# Untitled

Haoyu Wang

12/9/2020

## Abstract

Banks play a crucial role in market economies. They decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit. Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether a loan should be granted or not.

The credit score, which is a numerical value know as FICO score, ranges from 300 – 850. In general, the higher the score, the lower the risk. The lower score doesn't necessarily mean you can't get a loan, but you would probably pay a higher interest rate. What factors can affect the credit score is the main topic of my project.

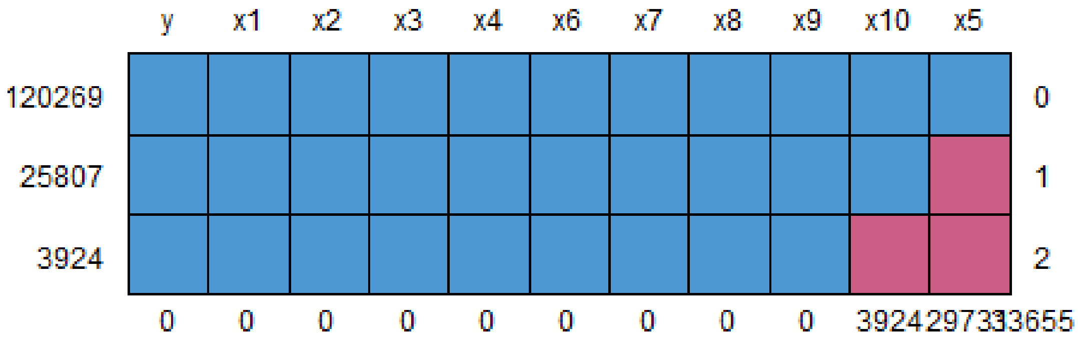
## Introduction

The data were collected from Kaggle: https://www.kaggle.com/c/GiveMeSomeCredit/data. In this website, Training, Test, Sample Entry and Submission Files are provided. I selected training as my dataset.

## EDA and data clean

For the convenience, I replace those names of variables into $y, x_1, x_2, ..., x_{10}$ respectively.

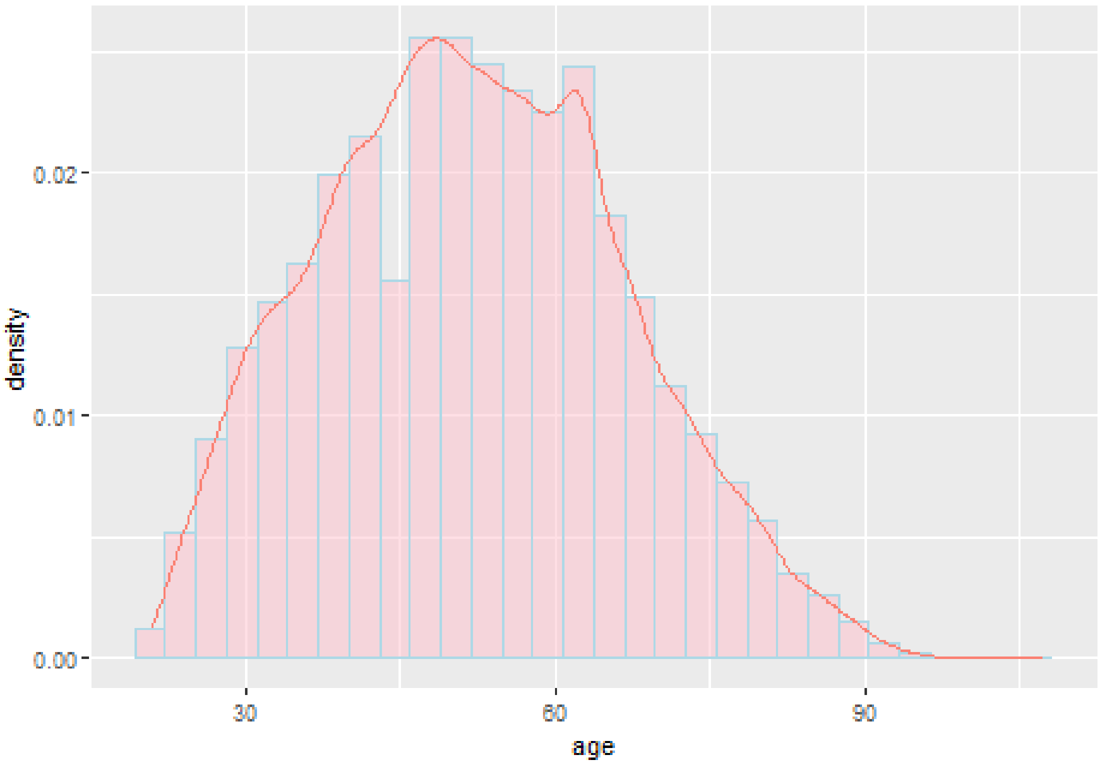
After visualizing the missing values, we can see that the variables $x_5$ and $x_{10}$ have missing values, i.e., there are missing values in the *MonthlyIncome* column and *NumberofDependents* column. The specific situation can be seen in the above table. There are 29,731 missing values in the *MonthlyIncome* column. , *NumberofDependents* has 3,924. For *MonthlyIncome*, since the missing values are kind of large, so I used *na.roughfix()* fuction to fill them up; and for the missing values in *NumberofDependents*, I just deleted them. (Plot is in Appendix)

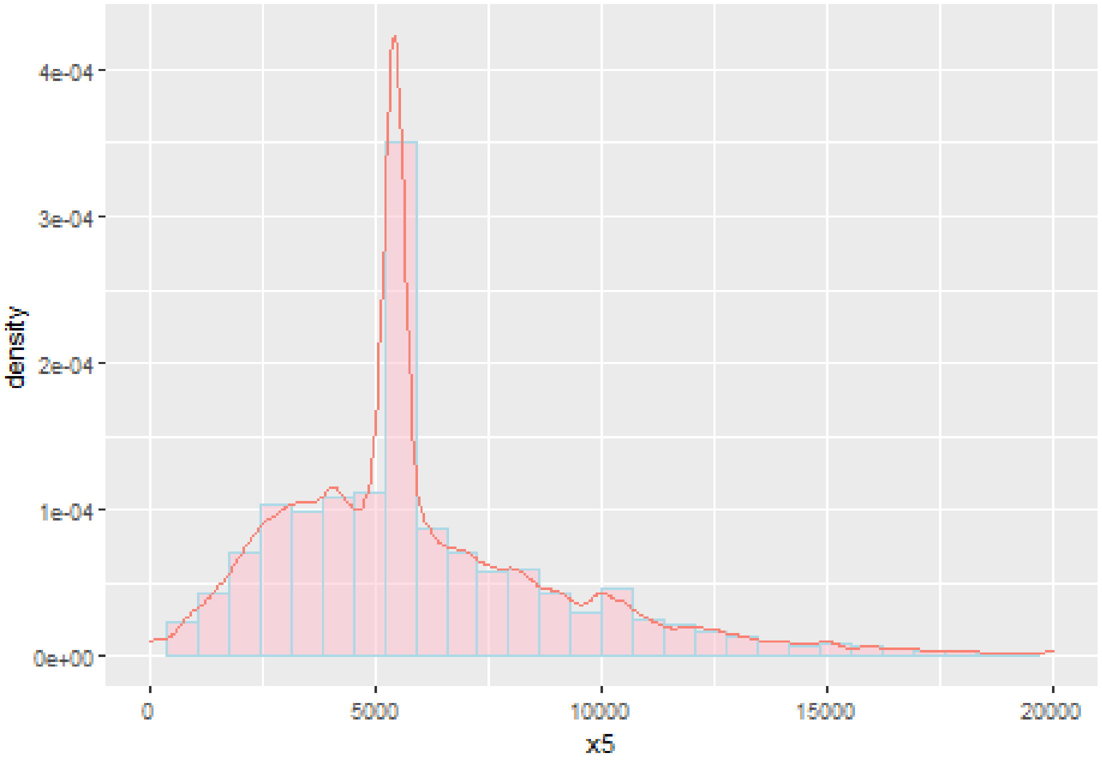


```
##         y x1 x2 x3 x4 x6 x7 x8 x9  x10    x5
## 120269 1  1  1  1  1  1  1  1  1    1     1     0
## 25807  1  1  1  1  1  1  1  1  1    1     0     1
## 3924   1  1  1  1  1  1  1  1  1    0     0     2
##         0  0  0  0  0  0  0  0  0 3924 29731 33655
```

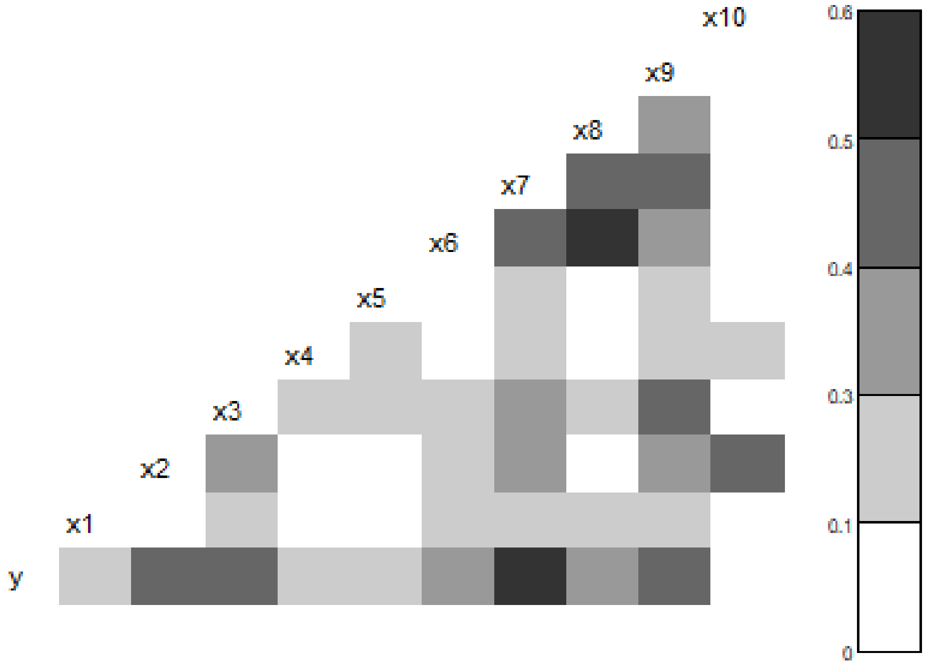
Then, find the outliers of variables in *NumberOfTime30-59DaysPastDueNotWorse*, *NumberOfTimes90DaysLate* and *NumberOfTime60-89DaysPastDueNotWorse*. It can be known that there are two outliers of 96 and 98, so they should be eliminated. Also, found that there was an 0 in *age* which doesn't make any sense, so eliminated it as well.

Then check the distribution of variables, here I took *age* and *MonthlyIncome* as examples(Plots are in Appendix). By the plots, those two variables are roughly normally distributed, which meet the needs of statistical analysis.





Before modeling, we must first check the correlation between variables. If the correlation between variables is significant, it will affect the prediction effect of the model. As can be seen from the figure below, the correlation between the variables is very small. In fact, Logistic regression also need to consider the issue of multicollinearity, however, the correlation of variables is very small, which can be inferred that there is no multicollinearity issue. After modeling, we can also use VIF (variance inflation factor) to test the multicollinearity problem. If there is a multicollinearity problem , then eliminate the variables.

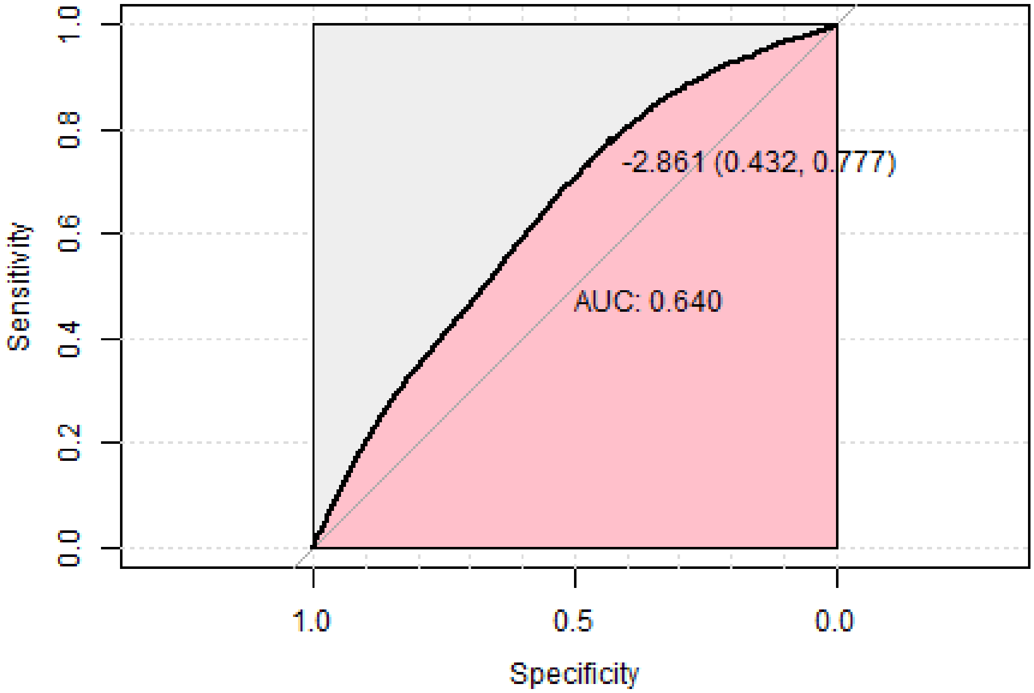


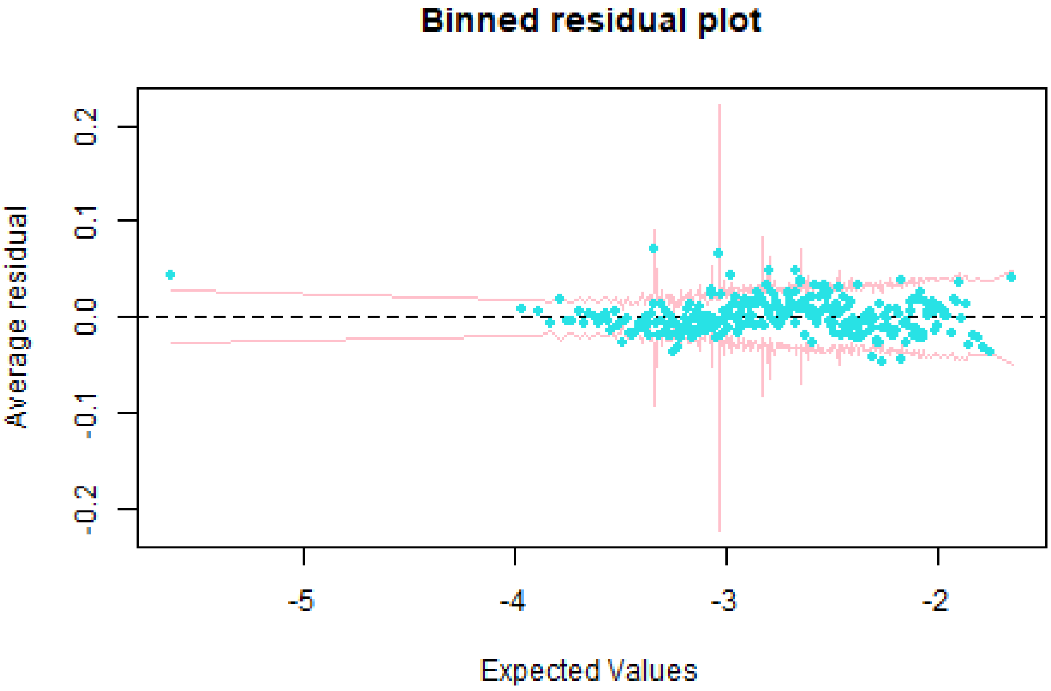
For the variable *SeriousDlqin2yrs*, there is an obvious imbalance issue. The observations that *SeriousDlqin2yrs* is equal to 1 are 9879, which is only 6.6% of all observations. Therefore, it is necessary to process the unbalanced data. Meanwhile, the total of observations are large to some extent, it might take forever when running the Logistic GLMM model, which means splitting is necessary.

```
##
##      0      1
## 136125   9712
```

# Modeling

## Logistic Model



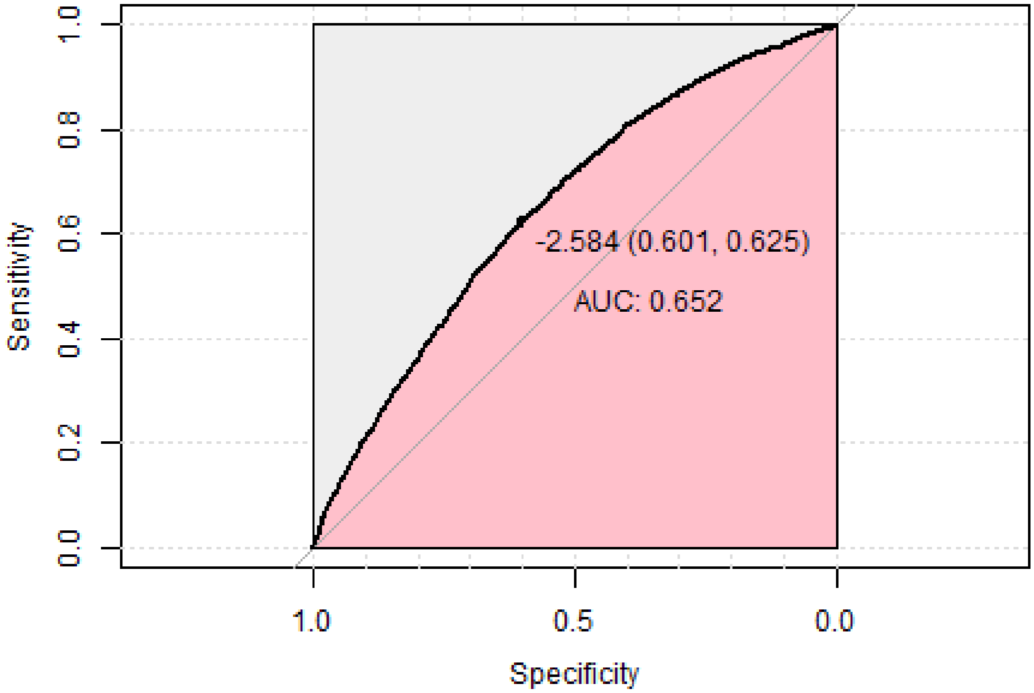


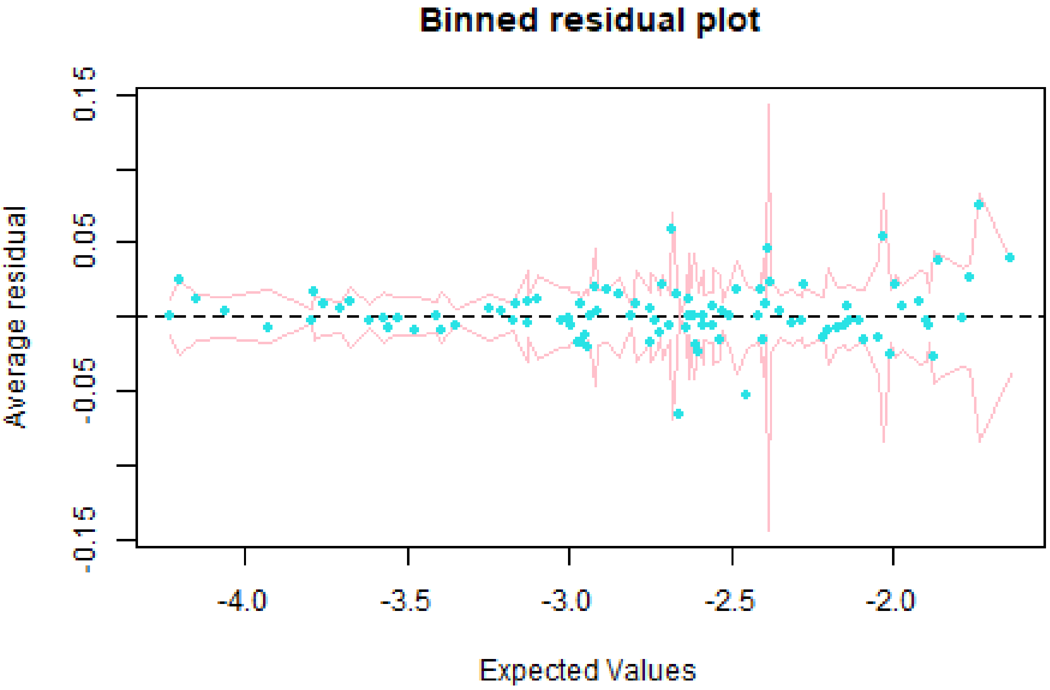
The area under the ROC curve is called the AUC statistic. The larger the statistic, the better the model effect. Generally speaking, AUC greater than 0.75 indicates that the model is very reliable. This model has AUC = 0.640, which is relatively reliable. From binned residual plot, we can find that most of residual plots are in the interval, which means this model is good but not the pefect model.

## Logistic GLMM model

Before using this model, the most important thing is that to make variables that I want to explore as factor. After binning the data, I made a new dataset for easier modeling.

Then, back to Logistic GLMM model:





This model has AUC = 0.652, which is more reliable. Moreover, from residual plot, we can see that almost plots are in the interval. Thus, Logistic GLMM model is more fit.

### Interpretation

I fitted a constant (intercept-only) logistic mixed model (estimated using ML and Nelder-Mead optimizer) to predict Result (formula: Result ~ 1). The model included Age, Income and Family Members as random effects (formula: list(~1 | Age, ~1 | Income, ~1 | Family Members)). . The model's intercept is at -2.75 (95% CI [-3.28, -2.22], p < .001). Within this model, standardized parameters were obtained by fitting the model on a standardized version of the dataset. 95% Confidence Intervals (CIs) and p-values were computed using the Wald approximation.

## Discussion

In this model, I only chose *age*, *MonthlyIncome*, *NumberOfDependents*, so the result might not be accurate. Further, I guess each variable in this dataset can be calculated as a specific number which is credit score. However, I have no idea about how to build a model to calculate the credit score. For increament, I can learn how to build a model to calcualte the score by using all the variables, after that, probably I can also use logistic or logistic GLMM model to chek if my creid score calculation model is correct or not.

## Appendix

The explanations of variables in this dataset:

*SeriousDlqin2yrs*: Person experienced 90 days past due delinquency or worse.

*RevolvingUtilizationOfUnsecuredLines*: Total balance on credit cards and personal lines of credit except real estate and no installment debt like car loans divided by the sum of credit limits.

*age*: Age of borrower in years.

*NumberOfTime30-59DaysPastDueNotWorse*: Number of times borrower has been 30-59 days past due but no worse in the last 2 years.

*DebtRatio*: Monthly debt payments, alimony,living costs divided by monthy gross income.

*MonthlyIncome*: Monthly income.

*NumberOfOpenCreditLinesAndLoans*: Number of Open loans (installment like car loan or mortgage) and Lines of credit (e.g. credit cards).

*NumberOfTimes90DaysLate*: Number of times borrower has been 90 days or more past due.

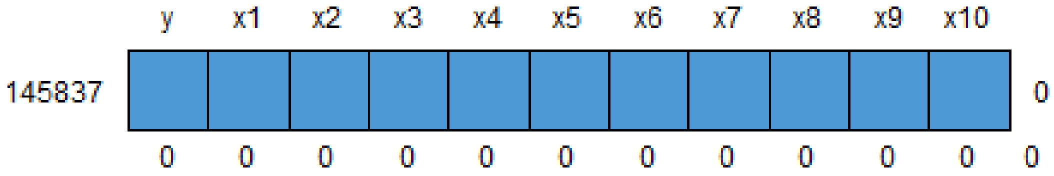
*NumberRealEstateLoansOrLines*: Number of mortgage and real estate loans including home equity lines of credit.

*NumberOfTime60-89DaysPastDueNotWorse*: Number of times borrower has been 60-89 days past due but no worse in the last 2 years.

*NumberOfDependents*:Number of dependents in family excluding themselves (spouse, children etc.).

Missing values in dataset:

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```



```
##        y x1 x2 x3 x4 x5 x6 x7 x8 x9 x10  
## 145837 1  1  1  1  1  1  1  1  1  1   1 0
##        0  0  0  0  0  0  0  0  0  0   0 0
```

Unique values in original dataset:

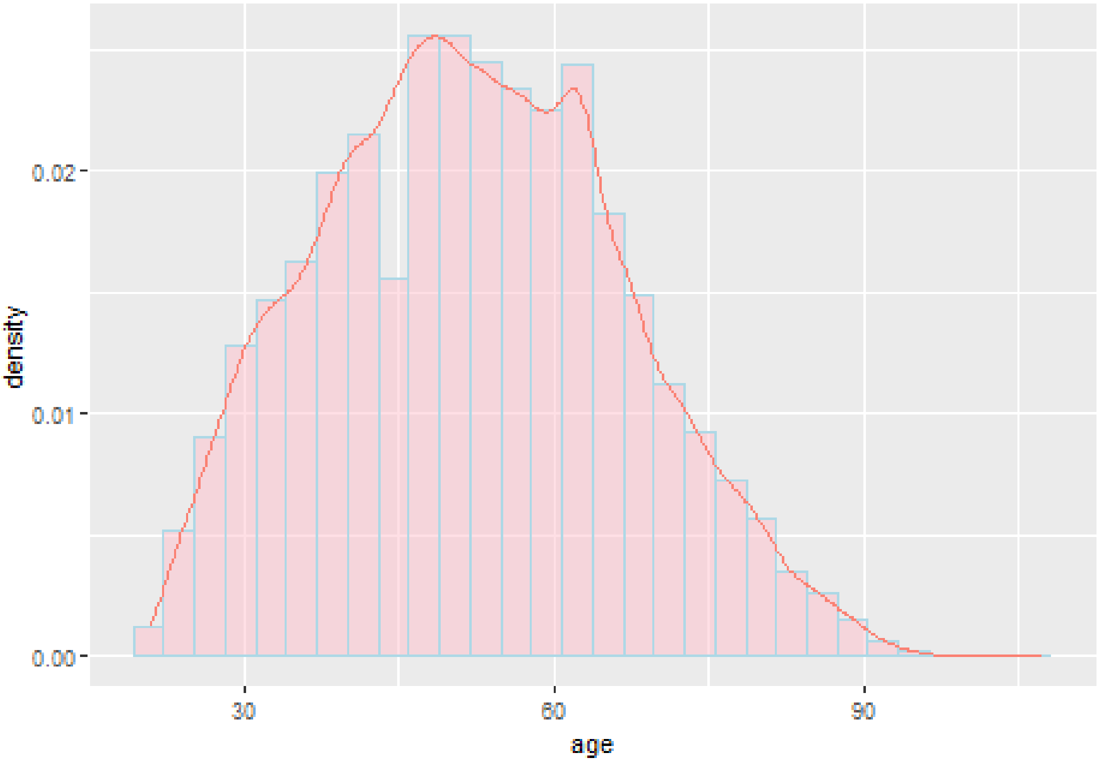
```
##  [1]  2  0  1  3  4  5  7 10  6 12  8  9 13 11
```

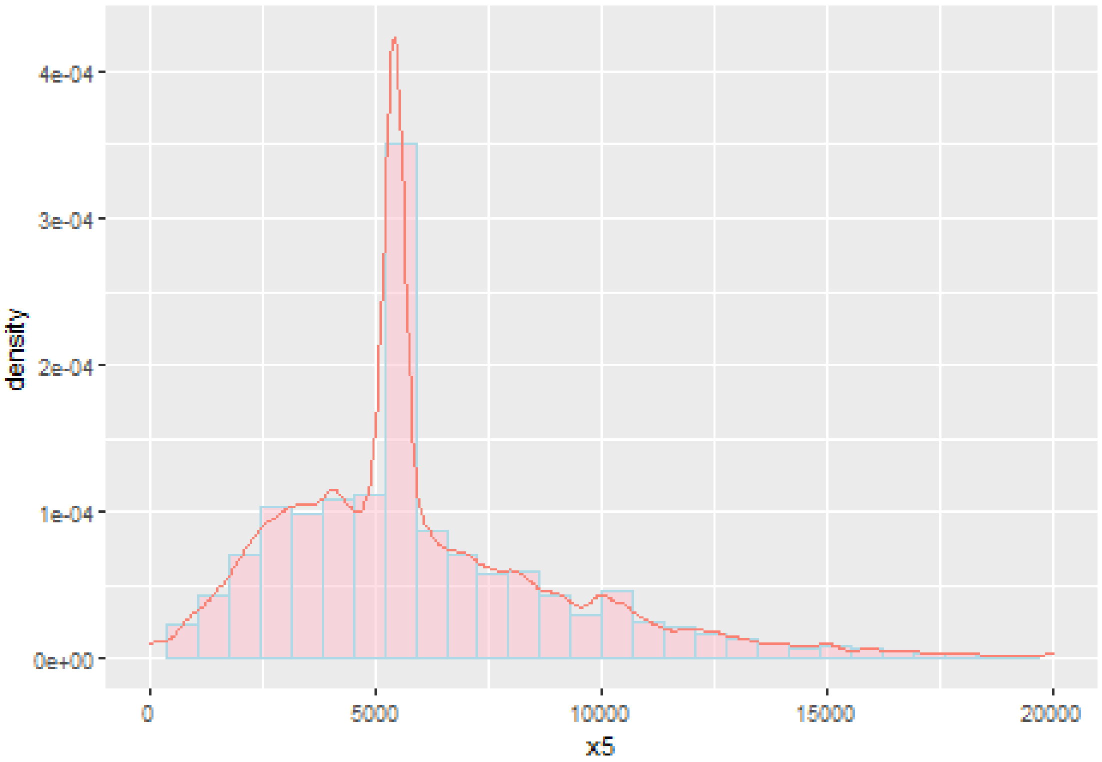
```
##  [1]  0  1  3  2  5  4 10  9  6  7  8 15 11 13 14 17 12
```

```
##  [1]  0  1  2  5  3  4  6  7  8 11  9
```

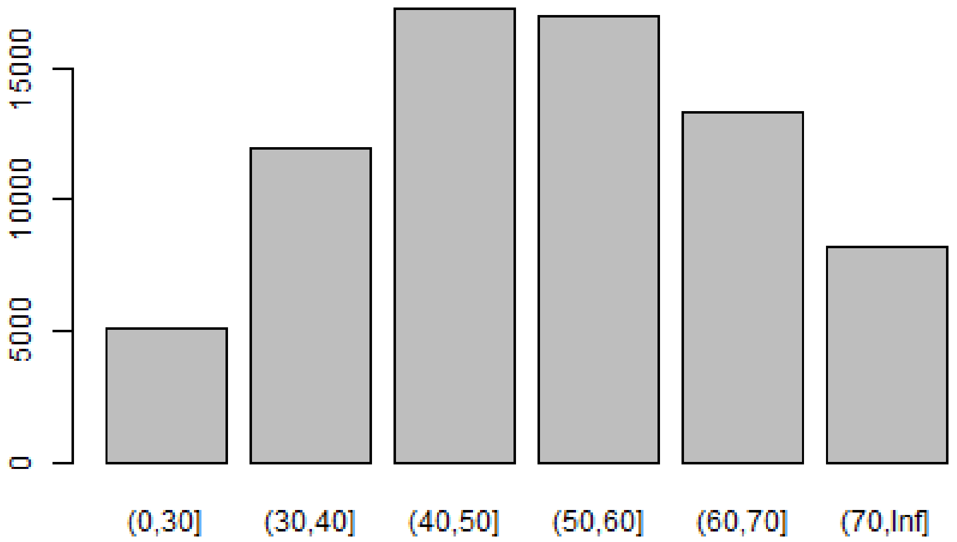
```
##  [1]  45  40  38  30  49  74  57  39  51  46  76  64  78  53  43  25  32  58  50
## [20]  69  24  28  62  42  75  26  52  41  81  31  68  70  73  29  55  35  72  60
## [39]  67  27  36  56  37  66  83  34  44  61  80  47  59  77  48  63  54  33  79
## [58]  65  86  92  23  87  71  90  84  82  22  89  91  85  88  21  93  96  94  95
## [77] 101  97  98 103 102  99 107
```

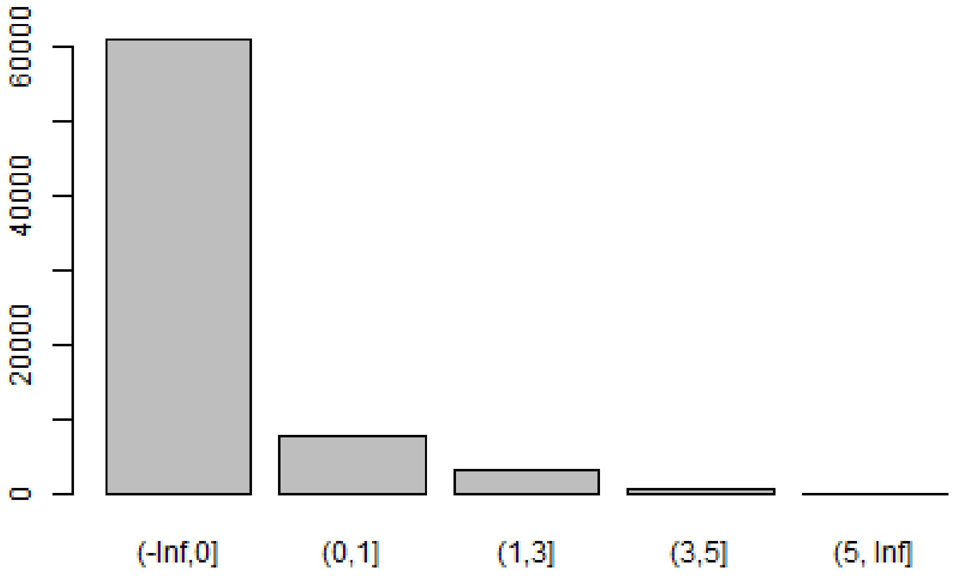
single value checking

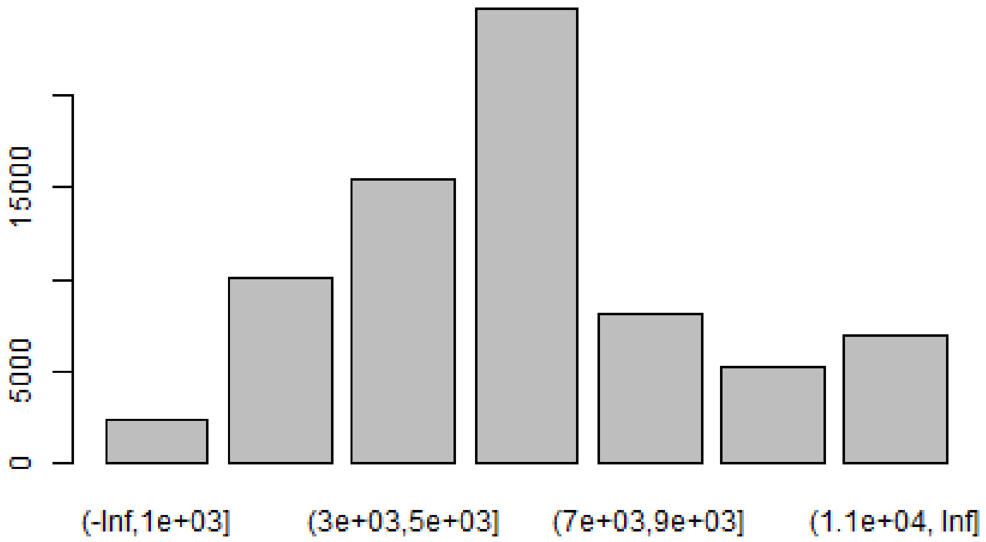


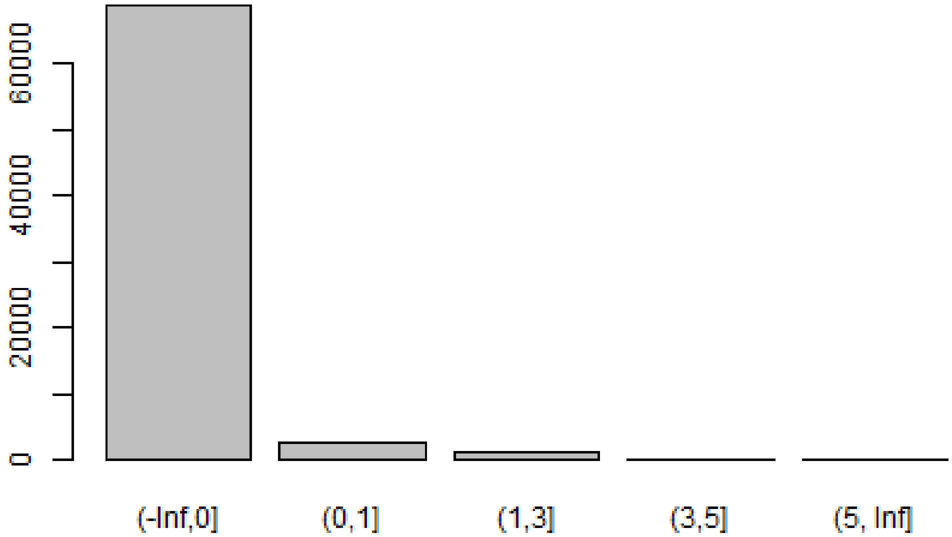


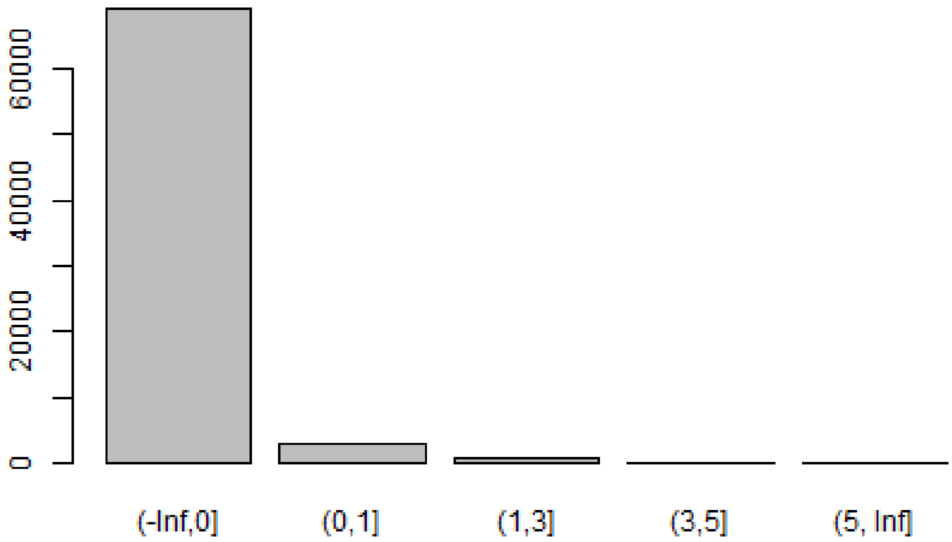
Plots for the variables after binning

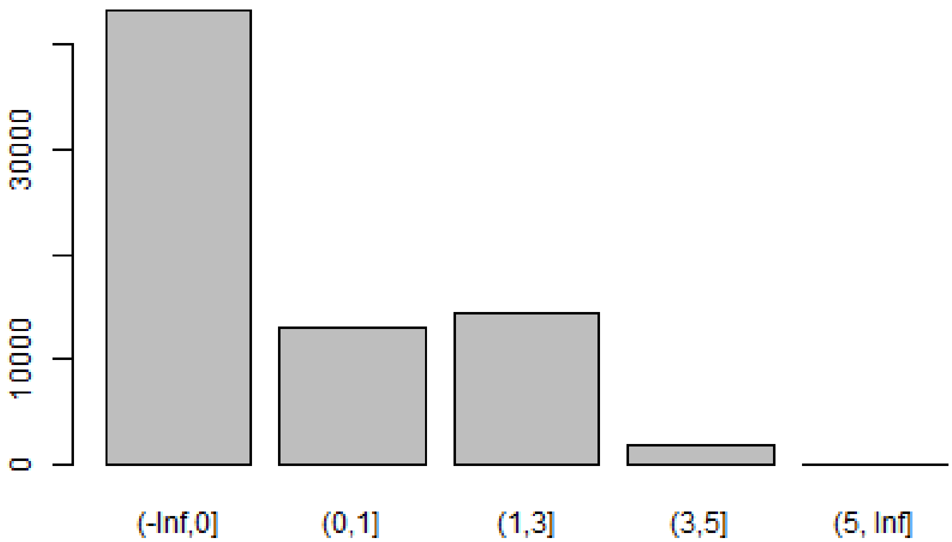



0
20000
40000
60000
(-Inf,0]
(0,1]
(1,3]
(3,5]
(5, Inf]


0
5000
15000
(-Inf,1e+03]
(3e+03,5e+03]
(7e+03,9e+03]
(1.1e+04, Inf]


0
20000
40000
60000
(-Inf,0]
(0,1]
(1,3]
(3,5]
(5, Inf]


0
20000
40000
60000
(-Inf,0]
(0,1]
(1,3]
(3,5]
(5, Inf]



0
10000
30000
(-Inf,0]
(0,1]
(1,3]
(3,5]
(5, Inf]