CS7DS1-Data Analytics

# Semester project: Data Imputation methods

Roman Shaikh [ Student Number: 18300989 | Email: shaikhr@tcd.ie]

12-26-2018

# Table of Contents

Note: All of the data analysis done in this project was performed in R using R Studio as the development environment. All the code for analysis can be found at https://github.com/romaan7/DataAnalyticsFinalAssignment

# 1. Introduction

The goal of this project is to find a prediction model to fit the given dataset. The dataset provided has a response variable labeled *RESPONSE*, which could be assumed to be anything. Since we do not have any information about the provided dataset, we assume the response variable to be *"Is the tumor malignant OR benign?"*. Therefore, our goal becomes to predict the presence of a cancerous tumor in a patient. We do this by examining the incomplete dataset.

Missing data pose a problem in every data scientist's daily work. Should we impute them? If so, which method is appropriate? Or can observations with missing data points simply be dropped? To answer these questions, one would need to know what the mechanism behind the missing data is. Detecting it with statistical tests is complex and sometimes only leads to vague statements.

There might be multiple reasons why a dataset is incomplete. It is crucial to investigate what the possible causes of missing data could be, as this can impact the way we tackle the problem. For instance, if non-response in job-related surveys is the concern, one could expect the very richest and the very poorest respondents not to disclose their earnings in the questionnaire, which means the missing data points are not spread out evenly over the dataset. If this is the case, simply removing incomplete observations before performing the analysis of interest would yield biased results. On the other hand, for example, if data are missing due to a failure of the data collecting device, it might be that the locations of the missing points in the dataset are purely random.

There are four distinct patterns according to which the data can be missing. They are typically referred to as missing data mechanisms.

## Missing completely at random (MCAR)

Under MCAR there is no systematic pattern in the location of the missing data points in the dataset: they occur entirely at random. In this case, dropping incomplete observations would not introduce bias to the results of the subsequent analysis.

Example: Temperature is continuously measured by a sensor that collects the data and sends them via the Internet to a database. Due to unknown reasons, Internet connection breaks sometimes.

## Missing at random (MAR)

Under MAR, the probability of an observation is missing is still independent on its own values, but it does depend on the values of other variables. In this case, removing incomplete observations makes the sample less representative.

Example: Some of the night-hours data are missing due to the sensor's maintenance works, which are always carried out overnight.

### Missing not at random (MNAR)

Under MNAR, the probability of an observation being missing depends on its own unobserved values. In this case, again, dropping incomplete data leads to a biased analysis.

Example: Sensor freezes in -20 degrees Celsius and does not measure temperature below this value.

### Logically Missing

Missing data is logically not possible.

Example: Certain values may not be applicable to a candidate in a survey.

**For this project, we assume the MAR case and apply some of the advance data imputation techniques for our prediction task.**

# 2. Data Analysis

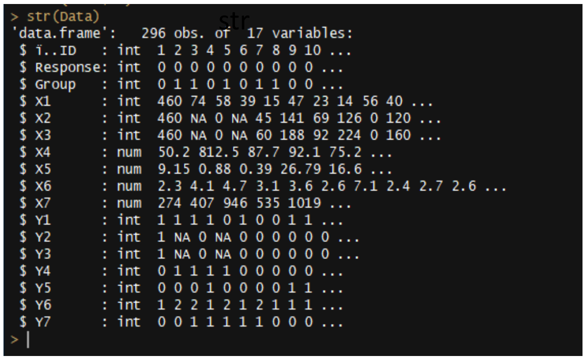
## 2.1. Data structure

The given dataset has 17 variables/rows with different datatypes. The continuous variables are labeled X and the corresponding categorical variables are labeled Y. Our response variable RESPONSE is taken to a categorical with BENIGN = 0 and MALIGNANT = 1. The categorical variable GROUP is assumed to be the gender of the patient Female = 0 and Male = 1.

| Variables | Data type | # missing values |
|---|---|---|
| ID | Index | 0 |
| Response (malignant OR benign) | Categorical | 0 |
| Group (Male OR Female) | Categorical | 0 |
| X1 | Continuous | 4 |
| X2 | Continuous | 130 |
| X3 | Continuous | 131 |
| X4 | Continuous | 0 |
| X5 | Continuous | 4 |
| X6 | Continuous | 63 |
| X7 | Continuous | 24 |
| Y1 | Categorical | 4 |
| Y2 | Categorical | 130 |
| Y3 | Categorical | 131 |
| Y4 | Categorical | 0 |
| Y5 | Categorical | 4 |
| Y6 | Categorical | 63 |
| Y7 | Categorical | 24 |

Table 1: Variables and data types.

Figure: 1 Structure of data using R



str

```
> str(Data)
'data.frame':   296 obs. of  17 variables:
 $ ï..ID   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Response: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Group   : int  0 1 1 0 1 0 1 1 0 0 ...
 $ X1      : int  460 74 58 39 15 47 23 14 56 40 ...
 $ X2      : int  460 NA 0 NA 45 141 69 126 0 120 ...
 $ X3      : int  460 NA 0 NA 60 188 92 224 0 160 ...
 $ X4      : num  50.2 812.5 87.7 92.1 75.2 ...
 $ X5      : num  9.15 0.88 0.39 26.79 16.6 ...
 $ X6      : num  2.3 4.1 4.7 3.1 3.6 2.6 7.1 2.4 2.7 2.6 ...
 $ X7      : num  274 407 946 535 1019 ...
 $ Y1      : int  1 1 1 1 0 1 0 0 1 1 ...
 $ Y2      : int  1 NA 0 NA 0 0 0 0 0 0 ...
 $ Y3      : int  1 NA 0 NA 0 0 0 0 0 0 ...
 $ Y4      : int  0 1 1 1 1 0 0 0 0 0 ...
 $ Y5      : int  0 0 0 1 0 0 0 0 1 1 ...
 $ Y6      : int  1 2 2 1 2 1 2 1 1 1 ...
 $ Y7      : int  0 0 1 1 1 1 1 0 0 0 ...
> |
```

## 2.2. Visualizing Missing data

Here we visualize the missing data and patterns using *naniar* and *VIM* libraries in R.

As mentioned earlier the data was assumed to be missing at random (MAR). We can see from the figure that the distribution of missingness across the dataset is uneven. With a total of around 14% of data missing in the whole dataset, we see X2, X3, and Y2, Y3 have significantly more missing values when compared to other variables. Also note that there are no missing values for variables Response, Group, X4, and Y4.

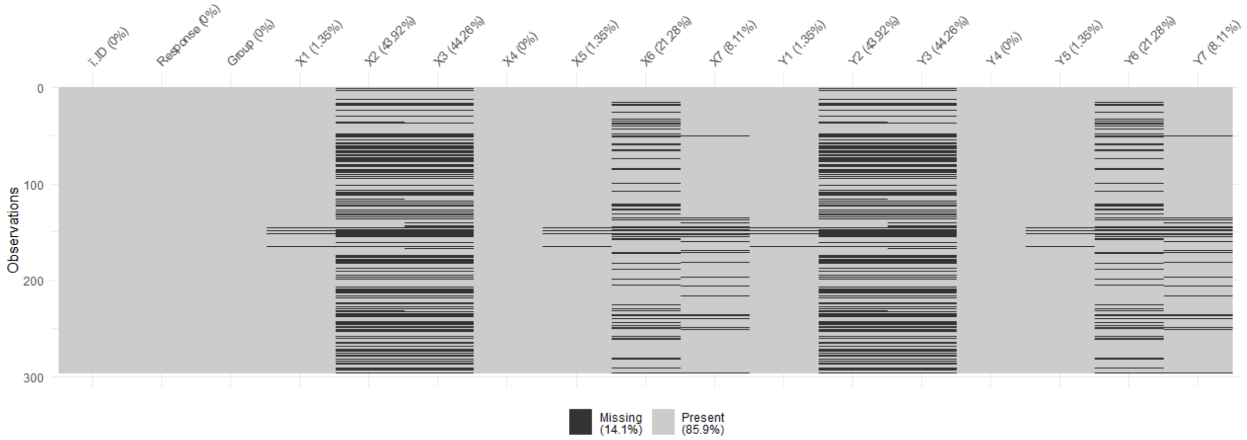


Figure 2: Visualizing missing data

Below figure shows us the pattern in missing data (RED CELLS = missing values and BLUE CELLS = accessible data). We see that around 44% of the entries are complete, and for X2, X3, Y2, Y3 around 30% are incomplete (most among other values).

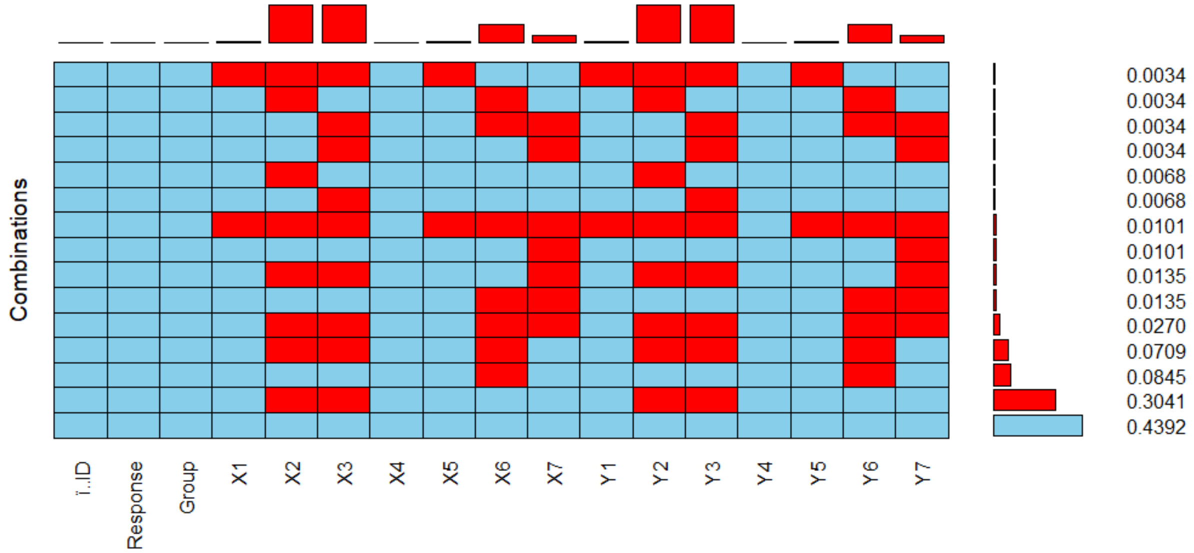


Figure 3: Missing data breakdown.

## 2.3. Analyzing Shape and Variance

For analyzing the variance in the dataset, we first plot the box plots of the whole dataset. As we see from the figure there is a huge number of outliers in data. Prediction algorithms are very sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results.

Outlier problem can be addressed in many ways like Multivariate method, Minkowski error Universal method etc. For our purpose, we will use the Top-Coding method which significantly reduces the effects of large values on the model.

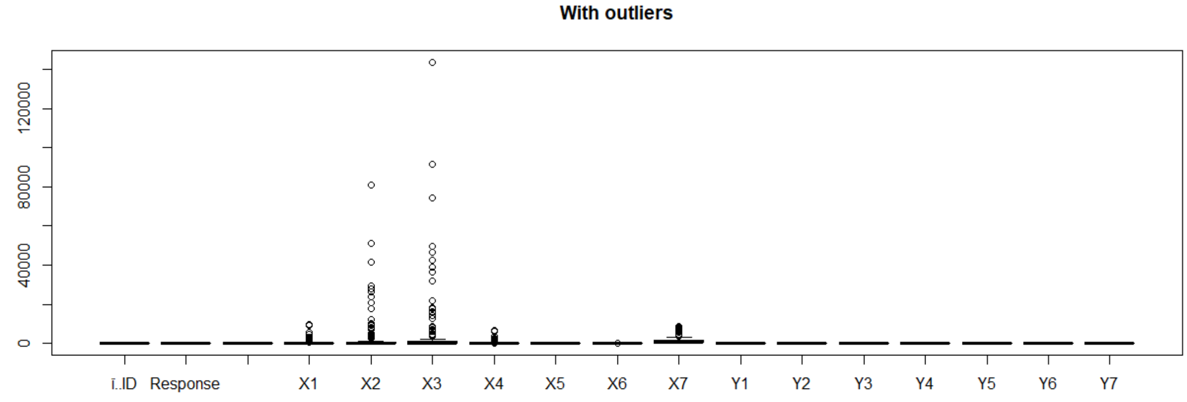


Figure 4: Boxplot with outliers

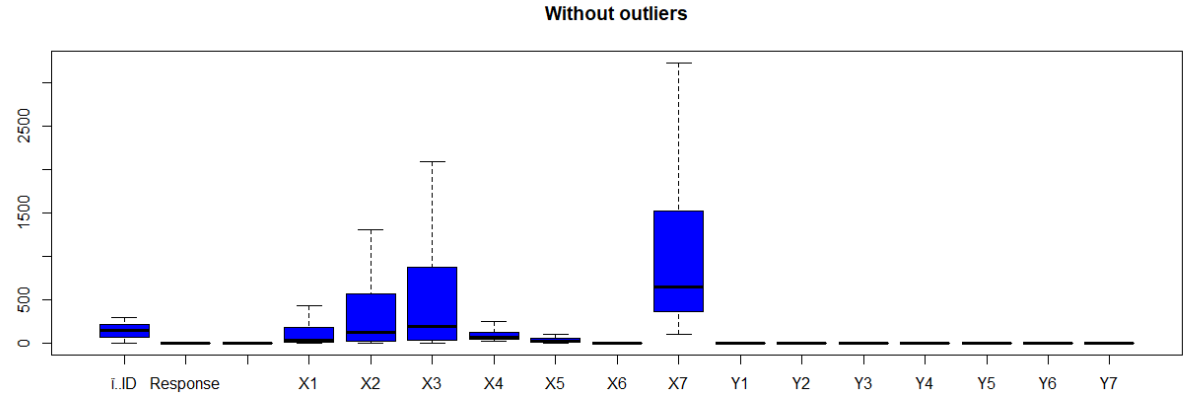
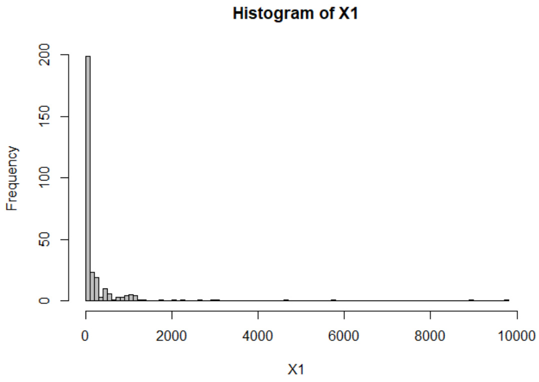


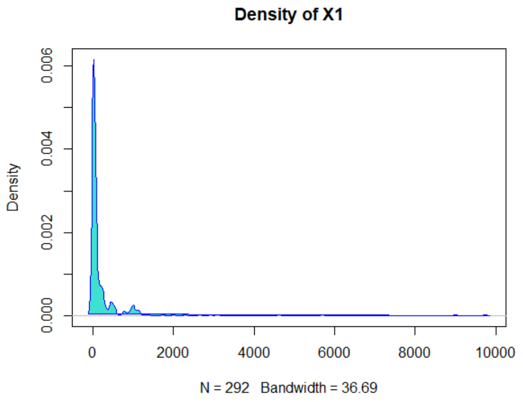
Figure 5: Boxplot without outliers

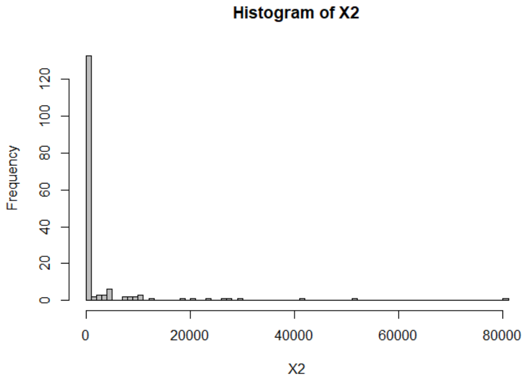
### Skewness of Data

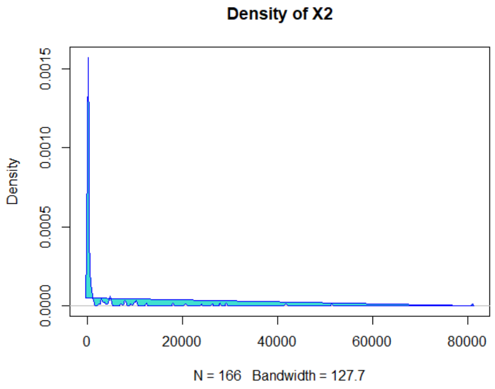
From plotting the histograms and density plots for each of the continuous variable, we see the skewness in data.

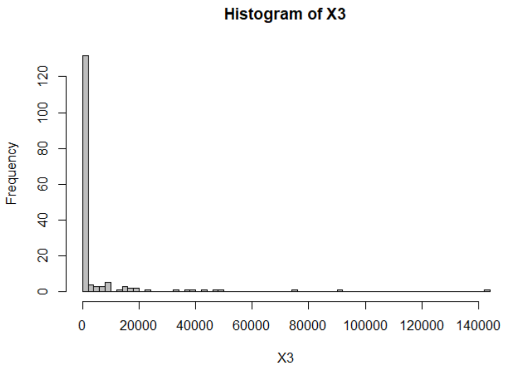
From below figures we clearly see that X1, X2, X3, X4, X7 are highly skewed variables. Moreover, the scales used to measure the variables are different. Example X1 ranges from 0 to 10,000 whereas X6 ranges from 0 to 10. For this, we propose standardization of the variables, which will help in getting the range of different variables on a similar scale for our predictive analysis. While imputing the data it is important to consider the skewness or the shape of data. One other point to note is that the group i.e. Gender does not have any impact on data as seen in below figures.

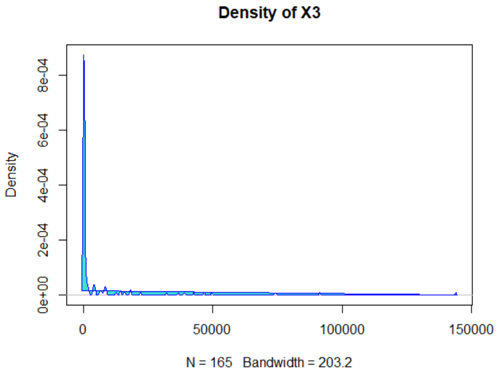

Histogram of X1


Density of X1


Histogram of X2


Density of X2


Histogram of X3


Density of X3

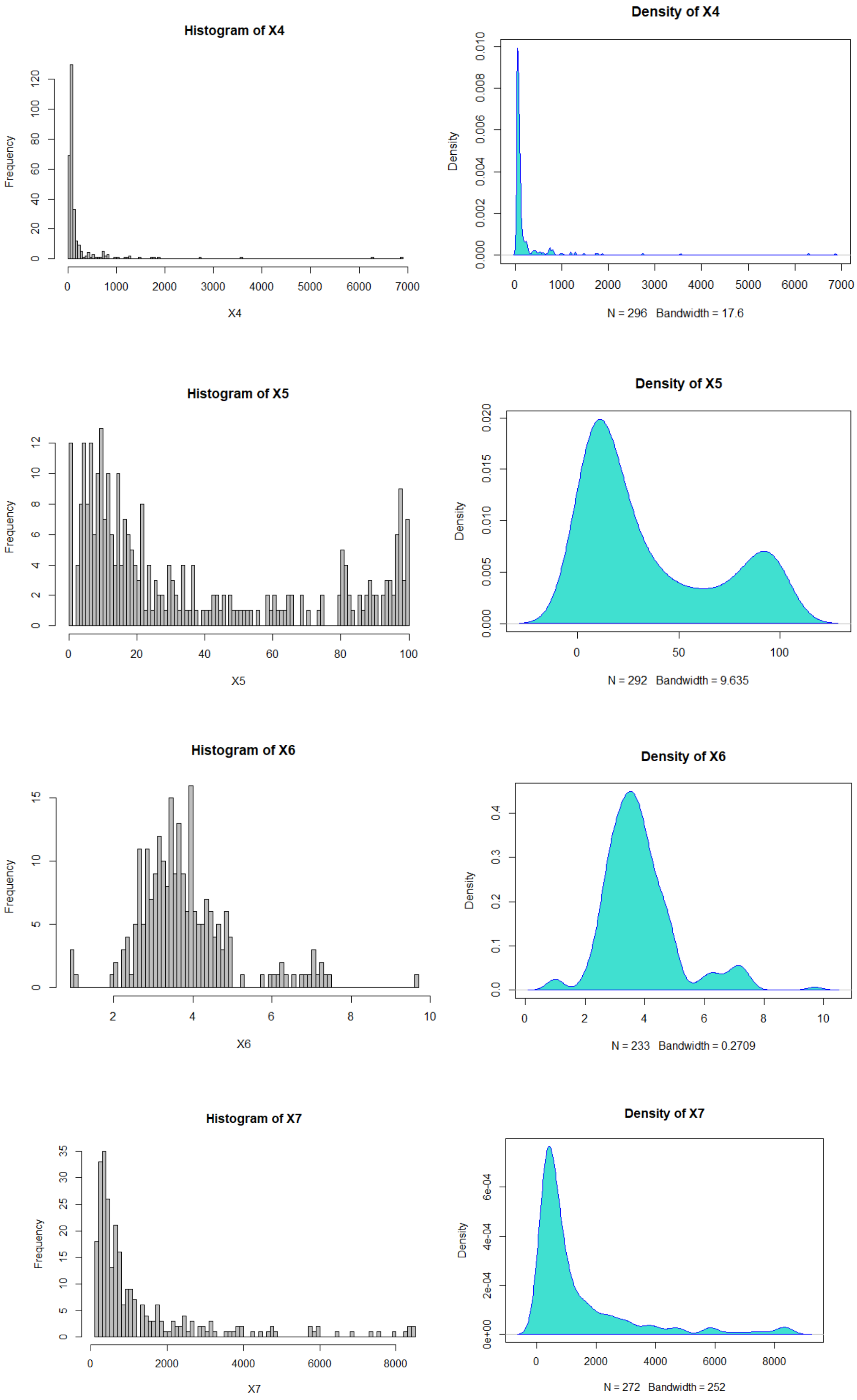
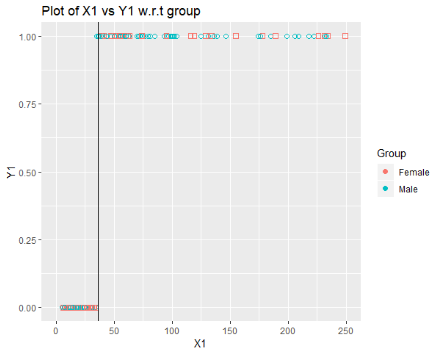
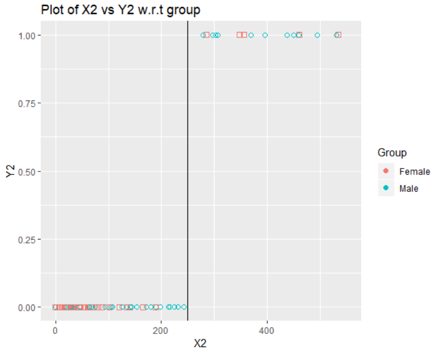
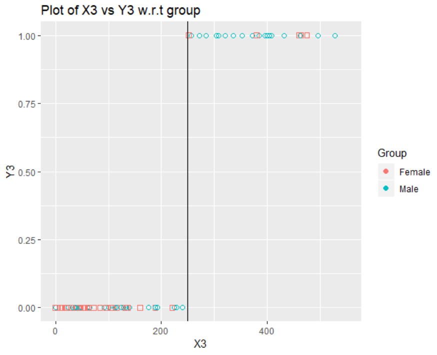


Figure 6: Histogram and Density plot of X variables.

## 2.4. X and Y Relationship

As we have the information (given with the dataset in question.) that there can be a relation between the X and corresponding Y variables i.e. X1-Y1, X2-Y2 etc. We examine the relationship between X and Y variables by plotting the scatterplots considering each group i.e. Male and Female. we can observe from below table that there some relationship between X and Y variables with Y jumping to value 1 beyond an approximate cutoff point. E.g. – For X1-Y1 pair after a cutoff of 36 for X1 values, the Y1 value jumps from 0 to 1 (Note that there is no clear cutoff point observed for X5-Y5 pair). Also, we see there is no effect of group variable on the relationship.

From this we can conclude that X and Y variables are related and information from Y is also held with X. Hence for our predictive analysis task, it is possible to ignore the y variables except for the Y5.

| Plots | R code | Cutoff points |
|---|---|---|
|  | `ggplot(Data, aes(x=X1, y=Y1,shape=Group, color=Group)) + ggtitle("Plot of X1 vs Y1 w.r.t group") +geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X1", limits=c(0, 250))+geom_vline(xintercept = 36)` | ~36 |
|  | `ggplot(Data, aes(x=X2, y=Y2,shape=Group, color=Group)) + ggtitle("Plot of X2 vs Y2 w.r.t group") + geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X2", limits=c(0, 550))+geom_vline(xintercept = 250)` | ~250 |
|  | `ggplot(Data, aes(x=X3, y=Y3,shape=Group, color=Group)) + ggtitle("Plot of X3 vs Y3 w.r.t group") + geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X3", limits=c(0, 550))+geom_vline(xintercept = 250)` | ~250 |

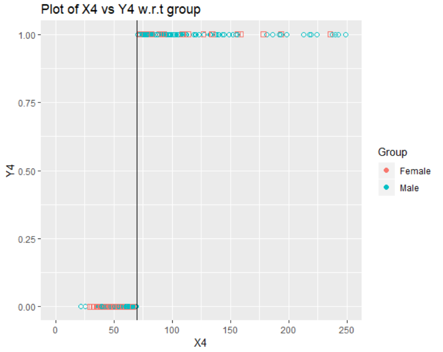
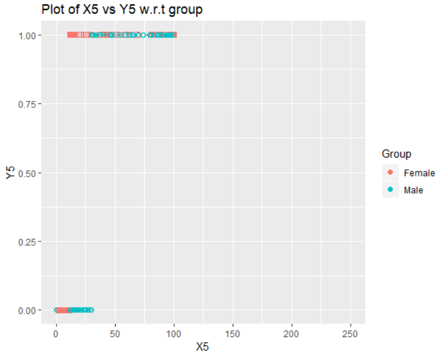
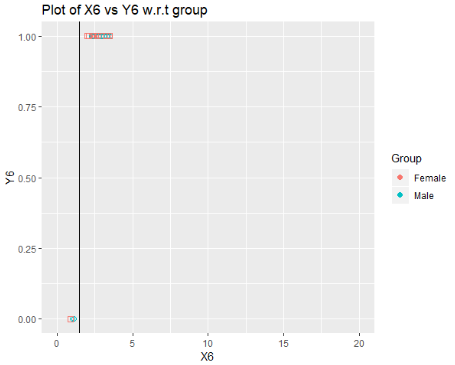
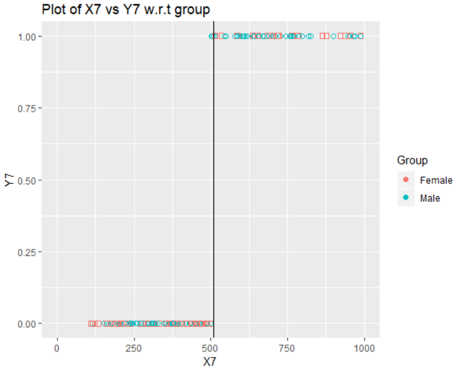
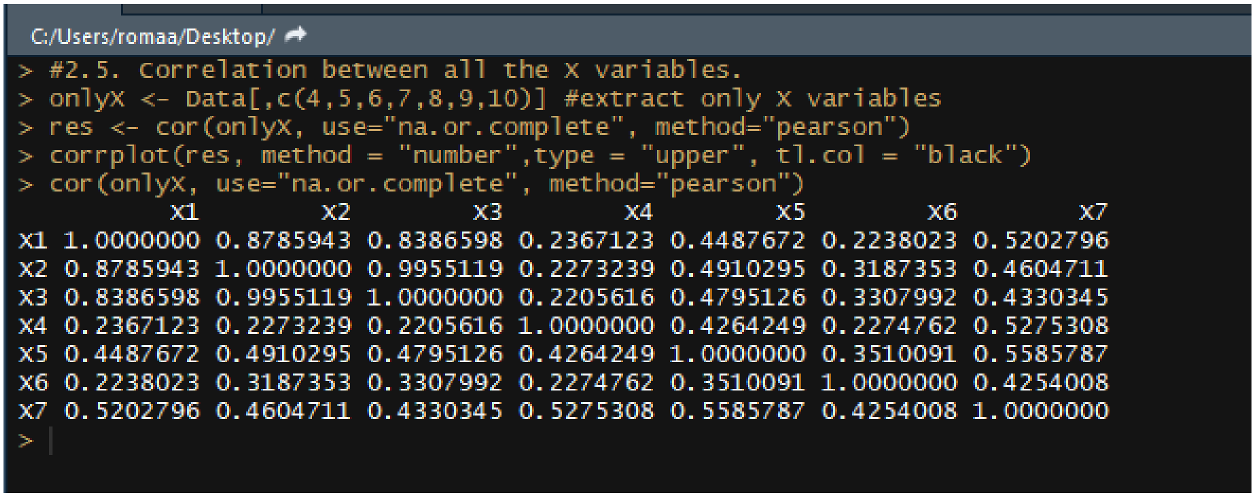
| | | |
|---|---|---|
|  | `ggplot(Data, aes(x=X4, y=Y4,shape=Group, color=Group)) + ggtitle("Plot of X4 vs Y4 w.r.t group") + geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X4", limits=c(0, 250))+geom_vline(xintercept = 70)` | ~70 |
|  | `ggplot(Data, aes(x=X5, y=Y5,shape=Group, color=Group)) + ggtitle("Plot of X5 vs Y5 w.r.t group") + geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X5", limits=c(0, 250))#+geom_vline(xintercept = 30)` | Not Found |
|  | `ggplot(Data, aes(x=X6, y=Y6,shape=Group, color=Group)) + ggtitle("Plot of X6 vs Y6 w.r.t group") + geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X6", limits=c(0, 20))+geom_vline(xintercept = 1.5)` | ~1.5 |
|  | `ggplot(Data, aes(x=X7, y=Y7,shape=Group, color=Group)) + ggtitle("Plot of X7 vs Y7 w.r.t group") + geom_point(size=2, shape=Group) + scale_y_continuous(limits=c(0, 1)) +scale_x_continuous(name="X7", limits=c(0, 1000))+geom_vline(xintercept = 510)` | ~510 |

Table 2: X-Y relationship and cutoff points.

## 2.5. Correlation between all the X variables.

Further, we check the correlation between all the X variables. This will help us to see if there is any strong correlation between any of these variables and if we can eliminate these variables for our predictive analysis task. Which will ultimately reduce the complexity of our model.

We first extract only X variables and generate the correlation matrix for it. When we plot this correlation matrix we see that there is a strong relationship between X1, X2 and X3 variables with correlation values above an approximate of 0.8. Which means we can use only one of them to predict the values. This will highly reduce our model's complexity. For this task, we will use X1.



```
> #2.5. Correlation between all the X variables.
> onlyX <- Data[,c(4,5,6,7,8,9,10)] #extract only X variables
> res <- cor(onlyX, use="na.or.complete", method="pearson")
> corrplot(res, method = "number",type = "upper", tl.col = "black")
> cor(onlyX, use="na.or.complete", method="pearson")
          X1        X2        X3        X4        X5        X6        X7
X1 1.0000000 0.8785943 0.8386598 0.2367123 0.4487672 0.2238023 0.5202796
X2 0.8785943 1.0000000 0.9955119 0.2273239 0.4910295 0.3187353 0.4604711
X3 0.8386598 0.9955119 1.0000000 0.2205616 0.4795126 0.3307992 0.4330345
X4 0.2367123 0.2273239 0.2205616 1.0000000 0.4264249 0.2274762 0.5275308
X5 0.4487672 0.4910295 0.4795126 0.4264249 1.0000000 0.3510091 0.5585787
X6 0.2238023 0.3187353 0.3307992 0.2274762 0.3510091 1.0000000 0.4254008
X7 0.5202796 0.4604711 0.4330345 0.5275308 0.5585787 0.4254008 1.0000000
> 
```

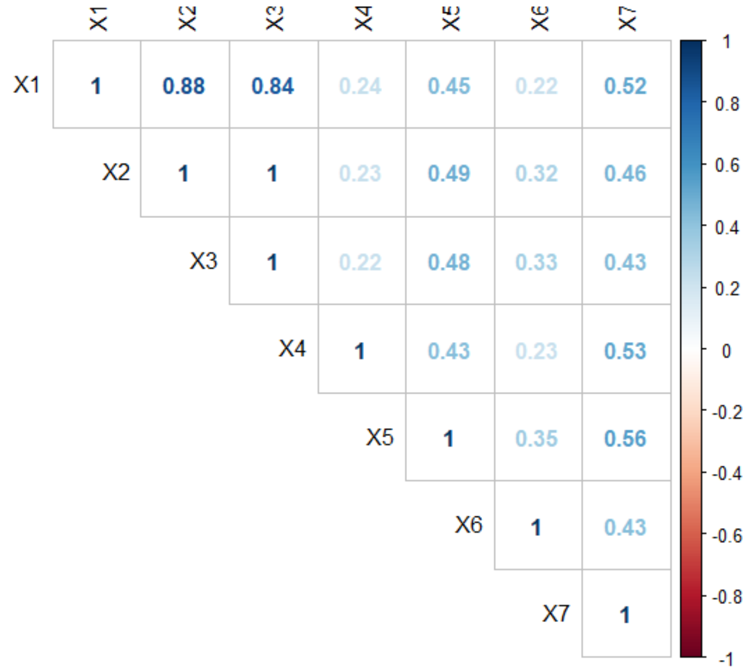


Figure 7: Correlation matrix and plot

## 2.6. Response vs X

Now that we have our X variables analyzed, let's examine to see if we can find any relationship between our RESPONSE variable and X variables.

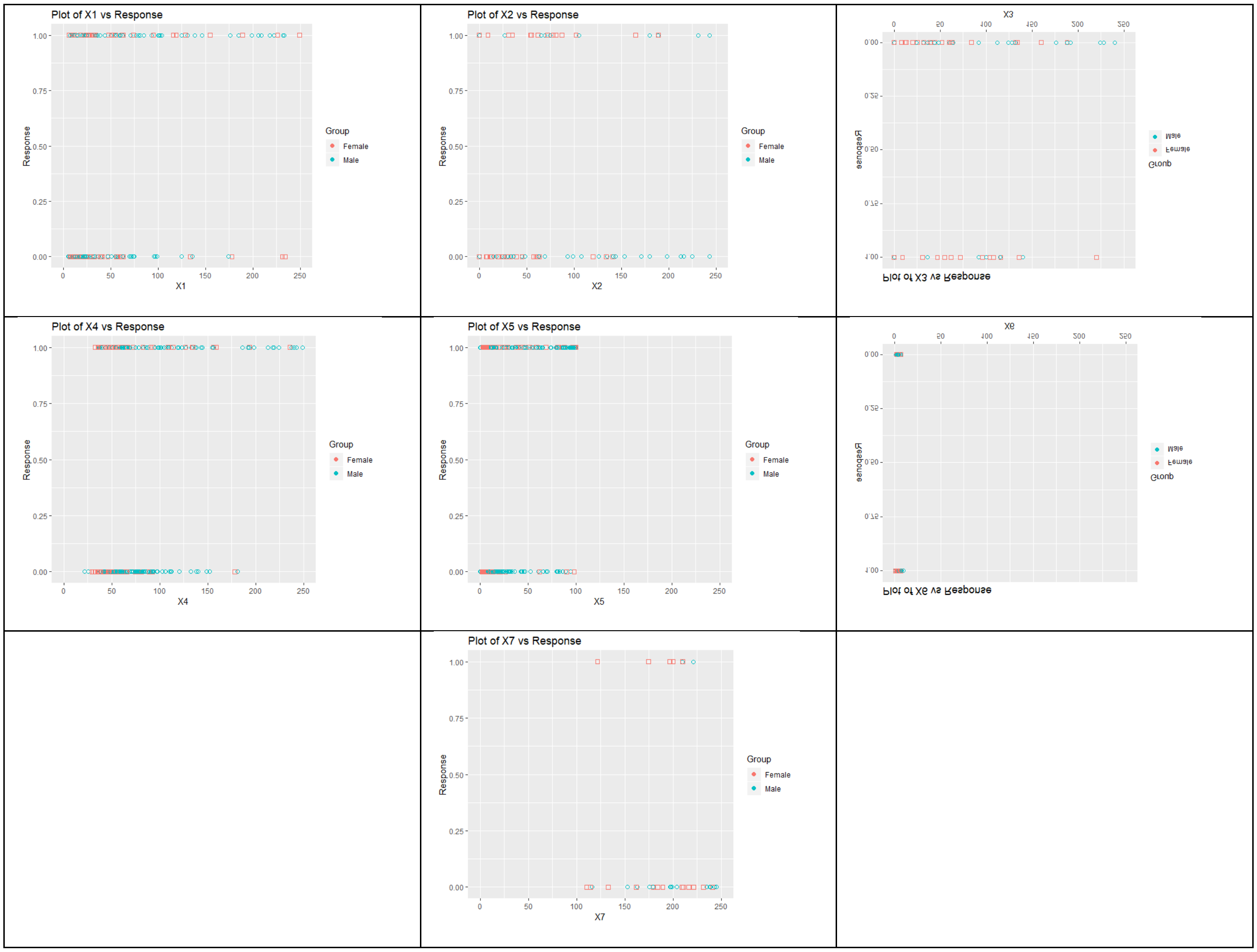


Figure 8: Response vs X variable plots

As we can see from the above graphs there is no clear correlation between the Response and X Continuous variables. This leads us to our objective of analyzing the relationship using predictive algorithms and models and predicting whether the tumor is benign or malignant.

# 3. Data Imputation, Cleaning, and Baselining.

Keeping our goal in mind we now compare various imputation methods used to replace missing data with different complexity levels. As discussed earlier we consider the MAR case for our missing data and will focus on three types of data imputation methods – 1. Median Imputation, 2. KNN (K-nearest neighbor) and 3. Predictive mean matching (provided my MICE package in R).

We will compare the performance of these methods using three predictive models – 1. Decision Tree, 2. Logistic Regression and 3. Random Forest. Also, a baseline will be established for comparison.

The performance of all these models will be based on four parameters precision, error, recall and accuracy.

## 3.1. Median Imputation

This is like mean imputation in which we calculate the mean of the observed values for that variable. But since there is a high degree of skewness in our data we to a median imputation. In this, we replace the missing values with the Median of the observed values. This method may introduce bias and error to the dataset.

## 3.2. KNN (K-Nearest Neighbor)

The assumption behind using KNN for missing values is that a point value can be approximated by the values of the points that are closest to it, based on other variables. Where K is the number of closest points to look for and is usually given by user based on the model. KNN is the most commonly used method for all type of data.

## 3.3. Predictive Mean Matching (PMM)

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 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 (Taken from Heitjan and Little 1991; Schenker and Taylor 1996).

The MICE package in R provides an implementation on PMM, which makes it easy to use (We use this here). Data imputed using PMM will generally follow the distribution pattern of the data without introducing much bias.

## 3.4. Baseline

Before we proceed for imputation, we need to have a baseline for comparison. This will help is understand the improvements in the results using different methods. We use Decision tree as our initial baseline model.

When we apply different optimization techniques like Top-Coding, Standardization and feature selection we can see the improvements in result from below the table.

| Method (Decision Tree) | Accuracy | Precision | Error | Recall |
|---|---|---|---|---|
| **Baseline** | 0.68 | 0.78 | 0.32 | 0.6 |
| **Top-coding** | 0.73 | 0.70 | 0.27 | 0.75 |
| **Top-coding + Standardization** | 0.75 | 0.69 | 0.25 | 0.8 |
| **Top-coding + Standardization + Feature Selection** | 0.76 | 0.69 | 0.24 | 0.83 |

Table 3: Baseline model performance metrics.

Top-coding - data observation is one for which data points whose values are above an upper bound are censored.

We observe that after top coding the outliers in X1, X2, X3, X4, X7 the accuracy increases to 0.73 from 0.68 of our baseline model. We perform standardization and feature selection to make sure that all the variables are on the same scale and highly correlated variables are eliminated like X2 and X3 the accuracy jumps to 0.76 and error is reduced to 0.24.

## 4. Comparing Data Imputation methods

We use three models to compare the above-discussed data imputation methods Decision tree, Random forest, and Logistic Regression.

| | **Imputation Method** | | **Precision** | **Accuracy** | **Error** | **Recall** |
|---|---|---|---|---|---|---|
| Decision Tree | | | | | | |
| | **Median Imputation** | | 0.7 | 0.73 | 0.27 | 0.76 |
| | **PMM Imputation** | | 0.64 | 0.73 | 0.27 | 0.81 |
| | **KNN imputation** | k=10 | 0.73 | 0.76 | 0.24 | 0.8 |
| | | k=3 | 0.71 | 0.78 | 0.22 | 0.84 |
| | | | | | | |
| Random Forest | **Median Imputation** | | 0.7 | 0.76 | 0.24 | 0.81 |
| | **PMM Imputation** | | 0.68 | 0.75 | 0.25 | 0.82 |
| | **KNN imputation** | k=10 | 0.73 | 0.8 | 0.2 | 0.86 |
| | | k=3 | 0.79 | 0.84 | 0.16 | 0.88 |
| | | | | | | |
| Logistic Regression | **Median Imputation** | | 0.56 | 0.72 | 0.28 | 0.86 |
| | **PMM Imputation** | | 0.64 | 0.72 | 0.28 | 0.79 |
| | **KNN imputation** | k=10 | 0.68 | 0.72 | 0.28 | 0.78 |
| | | k=3 | 0.66 | 0.74 | 0.26 | 0.81 |

Table 4 : Different model performance metrics comparison.

As we see from the above comparison table, for all three models i.e. Decision tree, Random Forest and Logistic regression thee KNN imputation performed the best giving highest accuracy of 0.84 with Random forest. We can also note that PMM in Decision tree had a high recall with better precision rate. Random Forest is the clear winner with the highest accuracy of 0.84 when compared to Logistic regression and Decision tree. KNN with k=3 we note the highest performance across all the models.

By comparing our results to the baseline, we can clearly see that imputing missing data can increase the overall performance of the prediction. It is also worthy to note that the group variable had no effect on the prediction of the response.

## 5. Analyzing each variable

For analyzing the performance of each individual variable together with the group we build a decision tree model using each variable separately first then along with the group. The performance was then measured using K-fold cross-validation with 10% in each fold as a test and remain as training. By measuring the performance before imputing the variables and after imputing we can see the difference in numbers. With a 0.04 increase in accuracy and 0.04 decrease in error, recall didn't change at all and precision increased by 0.10.

| | Before Imputation | | | | After Imputation | | | |
|---|---|---|---|---|---|---|---|---|
| **Variables** | **Precision** | **Accuracy** | **Error** | **Recall** | **Precision** | **Accuracy** | **Error** | **Recall** |
| **X1** | 0.66 | 0.68 | 0.32 | 0.7 | 0.63 | 0.68 | 0.32 | 0.74 |
| **X2** | 0.39 | 0.56 | 0.44 | 0.71 | 0.8 | 0.79 | 0.21 | 0.79 |
| **X3** | 0.31 | 0.57 | 0.43 | 0.81 | 0.72 | 0.72 | 0.28 | 0.73 |
| **X4** | 0.58 | 0.74 | 0.26 | 0.87 | 0.6 | 0.75 | 0.25 | 0.88 |
| **X5** | 0.61 | 0.72 | 0.28 | 0.83 | 0.61 | 0.73 | 0.7 | 0.85 |
| **X6** | 0.4 | 0.59 | 0.41 | 0.77 | 0.51 | 0.62 | 0.38 | 0.75 |
| **X7** | 0.51 | 0.63 | 0.37 | 0.73 | 0.6 | 0.68 | 0.32 | 0.74 |
| **Y1** | 0.72 | 0.71 | 0.29 | 0.68 | 0.72 | 0.71 | 0.29 | 0.67 |
| **Y2** | 0.3 | 0.62 | 0.38 | 0.89 | 0.49 | 0.69 | 0.31 | 0.88 |
| **Y3** | 0.25 | 0.6 | 0.4 | 0.83 | 0.64 | 0.69 | 0.31 | 0.75 |
| **Y4** | 0.67 | 0.67 | 0.33 | 0.64 | 0.69 | 0.67 | 0.33 | 0.66 |
| **Y5** | 0.68 | 0.72 | 0.28 | 0.74 | 0.7 | 0.72 | 0.28 | 0.76 |
| **Y6** | 0.72 | 0.57 | 0.43 | 0.44 | 0.69 | 0.62 | 0.38 | 0.56 |
| **Y7** | 0.78 | 0.64 | 0.36 | 0.52 | 0.77 | 0.66 | 0.34 | 0.55 |
| **Group + X1** | 0.63 | 0.72 | 0.28 | 0.8 | 0.62 | 0.67 | 0.33 | 0.71 |
| **Group + X2** | 0.35 | 0.59 | 0.41 | 0.82 | 0.79 | 0.77 | 0.23 | 0.77 |
| **Group + X3** | 0.31 | 0.57 | 0.43 | 0.8 | 0.77 | 0.72 | 0.28 | 0.67 |
| **Group + X4** | 0.59 | 0.75 | 0.25 | 0.89 | 0.6 | 0.74 | 0.26 | 0.87 |
| **Group + X5** | 0.62 | 0.72 | 0.28 | 0.8 | 0.63 | 0.72 | 0.28 | 0.8 |
| **Group + X6** | 0.39 | 0.6 | 0.4 | 0.8 | 0.52 | 0.63 | 0.37 | 0.74 |
| **Group + X7** | 0.6 | 0.63 | 0.37 | 0.68 | 0.65 | 0.66 | 0.34 | 0.67 |
| **Group + Y1** | 0.72 | 0.71 | 0.29 | 0.68 | 0.75 | 0.71 | 0.29 | 0.69 |
| **Group + Y2** | 0.31 | 0.62 | 0.38 | 0.9 | 0.49 | 0.69 | 0.31 | 0.89 |
| **Group + Y3** | 0.68 | 0.61 | 0.39 | 0.55 | 0.62 | 0.69 | 0.31 | 0.75 |
| **Group + Y4** | 0.69 | 0.67 | 0.33 | 0.64 | 0.69 | 0.67 | 0.33 | 0.65 |
| **Group + Y5** | 0.69 | 0.72 | 0.28 | 0.75 | 0.68 | 0.72 | 0.28 | 0.75 |
| **Group + Y6** | 0.67 | 0.6 | 0.4 | 0.54 | 0.67 | 0.62 | 0.38 | 0.56 |
| **Group + Y7** | 0.75 | 0.65 | 0.35 | 0.56 | 0.76 | 0.66 | 0.34 | 0.56 |

Table 5: Performance metrics for individual variable and group

## 6. Summary

We conclude by noting the following observations from this experiment/project.

- The assumption of MAR for this dataset. If more details are provided for the dataset, it might be easier to find correlations between variables and decide upon a better imputation method.
- Data imputation generally lead to a better performance metric when compared to the models with discarded values.
- Data preprocessing with methods like feature selection and top-coding helped in improving the performance metrics as well.
- As expected Random forest continually outperformed the other models. This is because of its Ensemble nature which reduces uncertainty.
- However, the least performer is not consistent as we observe in case of the Decision tree (KNN and Median imputation) and Logistic regression (PMM/MICE)
- The best performing imputation method, however, was PMM/MICE which was followed by KNN imputation. This may however not be generalized as the results may depend heavily on the shape and size of data.