Data 621 Assignment 4

Bridget Boakye, Hazal Gunduz and Farhana Akther

2024-04-14

Contents

Overview:

In this assignment, we will explore, analyze and model a data set containing approximately 8000 records, each representing a customer at an auto insurance company. Each record has two response variables. The first response variable, TARGET_FLAG, is binary. A "1" indicates that the customer was in a car crash while 0 indicates that they were not. The second response variable is TARGET_AMT. This value is 0 if the customer did not crash their car. However, if they did crash their car, this number will be a value greater than 0.

The objective is to build multiple linear regression and binary logistic regression models on the training data to predict whether a customer will crash their car and to predict the cost in the case of crash. We will only use the variables given to us (or variables that we derive from the variables provided).

Loading libraries:

```
library(stringr)
library(ggcorrplot)
library(dplyr)
library(GGally)
library(ggplot2)
library(readr)
library(reshape2)
library(purrr)
library(tidyr)
library(corrplot)
library(MASS)
library(e1071)
library(ROCR)
library(pROC)
library(car)
library(glmnet)
library(caTools)
library(leaps)
library(caret)
library(ROSE)
library(mice)
```

1. DATA EXPLORATION:

In this first step, we're going to look closely at the training data set to understand it better before we start preparing or modeling.

Loading Data:

The datasets (training and evaluation) has been uploaded to a GitHub repository, from which it has been loaded into the markdown using the code chunk provided below. The rationale behind uploading it to GitHub is to maintain the reproducibility of the work.

```r
set.seed(2024)

insurance_training <- read.csv("https://raw.githubusercontent.com/breboa/Data621/main/insurance_trainin
insurance_evaluation <- read.csv("https://raw.githubusercontent.com/breboa/Data621/main/insurance-evalu
```

Data Dimension:

```r
head(insurance_training)
```

```
##   INDEX TARGET_FLAG TARGET_AMT KIDSDRIV AGE HOMEKIDS YOJ   INCOME PARENT1
## 1     1           0          0        0  60        0  11  $67,349      No
## 2     2           0          0        0  43        0  11  $91,449      No
## 3     4           0          0        0  35        1  10  $16,039      No
## 4     5           0          0        0  51        0  14                No
## 5     6           0          0        0  50        0  NA $114,986      No
## 6     7           1       2946        0  34        1  12 $125,301     Yes
##   HOME_VAL MSTATUS SEX     EDUCATION           JOB TRAVTIME    CAR_USE BLUEBOOK
## 1       $0    z_No   M           PhD  Professional       14    Private  $14,230
## 2 $257,252    z_No   M z_High School z_Blue Collar       22 Commercial  $14,940
## 3 $124,191     Yes z_F z_High School      Clerical        5    Private   $4,010
## 4 $306,251     Yes   M  <High School z_Blue Collar       32    Private  $15,440
## 5 $243,925     Yes z_F           PhD        Doctor       36    Private  $18,000
## 6       $0    z_No z_F     Bachelors z_Blue Collar       46 Commercial  $17,430
##   TIF   CAR_TYPE RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS CAR_AGE
## 1  11    Minivan     yes   $4,461        2      No       3      18
## 2   1    Minivan     yes       $0        0      No       0       1
## 3   4      z_SUV      no  $38,690        2      No       3      10
## 4   7    Minivan     yes       $0        0      No       0       6
## 5   1      z_SUV      no  $19,217        2     Yes       3      17
## 6   1 Sports Car      no       $0        0      No       0       7
##            URBANICITY
## 1 Highly Urban/ Urban
## 2 Highly Urban/ Urban
## 3 Highly Urban/ Urban
## 4 Highly Urban/ Urban
## 5 Highly Urban/ Urban
## 6 Highly Urban/ Urban
```

```r
dim(insurance_training)
```

```
## [1] 8161   26
```

Remove index column

```r
insurance_training <- subset(insurance_training, select = -INDEX)
head(insurance_training)
```

```
##   TARGET_FLAG TARGET_AMT KIDSDRIV AGE HOMEKIDS YOJ   INCOME PARENT1 HOME_VAL
## 1           0          0        0  60        0  11  $67,349      No       $0
## 2           0          0        0  43        0  11  $91,449      No $257,252
## 3           0          0        0  35        1  10  $16,039      No $124,191
## 4           0          0        0  51        0  14                No $306,251
## 5           0          0        0  50        0  NA $114,986      No $243,925
## 6           1       2946        0  34        1  12 $125,301     Yes       $0
##   MSTATUS SEX     EDUCATION           JOB TRAVTIME    CAR_USE BLUEBOOK TIF
## 1    z_No   M           PhD  Professional       14    Private  $14,230  11
```

```
## 2    z_No   M z_High School z_Blue Collar       22 Commercial  $14,940   1
## 3     Yes z_F z_High School      Clerical        5    Private   $4,010   4
## 4     Yes   M  <High School z_Blue Collar       32    Private  $15,440   7
## 5     Yes z_F           PhD        Doctor       36    Private  $18,000   1
## 6    z_No z_F     Bachelors z_Blue Collar       46 Commercial  $17,430   1
##     CAR_TYPE RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS CAR_AGE
## 1    Minivan     yes   $4,461        2      No       3      18
## 2    Minivan     yes       $0        0      No       0       1
## 3      z_SUV      no  $38,690        2      No       3      10
## 4    Minivan     yes       $0        0      No       0       6
## 5      z_SUV      no  $19,217        2     Yes       3      17
## 6 Sports Car      no       $0        0      No       0       7
##          URBANICITY
## 1 Highly Urban/ Urban
## 2 Highly Urban/ Urban
## 3 Highly Urban/ Urban
## 4 Highly Urban/ Urban
## 5 Highly Urban/ Urban
## 6 Highly Urban/ Urban
```

Descriptive Summary Statistics:

```r
summary(insurance_training)
```

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

```
##  Length:8161
##  Class :character
##  Mode  :character
##
##
##
##
```

The summary confirms the following information about the predictors, which is also stated in their description:
There are 13 variables that contain discrete varibles (class: characters) while the remaining are continuous. Some variables that are categorized as discrete (eg. INCOME, HOME_VAL, OLDCLAIM, BLUEBOOK), however, are incorrect given the continuous values shown in the dataset head and will need to be categorized to the correct data type.
The continuous variables AGE, YOJ, and CAR_AGE containing missing variables. CAR_AGE also has a minimum value of -3 which does not make sense.
Some of the character values may also contain missing data but it isn't vible from summary.
Some of the character and numeric values have various prefixes that need to be cleaned.
Target variable, TARGET_FLAG, is characterized as continuous although it should be a factor (given the description of the variables in the assignment), as 0 and 1.

Missing values for numerical data

The following code calculates the percent of missing values across AGE, YOJ, and CAR_AGE.

```
insurance_training %>%
  summarize(across(everything(), ~sum(is.na(.)) / n()))
```

```
##   TARGET_FLAG TARGET_AMT KIDSDRIV         AGE HOMEKIDS        YOJ INCOME
## 1           0          0        0 0.000735204        0 0.05563044      0
##   PARENT1 HOME_VAL MSTATUS SEX EDUCATION JOB TRAVTIME CAR_USE BLUEBOOK TIF
## 1       0        0       0   0         0   0        0       0        0   0
##   CAR_TYPE RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS    CAR_AGE URBANICITY
## 1        0       0        0        0       0       0 0.06249234          0
```

It is clear that the missing values are only a low/moderate percentage of their respective variables:
Missing values for AGE: 7.35 %
YOJ: 5.56%
and CAR_AGE: 6.25%

Check missing values for categorical variables

```
insurance_training %>%
  select_if(~is.character(.x) | is.factor(.x)) %>%
  map_df(~sum(is.na(.)), .id = "Variable") %>%
  t()
```

```
##             [,1]
## INCOME         0
## PARENT1        0
```

```
## HOME_VAL      0
## MSTATUS       0
## SEX           0
## EDUCATION     0
## JOB           0
## CAR_USE       0
## BLUEBOOK      0
## CAR_TYPE      0
## RED_CAR       0
## OLDCLAIM      0
## REVOKED       0
## URBANICITY    0
```

When we check the missing values for our categorical variables, we see that there are no missing values. However, when we look in the training dataset, we see that there are some blanks for the JOB variable. So we will correct that in our data preparation.

2. DATA PREPRATION - LOGISTIC REGRESSION:

In our data preparation, we seek to address a number of issues that will prevent us for creating statistically sound models. We write functions to:

a. Fix formatting
b. Correct data types
c. Impute missing values using median and **Unspecified**
d. Skewness

a. Fix formatting - remove $ and z prefix

The presence of currency ($) notation for some columns (eg. INCOME, HOME_VAL, BLUVE_BOOK, AND OLDCLAIM) may disrupt our analysis and model building, necessitating the proper reformatting of those values

```r
strip_dollars <- function(x){
  x <- as.character(x)
  x <- gsub(",", "", x)
  x <- gsub("\\$", "", x)
  as.numeric(x)
}


fix_formatting <- function(training_df) {
  training_df <- training_df %>%
    mutate(across(c(INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM), strip_dollars))

  return(training_df)
}
```

```r
remove_value_prefixes <- function(training_df) {
  targeted_cols <- c("MSTATUS", "SEX", "EDUCATION", "JOB", "CAR_TYPE", "URBANICITY")
```

```r
  training_df <- training_df %>%
    mutate(across(all_of(targeted_cols), ~str_replace_all(.x, "^z_", "")))

  training_df$EDUCATION <- str_replace_all(training_df$EDUCATION, "<", "Below ")


   return(training_df)
}
```

b. Transform to numeric data types function

As discussed in the data exploration, INCOME, HOME_VAL, OLDCLAIM, and BLUEBOOK are categorized as discrete, character datatypes although their values are continuous. Here we can their datatype to numeric.

```r
transform_numeric <- function(training_df){
  training_df %>%
    mutate(across(c("INCOME", "HOME_VAL", "BLUEBOOK", "OLDCLAIM", "TARGET_AMT"),
                  ~as.numeric(as.character(.))))

   return(training_df)
}
```

c. Transform to factor data types function

```r
transform_to_factors <- function(training_df) {

  training_df$URBANICITY <- factor(ifelse(str_detect(training_df$URBANICITY, "Highly Urban"),
                                          "Urban",
                                          ifelse(str_detect(training_df$URBANICITY, "Highly Rural"),
                                                 "Rural",
                                                 NA_character_)))


  training_df$JOB <- ifelse(training_df$JOB == "" | is.na(training_df$JOB), "UNSPECIFIED", training_df$


  factor_vars <- c("TARGET_FLAG", "CAR_TYPE", "CAR_USE", "EDUCATION", "JOB",
                   "MSTATUS", "PARENT1", "RED_CAR", "REVOKED", "SEX", "URBANICITY")

  training_df[factor_vars] <- lapply(training_df[factor_vars], factor)

  return(training_df)

}
```

d. Correct values for CAR < 0

```r
correct_values <- function(training_df){
  training_df %>%
    rowwise() %>%
    mutate(CAR_AGE = ifelse(CAR_AGE < 0, NA, CAR_AGE))%>%
    ungroup()

   return(training_df)
}
```

e. Impute missing values

```r
impute_missing <- function(training_df) {
  training_df <- training_df %>%
    mutate(across(c(CAR_AGE, YOJ, AGE, INCOME, HOME_VAL), ~ifelse(is.na(.), median(., na.rm = TRUE), .)
  return(training_df)
}
```

F. We apply the processing steps by running both the training and evaluation datasets through the fuctions above

```r
clean_training <- insurance_training %>%
  fix_formatting()  %>%
  remove_value_prefixes()  %>%
  transform_numeric() %>%
  transform_to_factors () %>%
  correct_values() %>%
  impute_missing()
head(clean_training)
```

```
##   TARGET_FLAG TARGET_AMT KIDSDRIV AGE HOMEKIDS YOJ INCOME PARENT1 HOME_VAL
## 1           0          0        0  60        0  11  67349      No        0
## 2           0          0        0  43        0  11  91449      No   257252
## 3           0          0        0  35        1  10  16039      No   124191
## 4           0          0        0  51        0  14  54028      No   306251
## 5           0          0        0  50        0  11 114986      No   243925
## 6           1       2946        0  34        1  12 125301     Yes        0
##   MSTATUS SEX         EDUCATION          JOB TRAVTIME    CAR_USE BLUEBOOK TIF
## 1      No   M               PhD Professional       14    Private    14230  11
## 2      No   M       High School  Blue Collar       22 Commercial    14940   1
## 3     Yes   F       High School     Clerical        5    Private     4010   4
## 4     Yes   M Below High School  Blue Collar       32    Private    15440   7
## 5     Yes   F               PhD       Doctor       36    Private    18000   1
## 6      No   F         Bachelors  Blue Collar       46 Commercial    17430   1
##     CAR_TYPE RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS CAR_AGE URBANICITY
## 1    Minivan     yes     4461        2      No       3      18      Urban
## 2    Minivan     yes        0        0      No       0       1      Urban
## 3        SUV      no    38690        2      No       3      10      Urban
## 4    Minivan     yes        0        0      No       0       6      Urban
## 5        SUV      no    19217        2     Yes       3      17      Urban
```

```
## 6 Sports Car      no        0        0      No       0       7      Urban
```

```r
clean_evaluation <- insurance_evaluation %>%
  fix_formatting()  %>%
  remove_value_prefixes()  %>%
  transform_numeric() %>%
  transform_to_factors () %>%
  correct_values() %>%
  impute_missing()
head(clean_evaluation)
```

```
##   INDEX TARGET_FLAG TARGET_AMT KIDSDRIV AGE HOMEKIDS YOJ INCOME PARENT1
## 1     3        <NA>         NA        0  48        0  11  52881      No
## 2     9        <NA>         NA        1  40        1  11  50815     Yes
## 3    10        <NA>         NA        0  44        2  12  43486     Yes
## 4    18        <NA>         NA        0  35        2  11  21204     Yes
## 5    21        <NA>         NA        0  59        0  12  87460      No
## 6    30        <NA>         NA        0  46        0  14  51778      No
##   HOME_VAL MSTATUS SEX   EDUCATION          JOB TRAVTIME    CAR_USE BLUEBOOK
## 1        0      No   M   Bachelors      Manager       26    Private    21970
## 2        0      No   M High School      Manager       21    Private    18930
## 3        0      No   F High School  Blue Collar       30 Commercial     5900
## 4        0      No   M High School     Clerical       74    Private     9230
## 5        0      No   M High School      Manager       45    Private    15420
## 6   207519     Yes   M   Bachelors Professional        7 Commercial    25660
##   TIF    CAR_TYPE RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS CAR_AGE URBANICITY
## 1   1         Van     yes        0        0      No       2      10      Urban
## 2   6     Minivan      no     3295        1      No       2       1      Urban
## 3  10         SUV      no        0        0      No       0      10      Rural
## 4   6      Pickup      no        0        0     Yes       0       4      Rural
## 5   1     Minivan     yes    44857        2      No       4       1      Urban
## 6   1 Panel Truck      no     2119        1      No       2      12      Urban
```

```r
str(clean_training)
```

```
## 'data.frame':    8161 obs. of  25 variables:
##  $ TARGET_FLAG: Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 2 2 1 ...
##  $ TARGET_AMT : num  0 0 0 0 0 ...
##  $ KIDSDRIV   : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ AGE        : int  60 43 35 51 50 34 54 37 34 50 ...
##  $ HOMEKIDS   : int  0 0 1 0 0 1 0 2 0 0 ...
##  $ YOJ        : int  11 11 10 14 11 12 11 11 10 7 ...
##  $ INCOME     : num  67349 91449 16039 54028 114986 ...
##  $ PARENT1    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ HOME_VAL   : num  0 257252 124191 306251 243925 ...
##  $ MSTATUS    : Factor w/ 2 levels "No","Yes": 1 1 2 2 2 1 2 2 1 1 ...
##  $ SEX        : Factor w/ 2 levels "F","M": 2 2 1 2 1 1 1 2 1 2 ...
##  $ EDUCATION  : Factor w/ 5 levels "Bachelors","Below High School",..: 5 3 3 2 5 1 2 1 1 1 ...
##  $ JOB        : Factor w/ 9 levels "Blue Collar",..: 7 1 2 1 3 1 1 1 2 7 ...
##  $ TRAVTIME   : int  14 22 5 32 36 46 33 44 34 48 ...
##  $ CAR_USE    : Factor w/ 2 levels "Commercial","Private": 2 1 2 2 2 1 2 1 2 1 ...
##  $ BLUEBOOK   : num  14230 14940 4010 15440 18000 ...
##  $ TIF        : int  11 1 4 7 1 1 1 1 1 7 ...
```

```
##  $ CAR_TYPE   : Factor w/ 6 levels "Minivan","Panel Truck",..: 1 1 5 1 5 4 5 6 5 6 ...
##  $ RED_CAR    : Factor w/ 2 levels "no","yes": 2 2 1 2 1 1 1 2 1 1 ...
##  $ OLDCLAIM   : num  4461 0 38690 0 19217 ...
##  $ CLM_FREQ   : int  2 0 2 0 2 0 0 1 0 0 ...
##  $ REVOKED    : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 1 1 ...
##  $ MVR_PTS    : int  3 0 3 0 3 0 0 10 0 1 ...
##  $ CAR_AGE    : int  18 1 10 6 17 7 1 7 1 17 ...
##  $ URBANICITY : Factor w/ 2 levels "Rural","Urban": 2 2 2 2 2 2 2 2 2 1 ...
```

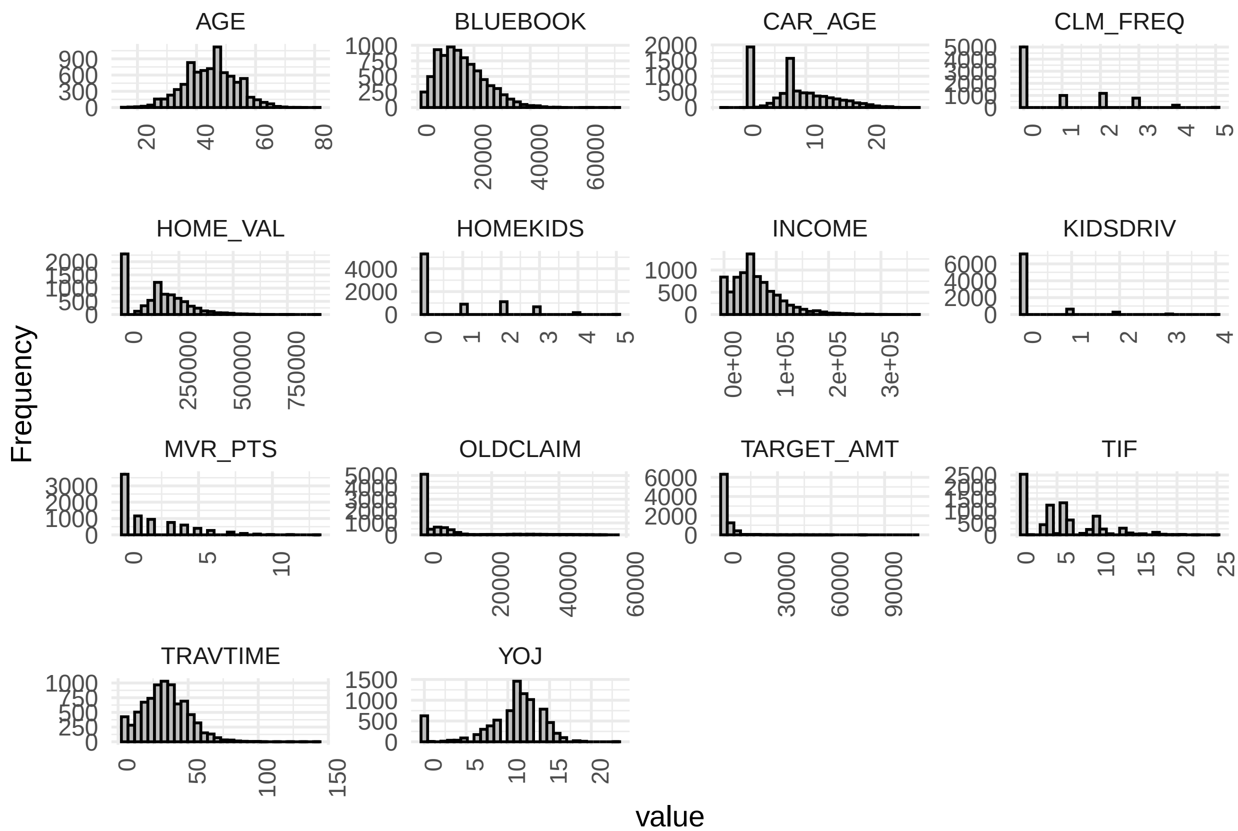
Check distribution of all the variables: with a fairly clean dataset, we examine the distribution of the data

Histogram

Histograms tell us how the data is distributed in the dataset (numeric fields).

```
data_long <- clean_training %>%
  select_if(is.numeric) %>%
  gather(key = "Variable", value = "Value")

ggplot(data_long, aes(x = Value)) +
  geom_histogram(bins = 30, fill = "gray", color = "black") +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(x = "value", y = "Frequency") +
 theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



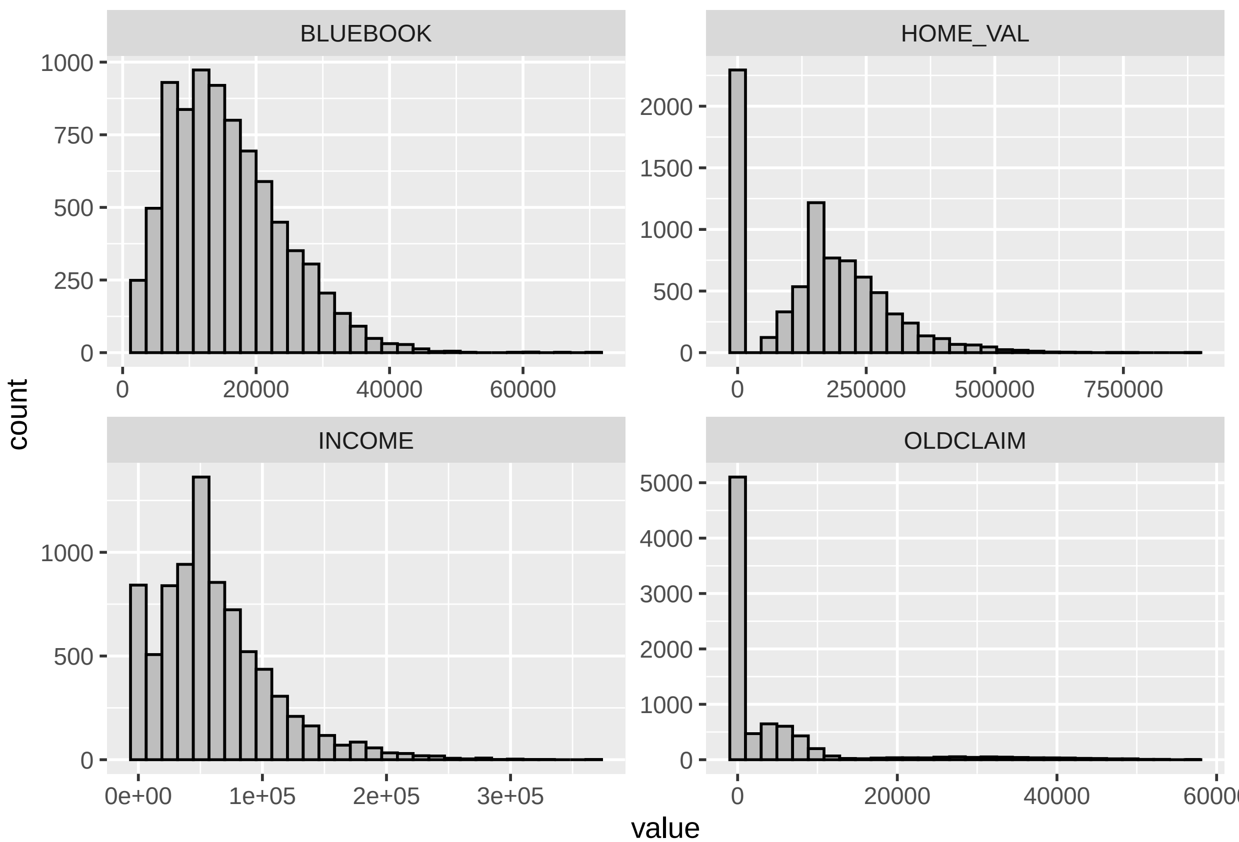
The histograms indicate that the distribution of AGE is roughly normal. The rest of the variables show some degree of skewness. Moreover, several variables have high occurrence of zeros.

Identifying highly skewed variabled:

From this "zoomed-in" histograms below, we see that the following variable -OLDCLAIM, INCOME, BLUEBOOK, HOME_VAL - are highly skewed. We will transform them during model building to access if that affects performance.

```
clean_training %>%
  dplyr::select(OLDCLAIM, INCOME, BLUEBOOK, HOME_VAL) %>%
  gather() %>%
  ggplot(aes(x= value)) +
  geom_histogram(fill='gray', color = "black") +
  facet_wrap(~key, scales = 'free')
```

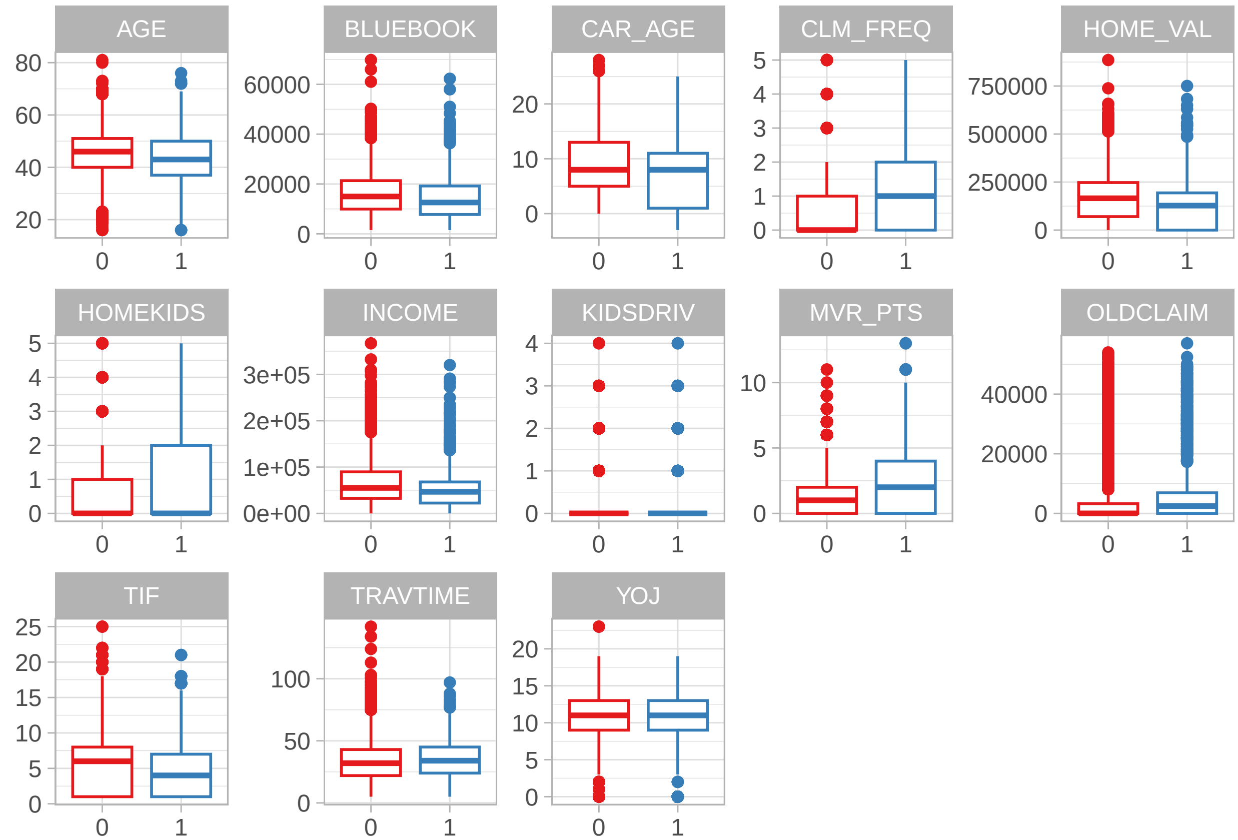
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Boxplots of feature variables

We examine boxplots of the variables to identify outliers.

```
plot_vars <- c("TARGET_FLAG", names(keep(clean_training, is.numeric)))
clean_training[plot_vars] %>%
 dplyr::select(-TARGET_AMT) %>%
 gather(variable, value, -TARGET_FLAG) %>%
 ggplot(aes(x = TARGET_FLAG, y = value, color = TARGET_FLAG)) +
 geom_boxplot() +
 scale_color_brewer(palette = "Set1") +
 theme_light() +
 theme(legend.position = "none") +
 facet_wrap(~variable, scales = "free", ncol = 5) +
 labs(x = NULL, y = NULL)
```



Boxplots of the feature variables shows that some variables have outliers. Let's examine where the outliers lie in response to the TARGET PAYOUT. We'll also test a model where we remove outliers to assess if that impacts performance.

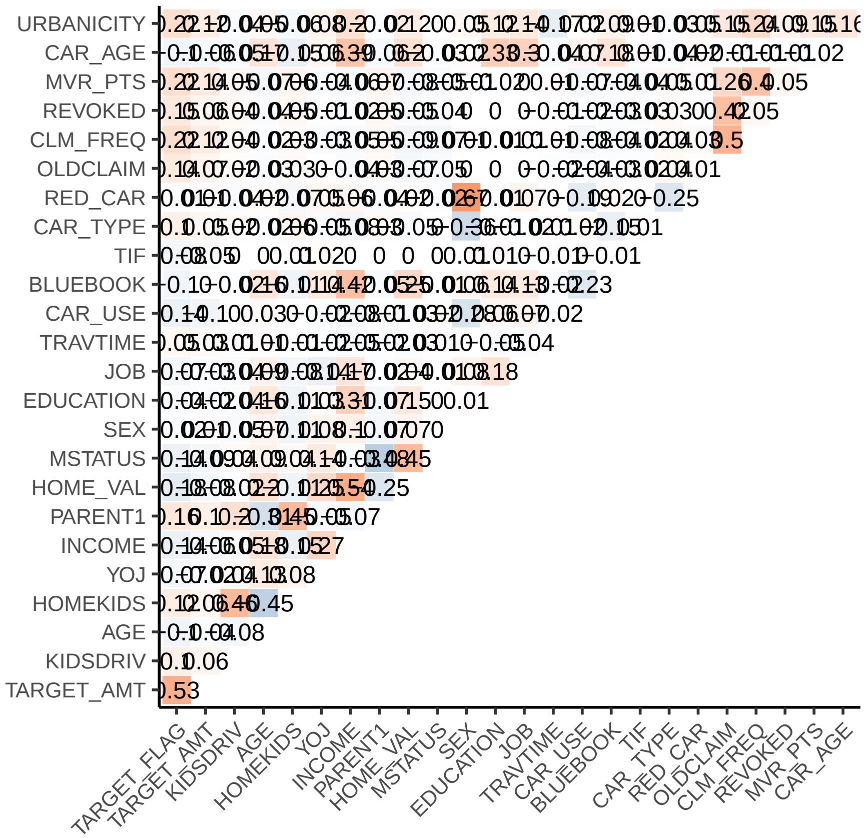
Correlation

We can also observe the correlation of our variables with each other and the target variable with a corrplot:

```
corr_dataframe = clean_training %>%
  mutate_if(is.factor, as.numeric) %>%
  select_if(is.numeric)

q <- cor(corr_dataframe)

ggcorrplot(q,
           type = "upper",
           outline.color = "white",
           ggtheme = ggplot2::theme_classic,
           colors = c("#6D9EC1", "white", "#E46726"),
           lab = TRUE,
           show.legend = FALSE,
           tl.cex = 8,
           lab_size = 3)
```



The correlation plot shows the relationships between various variables and the target variable (TARGET_AMT). The MVR_PTS (number of motor vehicle records points), CLM_FREQ (claim frequency), and OLDCLAIM (previous claims indicator) have the strongest positive correlations with TARGET_AMT, suggesting that higher values of these variables are associated with higher claim amounts.

On the other hand, CAR_AGE (age of the car) has a moderate negative correlation, implying that older cars tend to have lower claim amounts. BLUEBOOK (the resale value of the car) and HOME_VAL (home value) also show moderate positive correlations, indicating that higher resale values and home values are linked to higher claim amounts.

Variables like INCOME, YOJ (years on job), HOMEKIDS (number of kids at home), and AGE exhibit relatively weaker correlations with the target variable. KIDSDRIV (number of kids driving) and TARGET_FLAG (whether a claim was made or not) have very weak correlations.

Checking for Imbalance Data

We check for the balance of the data using our target variable, TARGET FLAG. If the data is imbalanced our model can be biased towards the target class that appears the most.

```
table(clean_training$TARGET_FLAG)
```

```
##
##    0    1
## 6008 2153
```

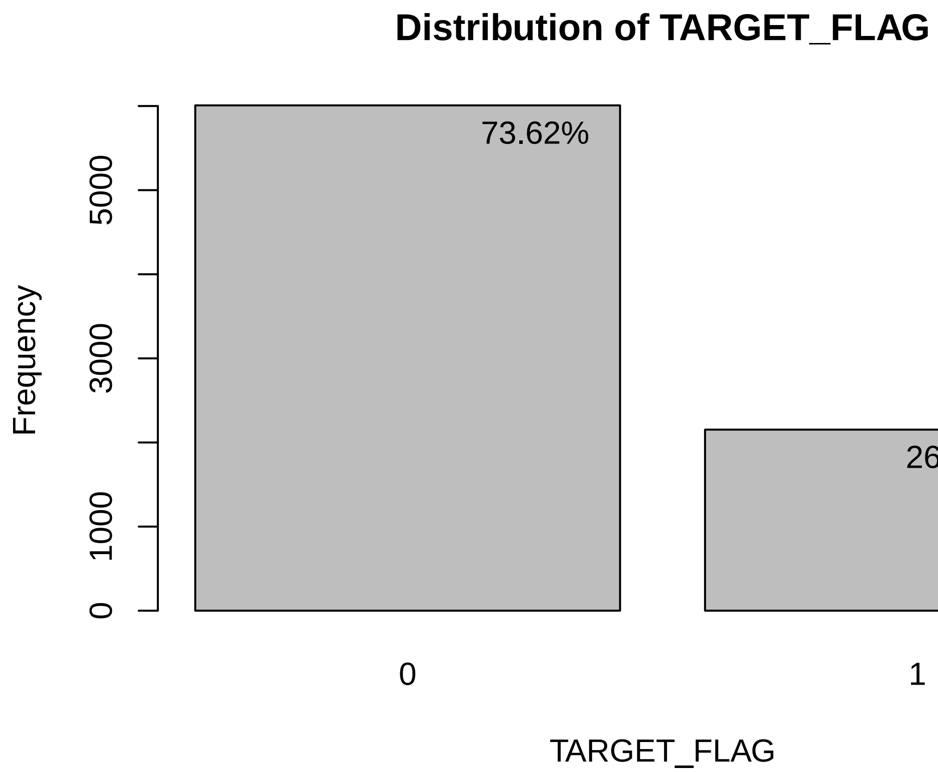
```
prop.table(table(clean_training$TARGET_FLAG))
```

```
##
##         0         1
## 0.7361843 0.2638157
```

```
# Calculate proportions
prop_table <- prop.table(table(clean_training$TARGET_FLAG)) * 100

# Bar plot of TARGET_FLAG distribution with percentages
barplot(table(clean_training$TARGET_FLAG),
        main = "Distribution of TARGET_FLAG",
        xlab = "TARGET_FLAG",
        ylab = "Frequency",
        col = "gray",
        border = "black")

# Add percentages on each bar
text(x = 1:length(prop_table),
     y = table(clean_training$TARGET_FLAG),
     labels = paste0(round(prop_table, 2), "%"),
     col = "black",
     pos = 1.5)
```



The data and the plot exhibits a significant class imbalance, with only 26% of instances belonging to the positive class (those who have experienced an accident), while the remaining 74% belong to the negative class (those who have not experienced an accident). This severe imbalance in the dataset could adversely impact the model's accuracy during the model building stage if left untreated.

To address this imbalance, we will employ an oversampling technique. Oversampling involves generating synthetic instances of the minority class (in this case, the positive class) to balance the class distribution. By increasing the representation of the minority class, the model will have an opportunity to learn patterns from both classes more effectively, potentially improving its overall performance and accuracy.

Oversampling and Splitting:

Before we oversample to account for the imbalanced dataset, lets split the dataset into 80% training and 20% testing.This way our test dataset will not be affected by the over sampled process. We can use the ovun.sample() from ROSE package in order to take care of the imbalanced data.

```
# Split data into training and test sets
set.seed(123)
train_index <- createDataPartition(clean_training$TARGET_FLAG, p = 0.8, list = FALSE)
train_data <- clean_training[train_index, ]
test_data <- clean_training[-train_index, ]

# Identify the minority class count
minority_count <- sum(train_data$TARGET_FLAG == 1)

# Determine the desired size of the oversampled dataset
```

```r
N <- max(2 * minority_count, nrow(train_data))

# Over-sample the minority class only in the training set
train_data_balanced <- ovun.sample(TARGET_FLAG ~ ., data = train_data, N = N, seed = 42, method = "over
```

3. BUILDING AND SELECTING MODELS:

Here we start with the binomial modeling that utilizes the feature set to predict the binary logistic regression model that includes all original feature predictor variables. TARGET_FLAG coded '1' is a car that was in a crash and '0' otherwise.

Binary Logistic Regression Model 1:

In Model 1, we'll exclude the TARGET_AMT column from our dataset because it represents the response variable for accident costs, making it unnecessary for our analysis.

```r
set.seed(456)
m1 <- glm(formula = TARGET_FLAG ~ . - TARGET_AMT, family = binomial(link = "logit"), data = train_data_
summary(m1)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ . - TARGET_AMT, family = binomial(link = "logit"),
##     data = train_data_balanced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5229  -0.7065  -0.3867   0.6221   2.9705
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -2.979e+00  3.272e-01  -9.103  < 2e-16 ***
## KIDSDRIV                   3.149e-01  7.021e-02   4.485 7.29e-06 ***
## AGE                       -8.111e-03  4.583e-03  -1.770 0.076733 .
## HOMEKIDS                   4.090e-02  4.177e-02   0.979 0.327571
## YOJ                       -1.943e-02  9.666e-03  -2.010 0.044449 *
## INCOME                    -2.674e-06  1.217e-06  -2.197 0.028049 *
## PARENT1Yes                 3.919e-01  1.258e-01   3.115 0.001838 **
## HOME_VAL                  -1.354e-06  3.876e-07  -3.493 0.000477 ***
## MSTATUSYes                -4.610e-01  9.405e-02  -4.902 9.50e-07 ***
## SEXM                       2.695e-01  1.271e-01   2.121 0.033953 *
## EDUCATIONBelow High School  3.521e-01  1.304e-01   2.700 0.006926 **
## EDUCATIONHigh School       5.155e-01  1.003e-01   5.138 2.78e-07 ***
## EDUCATIONMasters           2.631e-01  1.556e-01   1.691 0.090768 .
## EDUCATIONPhD               5.545e-01  1.997e-01   2.777 0.005483 **
## JOBClerical                1.735e-01  1.197e-01   1.449 0.147246
## JOBDoctor                 -1.605e+00  3.619e-01  -4.434 9.25e-06 ***
## JOBHome Maker             -9.421e-02  1.729e-01  -0.545 0.585899
## JOBLawyer                 -4.581e-01  2.131e-01  -2.150 0.031591 *
## JOBManager                -1.039e+00  1.610e-01  -6.457 1.07e-10 ***
```

```
## JOBProfessional           -1.111e-01  1.326e-01  -0.837 0.402337
## JOBStudent                -7.888e-03  1.440e-01  -0.055 0.956309
## JOBUNSPECIFIED            -6.062e-01  2.088e-01  -2.903 0.003691 **
## TRAVTIME                   1.911e-02  2.110e-03   9.053  < 2e-16 ***
## CAR_USEPrivate            -8.295e-01  1.010e-01  -8.216  < 2e-16 ***
## BLUEBOOK                  -1.842e-05  5.914e-06  -3.116 0.001836 **
## TIF                       -4.606e-02  8.020e-03  -5.743 9.31e-09 ***
## CAR_TYPEPanel Truck        4.105e-01  1.811e-01   2.266 0.023422 *
## CAR_TYPEPickup             4.788e-01  1.118e-01   4.281 1.86e-05 ***
## CAR_TYPESports Car         1.134e+00  1.476e-01   7.686 1.52e-14 ***
## CAR_TYPESUV                8.204e-01  1.271e-01   6.453 1.09e-10 ***
## CAR_TYPEVan                7.228e-01  1.384e-01   5.221 1.78e-07 ***
## RED_CARyes                -6.345e-02  9.571e-02  -0.663 0.507338
## OLDCLAIM                  -1.913e-05  4.450e-06  -4.299 1.71e-05 ***
## CLM_FREQ                   1.534e-01  3.232e-02   4.747 2.07e-06 ***
## REVOKEDYes                 8.765e-01  1.028e-01   8.526  < 2e-16 ***
## MVR_PTS                    1.196e-01  1.552e-02   7.707 1.29e-14 ***
## CAR_AGE                   -1.158e-03  8.525e-03  -0.136 0.891991
## URBANICITYUrban            2.513e+00  1.251e-01  20.090  < 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: 7536.3  on 6529  degrees of freedom
## Residual deviance: 5791.8  on 6492  degrees of freedom
## AIC: 5867.8
##
## Number of Fisher Scoring iterations: 5
```

AIC(Akaike Information Criterion) values of 5867.8 measure of the relative quality of a statistical model for a given set of data. It is used as a measure of the model's goodness of fit while penalizing for the number of model parameters. A lower AIC value indicates a better model fit, with the "best" model being the one with the lowest AIC value.

The Variance Inflation Factor (VIF): Lets check Variance Inflation Factor (VIF) to detect 'multi-collinearity' in our models as quantifies the correlation and its strength between independent variables in a regression model. The interpretation of VIF values is as follows:

- VIF $<$ 1: No correlation
- 1 $<$ VIF $<$ 5: Moderate correlation
- VIF $>$ 5: Severe correlation

```
knitr::kable(vif(m1))
```

	GVIF	Df	GVIF^(1/(2*Df))
KIDSDRIV	1.369005	1	1.170045
AGE	1.449315	1	1.203875
HOMEKIDS	2.200309	1	1.483344
YOJ	1.541351	1	1.241512
INCOME	2.571324	1	1.603535

	GVIF	Df	GVIF^(1/(2*Df))
PARENT1	1.947544	1	1.395544
HOME_VAL	1.879265	1	1.370863
MSTATUS	2.059042	1	1.434936
SEX	3.788703	1	1.946459
EDUCATION	9.584693	4	1.326469
JOB	21.906156	8	1.212789
TRAVTIME	1.065914	1	1.032431
CAR_USE	2.369786	1	1.539411
BLUEBOOK	2.263397	1	1.504459
TIF	1.012367	1	1.006164
CAR_TYPE	6.758179	5	1.210551
RED_CAR	1.803772	1	1.343046
OLDCLAIM	1.627665	1	1.275800
CLM_FREQ	1.472233	1	1.213356
REVOKED	1.300438	1	1.140368
MVR_PTS	1.173168	1	1.083129
CAR_AGE	2.010655	1	1.417976
URBANICITY	1.176659	1	1.084739

We will focus on the GVIF, as it measures how much the variance of the estimated regression coefficients is increased due to multicollinearity, as we can see from above that EDUCATION, JOB and CAR_TYPE represents severe correlation. We shall see if the log transformation will reduce the VIF in the next model.

Binary Logistic Regression Model 2:

INCOME, BLUEBOOK, OLDCLAIM and HOME_VAL are right-skewed. To make results normal, they are log-transformed (adding 1 to make sure that log-transformation is possible for 0 values).

```r
set.seed(789)
m2 <- glm(formula = TARGET_FLAG ~ KIDSDRIV + log(INCOME + 1) + PARENT1 + log(HOME_VAL + 1) + MSTATUS + E
summary(m2)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ KIDSDRIV + log(INCOME + 1) + PARENT1 +
##     log(HOME_VAL + 1) + MSTATUS + EDUCATION + JOB + TRAVTIME +
##     CAR_USE + log(BLUEBOOK + 1) + TIF + CAR_TYPE + log(OLDCLAIM +
##     1) + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY, family = binomial(link = "logit"),
##     data = train_data_balanced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5638  -0.7014  -0.3954   0.6113   3.0072
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.477353   0.622108   0.767 0.442894
## KIDSDRIV                     0.339970   0.063036   5.393 6.92e-08 ***
## log(INCOME + 1)             -0.092820   0.015534  -5.975 2.30e-09 ***
## PARENT1Yes                   0.490128   0.108063   4.536 5.74e-06 ***
```

```
## log(HOME_VAL + 1)          -0.029050   0.007766  -3.741 0.000184 ***
## MSTATUSYes                 -0.420492   0.092862  -4.528 5.95e-06 ***
## EDUCATIONBelow High School  0.373852   0.121309   3.082 0.002058 **
## EDUCATIONHigh School        0.547575   0.092720   5.906 3.51e-09 ***
## EDUCATIONMasters            0.202213   0.150383   1.345 0.178737
## EDUCATIONPhD                0.335759   0.189455   1.772 0.076355 .
## JOBClerical                 0.210738   0.118130   1.784 0.074431 .
## JOBDoctor                  -1.641540   0.361298  -4.543 5.53e-06 ***
## JOBHome Maker              -0.476013   0.186648  -2.550 0.010762 *
## JOBLawyer                  -0.481429   0.212264  -2.268 0.023325 *
## JOBManager                 -1.090221   0.160500  -6.793 1.10e-11 ***
## JOBProfessional            -0.156243   0.131738  -1.186 0.235620
## JOBStudent                 -0.401670   0.162874  -2.466 0.013658 *
## JOBUNSPECIFIED             -0.672716   0.207300  -3.245 0.001174 **
## TRAVTIME                    0.019580   0.002114   9.262  < 2e-16 ***
## CAR_USEPrivate             -0.803125   0.101236  -7.933 2.14e-15 ***
## log(BLUEBOOK + 1)          -0.348168   0.061678  -5.645 1.65e-08 ***
## TIF                        -0.046355   0.008021  -5.779 7.51e-09 ***
## CAR_TYPEPanel Truck         0.462867   0.159760   2.897 0.003764 **
## CAR_TYPEPickup              0.513639   0.111512   4.606 4.10e-06 ***
## CAR_TYPESports Car          0.914536   0.121278   7.541 4.67e-14 ***
## CAR_TYPESUV                 0.672752   0.096594   6.965 3.29e-12 ***
## CAR_TYPEVan                 0.804028   0.132413   6.072 1.26e-09 ***
## log(OLDCLAIM + 1)           0.028206   0.013980   2.018 0.043635 *
## CLM_FREQ                    0.014016   0.049293   0.284 0.776145
## REVOKEDYes                  0.637101   0.092606   6.880 6.00e-12 ***
## MVR_PTS                     0.105843   0.015966   6.629 3.38e-11 ***
## URBANICITYUrban             2.510831   0.126211  19.894  < 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: 7536.3  on 6529  degrees of freedom
## Residual deviance: 5789.7  on 6498  degrees of freedom
## AIC: 5853.7
##
## Number of Fisher Scoring iterations: 5
```

The difference in AIC between model 1 (AIC: 5867.8) and model 2 (AIC: 5853.7) is 14.1, suggesting that both models provide similar fits to the data. Therefore, there is no strong evidence to favor one model over the other based on AIC alone. Now lets check VIF for model 2.

```
knitr::kable(vif(m2))
```

	GVIF	Df	GVIF^(1/(2*Df))
KIDSDRIV	1.103841	1	1.050639
log(INCOME + 1)	2.494651	1	1.579446
PARENT1	1.432626	1	1.196924
log(HOME_VAL + 1)	1.835040	1	1.354637
MSTATUS	2.004190	1	1.415694
EDUCATION	6.541499	4	1.264619

	GVIF	Df	GVIF^(1/(2*Df))
JOB	29.371989	8	1.235224
TRAVTIME	1.066010	1	1.032477
CAR_USE	2.382695	1	1.543598
log(BLUEBOOK + 1)	1.507064	1	1.227625
TIF	1.010995	1	1.005483
CAR_TYPE	2.288106	5	1.086295
log(OLDCLAIM + 1)	3.605611	1	1.898845
CLM_FREQ	3.369256	1	1.835553
REVOKED	1.034575	1	1.017141
MVR_PTS	1.229831	1	1.108977
URBANICITY	1.189286	1	1.090544

After running the log transformation it looks like the VIF for 'EDUCATION' and 'CAR_TYPE' has reduced but 'JOB' has increased. So, the variance of the estimated regression coefficients is increased approximately by 8 due to multicollinearity. In our next model (m3) will shift our focus to high-P values. We will remove the variable with higher P-values on our final model.

Binary Logistic Regression Model 3:

Let's remove variables with higher P-values to create more models.

```
set.seed(1011)
m3 <-glm(formula = TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 +
    HOME_VAL + MSTATUS + TRAVTIME +
    CAR_USE + BLUEBOOK + TIF + CAR_TYPE + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY, family = binomial(l
    data = train_data_balanced)
summary(m3)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 + HOME_VAL +
##     MSTATUS + TRAVTIME + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
##     CLM_FREQ + REVOKED + MVR_PTS + URBANICITY, family = binomial(link = "logit"),
##     data = train_data_balanced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4488  -0.7387  -0.4236   0.6710   2.8560
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.607e+00  1.934e-01 -13.480  < 2e-16 ***
## KIDSDRIV               3.491e-01  6.223e-02   5.610 2.02e-08 ***
## INCOME                -7.399e-06  9.752e-07  -7.587 3.27e-14 ***
## PARENT1Yes             5.389e-01  1.051e-01   5.126 2.96e-07 ***
## HOME_VAL              -1.733e-06  3.705e-07  -4.677 2.91e-06 ***
## MSTATUSYes            -3.485e-01  8.738e-02  -3.988 6.66e-05 ***
## TRAVTIME               1.943e-02  2.063e-03   9.420  < 2e-16 ***
## CAR_USEPrivate        -1.019e+00  7.611e-02 -13.391  < 2e-16 ***
## BLUEBOOK              -2.700e-05  5.182e-06  -5.209 1.90e-07 ***
```

```
## TIF                  -4.527e-02  7.878e-03  -5.746 9.14e-09 ***
## CAR_TYPEPanel Truck  4.015e-01  1.558e-01   2.576 0.009988 **
## CAR_TYPEPickup       3.994e-01  1.068e-01   3.738 0.000186 ***
## CAR_TYPESports Car   8.865e-01  1.168e-01   7.588 3.25e-14 ***
## CAR_TYPESUV          6.400e-01  9.397e-02   6.811 9.69e-12 ***
## CAR_TYPEVan          7.085e-01  1.280e-01   5.534 3.12e-08 ***
## CLM_FREQ             8.244e-02  2.843e-02   2.900 0.003735 **
## REVOKEDYes           6.941e-01  8.948e-02   7.757 8.71e-15 ***
## MVR_PTS              1.211e-01  1.512e-02   8.004 1.21e-15 ***
## URBANICITYUrban      2.345e+00  1.238e-01  18.942  < 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: 7536.3  on 6529  degrees of freedom
## Residual deviance: 5975.8  on 6511  degrees of freedom
## AIC: 6013.8
##
## Number of Fisher Scoring iterations: 5
```

```
knitr::kable(vif(m3))
```

	GVIF	Df	GVIF^(1/(2*Df))
KIDSDRIV	1.096872	1	1.047317
INCOME	1.565178	1	1.251071
PARENT1	1.426236	1	1.194251
HOME_VAL	1.740115	1	1.319134
MSTATUS	1.844432	1	1.358099
TRAVTIME	1.058655	1	1.028909
CAR_USE	1.394294	1	1.180802
BLUEBOOK	1.751706	1	1.323520
TIF	1.006787	1	1.003388
CAR_TYPE	2.133688	5	1.078731
CLM_FREQ	1.180342	1	1.086435
REVOKED	1.006808	1	1.003398
MVR_PTS	1.151928	1	1.073279
URBANICITY	1.151514	1	1.073086

Model 3 excludes some variables that were significant in Model 1 and Model 2, resulting in a higher AIC and residual deviance. In our last model we can see that all the variables statistically significant and the VIF values are also show very low to moderate correlation. Now, let's move on to the models selection based on classification model metrics

Model Selection:

To begin, we adhered to the professor's instructions and set the threshold to 0.5. Subsequently, we converted probabilities into classes and transformed predicted labels into factors for each of our models. Finally, we conducted a Confusion Matrix analysis to assess their performance.

```r
# Set threshold as per instruction
threshold <- 0.5
```

```r
preds1 = predict(m1, newdata = test_data, type = "response")

# Convert probabilities to class preds1
predicted_labels1 <- ifelse(preds1 >= threshold, "1", "0")

# Convert predicted labels to factors
predicted_labels_factor1 <- factor(predicted_labels1, levels = c("0", "1"))
```

```r
# Drop original variables
test_data_trans <- subset(test_data, select = -c(INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM))

# Apply log transformation to INCOME
test_data_trans$INCOME <- log(test_data$INCOME + 1)

# Apply log transformation to HOME_VAL
test_data_trans$HOME_VAL <- log(test_data$HOME_VAL + 1)

# Apply log transformation to BLUEBOOK
test_data_trans$BLUEBOOK <- log(test_data$BLUEBOOK + 1)

# Apply log transformation to OLDCLAIM
test_data_trans$OLDCLAIM <- log(test_data$OLDCLAIM + 1)

preds2 = predict(m2, newdata = test_data_trans)

# Convert probabilities to class preds2
predicted_labels2 <- ifelse(preds2 >= threshold, "1", "0")

# Convert predicted labels to factors
predicted_labels_factor2 <- factor(predicted_labels2, levels = c("0", "1"))
```

```r
preds3 = predict(m3, newdata = test_data , type = "response")

# Convert probabilities to class preds3
predicted_labels3 <- ifelse(preds3 >= threshold, "1", "0")

# Convert predicted labels to factors
predicted_labels_factor3 <- factor(predicted_labels3, levels = c("0", "1"))
```

```r
cm_m1 <- confusionMatrix(data=predicted_labels_factor1, test_data$TARGET_FLAG, mode = "everything")
cat("Confusion Matrix Model 1:\n")
```

Confusion Matrix:

```
## Confusion Matrix Model 1:
```

```r
print(cm_m1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1091  247
##          1  110  183
##
##                Accuracy : 0.7811
##                  95% CI : (0.7602, 0.801)
##     No Information Rate : 0.7364
##     P-Value [Acc > NIR] : 1.658e-05
##
##                   Kappa : 0.372
##
##  Mcnemar's Test P-Value : 6.115e-13
##
##             Sensitivity : 0.9084
##             Specificity : 0.4256
##          Pos Pred Value : 0.8154
##          Neg Pred Value : 0.6246
##               Precision : 0.8154
##                  Recall : 0.9084
##                      F1 : 0.8594
##              Prevalence : 0.7364
##          Detection Rate : 0.6689
##    Detection Prevalence : 0.8204
##       Balanced Accuracy : 0.6670
##
##        'Positive' Class : 0
##
```

```r
cm_m2 <- confusionMatrix(data=predicted_labels_factor2, test_data_trans$TARGET_FLAG, mode = "everything
cat("Confusion Matrix Model 2:\n")
```

```
## Confusion Matrix Model 2:
```

```r
print(cm_m2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 269  10
##          1 932 420
##
##                Accuracy : 0.4224
##                  95% CI : (0.3983, 0.4468)
##     No Information Rate : 0.7364
##     P-Value [Acc > NIR] : 1
##
```

```
##                  Kappa : 0.1189
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.2240
##             Specificity : 0.9767
##          Pos Pred Value : 0.9642
##          Neg Pred Value : 0.3107
##               Precision : 0.9642
##                  Recall : 0.2240
##                      F1 : 0.3635
##              Prevalence : 0.7364
##          Detection Rate : 0.1649
##    Detection Prevalence : 0.1711
##       Balanced Accuracy : 0.6004
##
##        'Positive' Class : 0
##
```

```r
cm_m3 <- confusionMatrix(data=predicted_labels_factor3, test_data$TARGET_FLAG, mode = "everything")
cat("Confusion Matrix Model 3:\n")
```

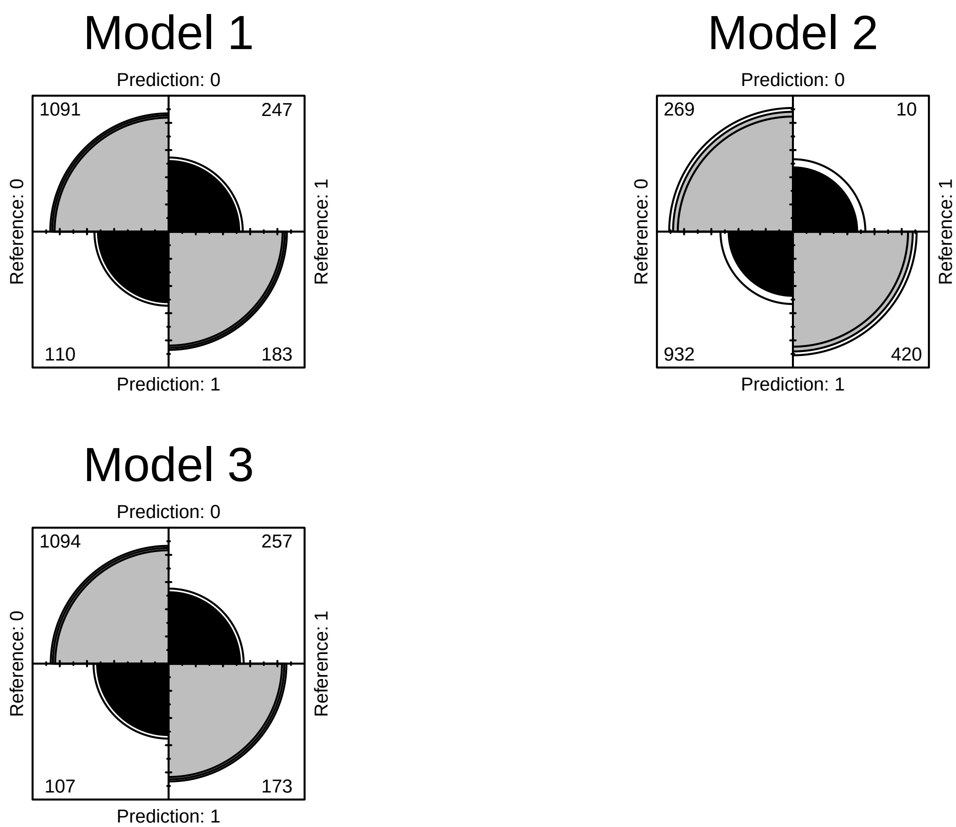
```
## Confusion Matrix Model 3:
```

```r
print(cm_m3)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1094  257
##          1  107  173
##
##                Accuracy : 0.7768
##                  95% CI : (0.7558, 0.7968)
##     No Information Rate : 0.7364
##     P-Value [Acc > NIR] : 9.139e-05
##
##                   Kappa : 0.3527
##
##  Mcnemar's Test P-Value : 5.731e-15
##
##             Sensitivity : 0.9109
##             Specificity : 0.4023
##          Pos Pred Value : 0.8098
##          Neg Pred Value : 0.6179
##               Precision : 0.8098
##                  Recall : 0.9109
##                      F1 : 0.8574
##              Prevalence : 0.7364
##          Detection Rate : 0.6708
##    Detection Prevalence : 0.8283
##       Balanced Accuracy : 0.6566
```

```
##
##        'Positive' Class : 0
##
```

```
par(mfrow=c(2,2))
fourfoldplot(cm_m1$table, color = c("black", "gray"), main="Model 1")
fourfoldplot(cm_m2$table, color = c("black", "gray"), main="Model 2")
fourfoldplot(cm_m3$table, color = c("black", "gray"), main="Model 3")
```




```
eval <- data.frame(cm_m1$byClass,
                   cm_m2$byClass,
                   cm_m3$byClass)
eval <- data.frame(t(eval))

# Add Accuracy to the evaluation dataframe
eval$Accuracy <- c(cm_m1$overall["Accuracy"], cm_m2$overall["Accuracy"], cm_m3$overall["Accuracy"])

eval <- dplyr::select(eval, Accuracy, Sensitivity, Specificity, Precision, Recall, F1, Balanced.Accurac
row.names(eval) <- c("Model 1", "Model 2", "Model 3")
knitr::kable(eval)
```

Logistic Regression Evaluation Metrics Summary:

	Accuracy	Sensitivity	Specificity	Precision	Recall	F1	Balanced.Accuracy
Model 1	0.7811159	0.9084097	0.4255814	0.8153961	0.9084097	0.8593935	0.6669955
Model 2	0.4224402	0.2239800	0.9767442	0.9641577	0.2239800	0.3635135	0.6003621
Model 3	0.7768240	0.9109076	0.4023256	0.8097705	0.9109076	0.8573668	0.6566166

Model Comparison:

- Model 1 demonstrates the highest accuracy among the three models. However, it exhibits relatively low specificity, suggesting a potential tendency to over-predict the positive class.
- Model 2 showcases remarkably high specificity but significantly low sensitivity, resulting in an overall lower accuracy.
- Model 3 presents a balanced performance, effectively balancing sensitivity and specificity, and consequently achieving the second-highest accuracy among the three models.

Upon comprehensive evaluation of performance metrics including accuracy, sensitivity, specificity, precision, recall, F1 score, and balanced accuracy, Model 2 emerges with notable specificity but compromised sensitivity, leading to an overall diminished accuracy. Conversely, Model 3 exhibits a more balanced performance across these metrics.

Considering model complexity, encompassing the number of variables and interpretability, we prioritize parsimony for ease of interpretation and reduced risk of overfitting. While Model 2 might initially appear more parsimonious based on AIC and residual deviance, this aspect needs to be weighed against its sensitivity and specificity performance.

Therefore, based on the comprehensive evaluation of metrics, Model 3 emerges as a favorable choice, offering a well-balanced trade-off between sensitivity and specificity, rendering it potentially the optimal choice overall.

Classification Error Rate of the Predictions:

Now lets take a took at the classification error rate and see if the accuracy and error rate sum up to one 1 for each of the models.

Classification Error Rate = FP + FN / TP + FP + TN + FN

Classification error rate of the predictions Model 1:

```
confusion_matrix_m1 <- as.data.frame(table("Actual Class" = test_data$TARGET_FLAG, "Predicted Class" = p
confusion_matrix_m1
```

```
##   Actual.Class Predicted.Class Freq
## 1            0               0 1091
## 2            1               0  247
## 3            0               1  110
## 4            1               1  183
```

```
class_error_rate1 <- (confusion_matrix_m1$Freq[3] + confusion_matrix_m1$Freq[2])/sum(confusion_matrix_m1
class_error_rate1
```

```
## [1] 0.2188841
```

Model 1 Verification of accuracy and error rate sum up to one 1

```
Accuracy1 <-  0.7921521
Verify <- round(Accuracy1+class_error_rate1)
print(Verify)
```

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

Classification error rate of the predictions Model 2:

```
confusion_matrix_m2 <- as.data.frame(table("Actual Class" = test_data_trans$TARGET_FLAG, "Predicted Clas
confusion_matrix_m2
```

```
##   Actual.Class Predicted.Class Freq
## 1            0               0  269
## 2            1               0   10
## 3            0               1  932
## 4            1               1  420
```

```
class_error_rate2 <- (confusion_matrix_m2$Freq[3] + confusion_matrix_m2$Freq[2])/sum(confusion_matrix_m2
class_error_rate2
```

```
## [1] 0.5775598
```

Model 2 Verification of accuracy and error rate sum up to one 1

```r
Accuracy2 <-  0.4800736
Verify <- round(Accuracy2+class_error_rate2)
print(Verify)
```

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

Classification error rate of the predictions Model 3:

```r
confusion_matrix_m3 <- as.data.frame(table("Actual Class" = test_data$TARGET_FLAG, "Predicted Class" = p
confusion_matrix_m3
```

```
##   Actual.Class Predicted.Class Freq
## 1            0               0 1094
## 2            1               0  257
## 3            0               1  107
## 4            1               1  173
```

```r
class_error_rate3 <- (confusion_matrix_m3$Freq[3] + confusion_matrix_m3$Freq[2])/sum(confusion_matrix_m3
class_error_rate3
```

```
## [1] 0.223176
```

Model 3 Verification of accuracy and error rate sum up to one 1

```r
Accuracy3 <-  0.7915389
Verify <- round(Accuracy3+class_error_rate3)
print(Verify)
```

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

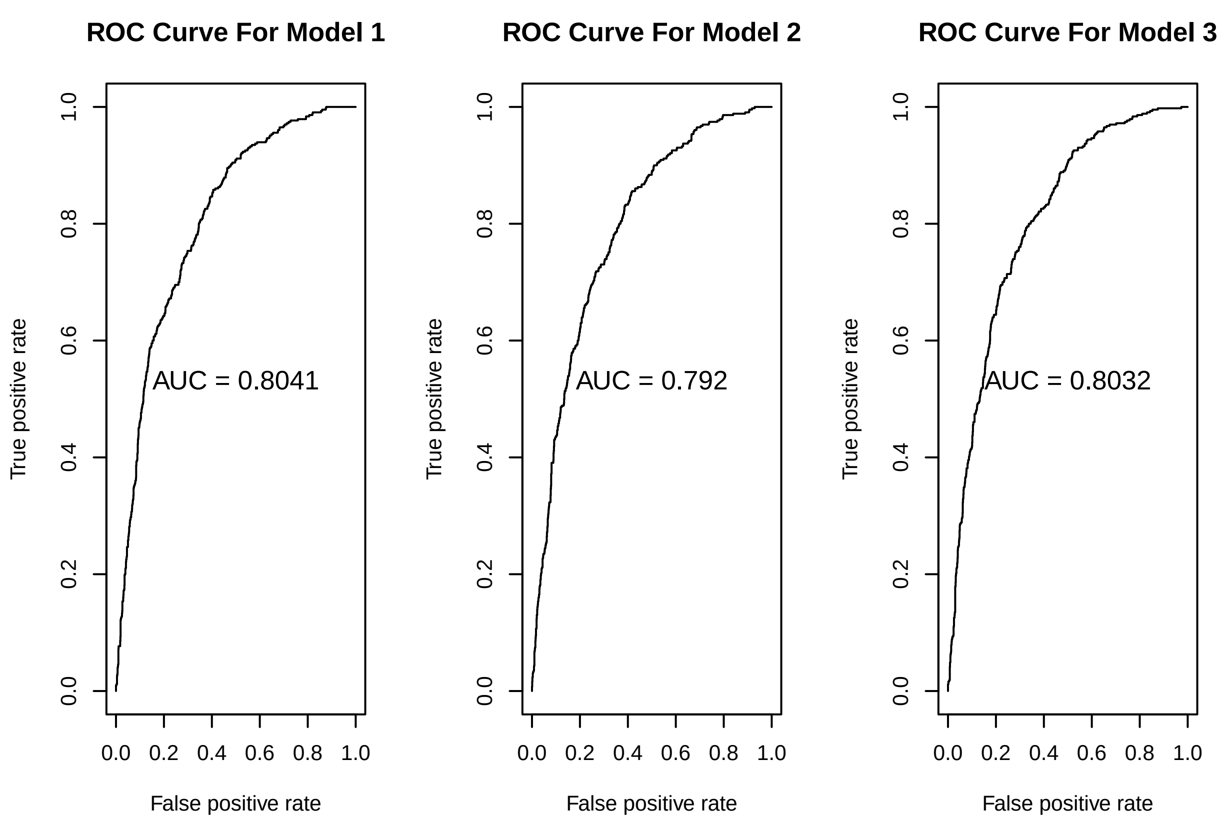
ROC/AUC Curves:

```r
par(mfrow = c(1,3))

# Plot for Model 1
pred_obj <- prediction(preds1, test_data$TARGET_FLAG)
auc_value <- performance(pred_obj, "auc")@y.values[[1]]
roc <- performance(pred_obj, "tpr", "fpr") # Calculate ROC curve
plot(roc, main = "ROC Curve For Model 1", colorize = FALSE) # Plot ROC curve
text(0.5, 0.5, paste("AUC =", round(auc_value, 4)), adj = c(0.5, -0.5), cex = 1.2)

# Plot for Model 2
pred_obj2 <- prediction(preds2, test_data_trans$TARGET_FLAG)
auc_value2 <- performance(pred_obj2, "auc")@y.values[[1]]
roc <- performance(pred_obj2, "tpr", "fpr")# Calculate ROC curve
plot(roc, main = "ROC Curve For Model 2", colorize = FALSE)# Plot ROC curve
text(0.5, 0.5, paste("AUC =", round(auc_value2, 4)), adj = c(0.5, -0.5), cex = 1.2)

# Plot for Model 3
pred_obj3 <- prediction(preds3, test_data$TARGET_FLAG)
auc_value3 <- performance(pred_obj3, "auc")@y.values[[1]]
roc <- performance(pred_obj3, "tpr", "fpr")# Calculate ROC curve
plot(roc, main = "ROC Curve For Model 3", colorize = FALSE)# Plot ROC curve
text(0.5, 0.5, paste("AUC =", round(auc_value3, 4)), adj = c(0.5, -0.5), cex = 1.2)
```



Indeed, the AUC values obtained from the ROC curve appear slightly higher than both the accuracy and balanced accuracy metrics. Generally, AUC values closer to 1 indicate superior model performance, particularly in terms of classification discrimination.

The discrepancy between AUC and accuracy metrics suggests that our model's predictions are well-ranked or well-discriminated across different thresholds. This phenomenon can occur if a dataset is imbalanced or if the mis-classifications made by the model are not evenly distributed among classes. Therefore, we can interpret AUC values of 0.80 as reasonable, indicating that our model's predictions are well-separated between classes, even if the overall accuracy is slightly lower.

4. EVALUATION:

LOGISTIC REGRESSION- EVALUATION:

Predictions: After reviewing the outcomes of the three models, it's evident that model three exhibits the strongest predictive capability and maintains a robust relationship with the underlying data. The data transformations applied have effectively mitigated any underlying skews and multicollinearity issues within the dataset. Despite not reaching perfection, the model achieves a commendable AUC of 0.8032, underscoring its formidable predictive prowess.

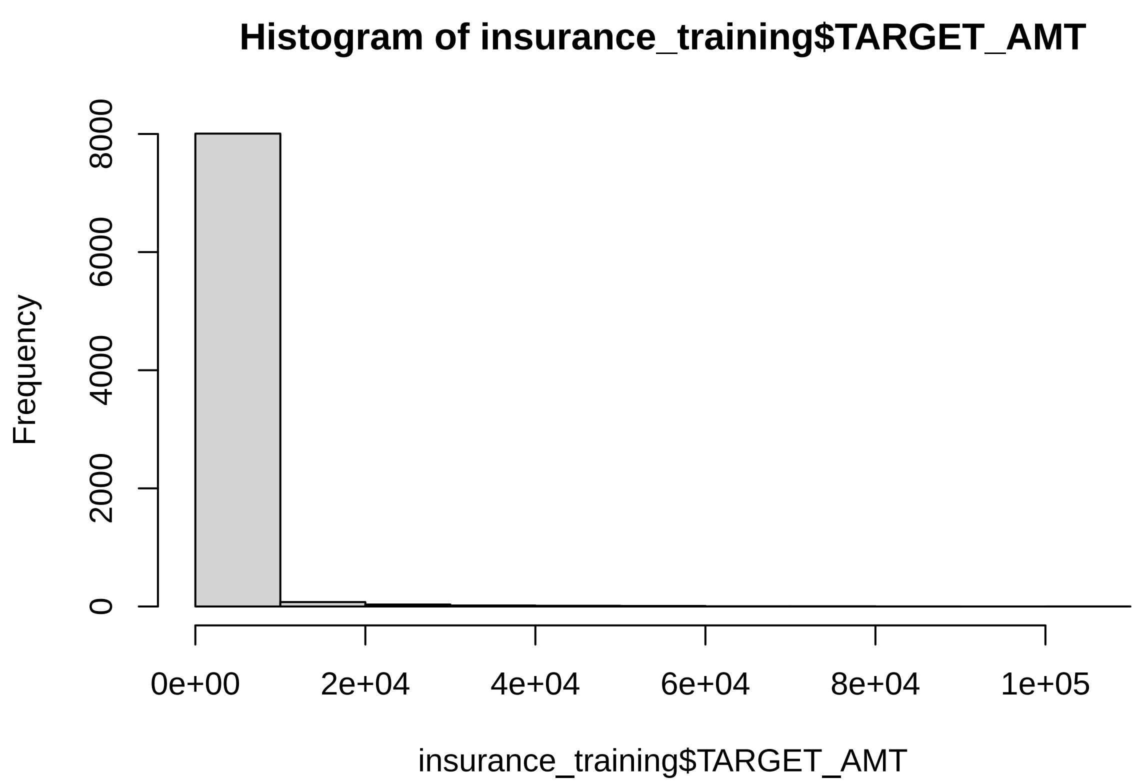
We'll proceed by applying this model three (m3) to our evaluation data and generating predictions accordingly. The subsequent results closely resemble the distributions observed in our training data.

```
A <- predict(m3, newdata = clean_evaluation, type = "response")
clean_evaluation$TARGET_FLAG <- ifelse(A >= threshold, "1", "0")
```

Multiple Linear Regression:

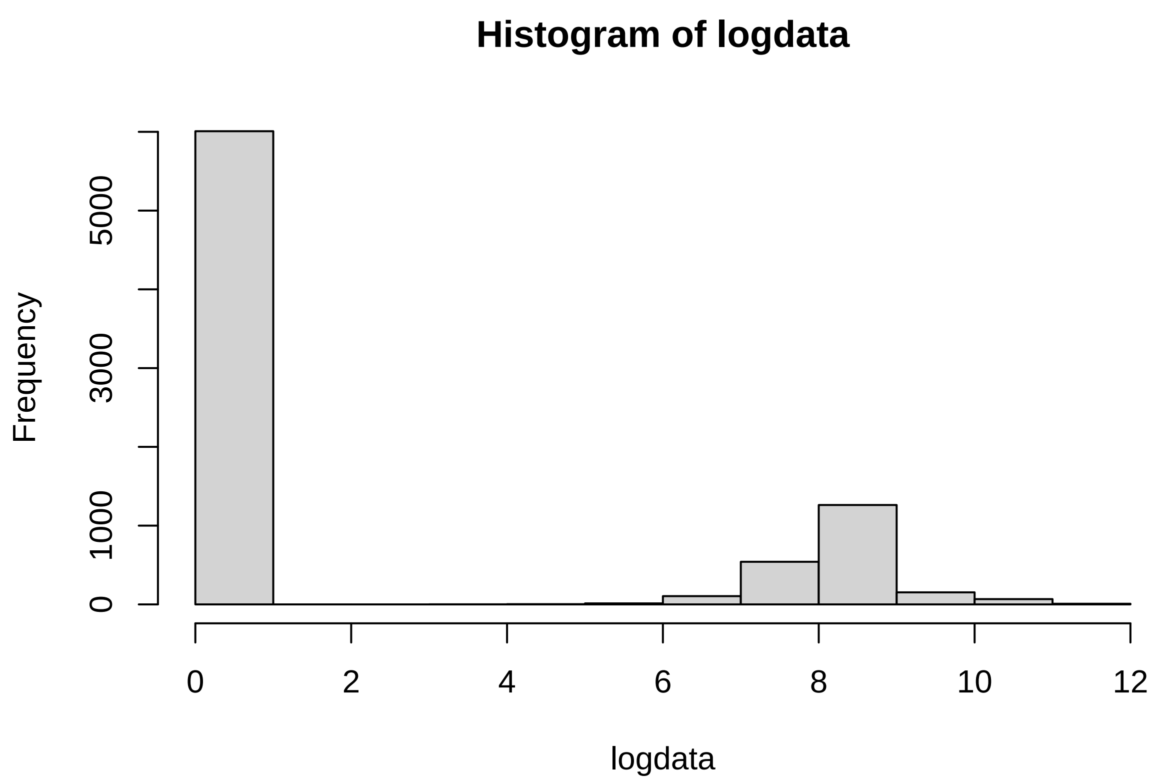
Before we build the multiple regression models, let take a look at our distribution for response variable in multiple linear regression the 'TAGET_AMT'

```
hist(insurance_training$TARGET_AMT)
```



We can wee that the our target variable for multiple regression has too many zeros.One way to deal with these too zeros is to remove them from the dataset. However, ethically we should not remove them since this is problem and we do not want to alter the problem that presents in our data. Thus, we will use log transformation particularly log1p() in order to *improve* the distribution of our target variable.

```
logdata <- log1p(insurance_training$TARGET_AMT)
hist(logdata)
```



The situation has improved significantly with the presence of a prominent outlier at 0, considering the dataset's substantial imbalance, which cannot be rectified.

Multiple Regression Model 1: For our first multiple linear regression, we will use the all predictors along with log transformation on the target variable. By including all predictors, we aim to capture combined effects and potential interactions between variables, thus providing a more detailed analysis of the data.

```
lm1 <- lm(formula = log1p(TARGET_AMT) ~., data = train_data[,-(1)])
summary(lm1)
```

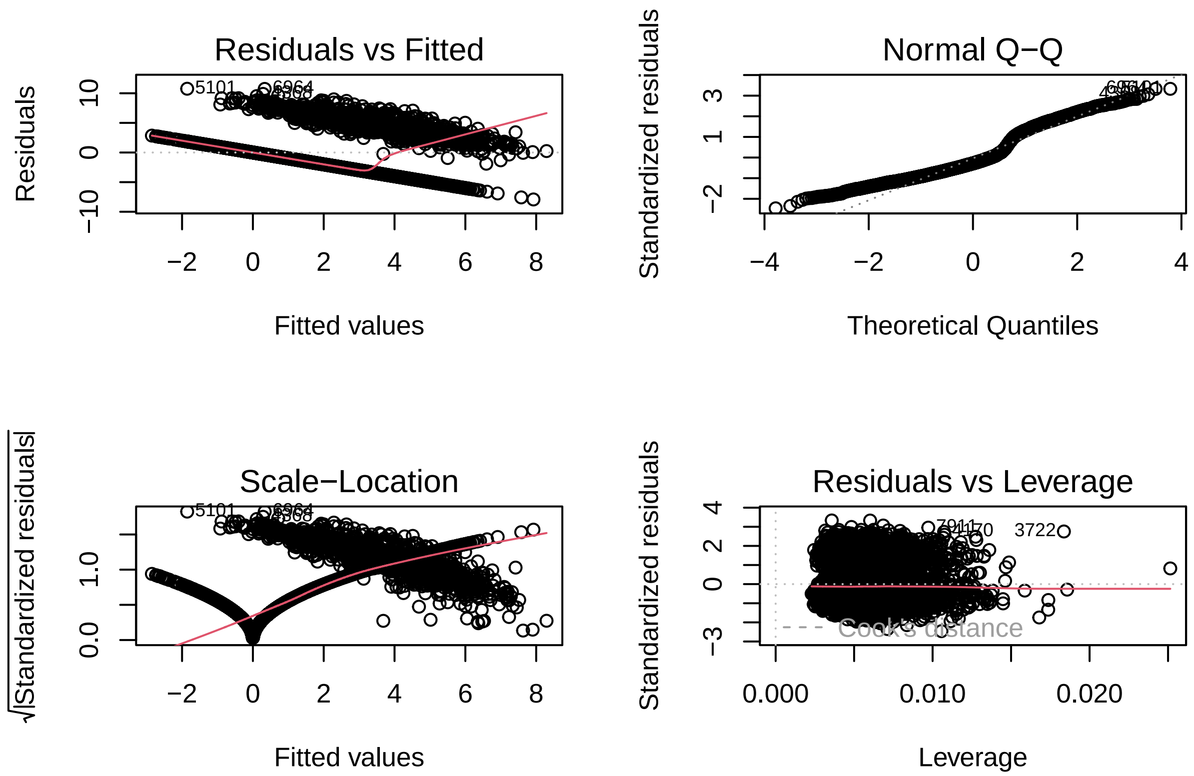
```
##
## Call:
## lm(formula = log1p(TARGET_AMT) ~ ., data = train_data[, -(1)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9229 -2.3374 -0.9123  2.0887 10.7611
##
## Coefficients:
```

```
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.540e-01  3.770e-01   1.735 0.082864 .
## KIDSDRIV                  4.299e-01  8.966e-02   4.795 1.66e-06 ***
## AGE                      -6.431e-03  5.650e-03  -1.138 0.254997
## HOMEKIDS                  3.495e-02  5.225e-02   0.669 0.503603
## YOJ                      -1.896e-02  1.206e-02  -1.572 0.115966
## INCOME                   -3.900e-06  1.447e-06  -2.696 0.007033 **
## PARENT1Yes                6.891e-01  1.618e-01   4.258 2.09e-05 ***
## HOME_VAL                 -1.195e-06  4.701e-07  -2.542 0.011048 *
## MSTATUSYes               -6.301e-01  1.147e-01  -5.491 4.14e-08 ***
## SEXM                      1.720e-01  1.460e-01   1.178 0.238693
## EDUCATIONBelow High School  4.270e-01  1.633e-01   2.615 0.008935 **
## EDUCATIONHigh School      4.917e-01  1.251e-01   3.931 8.56e-05 ***
## EDUCATIONMasters          3.164e-01  1.772e-01   1.785 0.074249 .
## EDUCATIONPhD              4.773e-01  2.304e-01   2.071 0.038375 *
## JOBClerical               1.811e-01  1.544e-01   1.173 0.240825
## JOBDoctor                -1.217e+00  3.444e-01  -3.534 0.000412 ***
## JOBHome Maker            -1.114e-01  2.141e-01  -0.520 0.602879
## JOBLawyer                -5.236e-01  2.489e-01  -2.104 0.035415 *
## JOBManager               -1.270e+00  1.865e-01  -6.809 1.07e-11 ***
## JOBProfessional          -2.037e-01  1.690e-01  -1.205 0.228105
## JOBStudent               -3.192e-02  1.847e-01  -0.173 0.862779
## JOBUNSPECIFIED           -8.189e-01  2.559e-01  -3.199 0.001384 **
## TRAVTIME                  1.872e-02  2.560e-03   7.314 2.91e-13 ***
## CAR_USEPrivate           -9.916e-01  1.306e-01  -7.595 3.52e-14 ***
## BLUEBOOK                 -1.646e-05  6.897e-06  -2.387 0.017005 *
## TIF                      -6.198e-02  9.656e-03  -6.419 1.47e-10 ***
## CAR_TYPEPanel Truck       4.463e-01  2.217e-01   2.013 0.044108 *
## CAR_TYPEPickup            5.477e-01  1.360e-01   4.028 5.69e-05 ***
## CAR_TYPESports Car        1.267e+00  1.719e-01   7.371 1.91e-13 ***
## CAR_TYPESUV               9.291e-01  1.421e-01   6.536 6.79e-11 ***
## CAR_TYPEVan               6.379e-01  1.712e-01   3.725 0.000197 ***
## RED_CARyes                4.490e-02  1.191e-01   0.377 0.706105
## OLDCLAIM                 -1.912e-05  5.888e-06  -3.246 0.001175 **
## CLM_FREQ                  2.152e-01  4.386e-02   4.907 9.45e-07 ***
## REVOKEDYes                1.256e+00  1.382e-01   9.085  < 2e-16 ***
## MVR_PTS                   1.927e-01  2.083e-02   9.251  < 2e-16 ***
## CAR_AGE                  -4.020e-03  1.013e-02  -0.397 0.691409
## URBANICITYUrban           2.508e+00  1.108e-01  22.632  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.237 on 6492 degrees of freedom
## Multiple R-squared:  0.2271, Adjusted R-squared:  0.2227
## F-statistic: 51.54 on 37 and 6492 DF,  p-value: < 2.2e-16
```

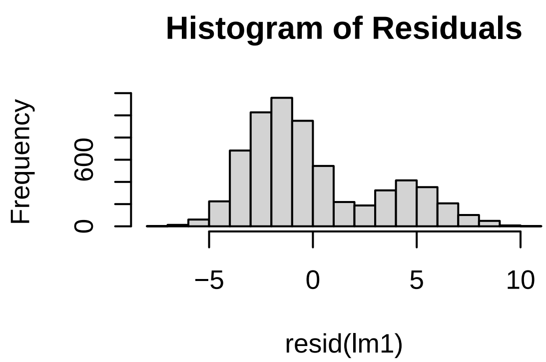
The model's overall fit is described by the R-squared value of 0.2271, suggesting that around 22.71% of the variance in TARGET_AMT can be explained by the predictors included in the model.The R-squared value indicates a moderate predictive power of the model. However since we are doing multiple regression it's generally more appropriate to look at the adjusted R-squared (adjusted R^2) rather than the regular R-squared (R^2) as it takes into account the number of predictors in the model, penalizing the addition of unnecessary variables that do not contribute significantly to the model's explanatory power. The adjusted R-squared is a more robust metric for assessing the overall effectiveness of a multiple regression model.

The model's adjusted R-squared value of 0.2227 indicates that approximately 22.27% of the variability in the target amount can be explained by the predictors, after adjusting for the number of predictors in the model. Additionally, the F-statistic of 51.54 with a p-value less than 2.2e-16 suggests that the overall model is statistically significant, implying that at least one predictor variable has a non-zero effect on the target amount. However, we should be careful when interpreting the model's coefficients and significance levels, as they rely on the assumptions and limitations of linear regression analysis.

```r
par(mfrow=c(2,2))
plot(lm1)
```




```r
hist(resid(lm1), main="Histogram of Residuals")
```



In the residuals vs fitted plot while there is a central tendency around the zero line, there are some visible patterns, such as a slight funnel shape. The diagnostic QQ plot also reveals a large deviation from normal in the upper quantiles that heavily affects the results. The residuals vs leverage plot shows there are significant outliers that are also affecting model performance. With the log transformation the histogram of the residuals appears to be more normal vs. without the log transformation that we have checked. We have decided not to include in our model since the log transformation has improved our model significantly.

Multiple Regression Model 2: In our subsequent multiple linear regression analysis, we adopted the log transformation approach for the response variable TARGET_AMT, coupled with stepwise feature selection, aimed at enhancing the previous findings. This iterative method enables us to enhance our model by focusing solely on the most significant features while accommodating the attributes of the transformed response variable. Our objective is to achieve improved performance compared to the initial analysis, thereby refining our understanding of the underlying relationships within the data.

```r
lm2 <- stepAIC(lm1, trace = FALSE, direction = 'backward')
summary(lm2)
```

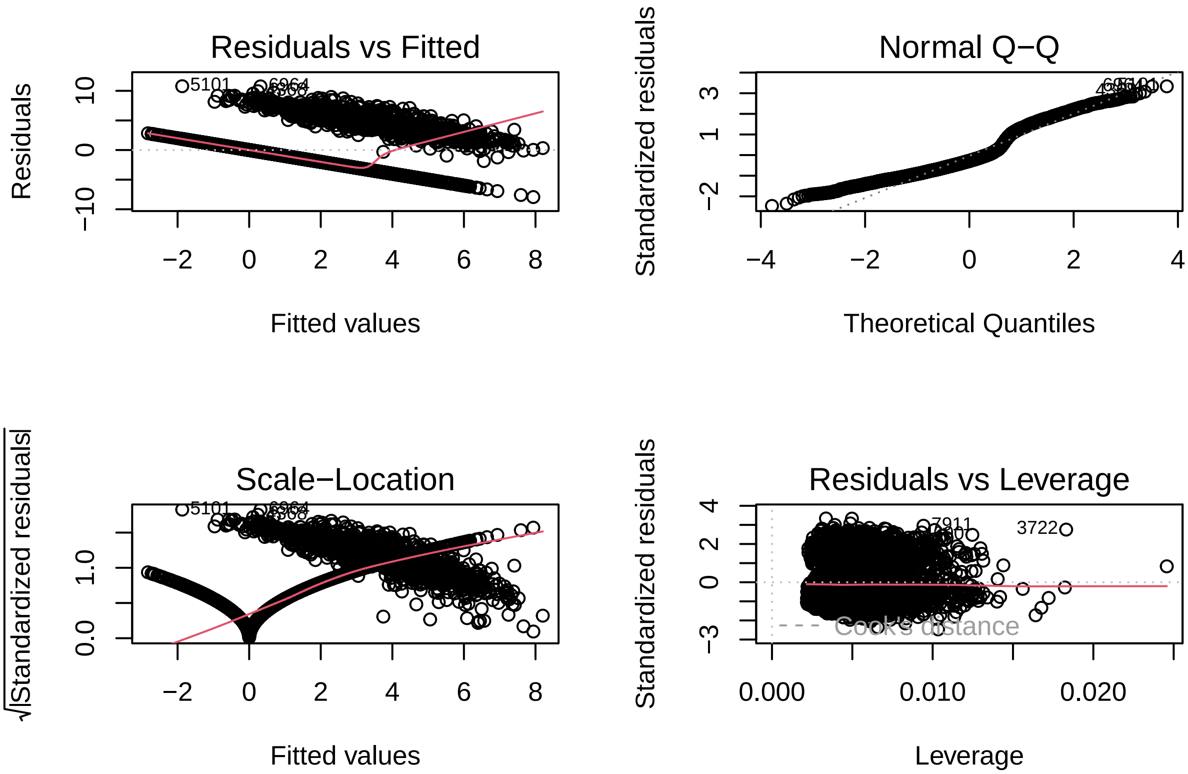
```
##
## Call:
## lm(formula = log1p(TARGET_AMT) ~ KIDSDRIV + AGE + YOJ + INCOME +
##     PARENT1 + HOME_VAL + MSTATUS + SEX + EDUCATION + JOB + TRAVTIME +
##     CAR_USE + BLUEBOOK + TIF + CAR_TYPE + OLDCLAIM + CLM_FREQ +
##     REVOKED + MVR_PTS + URBANICITY, data = train_data[, -(1)])
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -7.9388 -2.3417 -0.9085  2.0965 10.7763
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.714e-01  3.591e-01   1.870 0.061554 .
## KIDSDRIV                   4.555e-01  8.082e-02   5.635 1.82e-08 ***
## AGE                       -7.879e-03  5.254e-03  -1.499 0.133798
## YOJ                       -1.735e-02  1.184e-02  -1.466 0.142776
## INCOME                    -3.904e-06  1.446e-06  -2.700 0.006959 **
## PARENT1Yes                 7.329e-01  1.479e-01   4.954 7.44e-07 ***
## HOME_VAL                  -1.199e-06  4.696e-07  -2.554 0.010671 *
## MSTATUSYes                -6.130e-01  1.116e-01  -5.492 4.12e-08 ***
## SEXM                       1.973e-01  1.287e-01   1.533 0.125432
## EDUCATIONBelow High School  4.496e-01  1.547e-01   2.906 0.003668 **
## EDUCATIONHigh School       5.073e-01  1.182e-01   4.292 1.80e-05 ***
## EDUCATIONMasters           2.973e-01  1.709e-01   1.740 0.081879 .
## EDUCATIONPhD               4.595e-01  2.259e-01   2.035 0.041935 *
## JOBClerical                1.844e-01  1.543e-01   1.195 0.231940
## JOBDoctor                 -1.219e+00  3.444e-01  -3.540 0.000403 ***
## JOBHome Maker             -1.026e-01  2.136e-01  -0.480 0.631162
## JOBLawyer                 -5.267e-01  2.488e-01  -2.117 0.034314 *
## JOBManager                -1.271e+00  1.864e-01  -6.820 9.94e-12 ***
## JOBProfessional           -2.050e-01  1.689e-01  -1.214 0.224845
## JOBStudent                -1.691e-02  1.834e-01  -0.092 0.926524
## JOBUNSPECIFIED            -8.222e-01  2.559e-01  -3.213 0.001319 **
## TRAVTIME                   1.871e-02  2.558e-03   7.314 2.91e-13 ***
## CAR_USEPrivate            -9.926e-01  1.305e-01  -7.604 3.27e-14 ***
## BLUEBOOK                  -1.648e-05  6.890e-06  -2.392 0.016772 *
## TIF                       -6.187e-02  9.649e-03  -6.412 1.54e-10 ***
## CAR_TYPEPanel Truck        4.454e-01  2.216e-01   2.010 0.044479 *
## CAR_TYPEPickup             5.470e-01  1.359e-01   4.024 5.80e-05 ***
## CAR_TYPESports Car         1.271e+00  1.718e-01   7.398 1.56e-13 ***
## CAR_TYPESUV                9.291e-01  1.420e-01   6.544 6.45e-11 ***
## CAR_TYPEVan                6.380e-01  1.712e-01   3.726 0.000196 ***
## OLDCLAIM                  -1.909e-05  5.887e-06  -3.242 0.001191 **
## CLM_FREQ                   2.155e-01  4.384e-02   4.915 9.08e-07 ***
## REVOKEDYes                 1.258e+00  1.382e-01   9.101  < 2e-16 ***
## MVR_PTS                    1.927e-01  2.082e-02   9.255  < 2e-16 ***
## URBANICITYUrban            2.508e+00  1.108e-01  22.640  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.237 on 6495 degrees of freedom
## Multiple R-squared:  0.227,  Adjusted R-squared:  0.2229
## F-statistic: 56.09 on 34 and 6495 DF,  p-value: < 2.2e-16
```

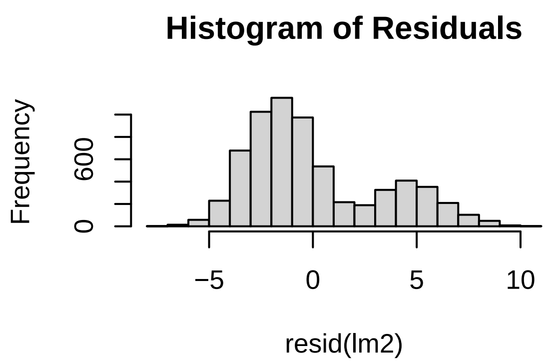
The second multiple regression (lm2), which incorporates log transformation of the response variable and stepwise feature selection, demonstrates an adjusted R-squared value of 0.2229, this suggests that it explains a slightly larger proportion of the variability in the target variable compared to the first model, indicating better predictive performance. The F-statistic of 56.09 with a p-value less than 2.2e-16 suggests that the overall model is statistically significant, implying that at least one predictor variable has a non-zero effect on the target amount.

Comparatively, model 2 (lm2) exhibits a slightly higher adjusted R-squared value and a larger F-statistic compared to model 1 (lm1), indicating a better fit and stronger overall predictive power. Therefore, the second model may provide more accurate predictions of the target amount compared to the initial analysis

```r
par(mfrow=c(2,2))
plot(lm2)
```




```r
hist(resid(lm2), main="Histogram of Residuals")
```



While the residuals vs fitted plot has shown a slight improvement, there is still some visible patterns, such as a slight funnel shape. The diagnostic QQ plot also reveals a large deviation from normal in the upper quantiles that heavily affects the results. The residuals vs leverage plot shows there are significant outliers that are also affecting model performance. Again, the log transformation of the histogram of the residuals appears to be more normal vs. without the log transformation that we have checked. We have decided not to include in our model since the log transformation has improved our model significantly.

MODEL SELECTION:

```
sum_lm1 <- summary(lm1)
RSS <- c(crossprod(lm1$residuals))
MSE <- RSS/length(lm1$residuals)
print(paste0("Mean Squared Error: ", MSE))
```

```
## [1] "Mean Squared Error: 10.4191997202696"
```

```
print(paste0("Root MSE: ", sqrt(MSE)))
```

```
## [1] "Root MSE: 3.22787851696275"
```

```
print(paste0("Adjusted R-squared: ", sum_lm1$adj.r.squared))
```

```
## [1] "Adjusted R-squared: 0.222659694503687"
```

```r
print(paste0("F-statistic: ",sum_lm1$fstatistic[1]))
```

```
## [1] "F-statistic: 51.5446701498941"
```

```r
sum_lm2 <- summary(lm2)
RSS <- c(crossprod(lm2$residuals))
MSE <- RSS/length(lm2$residuals)
print(paste0("Mean Squared Error: ", MSE))
```

```
## [1] "Mean Squared Error: 10.4204183833981"
```

```r
print(paste0("Root MSE: ", sqrt(MSE)))
```

```
## [1] "Root MSE: 3.22806728297261"
```

```r
print(paste0("Adjusted R-squared: ", sum_lm2$adj.r.squared))
```

```
## [1] "Adjusted R-squared: 0.222927864915498"
```

```r
print(paste0("F-statistic: ",sum_lm2$fstatistic[1]))
```

```
## [1] "F-statistic: 56.0897462833738"
```

From above, we can see that both models have very similar mean squared error, root mean squared error, and adjusted R-squared values, suggesting comparable predictive performance. However, Model 2 has a higher F-statistic compared to Model 1, indicating that Model 2 explains more variability in the target variable and is likely a better fit to the data.

Therefore, based on the F-statistic and adjusted R-squared, Model 2 appears to be the preferred choice for model selection. It offers slightly better explanatory power and potentially improved predictive performance compared to Model 1.Therefore, we will opt for model 2 (lm2) for our prediction of TARGET_AMT.

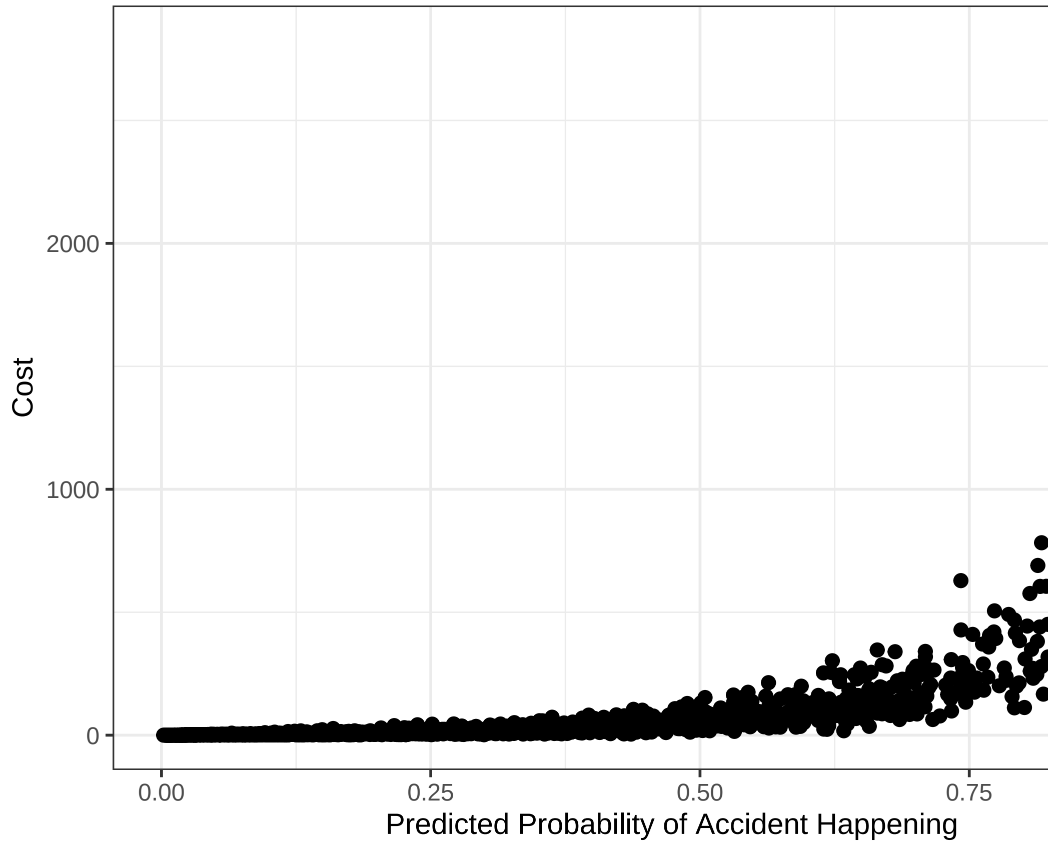
MULTIPLE REGRESSION- EVALUATION:

```r
clean_evaluation$TARGET_AMT <- exp(predict(lm2, newdata = clean_evaluation[,-(1:3)]))
```

Predictions:

```r
clean_evaluation <- cbind(clean_evaluation, A)
```

```r
ggplot(data = clean_evaluation,  mapping = aes(x= A, y = TARGET_AMT)) +
  geom_point(color = "black", size = 2)+labs(x="Predicted Probability of Accident Happening", y="Cost")
```



Visualization of Predictions
The graph above provides confirmation that our model consistently predicts higher costs as the probability of accidents increases.

5. CONCLUSION:

In this study, we aimed to understand the factors contributing to car crashes and predict repair costs using a dataset with 26 variables and over 8000 observations. We began by exploring and cleaning the data, then built two models: one for classification and one for regression. For classification, we trained three logistic regression models, prioritizing simplicity to avoid overfitting. Although Model 2 initially seemed simpler, we chose Model 3 for its balanced performance in sensitivity and specificity, resulting in the second-highest accuracy. For predicting repair costs, we created two multiple regression models. Model 2 showed better fit and predictive power than Model 1. Though our cost predictions weren't highly accurate, our model consistently predicted higher costs when the probability of accidents was higher. We believe our model could improve with predictor variable transformations. Additionally, we recommend exploring advanced modeling techniques like random forests or neural networks to enhance predictive performance further.

Appendix: