Bank Loan Approval Optimization Report

Mark Riley

April 23, 2019

Executive Summary

This report provides an analysis of our ability to predict which applicants are likely to default on their loans. Our selected method is logistic regression, which calculates the odds that a loan applicant will default. Each loan applicant is rated with odds between 0 (bad/default) and 1 (good/fully paid) using twenty five prediction variables such as loan amount, loan length, loan grade, income, and debt ratio. We then evaluated a number of odds threshold values to produce both the most accurate predicted results and to maximize the bank's profits, using a set of test data with known outcomes. A loan with a rating below the odds threshold classified the loan as a 'bad' loan, and a loan with a rating above or equal to the odds threshold classified the loan as 'good'.

Based on the data provided for this analysis, the bank's current level of accuracy is 78.1% of approved loans are fully paid, with a profit margin of 1.31%. When we optimized the accuracy of our model to predict loan status, we predicted that 80.7% of approved loans would be fully paid with a profit margin of 3.41%. When we optimized the model to produce the highest profit for the bank, we predicted that 84.1% of approved loans would be fully paid with a profit margin of 5.29%, a margin four times higher than the current loan approval methods.

Our recommendation is to deploy the model as designed in this project into production, with the threshold value optimal to maximizing the bank's profit.

The model's overall accuracy level for maximum profit does leave opportunities for further improvement. The following steps could be pursued to see if they improve the model's ability to further increase the bank's profit margins.

- Use additional data points about loan applicants beyond what what used in this analysis.
- This analysis only used data from loans that were approved. Conduct further analysis including details for loan applications that were denied.

Introduction

The dataset includes 32 variables for 50,000 randomly selected loans. This project will use logistic regression to predict which applicants are likely to default on their loans. Our process for the analysis will include:

1. Prepare the response variable based on the values of the status variable.
2. Remove observations from the dataset for loans that are late, current or in a grace period.
3. Eliminate variables that are not useful to the analysis or are redundant with other variables.
4. Consolidate categorical variables into meaningful groups.
5. Analyze and deal with missing values in the dataset.
6. Plot quantitative variables and transform heavily skewed results.
7. Explore the relationships between predictors and loan status to look for significant predictors.
8. Randomly divide the cleaned and transformed dataset into training (80% of observations) and testing (20% of observations).
9. Create a logistic model from the training data using all predictor variables.
10. Use the logistic model to predict the loan status for the test dataset with a threshold of 0.5.
11. Optimize the threshold value for predictive accuracy.
12. Optimize the threshold value for greatest net profit.

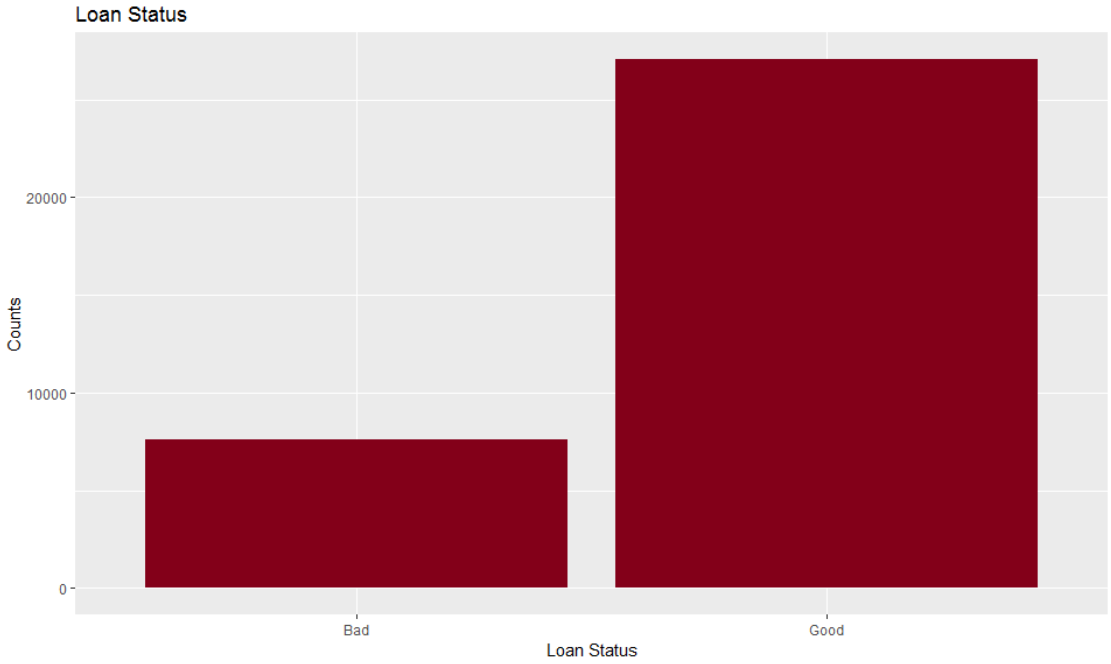
Preparing and Cleaning the Data

Response Variable

We will create a new response variable based on the existing status variable, loanStatus. The response variable will be a factor with two levels:

- Good -> loans with status of 'Fully Paid'
- Bad - > loans with a status of 'Charged Off'

Additionally we will remove the loans with status of 'Late,' 'Current', or 'Grace Period' from the dataset.



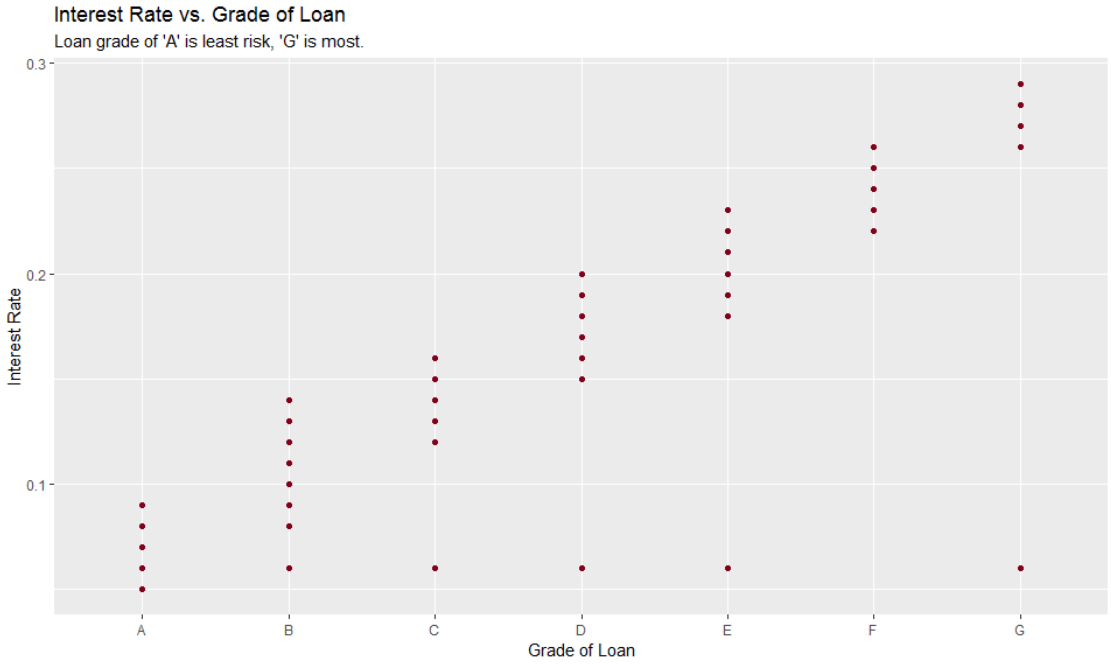

```
## [1] "Count of 'Good' loans: 27074 and 'Bad' loans: 7579"

## [1] "Proportion of 'Good' loans: 78.1 and 'Bad' loans 21.9"
```

We can see the Good to Bad loan ratio is roughly 78/22 after creating the response variable and filtering out the unwanted observations.

Eliminating Variables

Some variables will not be useful as predictors in our model. We reviewed the list of predictor variables in the dataset and determined the following variables could be removed:

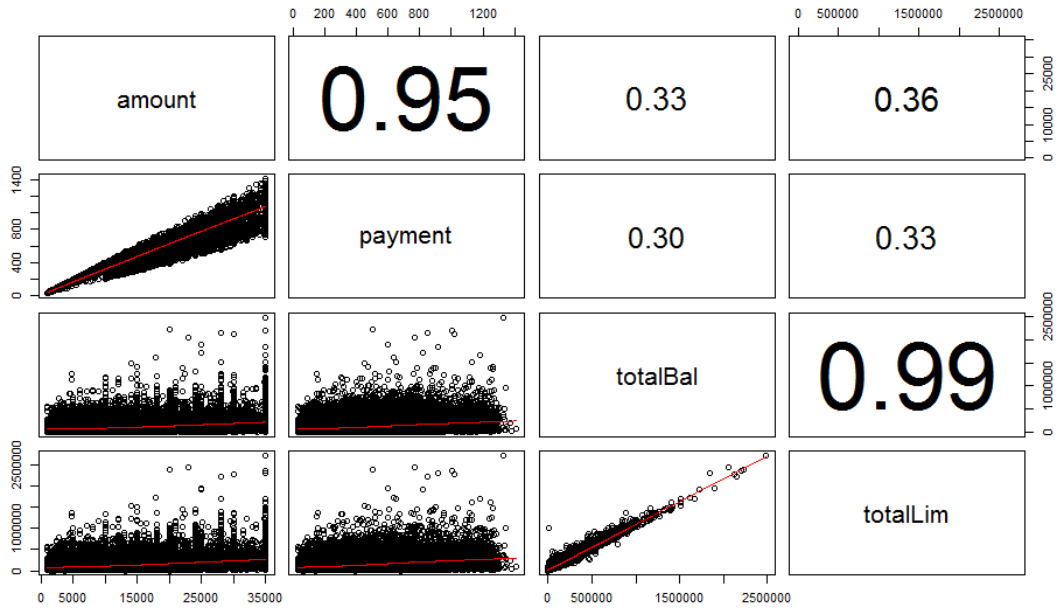
- The 'employment' variable in the dataset has too many different values of varying quality to be grouped into meaningful categories, so we will drop that column from the dataset.
- We will also drop the 'status' variable now that we have created a new response variable and filtered the observations based on those values.
- The 'loanID' variable is a unique identifier for each loan and is not applicable to predicting if an applicant will default on their loan, so we will remove that variable as well.



With a few exceptions, the 'rate' variable and the 'grade' variable have a strong positive linear correlation. Generally the most risk for the loan the higher the interest rate. We will remove the 'rate' variable from the dataset since that is a continuous variable.

Quantitative Variable Correlation

Using the pair() function we performed analysis of all combinations of qualitative variables to assess the amount of correlation.

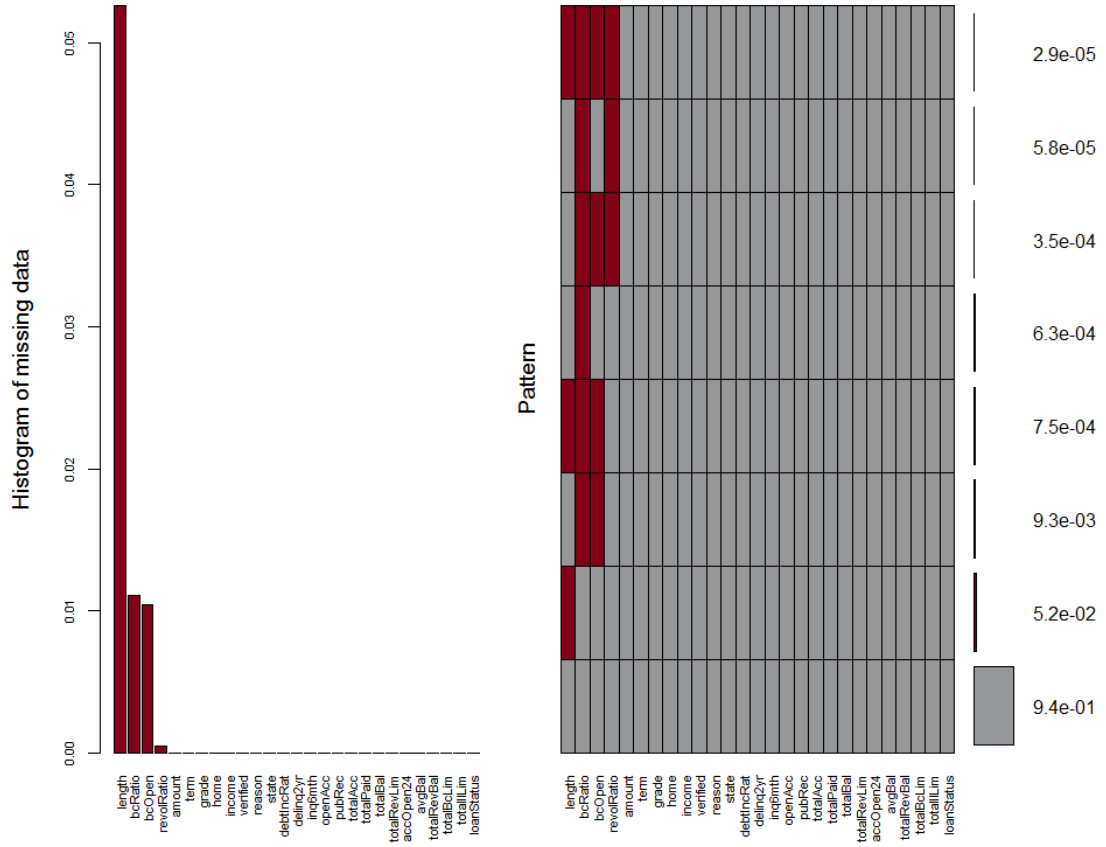


As we can see from the pair analysis the 'amount' and 'payment' variables have a strong positive correlation coefficient of 0.95. The 'totalBal' and 'totalLim' variables have a strong positive correlation coefficient of 0.99.

- We will remove the 'payment' variable since it is computed between the loan amount, loan length (term), and interest rate.
- We will also remove the 'totalLim' variable since it has very strong correlation to the 'totalBal' variable.

Missing Values

First we will check each variable in the dataset to see which variables have missing values.




```
##
##  Variables sorted by number of missings:
##     Variable       Count
##       length 0.052607278
##      bcRatio 0.011081292
##       bcOpen 0.010388711
##   revolRatio 0.000432863
##       amount 0.000000000
##         term 0.000000000
##        grade 0.000000000
```

```
##         home 0.000000000
##       income 0.000000000
##     verified 0.000000000
##       reason 0.000000000
##        state 0.000000000
##   debtIncRat 0.000000000
##    delinq2yr 0.000000000
##      inq6mth 0.000000000
##      openAcc 0.000000000
##       pubRec 0.000000000
##     totalAcc 0.000000000
##    totalPaid 0.000000000
##     totalBal 0.000000000
##  totalRevLim 0.000000000
##    accOpen24 0.000000000
##       avgBal 0.000000000
##  totalRevBal 0.000000000
##   totalBcLim 0.000000000
##   totalIlLim 0.000000000
##   loanStatus 0.000000000
```

Only a few variables have NA values including length, revolRatio, bcOpen, and bcRatio.

There is a 100% overlap between bcOpen missing values that have a corresponding bcRatio missing value. However there is not a strong correlation between missing bcOpen and bcRatio to other variables in the data set.

Because the ratio of missing values is relatively low for each variable (<= 5%), we will use the mice package to impute the missing values. The imputed values will be added back into the dataset to assist with additional analysis.

Feature Engineering

We will begin by consolidating the 'length' variable from 11 different non-NA values down to four levels:

- <1 year
- 1 - 4 years
- 5 - 9 years
- 10+ years

Some of the variables have values of 'n/a' and we will swap those for 'NA' in the dataset.

```
##  < 1 year 1-4 years 10+ years 5-9 years
##      2987     10863     11864      8939
```

The 'verified' variable has two values that are duplicates ('Source Verified' and 'Verified') to be consolidated into a single value of 'Verified'.

```
##                     Not Verified Source Verified        Verified
##                0           10235                0           24418
```

We have also summed up some of the smaller factors in the reason variable into the other factors as follows:

- wedding -> vacation
- renewable_energy -> other
- home_improvement -> house

```
##                  car        credit_card debt_consolidation
##                  281               7843              21044
##                house     major_purchase            medical
##                 2147                619                378
##               moving              other     small_business
##                  215               1598                317
##             vacation
##                  211
```

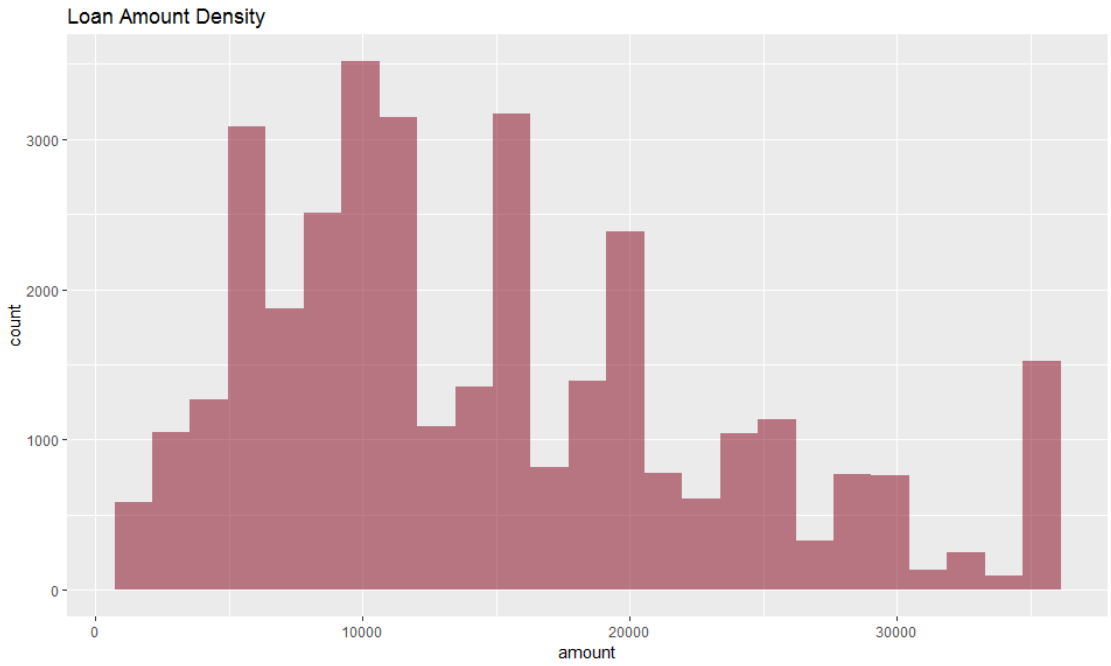
Finally we will group the states into regions using the following alignment:

- New England: (CT, ME, MA, NH, RI, VT)
- Mid-Atlantic: (NJ, NY, PA)
- East North Central (IL, IN, MI, OH, WI)
- West North Central (IA, KS, MN, MO, NE, ND, SD)
- South Atlantic (DE, FL, GA, MD, NC, SC, VA, DC, WV)
- East South Central (AL, KY, MS, TN)
- West South Central (AR, LA, OK, TX)
- Mountain (AZ, CO, ID, MT, NV, NM, UT, WY)
- Pacific (AK, CA, HI, OR, WA)

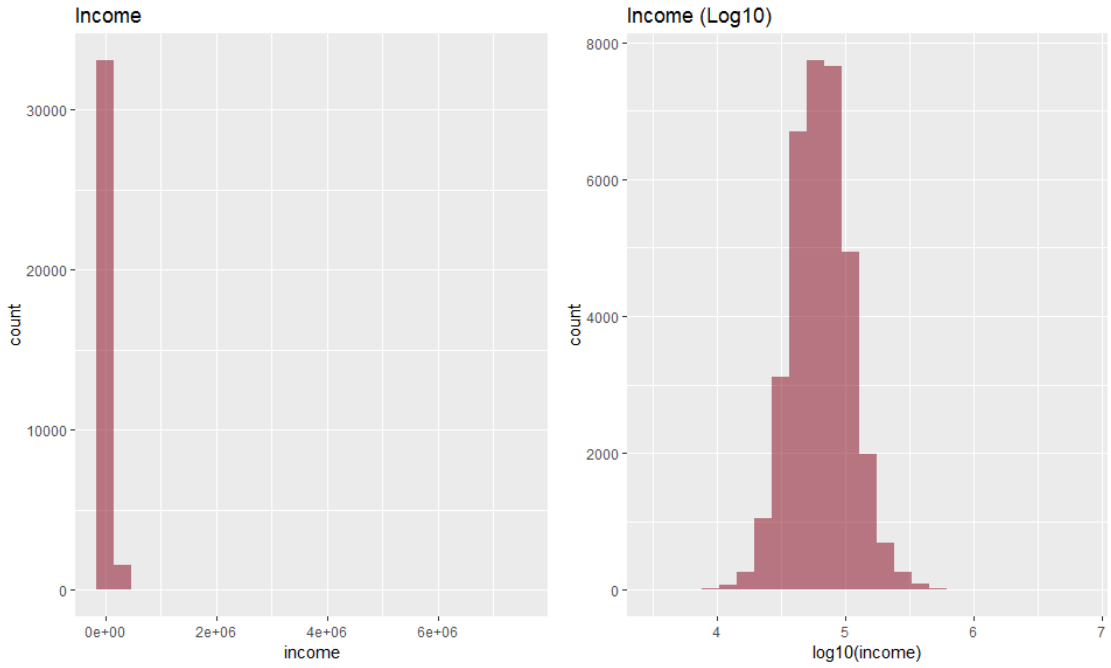
```
## East North Central East South Central       Mid-Atlantic
##               4434               1477               5310
##           Mountain        New England            Pacific
##               2733               1647               6549
##     South Atlantic West North Central West South Central
##               7058               1635               3810
```

Exploring and Transforming the Data

We will now examine the distributions of the quantitative predictor variables. If there is a strong skew we will attempt transformations such a reciprocals, logarithms, cube roots, and square roots to un-skew the data and replace that predictor variable in the dataset with the transformed value.

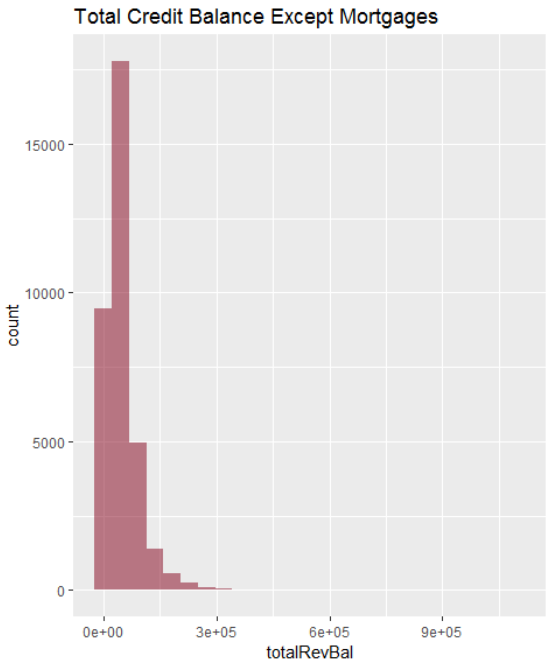


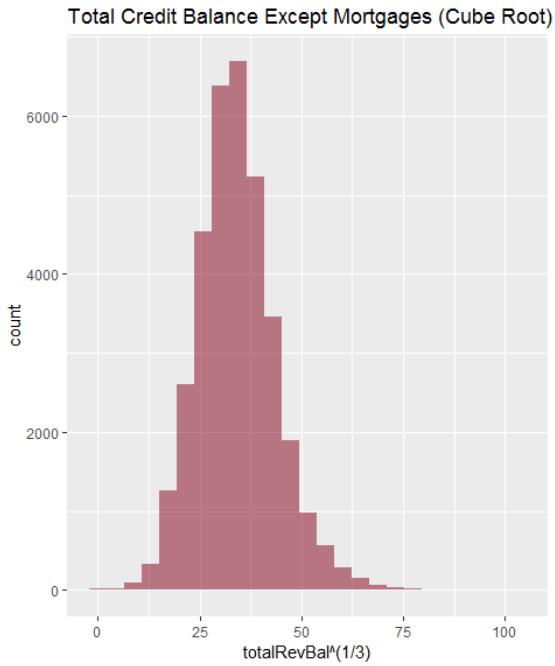
The ‘amount’ variable is only slightly right-skewed so we will not transform its values.

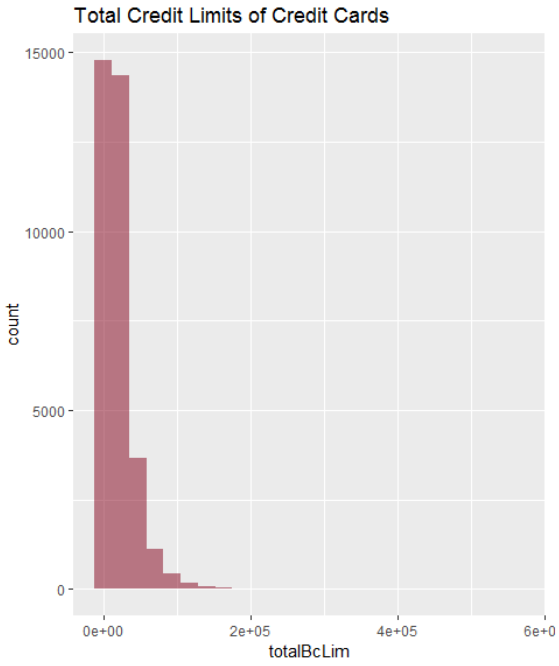


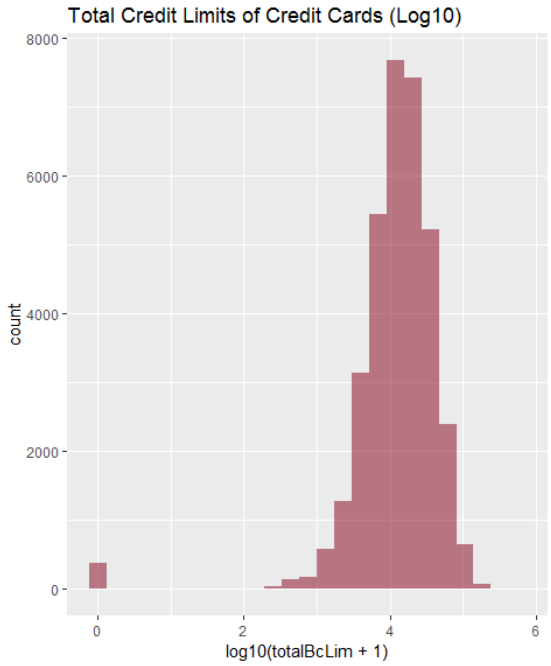
We can see from the density plot that ‘income’ has a strong right skew. Taking the logarithm base 10 of the ‘income’ produces a normally distributed density plot. We will replace the income value in the data set with the transformed value.

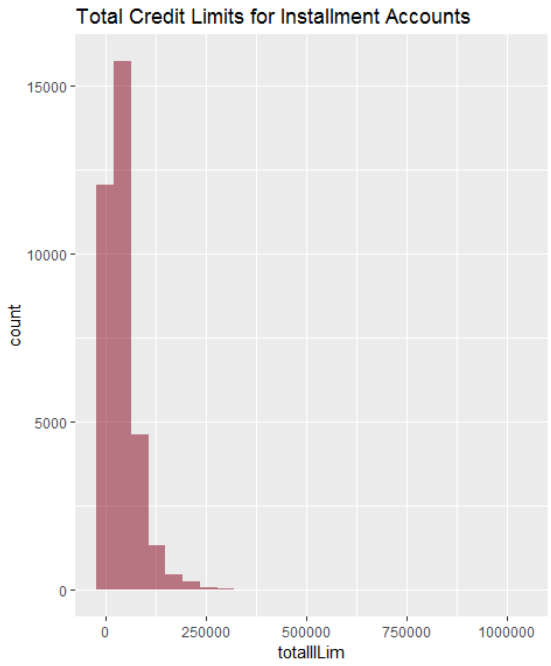
Below are some samples of the skewed and transformed distribution graphs for the remaining quantitative predictor variables.

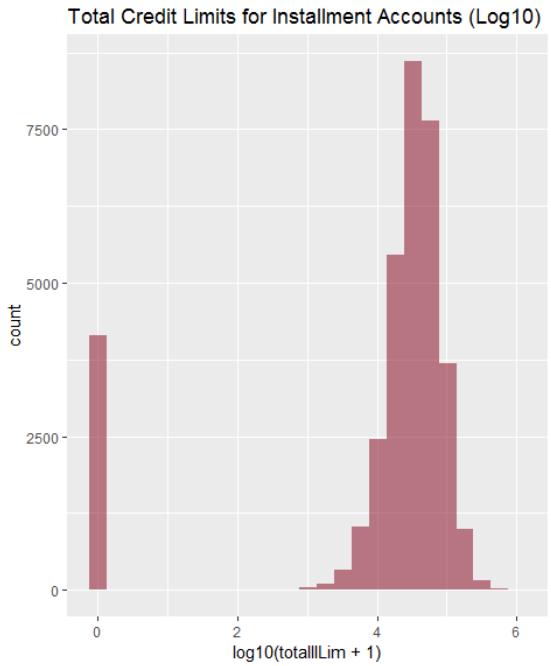

Total Credit Balance Except Mortgages
count
15000
10000
5000
0
0e+00
3e+05
6e+05
9e+05
totalRevBal


Total Credit Balance Except Mortgages (Cube Root)
count
6000
4000
2000
0
0
25
50
75
100
totalRevBal^(1/3)


Total Credit Limits of Credit Cards
count
15000
10000
5000
0
0e+00
2e+05
4e+05
6e+0
totalBcLim


Total Credit Limits of Credit Cards (Log10)
count
8000
6000
4000
2000
0
0
2
4
6
log10(totalBcLim + 1)

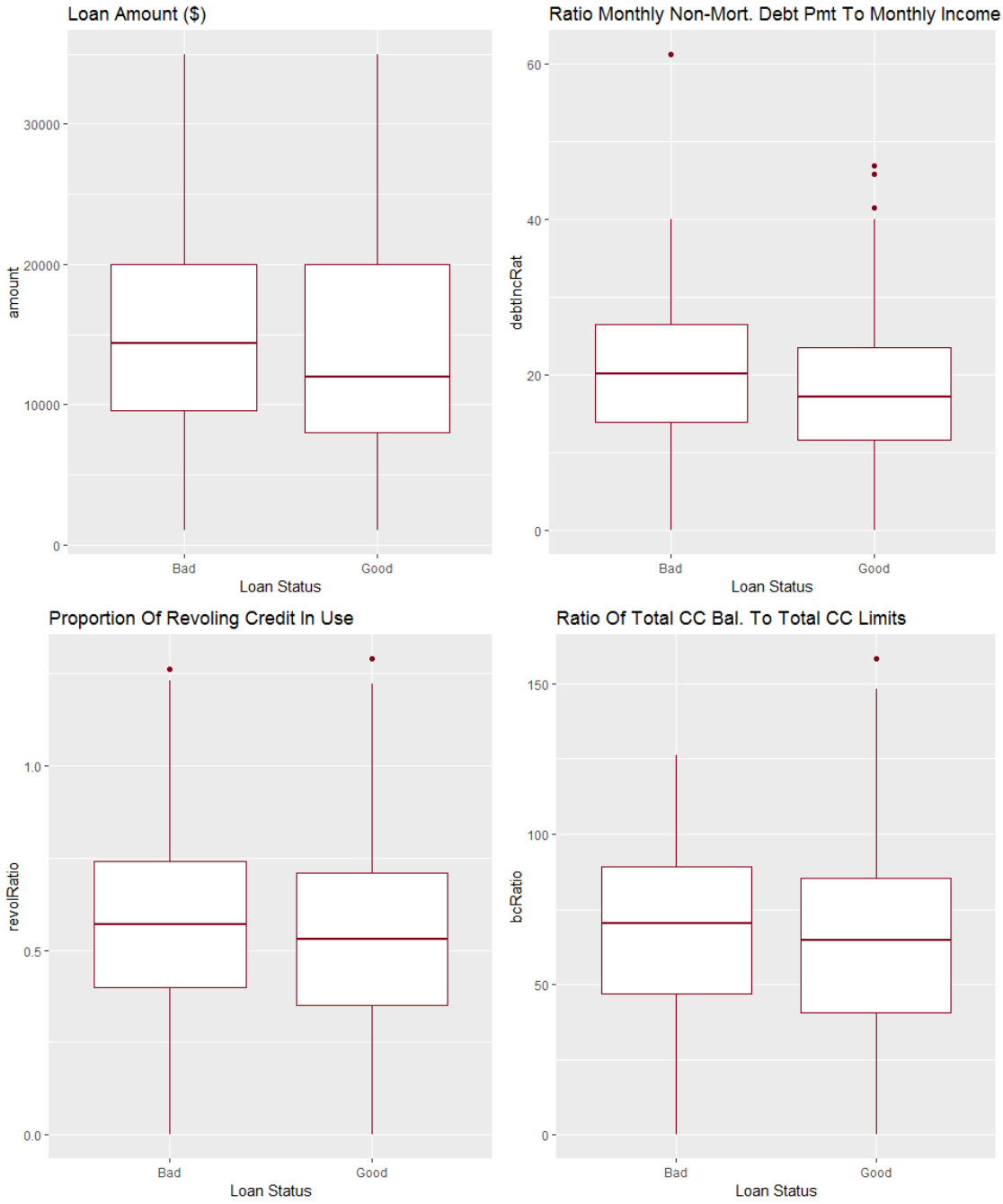




Data Exploration

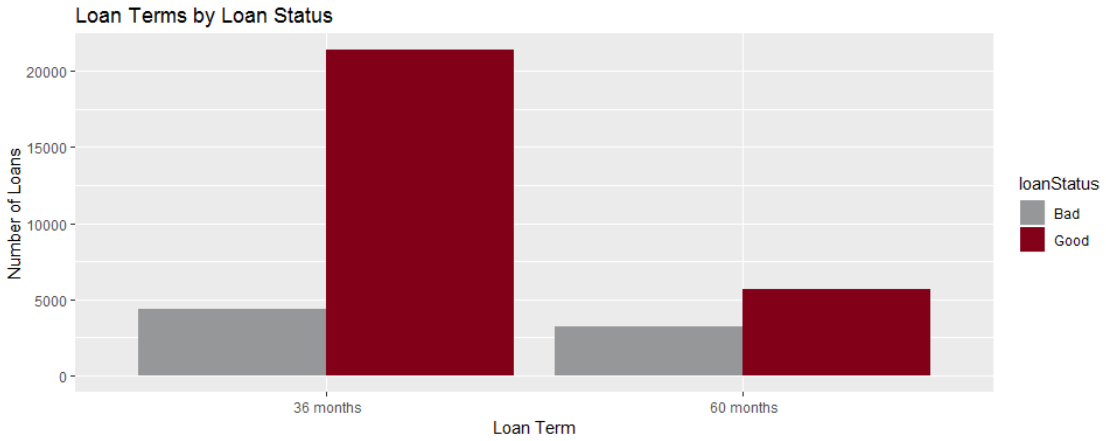
In this section we will make graphs to explore the relationships between the predictors and loan status.

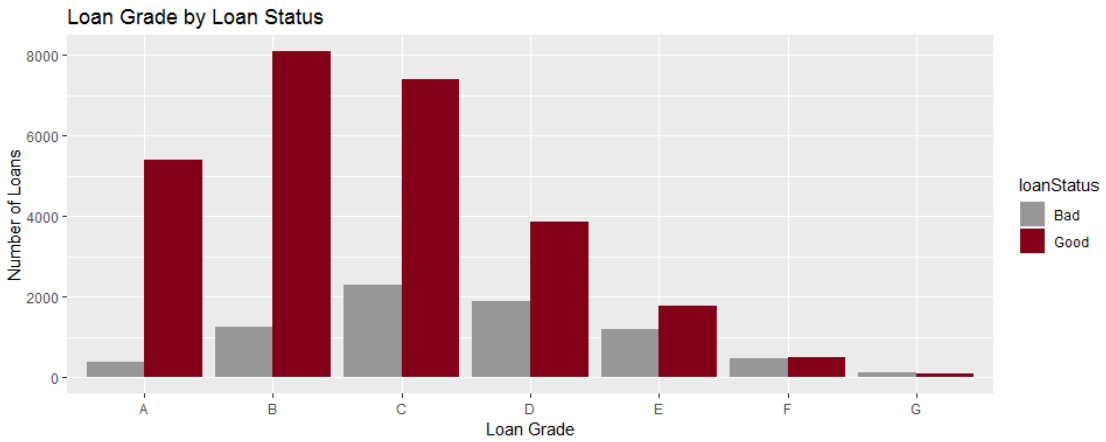
Quantitative Predictors vs. Loan Status



We plotted all of the quantitative predictors against the loan status. Above is a sample of the graphical analysis of the quantitative predictors. The boxplot graphs do not show any particularly strong relationships to the loanStatus variable.

Categorical Predictors





We plotted all of the categorical predictors against the loanStatus response variable. Above is a sampling of the interesting results. A loan term of 60 months seems to have a much higher proportion of 'Bad' loans than loans with a 36 month term. The loan grade also seems to have a higher proportion of 'Bad' loans as the level of risk rises, with a grade of 'G' appearing to have more 'Bad' loans than 'Good'.

The Logistic Model

The target response variable for the prediction model is an indicator of whether a loan will be 'Good' (i.e. paid in full) or 'Bad' (i.e. charged off or defaulted). We took the following steps to create the logistic model on the cleaned and transformed data:

1. Randomly select 80% of the records for a training dataset and the remaining 20% for a testing dataset.
2. Using all of the predictor variables, except for totalPaid, we ran a logistic regression using the training dataset.
3. Using the the model from step 2, we predicted the loan status for loans in the testing dataset.

4. We created a contingency table to determine the overall accuracy of the logistic model on the testing dataset, using a threshold of 0.5.

Logistic Model Results

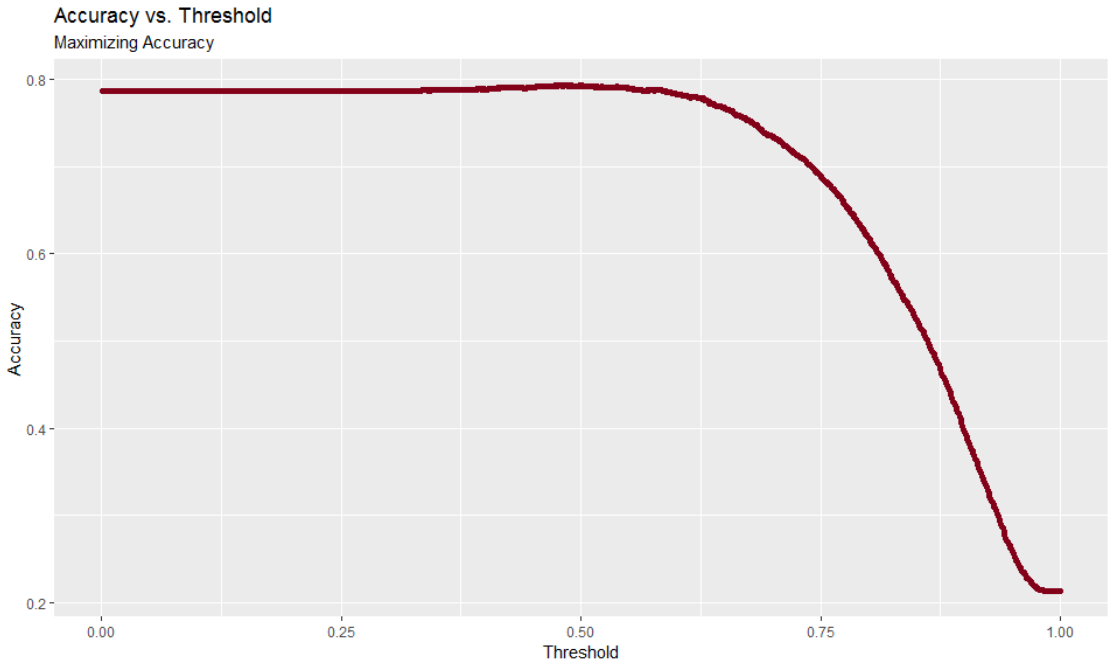
```
##        predGood
##         Bad Loan Good Loan  Sum
##    Bad        204       1270 1474
##    Good       161       5296 5457
##    Sum        365       6566 6931

## [1] "Proportion correctly predicted = 0.794"
```

The logistic model correctly classified 79.4% of the loan statuses in the testing dataset. This is a relatively good model for predicting loan status. The full dataset proportion of good loans was 78.1%. If we assume that the bank approves loans to all of the applicants who receive a prediction of Good from this model (6,566 loans), 80.7% percent of those loans (5,296) would be repaid in full. This model outperforms existing predictive measures.

Optimizing the Threshold for Accuracy

The analysis above uses a threshold value of 0.5. To test if there is a better threshold for predicting bad loans we wrote a procedure to loop through threshold values from 0.001 to 1.000 and check the accuracy at each threshold level.

Threshold Optimization Results




```
## [1] "The threshold value that produces the best accuracy is 0.484 with an
accuracy of 79.4%"
```

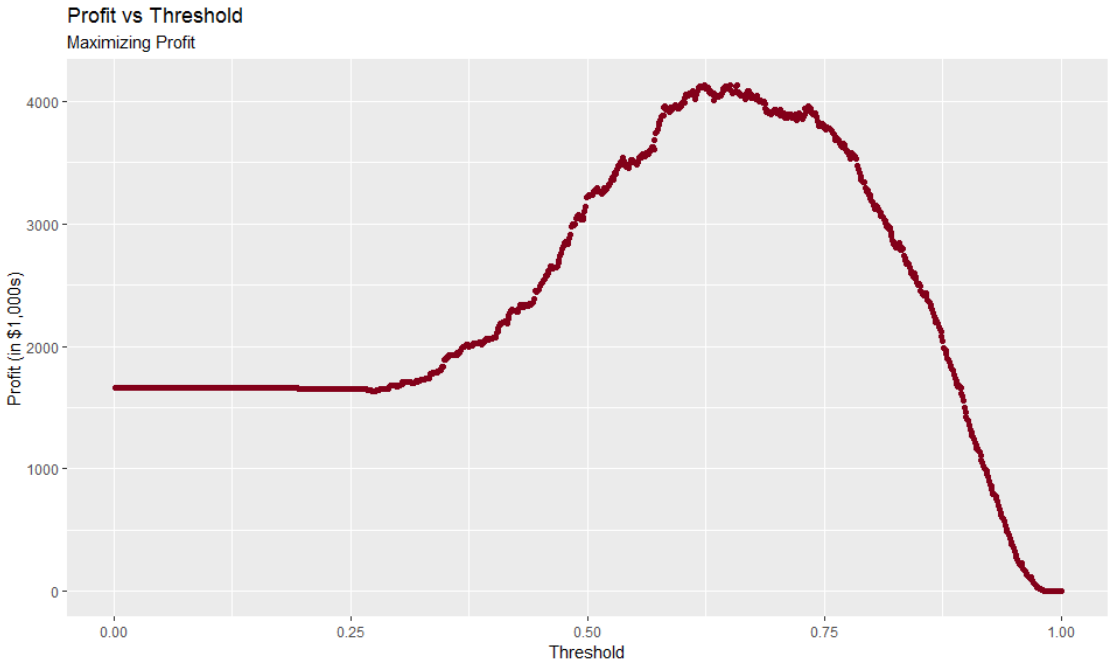
After testing 1,000 different threshold values we are unable to get a better result than the threshold value of 0.5 with an overall accuracy of 0.794. At that threshold level the accuracy for 'Good' loans is 80.7% (5296/6566) and the accuracy for 'Bad' loans is 55.9% (204/365).

Optimizing the Threshold for Profit

We will now test to see if there is a threshold value of that produces a better level of profitability for the bank. For each loan predicted as 'Good' we will calculate the profit as totalPaid - amount.

```
## # A tibble: 2 x 2
##   loanStatus profit
##   <fct>      <chr>
## 1 Bad        ($11,052,722)
## 2 Good       $12,715,887
```

The current level of profitability for 'Good' loans is $12,715,887 from the test dataset, but including losses from 'Bad' loans the net profitability is $1,663,165. Again we will loop through threshold values from 0.001 to 1.000 to determine the threshold value that results in the greatest level of profitability for the bank.




```
## [1] "The threshold with the highest loan profit is 0.658 with a total
profit of $4,133,014"
```

The maximum percentage increase in profit by using this model with a threshold value of 0.658 is 149% over the current method for approving or denying loans. Compared to the

profit level, $12,715,887, from a perfect model (approve all 'Good' loans and deny all 'Bad' loans), this model is only 33% of the perfect level of profitability using the test dataset.

```
## [1] "The overall accuracy of the threshold for highest profit is 76.4%"

##       pred
##        Bad Loan Good Loan  Sum
##   Bad       587       887 1474
##   Good      751      4706 5457
##   Sum      1338      5593 6931
```

The maximum profit threshold (0.658) does not coincide with the maximum accuracy threshold (0.500).

Results Summary

Our recommendation is to use the model as designed in this project, using a threshold level of 0.658 to maximize the bank's profit. The overall accuracy of the model at this threshold is 76.4%, accuracy for 'Good' loans is 84.1%, and accuracy for 'Bad' loans is 43.9%. The profitability at this threshold is $4,133,014, representing a 149% increase over the current method for approving loans.

Model Limitations

The model's overall accuracy level for maximum profit is 76.4% leaves opportunity for further improvement. The following steps could be pursued to see if they improve the model's accuracy.

- Use additional predictor variables to determine if they have significant predictive value
- Include observations on loan applications that were denied along with the existing dataset of approved loan applications