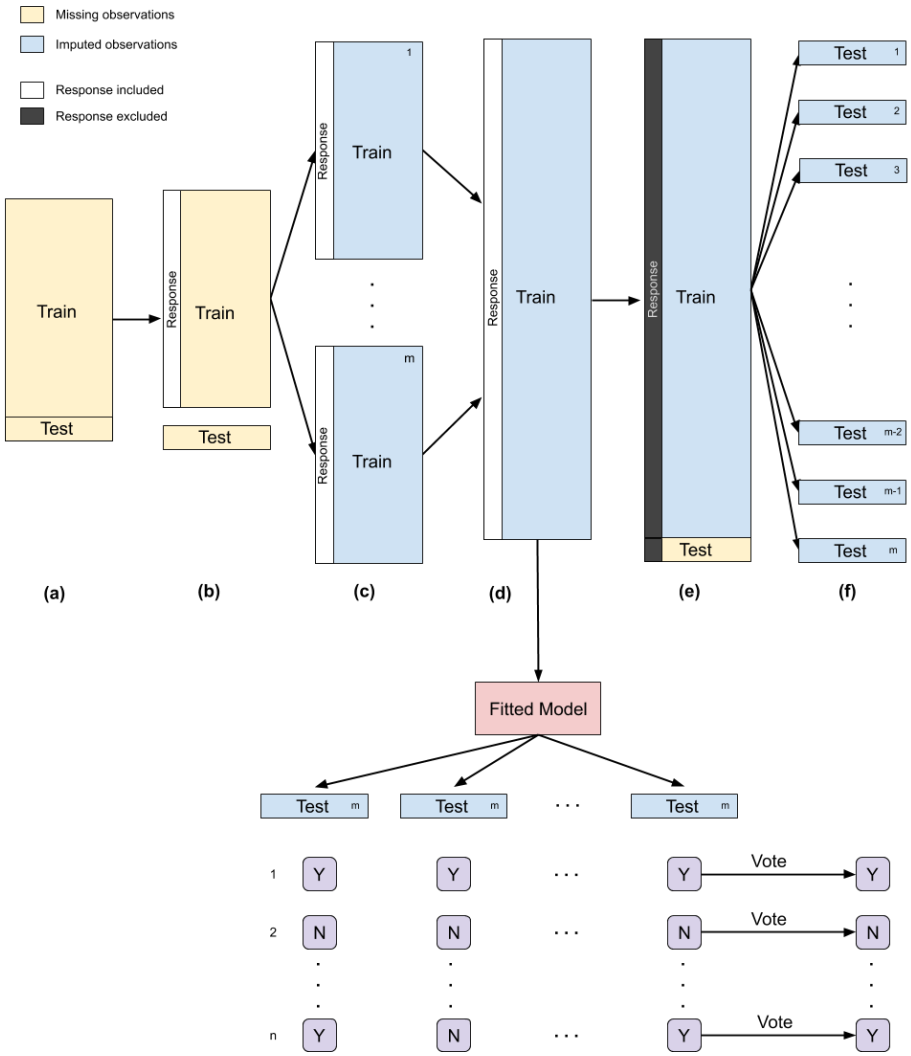

Figure 1: Outline of the algorithm used to pool predictions from multiple imputation. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4. (e) Step 5. (f) Step 6.

3.5.4 Number of Imputations

- Rubin's Rule
- Sample size calculator

$$\left(\hat{p} - 1.96\sqrt{\frac{0.25}{n}}, \quad \hat{p} + 1.96\sqrt{\frac{0.25}{n}}\right)$$

Flexible Imputation Book

"The classic advice is to use a low number of imputation, somewhere between 3 and 5 for moderate amounts of missing information. Several authors investigated the influence of m on various aspects of the results. The picture emerging from this work is that it is often beneficial to set m higher, somewhere in the range of 20-100 imputations.

Theoretically it is always better to use higher m, but this involves more computation and storage. Setting m very high (say m=200) may be useful for low-level estimands that are very uncertain, and for which we want to approximate the full distribution, or for parameters that are notoriously different to estimates, like variance components. On the other hand, setting m high may not be worth the extra wait if the primary interest is on the point estimates (and not on standard errors, p-values, and so on). In that case using m=5-20 will be enough under moderate missingness."

- Rubin's Rules allow the pooling of parameter estimates in GLMs but...
- To my knowledge, there has been insufficient work on estimating the required number of imputations for pooling posterior probabilities in classification problems.

- Cite the Dutch Master Thesis
- Adapt Rubin's Rules - arguing that

3.6 Voting

3.6.1 Majority Vote

The combination can be implemented using a variety of strategies, among which majority vote is by far the simplest, yet it has been found to be just as effective as more complicated schemes. (Lam and Suen, 1994).

(Alexandre et al. 2001) There has been some interest on the comparative performance of the sum and product rules (or the arithmetic and geometric means) (Kittler et al., 1996; Tax et al., 1997; Kittler et al., 1998). The arithmetic mean is one of the most frequently used combination rules since it is easy to implement and normally produces good results.

In (Kittler et al., 1998), the authors show that for combination rules based on the sum, such as the arithmetic mean, and for the case of classifiers working in different feature spaces, the arithmetic mean is less sensitive to errors than geometric mean.

In fact (Alexandre et al. 2001) show that for classification problems with two classes, that give estimates of the a posteriori probabilities that sum to one the combination rules

arithmetic mean (or the sum) and the geometric mean (or the product) are equivalent.

Result: Write here the result
Input : Write here the input
Output: Write here the output

```
while While condition do
    instructions
    if condition then
        instructions1
        instructions2
    else
        instructions3
    end
end
```

Algorithm 1: While loop with If/Else condition

4 Results

4.1 Exploratory Data Analysis

- describe the data:
- 3 Categorical variables
- 30 continuous variables
- Violin plots in appendix

4.2 Missing Data Patterns

Before imputation, and indeed multiple imputation, it is important to inspect the missingness patterns in the data and check assumptions. Figure 2 shows the missingness patterns in the dataset, where a black bar represents a missing value. Table **??** provides some measures about variable dependence in the dataset. The first row shows the probability of observed values for each variable. The following are coefficients that give insight into how the variables are connected in terms of missingness. **Influx** is the ratio of the number of variables pairs $(Y_j,\ Y_k)$ with Y_j missing and Y_k observed, divided by the total number of observed data. For a variable that is entirely missing, influx is 1, and 0 for if the variable is complete. **Outflux** is defined in the opposit manner, by dividing the number of pairs $(Y_j,\ Y_k)$ with Y_j observed and Y_k missing, by the total number of complete cells. For a completely observed variable, outflux will have a value of 1 and 0 if completely missing. Outflux gives an indication of how useful the variable will be for imputing other variables in the dataset, while influx is an indicator for how easily the variable can be imputed. We see that **all variables are useful except XXX**. A high outflux variable might turn out to be useless for the imputation procedure if it is unrelated to the incomplete variables, while the usefulness of a highly predictive variables is severely limited by a low outflux value (Van Buuren 2012).

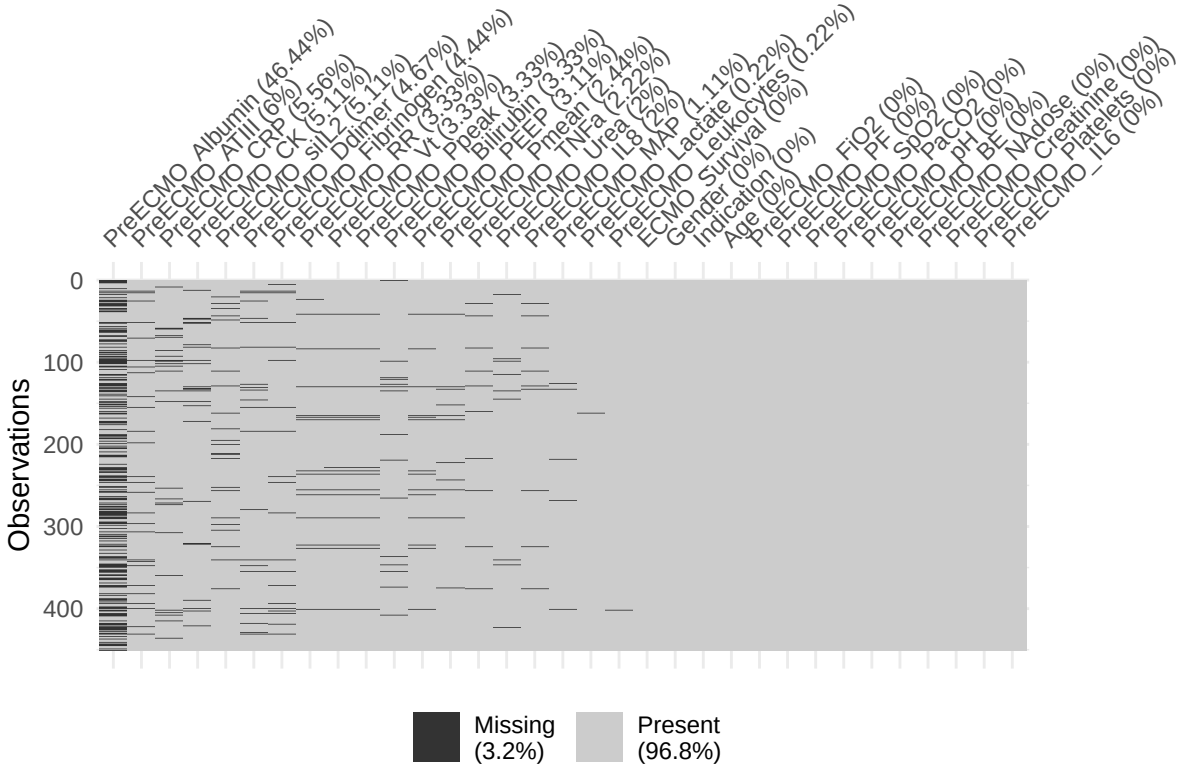

Figure 2: Visual representation of missing observations in the ARDS dataset.

Table 2: Missing pattern statistics for variables in dataset.

	Proportion	Influx	Outflux
ECMO_Survival	1.00	0.00	1.00
Gender	1.00	0.00	1.00
Indication	1.00	0.00	1.00
Age	1.00	0.00	1.00
PreECMO_RR	0.97	0.03	0.85
PreECMO_Vt	0.97	0.03	0.85
PreECMO_FiO2	1.00	0.00	1.00
PreECMO_Ppeak	0.97	0.03	0.85
PreECMO_Pmean	0.98	0.02	0.90
PreECMO_PEEP	0.97	0.03	0.85
PreECMO_PF	1.00	0.00	1.00
PreECMO_SpO2	1.00	0.00	1.00
PreECMO_PaCO2	1.00	0.00	1.00
PreECMO_pH	1.00	0.00	1.00
PreECMO_BE	1.00	0.00	1.00
PreECMO_Lactate	1.00	0.00	0.99
PreECMO_NAdose	1.00	0.00	1.00
PreECMO_MAP	0.99	0.01	0.97
PreECMO_Creatinine	1.00	0.00	1.00
PreECMO_Urea	0.98	0.02	0.94
PreECMO_CK	0.95	0.05	0.87
PreECMO_Bilirubin	0.97	0.03	0.91
PreECMO_Albumin	**0.54**	**0.46**	**0.26**
PreECMO_CRP	0.94	0.05	0.88
PreECMO_Fibrinogen	0.96	0.04	0.85
PreECMO_Ddimer	0.95	0.04	0.86
PreECMO_ATIII	0.94	0.06	0.84
PreECMO_Leukocytes	1.00	0.00	0.99
PreECMO_Platelets	1.00	0.00	1.00
PreECMO_TNFa	0.98	0.02	0.93
PreECMO_IL6	1.00	0.00	1.00
PreECMO_IL8	0.98	0.02	0.93
PreECMO_siIL2	0.95	0.05	0.87

Table 3: Averaged Cohen's Kappa for each model fitted in cross-validation. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Logit	LDA	QDA	KNN	RF
Complete Case	0.139	0.205	0.038	0.053	0.035
Mean	0.191	0.220	0.040	0.136	0.085
PMM	0.179	0.124	0.106	0.088	0.136

5 Discussion

Table 4: Complete case analysis accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=11, respectively.

	Accuracy	Kappa	Sensitivity	Specificity	Precision	Recall
Logit	0.658	0.015	0.20	0.814	0.267	0.20
LDA	0.684	0.054	0.20	0.847	0.308	0.20
QDA	0.722	-0.048	0.00	0.966	0.000	0.00
KNN	0.709	0.161	0.30	0.847	0.400	0.30
RF	0.734	0.022	0.05	0.966	0.333	0.05

Table 5: Mean imputation accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=11, respectively.

	Accuracy	Kappa	Sensitivity	Specificity	Precision	Recall
Logit	0.723	0.147	0.259	0.871	0.389	0.259
LDA	0.705	0.091	0.222	0.859	0.333	0.222
QDA	0.714	0.021	0.111	0.906	0.273	0.111
KNN	0.696	0.102	0.259	0.835	0.333	0.259
RF	0.759	0.071	0.074	0.976	0.500	0.074

Table 6: Predictively mean-matching imputation accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Accuracy	Kappa	Sensitivity	Specificity	Precision	Recall
Logit	0.732	0.162	0.259	0.882	0.412	0.259
LDA	0.732	0.162	0.259	0.882	0.412	0.259
QDA	0.714	0.021	0.111	0.906	0.273	0.111
KNN	0.714	0.131	0.259	0.859	0.368	0.259
RF	0.741	0.037	0.074	0.953	0.333	0.074

Table 7: 99 Predictively mean-matching imputation accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Accuracy	Kappa	Sensitivity	Specificity	Precision	Recall
Logit	0.741	0.178	0.259	0.894	0.438	0.259
LDA	0.741	0.202	0.296	0.882	0.444	0.296
QDA	0.714	0.021	0.111	0.906	0.273	0.111
KNN	0.714	0.106	0.222	0.871	0.353	0.222
RF	0.750	0.116	0.148	0.941	0.444	0.148

6 Conclusion

7 Bibliography

8 Appendices

8.1 Additional Material

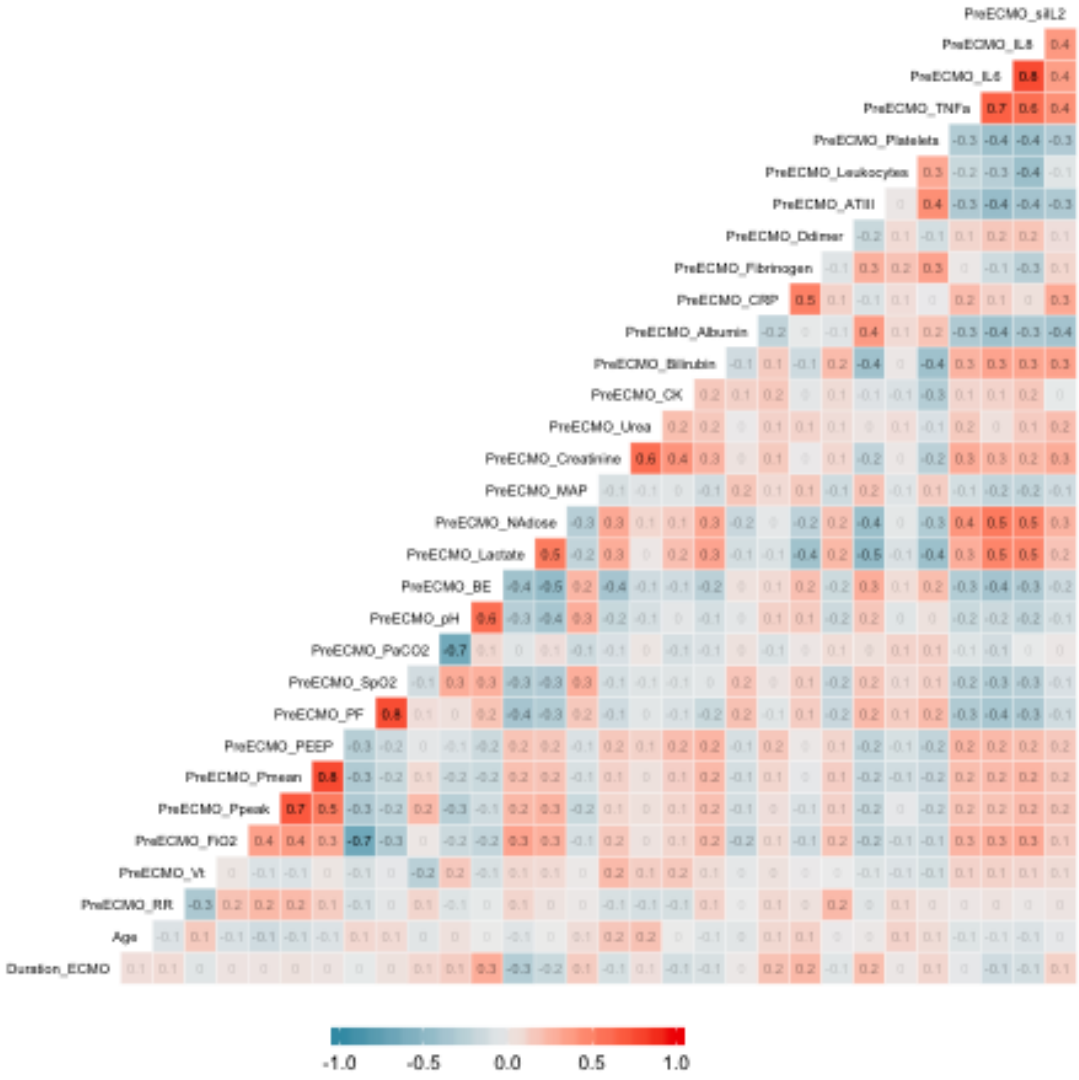

Figure 3: Heatmap of standardized and transformed variables.

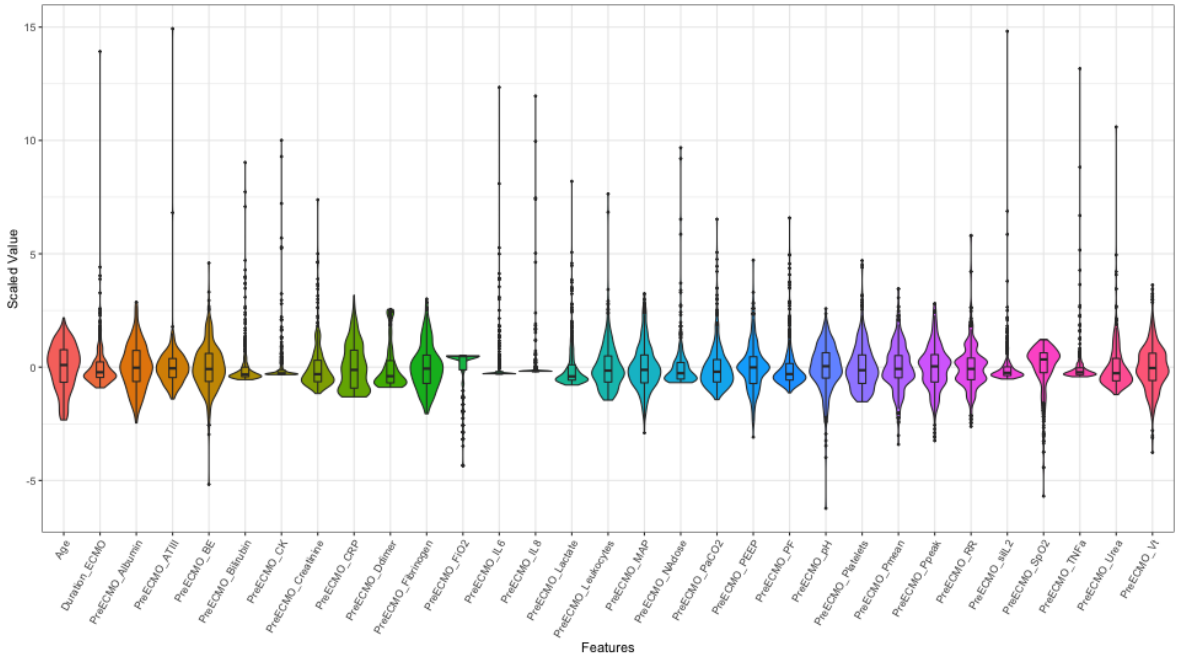

Figure 4: Violin plot of standardized variables.

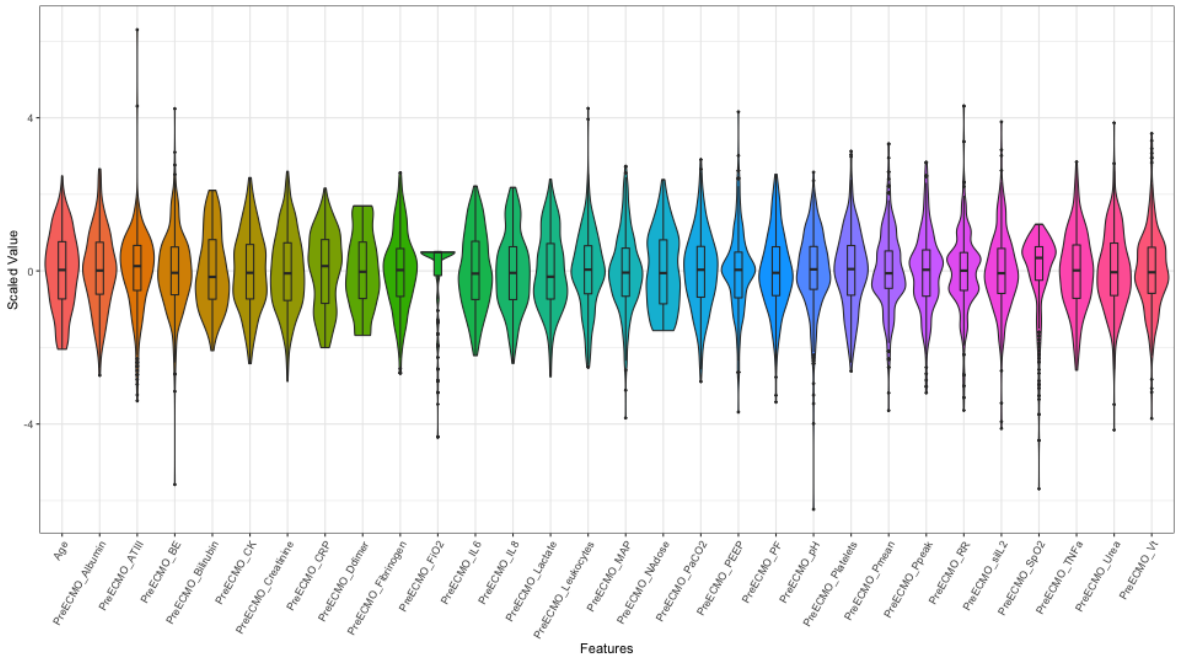

Figure 5: Violin plot of standardized and transformed variables.

8.2 R Code