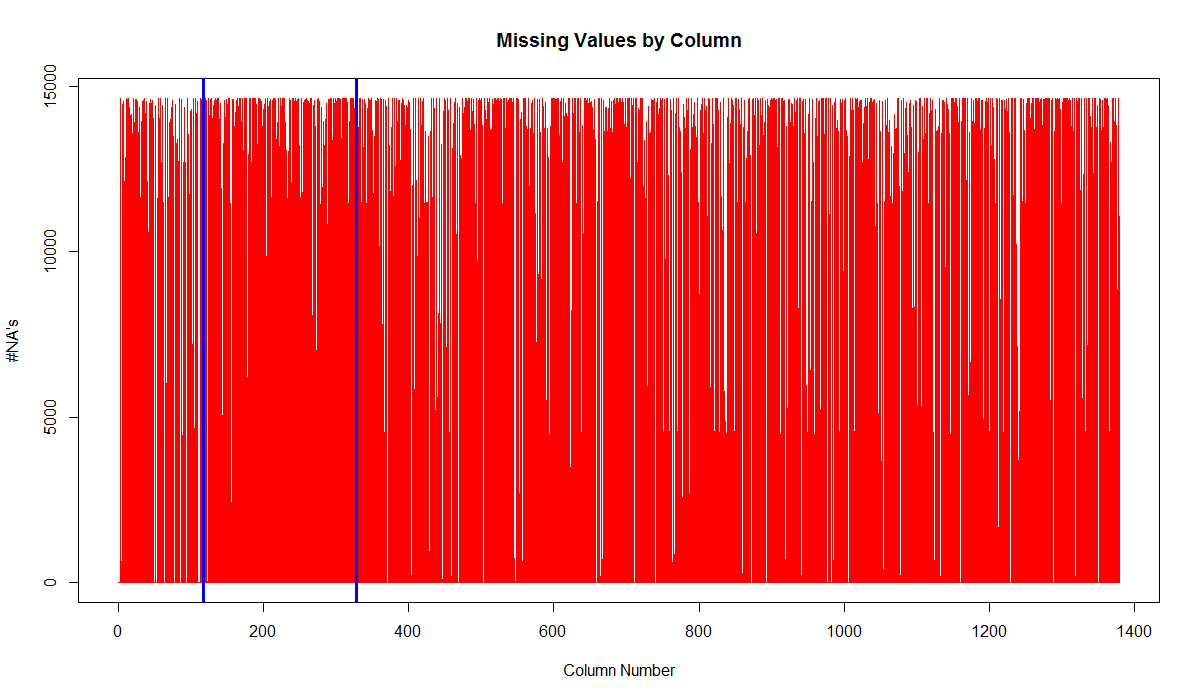
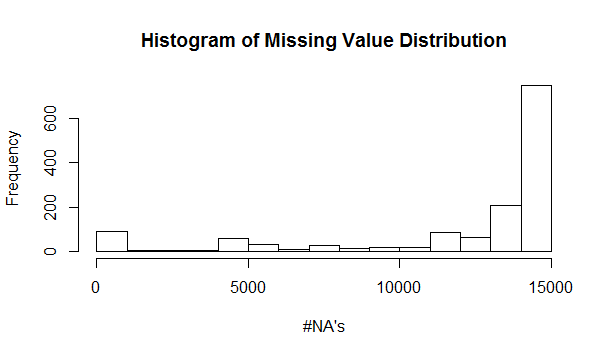
Missing Value Imputation

The dataset was filled with missing values. More specifically, there were more missing values than valid values within the inindependent variables. To explore into quantities and distributions of missing values among the columns, visualizations were made using line charts and histograms.



*Figure 1. Visualization of Missing Values by Columns. Blue vertical lines were used to separate among numeric (first 116 columns), ordinal (next 210 columns) and categorical (last 1052 columns) variables.*

**

*Figure 2. Histogram of Missing Value Distribution by Columns (Independent Variables).*

Figure 1 and Figure 2 showed that most columns were filled with missing values. In fact, 82.6% of the cells in train data were registered as "NA". Since columns with mostly missing values would do no good to the prediction but make the data redundant and slow to run, throwing away columns was considered before fitting the model. From Figure 2, most columns had 10,000+ missing values out of 14,662 observations and could thus be thrown away.

If the threshold was set to be 50%, i.e. columns with proportion of NA's exceeding 50% would be considered invalid, then 1159 out of 1378 independent variables would be discarded. To compare, an 80% cutoff threw away 1038 out of 1378 columns. In practice we experimented with both cutoffs for each model and compared the accuracy resulted from loss of data.

After removing invalid columns, NA values left were handled using different approaches for different datatypes (categorical, numeric or ordinal). Since more than one datatype was presented in the independent variables, plus that the dataset was huge and obfuscated, it was hard to use predictions from regression to fill in the NA values. Instead, generic methods were used for each datatype:

1. For numeric columns, NA's were converted to 0's;

2. For ordinal columns, NA's were converted to -1;

3. For categorical columns, new category called "missing" was introduced.

Data Processing (Dummy)

In order to fit certain algorithms (e.g. Ridge, SVM, logit), columns with data type "categorical" were required to be made into dummy variables. Function model.matrix in R was used to introduce dummy columns with categorical columns treated as factors.

Levels of each categorical variable were compared for train data and test data. Additional levels in test data that did not appear in train data were removed to make sure the models were trained properly. For levels that appeared in train data but not in test data, placeholder columns (columns with all 0's) were introduced to test data to take up the position of that level. This was necessary because it was required that train data and test data have the same number of columns, even after introducing dummy variables.

Ridge/Lasso

Shrinkage methods like Ridge was applied for each Y with lambda value being picked by ten-fold cross validation. The fourteen predictions were then combined and submitted to get a score. Figure 3 visualized the lambda values for each Y that minimized cross validation mean squared error.

lambdas.min

[1] 0.19818207 0.36686417 0.42816291 0.02344582 0.22931004 0.08761746 0.52432556

[8] 0.23928899 0.78558958 0.61596125 0.64794003 0.10522149 0.10637513 0.51174359

Discussion (Future Development)

One significant feature about this dataset was that it had not one, but fourteen dependent variables (Y's). And the dependent variables were correlated with each other since a woman usually orders more than one health services. Conventional regression models assumed the dimension of Y to be nx1, so did the algorithms in R. Therefore, 14 columns of data were fitted separately using the models, assuming they were independent, which was not actually true. Ignoring the correlation between Y variables may cause loss of accuracy. Application of multivariate methods might be able to increase prediction accuracy.

Methodology of imputing missing values might also result in loss of accuracy. Throwing away columns was a straightforward, but not elegant way to reduce dimension and save computer time. Ideally only columns with all NA's and columns with a single constant should be removed. Moreover, replacing missing values with a global value (e.g. 0 or -1) was a compromise due to the large amount of NA's and obfuscation of dataset. Provided more valid data and smaller size, regressing the missing values on other columns should be a better solution.

Procedure of creating dummy variables could be another source of problem. First of all, some levels in test data had to be dropped (these levels were replaced by "missing"). Secondly, introducing dummy variables dramatically increased dimension of data, without bringing in extra information. Lastly, the sole purpose of introducing placeholder columns (columns with only 0's) in test data was to make the number of columns match for test and train. Given that the model permitted, using factors instead of dummy variables should be more concise.

When regression models yielded prediction for probabilities, some of them might fall out of the range of 0 to 1. Converting these over-bound values to 0 or 1 would dramatically increase the score since log-loss was used for evaluation of prediction. To avoid this problem, values over-bound were replaced with column means. However, a finer way should be adopted to accommodate those values.