# Claims Prediction

Challenges in Business Analytics and Data Science

Zachary Palmore

4/17/2021

## Challenge 4

---

```r
# Packages
library(tidyverse)
library(kableExtra)
library(ggcorrplot)
library(reshape2)
library(bestNormalize)
library(caret)
library(MASS)
library(pROC)
library(stats)
library(ROCR)
theme_set(theme_minimal())
```

## Purpose

In this challenge we will explore, analyze and model a data set containing approximately 8000 records representing a customer 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 multiple linear regression and binary logistic regression models 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. We can only use the variables given (or variables derived from the variables provided). Below is a short description of the variables of interest in the data set:

```
# short descriptions of variables as table from matrix
vardesc <- data.frame(matrix(c(
'INDEX',    'Identification variable',
'TARGET_FLAG',  'Was car in a crash? 1 = Yes, 0 = No',
'TARGET_AMT',   'Cost of car crash',
'AGE',  'Age of driver',
'BLUEBOOK', 'Value of vehicle',
'CAR_AGE',  'Vehicle age',
'CAR_TYPE', 'Type of car',
'CAR_USE', 'Main purpose the vehicle is used for',
'CLM_FREQ', 'Number of claims filed in past five years',
'EDUCATION',    'Maximum education level',
'HOMEKIDS', 'Number of children at home',
'HOME_VAL', 'Value of driver\'s home',
'INCOME',   'Annual income of the driver',
'JOB',  'Type of job by standard collar categories',
'KIDSDRIV', 'Number of children who drive',
'MSTATUS',  'Marital status',
'MVR_PTS',  'Motor vehicle inspection points',
'OLDCLAIM', 'Total claims payout in past five years',
'PARENT1',  'Single parent status',
'RED_CAR',  '1 if car is red, 0 if not',
'REVOKED',  'License revoked in past 7 years status',
'SEX',  'Driver gender',
'TIF',  'Time in force',
'TRAVETIME',    'Distance to work in minutes',
'URBANICITY',   'Category of how urban the area the driver lives is',
'YOJ', 'Number of years on the job'
),  byrow = TRUE, ncol = 2))
colnames(vardesc) <- c('Variable', 'Description')
kbl(vardesc, booktabs = T, caption = "Variable Descriptions") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F)
```

Table 1: Variable Descriptions

| Variable | Description |
|---|---|
| INDEX | Identification variable |
| TARGET_FLAG | Was car in a crash? 1 = Yes, 0 = No |
| TARGET_AMT | Cost of car crash |
| AGE | Age of driver |
| BLUEBOOK | Value of vehicle |
| CAR_AGE | Vehicle age |
| CAR_TYPE | Type of car |
| CAR_USE | Main purpose the vehicle is used for |
| CLM_FREQ | Number of claims filed in past five years |
| EDUCATION | Maximum education level |
| HOMEKIDS | Number of children at home |
| HOME_VAL | Value of driver's home |
| INCOME | Annual income of the driver |
| JOB | Type of job by standard collar categories |
| KIDSDRIV | Number of children who drive |
| MSTATUS | Marital status |
| MVR_PTS | Motor vehicle inspection points |
| OLDCLAIM | Total claims payout in past five years |
| PARENT1 | Single parent status |
| RED_CAR | 1 if car is red, 0 if not |
| REVOKED | License revoked in past 7 years status |
| SEX | Driver gender |
| TIF | Time in force |
| TRAVETIME | Distance to work in minutes |
| URBANICITY | Category of how urban the area the driver lives is |
| YOJ | Number of years on the job |

## Introduction

There are 8161 observations of 26 variables in this data set. Each variable is a statistic describing the behavior of an individual driver. Presumably, they are connected to the presence or absence of an accident and contain enough information to estimate the cost of a claim for the accident. To begin, we read in two data sets, one for model training appropriately named, 'tdata,' and one for evaluation named 'edata.'

```
tdata <- read.csv(
  "https://raw.githubusercontent.com/palmorezm/msds/main/Business%20Analytics%20and%20Data%20Mining/HW4/
edata <- read.csv(
  "https://raw.githubusercontent.com/palmorezm/msds/main/Business%20Analytics%20and%20Data%20Mining/HW4/
```

We capture the first four initial observations of driver behavior. In this case, we are looking for any immediate or glaring problems with the first few rows. For example, we check that the data type matches the intended variable, that the observation makes logical sense for its intended column header, and that the data is organized appropriately into rows and columns among other big picture things. Those observations are displayed in table 2 titled "Initial Observations."

```
initialobs <- tdata[1:4,]
kbl(t(initialobs), booktabs = T, caption = "Initial Observations") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F) %>%
  add_header_above(c(" ", " ", "Row Number", " ", " ")) %>%
  footnote(c("Includes the first four observations of all variables in the data"))
```

Table 2: Initial Observations

| | | Row Number | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| INDEX | 1 | 2 | 4 | 5 |
| TARGET_FLAG | 0 | 0 | 0 | 0 |
| TARGET_AMT | 0 | 0 | 0 | 0 |
| KIDSDRIV | 0 | 0 | 0 | 0 |
| AGE | 60 | 43 | 35 | 51 |
| HOMEKIDS | 0 | 0 | 1 | 0 |
| YOJ | 11 | 11 | 10 | 14 |
| INCOME | $67,349 | $91,449 | $16,039 | |
| PARENT1 | No | No | No | No |
| HOME_VAL | $0 | $257,252 | $124,191 | $306,251 |
| MSTATUS | z_No | z_No | Yes | Yes |
| SEX | M | M | z_F | M |
| EDUCATION | PhD | z_High School | z_High School | <High School |
| JOB | Professional | z_Blue Collar | Clerical | z_Blue Collar |
| TRAVTIME | 14 | 22 | 5 | 32 |
| CAR_USE | Private | Commercial | Private | Private |
| BLUEBOOK | $14,230 | $14,940 | $4,010 | $15,440 |
| TIF | 11 | 1 | 4 | 7 |
| CAR_TYPE | Minivan | Minivan | z_SUV | Minivan |
| RED_CAR | yes | yes | no | yes |
| OLDCLAIM | $4,461 | $0 | $38,690 | $0 |
| CLM_FREQ | 2 | 0 | 2 | 0 |
| REVOKED | No | No | No | No |
| MVR_PTS | 3 | 0 | 3 | 0 |
| CAR_AGE | 18 | 1 | 10 | 6 |
| URBANICITY | Highly Urban/ Urban | Highly Urban/ Urban | Highly Urban/ Urban | Highly Urban/ Urban |

*Note:*

Includes the first four observations of all variables in the data

As a positive, each row does seem to contain the proper corresponding variable. There is no blatant mixing. However, there are noticeable issues. The variables ‘INCOME’ and ‘HOME_VAL’ contain ‘$’ character symbols which are completely irrelevant and will cause major problems if we were to interpret or analyze with the data as is. Interestingly, there is also a ‘z_’ present in some categories and not in others. To understand how these variables may affect the analysis as is and to learn exactly what must be done to prepare the data, we must explore further.

## Data Exploration

Before we delve into the nitty gritty of this data set, we should consider what effect each of these variables might exert on the outcome. Since there are two targets of different types, and thus two models (one binary logistic classifier and one multiple linear regression) there could be an influence on either or both models. As we understand it, the theoretical effects of each variable are recorded in the table below.

```r
# theoretical effects
vareffects <- data.frame(matrix(c(
'INDEX',    'None',
'TARGET_FLAG',  'None',
'TARGET_AMT',   'None',
'AGE',  'Youngest and Oldest may have higher risk of accident',
'BLUEBOOK', 'Unknown on probability of collision but correlated with payout',
'CAR_AGE',  'Unknown on probability of collision but correlated with payout',
'CAR_TYPE', 'Unknown on probability of collision but correlated with payout',
'CAR_USE', 'Commerical vehicles might increase risk of accident',
'CLM_FREQ', 'Higher claim frequency increases likelihood of future claims',
'EDUCATION',    'Theoretically higher education levels lower risk',
'HOMEKIDS', 'Unknown',
'HOME_VAL', 'Theoretically home owners reduce risk due to more responsible driving',
'INCOME',   'Theoretically wealthier drivers have fewer accidents',
'JOB',  'Theoretically white collar+ jobs are safer',
'KIDSDRIV', 'Increased risk of accident from inexperienced driver',
'MSTATUS',  'Theoretically married people drive safer',
'MVR_PTS',  'Increased risk of accident',
'OLDCLAIM', 'Increased risk of higher payout with previous payout',
'PARENT1',  'Unknown',
'RED_CAR',  'Theoretically increased risk of accident based on urban legend',
'REVOKED',  'Increased risk of accident if revoked',
'SEX',  'Theoretically increased risk of accident for women based on urban legend',
'TIF',  'Decreased risk for those who have greater loyalty',
'TRAVETIME',    'Longer distances increase risk of accident',
'URBANICITY',   'The more urban the area the greater the risk of accident',
'YOJ', 'Decreased risk for those with greater longevity'
),  byrow = TRUE, ncol = 2))
colnames(vareffects) <- c('Variable', 'Effect')
kbl(vareffects, booktabs = T, caption = "Theoretical Variable Effects") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F)
```

Table 3: Theoretical Variable Effects

| Variable | Effect |
|---|---|
| INDEX | None |
| TARGET_FLAG | None |
| TARGET_AMT | None |
| AGE | Youngest and Oldest may have higher risk of accident |
| BLUEBOOK | Unknown on probability of collision but correlated with payout |
| CAR_AGE | Unknown on probability of collision but correlated with payout |
| CAR_TYPE | Unknown on probability of collision but correlated with payout |
| CAR_USE | Commerical vehicles might increase risk of accident |
| CLM_FREQ | Higher claim frequency increases likelihood of future claims |
| EDUCATION | Theoretically higher education levels lower risk |
| HOMEKIDS | Unknown |
| HOME_VAL | Theoretically home owners reduce risk due to more responsible driving |
| INCOME | Theoretically wealthier drivers have fewer accidents |
| JOB | Theoretically white collar+ jobs are safer |
| KIDSDRIV | Increased risk of accident from inexperienced driver |
| MSTATUS | Theoretically married people drive safer |
| MVR_PTS | Increased risk of accident |
| OLDCLAIM | Increased risk of higher payout with previous payout |
| PARENT1 | Unknown |
| RED_CAR | Theoretically increased risk of accident based on urban legend |
| REVOKED | Increased risk of accident if revoked |
| SEX | Theoretically increased risk of accident for women based on urban legend |
| TIF | Decreased risk for those who have greater loyalty |
| TRAVETIME | Longer distances increase risk of accident |
| URBANICITY | The more urban the area the greater the risk of accident |
| YOJ | Decreased risk for those with greater longevity |

This table considers the effects of both models but they are only theoretical and may not necessarily reflect the true influence. We will evaluate these directly in the model selection process. For now, they will serve as general baseline expectations for exploration and preparation. We continue by exploring the data to determine where munging may be necessary.

Unfortunately, this data needs work even before we are able to make visualizations and contemplate improvements to the model. We consider the amount of missing values in relative proportions to each variable, followed by their respective data types, an example observation of each type, and the quantity of unique factors to each variable. We already know that some major work needs to be done in readjusting data types but we do not know the extent to which each variable needs improvement. This will help narrow down what is needed to prepare the data for modeling. Results are shown in the table:

```r
tdata.nas <- lapply(tdata, function(x) sum(is.na(x)))
tdata.len <- lapply(tdata, function(x) length(x))
tdata.permis <- lapply(tdata, function(x) round(sum(is.na(x))/length(x)*100, 1))
tdata.types <- lapply(tdata, function(x) class(x))
tdata.firstob <- lapply(tdata, function(x) head(x, 1))
tdata.uniques <- lapply(tdata, function(x) length(unique(factor(x))))
tdata.tbl.natypes <- cbind(tdata.nas, tdata.len, tdata.permis, tdata.types, tdata.firstob, tdata.uniques
colnames(tdata.tbl.natypes) <- c("Missing", "Total", "%", "Data Type", "Example", "Factors")
```

```
kbl(tdata.tbl.natypes, booktabs = T, caption = "Data Characteristics") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), full_width = F)
```

Table 4: Data Characteristics

| | Missing | Total | % | Data Type | Example | Factors |
|---|---|---|---|---|---|---|
| INDEX | 0 | 8161 | 0 | integer | 1 | 8161 |
| TARGET_FLAG | 0 | 8161 | 0 | integer | 0 | 2 |
| TARGET_AMT | 0 | 8161 | 0 | numeric | 0 | 1949 |
| KIDSDRIV | 0 | 8161 | 0 | integer | 0 | 5 |
| AGE | 6 | 8161 | 0.1 | integer | 60 | 61 |
| HOMEKIDS | 0 | 8161 | 0 | integer | 0 | 6 |
| YOJ | 454 | 8161 | 5.6 | integer | 11 | 22 |
| INCOME | 0 | 8161 | 0 | character | $67,349 | 6613 |
| PARENT1 | 0 | 8161 | 0 | character | No | 2 |
| HOME_VAL | 0 | 8161 | 0 | character | $0 | 5107 |
| MSTATUS | 0 | 8161 | 0 | character | z_No | 2 |
| SEX | 0 | 8161 | 0 | character | M | 2 |
| EDUCATION | 0 | 8161 | 0 | character | PhD | 5 |
| JOB | 0 | 8161 | 0 | character | Professional | 9 |
| TRAVTIME | 0 | 8161 | 0 | integer | 14 | 97 |
| CAR_USE | 0 | 8161 | 0 | character | Private | 2 |
| BLUEBOOK | 0 | 8161 | 0 | character | $14,230 | 2789 |
| TIF | 0 | 8161 | 0 | integer | 11 | 23 |
| CAR_TYPE | 0 | 8161 | 0 | character | Minivan | 6 |
| RED_CAR | 0 | 8161 | 0 | character | yes | 2 |
| OLDCLAIM | 0 | 8161 | 0 | character | $4,461 | 2857 |
| CLM_FREQ | 0 | 8161 | 0 | integer | 2 | 6 |
| REVOKED | 0 | 8161 | 0 | character | No | 2 |
| MVR_PTS | 0 | 8161 | 0 | integer | 3 | 13 |
| CAR_AGE | 510 | 8161 | 6.2 | integer | 18 | 31 |
| URBANICITY | 0 | 8161 | 0 | character | Highly Urban/ Urban | 2 |

Three variables contain incomplete records including ‘AGE’, ‘YOJ’, and ‘CAR_AGE’ with 0.1%, 5.6%, and 6.2% of their data missing respectively. Theoretically each variable would have 8161 total observations as noted in the table. The data types are either integer or numeric and the examples display what the type looks like for easy referencing. A calculation of the unique factors for each variable is included to gauge whether converting to a factor data type would be right for the variable and count the number of unique values to each. These are major concerns.

Minima, quartiles, averages, and maximums were computed to compare the numeric integer variables. Although the order of the variables remains the same as in the previous table, we added a missing values column with the row identifier ‘NA’ to count the number missing for tracking purposes. We put this together in a table called Summary Characteristics. Of course, several of the variables will need to be altered before we can evaluate if the data makes sense in a real-life scenario. These are shown as NA in the table.

```
tdata.summary.tbl <- summary(tdata)
kbl(t(tdata.summary.tbl), booktabs = T, caption = "Summary Characteristics") %>%
  kable_styling(latex_options = c("striped", "scale_down", "hold_position"), full_width = F)
```

Table 5: Summary Characteristics

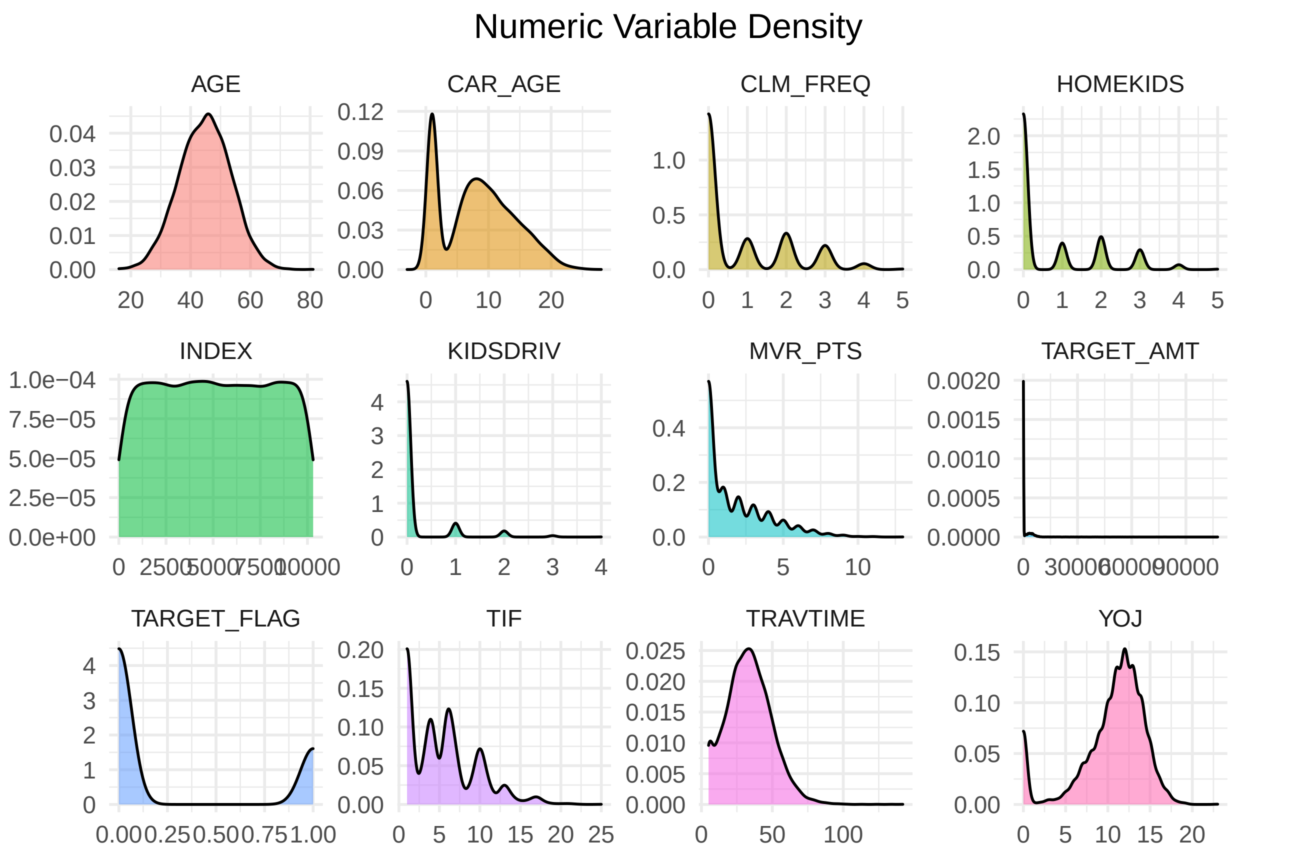
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| INDEX | Min. : 1 | 1st Qu.: 2559 | Median : 5133 | Mean : 5152 | 3rd Qu.: 7745 | Max. :10302 | NA |
| TARGET_FLAG | Min. :0.0000 | 1st Qu.:0.0000 | Median :0.0000 | Mean :0.2638 | 3rd Qu.:1.0000 | Max. :1.0000 | NA |
| TARGET_AMT | Min. : 0 | 1st Qu.: 0 | Median : 0 | Mean : 1504 | 3rd Qu.: 1036 | Max. :107586 | NA |
| KIDSDRIV | Min. :0.0000 | 1st Qu.:0.0000 | Median :0.0000 | Mean :0.1711 | 3rd Qu.:0.0000 | Max. :4.0000 | NA |
| AGE | Min. :16.00 | 1st Qu.:39.00 | Median :45.00 | Mean :44.79 | 3rd Qu.:51.00 | Max. :81.00 | NA's :6 |
| HOMEKIDS | Min. :0.0000 | 1st Qu.:0.0000 | Median :0.0000 | Mean :0.7212 | 3rd Qu.:1.0000 | Max. :5.0000 | NA |
| YOJ | Min. : 0.0 | 1st Qu.: 9.0 | Median :11.0 | Mean :10.5 | 3rd Qu.:13.0 | Max. :23.0 | NA's :454 |
| INCOME | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| PARENT1 | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| HOME_VAL | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| MSTATUS | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| SEX | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| EDUCATION | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| JOB | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| TRAVTIME | Min. : 5.00 | 1st Qu.: 22.00 | Median : 33.00 | Mean : 33.49 | 3rd Qu.: 44.00 | Max. :142.00 | NA |
| CAR_USE | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| BLUEBOOK | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| TIF | Min. : 1.000 | 1st Qu.: 1.000 | Median : 4.000 | Mean : 5.351 | 3rd Qu.: 7.000 | Max. :25.000 | NA |
| CAR_TYPE | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| RED_CAR | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| OLDCLAIM | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| CLM_FREQ | Min. :0.0000 | 1st Qu.:0.0000 | Median :0.0000 | Mean :0.7986 | 3rd Qu.:2.0000 | Max. :5.0000 | NA |
| REVOKED | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |
| MVR_PTS | Min. : 0.000 | 1st Qu.: 0.000 | Median : 1.000 | Mean : 1.696 | 3rd Qu.: 3.000 | Max. :13.000 | NA |
| CAR_AGE | Min. :-3.000 | 1st Qu.: 1.000 | Median : 8.000 | Mean : 8.328 | 3rd Qu.:12.000 | Max. :28.000 | NA's :510 |
| URBANICITY | Length:8161 | Class :character | Mode :character | NA | NA | NA | NA |

Notice, there are quite a few NA values and our binary outcomes such as 'KIDSDRIV',and even our 'TARGET_FLAG' appear to be skewed heavily. It will require a closer at their distributions look to be sure of this but regardless, they will need to be dealt with if we plan to use them in our models. Factors like the single parent indicator, 'PARENT1', the individuals binary gender 'SEX', whether a driver's license was revoked 'REVOKED' and others like 'MSTATUS,''URBANICITY,' and 'CAR_USE,' take on a character data types that are not useful in modeling. For practical purposes, they should be made factors so that their categories may be understood fully.

Several of these variables also have statistics that do not make logical sense. For example, 'CAR_AGE' has a minimum value of -3. This is not possible. Other numeric variables are clearly sets of dollar values but show as character strings. These will need to converted to numeric data types and the quantitative value portion of the string extracted. This and the factorization of cateogircal variables without a necessary order should help eliminate the many missing or 'NA' calculations performed above.

Generally, it is best to avoid reviewing the distribution of variables prior to munging the full data set to visualize all variables simultaneously.However, we have enough variables that are already of the numeric type that plotting them each together might overcrowd the chart. A new strategy should be considered. In this effort, we review the density of the numeric variables as they exist now to evaluate agreement with the assumptions of linearity.

```
tdata %>%
  select_if(is.numeric) %>%
  gather %>%
  ggplot() +
  facet_wrap(~ key, scales = "free") +
  geom_density(aes(value, color = value, fill = key, alpha = .5)) + theme(axis.title = element_blank(),
```



Variables ‘CLM_FREQ,’ ‘HOMEKIDS,’ and ‘KIDSDRIV’ show integer values for a discrete distribution. These will be treated differently in the preparation process because their non-conformity with a linearity will not effect their ability to predict in the binary logistic classification model. Results in that model to examine the probability of ‘TARGET_FLAG’ will remain stable. However, the multiple linear regression model suffers a loss of potential predictors due to their inability to meet the assumptions of linear regression.

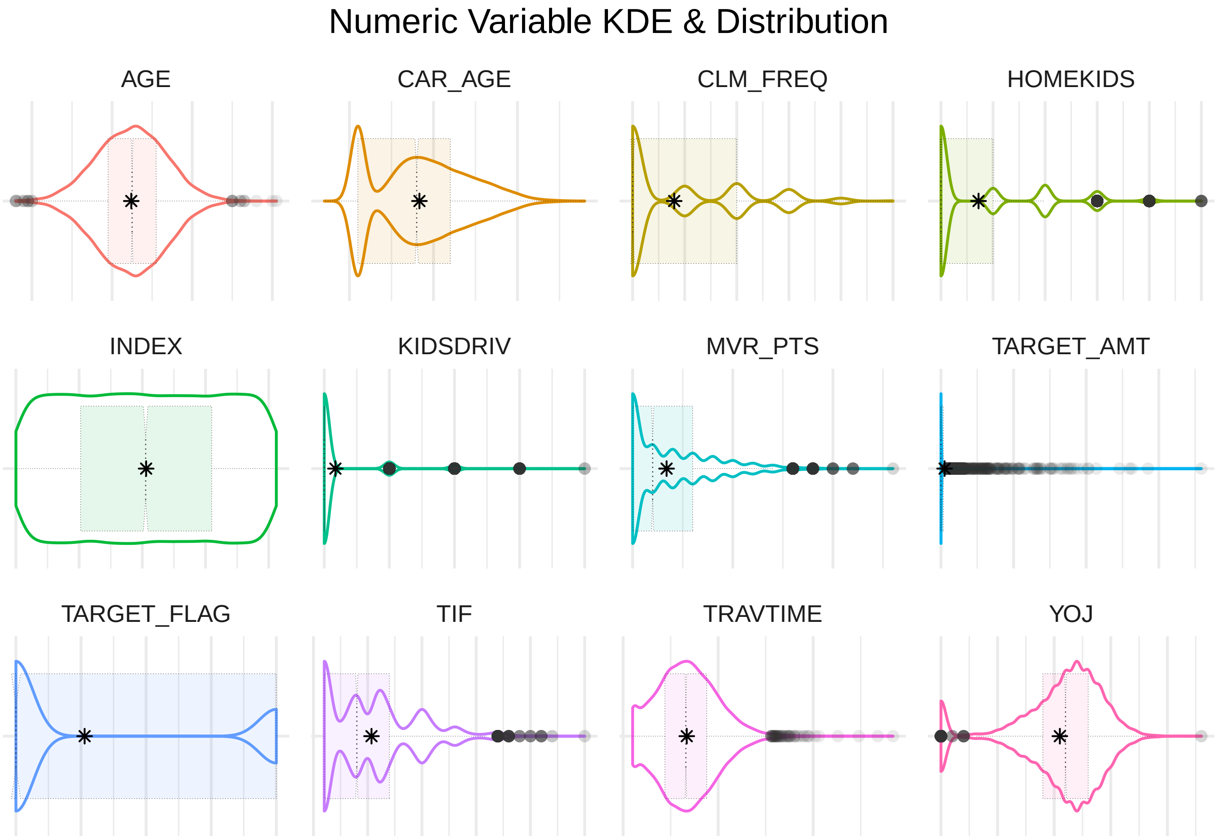
Our continuous variables, ‘AGE,’ ‘CAR_AGE,’ ‘TRAVTIME,’ ‘MVR_PTS,’ and ‘YOJ’ are much better suited to predict ‘TARGET_AMT.’ To our benefit, age is normally distributed with the bulk of the driver ages’ falling between 30 and 60. Given the theoretical effect of younger drivers on the likelihood of an accident this variable could be useful in our logistic regression model especially when converted to a binary outcome of young or not. But there are now reasons to doubt their agreement with the assumptions.

Consider the assumption of normality where only the ‘AGE’ variable satisfies. All other non-discrete and non-integer values, including ‘CAR_AGE,’ ‘YOJ,’ ‘MVR_PTS,’ ‘TIF,’ and ‘TRAVTIME’ are poorly classified as normal, if at all. There are at least two distinct peaks in the distribution of ‘YOJ,’ ‘TIF,’ and ‘CAR_AGE.’ Our ‘MVR_PTS’ is too dense at the front of the distribution but still manages to have less skewness than our ‘TARGET_AMT.’ The variable closest to satisfying this assumption is perhaps ‘TRAVTIME’ which has an unproven theoretical effect to increase the chance of an accident as the time increases and may also be bimodal.

Given the problematic nature of a major assumption of linear regression it would be tough to say modeling with any of these could make useful predictions. Without intense transformations we risk guessing wildly at the resultant amount. However, the degree to which transformations must be performed to cause this data to appear normal would grossly misrepresent the data and greatly increase our error rate in both models if we chose to use assign them places in each. However, we must continue knowing this and attempt to improve upon the expectation. In this endevaour, we also check a few other assumptions with violin plots and a boxplot estimation.

```
tdata %>%
  select_if(is.numeric) %>%
  gather %>%
  ggplot(aes(value, key)) +
  facet_wrap(~ key, scales = "free") +
  geom_violin(aes(color = key, alpha = 1)) +
  geom_boxplot(aes(fill = key, alpha = .5), notch = TRUE, size = .1, lty = 3) +
  stat_summary(fun.y = mean, geom = "point",
               shape = 8, size = 1.5, color = "#000000") +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        legend.position = "none") +
  ggtitle("Numeric Variable KDE & Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```

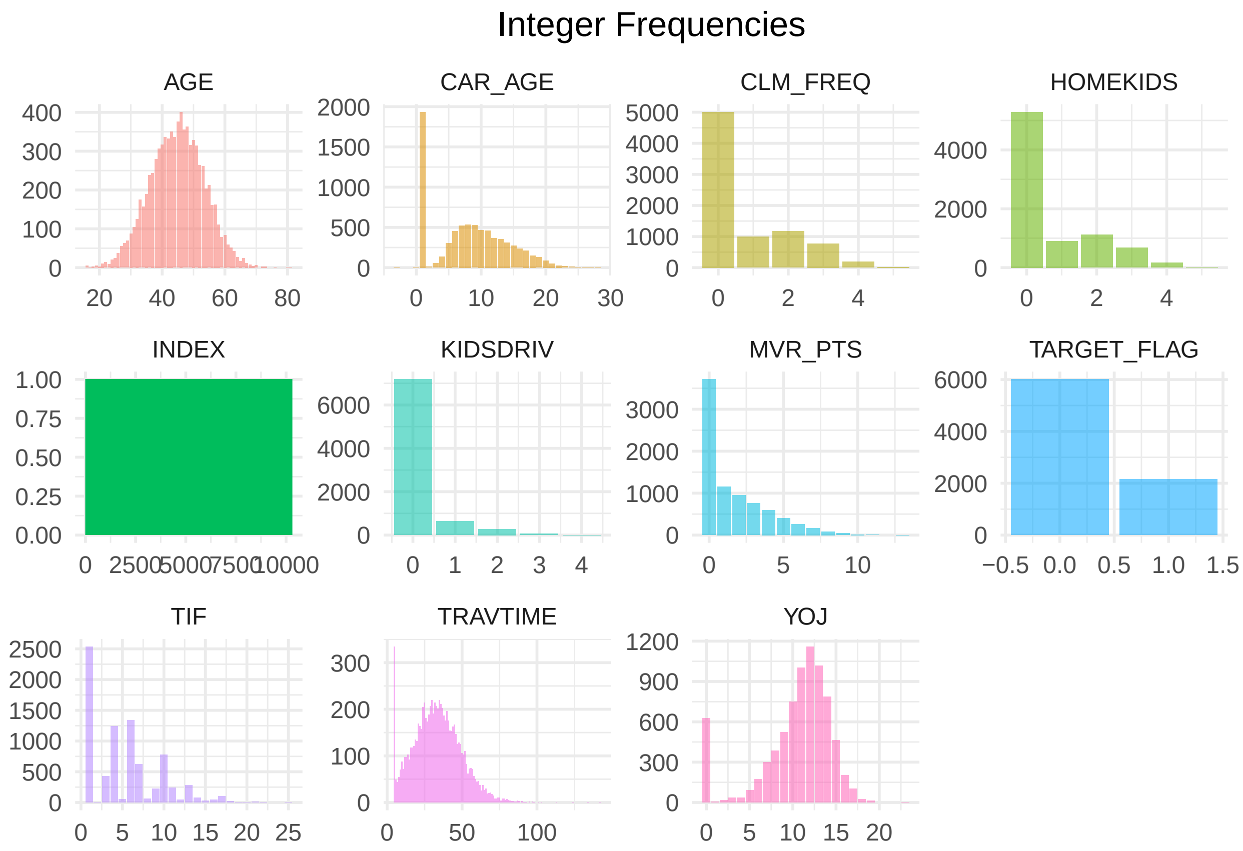
```
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```



While the good news is our discrete integer values (that will be partially reorganized into unordered factors) continue to give us hope that model accuracy will be reasonable for our binary model, the existence of numerous outliers, high levels of variation in some distributions, and little to no normality, reduce the ability of our multiple linear regression model to produce accurate results. In this visual we can clearly note the presence of outliers for our 'TARGET_AMT' variable. Unfortunately, it appears nearly all of our useful data (where the amount is greater than zero) are considered outliers. A selection of all values greater than zero might be useful but it eliminates the option of the multiple linear regression model to predict a value of zero when there is no claim made. This is an interesting conundrum and one we should consider of the utmost importance when preparing the data and building models.

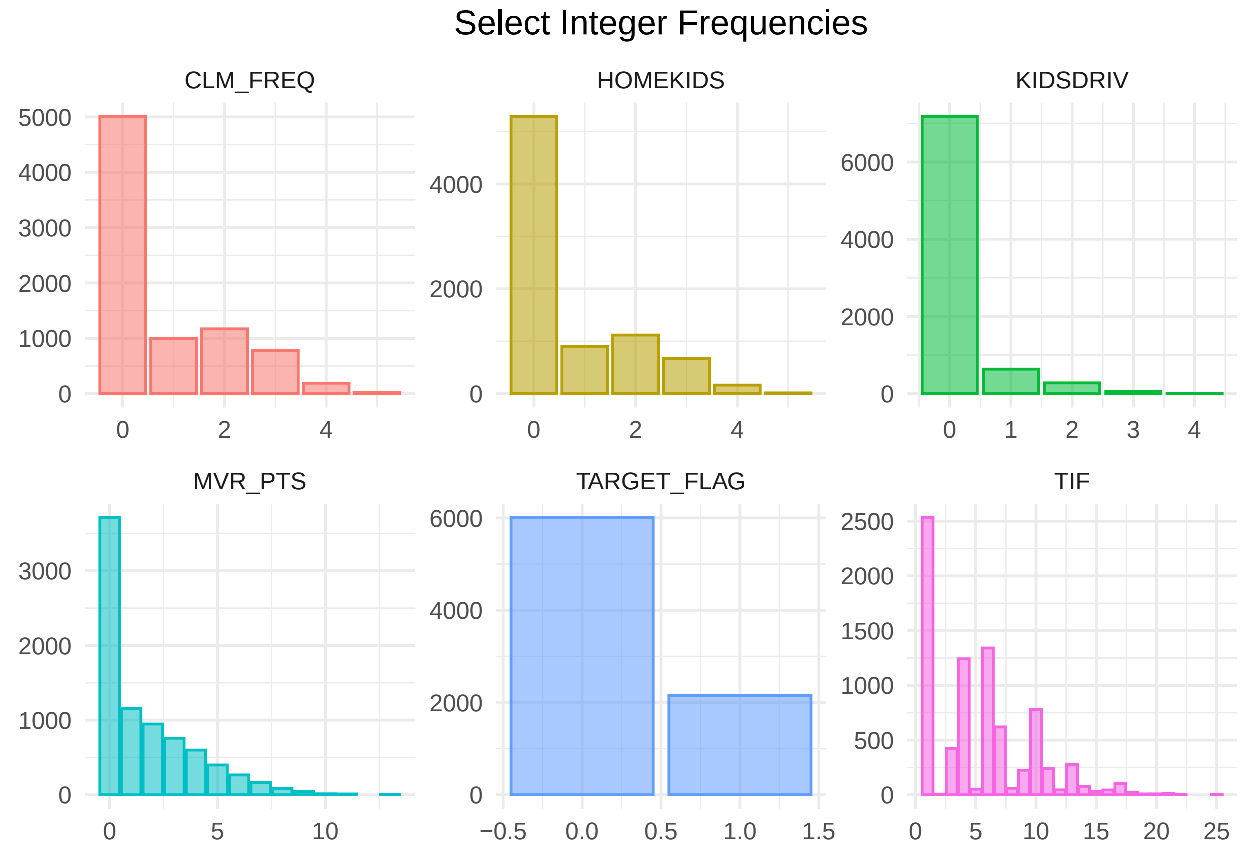
While some other numeric variables appear to confirm our density plots, others show new issues. The normally distributed ‘AGE’ variable contains many outliers above and below its distribution with a slightly larger number of older drivers stretching the distribution upwards. The outliers for our discrete integer values shown as black dots would be much better suited to a bar chart. This might also let any other variables that exhibit a discrete pattern with categorical values show through a bit better.

```
tdata %>%
  select_if(is.integer) %>%
  gather() %>%
  filter(value == 0 | 1) %>%
  group_by(key) %>%
  ggplot() +
  facet_wrap(~ key, scales = "free") +
  geom_bar(aes(value, color = value, fill = key, alpha = .5)) + theme(axis.title = element_blank(), leg
```



Outliers contained in the ‘TRAVTIME’ variable in the violin plot were mainly commute times of the ‘greater than’ kind. However, in this we notice there are more drivers with a zero minute commute time than at the average of all commuters. A similar spike in the years on job or ‘YOJ’ variable indicates that there are just as many drivers with near zero year on the job as there are drivers with 10 years on the job. We select a few of these to take a closer look at their categories and how they fall into equally weighted bars.

```
tdata %>%
  dplyr::select(TARGET_FLAG, MVR_PTS, CLM_FREQ, HOMEKIDS, KIDSDRIV, TIF) %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_bar(aes(value, color = key, fill = key, alpha = .5)) + theme(axis.title = element_blank(), legen
```



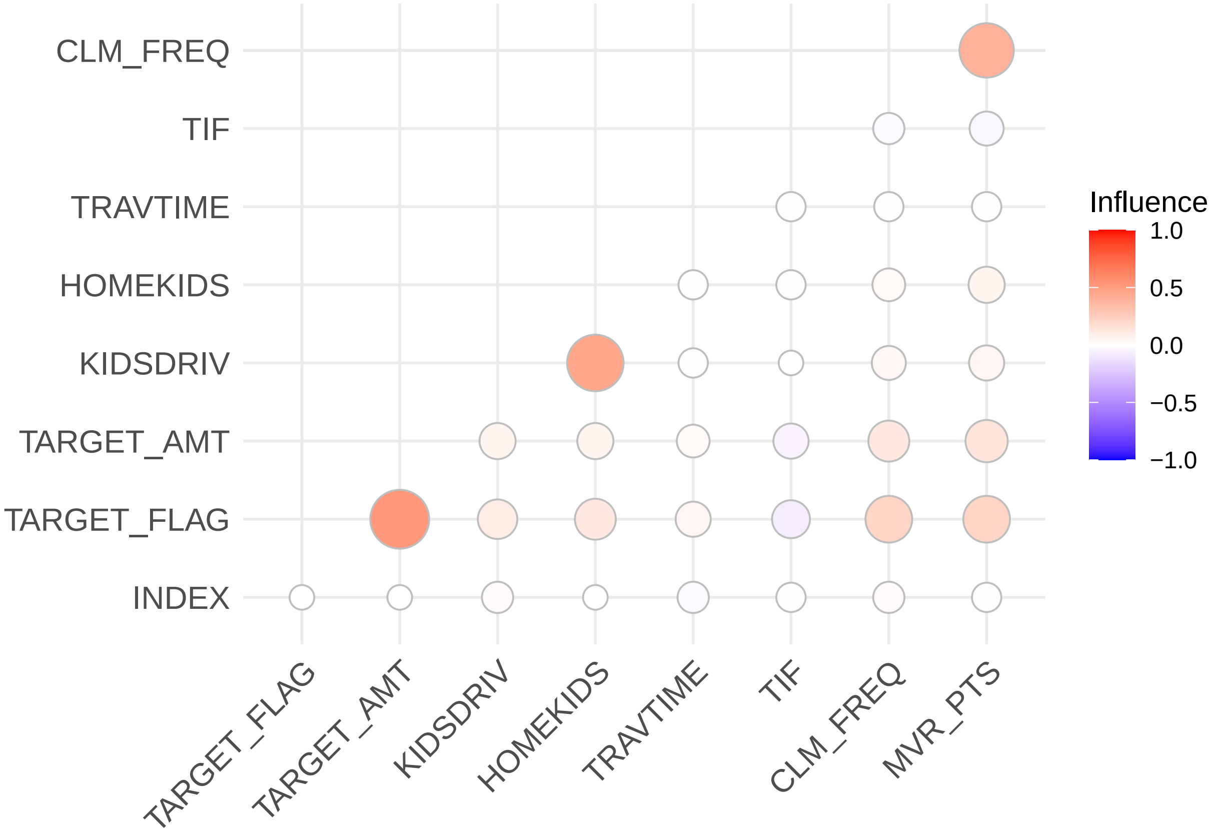
The quantity of drivers who do not have kids that drive is much larger than any driver that has kids. The same applies to the number of kids at home and the claim frequencies. The magnitude of difference in these categories is severe and may cause issues when trying to predict the probability of an accident. If the claim frequency is majority zero, along with the number of kids at home and almost every other variable follows the same pattern, then predicting the minority class becomes more difficult. We should expect a noise-filled model with faint signal. This issue in the binary logistic classifier makes it even worse for the multiple linear regression model.

Building on these insights, this infant minority class must be focused on to improve the accuracy and precision of both model types. Some might consider it productive to oversample the minority class, however, if we cannot reasonably assume the majority class fits the assumptions of linear regression nor that either case is particularly well-cleaned (transformed, prepared, and error-controled) enough to base real predictions on than the purpose of oversampling is moot. Otherwise, we would be trying to build a model off of incorrect assumptions and inevitably come to the wrong conclusions. Not to mention, these conclusions would be far from the reality on which the data is supposed to be representative.

To get a better sense of where these data may overlap and the strength of their relationships we consider the correlations of the numeric variables. This includes the discrete integer values with categories and the continuous numeric variables that have not undergone data type changes. There is less of a need to consider the other non-numeric variables at this time since the binary logistic classification model does not rely on the same rules as multiple linear regression to interpret the data.

```
tdata %>%
  select_if(is.numeric) %>%
  cor() %>%
  ggcorrplot(method = "circle", type="upper",
             ggtheme = ggplot2::theme_minimal, legend.title = "Influence") + coord_flip()
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



Again, since this is prior to the data preparation stage, we include the index of each observation. This is a good base indicator of variables with little to no correlation with our target variables 'TARGET_FLAG' and 'TARGET_AMT.' The directional vector of the influence in each variable is color coded from -1 to 1 with the most positive correlations being closest to 1. The size of each circle represents the strength in magnitude of the relationship between the variables.

Of course, our two target variables, 'TARGET_FLAG' and 'TARGET_AMT' are quite strongly positively correlated. The same applies for the variables 'KIDSDRIV' and 'HOMEKIDS' given that one driver should not be able to have a kid driving without having at least one kid at home. These stronger correlations pass the sanity check.

Alternatively, the weaker, but perhaps more interesting variables, include mostly positive correlations. It seems it is much easier to increase your chances of an accident than it is to reduce them. Our only negatively correlated (blue) relationship comes from 'TIF' and our targets. Looking at this variable with 'TARGET_FLAG' you can reduce your chance of getting into an accident increasing the time you spend at the same insurance company. This is likely a result of people who stick around because they have not had an accident and their rate has remained steady. Otherwise, the driver would probably search for new insurance.

These correlations elucidate the relationship of our ‘TARGET_FLAG’ or the chance of an accident better than the relationship of the variables with the ‘TARGET_AMT.’ Keep in mind that all of these variables are numeric types that could be used model with multiple linear regression. They are not going to be, since several of them are categorical factors and not continuous distributions, but recall that all but one fails at least one assumption of linear regression. As mentioned, the accuracy of this linear regression model will be affected by these poorly correlated data.

---

## Data Preparation

This section will implement the changes necessary to build models with the data. Since we have two model types, we must separate the data into groups. One group will have data specialized for the binary logistic classification model and the other for the multiple linear regression. We have already determined that we will have a limited pool of data to model with for the multiple linear regression. With that said, we still attempt to scrape together some resemblance of a realistic linear regression model ignoring the fact that the data is not in accordance with the assumptions of linear regression.

To begin, we extract the numeric variables present in 'INCOME,' 'HOME_VAL,' and others like it. In doing so we drop the '$' sign but retain the value. Then we impute the median value for those that had missing values and evaluate the changes with summary statistics. These are recorded and calculated separately. Once the differences are checked to ensure that the variables are not drastically far off from the original values we recombine them with the training data. This process is shown below along with the first table of summary statistics from the numeric imputed variables.

```r
# Select character variables
chars <- tdata %>%
  dplyr::select_if(is.character)
# Use function to extract dollars
to_num <- function(x){
  x <- as.character(x)
  x <- gsub(",", "", x)
  x <- gsub("\\$", "", x)
  as.numeric(x)
}
# Specify those dollar variables
income.values <- to_num(chars$INCOME)
home.values <- to_num(chars$HOME_VAL)
bluebook.values <- to_num(chars$BLUEBOOK)
oldclaim.values <- to_num(chars$OLDCLAIM)
concept_df <- as.data.frame(cbind(income.values,
                    home.values,
                    bluebook.values,
                    oldclaim.values))
income.values.stat <- to_num(chars$INCOME)
home.values.stat <- to_num(chars$HOME_VAL)
bluebook.values.stat <- to_num(chars$BLUEBOOK)
oldclaim.values.stat <- to_num(chars$OLDCLAIM)
# impute median values for missing variables
income.values[is.na(income.values)] <-
  median(income.values, na.rm = TRUE)
home.values[is.na(home.values)] <-
  median(home.values, na.rm = TRUE)
bluebook.values[is.na(bluebook.values)] <-
  median(bluebook.values, na.rm = TRUE)
oldclaim.values[is.na(oldclaim.values)] <-
  median(oldclaim.values, na.rm = TRUE)
# Recombine into data frame
dollar.values <-
  data.frame(cbind(income.values,
                   home.values,
                   bluebook.values,
                   oldclaim.values))
dollar.values.stats <-
```

```
  data.frame(cbind(income.values.stat,
                   home.values.stat,
                   bluebook.values.stat,
                   oldclaim.values.stat))
# Join with training data
tdata <- data.frame(cbind(tdata, dollar.values))
# Check the difference
dollar.values.tbl <- summary(dollar.values)
dollar.values.stats.tbl <- summary(dollar.values.stats)
kbl(dollar.values.tbl, booktabs = T, caption = "Imputed Summary Statistics") %>%
kable_styling(latex_options = c("striped", "hold_position"), full_width = F)
```

Table 6: Imputed Summary Statistics

| income.values | home.values | bluebook.values | oldclaim.values |
|---|---|---|---|
| Min. : 0 | Min. : 0 | Min. : 1500 | Min. : 0 |
| 1st Qu.: 29707 | 1st Qu.: 0 | 1st Qu.: 9280 | 1st Qu.: 0 |
| Median : 54028 | Median :161160 | Median :14440 | Median : 0 |
| Mean : 61469 | Mean :155225 | Mean :15710 | Mean : 4037 |
| 3rd Qu.: 83304 | 3rd Qu.:233352 | 3rd Qu.:20850 | 3rd Qu.: 4636 |
| Max. :367030 | Max. :885282 | Max. :69740 | Max. :57037 |

We compare this to the second table and observe the changes. If the differences between these is small, we can add the imputed variables to our original training data rather than eliminating the rows with missing variables altogether. The second table is shown here:

```
kbl(dollar.values.stats.tbl, booktabs = T, caption = "Original Summary Statistics") %>%
kable_styling(latex_options = c("striped", "hold_position"), full_width = F)
```

Table 7: Original Summary Statistics

| income.values.stat | home.values.stat | bluebook.values.stat | oldclaim.values.stat |
|---|---|---|---|
| Min. : 0 | Min. : 0 | Min. : 1500 | Min. : 0 |
| 1st Qu.: 28097 | 1st Qu.: 0 | 1st Qu.: 9280 | 1st Qu.: 0 |
| Median : 54028 | Median :161160 | Median :14440 | Median : 0 |
| Mean : 61898 | Mean :154867 | Mean :15710 | Mean : 4037 |
| 3rd Qu.: 85986 | 3rd Qu.:238724 | 3rd Qu.:20850 | 3rd Qu.: 4636 |
| Max. :367030 | Max. :885282 | Max. :69740 | Max. :57037 |
| NA's :445 | NA's :464 | NA | NA |

Since the differences between these variables is small and the values that were missing are no longer missing, we recombine the imputed data with the training data for later use. It turns out, it was reasonable to make these imputations, even though the data sets are skewed and do not contain total linearity or normality.

Next, we convert the categorical variables to factors rather than the integer or characters they were. Although the analysis will still interpret these factors as integers to perform computations, we will be able to recognize the various levels associated with each variable rather than a non-descriptive number. As a special note, we are going to interpret each of these factors as an unordered set.

```
# Covert categorical variables to factors
factors <- tdata %>%
  dplyr::select("PARENT1",
         "MSTATUS",
         "SEX",
         "EDUCATION",
         "JOB",
         "CAR_USE",
         "CAR_TYPE",
         "RED_CAR",
         "REVOKED",
         "URBANICITY")
factors <- data.frame(lapply(factors, function(x) as.factor(x)))
factors <- factors %>%
  rename("parent1" = "PARENT1",
         "mstatus" = "MSTATUS",
         "sex" = "SEX",
         "education" = "EDUCATION",
         "job" = "JOB",
         "car_use" = "CAR_USE",
         "car_type" = "CAR_TYPE",
         "red_car" = "RED_CAR",
         "revoked" = "REVOKED",
         "urbanicity" = "URBANICITY")
tdata <- cbind(tdata, factors)
```

There was a highly unrealistic value for the variable ‘CAR_AGE.’ This should be excluded from the data set to avoid further damage to the modeling process. We simply find where the value is less than zero and set that value to NA. A summary of both is run to see how this changes the data.

```
# Exclude unrealistic values
tdata <- tdata %>%
  mutate(car_age = ifelse(CAR_AGE<0, NA, CAR_AGE))
summary(tdata$car_age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    1.00    8.00    8.33   12.00   28.00     511
```

```
summary(tdata$CAR_AGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -3.000   1.000   8.000   8.328  12.000  28.000     510
```

As we can see there was only one value in the ‘CAR_AGE’ variable that contained a negative number. For our purposes, we will consider the rest realistic. Another variable that we do not need as it provide not value to the model is the ‘INDEX’ variable. We remove this and other unnecessary variables from the training data by selecting the variables that we suspect we may need. The total number of variables present after this selection process is shown.

```
# Drop INDEX and other unnecessary columns
tdata <- tdata %>%
  dplyr::select("TARGET_FLAG",
         "TARGET_AMT",
         "KIDSDRIV",
         "AGE",
         "HOMEKIDS",
         "YOJ",
         "TRAVTIME",
         "TIF",
         "CLM_FREQ",
         "MVR_PTS",
         "income.values",
         "home.values",
         "bluebook.values",
         "oldclaim.values",
         "parent1",
         "mstatus",
         "sex",
         "education",
         "job",
         "car_use",
         "car_age",
         "car_type",
         "red_car",
         "revoked",
         "urbanicity")
# Check total variables present
length(colnames(tdata))
```

```
## [1] 25
```

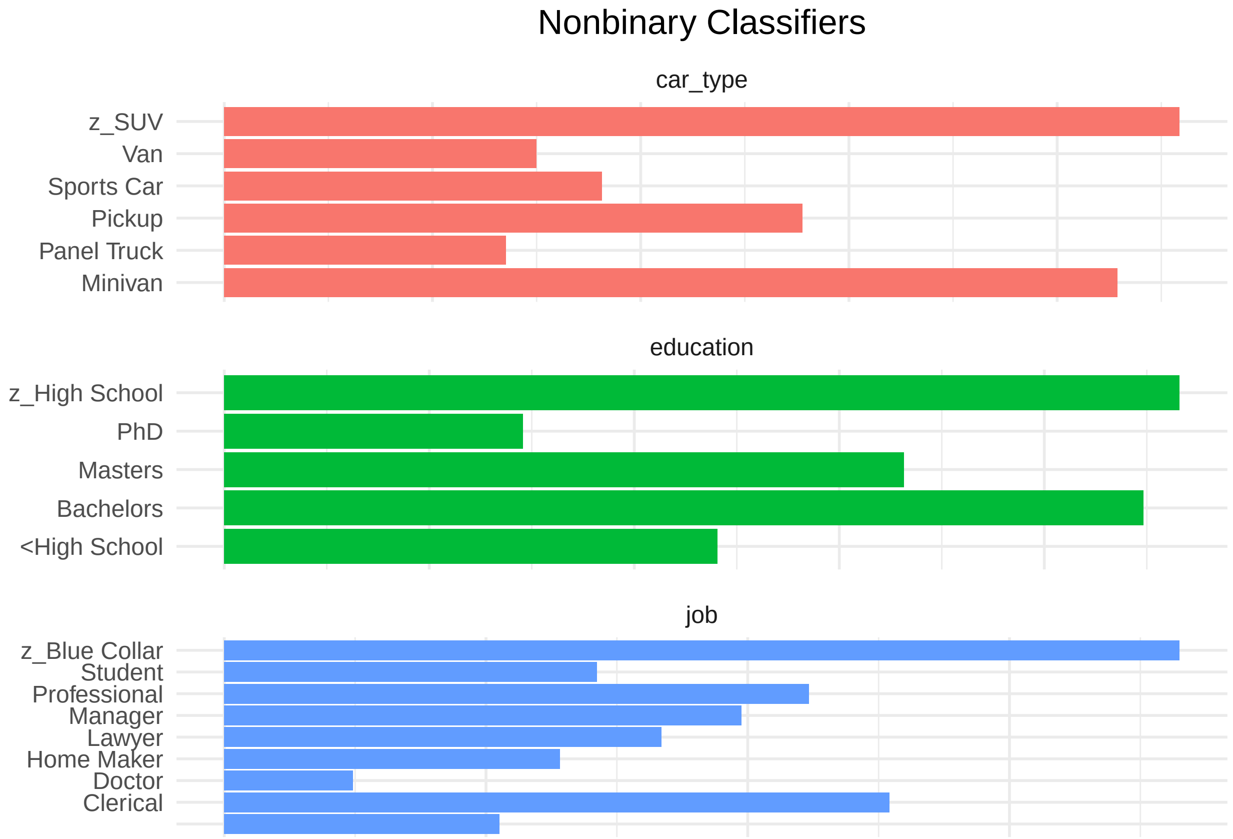
Because these variables still have missing values, further imputation is needed. The variables of 'AGE,' 'YOJ,' and 'CAR_AGE' are filled with the median value from each of their respective distributions. Of course, we created one of the missing value for 'CAR_AGE' but this one value does not effect the overall variable distribution but by 0.01% or about one in our total observations nor does it change the other 510 missing observations. This difference is negligible but we impute, continue, and repeat the process of imputation on the other aforementioned variables.

```
# More imputation
tdata$AGE[is.na(tdata$AGE)] <-
  median(tdata$AGE, na.rm = T)
tdata$YOJ[is.na(tdata$YOJ)] <-
  median(tdata$YOJ, na.rm = T)
tdata$car_age[is.na(tdata$car_age)] <-
  median(tdata$car_age, na.rm = T)
sum(is.na(tdata))
```

```
## [1] 0
```

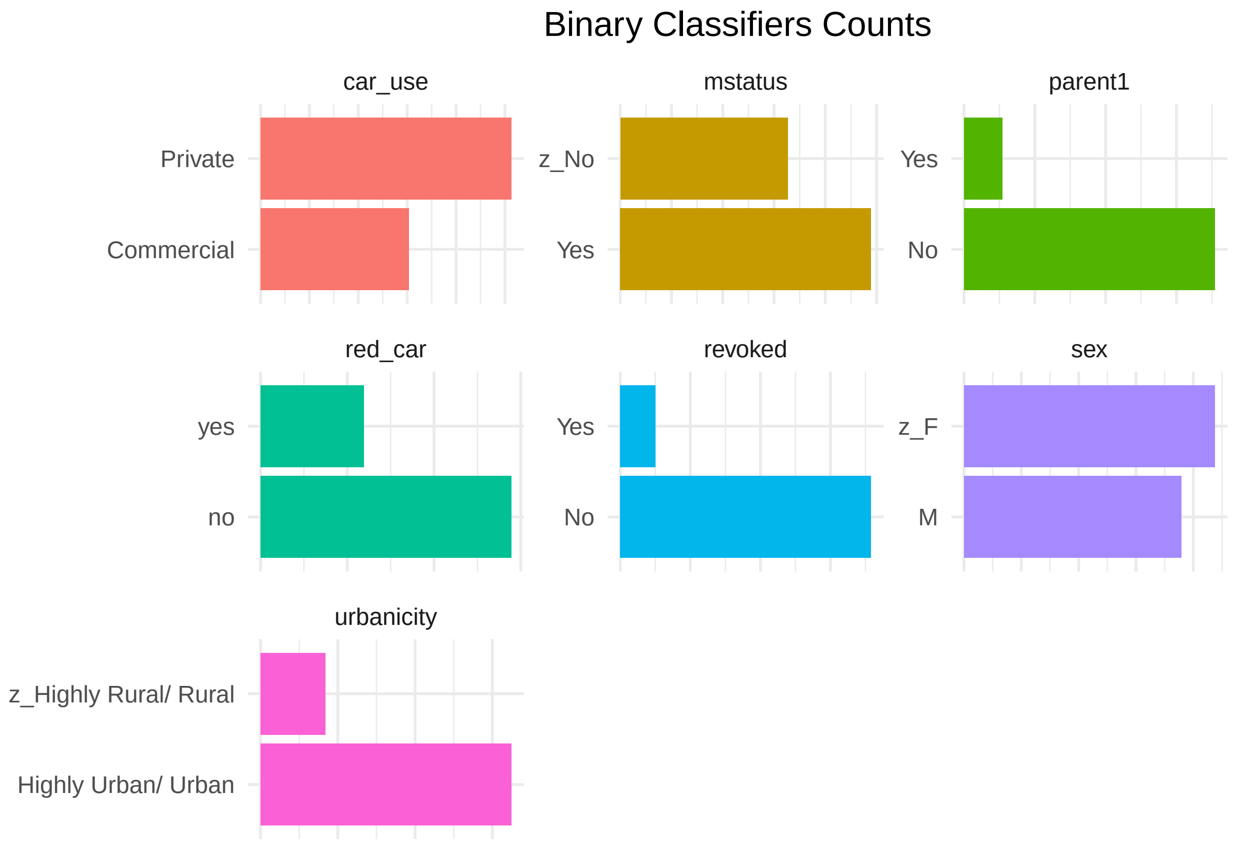
As expected, there are exactly 510 missing values. These all fall into the 'CAR_AGE' variable that will no longer need to be used. At this point, we consider excluding the non-imputed variable for the age of the driver's car from the data set to avoid confusion later. While we do not perform the exclusionary action, it is best to think of it as such. We do this because it may also be useful to our classification model's accuracy given that some of our new features, like if we include a young driver categorical factor, would work well with the driver's car age. Consider the nonbinary classifiers to briefly explain.

```
tdata %>%
  dplyr::select(is.factor) %>%
  dplyr::select("car_type", "education", "job") %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, nrow = 3, scales = "free") +
  geom_bar(aes(, fill = key )) + theme(axis.title = element_blank(), axis.text.x = element_blank(), lege
```



We suspect that based on our exploration, these three categories will have the most influence over the probability of an accident if that person is a student with a high school or less than high school education, and if they drive a sports car. These are based on the notion that, citation rates are higher for those who operate sports cars (such as by driver behavior or oversampling by police forces and ultimately police behavior), that less education leads to more risks, and that students are the most likely to take those risks because they are often younger and more susceptible to unintentional reactions from new situations that require experience to navigate safely. These circumstances, create what we might refer to as the risky people category but we refrain from categorizing because such a notion is, at this time, unfounded.

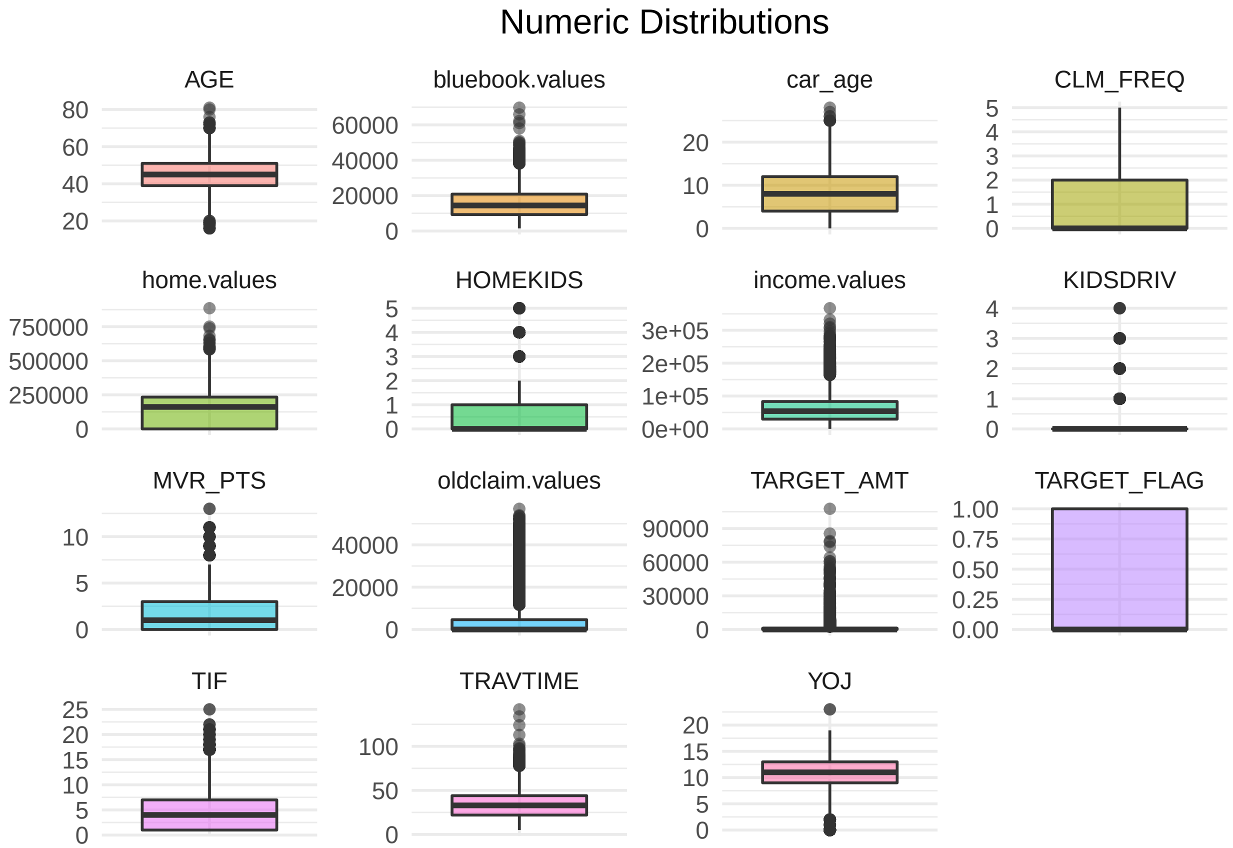
```
tdata %>%
  dplyr::select(is.factor) %>%
  dplyr::select("car_use",
         "mstatus",
         "parent1",
         "red_car",
         "revoked",
         "sex",
         "urbanicity") %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_bar(aes(, fill = key )) + theme(axis.title = element_blank(), axis.text.x = element_blank(), lege
```



Taking a closer look at the counts of the binary classifiers, we can easily see how a model would unfairly and unintentionally pick from the counts that hold the majority of the data. This is problematic but without drastically changing the data and grossly misrepresenting reality, we cannot improve this. Oversampling to try and adjust for the small amount of minority vectors would likely exacerbate the misrepresentation of reality and increase the standard error rate. Instead we keep the variables as is and will place emphasis on using these to make accurate predictions in our binary logistic classification model.

For our multiple linear regression model, things are a little more complicated. None of our variables are well-suited to predict the 'TARGET_AMT' and we are reliant on the results of our first model to inform it. Given this dependent relationship between models, we should reexamine their distributions with our newly formed 'INCOME,' 'HOME_VAL,' and other numeric variables to see if we have anything worthwhile to pull from.

```
tdata %>%
  select_if(is.numeric) %>%
  gather() %>%
  ggplot(aes(key)) +
  facet_wrap(~ key, scales = "free") +
  geom_boxplot(aes(key, value, fill = key, alpha = .5)) + theme(axis.title = element_blank(), axis.text
```



It is going to difficult to pull anything of value from the 'TARGET_AMT' variable. However, we may still be able to increase the accuracy of the model overall with a transformation of the target and a little bit of history. Consider how this model works in a real-life scenario. Often, there is a host of existing data to pull from wherein the analyst can use this data to draw conclusions. This is what we are going to do. In treating some of the data as a historical reference, we will be able to increase the accuracy of the model. However, this is not likely going to improve the ability of the regression to predict future values due to the assumption failure of the values in linear regression. To extract useful information from this we will need a few new features.

```r
# New features
tdata <- tdata %>%
  mutate(city = ifelse(urbanicity == "Highly Urban/ Urban", 0, 1)) %>%
  mutate(young = ifelse(AGE < 25, 1, 0)) %>%
  mutate(clean_rec = ifelse(MVR_PTS == 0, 1, 0)) %>%
  mutate(previous_accident = ifelse(CLM_FREQ == 0 & oldclaim.values == 0, 0, 1)) %>%
  mutate(educated = ifelse(education %in% c("Bachelors", "Masters", "PhD"), 1, 0)) %>%
  mutate(avg_claim = ifelse(CLM_FREQ > 0, oldclaim.values/CLM_FREQ, 0))
```

New features are created to increase the availability of useful prediction points that our first model, the binary classifier, can implement to make the best predictions from. Since our second model relies on the outcome of the first, we should make every effort to improve this model as much as possible. Without an impeccable source of data to pull from, we cannot hope make realistic predictions for our multiple linear regression model when predicting the target amounts of each claim.

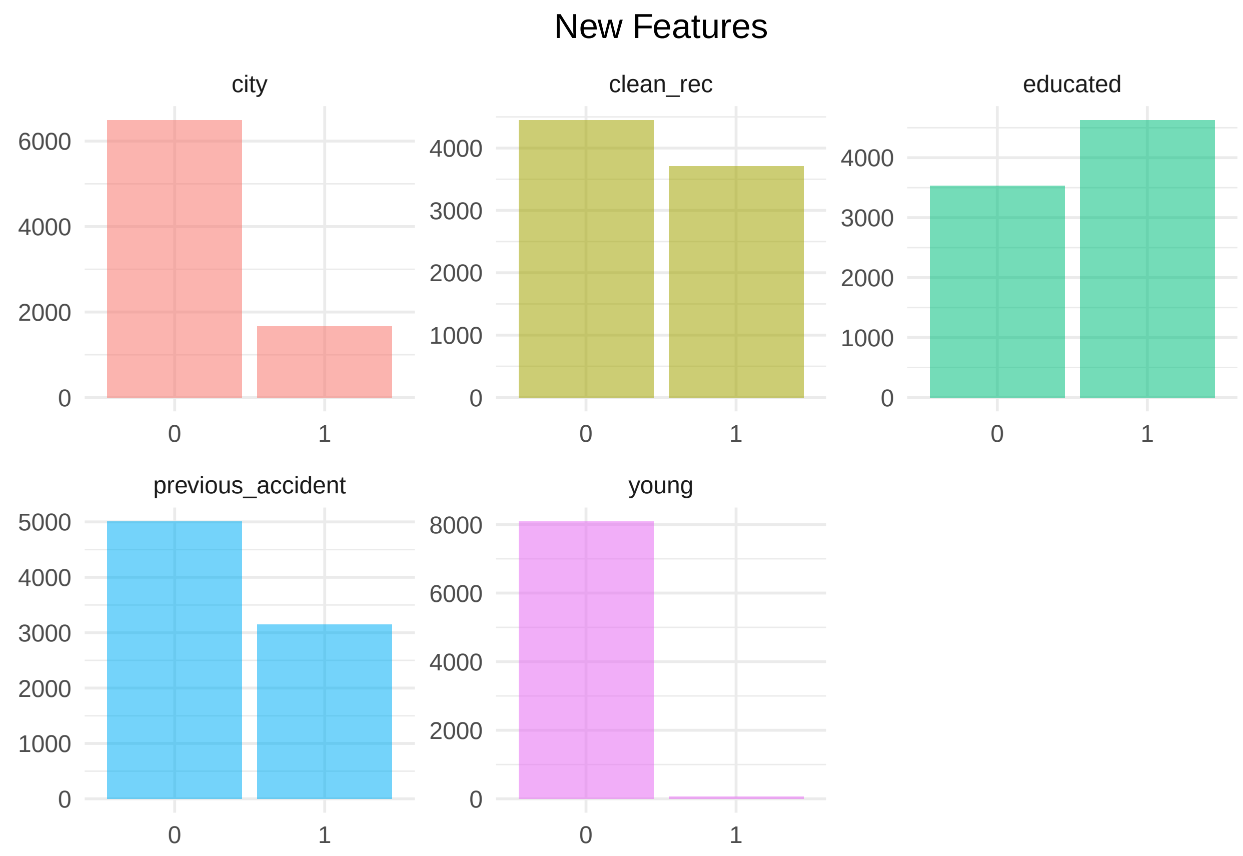
The features include a value for city drivers, since we suspect based on the theoretical effect on the driver from our exploration that the probability of accidents for drivers who drive within areas that are mostly urban increases compared to their less urban counterparts. This likely comes down to car density and the greater the number of cars on the road in a smaller space, the higher the chance of one of those cars getting into an accident.

We also create the feature young, for those who are aged 25 or younger, and clean_rec for those who have zero motor vehicle points. This is based mainly on car rental agreements that state anyone under 25 must pay a higher rate, that is, if they are allowed to drive the vehicle. For those who have a record, which many of them check, they are often prohibited from operating that vehicle until their record is cleared. We do not know what effect these will have on the outcome, but we hypothesize that these features will increase the chance of accidents if the data indicates a yes to any of these features. To enhance their practicality, we make their data type factors.

```
# Convert to factors
tdata$city <- as.factor(tdata$city)
tdata$young <- as.factor(tdata$young)
tdata$clean_rec <- as.factor(tdata$clean_rec)
tdata$previous_accident <- as.factor(tdata$previous_accident)
tdata$educated <- as.factor(tdata$educated)
```

Additionally, features are created to indicate whether a driver has been in a previous accident and the education level of the driver. Hypothetically, the less educated would take more risk. Meanwhile, we also are testing the notion that those who have been a previous accident are more likely to have one in the future. These new binary features are showns in the bar chart of ‘New ’Features’ to demonstrate their amount of drivers in each bin.

```
tdata[26:31] %>%
  select_if(is.factor) %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~key, scales = "free") +
  geom_bar(aes(fill = key, alpha = .5)) + theme(legend.position = "none", axis.title = element_blank())
```



Our highly urban group compared to less urban is weighted in favor of those who a less urban. Indicating that most of the drivers in this study drive in areas that are not highly urbanized. Also, more people are educated with at least a four year degree than not. We would guess that this is likely due to lower rates of insurance for people who do not have at least a four year degree. Very few drivers are less than 25 years old and about half of the drivers have been in a previous accident. Comprehending this hints that the drivers in this study are less likely to have accidents and how results will conclude for the amount. At this time, we suspect the target amount of claims that do get processed will either be a larger than a few thousand dollars or no claim. These drivers' behavior will have a significant impact on the predicted claim amounts since they are directly dependent on the occurrence of an accident.

To deal with the failed assumption of linear regression, we will transform the data but attempt to keep some resemblance of realism for practical purposes. Rather than guess at the best transformation, we use a function to run through many kinds of transformations and have it tell us what the most gaussian distribution is. Then, we reassign those values to a variable 'accident_costs' where the costs are greater than zero. This will be used later. Finally, we chose to focus on select variables for the transformations because none of them were great choices for prediction and transforming each send the error rate even higher.

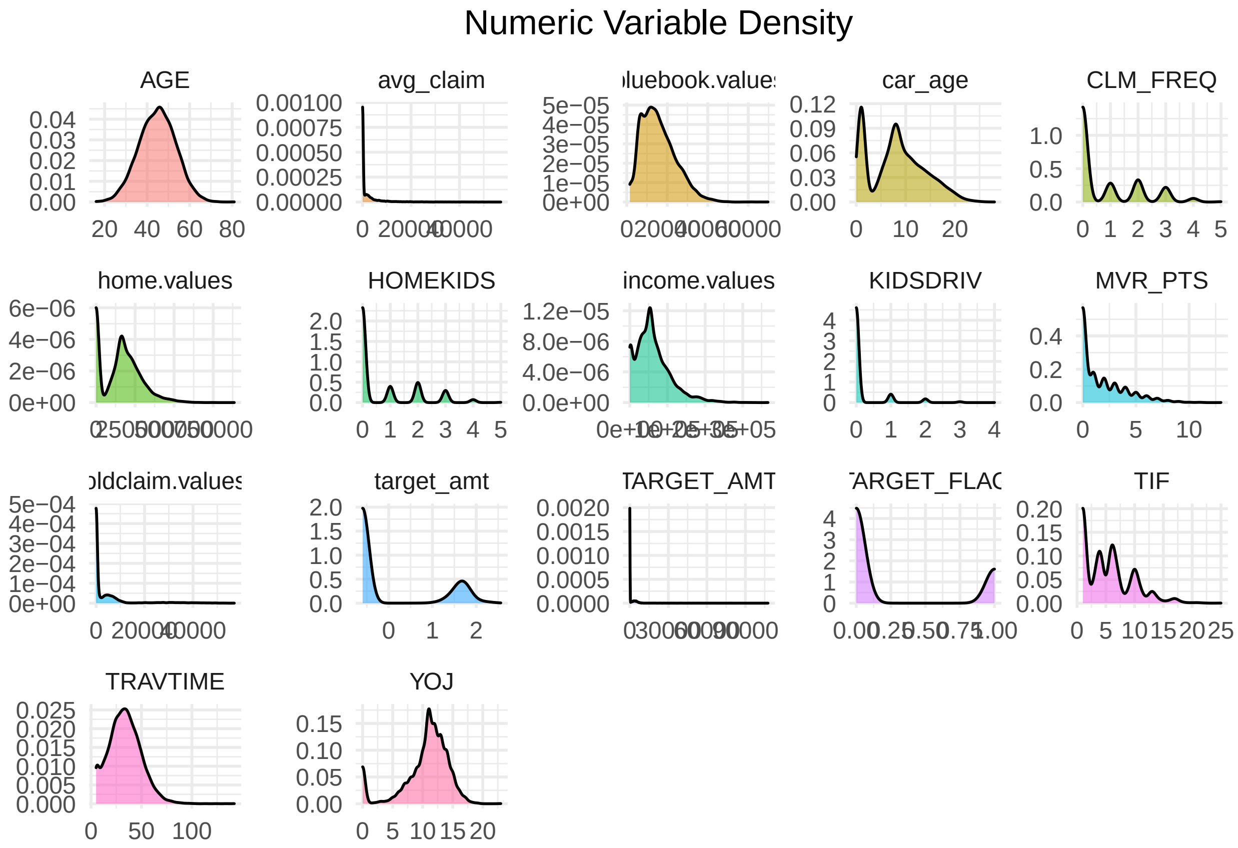
```
# Produce recommended transformations
bestNorms <- tdata[1:11,1:16]
df <- tdata %>%
  select_if(is.numeric)
for (i in colnames(df)) {
  bestNorms[[i]] <- bestNormalize(df[[i]],
                                  allow_orderNorm = FALSE,
                                  out_of_sample =FALSE)
}
```

```
# Continue focusing on realistic values
accident_costs <- tdata$TARGET_AMT[tdata$TARGET_AMT>.0]
```

```
# Focus on selected variables
bestNorms$target_amt$chosen_transform
```

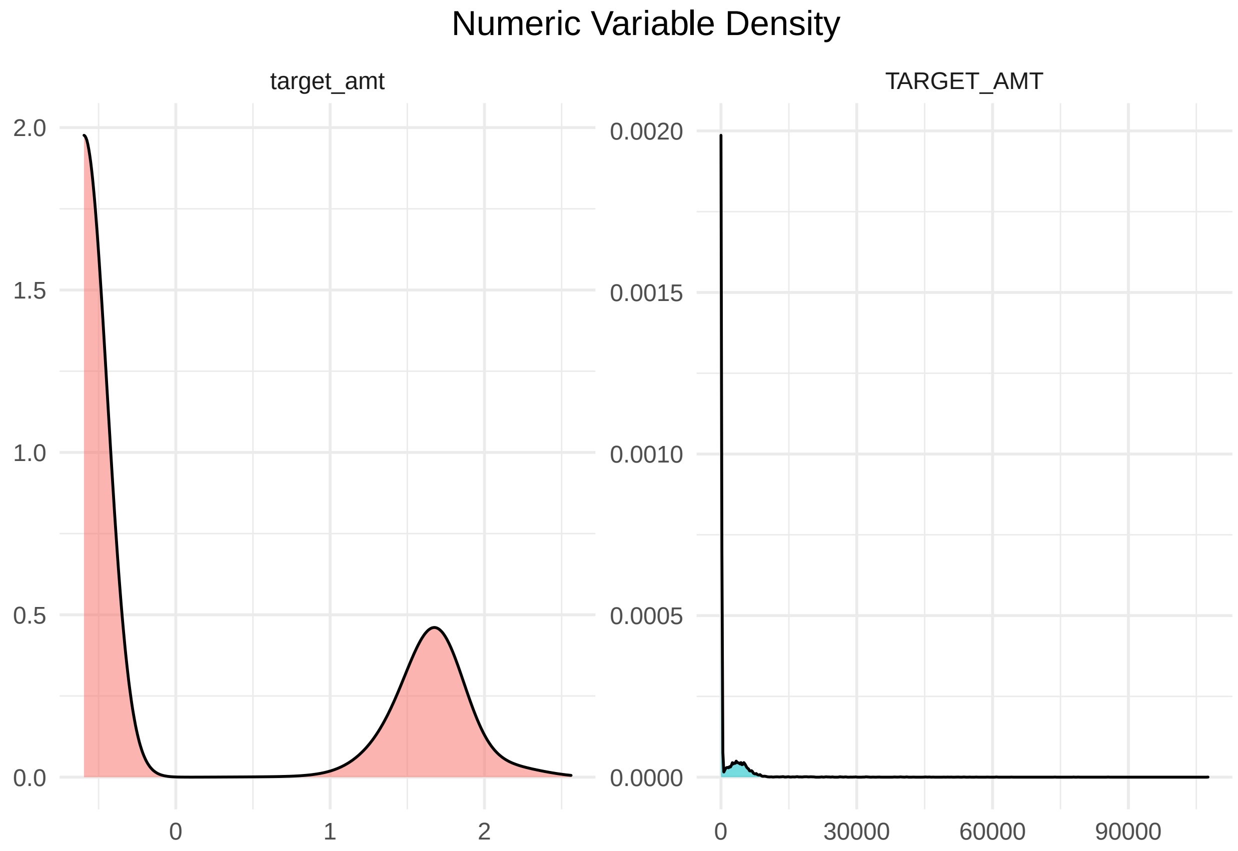
```
## NULL
```

```
tdata$target_amt <- scale(log(tdata$TARGET_AMT + 1))
tdata %>%
  dplyr::select(where(is.numeric)) %>%
  gather %>%
  ggplot() +
  facet_wrap(~ key, scales = "free") +
  geom_density(aes(value, color = value, fill = key, alpha = .5)) + theme(axis.title = element_blank(),
```



Although these are still a long shot from adhering to the condition necessary for performing linear regression, they have improved at the expense of our standard error and its dependencies. For example, the display below shows the differences between the transformed predictor ‘TARGET_AMT’ and its original form. In some ways the data is more normal since everything above zero resembles a near perfect bell-curve. However, there is an ominous presence of values less than zero, which for a graph where we are displaying the dollar amount of a claim, is completely useless. These changes are strictly for modeling purposes and the true values will need a reversal of this transformation to make sense of them. This is the cost of multiple linear regression without normality and it assumes the other conditions of independence, homoscedasticity, and linearity are fulfilled which, we have reasons to doubt.

```
tdata %>%
  dplyr::select(where(is.numeric)) %>%
  dplyr::select("TARGET_AMT","target_amt") %>%
  gather %>%
  ggplot() +
  facet_wrap(~ key, scales = "free") +
  geom_density(aes(value, color = value, fill = key, alpha = .5)) + theme(axis.title = element_blank(),
```



Bearing that in mind we split these variables into training data sets and testing data sets for the model. A 70-30 split should do just fine and one set of the data is used for each model. This makes it easier to differentiate the model types and statistics later. With this split, we can begin building models.

```
# Split 70-30 training test
set.seed(1102)
tindex <- createDataPartition(tdata$TARGET_FLAG, p = .7, list = FALSE, times = 1)
train <- tdata[tindex,]
test <- tdata[-tindex,]
rindex <- tdata %>%
  filter(TARGET_FLAG == 1)
reg.tindex <- createDataPartition(rindex$TARGET_AMT, p = .7, list = FALSE, times = 1)
reg.train <- rindex[reg.tindex,]
reg.test <- rindex[-reg.tindex,]
```

## Model Building

In this first model we are only going to consider how well a previous accident can predict if a driver will have a future accident. We specify that it is a binomial model and exercise the training data set with it. A summary is shown for reference although it will be discussed in more detail in the selection process.

```
model1 <- glm(TARGET_FLAG ~ previous_accident,
              family = binomial(link = "logit"), train)
summary(model1)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ previous_accident, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0069  -0.6351  -0.6351   1.3581   1.8441
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.49869    0.04369  -34.30   <2e-16 ***
## previous_accident1  1.08338    0.06167   17.57   <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: 6613.0  on 5712  degrees of freedom
## Residual deviance: 6297.9  on 5711  degrees of freedom
## AIC: 6301.9
##
## Number of Fisher Scoring iterations: 4
```

The coefficient is significant, however, it is clear this would not be the best for the model. Our standard error is only marginally greater than that of a model without any coefficients. Using the previous accident as a predictor results in a higher chance of the driver having an accident as shown with the estimate above. This is only meant to be a dummy test though because we know that there are more factors to consider in the probability of having an accident as a driver. We should expect improvement in this classification from here.

```
model2 <- glm(TARGET_FLAG ~ previous_accident +
                city + young + clean_rec +
                educated, family = binomial(link = "logit"), train)
summary(model2)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ previous_accident + city + young +
##     clean_rec + educated, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max  
## -1.8218  -0.8464  -0.5816   1.1001   2.6595  
## 
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -0.52339    0.06962  -7.518 5.57e-14 ***
## previous_accident1  0.70805    0.06737  10.510  < 2e-16 ***
## city1               -1.81562    0.12542 -14.477  < 2e-16 ***
## young1               1.26389    0.29142   4.337 1.44e-05 ***
## clean_rec1          -0.31886    0.06865  -4.645 3.41e-06 ***
## educated1           -0.84903    0.06517 -13.027  < 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: 6613  on 5712  degrees of freedom
## Residual deviance: 5859  on 5707  degrees of freedom
## AIC: 5871
## 
## Number of Fisher Scoring iterations: 5
```

This model consider what we would consider the riskiest model. It takes the young, city driver, who we think would take more risks and compares the coefficients. Each one is a significat predictor at the .001 alpha level. All but one of these predictors had a better standard error than the null model. Education stood out as a negative estimator with those in this category probably being mostly students or having less than a bachelors degree and less experience driving overall. We suspect this model is better than the previous one.

```
model3 <- glm(TARGET_FLAG ~ previous_accident +
                city + mstatus + income.values +
                sex + car_use + educated + KIDSDRIV +
                revoked, family = binomial(link = "logit"),
              train)
summary(model3)
```

```
## 
## Call:
## glm(formula = TARGET_FLAG ~ previous_accident + city + mstatus + 
##     income.values + sex + car_use + educated + KIDSDRIV + revoked, 
##     family = binomial(link = "logit"), data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.0469  -0.7604  -0.4518   0.7784   2.8387  
## 
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -5.071e-01  9.509e-02  -5.333 9.66e-08 ***
## previous_accident1   6.289e-01  6.806e-02   9.240  < 2e-16 ***
## city1               -2.106e+00  1.306e-01 -16.128  < 2e-16 ***
## mstatusz_No          8.075e-01  6.805e-02  11.866  < 2e-16 ***
## income.values       -8.460e-06  9.462e-07  -8.941  < 2e-16 ***
## sexz_F               3.020e-01  7.077e-02   4.268 1.98e-05 ***
## car_usePrivate     -7.490e-01  7.226e-02 -10.366  < 2e-16 ***
## educated1          -5.469e-01  7.762e-02  -7.046 1.84e-12 ***
## KIDSDRIV            4.418e-01  6.038e-02   7.318 2.52e-13 ***
## revokedYes          7.863e-01  9.258e-02   8.494  < 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: 6613.0  on 5712  degrees of freedom
## Residual deviance: 5447.3  on 5703  degrees of freedom
## AIC: 5467.3
##
## Number of Fisher Scoring iterations: 5
```

Since there are a few other factors to consider, we added the predictors of ‘mstatus,’ factorized ‘car_use,’ ‘KIDSDRIV,’ and other important considerations to us when estimating the risk of a driver getting into an accident. This model has a lower AIC and should be better at predicting the target because it has a better fit. Interesting the addition of the factorized ‘sex’ predictor seems to have a positive effect on driving (lowering the chance of accident). This may be due to the sample sizes and how the counts are created, but it does not agree with the theory that females are worse drivers, in implies the opposite.

From here we begin the multiple linear regression modeling. In this first model, we throw everything we have at it to see what works. This model will give us a good indication of what could be useful to us and although it is not likely a good predictive model, we will try to find the best predictors possible.

```
model4 <- lm(target_amt ~ ., train)
summary(model4)
```

```
##
## Call:
## lm(formula = target_amt ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.16361 -0.00472  0.00052  0.00713  0.15233
##
## Coefficients: (2 not defined because of singularities)
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -5.890e-01  1.112e-02 -52.981   <2e-16 ***
## TARGET_FLAG                      2.110e+00  2.952e-03 714.825   <2e-16 ***
## TARGET_AMT                       2.538e-05  2.834e-07  89.561   <2e-16 ***
## KIDSDRIV                        -1.115e-03  2.188e-03  -0.510   0.6102
## AGE                              3.253e-05  1.400e-04   0.232   0.8163
## HOMEKIDS                         2.005e-04  1.276e-03   0.157   0.8752
## YOJ                             -4.639e-04  2.924e-04  -1.586   0.1127
## TRAVTIME                         2.985e-05  6.247e-05   0.478   0.6327
## TIF                             -7.299e-05  2.354e-04  -0.310   0.7565
## CLM_FREQ                        -5.426e-03  2.161e-03  -2.511   0.0121 *
## MVR_PTS                          4.976e-04  7.063e-04   0.704   0.4812
## income.values                   -2.435e-08  3.468e-08  -0.702   0.4826
## home.values                     -9.550e-09  1.135e-08  -0.841   0.4001
## bluebook.values                  1.247e-07  1.661e-07   0.750   0.4530
## oldclaim.values                  5.064e-07  3.252e-07   1.557   0.1195
## parent1Yes                      -4.423e-05  3.880e-03  -0.011   0.9909
## mstatusz_No                     -4.518e-05  2.811e-03  -0.016   0.9872
## sexz_F                           2.346e-04  3.573e-03   0.066   0.9477
## educationBachelors              -3.203e-03  3.962e-03  -0.808   0.4188
## educationMasters                -3.571e-04  5.732e-03  -0.062   0.9503
## educationPhD                    -5.652e-04  6.885e-03  -0.082   0.9346
## educationz_High School           2.997e-03  3.306e-03   0.906   0.3647
## jobClerical                     -7.368e-03  6.615e-03  -1.114   0.2654
## jobDoctor                        2.577e-03  7.818e-03   0.330   0.7417
## jobHome Maker                   -1.657e-02  7.070e-03  -2.344   0.0191 *
## jobLawyer                       -6.127e-03  5.702e-03  -1.074   0.2827
## jobManager                      -1.398e-03  5.581e-03  -0.251   0.8022
## jobProfessional                 -2.482e-03  5.923e-03  -0.419   0.6752
## jobStudent                      -7.713e-03  7.278e-03  -1.060   0.2893
## jobz_Blue Collar                -9.370e-03  6.238e-03  -1.502   0.1332
## car_usePrivate                   1.575e-03  3.182e-03   0.495   0.6206
## car_age                          1.995e-04  2.468e-04   0.808   0.4189
## car_typePanel Truck              1.576e-03  5.386e-03   0.293   0.7698
## car_typePickup                   1.829e-03  3.311e-03   0.552   0.5807
## car_typeSports Car               2.875e-04  4.172e-03   0.069   0.9451
## car_typeVan                      4.292e-04  4.067e-03   0.106   0.9159
## car_typez_SUV                    2.666e-03  3.480e-03   0.766   0.4437
## red_caryes                       3.818e-03  2.895e-03   1.319   0.1873
## revokedYes                      -1.898e-05  3.417e-03  -0.006   0.9956
## urbanicityz_Highly Rural/ Rural  5.741e-04  2.821e-03   0.203   0.8388
## city1                                   NA         NA      NA       NA
## young1                           5.431e-03  1.033e-02   0.526   0.5989
## clean_rec1                      -1.693e-03  2.809e-03  -0.603   0.5467
## previous_accident1               7.992e-03  5.323e-03   1.501   0.1333
## educated1                               NA         NA      NA       NA
## avg_claim                       -6.315e-07  4.650e-07  -1.358   0.1745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0733 on 5669 degrees of freedom
## Multiple R-squared:  0.9947, Adjusted R-squared:  0.9946
## F-statistic: 2.461e+04 on 43 and 5669 DF,  p-value: < 2.2e-16
```

Since the many of the model's predictors are transformed the estimate of those predictors will not be a useful indicator of anything. However, there are two values that are signifcant at the 0.05 alpha level. These coefficients are 'CLM_FREQ' and 'jobHome_Maker.' These indicate that their presence is a significant addition to the model to help determine the amount of a claim. However, from our exploration and preparation of these points, these particular predictor confer no numeric value to estimate the claim amount by. This renders them useless for our purposes but significantly so. We will rearrange with a select few numeric variables next to see if this improves.

```
model5 <- lm(target_amt ~ income.values +
               home.values + bluebook.values +
               oldclaim.values + avg_claim,
             train)
summary(model5)
```

```
##
```

```
## Call:
## lm(formula = target_amt ~ income.values + home.values + bluebook.values +
##     oldclaim.values + avg_claim, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.617 -0.655 -0.472  1.050  2.803
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.676e-01  3.014e-02   8.877  < 2e-16 ***
## income.values   -7.328e-07  3.502e-07  -2.092 0.036450 *
## home.values     -1.224e-06  1.218e-07 -10.055  < 2e-16 ***
## bluebook.values -5.320e-06  1.678e-06  -3.170 0.001532 **
## oldclaim.values  1.224e-05  3.347e-06   3.656 0.000258 ***
## avg_claim        2.875e-06  4.850e-06   0.593 0.553302
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9728 on 5707 degrees of freedom
## Multiple R-squared:  0.05508,    Adjusted R-squared:  0.05425
## F-statistic: 66.53 on 5 and 5707 DF,  p-value: < 2.2e-16
```

These continuous numeric predictors are significant. The ‘avg_claim’ predictor is present soley to assist in the control of unrealistic estimates. It is not a signifcant predictor. However, the others are. Additionaly, ‘bluebook.values’ a near natural indicator of the value of a car is a ironically, a good indication of the claim value. This model is much better since it does not pretend to have historical data to build on. However, as expected, the realistic ability of this model to predict is impractical with a coefficient of determination around .05. Such a model is worse at predicting true values on the surface because the data’s transformation must be reversed to see logical values. Unfortunately, this model does not bode well with such a reversal as it repeatedly fails conditonal assumptions of linear regression.

```
model6 <- lm(target_amt ~ . -TARGET_AMT -TARGET_FLAG, train)
pm <- stepAIC(model6, trace = F, direction = "both")
summary(pm)
```

```
##
## Call:
## lm(formula = target_amt ~ KIDSDRIV + HOMEKIDS + TRAVTIME + TIF +
##     CLM_FREQ + MVR_PTS + income.values + home.values + bluebook.values +
##     oldclaim.values + parent1 + mstatus + education + job + car_use +
##     car_type + revoked + urbanicity + young + avg_claim, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9709 -0.6370 -0.2437  0.6152  2.8975
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -5.839e-02  1.024e-01  -0.570 0.568598
## KIDSDRIV                         1.212e-01  2.589e-02   4.680 2.94e-06 ***
## HOMEKIDS                         2.363e-02  1.400e-02   1.687 0.091643 .
## TRAVTIME                         4.161e-03  7.497e-04   5.550 2.99e-08 ***
## TIF                           -1.612e-02  2.822e-03  -5.711 1.18e-08 ***
## CLM_FREQ                       9.387e-02  1.388e-02   6.764 1.48e-11 ***
## MVR_PTS                        4.286e-02  6.062e-03   7.070 1.74e-12 ***
## income.values                 -7.632e-07  4.154e-07  -1.837 0.066250 .
## home.values                   -4.547e-07  1.360e-07  -3.343 0.000835 ***
## bluebook.values               -5.873e-06  1.803e-06  -3.257 0.001133 **
## oldclaim.values               -1.847e-05  3.727e-06  -4.957 7.39e-07 ***
## parent1Yes                     1.443e-01  4.650e-02   3.104 0.001920 **
## mstatusz_No                    1.861e-01  3.358e-02   5.540 3.15e-08 ***
## educationBachelors            -1.306e-01  4.519e-02  -2.890 0.003863 **
## educationMasters              -9.238e-02  6.260e-02  -1.476 0.140030
## educationPhD                  -1.123e-01  7.774e-02  -1.445 0.148627
## educationz_High School         5.226e-03  3.969e-02   0.132 0.895246
## jobClerical                    1.916e-01  7.951e-02   2.410 0.015982 *
## jobDoctor                     -5.598e-02  9.397e-02  -0.596 0.551349
## jobHome Maker                  1.420e-01  8.276e-02   1.716 0.086256 .
## jobLawyer                      5.229e-02  6.852e-02   0.763 0.445413
## jobManager                    -1.522e-01  6.708e-02  -2.268 0.023346 *
## jobProfessional                8.451e-02  7.125e-02   1.186 0.235625
## jobStudent                     1.792e-01  8.604e-02   2.083 0.037315 *
## jobz_Blue Collar               1.803e-01  7.503e-02   2.403 0.016312 *
## car_usePrivate                -2.224e-01  3.818e-02  -5.825 6.03e-09 ***
## car_typePanel Truck            2.003e-01  6.092e-02   3.288 0.001015 **
## car_typePickup                 1.682e-01  3.978e-02   4.228 2.40e-05 ***
## car_typeSports Car             3.131e-01  4.246e-02   7.374 1.90e-13 ***
## car_typeVan                    1.931e-01  4.732e-02   4.082 4.53e-05 ***
## car_typez_SUV                  2.248e-01  3.274e-02   6.867 7.24e-12 ***
## revokedYes                     3.823e-01  4.039e-02   9.465  < 2e-16 ***
## urbanicityz_Highly Rural/ Rural -6.588e-01  3.253e-02 -20.254  < 2e-16 ***
## young1                         3.380e-01  1.197e-01   2.824 0.004759 **
## avg_claim                      1.749e-05  4.780e-06   3.659 0.000255 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8824 on 5678 degrees of freedom
## Multiple R-squared:  0.2264, Adjusted R-squared:  0.2218
## F-statistic: 48.89 on 34 and 5678 DF,  p-value: < 2.2e-16
```

Lastly, we took a new approach. Our kitchen sink model is the model that contain the most realistic expectations and so we optimize the model to perform a a stepAIC in both directions and ideally improve accuracy. With this, we attempt to create the most realistic model with a higher AUC regardless of coefficient significance. This method should produce the highest coefficient of determination without compromising the data's integrity. We will review these results in the model selection process.

---

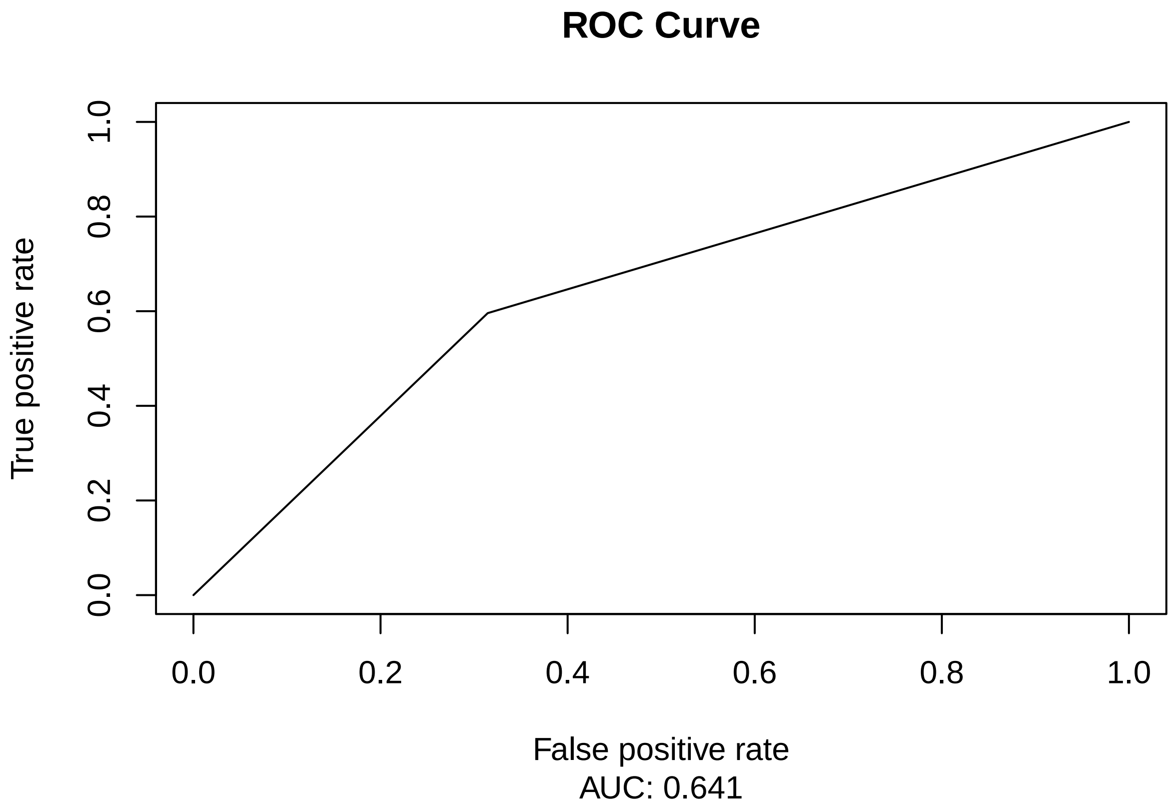
## Model Selection

To select the best model we will run some statistics on each. We will utilize a prediction model statistics function called 'modstat' rather than repeating the same estimates by hand. This will put all models on the same level of focus. It includes and confusion matrix, predicted probability and amount values, the AUC, F1 scores, a ROC plot for each model. The process is documented in the function below. We start with the binary logistic classification models and then move onto the multiple linear regression.

```r
# Calculate predicted values
# Classifier Model
mod1.pred <- predict.glm(model1, test)
mod2.pred <- predict.glm(model2, test)
mod3.pred <- predict.glm(model3, test)
# Regression Model
mod4.pred <- predict(model4, test, interval = "prediction")
mod5.pred <- predict(model5, test, interval = "prediction")
mod6.pred <- predict(model6, test, interval = "prediction")
```

```r
modstat <- function(model, test, target = "TARGET_FLAG", threshold = 0.5){
  test$new <- ifelse(predict.glm(model, test, "response") >= threshold, 1, 0)
  cm <- confusionMatrix(factor(test$new), factor(test[[target]]), "1")
  df <- data.frame(obs = test$TARGET_FLAG, predicted = test$new, probs = predict(model, test))
  Pscores <- prediction(df$probs, df$obs)
  AUC <- performance(Pscores, measure = "auc")@y.values[[1]]
  pscores <- performance(Pscores, "tpr", "fpr")
  plot(pscores,main="ROC Curve", sub = paste0("AUC: ", round(AUC, 3)))
  results <- paste(cat("F1 = ", cm$byClass[7], " "), cm)
  return(results)
}
```

```r
modstat(model1, test)
```

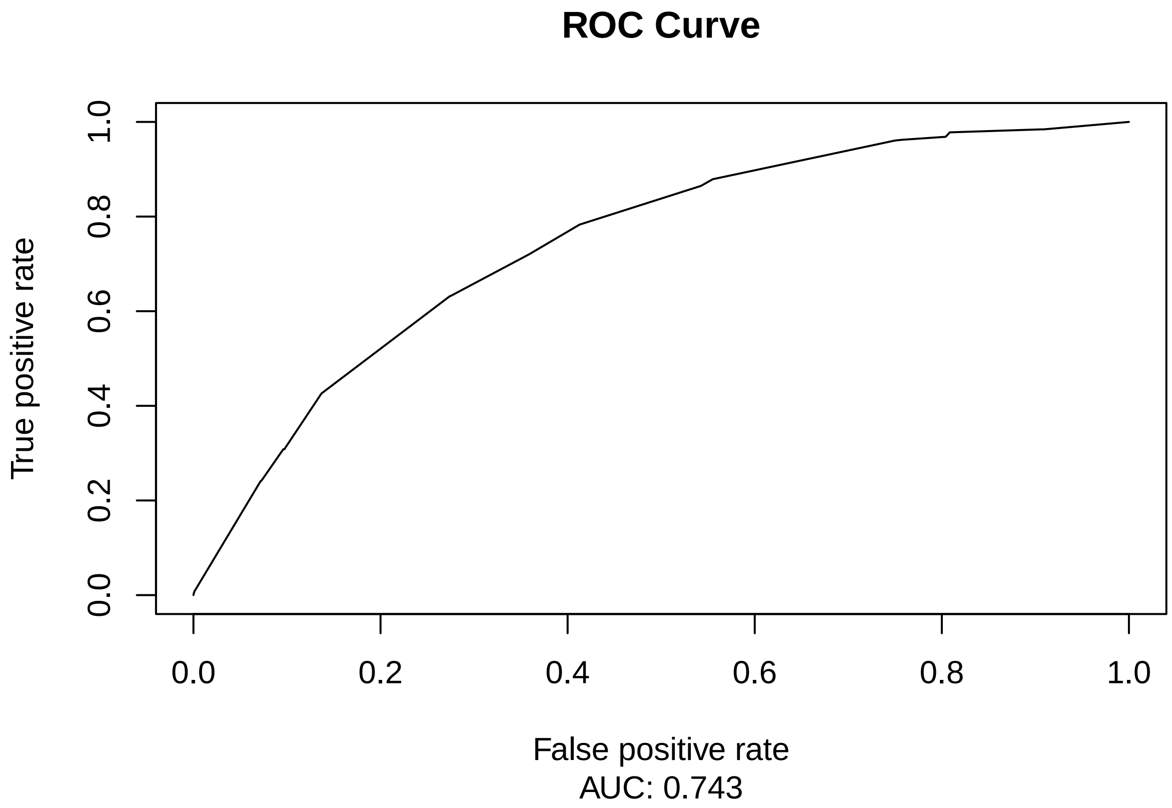


```
## F1 =  NA
```

```
## [1] " 1"
## [2] " c(1812, 0, 636, 0)"
## [3] " c(Accuracy = 0.740196078431373, Kappa = 0, AccuracyLower = 0.722339505019352, AccuracyUpper = 0
## [4] " c(Sensitivity = 0, Specificity = 1, ‘Pos Pred Value‘ = NaN, ‘Neg Pred Value‘ = 0.74019607843137
## [5] " sens_spec"
## [6] " list()"
```

This model was our dummy run. It is not expected to perform as well as the intentionally hand-picked models but it did give us some baseline information to go on. Perhaps the first thing we notice in the nonapplicable F1 score because this model correctly classified 1812 as true positives while 636 were false positive. None were present in the other categories due to the misrepresentation of the minority class. This model is not particularly useful for prediction but it is a great start to see how influential the majority class is.

```
modstat(model2, test)
```

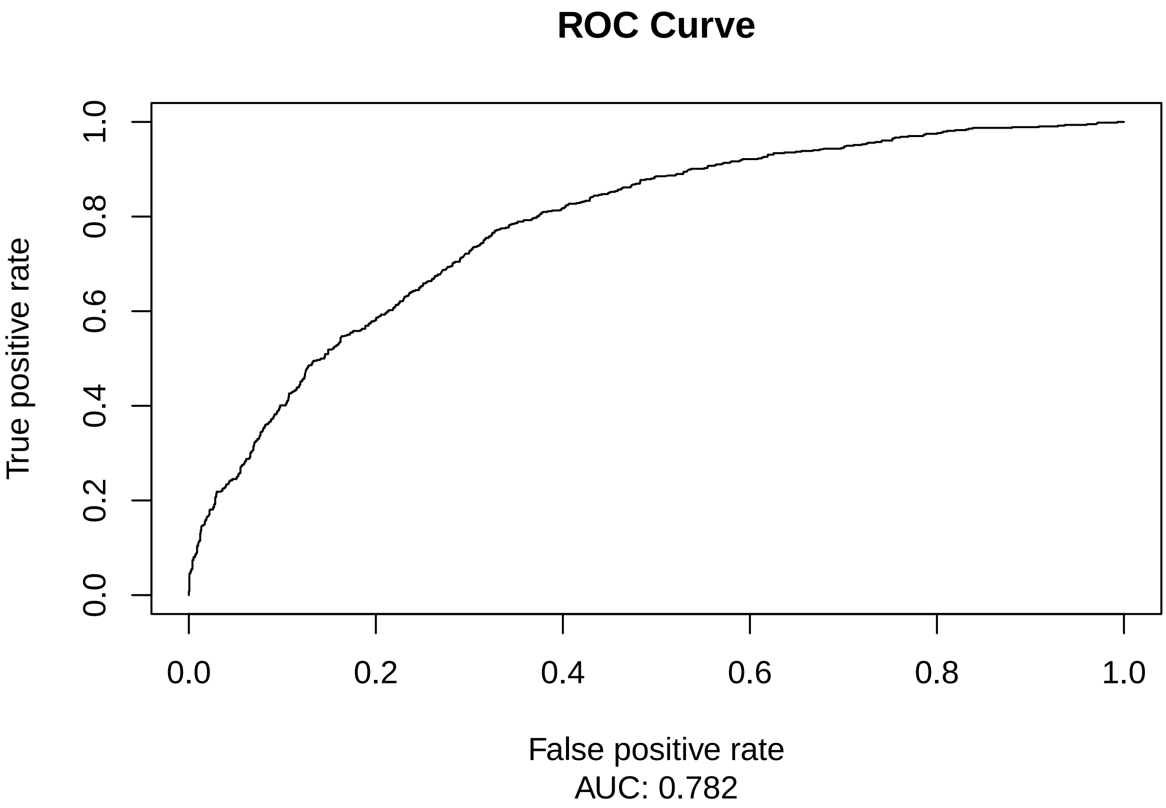


```
## F1 =  0.3329706
```

```
## [1] " 1"
## [2] " c(1682, 130, 483, 153)"
## [3] " c(Accuracy = 0.749591503267974, Kappa = 0.205908126849465, AccuracyLower = 0.731932472926179, 
## [4] " c(Sensitivity = 0.240566037735849, Specificity = 0.928256070640177, ‘Pos Pred Value‘ = 0.54063
## [5] " sens_spec"
## [6] " list()"
```

In this model we consider a few more predictors and ended with a reasonable espectation of sensitivity (24%) specificity (93%) and accuracy at 74%. This is an improvement on the previous model and garners an AUC of 0.743. Our trouble lies in the F1 score where we have relatively poor precision and poor recall when predicting the probability of an accident. Let's see how we improved upon this.

```
modstat(model3, test)
```

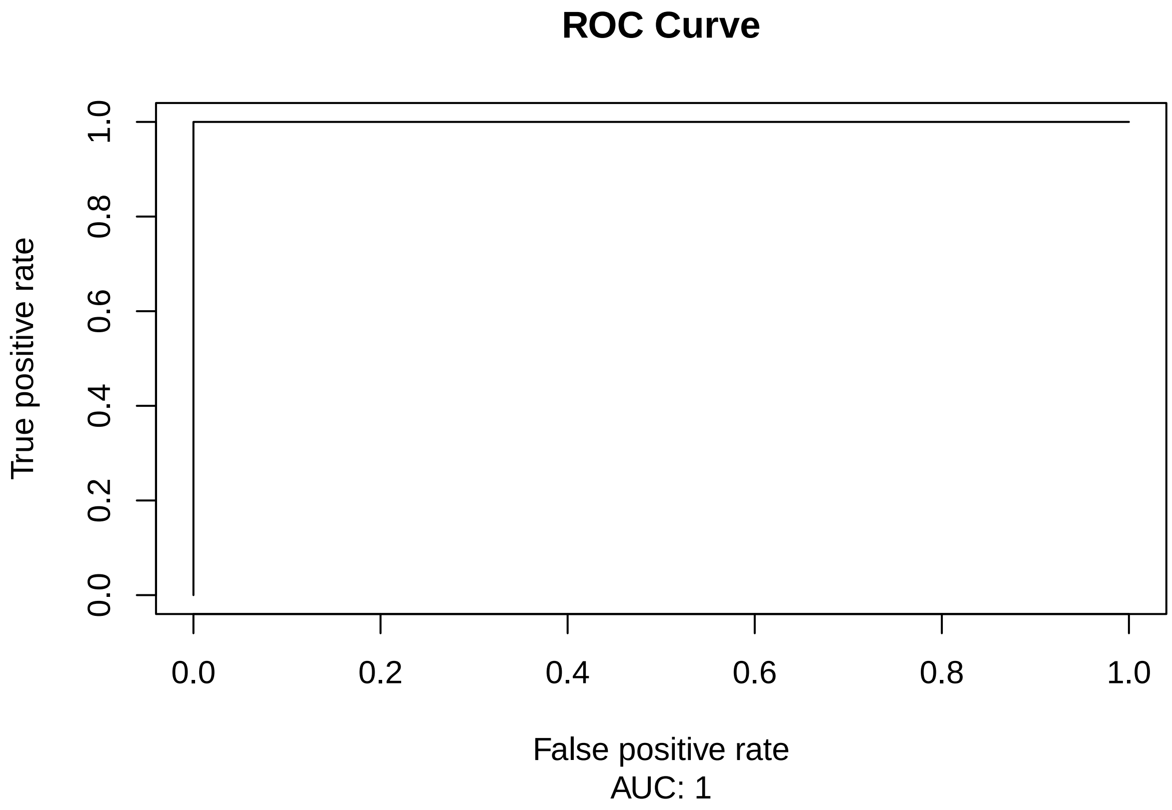


```
## F1 =  0.4219235
```

```
## [1] " 1"
## [2] " c(1685, 127, 432, 204)"
## [3] " c(Accuracy = 0.771650326797386, Kappa = 0.296864016968591, AccuracyLower = 0.754498018991632, A
## [4] " c(Sensitivity = 0.320754716981132, Specificity = 0.929911699779249, ‘Pos Pred Value‘ = 0.61631
## [5] " sens_spec"
## [6] " list()"
```

For our final binary classification model, this is an improvement. the F1 score did increase to 0.42 with a better accuracy of about 77%. It appears addint factors beyond just those that many would consider risky, determines the probability of an accident better than the risk behavior alone. Our AUC for this model is 0.782.

```
modstat(model4, test)
```

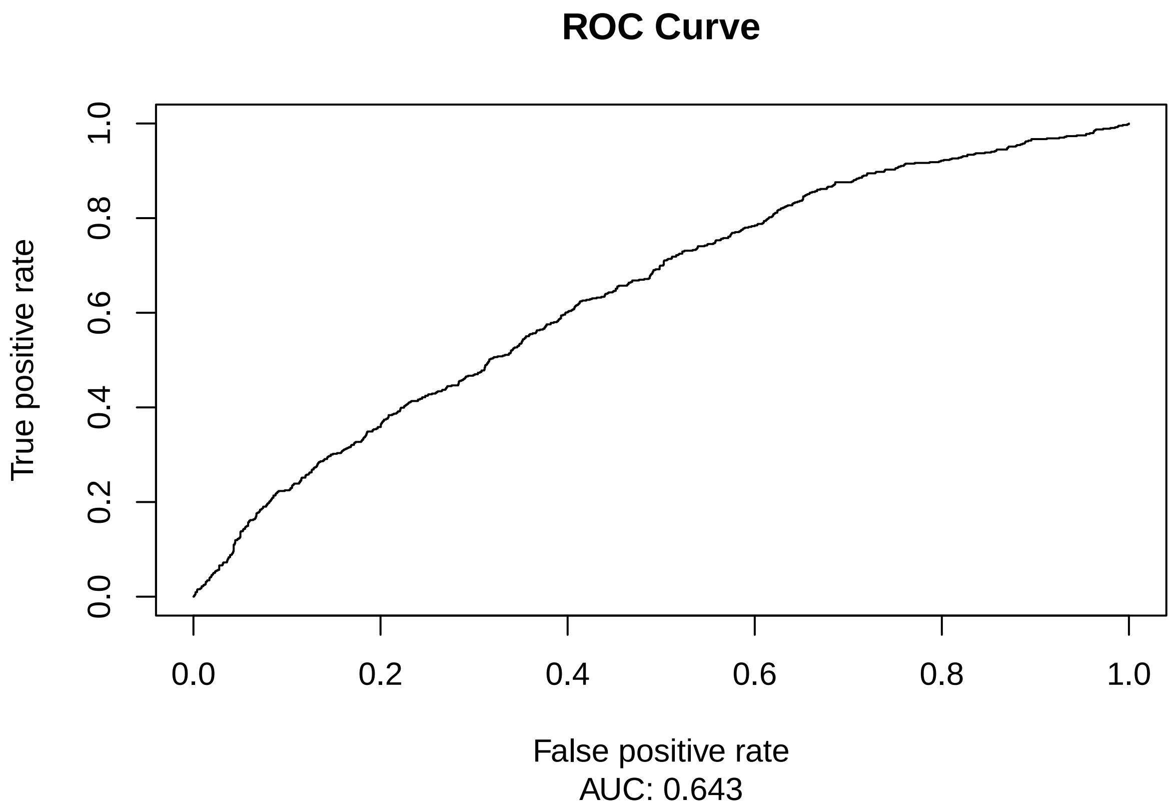


```
## F1 =  1
```

```
## [1] " 1"
## [2] " c(1812, 0, 0, 636)"
## [3] " c(Accuracy = 1, Kappa = 1, AccuracyLower = 0.998494239594653, AccuracyUpper = 1, AccuracyNull =
## [4] " c(Sensitivity = 1, Specificity = 1, ‘Pos Pred Value‘ = 1, ‘Neg Pred Value‘ = 1, Precision = 1,
## [5] " sens_spec"
## [6] " list()"
```

Here, we purposefully made a prediction from a rank-deficient fit that contained misrepresentative variables. For example, our target contained values that were between -2 and 2 when they should have been on the scale of 0-100,000 as dollars. For this reason, we made this model to show how we could improve the model's accuracy to a perfect value by including many transformed values in the model itself. However, the reality is that this model is completely useless because to gather any information from it that is real, the data must be reversed from its transformation and reformed to represent a true value of the claim amounts. Rather than go through this effort, we let this one inform our next model.

```
modstat(model5, test)
```

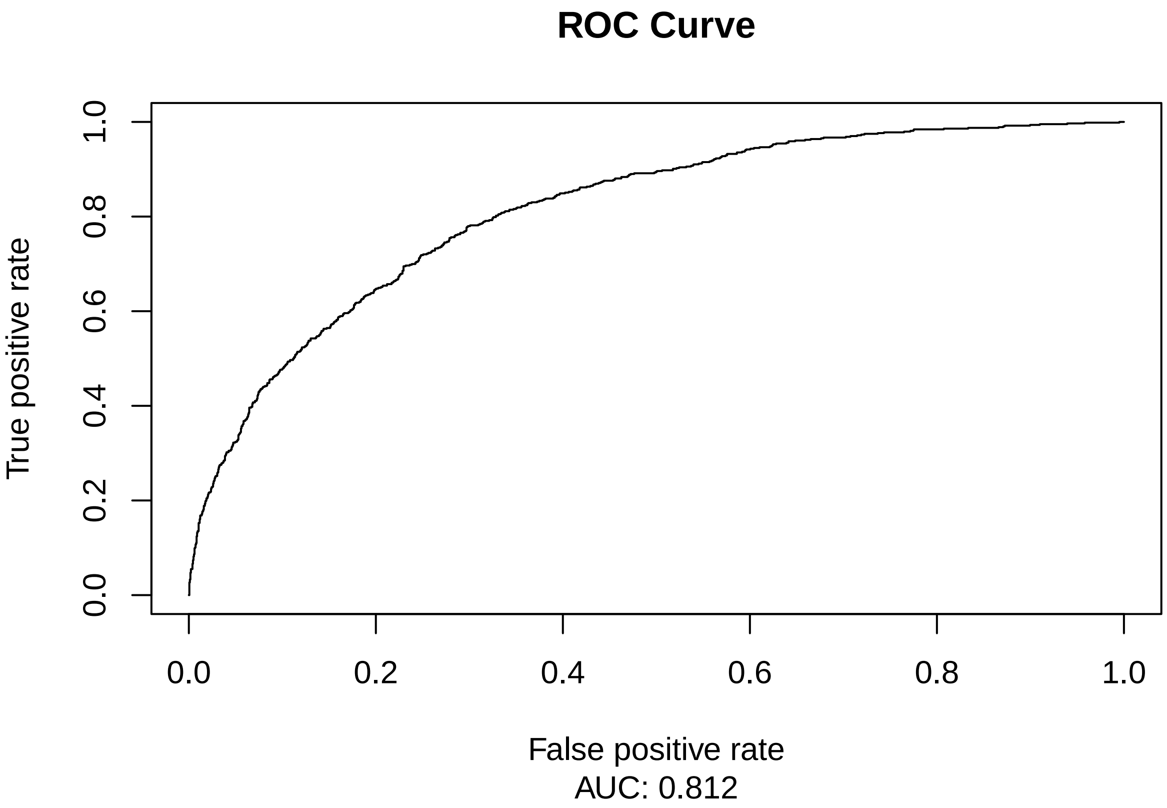


```
## F1 =  0.09014085
```

```
## [1] " 1"
## [2] " c(1770, 42, 604, 32)"
## [3] " c(Accuracy = 0.736111111111111, Kappa = 0.0380449064206253, AccuracyLower = 0.718171898328195,
## [4] " c(Sensitivity = 0.050314465408805, Specificity = 0.97682119205298, ‘Pos Pred Value‘ = 0.4324324
## [5] " sens_spec"
## [6] " list()"
```

Here again we are pulling from a tough data set. The ability of this model to predict true positives is great and with an accuracy of 73% we are gaining traction at predicting with it. Our AUC was also 0.643. However, this model incorrectly assumes almost all drivers are going to get into accidents. this comes from the distribution and sampling of the data. The majority class greatly outnumbers the minoirity causing them to be misrepresented. This should be improved upon, which mean improving our F1 score from 0.090 to something closer to 1.

```
modstat(model6, test)
```



```
## F1 =  0.4838057
```

```
## [1] " 1"
## [2] " c(1699, 113, 397, 239)"
## [3] " c(Accuracy = 0.791666666666667, Kappa = 0.36653677545056, AccuracyLower = 0.775030346732191, A
## [4] " c(Sensitivity = 0.375786163522013, Specificity = 0.937637969094923, ‘Pos Pred Value‘ = 0.67897
## [5] " sens_spec"
## [6] " list()"
```

Importantly, this model acheives the goal we set with our last model. Our F1 score improved to 0.484 which, although it is still not great, is better than 0.090. Additionaly this model offers the best accuracy of the multiple linear regression models with about 79%. Our AUC is also 0.812 which serves us better in prediction than the other models. However, it should be noted that the ability of this model to make accurate predictions is still poor and that if data were to be run through this model, it would need a reversal of its transformation to make logical sense and be used in practice.

## Conclusion

Given insurance data of the same types and identities, the third and sixth models should be used to export the most favorable data sets for making realistic predictions. The third runs a binary logistic classifier and the sixth the multiple linear regression model. However, it should be noted that the use of the sixth model to produce results for informing insurance practices is ill-advised with results as is. These data should be reformed into useful figures by reversing the transformation of the target, the claim amount.

Additionally, reliance on the third model is minimized as much as possible but it should be known that better data collection may be necessary to produce more realistic results. The current rate of accuracy is only about 77% without more gaussian data. Prediction of the claim amount after predicting whether an individual will get into an accident is also tedious and tempermental due to the data's failure to comply with the conditional assumptions of linear regression. The best accuracy we could produce with this claim prediction model is about 79% but the fit of the model judging by the coefficient of determination is only about 0.22.

Predicting the claim amount is more difficult than determining whether an individual will get into an accident. Future models should consider how to limit the influence of the majority class on the surpression of the minority class to make better predictions since the minority class is where the accidents and claim values larger than zero are. We also recommend a better collection method in which people can be more honest and open about their responses. If this data did come from a valid insurance provider, which it appear to have been, then the models will be skewed based on the data, as is demonstrated.