

# Data 621 - Homework 4

Group 4 Layla Quinones, Ian Costello, Dmitriy Burtsev & Esteban Aramayo

11/21/2021

## Overview

In this homework assignment, we explore, analyze and model a data set containing approximately 8,000 records representing customers at an auto insurance company. Each record has two response variables: The first response variable, **TARGET\_FLAG**, is a 1 or a 0. A “1” means that the person was in a car crash. A zero means that the person was not in a car crash. The second response variable is **TARGET\_AMT**. This value is zero if the person did not crash their car. But if they did crash their car, this number will be a value greater than zero.

Our objective is to build a multiple linear regression model and a binary logistic regression model on the training data to predict the probability that a person will crash their car, and also the amount of money it will cost if the person does crash their car.

## Libraries Used

We use the standard libraries such as **tidyverse**, **ggplot2**, and **caret**. We also make use of the **pROC** package to construct the curves. (**Code Appendix 0.1.1**)

## Data Exploration

As usual, our data are stored on GitHub at our team’s main repository for easy access across team members (**Code Appendix 1.2**). With our initial glimpse of the data, we noticed that there were variables that should be doubles that are classed as character strings (**Code Appendix 1.3**). These seem to be on values with currency, such as income, cars values, and home values. We’ll have to do some data cleaning before we continue with the exploration.

There are 8,161 observations in this data set and 26 columns. We know that **INDEX**, **TARGET\_FLAG** and **TARGET\_AMT** are not predictor variables. This gives us **8,161 observations** with **23 predictors** that are a combination of int, double and character data types. We also see that the character variables will have to be converted to factors in order for us to explore their distributions. Variables such as **INCOME**, **HOME\_VAL**, **BLUEBOOK**, **OLDCLAIM** will be converted to numeric because they are numbers with values that have meaning in their hierarchy.

```
## Rows: 8,161
## Columns: 26
## $ INDEX      <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 19, 20, 2-
## $ TARGET_FLAG <int> 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1-
## $ TARGET_AMT  <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 2946.000, 0.000, 4021.0-
## $ KIDSDRV     <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0-
## $ AGE         <int> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 43, 55, 53, 45-
## $ HOMEKIDS    <int> 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 3, 2, 1-
## $ YOJ         <int> 11, 11, 10, 14, NA, 12, NA, 10, 7, 14, 5, 11, 11, 0, 1-
## $ INCOME       <chr> "$67,349", "$91,449", "$16,039", "", "$114,986", "$125,301-
## $ PARENT1      <chr> "No", "No", "No", "No", "Yes", "No", "No", "No", "No", "No-
## $ HOME_VAL     <chr> "$0", "$257,252", "$124,191", "$306,251", "$243,925", "$0"-
## $ MSTATUS      <chr> "z_No", "z_No", "Yes", "Yes", "Yes", "z_No", "Yes", "Yes", "Yes", "-
## $ SEX          <chr> "M", "M", "z_F", "M", "z_F", "z_F", "z_F", "M", "z_F", "M"-
## $ EDUCATION    <chr> "PhD", "z_High School", "z_High School", "<High School", "-
## $ JOB          <chr> "Professional", "z_Blue Collar", "Clerical", "z_Blue Colla-
```

```

## $ TRAVTIME      <int> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36, 25, 64, 48, ~
## $ CAR_USE       <chr> "Private", "Commercial", "Private", "Private", "Private", ~
## $ BLUEBOOK      <chr> "$14,230", "$14,940", "$4,010", "$15,440", "$18,000", "$17~
## $ TIF           <int> 11, 1, 4, 7, 1, 1, 1, 1, 7, 1, 7, 7, 6, 1, 6, 6, 7, 4, ~
## $ CAR_TYPE       <chr> "Minivan", "Minivan", "z_SUV", "Minivan", "z_SUV", "Sports~
## $ RED_CAR        <chr> "yes", "yes", "no", "yes", "no", "no", "no", "yes", "no", ~
## $ OLDCLAIM       <chr> "$4,461", "$0", "$38,690", "$0", "$19,217", "$0", "$0", "$0", ~
## $ CLM_FREQ        <int> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2~
## $ REVOKED        <chr> "No", "No", "No", "Yes", "No", "Yes", "No", "Yes", "No", "N~
## $ MVR_PTS         <int> 3, 0, 3, 0, 3, 0, 0, 10, 0, 1, 0, 0, 3, 3, 3, 0, 0, 0, 0, ~
## $ CAR_AGE         <int> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 10, 5, 13, 16, ~
## $ URBANICITY      <chr> "Highly Urban/ Urban", "Highly Urban/ Urban", "Highly Urba~

```

## Missing Values

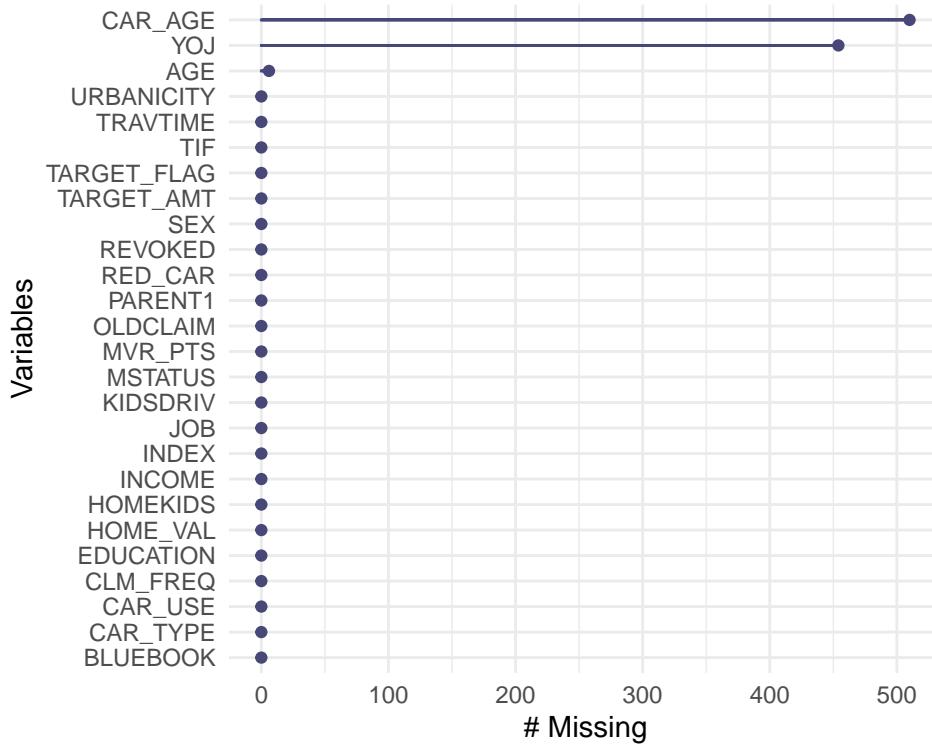


Figure 1. Plot of missing values

As shown in figure 1 ([Code Appendix 1.4](#)), there are missing variables in the columns Car\_AGE, AGE and YOJ. None of these exceed the 10% missing data, so we will continue with all variables for now and not dropping any of them.

## Data Cleaning - Converting Data Types

Before getting too far ahead, using regular expressions, we'll have to do some standard cleaning to remove the \$, z\_, and , and put in a different variable name from numeric strings. We'll also change all other character variables into factors. We'll glimpse the data again to confirm these changes. ([Code Appendix 1.5; 1.6](#))

```

## Rows: 8,161
## Columns: 26
## $ INDEX          <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 19, 20, 2~
## $ TARGET_FLAG    <fct> 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1~
## $ TARGET_AMT     <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 2946.000, 0.000, 4021.0~
## $ KIDSDRIV       <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ AGE            <int> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 43, 55, 53, 45~
## $ HOMEKIDS       <int> 0, 0, 1, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~

```

```

## $ YOJ          <int> 11, 11, 10, 14, NA, 12, NA, NA, 10, 7, 14, 5, 11, 11, 0, 1~
## $ INCOME       <dbl> 67349, 91449, 16039, NA, 114986, 125301, 18755, 107961, 62~
## $ PARENT1      <fct> No, No, No, No, Yes, No, No, No, No, No, No, No, N~
## $ HOME_VAL     <dbl> 0, 257252, 124191, 306251, 243925, 0, NA, 333680, 0, 0, 0, ~
## $ MSTATUS      <fct> No, No, Yes, Yes, No, Yes, Yes, No, No, No, Yes, Yes, ~
## $ SEX           <fct> M, M, F, M, F, F, M, F, M, F, M, M, F, F, M, F, F, F~
## $ EDUCATION     <fct> PhD, High School, High School, <High School, PhD, Bachelor~<fct> Professional, Blue Collar, Clerical, Blue Collar, Doctor, ~
## $ JOB            <int> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36, 25, 64, 48, ~
## $ TRAVTIME      <fct> Private, Commercial, Private, Private, Commercial~<dbl> 14230, 14940, 4010, 15440, 18000, 17430, 8780, 16970, 1120~<int> 11, 1, 4, 7, 1, 1, 1, 7, 1, 7, 7, 6, 1, 6, 6, 7, 4, ~<fct> Minivan, Minivan, SUV, Minivan, SUV, Sports Car, SUV, Van, ~<fct> yes, yes, no, yes, no, no, no, yes, no, no, no, no, yes, y~<dbl> 4461, 0, 38690, 0, 19217, 0, 0, 2374, 0, 0, 0, 0, 0, 0, 5028, 0, ~<int> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2~<fct> No, No, No, No, Yes, No, No, Yes, No, No, No, No, Yes, No, ~<int> 3, 0, 3, 0, 3, 0, 0, 10, 0, 1, 0, 0, 3, 3, 3, 0, 0, 0, 0, ~<int> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 10, 5, 13, 16, ~<fct> Highly Urban/ Urban, Highly Urban/ Urban, Highly Urban/ Ur~
## $ URBANICITY    <fct> Highly Urban/ Urban, Highly Urban/ Urban, Highly Urban/ Ur~

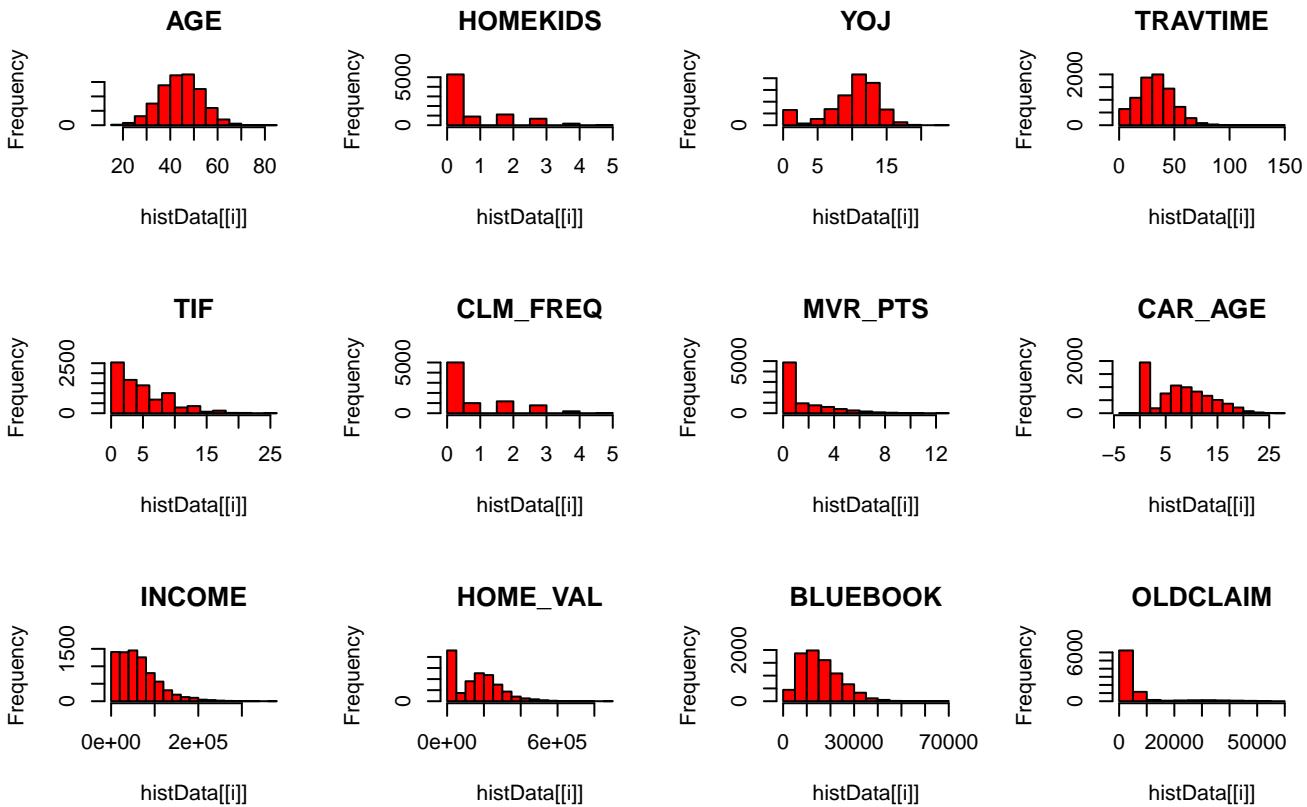
```

## Summary Statistics

Displaying summary statistics ([Code Appendix 1.7](#)) again to confirm data cleaning and get a sense of the spread of the data elements.

## Feature Histograms

For each of the variables, these histograms in figure 2 ([Code Appendix 1.8](#)) provide a nice overview of each feature, its variation, and paths for potential transformations later on for model construction. Histograms are a quick way to see the shape of the distributions for each feature. Of note is the only normally distributed variable, age. The other features appear skewed to some degree. We can also begin to see the affect of outliers that we'll have to account for later on.



**Figure 2. Feature histograms**

## Outlier Analysis

### Feature Box Plots

Let's identify the variables with outlier values using boxplots. (**Code Appendix 1.9**) From these initial box plots we can see that there are some outliers. In particular, TRAVTIME, INCOME, and HOME\_VAL have many outliers which are spread out more compared to the other variables.

## Insurance Data Variables – PART 1

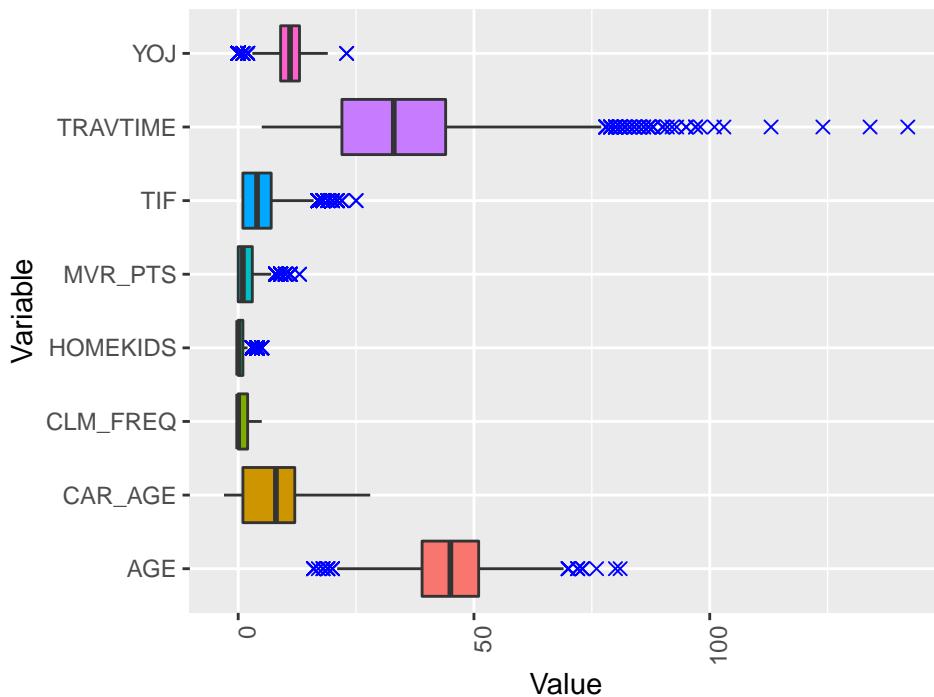


Figure 3. Feature box plot, part 1\*

## Insurance Data Variables – PART 2

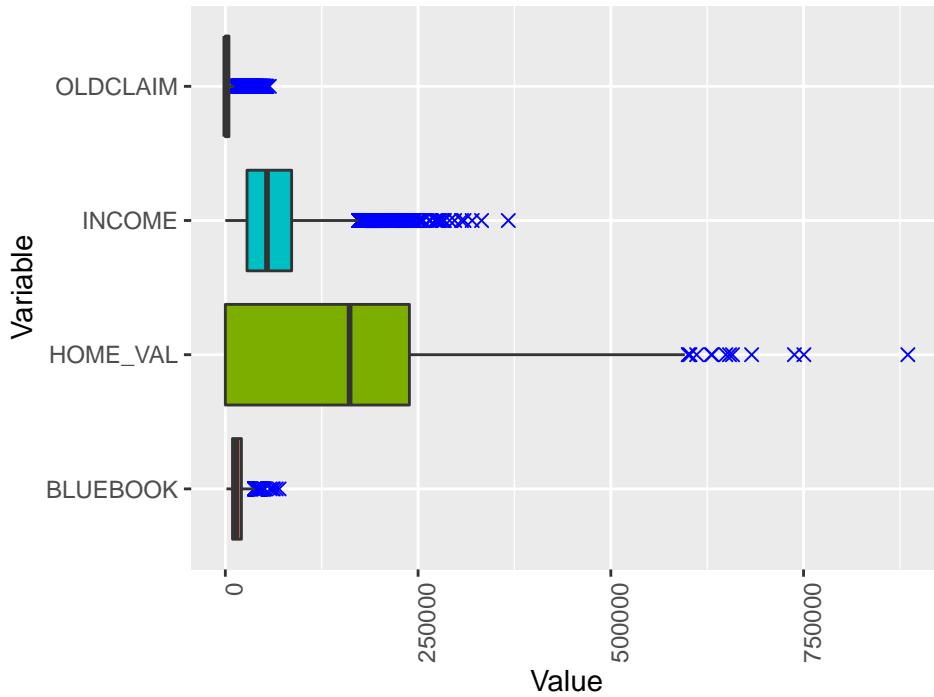


Figure 4. Feature box plot, part 2

## Categorical Predictors - with target variable

The figures below (Code Appendix 1.10) show each categorical feature and the prevalence of each factor. The balance and imbalance of factors may affect how models are built and function in the later part of this project. In reviewing each factor, the imbalance may lead us to select or exclude features.

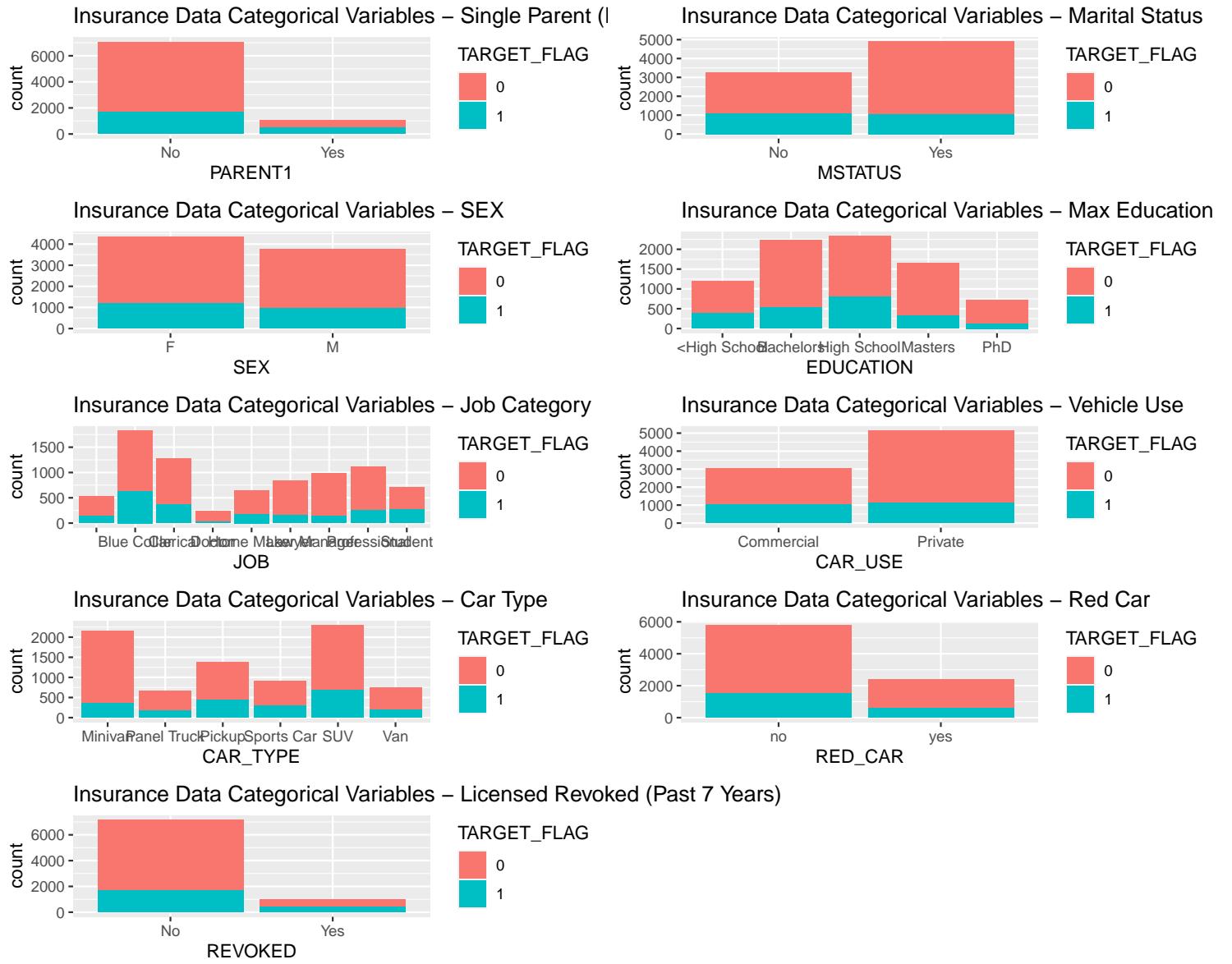
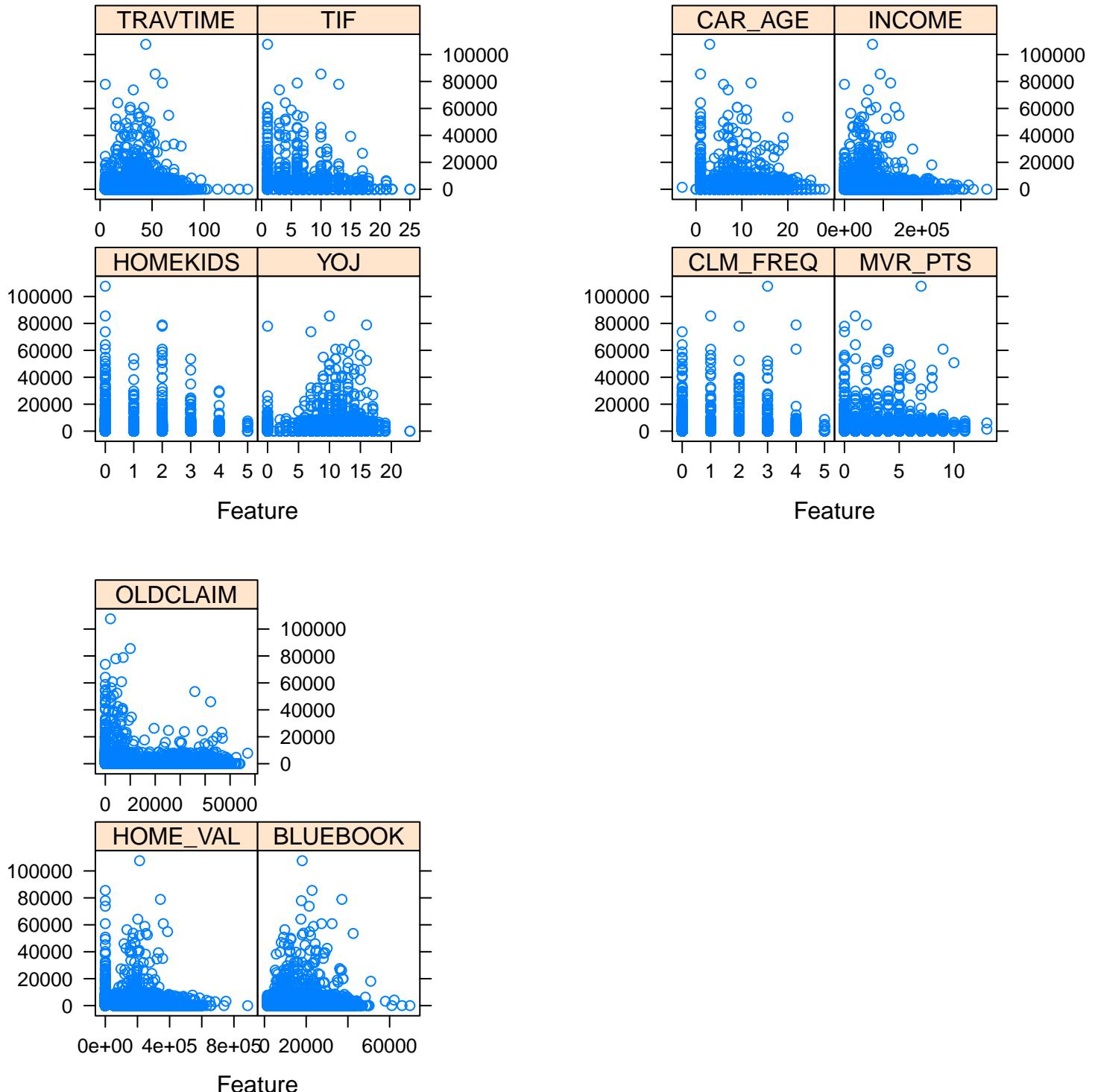


Figure 5. Categorical variables showing balance/imbalance

## Numeric Data - Relationship to Target

These plot visualizations (Figure 6) ([Code Appendix 1.11](#)) gives us an idea of the outliers we have in each variable, but does not give us a good sense of the distribution. We can use the histograms (Figure 2) above to interpret shape. If the notches of two boxes do not overlap, then this suggests that the medians are significantly different.

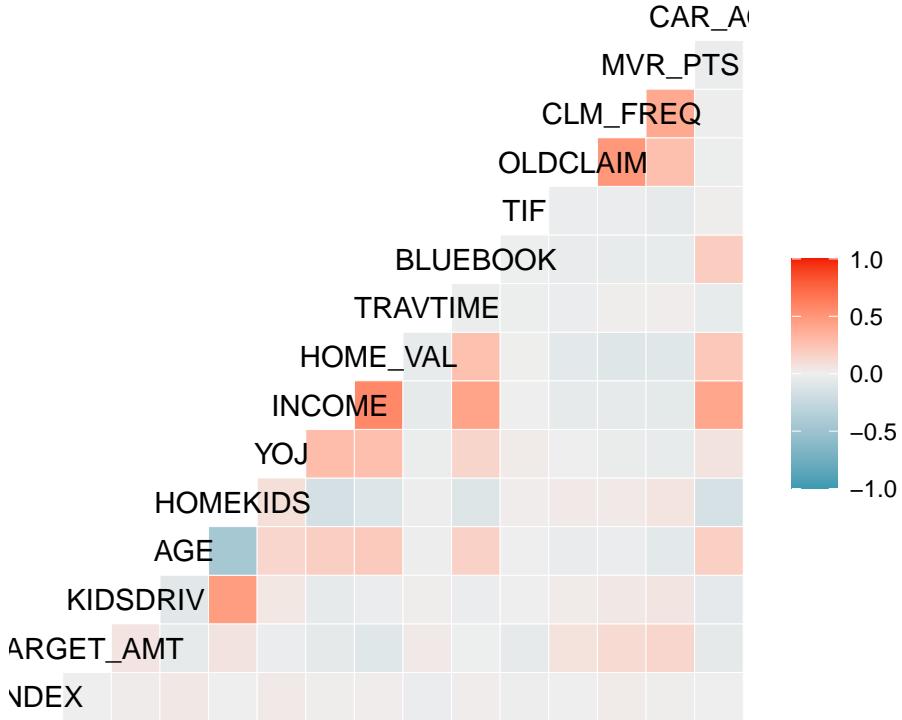
For the features we see some outliers, we can decide to either throw out that variable out altogether and not consider it in our models or impute the outliers with median values. Before deciding on a course of action, we'll look at a few other things.



**Figure 6. Feature plots**

## Correlation

Let's use a heat map ([Code Appendix 1.12.1](#)) to see the level of correlation of the numeric predictor variables. From the correlation matrix (figure 7) below and from the `findCorrelation()` function ([Code Appendix 1.12.2](#)), there does not appear to be any multiple collinearity we have to account for.



**Figure 7.** Correlation matrix

```
## All correlations <= 0.75
```

```
## character(0)
```

## Data Preparation

### Impute Missing Values

For the missing values in `Age` and `YOJ` we will impute with the mean, and for `HOME_VAL`, `INCOME`, and `CAR_AGE` we will use the median. We determine this by observing their distribution and if there is more skew we'll take the median, more normal and we'll use mean. ([Code Appendix 2.2; 2.3](#))

### Variable Importance

To determine the variable importance the following steps were taken ([Code Appendix 2.4; 2.5](#)):

- A training data frame `prepTrainA` was prepared for the `TARGET_FLAG` response variable and its associated predictor variables.
- A training data frame `prepTrainB` was prepared for the `TARGET_AMT` response variable and its associated predictor variables.
- Using the `prepTrainA` data frame, a classification model `modelA` was trained using the Learning Vector Quantization (`lvq`) method. From it, the variable importance was summarized and plotted.

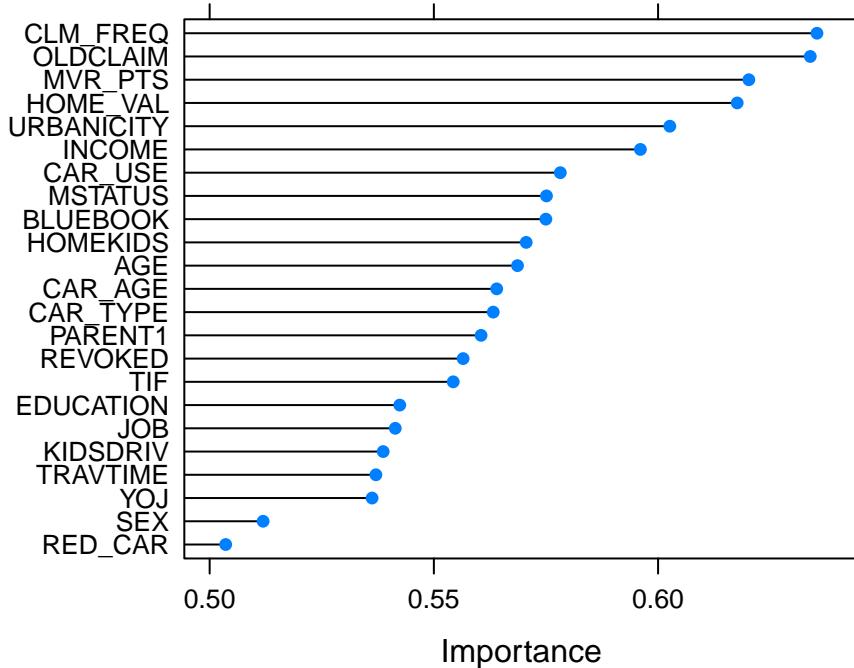
```
## ROC curve variable importance
##
##          Importance
## CLM_FREQ      0.6354
## OLDCLAIM     0.6339
## MVR PTS      0.6202
```

```

## HOME_VAL      0.6176
## URBANICITY   0.6026
## INCOME        0.5961
## CAR_USE       0.5782
## MSTATUS       0.5751
## BLUEBOOK      0.5750
## HOMEKIDS     0.5706
## AGE           0.5686
## CAR_AGE       0.5640
## CAR_TYPE      0.5632
## PARENT1       0.5605
## REVOKED       0.5565
## TIF           0.5543
## EDUCATION     0.5424
## JOB           0.5414
## KIDSDRV       0.5387
## TRAVTIME      0.5371
## YOJ           0.5362
## SEX           0.5119
## RED_CAR       0.5036

```

**Table 1.** Feature importance - part A



**Figure 8.** Plot of feature importance - part A

According to the plots above (figure 8), we can predict which variables would contribute best to the categorical predictions for TARGET\_FLAG. We can use this to inform our data transformations.

Using the prepTrainB data frame, a classification/regression model `modelB` was trained using the Generalized Linear Model (`glm`) method. From it, the variable importance was summarized and plotted.

```

## glm variable importance
##
## only 23 most important variables shown (out of 37)
##
##                                     Overall

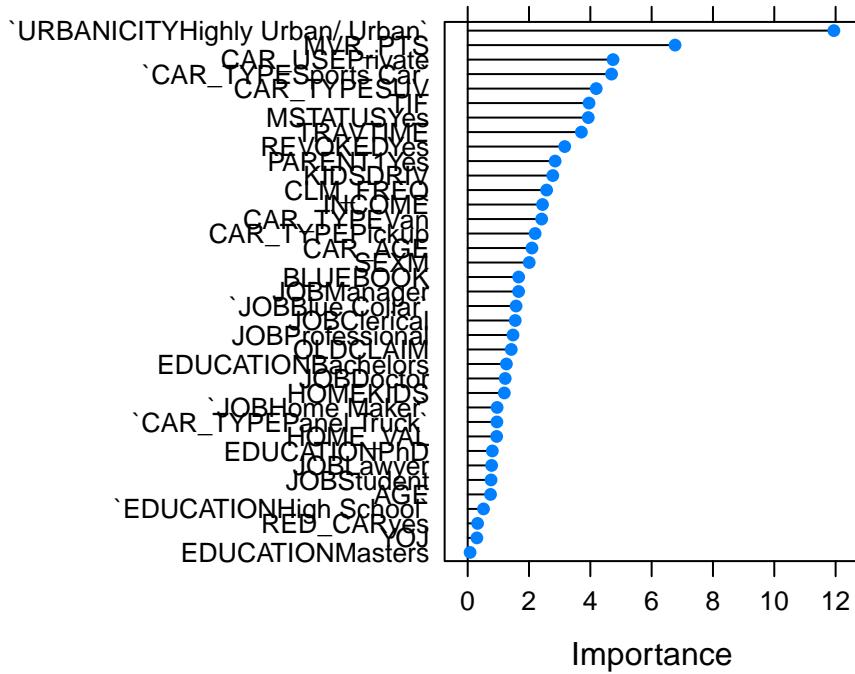
```

```

## 'URBANICITYHighly Urban/ Urban' 11.944
## MVR PTS 6.764
## CAR_USEPrivate 4.741
## 'CAR_TYPESports Car' 4.692
## CAR_TYPESUV 4.193
## TIF 3.958
## MSTATUSYes 3.932
## TRAVTIME 3.708
## REVOKEDYes 3.166
## PARENT1Yes 2.852
## KIDSDRV 2.776
## CLM_FREQ 2.574
## INCOME 2.441
## CAR_TYPEVan 2.413
## CAR_TYPEPickup 2.200
## CAR_AGE 2.096
## SEXM 2.007
## BLUEBOOK 1.663
## JOBManager 1.660
## 'JOBBlue Collar' 1.578
## JOBClerical 1.550
## JOBProfessional 1.478
## OLDCLAIM 1.420

```

**Table 2.** Feature importance - part B



**Figure 9.** Plot of feature importance - part B

According to the plots above, we can predict which variables would contribute best to the numerical predictions for TARGET\_AMT. We can use this to inform our data transformations.

## Train Test Split

We partition (**Code Appendix 2.6**) the training data and set a seed in two data sets. One to be used for training purposes and one for validation/testing purposes.

# Model Building

## Binary Logistic Regression for dependent variable TARGET\_FLAG

### Binary Logistic Regression Model 1

For this model (**Code Appendix 3.2.1**), we only include the predictor variables that have **theoretical effect on probability of collision**, which was provided as part of the definition of the variables.

Additionally, we remove the variables that were deemed as “urban legends,” such as **RED\_CAR** and **SEX**. From our importance variable model **importanceA**, we know that the variables **RED\_CAR** and **SEX** ranked in the bottom 2 items of the importance list of 23 items. Hence, we don’t include them. We also exclude variables having a theoretical “unknown effect” on probability of collision, such as **EDUCATION**.

$AIC = 6570.6$

### Binary Logistic Regression Model 2

In order to improve on our first model, we use all the variables from Model 1, but we exclude the variables **Y0J**, which proved to be the least statistically significant for our Model 1. (**Code Appendix 3.2.2**)

Additionally, we include the variables **OLDCLAIM** and **URBANICITY**, which ranked 4th and 5th in our list of 23 predictor variable importance model **importanceA**.

$AIC = 6053.2$

```
## ROC curve variable importance
##
##      only 5 most important variables shown (out of 23)
##
##              Importance
## CLM_FREQ      0.6354
## OLDCLAIM      0.6339
## MVR_PTS       0.6202
## HOME_VAL      0.6176
## URBANICITY    0.6026
```

**Table 3. Feature importance - Binary model 2**

We can see a significant improvement on the **residual deviance** and **AIC** values.

### Binary Logistic Regression Model 3

In order to improve even more on our previous model, we add the variables **BLUEBOOK** and **HOMEKIDS**, which ranked 9th and 10th in our list of 23 predictor variable importance model **importanceA**. (**Code Appendix 3.2.3**)

At this point, the top 10 most statistically important of our set of 23 predictor variables are included in this model. This time, we can see an even more significant improvement on the **residual deviance** and **AIC** values.

$AIC = 6015$

```
## ROC curve variable importance
##
##      only 10 most important variables shown (out of 23)
##
##              Importance
## CLM_FREQ      0.6354
## OLDCLAIM      0.6339
## MVR_PTS       0.6202
## HOME_VAL      0.6176
## URBANICITY    0.6026
```

```

## INCOME      0.5961
## CAR_USE    0.5782
## MSTATUS     0.5751
## BLUEBOOK   0.5750
## HOMEKIDS   0.5706

```

**Table 4. Feature importance - Binary model 3**

#### Binary Logistic Regression Model 4

Taking what we've learned from models 1-3, we add the variables `CAR_AGE`, `PARENT1` and `EDUCATION`, which ranked 12th, 14th and 17th in our list of 23 predictor variable importance model `importanceA`. (**Code Appendix 3.2.4**)

We also remove the variables `AGE` and `HOMEKIDS`, which from the previous models do not seem to contribute much, i.e., do not seem to be statistically significant for most of the models.

$AIC = 5897.7$

At this point, we can see most significant improvement on the `residual deviance` and `AIC` values.

#### Binary Logistic Regression Model 5

Just out of curiosity, what if we ignored all the statistical correlation and variable importance that we used for the previous four models. We use a model that includes all the predictor variables and the response variable `TARGET_FLAG`. (**Code Appendix 3.2.5**)

The results above show the best improvement so far.

Even after seeing the most significant improvement of all models, we still see that variables `AGE`, `HOMEKIDS`, `SEX`, and `RED_CAR (yes)` are not statistically significant. Which, lead us to believe that it might be true that deeming the variables `RED_CAR` and `SEX` as "urban legends" might be just urban legends. Those variable show little to no correlation to the probability of collision.

The variable `EDUCATION` seems to be statistically significant. At least for the values "Bachelors" and "Masters" we see that, based on the sign of their coefficients, they have a negative correlation to the theoretical probability of collision. So, it appears that people with higher education tend to have fewer accidents.

$AIC = 5903.2$

---

### Linear Regression Models for dependent variable `TARGET_AMT`

#### Linear Regression Model 1

For our first model (**Code Appendix 3.3.1**), we only include the predictor variables that have `theoretical probably of effecting the payout if there is a crash`, which was provided as part of the definition of the variables.

From the summary results we can see that we obtained low values for

Multiple R-squared: 0.01945 and **Adjusted R-squared: 0.01809**

Which, shows that using only predictor variables that have `theoretical probably of effecting the payout if there is a crash` is not a good way to go, for those variables do not seem to be enough to provide statistically significant results.

#### Linear Regression Model 2

For our second model, we only include the top 10 most important predictor variables that we gathered from our variable importance trained model `modelB`. (**Code Appendix 3.3.2**)

```

## glm variable importance
##
##   only 10 most important variables shown (out of 37)
##
##                               Overall
## 'URBANICITYHighly Urban/ Urban' 11.944
## MVR PTS                         6.764
## CAR USEPrivate                   4.741
## 'CAR_TYPESports Car'             4.692
## CAR_TYPESUV                      4.193
## TIF                             3.958
## MSTATUSYes                       3.932
## TRAVTIME                         3.708
## REVOKEDYYes                      3.166
## PARENT1Yes                       2.852

```

**Table 5. Top 10 predictor variables from importance model B**

Below are the results of applying our linear model 2 of TARGET\_AMT vs Top 10 predictor variables.

From the summary results we can see that we obtained much better values for

Multiple R-squared: 0.05666, **Adjusted R-squared: 0.05478**

### Linear Regression Model 3

We begin with a **baseline** model that includes all the predictor variables from Model 2 and the response variable TARGET\_AMT. We remove the variables CLM\_FREQ because its Pr value is 0.157159, which exceeds our requested 0.05 threshold. We will also add the next 6 variables from our variable importance model **modelB**. The added variables are: KIDSDRV, CLM\_FREQ, INCOME, CAR\_AGE, SEX, and BLUEBOOK. (**Code Appendix 3.3.3**)

From the summary results above, we can see that the added variables have helped improve the values of our key indicators

Multiple R-squared: 0.06527, **Adjusted R-squared: 0.06254**

However, now we have to be skeptical about adding too many predictor variables for we do not want to end up with potential multi-collinearity issues.

## Model Selection

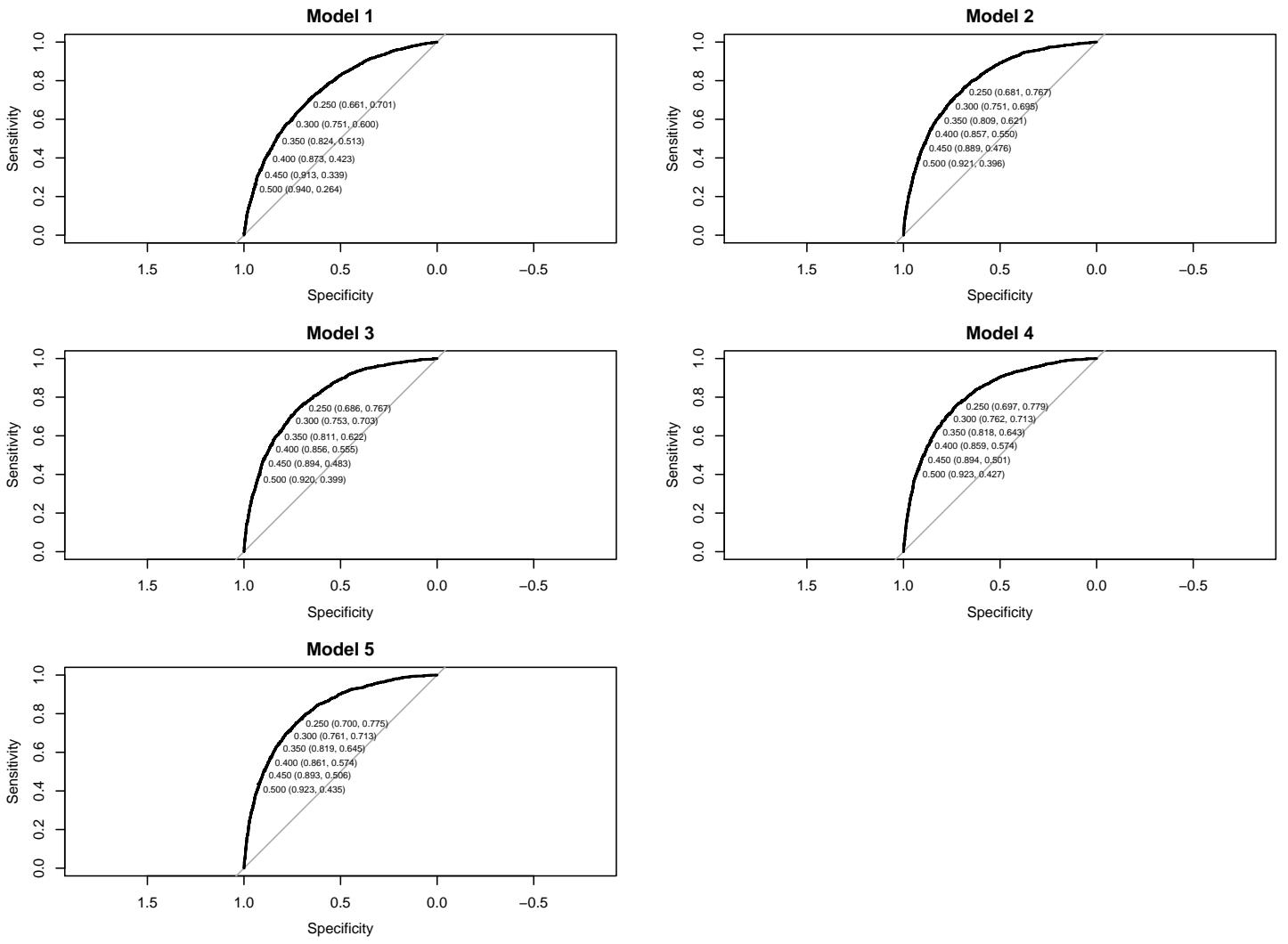
### Binary logistic regression

#### Confusion Matrices

We generate confusion matrices for our five models using a  $p = 0.5$  threshold. (**Code Appendix 4.2.1**)

#### ROC Curves

We generate the ROC curves for all of our models. (**Code Appendix 4.2.2**)



**Figure 9. ROC curves for models 1-5**

The ROC curves in figure 9, show how the different binary models perform. Generally, a lower AIC and a steeper curve are indicators of better models. Model 5, while having a slightly high AIC than model 4, has a much better ROC curve. So we will

likely select this for our binary model.

## Linear regression model option summary of TARGET\_AMT

### Model options summary

Below is a summary of the key indicators for all three models to help us decide which model is the best. Based on the summary, we will select model one, with the lowest  $R^2$  value.

Model	F-statistic	p-value	Adjusted R-squared	Multiple R-squared
Model 1	14.36	< 2.2e-16	0.01809	0.01945
Model 2	30.10	< 2.2e-16	0.05478	0.05666
Model 3	23.92	< 2.2e-16	0.06254	0.06527

Table 6. Summary statistics of linear models

### Residual Plots

We now compare the Residual plots (**Code Appendix 4.3.1**) to help us decide on the selection of the best model. The residual plots in figures 10 through 12, show that the distributions in the models are not looking that normal even after outlier imputation. (**Code Appendix 4.3.2**) Invoking the central limit theorem (CLT) we will do nothing further as we have sufficient data to understand that the variance of the errors in these models is likely finite.

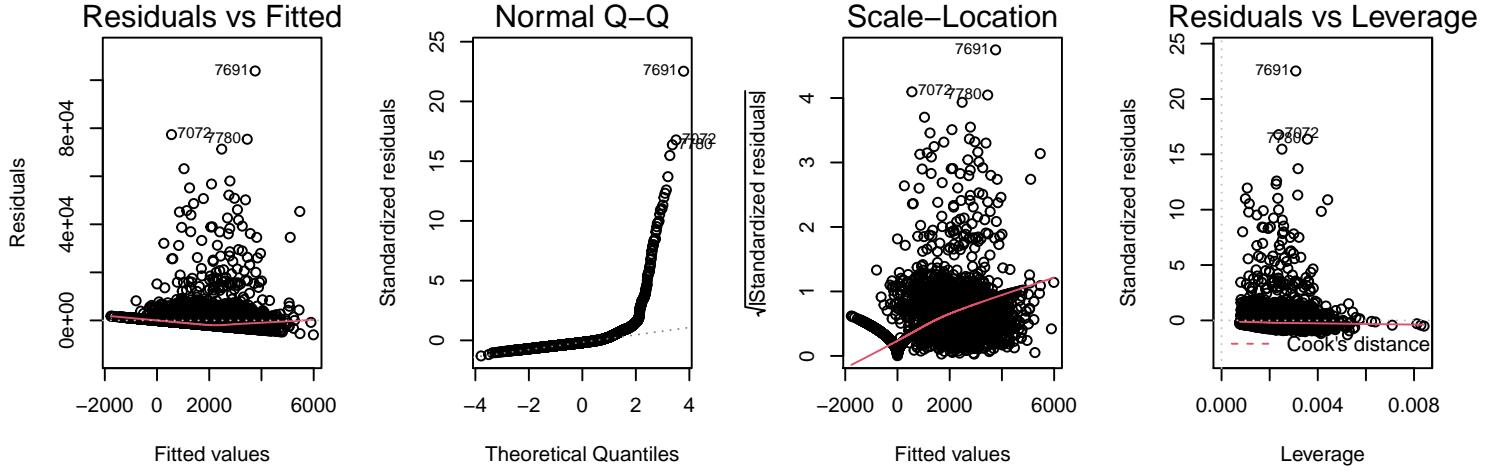
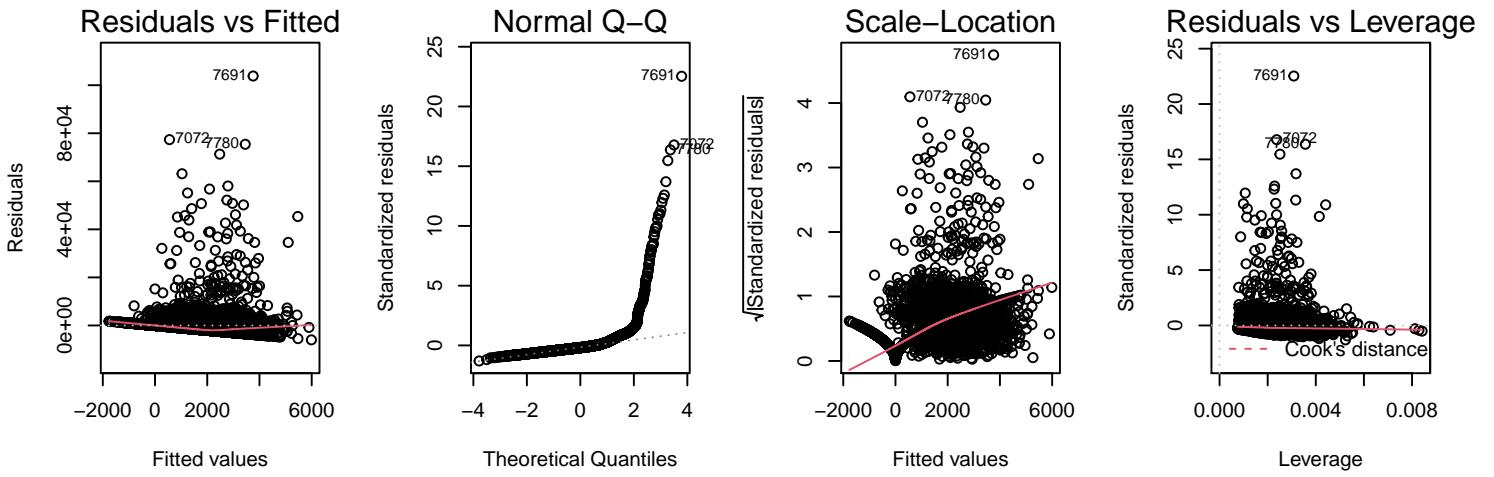


Figure 10. Model 1 plots



**Figure 11. Model 2 plots**

```

## 
## Call:
## lm(formula = TARGET_AMT ~ URBANICITY + MVR PTS + CAR_USE + CAR_TYPE +
##     CAR_TYPE + TIF + MSTATUS + TRAVTIME + REVOKED + PARENT1 +
##     KIDSDRV + CLM_FREQ + INCOME + CAR_AGE + SEX + BLUEBOOK,
##     data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5748 -1683    -800    313 103642
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.247e+02 3.525e+02  0.921 0.357028  
## URBANICITYHighly Urban/ Urban 1.581e+03 1.523e+02 10.382 < 2e-16 ***
## MVR PTS     1.805e+02 2.911e+01  6.199 6.03e-10 ***
## CAR USEPrivate -8.952e+02 1.403e+02 -6.378 1.92e-10 ***
## CAR TYPEPanel Truck -7.406e+01 2.958e+02 -0.250 0.802279  
## CAR TYPEPickup  3.513e+02 1.861e+02  1.888 0.059115 .  
## CAR TYPESports Car  9.632e+02 2.440e+02  3.947 7.99e-05 ***
## CAR TYPESUV    7.759e+02 2.013e+02  3.855 0.000117 *** 
## CAR TYPEVan    5.817e+02 2.346e+02  2.480 0.013173 *  
## TIF          -5.309e+01 1.362e+01 -3.897 9.82e-05 *** 
## MSTATUSYes    -5.762e+02 1.347e+02 -4.276 1.93e-05 *** 
## TRAVTIME      1.210e+01 3.621e+00  3.341 0.000839 *** 
## REVOKEDYes    3.298e+02 1.757e+02  1.877 0.060576 .  
## PARENT1Yes    7.486e+02 1.983e+02  3.775 0.000162 *** 
## KIDSDRV       3.741e+02 1.162e+02  3.219 0.001291 ** 
## CLM_FREQ      7.797e+01 5.511e+01  1.415 0.157159  
## INCOME        -6.445e-03 1.463e-03 -4.405 1.08e-05 *** 
## CAR_AGE       -3.480e+01 1.133e+01 -3.072 0.002135 ** 
## SEXM          3.297e+02 1.798e+02  1.834 0.066718 .  
## BLUEBOOK      1.638e-02 9.656e-03  1.696 0.089849 .  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4597 on 6508 degrees of freedom
## Multiple R-squared:  0.06527, Adjusted R-squared:  0.06254 
## F-statistic: 23.92 on 19 and 6508 DF, p-value: < 2.2e-16

```

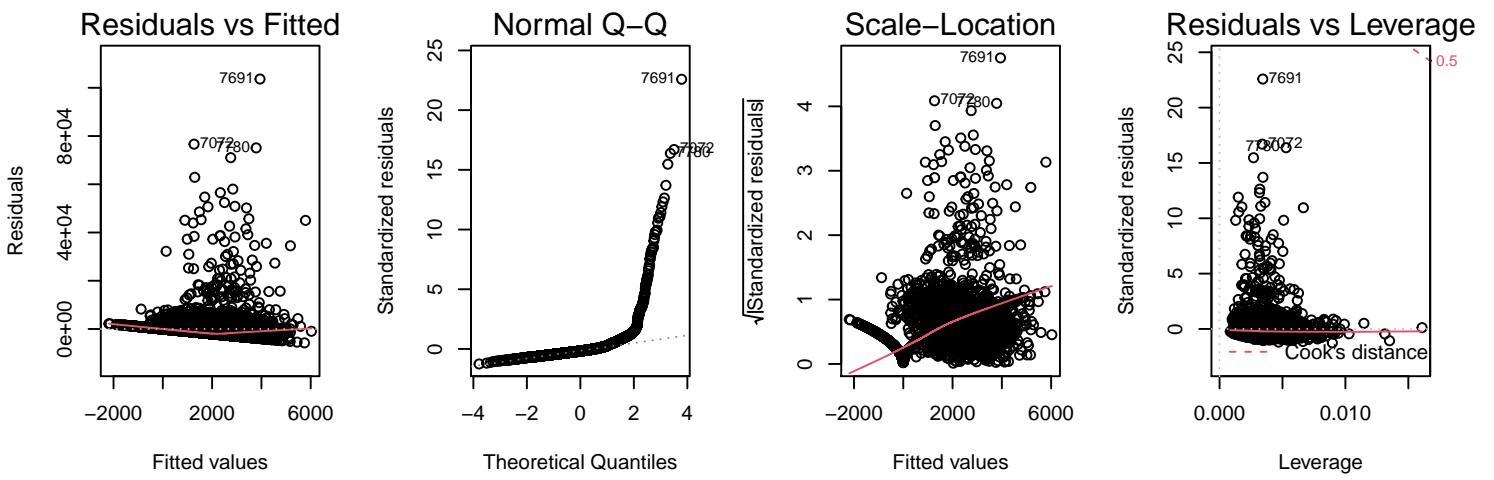


Figure 12. Model 3 plots

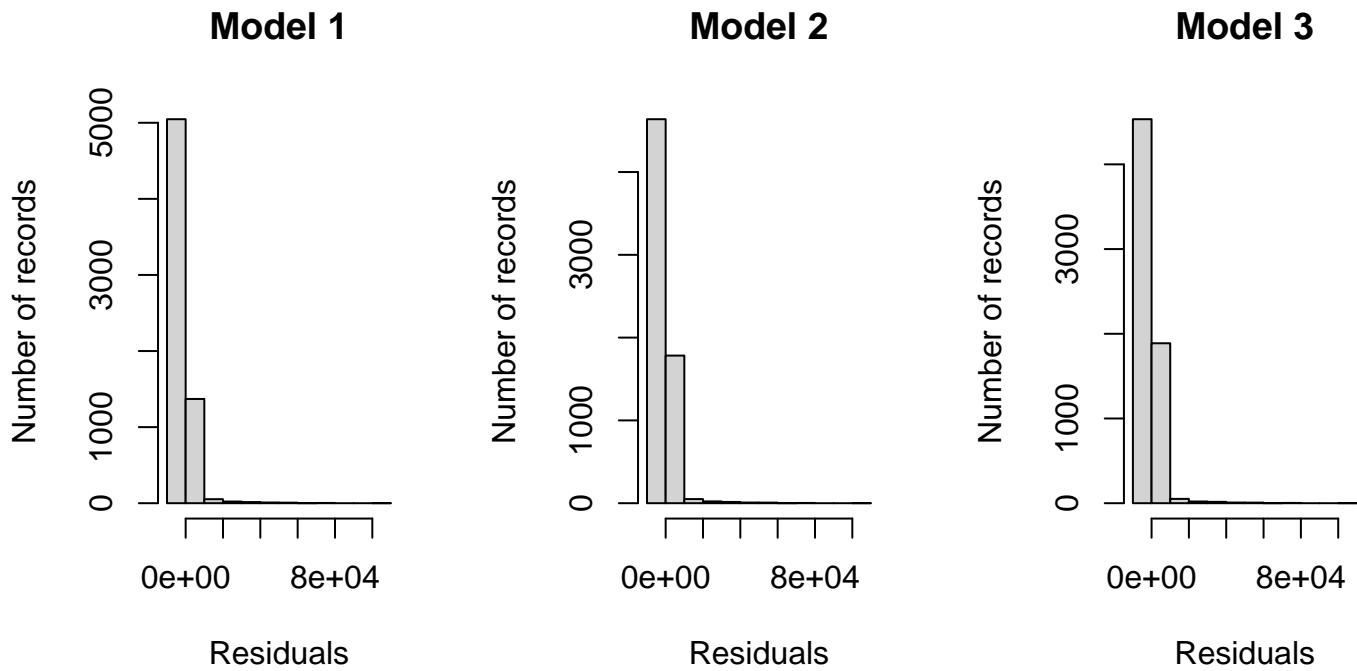


Figure 13. Plots of model residuals

## Conclusions

With our models selected, we conclude this project by writing our predictions to the test data set. ([Code Appendix 5.1](#))

## Code Appendix

### 0.1 Overview

### 0.1.1 Libraries Used

```
library(tidyverse) library(ggplot2) library(VIM) library(GGally) library(caret) library(broom) library(naniar) library(stringr)  
library(pROC) library(ggpubr)
```

## 1.1 Data Exploration

### 1.2 Data Import

```
rawTrain <- read.csv("https://raw.githubusercontent.com/MsQCompSci/Data621Group4/main/HW4/insurance_training_  
data.csv", header = TRUE, stringsAsFactors = FALSE) rawTest <- read.csv("https://raw.githubusercontent.com/  
MsQCompSci/Data621Group4/main/HW4/insurance-evaluation-data.csv")
```

### 1.3 Glimpse on Training Data

```
glimpse(rawTrain)
```

```
## Rows: 8,161  
## Columns: 26  
## $ INDEX      <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 19, 20, 2~  
## $ TARGET_FLAG <fct> 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1~  
## $ TARGET_AMT  <dbl> 0.000, 0.000, 0.000, 0.000, 2946.000, 0.000, 4021.0~  
## $ KIDSDRV     <int> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  
## $ AGE         <int> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 43, 55, 53, 45~  
## $ HOMEKIDS    <int> 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 3, 2, 1~  
## $ YOJ         <int> 11, 11, 10, 14, NA, 12, NA, NA, 10, 7, 14, 5, 11, 11, 0, 1~  
## $ INCOME       <dbl> 67349, 91449, 16039, NA, 114986, 125301, 18755, 107961, 62~  
## $ PARENT1     <fct> No, No, No, No, Yes, No, No, No, No, No, No, No, N~  
## $ HOME_VAL     <dbl> 0, 257252, 124191, 306251, 243925, 0, NA, 333680, 0, 0, 0, ~  
## $ MSTATUS      <fct> No, No, Yes, Yes, Yes, No, Yes, No, No, Yes, Yes, Yes, ~  
## $ SEX          <fct> M, M, F, M, F, F, M, F, F, M, M, F, F, M, F, F, F~  
## $ EDUCATION    <fct> PhD, High School, High School, PhD, Bachelor~  
## $ JOB          <fct> Professional, Blue Collar, Clerical, Blue Collar, Doctor, ~  
## $ TRAVTIME     <int> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36, 25, 64, 48, ~  
## $ CAR_USE      <fct> Private, Commercial, Private, Private, Private, Commercial~  
## $ BLUEBOOK     <dbl> 14230, 14940, 4010, 15440, 18000, 17430, 8780, 16970, 1120~  
## $ TIF          <int> 11, 1, 4, 7, 1, 1, 1, 1, 7, 1, 7, 6, 1, 6, 6, 7, 4, ~  
## $ CAR_TYPE     <fct> Minivan, Minivan, SUV, Minivan, SUV, Sports Car, SUV, Van, ~  
## $ RED_CAR      <fct> yes, yes, no, yes, no, no, yes, no, no, no, yes, y~  
## $ OLDCLAIM     <dbl> 4461, 0, 38690, 0, 19217, 0, 0, 2374, 0, 0, 0, 0, 5028, 0, ~  
## $ CLM_FREQ     <int> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2~  
## $ REVOKED      <fct> No, No, No, Yes, No, No, Yes, No, No, No, Yes, No, ~  
## $ MVR_PTS      <int> 3, 0, 3, 0, 3, 0, 0, 10, 0, 1, 0, 0, 3, 3, 3, 0, 0, 0, 0, ~  
## $ CAR_AGE      <int> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 10, 5, 13, 16, ~  
## $ URBANICITY   <fct> Highly Urban/ Urban, Highly Urban/ Urban, Highly Urban/ Ur~
```

### 1.4 Figure 1. Plot of missing values

```
gg_miss_var(rawTrain)
```

### 1.5 Data Cleaning - Converting Data Types

```
rawTrain <- rawTrain %>% mutate(INCOME = gsub("\$", "", INCOME), #Remove $ HOME_VAL = gsub("\\", "", HOMEVAL), BL  
gsub("
```

```

,"", BLUEBOOK), OLDCLAIM = gsub("\$","", OLDCLAIM), MSTATUS = gsub("z_","", MSTATUS), SEX = gsub("z_","", SEX), EDUCATION= gsub("z_","", EDUCATION), JOB= gsub("z_","", JOB), CAR_TYPE= gsub("z_","", CAR_TYPE), URBANICITY= gsub("z_","", URBANICITY), INCOME = as.numeric(gsub(",","", INCOME)), #remove , and cast to numeric HOME_VAL = as.numeric(gsub(",","", HOME_VAL)), BLUEBOOK = as.numeric(gsub(",","", BLUEBOOK)), OLDCLAIM = as.numeric(gsub(",","", OLDCLAIM)), TARGET_FLAG = as.factor(TARGET_FLAG))

rawTrain[sapply(rawTrain, is.character)] <- lapply(rawTrain[sapply(rawTrain, is.character)], as.factor)

rawTest <- rawTest %>% mutate(INCOME = gsub("\$","", INCOME), #Remove $ HOME_VAL = gsub("\$","", HOMEVAL), BLUEBOOK = gsub("\$","", BLUEBOOK), OLDCLAIM = gsub("\$","", OLDCLAIM), MSTATUS = gsub("z_","", MSTATUS), SEX = gsub("z_","", SEX), EDUCATION= gsub("z_","", EDUCATION), JOB= gsub("z_","", JOB), CAR_TYPE= gsub("z_","", CAR_TYPE), URBANICITY= gsub("z_","", URBANICITY), INCOME = as.numeric(gsub(",","", INCOME)), #remove , and cast to numeric HOME_VAL = as.numeric(gsub(",","", HOME_VAL)), BLUEBOOK = as.numeric(gsub(",","", BLUEBOOK)), OLDCLAIM = as.numeric(gsub(",","", OLDCLAIM)), TARGET_FLAG = as.factor(TARGET_FLAG), TARGET_AMT = as.numeric(TARGET_AMT))

rawTest[sapply(rawTest, is.character)] <- lapply(rawTest[sapply(rawTest, is.character)], as.factor)

```

## 1.6 Confirmation glimpse

```
glimpse(rawTrain)
```

```

## Rows: 8,161
## Columns: 26
## $ INDEX      <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 19, 20, 2~  

## $ TARGET_FLAG <fct> 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1~  

## $ TARGET_AMT <dbl> 0.000, 0.000, 0.000, 0.000, 2946.000, 0.000, 4021.0~  

## $ KIDSDRV    <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~  

## $ AGE        <int> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 43, 55, 53, 45~  

## $ HOMEKIDS   <int> 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 3, 2, 1~  

## $ YOJ        <int> 11, 11, 10, 14, NA, 12, NA, NA, 10, 7, 14, 5, 11, 11, 0, 1~  

## $ INCOME     <dbl> 67349, 91449, 16039, NA, 114986, 125301, 18755, 107961, 62~  

## $ PARENT1    <fct> No, No, No, No, Yes, No, No, No, No, No, No, No, N~  

## $ HOME_VAL   <dbl> 0, 257252, 124191, 306251, 243925, 0, NA, 333680, 0, 0, 0~  

## $ MSTATUS    <fct> No, No, Yes, Yes, No, Yes, Yes, No, No, No, Yes, Yes, ~  

## $ SEX         <fct> M, M, F, M, F, F, M, F, F, M, M, F, F, M, F, F, F~  

## $ EDUCATION   <fct> PhD, High School, High School, <High School, PhD, Bachelor~  

## $ JOB         <fct> Professional, Blue Collar, Clerical, Blue Collar, Doctor, ~  

## $ TRAVTIME   <int> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36, 25, 64, 48, ~  

## $ CAR_USE     <fct> Private, Commercial, Private, Private, Private, Commercial~  

## $ BLUEBOOK    <dbl> 14230, 14940, 4010, 15440, 18000, 17430, 8780, 16970, 1120~  

## $ TIF         <int> 11, 1, 4, 7, 1, 1, 1, 7, 1, 7, 7, 6, 1, 6, 6, 7, 4, ~  

## $ CAR_TYPE    <fct> Minivan, Minivan, SUV, Minivan, SUV, Sports Car, SUV, Van, ~  

## $ RED_CAR     <fct> yes, yes, no, yes, no, no, yes, no, no, no, yes, y~  

## $ OLDCLAIM    <dbl> 4461, 0, 38690, 0, 19217, 0, 0, 2374, 0, 0, 0, 0, 5028, 0, ~  

## $ CLM_FREQ    <int> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2~  

## $ REVOKED    <fct> No, No, No, Yes, No, No, Yes, No, No, No, Yes, No, ~  

## $ MVR_PTS    <int> 3, 0, 3, 0, 3, 0, 10, 0, 1, 0, 0, 3, 3, 3, 0, 0, 0, 0, ~  

## $ CAR_AGE    <int> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 10, 5, 13, 16, ~  

## $ URBANICITY <fct> Highly Urban/ Urban, Highly Urban/ Urban, Highly Urban/ Ur~
```

## 1.7 Summary Statistics

```
summary(rawTrain)
```

##	INDEX	TARGET_FLAG	TARGET_AMT	KIDSDRV	AGE
----	-------	-------------	------------	---------	-----

```

## Min. : 1 0:6008   Min. : 0 Min. :0.0000 Min. :16.00
## 1st Qu.: 2559 1:2153   1st Qu.: 0 1st Qu.:0.0000 1st Qu.:39.00
## Median : 5133           Median : 0 Median :0.0000 Median :45.00
## Mean   : 5152           Mean   : 1504 Mean   :0.1711 Mean   :44.79
## 3rd Qu.: 7745           3rd Qu.: 1036 3rd Qu.:0.0000 3rd Qu.:51.00
## Max.   :10302          Max.   :107586 Max.   :4.0000 Max.   :81.00
##
##      HOMEKIDS        YOJ        INCOME      PARENT1      HOME_VAL
## Min. :0.0000  Min. : 0.0  Min. : 0  No :7084  Min. :    0
## 1st Qu.:0.0000 1st Qu.: 9.0  1st Qu.: 28097 Yes:1077 1st Qu.:    0
## Median :0.0000 Median :11.0  Median : 54028           Median :161160
## Mean   :0.7212  Mean   :10.5  Mean   : 61898           Mean   :154867
## 3rd Qu.:1.0000 3rd Qu.:13.0  3rd Qu.: 85986           3rd Qu.:238724
## Max.   :5.0000  Max.   :23.0  Max.   :367030          Max.   :885282
## NA's   :454     NA's   :445  NA's   :464
##
##      MSTATUS       SEX        EDUCATION      JOB        TRAVTIME
## No :3267  F:4375  <High School:1203  Blue Collar :1825  Min. : 5.00
## Yes:4894 M:3786  Bachelors   :2242  Clerical    :1271  1st Qu.: 22.00
##                   High School :2330  Professional:1117  Median : 33.00
##                   Masters    :1658  Manager     : 988  Mean   : 33.49
##                   PhD       : 728  Lawyer      : 835  3rd Qu.: 44.00
##                   Student   : 712  Student    : 712  Max.   :142.00
##                   (Other)   :1413
##
##      CAR_USE        BLUEBOOK      TIF        CAR_TYPE
## Commercial:3029  Min. : 1500  Min. : 1.000  Minivan   :2145
## Private   :5132   1st Qu.: 9280  1st Qu.: 1.000  Panel Truck: 676
##                   Median :144440  Median : 4.000  Pickup    :1389
##                   Mean   :15710  Mean   : 5.351  Sports Car : 907
##                   3rd Qu.:20850  3rd Qu.: 7.000  SUV       :2294
##                   Max.   :69740  Max.   :25.000  Van      : 750
##
##      RED_CAR        OLDCLAIM      CLM_FREQ      REVOKED      MVR_PTS
## no :5783     Min. : 0  Min. :0.0000  No :7161  Min. : 0.000
## yes:2378    1st Qu.: 0  1st Qu.:0.0000 Yes:1000  1st Qu.: 0.000
##                   Median : 0  Median :0.0000           Median : 1.000
##                   Mean   : 4037  Mean   :0.7986  Mean   : 1.696
##                   3rd Qu.: 4636  3rd Qu.:2.0000  3rd Qu.: 3.000
##                   Max.   :57037  Max.   :5.0000  Max.   :13.000
##
##      CAR_AGE        URBANICITY
## Min. :-3.000  Highly Rural/ Rural:1669
## 1st Qu.: 1.000  Highly Urban/ Urban:6492
## Median : 8.000
## Mean   : 8.328
## 3rd Qu.:12.000
## Max.   :28.000
## NA's   :510

```

## 1.8 Figure 2. Feature histograms

```
histData <- rawTrain %>% select(AGE, HOMEKIDS, YOJ, TRAVTIME, TIF, CLM_FREQ, MVR PTS, CAR_AGE, INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM)
```

```
par(mfrow = c(3,4)) for(i in 1:ncol(histData)) {#distribution of each variable hist(histData[[i]], main = colnames(histData[i]), col = "red")}
```

## 1.9 Outlier Analysis

### 1.9.1 Figure 3. Feature box plot, part 1

```
longData <- histData %>% select(-HOME_VAL, -INCOME, -BLUEBOOK, -OLDCLAIM) %>% # remove this for scale issue  
will plot below gather(key = Variable, value = Value)  
  
ggplot(longData, aes(Variable, Value, fill = Variable)) +geom_boxplot(outlier.colour="blue", outlier.shape=4, outlier.size=2,  
show.legend=FALSE) + theme(axis.text.x = element_text(angle = 90, hjust = 1)) + coord_flip() + labs(title="Insurance Data  
Variables - PART 1", y="Value")
```

### 1.9.2 Figure 4. Feature box plot, part 2

```
longData2 <- histData %>% select(HOME_VAL, INCOME, BLUEBOOK, OLDCLAIM) %>% # remove this for scale issue  
will plot below gather(key = Variable, value = Value)  
  
ggplot(longData2, aes(Variable, Value, fill = Variable)) +geom_boxplot(outlier.colour="blue", outlier.shape=4, outlier.size=2,  
show.legend=FALSE) + theme(axis.text.x = element_text(angle = 90, hjust = 1)) + coord_flip() + labs(title="Insurance Data  
Variables - PART 2", y="Value")
```

## 1.10 Figure 5. Categorical variables showing balance/imbalance

```
p1 <- ggplot(rawTrain, aes(x = PARENT1, fill = TARGET_FLAG)) + geom_bar() + labs(title="Insurance Data Categorical  
Variables - Single Parent (Parent 1)") p2 <- ggplot(rawTrain, aes(x = MSTATUS, fill = TARGET_FLAG)) + geom_bar()  
+ labs(title="Insurance Data Categorical Variables - Marital Status") p3 <- ggplot(rawTrain, aes(x = SEX, fill = TAR-  
GET_FLAG)) + geom_bar() + labs(title="Insurance Data Categorical Variables - SEX") p4 <- ggplot(rawTrain, aes(x =  
EDUCATION, fill = TARGET_FLAG)) + geom_bar() + labs(title="Insurance Data Categorical Variables - Max Education  
Level") p5 <- ggplot(rawTrain, aes(x = JOB, fill = TARGET_FLAG)) + geom_bar() + labs(title="Insurance Data Cate-  
gorical Variables - Job Category") p6 <- ggplot(rawTrain, aes(x = CAR_USE, fill = TARGET_FLAG)) + geom_bar()  
+ labs(title="Insurance Data Categorical Variables - Vehicle Use") p7 <- ggplot(rawTrain, aes(x = CAR_TYPE, fill = TAR-  
GET_FLAG)) + geom_bar() + labs(title="Insurance Data Categorical Variables - Car Type") p8 <- ggplot(rawTrain, aes(x =  
RED_CAR, fill = TARGET_FLAG)) + geom_bar() + labs(title="Insurance Data Categorical Variables - Red Car") p9  
<- ggplot(rawTrain, aes(x = REVOKED, fill = TARGET_FLAG)) + geom_bar() + labs(title="Insurance Data Categorical  
Variables - Licensed Revoked (Past 7 Years)") ggarrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, ncol = 2, nrow = 5)
```

## 1.11 Numeric Data - Relationship to Target / Figure 6. Feature plots

```
par(mfrow = c(5,3))  
  
histData <- rawTrain %>% select(TARGET_AMT, AGE, HOMEKIDS, YOJ, TRAVTIME, TIF, CLM_FREQ, MVR PTS,  
CAR AGE, INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM)  
  
featurePlot(x= histData[3:6], y = histData[['TARGET_AMT']]) featurePlot(x= histData[7:10], y = histData[['TARGET_AMT']])  
featurePlot(x= histData[11:13], y = histData[['TARGET_AMT']])
```

## 1.12 Correlation

### 1.12.1 Figure 7. Correlation matrix

```
ggcorr(rawTrain)
```

### 1.12.2 findCorrelation() function

```
findCorrelation(cor(histData), cutoff = 0.75, verbose = TRUE, names = TRUE)
```

## 2.1 Data Preparation

### 2.2 Impute Missing Values

```
#due to skew home_val, income will be imputed with median
#Age YOJ with the mean

#new DF
prepTrain <- rawTrain %>%
  select(-INDEX)

#impute NAs
prepTrain$AGE[is.na(prepTrain$AGE)] <- mean(prepTrain$AGE, na.rm=TRUE)
prepTrain$YOJ[is.na(prepTrain$YOJ)] <- mean(prepTrain$YOJ, na.rm=TRUE)
prepTrain$HOME_VAL[is.na(prepTrain$HOME_VAL)] <- median(prepTrain$HOME_VAL, na.rm=TRUE)
prepTrain$INCOME[is.na(prepTrain$INCOME)] <- median(prepTrain$INCOME, na.rm=TRUE)
prepTrain$CAR_AGE[is.na(prepTrain$CAR_AGE)] <- mean(prepTrain$CAR_AGE, na.rm=TRUE)
```

### 2.3 Outlier detection

```
outlier_norm <- function(x){ if (class(x) != "factor"){ qntile <- quantile(x, probs=c(.25, .75)) caps <- quantile(x, probs=c(.05,
.95)) H <- 1.5 * IQR(x, na.rm = T) x[x < (qntile[1] - H)] <- caps[1] x[x > (qntile[2] + H)] <- caps[2] return(x) } }
sapply(prepTrain, outlier_norm)
```

### 2.4 Variable Importance - Part A

```
prepTrainA <- prepTrain %>% select(-TARGET_AMT) prepTrainB <- prepTrain %>% select(-TARGET_FLAG) control
<- trainControl(method="repeatedcv", number=10, repeats=3) modelA <- train(TARGET_FLAG~., data=prepTrainA,
method="lvq", preProcess="scale", trControl=control) saveRDS(modelA, "./varimportance_modelA.rds")
```

#### 2.4.1 Table 1. Feature importance - part A

```
modelA = readRDS("./varimportance_modelA.rds") importanceA <- varImp(modelA, scale=FALSE) print(importanceA, 23)
```

#### 2.4.2 Figure 8. Plot of feature importance - part A

```
plot(importanceA)
```

### 2.4 Variable Importance - Part B

```
control <- trainControl(method="repeatedcv", number=10, repeats=3) modelB <- train(TARGET_AMT~, data=prepTrainB,
method="glm", preProcess="scale", trControl=control) saveRDS(modelB, "./varimportance_modelB.rds")
```

#### 2.5.1 Table 2. Feature importance - part B

```
modelB = readRDS("./varimportance_modelB.rds") importanceB <- varImp(modelB, scale=FALSE) print(importanceB, 23)
```

#### 2.5.2 Figure 9. Plot of feature importance - part B

```
plot(importanceB)
```

## 2.6 Train Test Split

```
set.seed(123) trainIndex<- sort(sample(nrow(prepTrain), nrow(prepTrain)*.8))
train <- prepTrain[trainIndex, ] test <- prepTrain[-trainIndex, ]
```

## 3.1 Model Building

### 3.2 Binary Logistic Regression for dependent variable TARGET\_FLAG

#### 3.2.1 Binary Logistic Regression Model 1

```
#(logistic regression)
logRegModel1 <- glm(formula = TARGET_FLAG ~ AGE +
CAR_USE +
CLM_FREQ +
HOME_VAL +
INCOME +
JOB +
KIDSDRIV +
MSTATUS +
MVR_PTS +
REVOKE +
TIF +
TRAVTIME +
YOJ,
data=train, family = "binomial" (link="logit"))
summary(logRegModel1)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + HOME_VAL +
##      INCOME + JOB + KIDSDRIV + MSTATUS + MVR_PTS + REVOKE + TIF +
##      TRAVTIME + YOJ, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.0607 -0.7577 -0.5370  0.8177  2.8031
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.265e-02 2.593e-01 -0.242 0.809111
## AGE         -1.202e-02 3.727e-03 -3.225 0.001258 **
## CAR_USEPrivate -5.537e-01 7.862e-02 -7.043 1.88e-12 ***
## CLM_FREQ      2.581e-01 2.677e-02  9.641 < 2e-16 ***
## HOME_VAL      -1.327e-06 3.610e-07 -3.675 0.000238 ***
## INCOME        -4.107e-06 1.080e-06 -3.804 0.000143 ***
## JOBBlue Collar 2.759e-01 1.452e-01  1.900 0.057447 .
## JOBClerical   1.945e-01 1.690e-01  1.151 0.249816
## JOBDoctor     -2.060e-01 2.650e-01 -0.777 0.436982
## JOBHome Maker  4.789e-02 2.004e-01  0.239 0.811094
## JOBLawyer     1.522e-02 1.784e-01  0.085 0.932011
## JOBManager    -4.282e-01 1.715e-01 -2.496 0.012546 *
## JOBProfessional 3.635e-02 1.598e-01  0.227 0.820111
## JOBStudent    2.181e-02 1.899e-01  0.115 0.908548
## KIDSDRIV      3.511e-01 5.551e-02  6.324 2.54e-10 ***
## MSTATUSYes     -5.051e-01 7.465e-02 -6.765 1.33e-11 ***
```

```

## MVR PTS      1.374e-01  1.435e-02  9.579 < 2e-16 ***
## REVOKEDYes   7.957e-01  8.529e-02  9.329 < 2e-16 ***
## TIF          -5.016e-02  7.687e-03 -6.525 6.78e-11 ***
## TRAVTIME     6.508e-03  1.903e-03  3.420 0.000625 ***
## YOJ          -9.587e-03  8.753e-03 -1.095 0.273386
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 7533.1 on 6527 degrees of freedom
## Residual deviance: 6528.6 on 6507 degrees of freedom
## AIC: 6570.6
##
## Number of Fisher Scoring iterations: 4

```

### 3.2.2 Binary Logistic Regression Model 2

```

#(logistic regression)
logRegModel2 <- glm(formula = TARGET_FLAG ~ AGE +
  CAR_USE +
  CLM_FREQ +
  HOME_VAL +
  INCOME +
  JOB +
  KIDSDRV +
  MSTATUS +
  MVR PTS +
  REVOKED +
  TIF +
  TRAVTIME +
  OLDCLAIM +
  URBANICITY,
  data=train, family = "binomial" (link="logit"))
summary(logRegModel2)

```

```

##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + HOME_VAL +
##       INCOME + JOB + KIDSDRV + MSTATUS + MVR PTS + REVOKED + TIF +
##       TRAVTIME + OLDCLAIM + URBANICITY, family = binomial(link = "logit"),
##       data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q      Max
## -2.4277 -0.7374 -0.4179  0.7057  2.9863
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.375e+00  2.857e-01 -8.313 < 2e-16 ***
## AGE                      -1.132e-02  3.890e-03 -2.911 0.003607 **
## CAR_USEPrivate            -6.694e-01  8.269e-02 -8.095 5.70e-16 ***
## CLM_FREQ                  1.795e-01  3.138e-02  5.721 1.06e-08 ***
## HOME_VAL                 -1.341e-06  3.724e-07 -3.602 0.000316 ***
## INCOME                   -4.800e-06  1.107e-06 -4.338 1.44e-05 ***
## JOBBlue Collar           5.527e-01  1.475e-01  3.748 0.000178 ***
## JOBClerical               6.551e-01  1.737e-01  3.771 0.000162 ***
## JOBDoctor                -1.763e-01  2.650e-01 -0.665 0.506038

```

```

## JOBHome Maker      5.915e-01  2.032e-01  2.910 0.003610 **
## JOBLawyer         9.474e-02  1.805e-01  0.525 0.599711
## JOBManager        -4.475e-01  1.721e-01  -2.601 0.009296 **
## JOBProfessional   2.014e-01  1.621e-01  1.242 0.214274
## JOBStudent        5.903e-01  1.941e-01  3.041 0.002357 **
## KIDSDRV           4.586e-01  5.971e-02  7.681 1.57e-14 ***
## MSTATUSYes        -6.311e-01  7.810e-02  -8.080 6.48e-16 ***
## MVR_PTS            1.194e-01  1.496e-02  7.976 1.51e-15 ***
## REVOKEDYes        8.145e-01  1.006e-01  8.096 5.66e-16 ***
## TIF                -5.477e-02  7.983e-03  -6.862 6.80e-12 ***
## TRAVTIME           1.408e-02  2.061e-03  6.833 8.31e-12 ***
## OLDCLAIM           -1.145e-05  4.332e-06  -2.643 0.008227 **
## URBANICITYHighly Urban/ Urban  2.329e+00  1.237e-01  18.819 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 7533.1 on 6527 degrees of freedom
## Residual deviance: 6009.2 on 6506 degrees of freedom
## AIC: 6053.2
##
## Number of Fisher Scoring iterations: 5

```

### 3.2.3 Binary Logistic Regression Model 3

```

#(logistic regression)
logRegModel3 <- glm(formula = TARGET_FLAG ~ AGE +
CAR_USE +
CLM_FREQ +
HOME_VAL +
INCOME +
JOB +
KIDSDRV +
MSTATUS +
MVR_PTS +
REVOKED +
TIF +
TRAVTIME +
OLDCLAIM +
URBANICITY +
BLUEBOOK +
HOMEKIDS,
data=train, family = "binomial" (link="logit"))
summary(logRegModel3)

##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + HOME_VAL +
##     INCOME + JOB + KIDSDRV + MSTATUS + MVR_PTS + REVOKED + TIF +
##     TRAVTIME + OLDCLAIM + URBANICITY + BLUEBOOK + HOMEKIDS, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q      Max
## -2.4244 -0.7320 -0.4164  0.6821  2.9748
##
## Coefficients:

```

```

##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -2.258e+00 3.086e-01 -7.315 2.58e-13 ***
## AGE                          -4.528e-03 4.333e-03 -1.045 0.295999
## CAR_USEPrivate              -7.865e-01 8.546e-02 -9.204 < 2e-16 ***
## CLM_FREQ                      1.739e-01 3.151e-02  5.520 3.39e-08 ***
## HOME_VAL                      -1.337e-06 3.743e-07 -3.571 0.000356 ***
## INCOME                        -3.486e-06 1.133e-06 -3.076 0.002101 **
## JOBBlue Collar                4.131e-01 1.492e-01  2.769 0.005626 **
## JOBCLerical                   5.702e-01 1.746e-01  3.265 0.001093 **
## JOBDoctor                      -2.424e-01 2.667e-01 -0.909 0.363280
## JOBHome Maker                  5.065e-01 2.042e-01  2.481 0.013105 *
## JOBLawyer                      3.755e-02 1.811e-01  0.207 0.835692
## JOBManager                     -4.971e-01 1.727e-01 -2.879 0.003992 **
## JOBProfessional                 1.571e-01 1.628e-01  0.965 0.334521
## JOBStudent                     4.097e-01 1.966e-01  2.084 0.037134 *
## KIDSDRV                         3.808e-01 6.792e-02  5.607 2.06e-08 ***
## MSTATUSYes                      -6.547e-01 7.869e-02 -8.320 < 2e-16 ***
## MVR PTS                         1.184e-01 1.502e-02  7.880 3.28e-15 ***
## REVOKEDYes                      8.031e-01 1.011e-01  7.942 1.99e-15 ***
## TIF                            -5.633e-02 8.020e-03 -7.024 2.15e-12 ***
## TRAVTIME                        1.439e-02 2.069e-03  6.953 3.57e-12 ***
## OLDCLAIM                        -1.120e-05 4.345e-06 -2.578 0.009938 **
## URBANICITYHighly Urban/ Urban   2.358e+00 1.243e-01 18.965 < 2e-16 ***
## BLUEBOOK                        -2.650e-05 4.511e-06 -5.875 4.22e-09 ***
## HOMEKIDS                        9.684e-02 3.580e-02  2.705 0.006830 **
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 7533.1 on 6527 degrees of freedom
## Residual deviance: 5967.0 on 6504 degrees of freedom
## AIC: 6015
##
## Number of Fisher Scoring iterations: 5

```

### 3.2.4 Binary Logistic Regression Model 4

```

#(logistic regression)
logRegModel4 <- glm(formula = TARGET_FLAG ~ CAR_USE +
CLM_FREQ +
HOME_VAL +
INCOME +
JOB +
KIDSDRV +
MSTATUS +
MVR PTS +
REVOKED +
TIF +
TRAVTIME +
OLDCLAIM +
URBANICITY +
BLUEBOOK +
CAR_AGE +
CAR_TYPE +
PARENT1 +
EDUCATION,
data=train, family = "binomial" (link="logit"))
summary(logRegModel4)

```

```

## 
## Call:
## glm(formula = TARGET_FLAG ~ CAR_USE + CLM_FREQ + HOME_VAL + INCOME +
##      JOB + KIDSDRIV + MSTATUS + MVR_PTS + REVOKED + TIF + TRAVTIME +
##      OLDCLAIM + URBANICITY + BLUEBOOK + CAR_AGE + CAR_TYPE + PARENT1 +
##      EDUCATION, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##    Min      1Q   Median      3Q     Max
## -2.6408 -0.7109 -0.3978  0.6333  3.1562
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -3.027e+00  3.095e-01 -9.780 < 2e-16 ***
## CAR_USEPrivate             -7.907e-01  1.023e-01 -7.733 1.05e-14 ***
## CLM_FREQ                   1.722e-01  3.194e-02  5.393 6.94e-08 ***
## HOME_VAL                   -1.425e-06 3.780e-07 -3.769 0.000164 ***
## INCOME                      -2.484e-06 1.189e-06 -2.088 0.036772 *
## JOBBlue Collar              3.677e-01 2.080e-01  1.767 0.077146 .
## JOBClerical                 4.621e-01 2.200e-01  2.100 0.035696 *
## JOBDoctor                  -2.734e-01 2.895e-01 -0.944 0.344937
## JOBHome Maker               3.776e-01 2.293e-01  1.647 0.099562 .
## JOBLawyer                   1.694e-01 1.913e-01  0.886 0.375636
## JOBManager                 -4.761e-01 1.925e-01 -2.473 0.013383 *
## JOBProfessional              2.565e-01 2.001e-01  1.282 0.199867
## JOBStudent                  3.529e-01 2.366e-01  1.491 0.135860
## KIDSDRIV                     4.086e-01 6.263e-02  6.524 6.84e-11 ***
## MSTATUSYes                  -4.904e-01 8.931e-02 -5.491 4.00e-08 ***
## MVR_PTS                      1.135e-01 1.520e-02  7.470 8.05e-14 ***
## REVOKEDYes                  7.962e-01 1.025e-01  7.769 7.92e-15 ***
## TIF                           -5.623e-02 8.136e-03 -6.912 4.79e-12 ***
## TRAVTIME                     1.484e-02 2.102e-03  7.061 1.66e-12 ***
## OLDCLAIM                      -1.135e-05 4.391e-06 -2.586 0.009714 **
## URBANICITYHighly Urban/ Urban 2.449e+00 1.263e-01 19.386 < 2e-16 ***
## BLUEBOOK                      -2.274e-05 5.312e-06 -4.282 1.85e-05 ***
## CAR_AGE                       -3.642e-03 8.405e-03 -0.433 0.664791
## CAR_TYPEPanel Truck            5.734e-01 1.706e-01  3.360 0.000779 ***
## CAR_TYPEPickup                 5.400e-01 1.124e-01  4.805 1.55e-06 ***
## CAR_TYPESports Car             1.026e+00 1.194e-01  8.593 < 2e-16 ***
## CAR_TYPESUV                    7.482e-01 9.578e-02  7.812 5.63e-15 ***
## CAR_TYPEVan                   7.281e-01 1.349e-01  5.399 6.71e-08 ***
## PARENT1Yes                   5.309e-01 1.049e-01  5.060 4.19e-07 ***
## EDUCATIONBachelors            -4.417e-01 1.293e-01 -3.416 0.000636 ***
## EDUCATIONHigh School           -5.730e-02 1.069e-01 -0.536 0.591870
## EDUCATIONMasters              -3.846e-01 2.007e-01 -1.916 0.055385 .
## EDUCATIONPhD                  -2.537e-01 2.364e-01 -1.073 0.283095
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 7533.1 on 6527 degrees of freedom
## Residual deviance: 5831.7 on 6495 degrees of freedom
## AIC: 5897.7
##
## Number of Fisher Scoring iterations: 5

```

### 3.2.5 Binary Logistic Regression Model 5

```
#Baseline (logistic regression)
logRegModel5 <- glm(formula = TARGET_FLAG ~ . - TARGET_AMT, data=train, family = "binomial" (link="logit"))
summary(logRegModel5)

##
## Call:
## glm(formula = TARGET_FLAG ~ . - TARGET_AMT, family = binomial(link = "logit"),
##      data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max 
## -2.6207 -0.7138 -0.3982  0.6320  3.1760 
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)           -2.794e+00  3.811e-01 -7.331 2.29e-13 ***
## KIDSDRV              3.954e-01  6.933e-02  5.703 1.18e-08 ***
## AGE                  -3.360e-03  4.509e-03 -0.745 0.456212    
## HOMEKIDS             2.628e-02  4.177e-02  0.629 0.529287    
## YOJ                  -1.639e-02  9.646e-03 -1.699 0.089301 .  
## INCOME               -2.356e-06  1.194e-06 -1.972 0.048596 *  
## PARENT1Yes            4.746e-01  1.226e-01  3.871 0.000108 *** 
## HOME_VAL              -1.381e-06  3.795e-07 -3.640 0.000273 *** 
## MSTATUSYes             4.922e-01  9.386e-02 -5.244 1.57e-07 *** 
## SEXM                 6.883e-02  1.256e-01  0.548 0.583642    
## EDUCATIONBachelors   -4.420e-01  1.295e-01 -3.413 0.000643 *** 
## EDUCATIONHigh School  -5.567e-02  1.070e-01 -0.520 0.602836    
## EDUCATIONMasters      -3.802e-01  2.010e-01 -1.891 0.058579 .  
## EDUCATIONPhD          -2.484e-01  2.370e-01 -1.048 0.294649    
## JOBBlue Collar        3.697e-01  2.081e-01  1.777 0.075644 .  
## JOBClerical            4.590e-01  2.202e-01  2.085 0.037058 *  
## JOBDoctor              -2.672e-01  2.901e-01 -0.921 0.357022    
## JOBHome Maker           3.097e-01  2.358e-01  1.314 0.188979    
## JOBLawyer              1.798e-01  1.916e-01  0.938 0.348195    
## JOBManager             -4.673e-01  1.928e-01 -2.424 0.015348 *  
## JOBProfessional         2.623e-01  2.002e-01  1.310 0.190294    
## JOBStudent              2.746e-01  2.409e-01  1.140 0.254280    
## TRAVTIME                1.493e-02  2.105e-03  7.091 1.33e-12 *** 
## CAR_USEPrivate          -7.869e-01  1.025e-01 -7.680 1.59e-14 *** 
## BLUEBOOK                2.070e-05  5.921e-06 -3.496 0.000473 *** 
## TIF                     -5.618e-02  8.141e-03 -6.901 5.17e-12 *** 
## CAR_TYPEPanel Truck     5.310e-01  1.829e-01  2.903 0.003694 ** 
## CAR_TYPEPickup          5.420e-01  1.125e-01  4.818 1.45e-06 *** 
## CAR_TYPESports Car      1.067e+00  1.446e-01  7.377 1.62e-13 *** 
## CAR_TYPESUV              7.894e-01  1.239e-01  6.369 1.91e-10 *** 
## CAR_TYPEVan              7.015e-01  1.403e-01  5.002 5.68e-07 *** 
## RED_CARyes              -1.634e-02  9.674e-02 -0.169 0.865834    
## OLDCLAIM                -1.115e-05  4.394e-06 -2.537 0.011172 *  
## CLM_FREQ                 1.718e-01  3.196e-02  5.377 7.55e-08 *** 
## REVOKEDYes              7.916e-01  1.026e-01  7.715 1.21e-14 *** 
## MVR PTS                 1.124e-01  1.523e-02  7.381 1.57e-13 *** 
## CAR_AGE                  -3.696e-03  8.409e-03 -0.440 0.660251    
## URBANICITYHighly Urban/ Urban  2.449e+00  1.263e-01  19.388 < 2e-16 *** 
## ---                        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
```

```

## Null deviance: 7533.1 on 6527 degrees of freedom
## Residual deviance: 5827.2 on 6490 degrees of freedom
## AIC: 5903.2
##
## Number of Fisher Scoring iterations: 5

```

---

### 3.3 Linear Regression Models for dependent variable TARGET\_AMT

#### 3.3.1 Linear Regression Model 1

```

#Baseline (linear regression)
linearRegModel1 <- lm(formula = TARGET_AMT ~ BLUEBOOK +
                      CAR_AGE +
                      CAR_TYPE +
                      CLM_FREQ +
                      OLDCLAIM,
                      data=train)
summary(linearRegModel1)

##
## Call:
## lm(formula = TARGET_AMT ~ BLUEBOOK + CAR_AGE + CAR_TYPE + CLM_FREQ +
##     OLDCLAIM, data = train)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -3763   -1597   -1117   -297 104469
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.042e+03 1.967e+02  5.295 1.23e-07 ***
## BLUEBOOK    1.810e-03 8.597e-03  0.210 0.833307    
## CAR_AGE     -4.808e+01 1.072e+01 -4.486 7.37e-06 ***
## CAR_TYPEPanel Truck 7.741e+02 2.612e+02  2.963 0.003054 ** 
## CAR_TYPEPickup 6.882e+02 1.822e+02  3.777 0.000160 ***  
## CAR_TYPESports Car 7.034e+02 2.115e+02  3.326 0.000886 ***  
## CAR_TYPESUV   5.532e+02 1.611e+02  3.435 0.000597 ***  
## CAR_TYPEVan   9.643e+02 2.268e+02  4.253 2.14e-05 ***
## CLM_FREQ      4.042e+02 5.779e+01  6.995 2.93e-12 ***
## OLDCLAIM     4.720e-03 7.728e-03  0.611 0.541369    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4705 on 6518 degrees of freedom
## Multiple R-squared:  0.01945,    Adjusted R-squared:  0.01809
## F-statistic: 14.36 on 9 and 6518 DF,  p-value: < 2.2e-16

```

#### 3.3.2 Linear Regression Model 2

```

#Baseline (linear regression)
linearRegModel2 <- lm(formula = TARGET_AMT ~ URBANICITY +
                      MVR PTS +

```

```

CAR_USE +
CAR_TYPE +
CAR_TYPE +
TIF +
MSTATUS +
TRAVTIME +
REVOKE +
PARENT1,
data=train)
summary(linearRegModel2)

##
## Call:
## lm(formula = TARGET_AMT ~ URBANICITY + MVR PTS + CAR_USE + CAR_TYPE +
##      CAR_TYPE + TIF + MSTATUS + TRAVTIME + REVOKE + PARENT1,
##      data = train)
##
## Residuals:
##    Min      1Q Median      3Q     Max
##   -5989   -1671    -852     249 103828
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                227.510   267.099   0.852 0.394365
## URBANICITYHighly Urban/ Urban 1407.279   145.912   9.645 < 2e-16 ***
## MVR PTS                   210.704    27.108   7.773 8.86e-15 ***
## CAR USEPrivate            -971.332   139.845  -6.946 4.13e-12 ***
## CAR TYPEPanel Truck        -43.760   255.789  -0.171 0.864166
## CAR TYPEPickup             369.661   185.113   1.997 0.045872 *
## CAR TYPESports Car         799.531   204.728   3.905 9.50e-05 ***
## CAR TYPESUV                615.412   155.300   3.963 7.49e-05 ***
## CAR TYPEVan                590.626   225.135   2.623 0.008725 **
## TIF                         -53.139   13.671  -3.887 0.000103 ***
## MSTATUSYes                 -454.592   132.811  -3.423 0.000624 ***
## TRAVTIME                     12.849    3.632   3.537 0.000407 ***
## REVOKE Yes                  384.468   176.236   2.182 0.029178 *
## PARENT1 Yes                 990.605   192.623   5.143 2.79e-07 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4616 on 6514 degrees of freedom
## Multiple R-squared:  0.05666,    Adjusted R-squared:  0.05478
## F-statistic:  30.1 on 13 and 6514 DF,  p-value: < 2.2e-16

```

### 3.3.3 Linear Regression Model 3

```

#Baseline (linear regression)
linearRegModel3 <- lm(formula = TARGET_AMT ~ URBANICITY +
                      MVR PTS +
                      CAR_USE +
                      CAR_TYPE +
                      CAR_TYPE +
                      TIF +
                      MSTATUS +
                      TRAVTIME +
                      REVOKE +
                      PARENT1 +

```

```

KIDSDRV +
CLM_FREQ +
INCOME +
CAR_AGE +
SEX +
BLUEBOOK
, data=train)
summary(linearRegModel3)

##
## Call:
## lm(formula = TARGET_AMT ~ URBANICITY + MVR PTS + CAR_USE + CAR_TYPE +
##     CAR_TYPE + TIF + MSTATUS + TRAVTIME + REVOKED + PARENT1 +
##     KIDSDRV + CLM_FREQ + INCOME + CAR_AGE + SEX + BLUEBOOK,
##     data = train)
##
## Residuals:
##    Min      1Q Median      3Q      Max
##   -5748   -1683    -800    313 103642
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.247e+02  3.525e+02   0.921 0.357028
## URBANICITYHighly Urban/ Urban 1.581e+03  1.523e+02  10.382 < 2e-16 ***
## MVR PTS                1.805e+02  2.911e+01   6.199 6.03e-10 ***
## CAR USEPrivate          -8.952e+02  1.403e+02  -6.378 1.92e-10 ***
## CAR TYPEPanel Truck     -7.406e+01  2.958e+02  -0.250 0.802279
## CAR TYPEPickup          3.513e+02  1.861e+02   1.888 0.059115 .
## CAR TYPESports Car      9.632e+02  2.440e+02   3.947 7.99e-05 ***
## CAR TYPESUV              7.759e+02  2.013e+02   3.855 0.000117 ***
## CAR TYPEVan              5.817e+02  2.346e+02   2.480 0.013173 *
## TIF                      -5.309e+01  1.362e+01  -3.897 9.82e-05 ***
## MSTATUSYes               -5.762e+02  1.347e+02  -4.276 1.93e-05 ***
## TRAVTIME                 1.210e+01  3.621e+00   3.341 0.000839 ***
## REVOKEDYes               3.298e+02  1.757e+02   1.877 0.060576 .
## PARENT1Yes               7.486e+02  1.983e+02   3.775 0.000162 ***
## KIDSDRV                  3.741e+02  1.162e+02   3.219 0.001291 **
## CLM_FREQ                 7.797e+01  5.511e+01   1.415 0.157159
## INCOME                   -6.445e-03  1.463e-03  -4.405 1.08e-05 ***
## CAR_AGE                  -3.480e+01  1.133e+01  -3.072 0.002135 **
## SEXM                      3.297e+02  1.798e+02   1.834 0.066718 .
## BLUEBOOK                 1.638e-02  9.656e-03   1.696 0.089849 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4597 on 6508 degrees of freedom
## Multiple R-squared:  0.06527,    Adjusted R-squared:  0.06254
## F-statistic: 23.92 on 19 and 6508 DF,  p-value: < 2.2e-16

```

## 4.1 Model Selection

### 4.2 Binary logistic regression

#### 4.2.1 Confusion Matrices

```

logRegModel1_prediction <- predict(object = logRegModel1,data=train,type="response")
logRegModel2_prediction <- predict(object = logRegModel2,data=train,type="response")
logRegModel3_prediction <- predict(object = logRegModel3,data=train,type="response")
logRegModel4_prediction <- predict(object = logRegModel4,data=train,type="response")
logRegModel5_prediction <- predict(object = logRegModel5,data=train,type="response")

lvl <- levels(train$TARGET_FLAG)

```

### Confusion Matrix for Model 1:

```

confusionMatrix(data= factor(ifelse(logRegModel1_prediction > 0.5,1,0), levels = lvl),
reference=train$TARGET_FLAG,
positive="1")

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction      0      1
##           0 4517 1267
##           1   289  455
##
##             Accuracy : 0.7616
##                 95% CI : (0.7511, 0.7719)
##     No Information Rate : 0.7362
##     P-Value [Acc > NIR] : 1.325e-06
##
##             Kappa : 0.2496
##
## Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.2642
##             Specificity  : 0.9399
##     Pos Pred Value : 0.6116
##     Neg Pred Value : 0.7809
##             Prevalence : 0.2638
##             Detection Rate : 0.0697
##     Detection Prevalence : 0.1140
##             Balanced Accuracy : 0.6020
##
##             'Positive' Class : 1
##

```

### Confusion Matrix for Model 2:

```

confusionMatrix(data= factor(ifelse(logRegModel2_prediction > 0.5,1,0), levels = lvl),
reference=train$TARGET_FLAG,
positive="1")

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction      0      1
##           0 4428 1040
##           1   378  682
##
##             Accuracy : 0.7828
##                 95% CI : (0.7726, 0.7927)

```

```

##      No Information Rate : 0.7362
##      P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.3621
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##      Sensitivity : 0.3961
##      Specificity : 0.9213
##      Pos Pred Value : 0.6434
##      Neg Pred Value : 0.8098
##      Prevalence : 0.2638
##      Detection Rate : 0.1045
##      Detection Prevalence : 0.1624
##      Balanced Accuracy : 0.6587
##
##      'Positive' Class : 1
##

```

### Confusion Matrix for Model 3:

```

confusionMatrix(data=factor(ifelse(logRegModel3_prediction > 0.5,1,0), levels = lvl),
reference=train$TARGET_FLAG,
positive="1")

```

```

## Confusion Matrix and Statistics
##
##          Reference
## Prediction 0     1
##           0 4422 1035
##           1   384  687
##
##          Accuracy : 0.7826
##          95% CI : (0.7724, 0.7926)
##      No Information Rate : 0.7362
##      P-Value [Acc > NIR] : < 2.2e-16
##
##          Kappa : 0.3631
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##      Sensitivity : 0.3990
##      Specificity : 0.9201
##      Pos Pred Value : 0.6415
##      Neg Pred Value : 0.8103
##      Prevalence : 0.2638
##      Detection Rate : 0.1052
##      Detection Prevalence : 0.1641
##      Balanced Accuracy : 0.6595
##
##      'Positive' Class : 1
##

```

### Confusion Matrix for Model 4:

```

confusionMatrix(data=factor(ifelse(logRegModel4_prediction > 0.5,1,0), levels = lvl),
reference=train$TARGET_FLAG,
positive="1")

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0     1
##           0 4434  986
##           1  372  736
##
##             Accuracy : 0.792
##                 95% CI : (0.7819, 0.8018)
## No Information Rate : 0.7362
## P-Value [Acc > NIR] : < 2.2e-16
##
##             Kappa : 0.3952
##
## McNemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4274
##             Specificity  : 0.9226
## Pos Pred Value : 0.6643
## Neg Pred Value : 0.8181
## Prevalence    : 0.2638
## Detection Rate : 0.1127
## Detection Prevalence : 0.1697
## Balanced Accuracy : 0.6750
##
## 'Positive' Class : 1
##

```

#### Confusion Matrix for Model 5:

```

confusionMatrix(data=factor(ifelse(logRegModel5_prediction > 0.5,1,0), levels = lvl),
reference=train$TARGET_FLAG,
positive="1")

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   0     1
##           0 4434  973
##           1  372  749
##
##             Accuracy : 0.794
##                 95% CI : (0.7839, 0.8037)
## No Information Rate : 0.7362
## P-Value [Acc > NIR] : < 2.2e-16
##
##             Kappa : 0.4026
##
## McNemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4350
##             Specificity  : 0.9226
## Pos Pred Value : 0.6682
## Neg Pred Value : 0.8200
## Prevalence    : 0.2638
## Detection Rate : 0.1147
## Detection Prevalence : 0.1717
## Balanced Accuracy : 0.6788
##
```

```
##      'Positive' Class : 1
##
```

#### 4.2.2 Figure 9. ROC curves for models 1-5

```
par(mfrow = c(5,2))

roc1 <- roc(predictor = logRegModel1_prediction, response = train$TARGET_FLAG, print.thres=c(0.25,0.30,0.35,0.40,0.45,0.5),
main="Model 1", plot=TRUE)

roc2 <- roc(predictor = logRegModel2_prediction, response = train$TARGET_FLAG, print.thres=c(0.25,0.30,0.35,0.40,0.45,0.5),
main="Model 2", plot=TRUE)

roc3 <- roc(predictor = logRegModel3_prediction, response = train$TARGET_FLAG, print.thres=c(0.25,0.30,0.35,0.40,0.45,0.5),
main="Model 3", plot=TRUE)

roc4 <- roc(predictor = logRegModel4_prediction, response = train$TARGET_FLAG, print.thres=c(0.25,0.30,0.35,0.40,0.45,0.5),
main="Model 4", plot=TRUE)

roc5 <- roc(predictor = logRegModel5_prediction, response = train$TARGET_FLAG, print.thres=c(0.25,0.30,0.35,0.40,0.45,0.5),
main="Model 5", plot=TRUE)
```

### 4.3 Linear regression model option summary of TARGET\_AMT

#### 4.3.1 Figure 12. Residual Plots

```
par(mfrow=c(1, 4)) plot(linearRegModel2)
par(mfrow=c(1, 4)) plot(linearRegModel2)
par(mfrow=c(1, 4)) plot(linearRegModel3)
```

#### 4.3.2 Figure 13. Plots of model residuals

```
par(mfrow=c(1,2)) hist(linearRegModel1$residuals, xlab = "Residuals", ylab = "Number of records", main = "Model 1") hist(linearRegModel2$residuals, xlab = "Residuals", ylab = "Number of records", main = "Model 2") hist(linearRegModel3$residuals, xlab = "Residuals", ylab = "Number of records", main = "Model 3")
```

## 5.1 Conclusions

```
rawTest$target_prob <- predict(logRegModel5, newdata = rawTest) rawTest$TARGET_FLAG_PRED <- ifelse(rawTest$target_prob >= -1.362578885, 1, 0) rawTest$TARGET_AMT_PRED <- ifelse(rawTest$TARGET_FLAG_PRED == 0, 0, predict(linearRegModel1,
newdata = rawTest))

rawTest %>% write.csv(., "insurance_pred.csv", row.names = F)
```