GROUP 2 HW4: Insurance - Data 621 Assignment 4

GROUP 2 MEMBERS: Banu Boopalan, Gregg Maloy, Alexander Moyse, Umais Siddiqui

10/26/2024

Contents

Data Exploration

Overview

In this assignment, you'll dive into a rich dataset of approximately 8,000 customer records from an auto insurance company. Each record represents a customer and includes two key response variables:

TARGET_FLAG - A binary indicator where a "1" signifies the customer was involved in a car crash, while a "0" means they were not. TARGET_AMT - This variable represents the cost incurred in the event of a crash. If there was no crash, this value is zero. If a crash occurred, this variable holds the associated monetary cost, which is greater than zero. Your goal is to develop predictive models that provide insights on two fronts:

The likelihood of a customer being involved in a car crash (using binary logistic regression). The potential cost of a crash, if it occurs (using multiple linear regression). For this task, you'll leverage the variables in the dataset—and any additional variables you derive from them—to create, train, and evaluate your models on a training dataset.

Crash Data Insights

Dataset Variables Overview:

This table outlines key attributes in our insurance dataset, detailing both the target variables and the predictor variables, along with their expected impacts on insurance outcomes. We aldo find a brief description of each variable in the dataset to help guide your exploratory analysis and feature engineering efforts.

Target Variables

Attribute	Description	Expected Impact
TARGET_FLAG	Indicates if the customer was involved in a crash (1 = Yes, 0 = No)	None at this stage
TARGET_AMT	Cost incurred in the event of a crash (0 if no crash)	None at this stage

---

Predictor Variables

Attribute	Description	Theoretical Influence
AGE	Driver's age	Young and very old drivers may have higher risks
BLUEBOOK	Vehicle market value	May affect payout size if a crash occurs
CAR_AGE	Vehicle's age	Possibly influences payout but unclear on crash likelihood
CAR_TYPE	Vehicle type	Potential influence on payout if a crash occurs
CAR_USE	Vehicle's primary use	Commercial usage may increase crash probability
CLM_FREQ	Claims made in past 5 years	More past claims may predict higher future claims
EDUCATION	Highest education level attained	Higher education might correlate with safer driving
HOMEKIDS	Number of children at home	Impact unknown
HOME_VAL	Value of home	Homeownership could correlate with responsible driving

Attribute	Description	Theoretical Influence
INCOME	Annual income	Wealthier individuals may experience fewer crashes
JOB	Employment category	White-collar jobs might suggest safer driving
KIDSDRIV	Number of young drivers in household	Teen drivers could increase crash risk
MSTATUS	Marital status	Married individuals may drive more cautiously
MVR_PTS	Points on motor vehicle record	Higher points suggest increased crash likelihood
OLDCLAIM	Cumulative claims in past 5 years	High past payouts may predict future claims
PARENT1	Single-parent household indicator	Impact unknown
RED_CAR	Indicator for a red car	Potential correlation with risky driving (myth)
REVOKED	Past license revocation (in last 7 years)	Suggests increased risk
SEX	Driver's gender	Myth suggests women may experience fewer crashes
TIF	Policy duration (years)	Long-term policyholders may have safer driving patterns
TRAVTIME	Commute duration	Longer commutes may indicate higher risk
URBANICITY	Urban or rural setting	Impact unknown
YOJ	Years in current job	Stable employment may suggest safer driving habits

The dataset includes 8,161 records with 23 feature variables and 2 target variables, providing detailed information on customers and their insurance claims history.

On preliminary inspection, we note that several columns contain issues such as incompatible punctuation in financial values, and categorical variables require conversion to factors with clearer labels.

```
## Rows: 8,161
## Columns: 26
## $ INDEX       <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 19, 20, 2~
## $ TARGET_FLAG <int> 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1~
## $ TARGET_AMT  <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 2946.000, 0.000, 4021.0~
## $ KIDSDRIV    <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ AGE         <int> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 43, 55, 53, 45~
## $ HOMEKIDS    <int> 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 2, 1~
## $ YOJ         <int> 11, 11, 10, 14, NA, 12, NA, NA, 10, 7, 14, 5, 11, 11, 0, 1~
## $ INCOME      <chr> "$67,349", "$91,449", "$16,039", "", "$114,986", "$125,301~
## $ PARENT1     <chr> "No", "No", "No", "No", "No", "Yes", "No", "No", "No", "No~
## $ HOME_VAL    <chr> "$0", "$257,252", "$124,191", "$306,251", "$243,925", "$0"~
## $ MSTATUS     <chr> "z_No", "z_No", "Yes", "Yes", "Yes", "z_No", "Yes", "Yes",~
## $ SEX         <chr> "M", "M", "z_F", "M", "z_F", "z_F", "z_F", "M", "z_F", "M"~
## $ EDUCATION   <chr> "PhD", "z_High School", "z_High School", "<High School", "~
## $ JOB         <chr> "Professional", "z_Blue Collar", "Clerical", "z_Blue Colla~
## $ TRAVTIME    <int> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36, 25, 64, 48,~
## $ CAR_USE     <chr> "Private", "Commercial", "Private", "Private", "Private", ~
## $ BLUEBOOK    <chr> "$14,230", "$14,940", "$4,010", "$15,440", "$18,000", "$17~
## $ TIF         <int> 11, 1, 4, 7, 1, 1, 1, 1, 1, 7, 1, 7, 7, 6, 1, 6, 6, 7, 4, ~
## $ CAR_TYPE    <chr> "Minivan", "Minivan", "z_SUV", "Minivan", "z_SUV", "Sports~
## $ RED_CAR     <chr> "yes", "yes", "no", "yes", "no", "no", "no", "yes", "no", ~
## $ OLDCLAIM    <chr> "$4,461", "$0", "$38,690", "$0", "$19,217", "$0", "$0", "$~
## $ CLM_FREQ    <int> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2~
```

```
## $ REVOKED     <chr> "No", "No", "No", "No", "Yes", "No", "No", "Yes", "No", "N~
## $ MVR_PTS     <int> 3, 0, 3, 0, 3, 0, 0, 10, 0, 1, 0, 0, 3, 3, 3, 0, 0, 0, 0, ~
## $ CAR_AGE     <int> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 10, 5, 13, 16,~
## $ URBANICITY  <chr> "Highly Urban/ Urban", "Highly Urban/ Urban", "Highly Urba~
```

```
##   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
```

```
##      INDEX        TARGET_FLAG       TARGET_AMT        KIDSDRIV
##  Min.   :    1   Min.   :0.0000   Min.   :     0   Min.   :0.0000
##  1st Qu.: 2559   1st Qu.:0.0000   1st Qu.:     0   1st Qu.:0.0000
##  Median : 5133   Median :0.0000   Median :     0   Median :0.0000
##  Mean   : 5152   Mean   :0.2638   Mean   :  1504   Mean   :0.1711
##  3rd Qu.: 7745   3rd Qu.:1.0000   3rd Qu.:  1036   3rd Qu.:0.0000
##  Max.   :10302   Max.   :1.0000   Max.   :107586   Max.   :4.0000
##
##       AGE           HOMEKIDS           YOJ          INCOME
##  Min.   :16.00   Min.   :0.0000   Min.   : 0.0   Length:8161
##  1st Qu.:39.00   1st Qu.:0.0000   1st Qu.: 9.0   Class :character
##  Median :45.00   Median :0.0000   Median :11.0   Mode  :character
##  Mean   :44.79   Mean   :0.7212   Mean   :10.5
##  3rd Qu.:51.00   3rd Qu.:1.0000   3rd Qu.:13.0
##  Max.   :81.00   Max.   :5.0000   Max.   :23.0
##  NA's   :6                        NA's   :454
##    PARENT1            HOME_VAL           MSTATUS              SEX
##  Length:8161        Length:8161        Length:8161        Length:8161
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
##
##
##
##
##   EDUCATION             JOB                TRAVTIME        CAR_USE
##  Length:8161        Length:8161        Min.   :  5.00   Length:8161
##  Class :character   Class :character   1st Qu.: 22.00   Class :character
##  Mode  :character   Mode  :character   Median : 33.00   Mode  :character
##                                        Mean   : 33.49
##                                        3rd Qu.: 44.00
##                                        Max.   :142.00
##
##    BLUEBOOK              TIF           CAR_TYPE           RED_CAR
##  Length:8161        Min.   : 1.000   Length:8161        Length:8161
##  Class :character   1st Qu.: 1.000   Class :character   Class :character
##  Mode  :character   Median : 4.000   Mode  :character   Mode  :character
##                     Mean   : 5.351
##                     3rd Qu.: 7.000
##                     Max.   :25.000
##
##    OLDCLAIM            CLM_FREQ        REVOKED             MVR_PTS
##  Length:8161        Min.   :0.0000   Length:8161        Min.   : 0.000
##  Class :character   1st Qu.:0.0000   Class :character   1st Qu.: 0.000
##  Mode  :character   Median :0.0000   Mode  :character   Median : 1.000
##                     Mean   :0.7986                      Mean   : 1.696
##                     3rd Qu.:2.0000                      3rd Qu.: 3.000
##                     Max.   :5.0000                      Max.   :13.000
##
##     CAR_AGE        URBANICITY
##  Min.   :-3.000   Length:8161
##  1st Qu.: 1.000   Class :character
##  Median : 8.000   Mode  :character
##  Mean   : 8.328
##  3rd Qu.:12.000
##  Max.   :28.000
##  NA's   :510

##   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        HOME_VAL
##  Min.   :0.0000   Min.   : 0.0   Min.   :     0   No :7084   Min.   :     0
##  1st Qu.:0.0000   1st Qu.: 9.0   1st Qu.: 28097   Yes:1077   1st Qu.:     0
##  Median :0.0000   Median :11.0   Median : 54028              Median :161160
##  Mean   :0.7212   Mean   :10.5   Mean   : 61898              Mean   :154867
##  3rd Qu.:1.0000   3rd Qu.:13.0   3rd Qu.: 85986              3rd Qu.:238724
##  Max.   :5.0000   Max.   :23.0   Max.   :367030              Max.   :885282
##                   NA's   :454    NA's   :445                 NA's   :464
##  MSTATUS    SEX                       EDUCATION              JOB
```

```
##  No :3267   F:4375   Bachelors           :2242   Blue Collar :1825
##  Yes:4894   M:3786   High School         :2330   Clerical    :1271
##                      Less than High School:1203   Professional:1117
##                      Masters             :1658   Manager     : 988
##                      PhD                 : 728   Lawyer      : 835
##                                                  Student     : 712
##                                                  (Other)     :1413
##     TRAVTIME           CAR_USE       BLUEBOOK          TIF
##  Min.   :  5.00   Commercial:3029   Min.   : 1500   Min.   : 1.000
##  1st Qu.: 22.00   Private   :5132   1st Qu.: 9280   1st Qu.: 1.000
##  Median : 33.00                     Median :14440   Median : 4.000
##  Mean   : 33.49                     Mean   :15710   Mean   : 5.351
##  3rd Qu.: 44.00                     3rd Qu.:20850   3rd Qu.: 7.000
##  Max.   :142.00                     Max.   :69740   Max.   :25.000
##
##        CAR_TYPE    RED_CAR       OLDCLAIM        CLM_FREQ      REVOKED
##  Minivan    :2145   no :5783   Min.   :    0   Min.   :0.0000   No :7161
##  Panel Truck: 676   yes:2378   1st Qu.:    0   1st Qu.:0.0000   Yes:1000
##  Pickup     :1389              Median :    0   Median :0.0000
##  Sports Car : 907              Mean   : 4037   Mean   :0.7986
##  SUV        :2294              3rd Qu.: 4636   3rd Qu.:2.0000
##  Van        : 750              Max.   :57037   Max.   :5.0000
##
##     MVR_PTS          CAR_AGE                    URBANICITY
##  Min.   : 0.000   Min.   :-3.000   Highly Rural/ Rural:1669
##  1st Qu.: 0.000   1st Qu.: 1.000   Highly Urban/ Urban:6492
##  Median : 1.000   Median : 8.000
##  Mean   : 1.696   Mean   : 8.328
##  3rd Qu.: 3.000   3rd Qu.:12.000
##  Max.   :13.000   Max.   :28.000
##                   NA's   :510
```

The updated data frame now comprises only numeric and factor columns. It is observed that the car age variable contains values less than 1, including negative values. These will be replaced with a mode value of 1 to ensure data integrity.

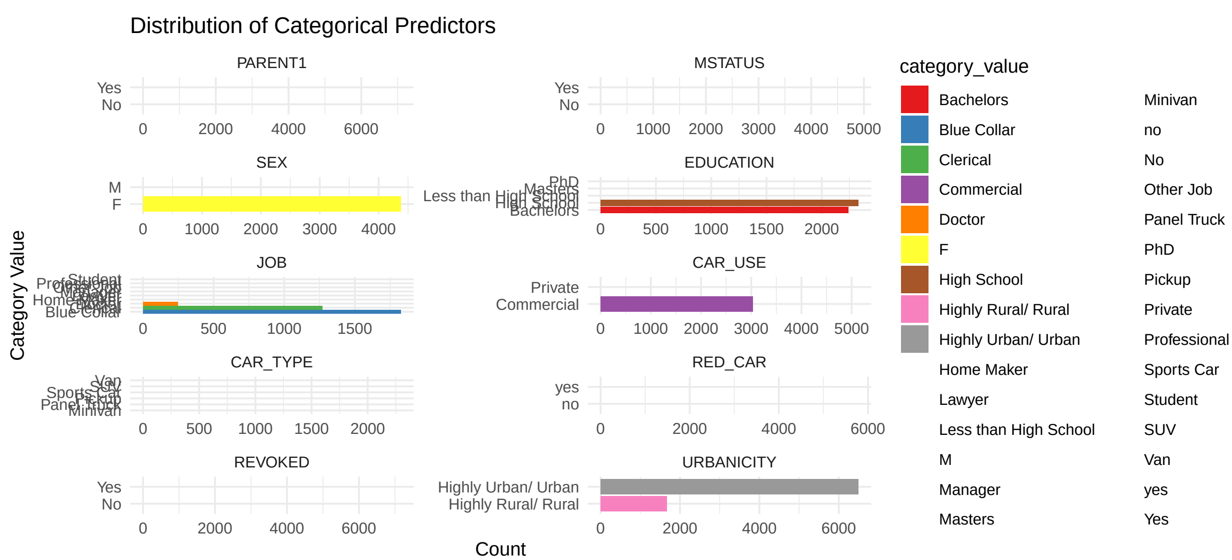
Categorical variables

```
## Exploring Categorical Features:

## Feature: PARENT1
## Levels: No, Yes
##
## Feature: MSTATUS
## Levels: No, Yes
##
## Feature: SEX
## Levels: F, M
##
## Feature: EDUCATION
## Levels: Bachelors, High School, Less than High School, Masters, PhD
##
## Feature: JOB
```

```
## Levels: Blue Collar, Clerical, Doctor, Home Maker, Lawyer, Manager, Other Job, Professional, Student
##
## Feature: CAR_USE
## Levels: Commercial, Private
##
## Feature: CAR_TYPE
## Levels: Minivan, Panel Truck, Pickup, Sports Car, SUV, Van
##
## Feature: RED_CAR
## Levels: no, yes
##
## Feature: REVOKED
## Levels: No, Yes
##
## Feature: URBANICITY
## Levels: Highly Rural/ Rural, Highly Urban/ Urban
```

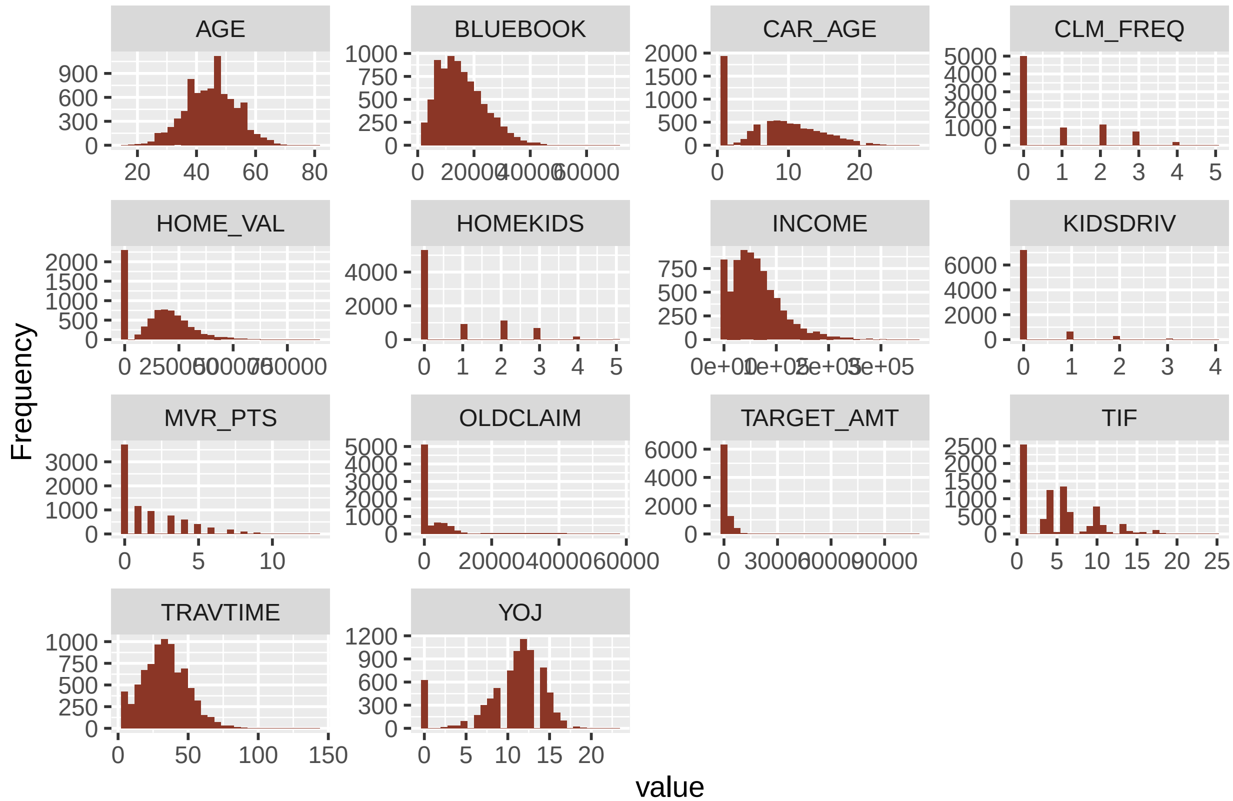
Upon examining the categorical variables, it is observed that the majority of the columns are binary in nature.

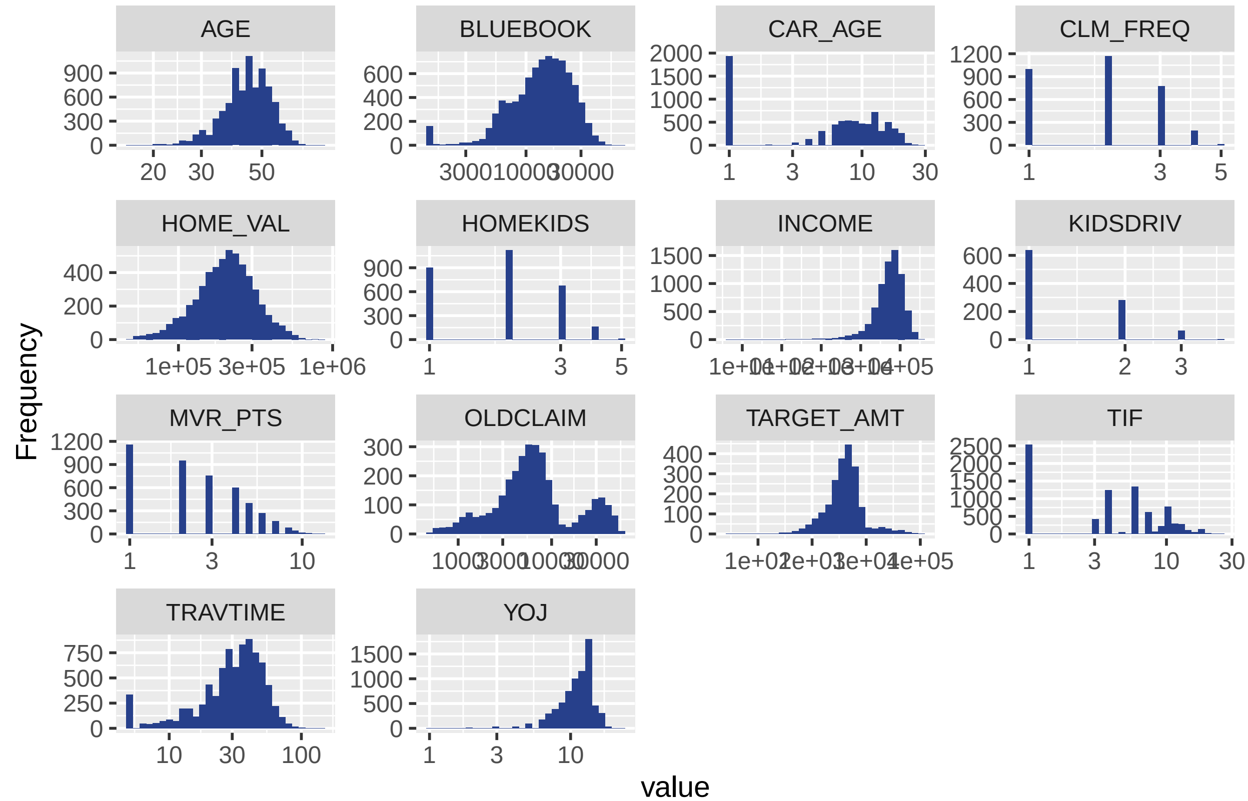
The following graphs illustrate the distribution of all categorical predictors.



Numeric Variables

The following two graphs illustrate the distribution of the numeric variables in our dataset. The first set of histograms, represented in red, displays the distributions on a normal scale, while the second set, depicted in blue, presents the distributions on a log10 scale. Notably, many numeric variables exhibit a mode value of zero, which may warrant further investigation.

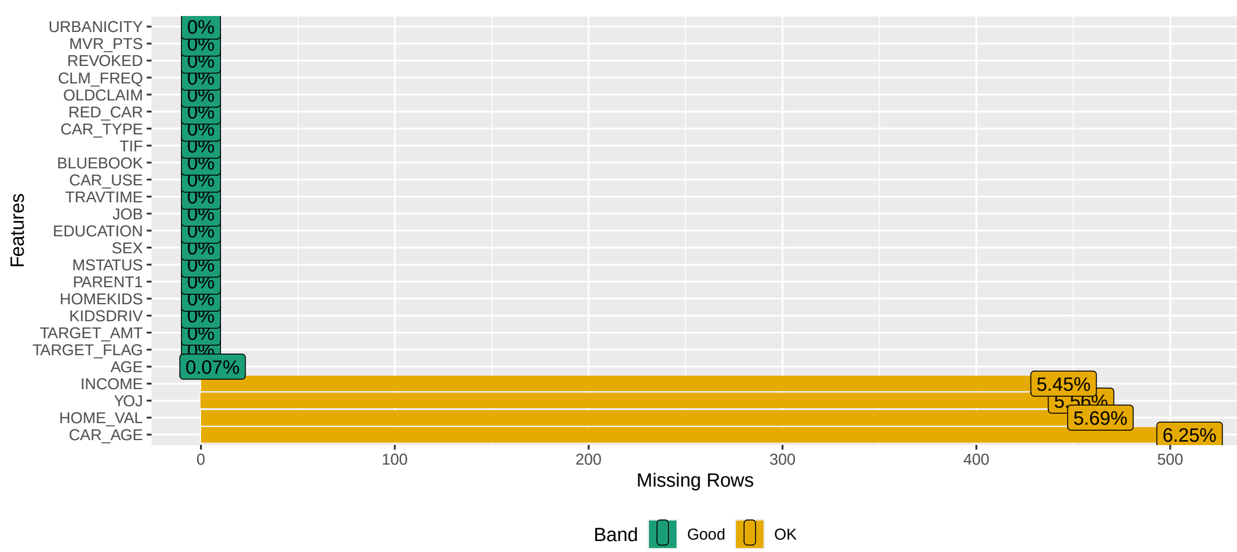

AGE
BLUEBOOK
CAR_AGE
CLM_FREQ
HOME_VAL
HOMEKIDS
INCOME
KIDSDRIV
MVR_PTS
OLDCLAIM
TARGET_AMT
TIF
TRAVTIME
YOJ
Frequency
value



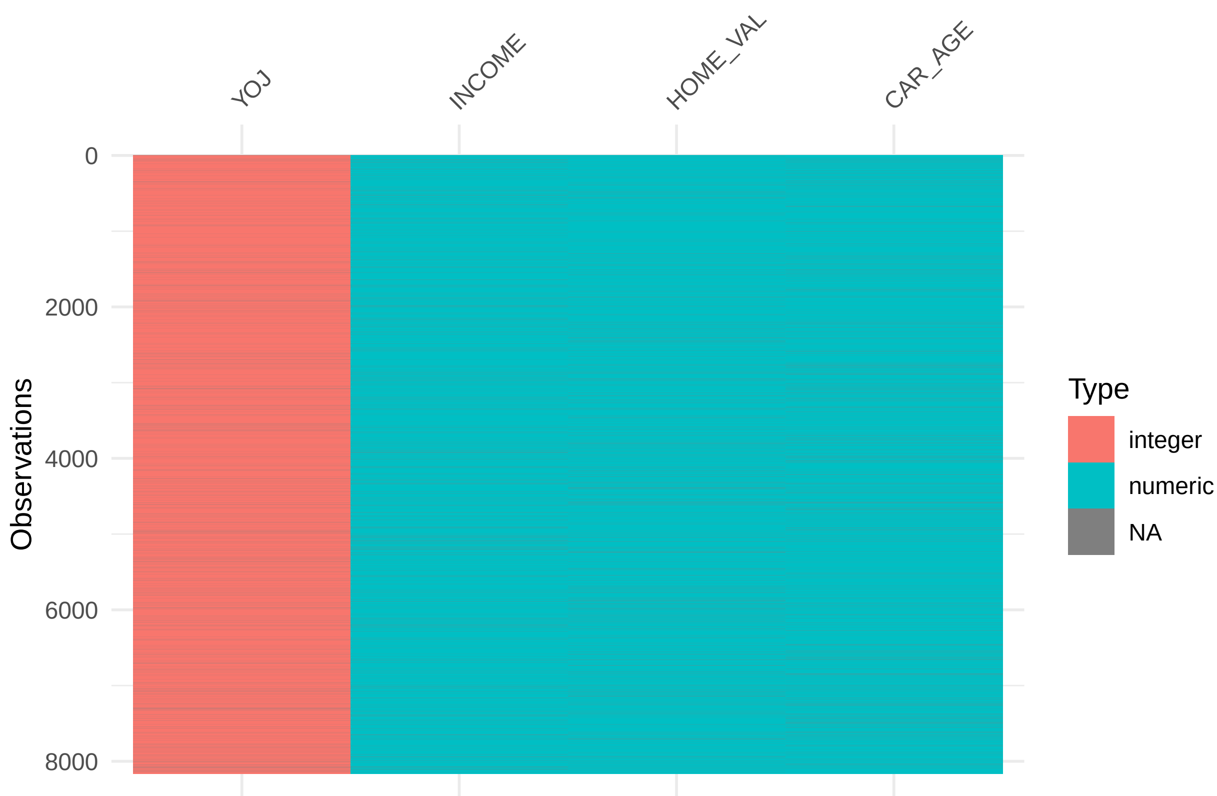
Assessment of Incomplete Data

This section identifies columns within the dataset that contain missing values, denoted as NA:

```
##   AGE YOJ INCOME HOME_VAL CAR_AGE
## 1   6 454    445      464     510
```



```
## TARGET_FLAG  TARGET_AMT    KIDSDRIV         AGE    HOMEKIDS         YOJ
##       0.000       0.000       0.000       0.001       0.000       0.056
##      INCOME     PARENT1    HOME_VAL     MSTATUS         SEX   EDUCATION
##       0.055       0.000       0.057       0.000       0.000       0.000
##         JOB    TRAVTIME     CAR_USE    BLUEBOOK         TIF    CAR_TYPE
##       0.000       0.000       0.000       0.000       0.000       0.000
##     RED_CAR    OLDCLAIM    CLM_FREQ     REVOKED     MVR_PTS     CAR_AGE
##       0.000       0.000       0.000       0.000       0.000       0.062
##  URBANICITY
##       0.000
```



The analysis reveals that five variables contain missing values. However, there does not appear to be a discernible pattern associated with these missing entries, which suggests they are likely missing at random (MAR). This conclusion allows us to proceed with standard imputation techniques or analyses without significant concern regarding bias introduced by the missing data.

Handling Missing Values And Correlation Analysis

Multiple Imputation by Chained Equations (MICE) is a powerful method for handling missing data, as it generates multiple complete datasets by predicting missing values based on other available data. This method accounts for uncertainty in the imputations and allows for more reliable statistical inference.

```
## [1] "Missing Values Before Imputation:"
```

```
## TARGET_AMT        AGE        YOJ     INCOME   HOME_VAL   TRAVTIME   BLUEBOOK
```

```
##          0          6        454        445        464          0          0
##        TIF   OLDCLAIM   CLM_FREQ    MVR_PTS    CAR_AGE
##          0          0          0          0        510
```

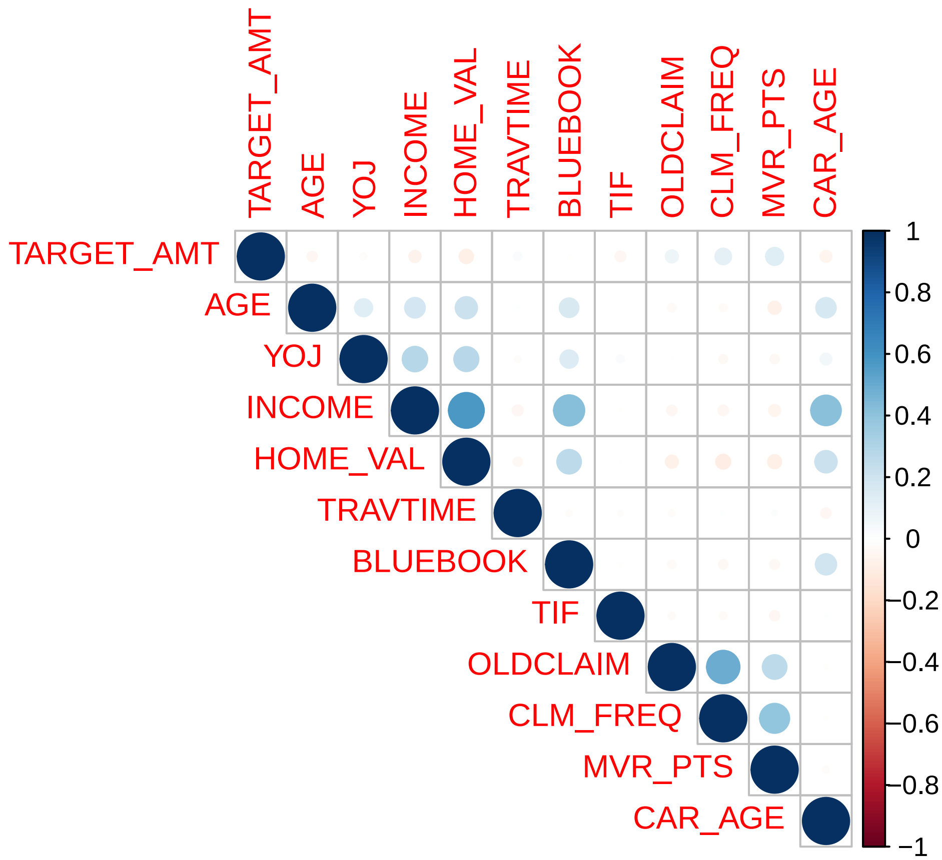
```
## 
##  iter imp variable
##   1   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
```

```
## 
##  iter imp variable
##   1   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
```

```
##   5   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
```

```
## [1] "Missing Values After Imputation:"
```

```
## TARGET_AMT        AGE        YOJ     INCOME   HOME_VAL   TRAVTIME   BLUEBOOK
##          0          0          0          0          0          0          0
##        TIF   OLDCLAIM   CLM_FREQ    MVR_PTS    CAR_AGE
##          0          0          0          0          0
```

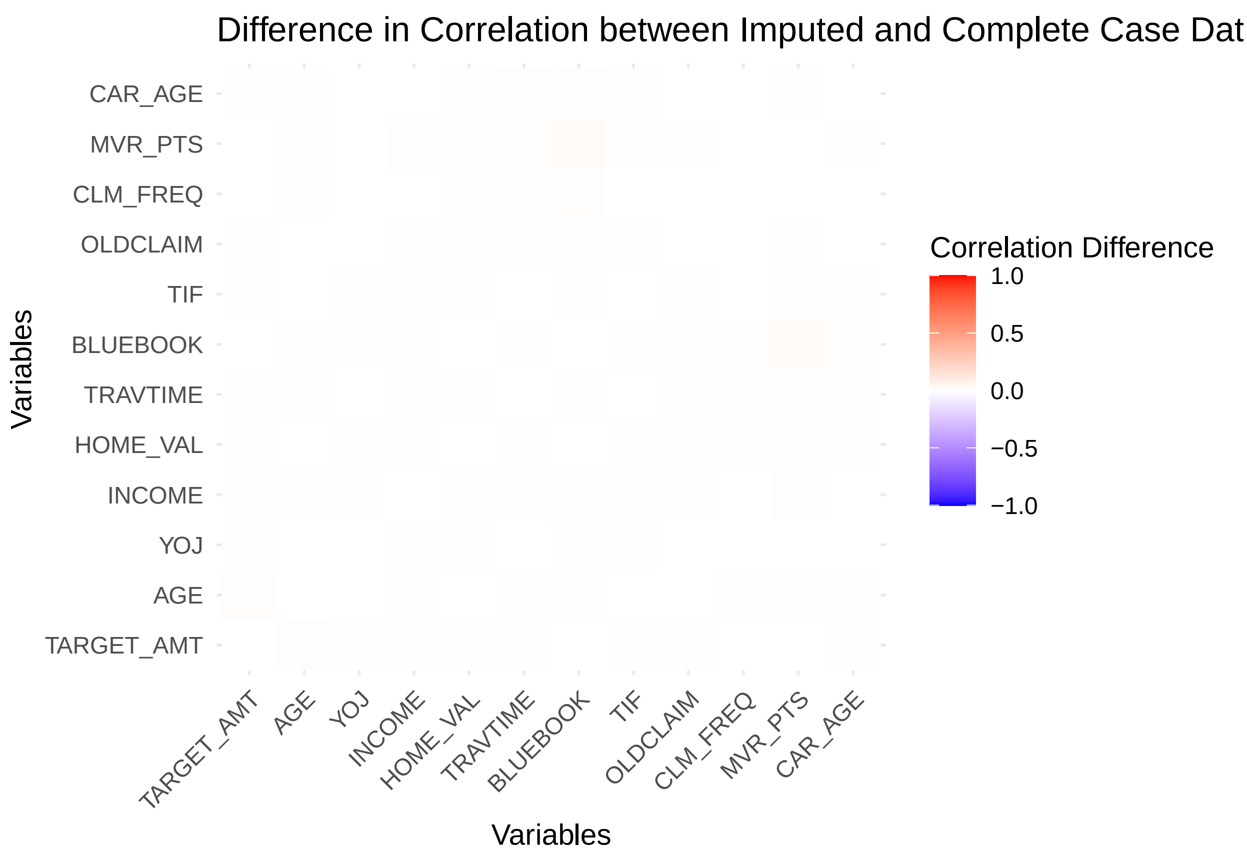



```
## [1] "Correlation Matrix for Complete Case Analysis:"
```

```
##              TARGET_AMT          AGE          YOJ        INCOME    HOME_VAL
## TARGET_AMT  1.000000000 -0.052348528 -0.022196571 -0.0562601493 -0.09056112
## AGE        -0.052348528  1.000000000  0.137847876  0.1876862059  0.21598562
## YOJ        -0.022196571  0.137847876  1.000000000  0.2783277152  0.26980907
## INCOME     -0.056260149  0.187686206  0.278327715  1.0000000000  0.57970674
## HOME_VAL   -0.090561124  0.215985625  0.269809074  0.5797067363  1.00000000
## TRAVTIME    0.032287806  0.007807727 -0.015740963 -0.0413200825 -0.03014163
## BLUEBOOK   -0.003183645  0.171170247  0.136335894  0.4332521829  0.26161690
## TIF        -0.041860052  0.000408708  0.030813700  0.0007376252 -0.00460570
## OLDCLAIM    0.080067386 -0.030707066  0.001634368 -0.0377131052 -0.05863833
```

```
## CLM_FREQ    0.116939123 -0.027125254 -0.028669411 -0.0451604051 -0.09695212
## MVR_PTS     0.137030840 -0.075556608 -0.035432609 -0.0709892627 -0.09418684
## CAR_AGE    -0.062828101  0.184019005  0.057768248  0.4117386242  0.21531374
##               TRAVTIME     BLUEBOOK           TIF     OLDCLAIM     CLM_FREQ
## TARGET_AMT  0.032287806 -0.003183645 -0.0418600523  0.080067386  0.116939123
## AGE         0.007807727  0.171170247  0.0004087080 -0.030707066 -0.027125254
## YOJ        -0.015740963  0.136335894  0.0308136996  0.001634368 -0.028669411
## INCOME     -0.041320082  0.433252183  0.0007376252 -0.037713105 -0.045160405
## HOME_VAL   -0.030141625  0.261616901 -0.0046056998 -0.058638327 -0.096952119
## TRAVTIME    1.000000000 -0.010979136 -0.0117716399 -0.022707967  0.009510331
## BLUEBOOK   -0.010979136  1.000000000  0.0045237917 -0.032654587 -0.046002944
## TIF        -0.011771640  0.004523792  1.0000000000 -0.018249702 -0.023758956
## OLDCLAIM   -0.022707967 -0.032654587 -0.0182497019  1.000000000  0.494017156
## CLM_FREQ    0.009510331 -0.046002944 -0.0237589564  0.494017156  1.000000000
## MVR_PTS     0.003815401 -0.061216939 -0.0380976659  0.272706265  0.397847352
## CAR_AGE    -0.030726192  0.185550420  0.0124643954 -0.010610234 -0.006339303
##                MVR_PTS      CAR_AGE
## TARGET_AMT  0.137030840 -0.062828101
## AGE        -0.075556608  0.184019005
## YOJ        -0.035432609  0.057768248
## INCOME     -0.070989263  0.411738624
## HOME_VAL   -0.094186838  0.215313740
## TRAVTIME    0.003815401 -0.030726192
## BLUEBOOK   -0.061216939  0.185550420
## TIF        -0.038097666  0.012464395
## OLDCLAIM    0.272706265 -0.010610234
## CLM_FREQ    0.397847352 -0.006339303
## MVR_PTS     1.000000000 -0.023995843
## CAR_AGE    -0.023995843  1.000000000

## [1] "Correlation Matrix for Imputed Data:"

##             TARGET_AMT           AGE          YOJ       INCOME      HOME_VAL
## TARGET_AMT  1.000000000 -0.0418264191 -0.017860070 -0.060983939 -0.0861878936
## AGE        -0.041826419  1.0000000000  0.138761497  0.182928284  0.2143812514
## YOJ        -0.017860070  0.1387614968  1.000000000  0.282659508  0.2733640677
## INCOME     -0.060983939  0.1829282842  0.282659508  1.000000000  0.5723473522
## HOME_VAL   -0.086187894  0.2143812514  0.273364068  0.572347352  1.0000000000
## TRAVTIME    0.027987016  0.0053547772 -0.016038747 -0.048890357 -0.0352608342
## BLUEBOOK   -0.004699523  0.1651777923  0.142660165  0.428970852  0.2630568135
## TIF        -0.046480831 -0.0003363674  0.024330425 -0.002846146  0.0006303218
## OLDCLAIM    0.070953287 -0.0297096301  0.001866237 -0.042264940 -0.0701071116
## CLM_FREQ    0.116419159 -0.0239127328 -0.030361314 -0.044365798 -0.0920016863
## MVR_PTS     0.137865509 -0.0717218955 -0.034559684 -0.058716119 -0.0830885507
## CAR_AGE    -0.058658346  0.1791602948  0.057592905  0.413684204  0.2182023139
##               TRAVTIME     BLUEBOOK           TIF     OLDCLAIM     CLM_FREQ
## TARGET_AMT  0.027987016 -0.004699523 -0.0464808306  0.070953287  0.116419159
## AGE         0.005354777  0.165177792 -0.0003363674 -0.029709630 -0.023912733
## YOJ        -0.016038747  0.142660165  0.0243304249  0.001866237 -0.030361314
## INCOME     -0.048890357  0.428970852 -0.0028461456 -0.042264940 -0.044365798
## HOME_VAL   -0.035260834  0.263056814  0.0006303218 -0.070107112 -0.092001686
## TRAVTIME    1.000000000 -0.017001298 -0.0116046256 -0.019267169  0.006560211
## BLUEBOOK   -0.017001298  1.000000000 -0.0054245723 -0.029517568 -0.036341497
## TIF        -0.011604626 -0.005424572  1.0000000000 -0.021958198 -0.023022955
```

```
## OLDCLAIM   -0.019267169 -0.029517568 -0.0219581980  1.000000000  0.495130810
## CLM_FREQ    0.006560211 -0.036341497 -0.0230229550  0.495130810  1.000000000
## MVR_PTS     0.010598511 -0.039130846 -0.0410457340  0.264485025  0.396638373
## CAR_AGE    -0.042936990  0.195606786  0.0058816228 -0.009906046 -0.005909096
##                MVR_PTS      CAR_AGE
## TARGET_AMT  0.13786551 -0.058658346
## AGE        -0.07172190  0.179160295
## YOJ        -0.03455968  0.057592905
## INCOME     -0.05871612  0.413684204
## HOME_VAL   -0.08308855  0.218202314
## TRAVTIME    0.01059851 -0.042936990
## BLUEBOOK   -0.03913085  0.195606786
## TIF        -0.04104573  0.005881623
## OLDCLAIM    0.26448503 -0.009906046
## CLM_FREQ    0.39663837 -0.005909096
## MVR_PTS     1.00000000 -0.018823869
## CAR_AGE    -0.01882387  1.000000000
```



After completing the data, we have calculated the correlation matrix on the fully imputed dataset. This provides a more accurate representation of the relationships between variables without the bias that could be introduced by simple imputation methods.

It is evident that there are notable positive correlations among the following variables:

Income and Home Value Income and Bluebook Value Income and Car Age Claim Frequency and Old Claims Claim Frequency and MVR Points

The heatmap provides a visual representation of the differences in correlations between the imputed data and complete case data, helping to understand the impact of the missing data handling method.

Data Preparation for Multiple Linear Regression

Removing TARGET_FLAG

Since, for multiple linear regression our objective is to predict the monetary amount of how much it will cost in the event of a crash, we will exclude the TARGET_FLAG variable from our analysis.

Handling Missing Data - Multiple Linear Regression

Before proceeding with imputation, let's assess the missing values in our dataset. We will then handle the missing data using multiple imputation, which is a more robust method than simply replacing missing values with the median.

```
## [1] "Missing Values Before Imputation:"
```

```
## TARGET_AMT   KIDSDRIV        AGE   HOMEKIDS        YOJ     INCOME    PARENT1
##          0          0          5          0        123        110          0
##   HOME_VAL    MSTATUS        SEX  EDUCATION        JOB   TRAVTIME    CAR_USE
##        121          0          0          0          0          0          0
##   BLUEBOOK        TIF   CAR_TYPE    RED_CAR   OLDCLAIM   CLM_FREQ    REVOKED
##          0          0          0          0          0          0          0
##    MVR_PTS    CAR_AGE URBANICITY
##          0        142          0
```

```
##
##  iter imp variable
##   1   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   1   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   2   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   3   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   4   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   1  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   2  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
```

```
##   5   3  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   4  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
##   5   5  AGE  YOJ  INCOME  HOME_VAL  CAR_AGE
```

```
## [1] "Missing Values After Imputation:"
```

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

Transformatiions - Multiple Linear Regression

We will be performing transformations and create histograms for several variables, which helps visualize the effect of the transformations on data distribution. Here's a breakdown of how these transformations aid in model building and potential outcomes:

Handling Skewness:

Many of these variables (e.g., INCOME, HOME_VAL, OLDCLAIM) may be right-skewed due to outliers or a large range of values. Transformations like log, square root, and Yeo-Johnson help normalize the distribution, reducing skewness. Normalized distributions (closer to normal) are beneficial for regression-based models, as they assume linear relationships and normally distributed residuals.

Improving Model Fit:

Log and Square Root transformations compress the range of values, which can make the data easier for linear models to handle. For instance, high-income values may dominate the predictive power of INCOME if not transformed. Box-Cox and Yeo-Johnson transformations (which automatically choose an optimal transformation) can help produce more linearly related predictors, which improves linear regression model accuracy.

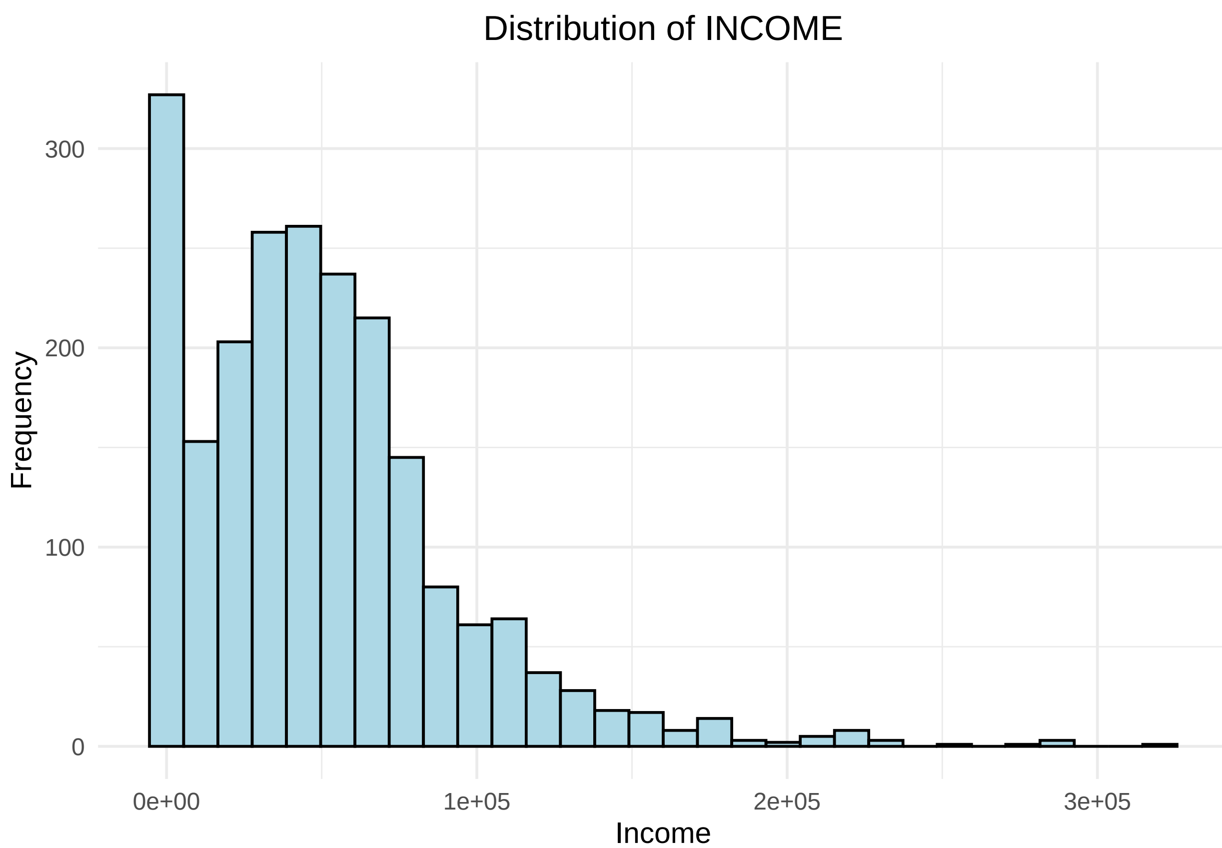
Comparing the Effect of Transformations:

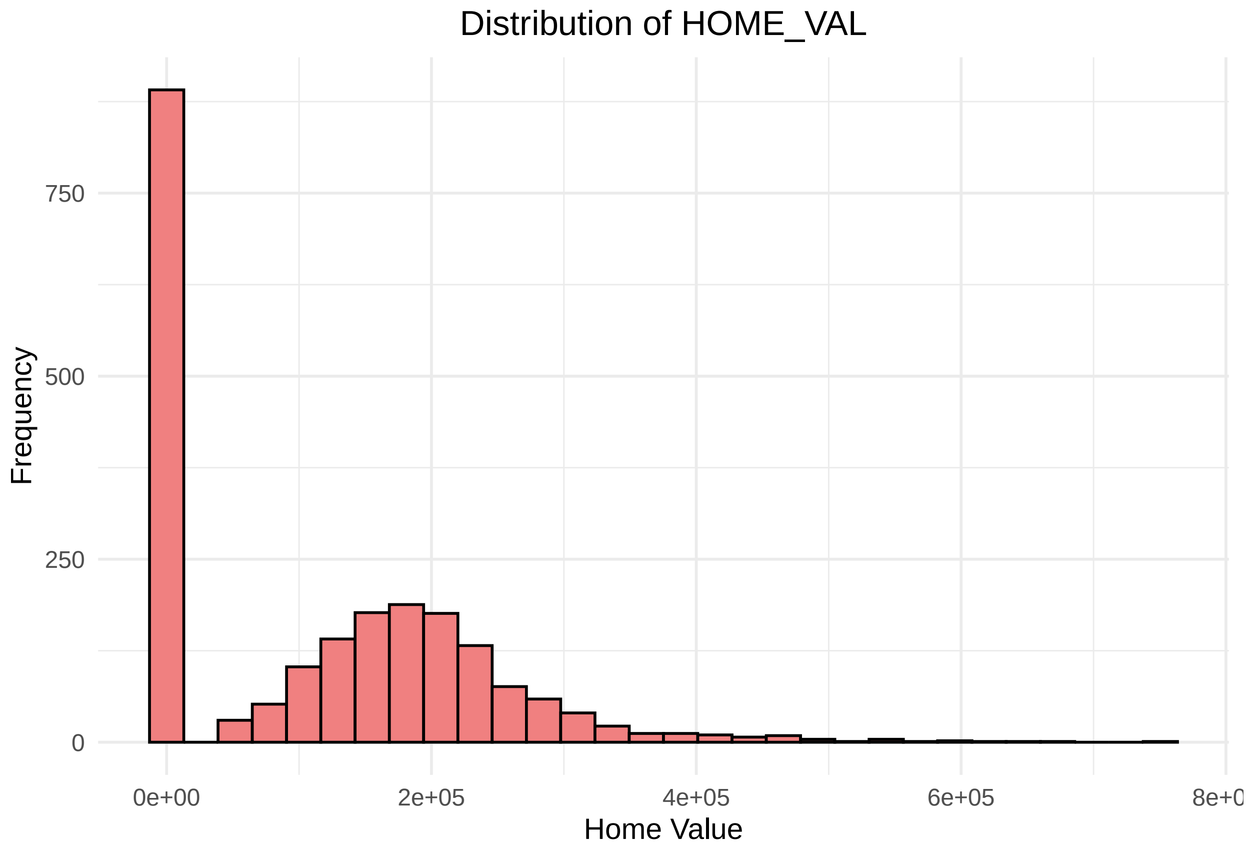
Creating side-by-side histograms allows you to compare the original and transformed distributions. This visual analysis is important for selecting the transformation that brings the distribution closest to normality, which can ultimately improve the performance and interpretability of the model.

Categorizing Continuous Variables:

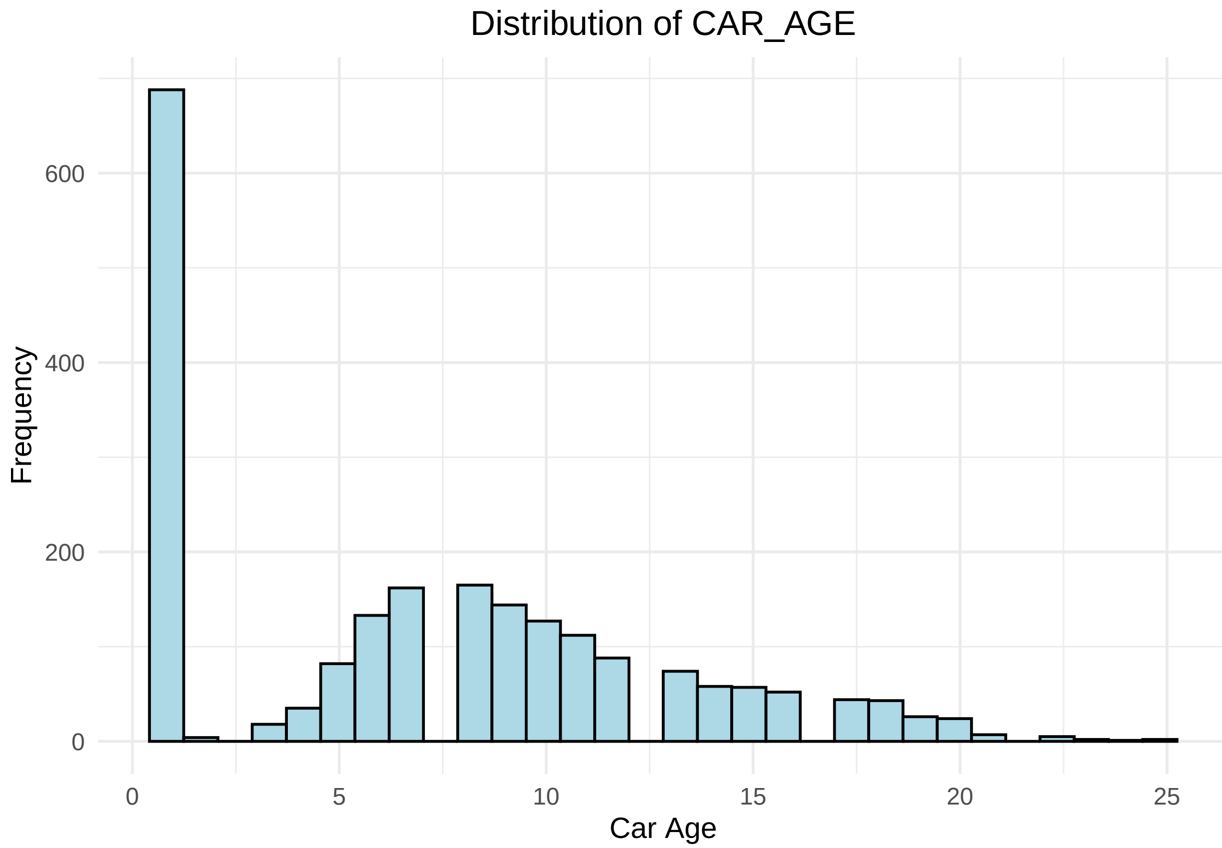
The cut function is used to create binned categories for TIF (Years with Policy) and MVR_PTS (Driving Record Points), which converts continuous variables into categorical bins. This is useful if there are distinct groups within the data that are meaningful (e.g., "Less than 1 year" in TIF). Using Transformed Variables for Modeling After determining the most effective transformation for each variable, we can replace the original variables with the transformed ones in our model. However, it's also useful to keep both versions to allow for comparison in model performance. Here's how to proceed with this:

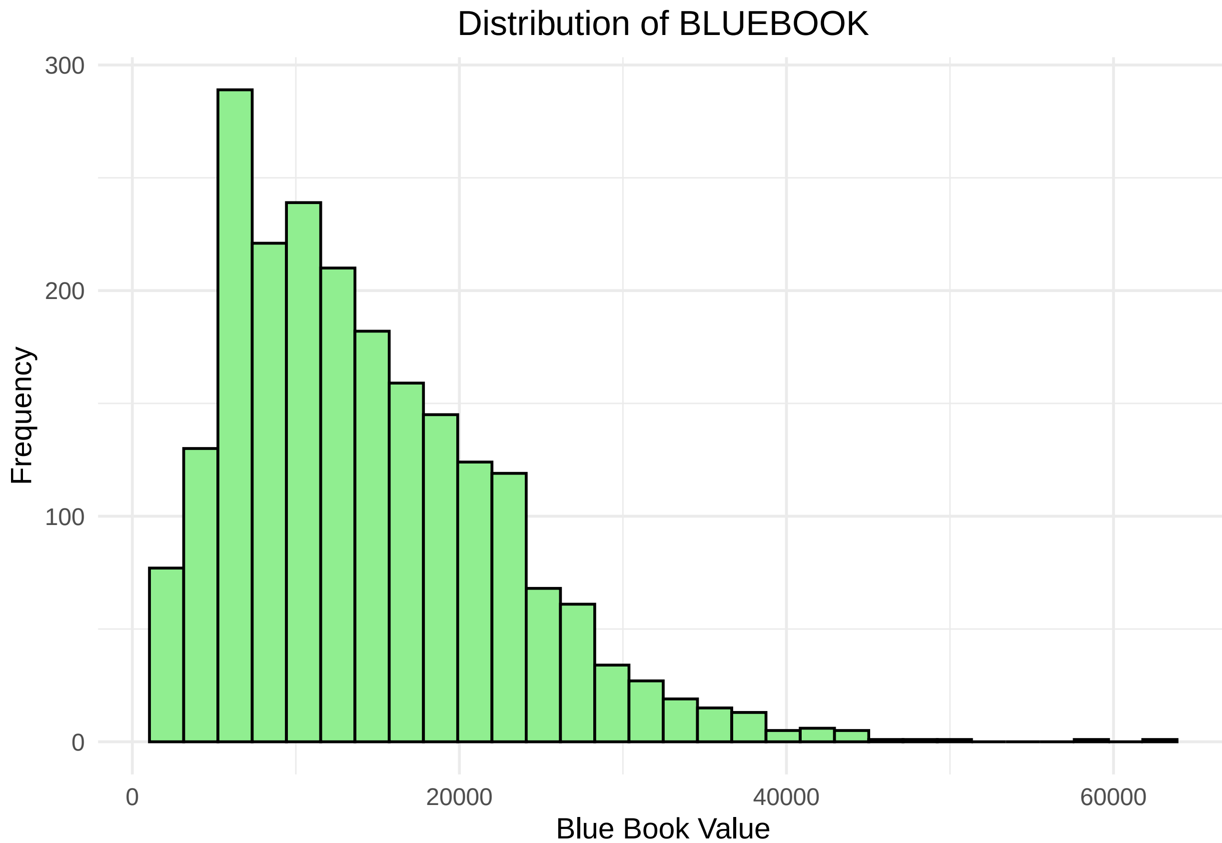
r

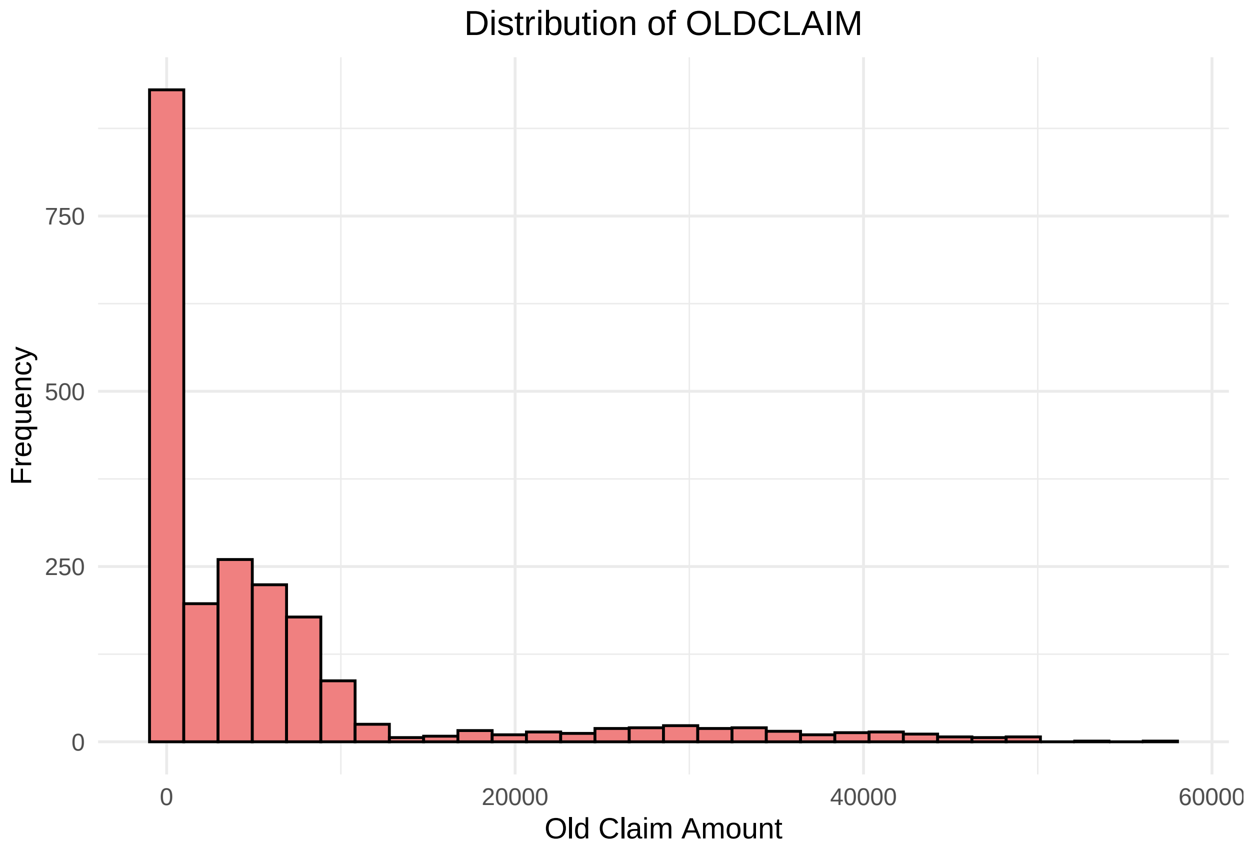

Distribution of INCOME
Frequency
300
200
100
0
0e+00
1e+05
2e+05
3e+05
Income

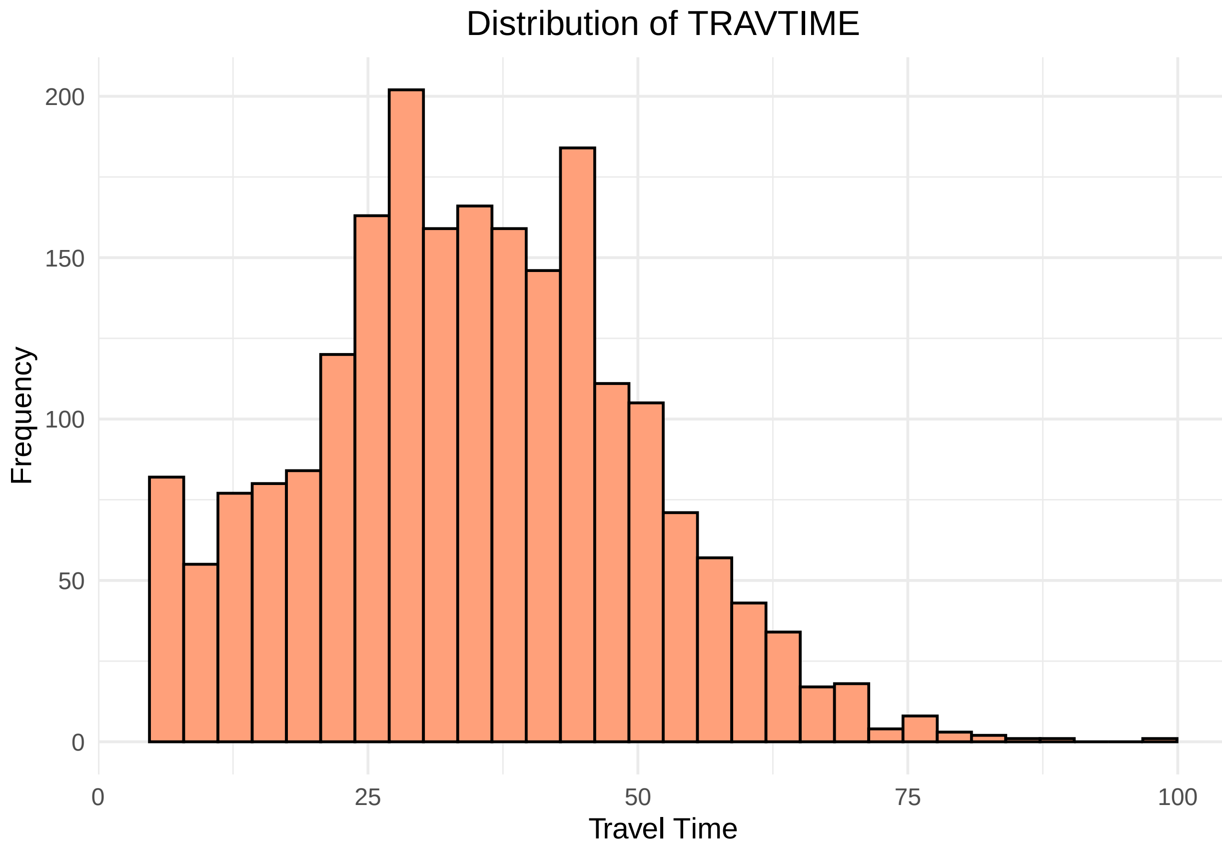


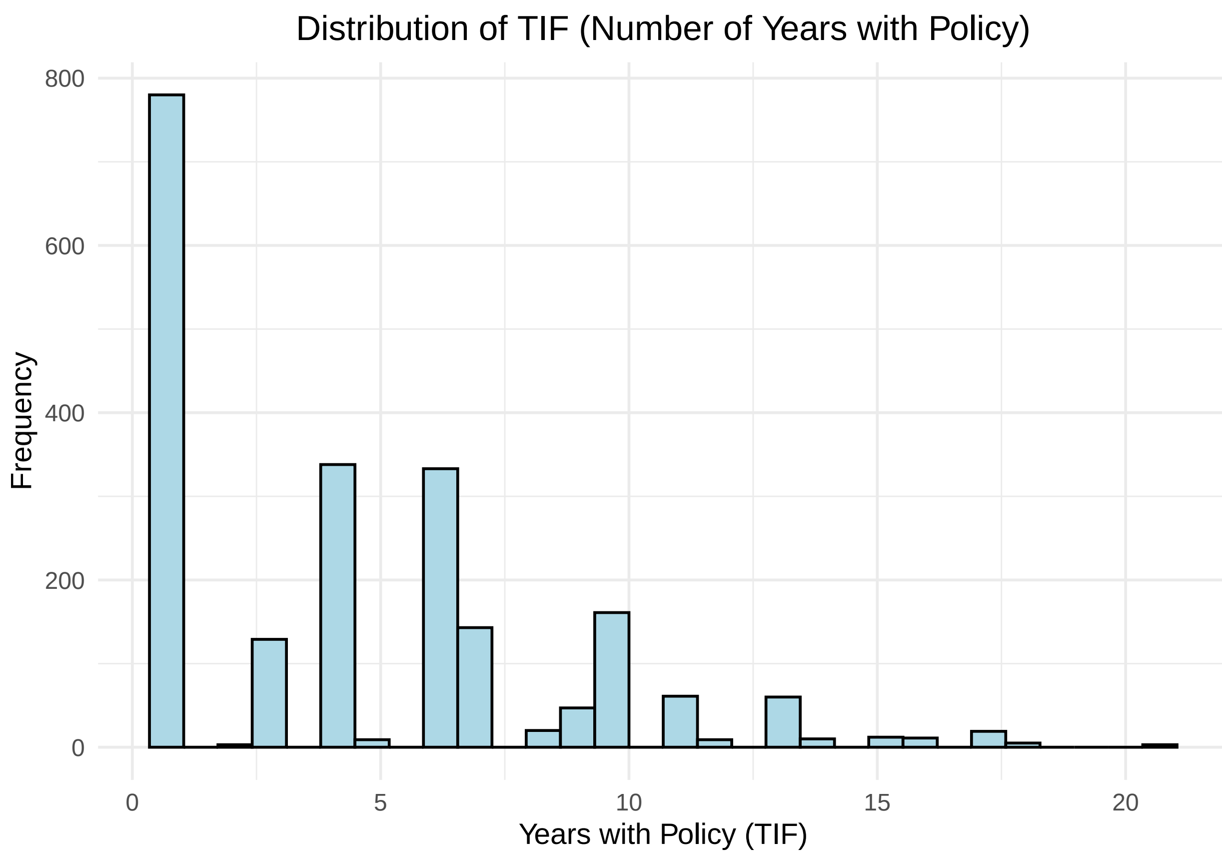
Distribution of HOME_VAL
Frequency
750
500
250
0
0e+00
2e+05
4e+05
6e+05
8e+0
Home Value

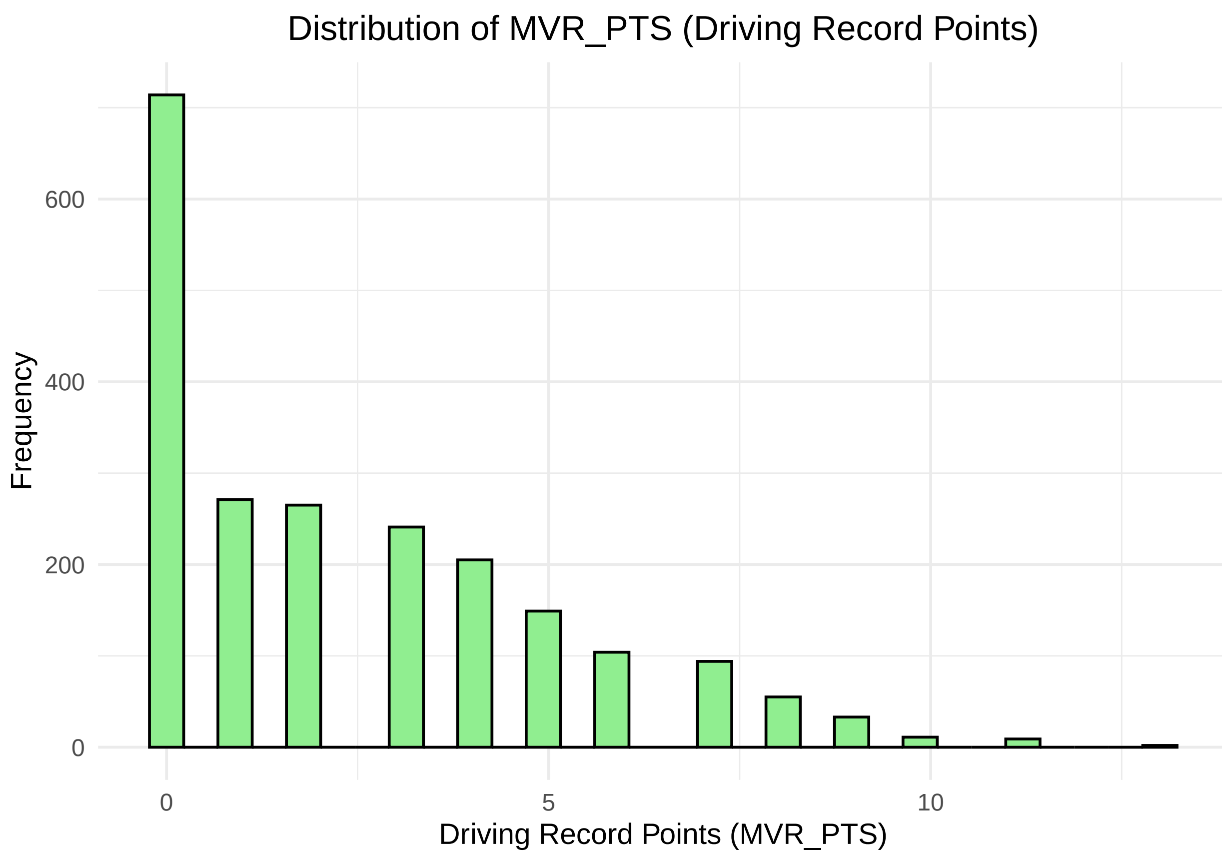

Distribution of CAR_AGE
Frequency
0
200
400
600
0
5
10
15
20
25
Car Age

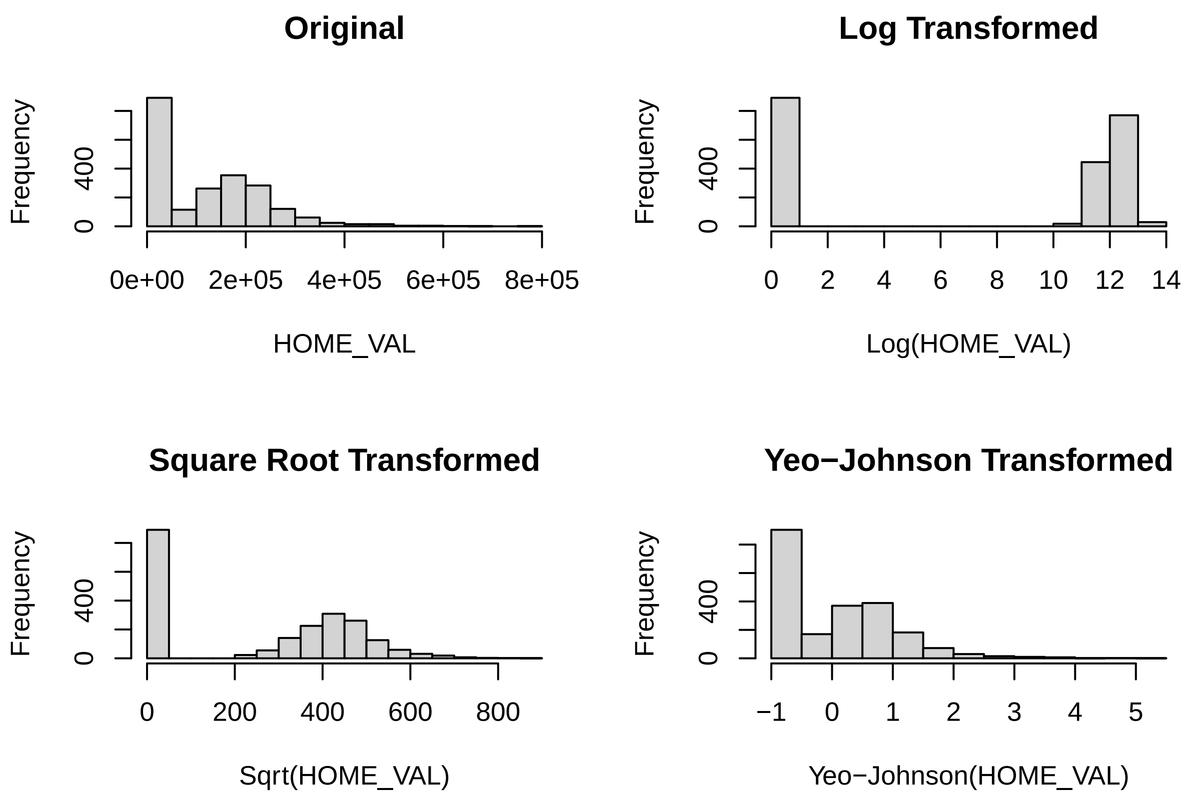

Distribution of BLUEBOOK
Frequency
300
200
100
0
0
20000
40000
60000
Blue Book Value

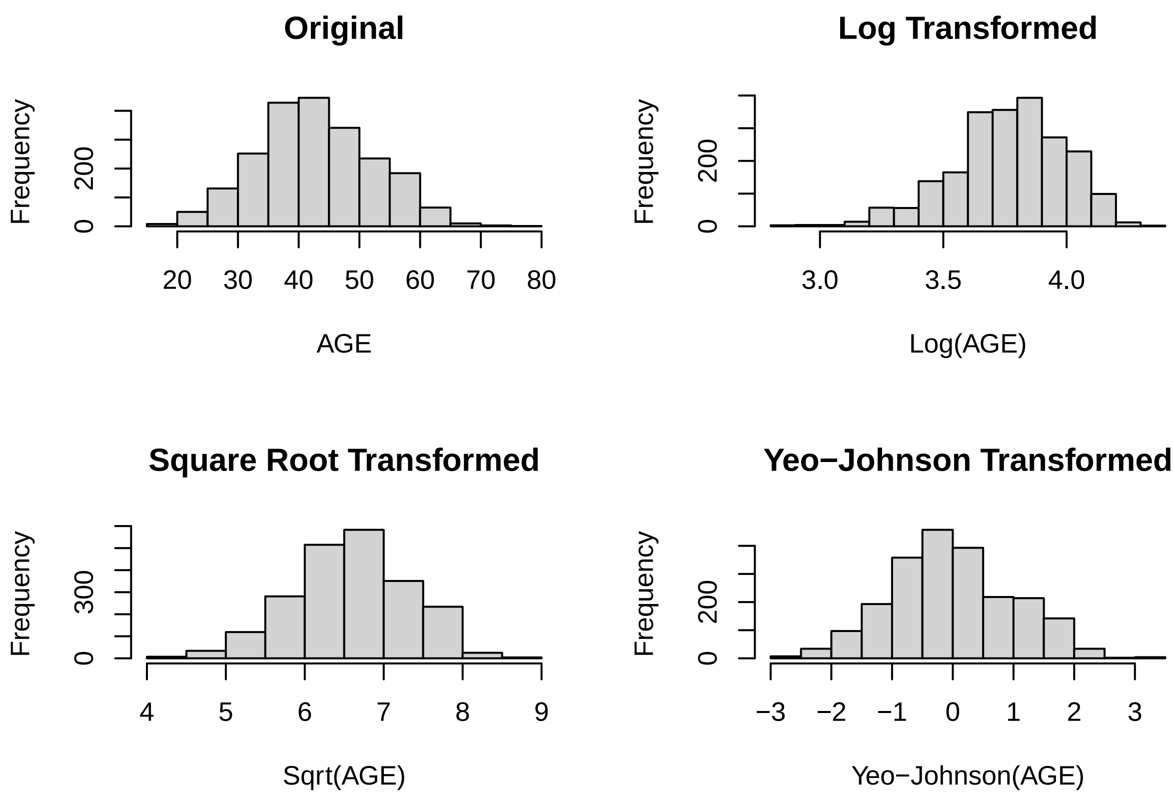

Distribution of OLDCLAIM
Frequency
750
500
250
0
0
20000
40000
60000
Old Claim Amount

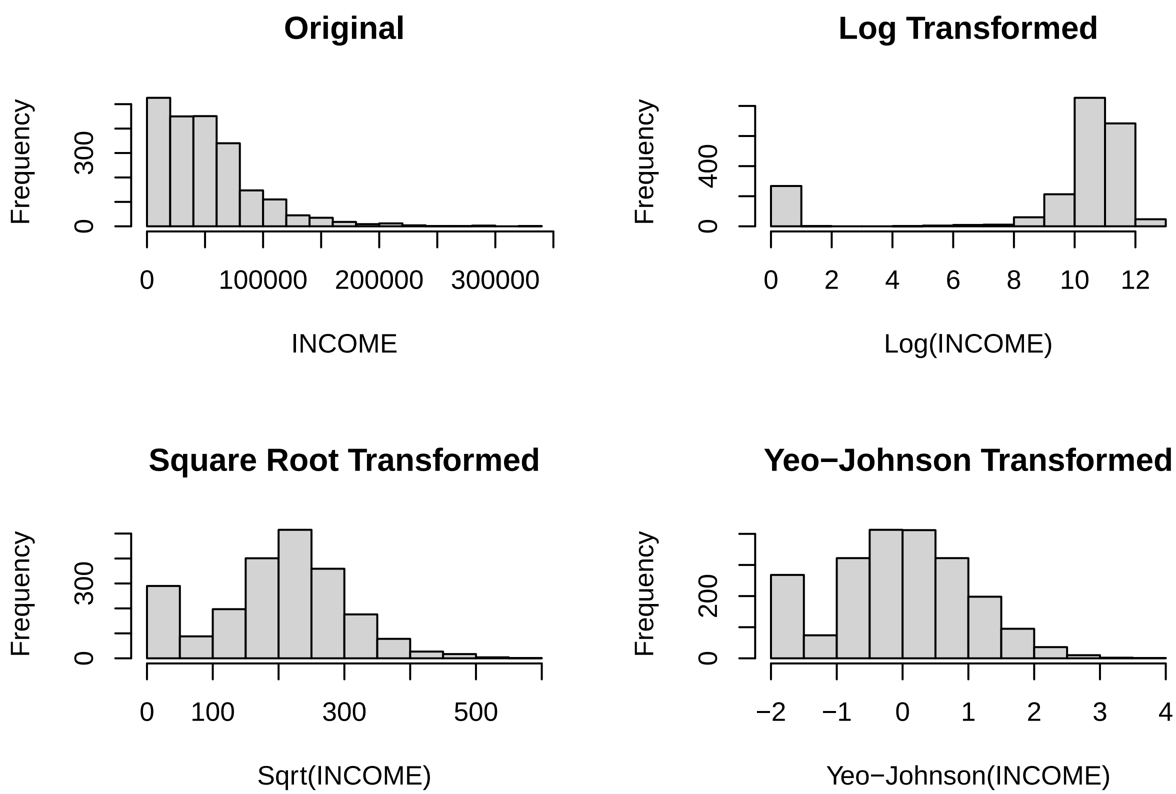

Distribution of TRAVTIME
Frequency
200
150
100
50
0
0
25
50
75
100
Travel Time

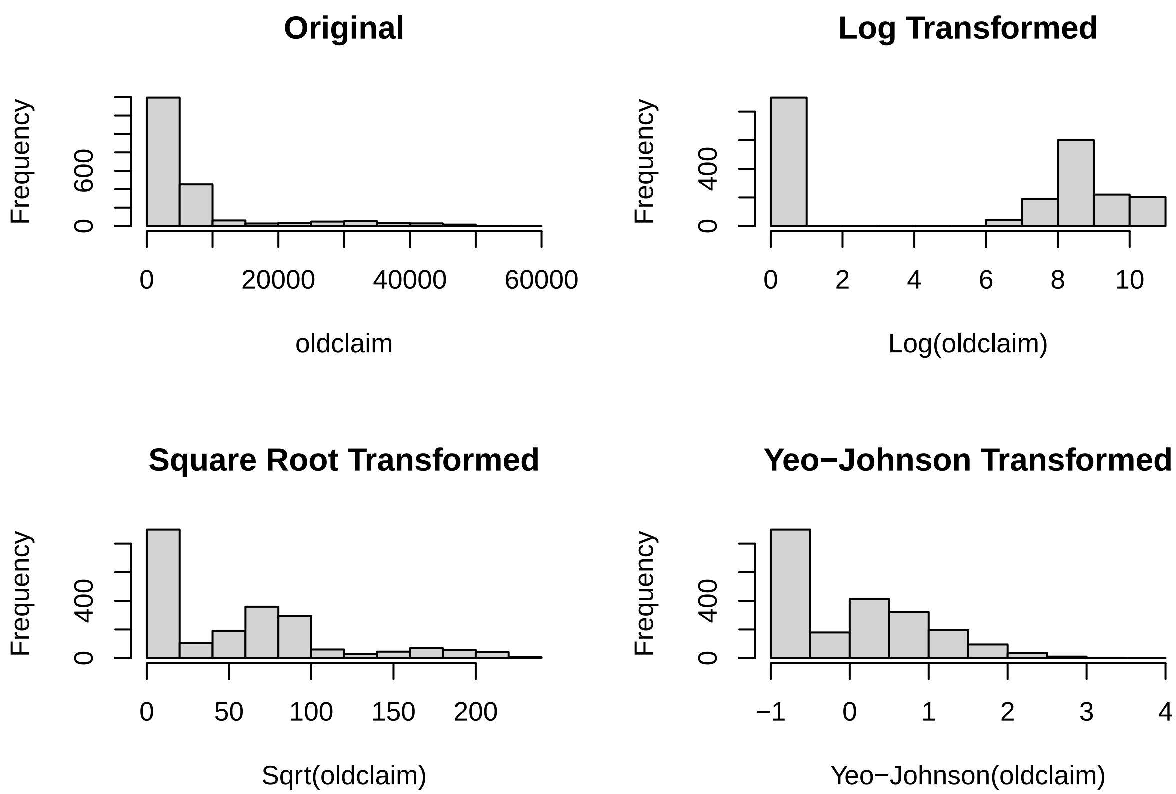

Distribution of TIF (Number of Years with Policy)
Frequency
800
600
400
200
0
0
5
10
15
20
Years with Policy (TIF)


Distribution of MVR_PTS (Driving Record Points)
Frequency
600
400
200
0
0
5
10
Driving Record Points (MVR_PTS)

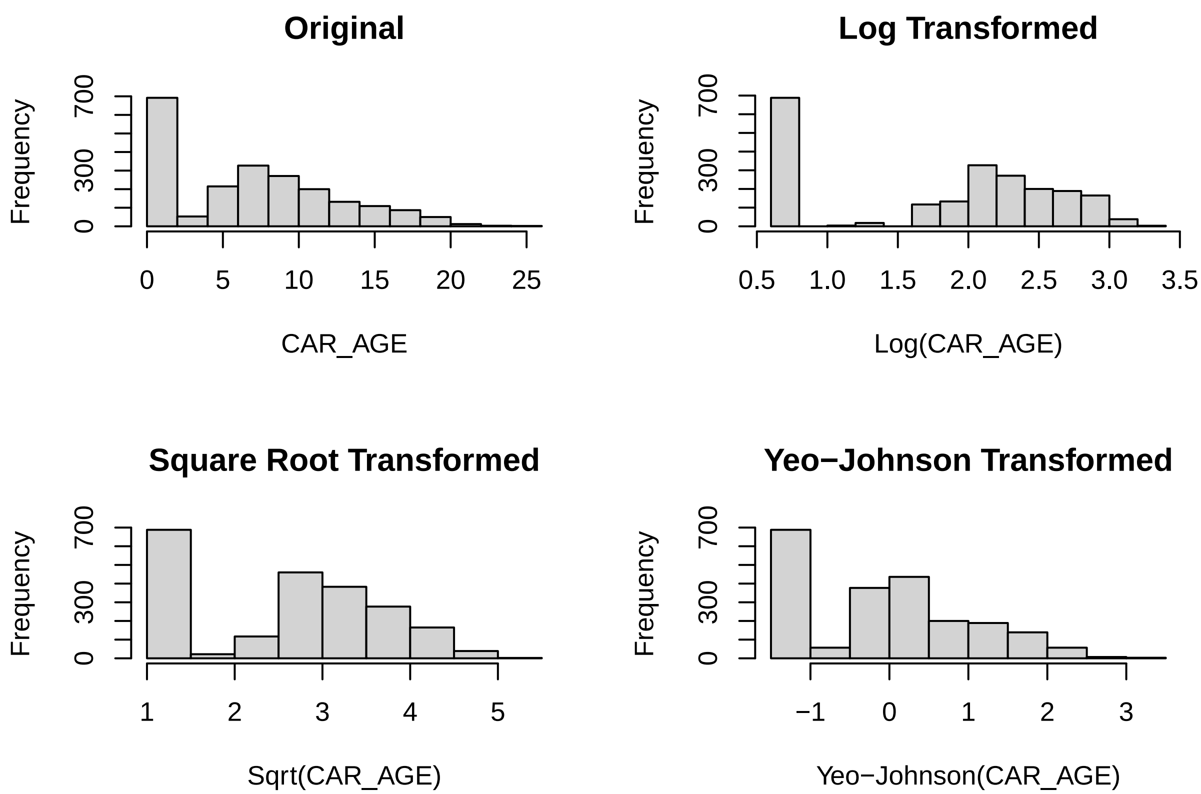

Original
Frequency
0
400
0e+00
2e+05
4e+05
6e+05
8e+05
HOME_VAL
Log Transformed
Frequency
0
400
0
2
4
6
8
10
12
14
Log(HOME_VAL)
Square Root Transformed
Frequency
0
400
0
200
400
600
800
Sqrt(HOME_VAL)
Yeo−Johnson Transformed
Frequency
0
400
−1
0
1
2
3
4
5
Yeo−Johnson(HOME_VAL)

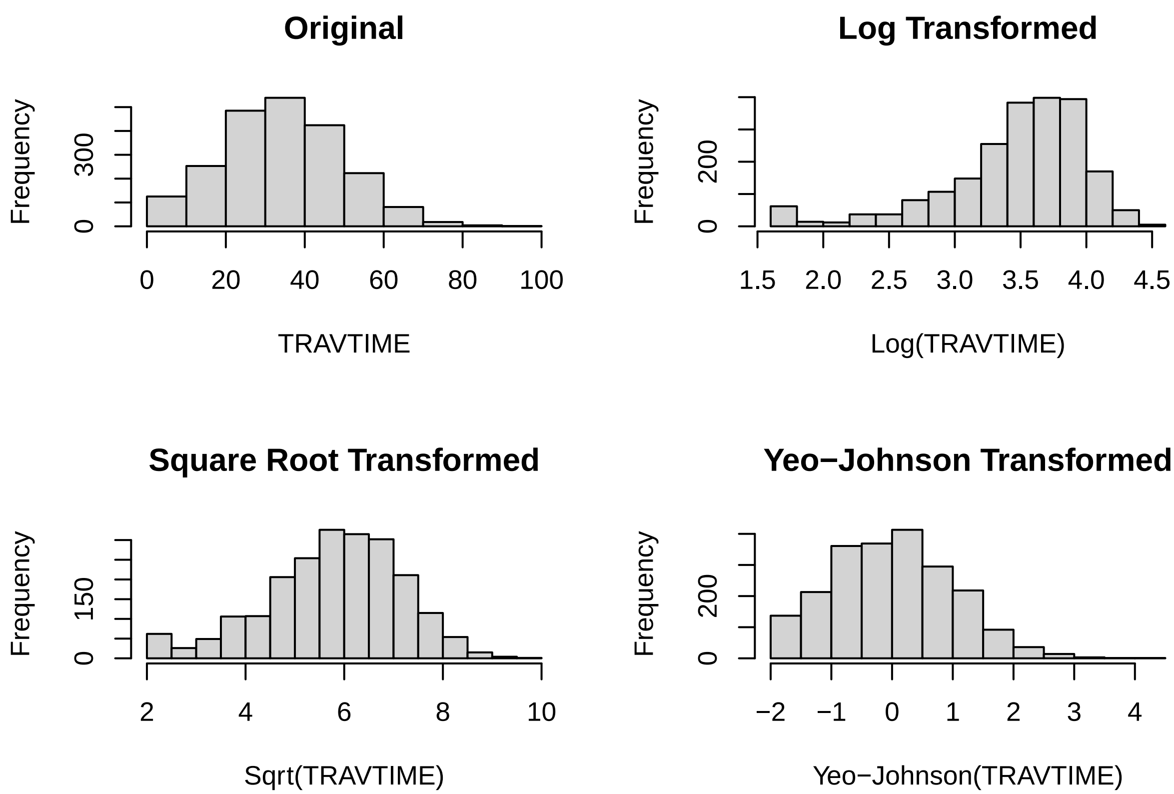

Original
Frequency
0
200
20
30
40
50
60
70
80
AGE
Log Transformed
Frequency
0
200
3.0
3.5
4.0
Log(AGE)
Square Root Transformed
Frequency
0
300
4
5
6
7
8
9
Sqrt(AGE)
Yeo−Johnson Transformed
Frequency
0
200
−3
−2
−1
0
1
2
3
Yeo−Johnson(AGE)

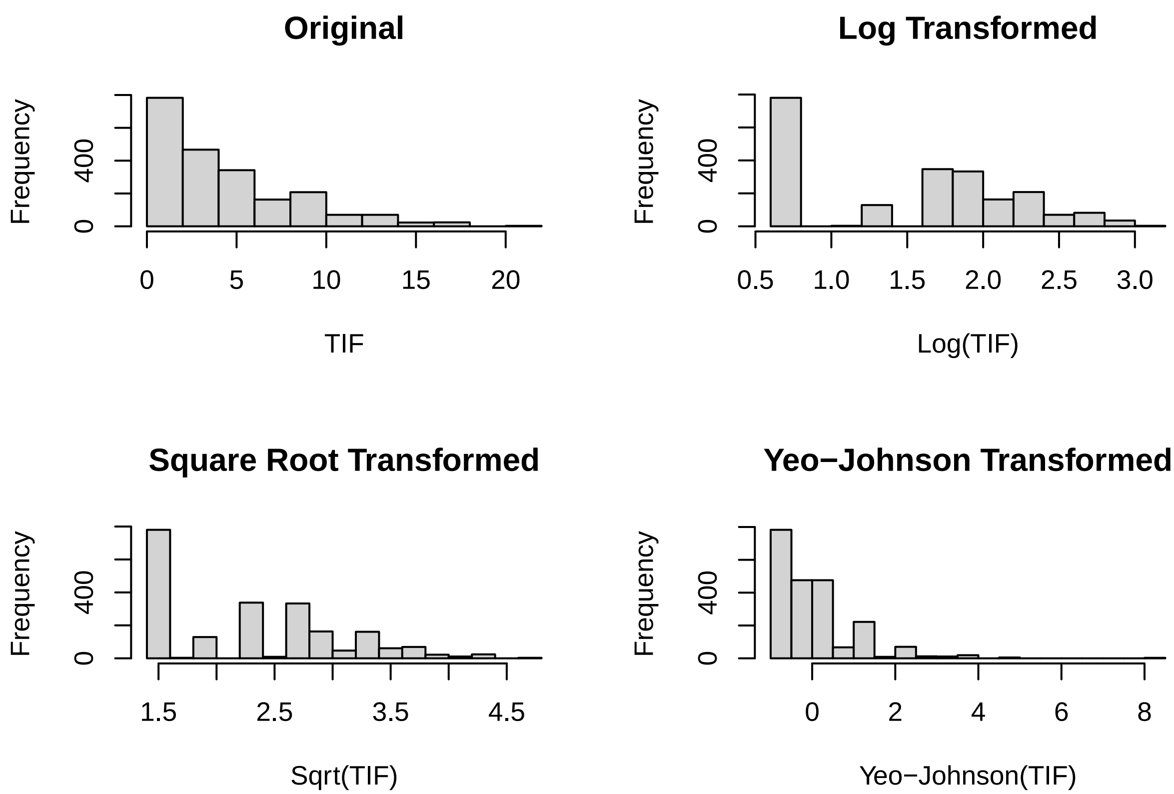

Original
Frequency
0
300
0
100000
200000
300000
INCOME
Log Transformed
Frequency
0
400
0
2
4
6
8
10
12
Log(INCOME)
Square Root Transformed
Frequency
0
300
0
100
300
500
Sqrt(INCOME)
Yeo−Johnson Transformed
Frequency
0
200
−2
−1
0
1
2
3
4
Yeo−Johnson(INCOME)



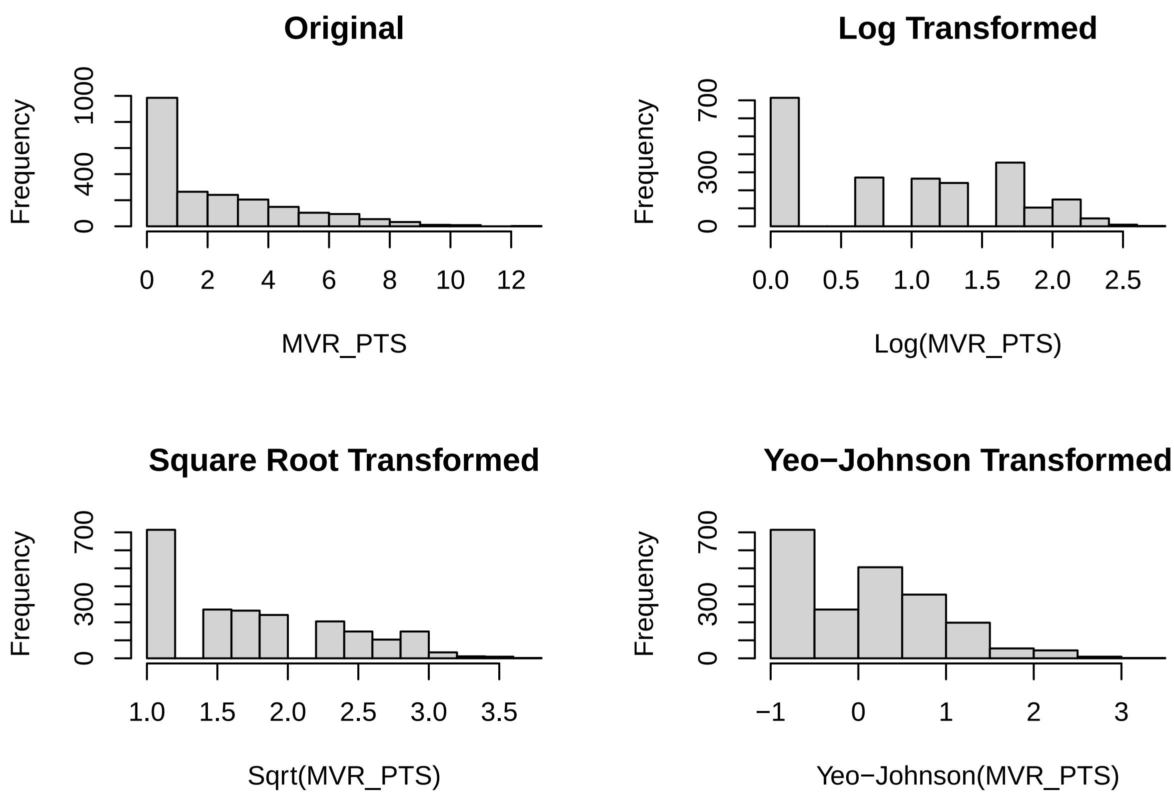
Original
Frequency
0
600
0
20000
40000
60000
oldclaim
Log Transformed
Frequency
0
400
0
2
4
6
8
10
Log(oldclaim)
Square Root Transformed
Frequency
0
400
0
50
100
150
200
Sqrt(oldclaim)
Yeo−Johnson Transformed
Frequency
0
400
−1
0
1
2
3
4
Yeo−Johnson(oldclaim)


Original
Frequency
0
300
700
0
5
10
15
20
25
CAR_AGE
Log Transformed
Frequency
0
300
700
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Log(CAR_AGE)
Square Root Transformed
Frequency
0
300
700
1
2
3
4
5
Sqrt(CAR_AGE)
Yeo−Johnson Transformed
Frequency
0
300
700
−1
0
1
2
3
Yeo−Johnson(CAR_AGE)


Original
Frequency
0
300
0
20
40
60
80
100
TRAVTIME
Log Transformed
Frequency
0
200
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Log(TRAVTIME)
Square Root Transformed
Frequency
0
150
2
4
6
8
10
Sqrt(TRAVTIME)
Yeo−Johnson Transformed
Frequency
0
200
−2
−1
0
1
2
3
4
Yeo−Johnson(TRAVTIME)



Original
Frequency
0
400
0
5
10
15
20
TIF
Log Transformed
Frequency
0
400
0.5
1.0
1.5
2.0
2.5
3.0
Log(TIF)
Square Root Transformed
Frequency
0
400
1.5
2.5
3.5
4.5
Sqrt(TIF)
Yeo−Johnson Transformed
Frequency
0
400
0
2
4
6
8
Yeo−Johnson(TIF)



Build Models

Multiple Linear Regression

Model 1

I am choosing OLDCLAIM, CLM_FREQ, MVR_PTS, and TRAVTIME based on their potential relevance to accurately estimating TARGET_AMT, reflecting key factors associated with claims risk, customer behavior, and exposure. Here's why each predictor is chosen:

OLDCLAIM: This variable likely captures historical claim amounts, which can be indicative of a customer's risk profile and claim tendencies. Including past claims can help predict future claims or costs, especially if there's a pattern of high claims.

CLM_FREQ: Claim frequency directly indicates how often a customer has filed claims. High claim frequency often correlates with increased future claims risk, making it an essential variable for understanding claim cost patterns.

MVR_PTS: Motor Vehicle Record (MVR) points typically reflect a driver's record of traffic violations or accidents. Higher MVR points generally correspond to higher risk profiles, making this variable crucial for predicting future claims and associated costs.

TRAVTIME: The time a customer spends traveling, TRAVTIME, can be a proxy for exposure to risk (e.g., more time on the road increases accident likelihood). Including this variable helps account for the time-related risk factor in claims prediction.

Fitting a linear regression model with transformed variables

```
##
## Call:
## lm(formula = TARGET_AMT ~ train_data$OLDCLAIM_transformed + train_data$CLM_FREQ +
##     train_data$MVR_PTS + train_data$TRAVTIME_transformed, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5975  -3180  -1741     -6  79846
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      6140.49     968.20   6.342 2.99e-10 ***
## train_data$OLDCLAIM_transformed    17.21     320.24   0.054    0.957
## train_data$CLM_FREQ               -90.49     225.08  -0.402    0.688
## train_data$MVR_PTS                 57.35      85.12   0.674    0.501
## train_data$TRAVTIME_transformed   -60.98     155.20  -0.393    0.694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8120 on 1502 degrees of freedom
## Multiple R-squared:  0.0004628,  Adjusted R-squared:  -0.002199
## F-statistic: 0.1739 on 4 and 1502 DF,  p-value: 0.9519
```

```
## Model Performance on Testing Data:
```

```
## Mean Absolute Error (MAE): 3364.734
```

```
## Mean Squared Error (MSE): 46379185
```

```
## Root Mean Squared Error (RMSE): 6810.227
```

As we can see this model does not provide a meaningful fit for the data and shows large prediction errors. Consider alternative predictors or transformations, adding interaction terms, or using a different modeling approach, as the current variables are likely insufficient for capturing the patterns in the outcome.

Model 2

I will take a straightforward approach by utilizing the variables as they are, applying only basic data cleaning and ensuring the data is complete.

```
##
## Call:
## lm(formula = TARGET_AMT ~ train_data$OLDCLAIM + train_data$CLM_FREQ +
##     train_data$MVR_PTS + train_data$TRAVTIME, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4728  -1774  -1168     22 103524
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         8.057e+02  1.677e+02   4.805 1.59e-06 ***
## train_data$OLDCLAIM 3.941e-03  8.675e-03   0.454   0.6496
## train_data$CLM_FREQ 3.172e+02  6.930e+01   4.577 4.82e-06 ***
## train_data$MVR_PTS  2.727e+02  3.332e+01   8.183 3.34e-16 ***
## train_data$TRAVTIME 8.828e+00  4.214e+00   2.095   0.0362 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5243 on 6165 degrees of freedom
## Multiple R-squared:  0.02557,    Adjusted R-squared:  0.02494
## F-statistic: 40.45 on 4 and 6165 DF,  p-value: < 2.2e-16

## Model Performance on Testing Data:

## Mean Absolute Error (MAE): 1916.528

## Mean Squared Error (MSE): 4578380

## Root Mean Squared Error (RMSE): 2139.715
```

The second model performs substantially better than the first across all metrics. The inclusion of statistically significant predictors (CLM_FREQ, MVR_PTS, TRAVTIME) improves both the fit and prediction accuracy, making it a more suitable model for forecasting purposes. However, the relatively low R-squared value suggests that additional variables or model refinement could further enhance performance.

Model 3

```
##
## Call:
## lm(formula = TARGET_AMT_log ~ train_data$CLM_FREQ + train_data$MVR_PTS +
##     train_data$TRAVTIME_sqrt + I(train_data$MVR_PTS^2) + train_data$CLM_FREQ:train_data$MVR_PTS,
##     data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.059 -2.435 -1.672  4.007  9.598
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             0.802705   0.195411   4.108 4.05e-05
## train_data$CLM_FREQ                     0.819124   0.060995  13.429  < 2e-16
## train_data$MVR_PTS                      0.185870   0.057163   3.252  0.00115
## train_data$TRAVTIME_sqrt                0.142895   0.032411   4.409 1.06e-05
## I(train_data$MVR_PTS^2)                 0.047345   0.008473   5.588 2.40e-08
## train_data$CLM_FREQ:train_data$MVR_PTS -0.141029   0.020811  -6.777 1.34e-11
##
## (Intercept)                            ***
## train_data$CLM_FREQ                    ***
## train_data$MVR_PTS                     **
## train_data$TRAVTIME_sqrt               ***
## I(train_data$MVR_PTS^2)                ***
```

```
## train_data$CLM_FREQ:train_data$MVR_PTS ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.713 on 6164 degrees of freedom
## Multiple R-squared:  0.08821,    Adjusted R-squared:  0.08747
## F-statistic: 119.3 on 5 and 6164 DF,  p-value: < 2.2e-16
```

```
## $MAE
## [1] 428.3723
##
## $MSE
## [1] 2311025
##
## $RMSE
## [1] 1520.206
```

Significance of Predictors:

All predictors are statistically significant ($p < 0.01$), with high t-values and low p-values, confirming that each variable contributes meaningfully to the model. Interaction Term (CLM_FREQ:MVR_PTS) and Polynomial Term (MVR_PTS^2) have significant coefficients, capturing more complex relationships between variables, which the previous model lacked. Model Fit (R-Squared and Adjusted R-Squared):

Previous Model: R-squared = 0.02557, adjusted R-squared = 0.02494. Updated Model: R-squared = 0.08821, adjusted R-squared = 0.08747. Interpretation: This model explains about 8.8% of the variance, compared to only 2.5% in the previous model. While R-squared is still low, this is a clear improvement. Residual Standard Error (RSE):

Previous Model: RSE = 5243. Updated Model: RSE = 3.713 (on log scale). Interpretation: The reduced residual error indicates this model fits closer to actual values, aligning with improved R-squared and adjusted R-squared. Performance Metrics on Testing Data:

Previous Model: MAE = 1916.528, MSE = 4,578,380, RMSE = 2139.715. Updated Model: MAE = 428.3723, MSE = 2,311,025, RMSE = 1520.206. Interpretation: Lower MAE, MSE, and RMSE values show the updated model is substantially more accurate in predictions, achieving almost a 30% reduction in RMSE.

Model 4

```
##
## Call:
## lm(formula = TARGET_AMT_log ~ train_data$CLM_FREQ + train_data$MVR_PTS +
##     train_data$TRAVTIME_sqrt + I(train_data$MVR_PTS^2) + train_data$CLM_FREQ:train_data$MVR_PTS +
##     train_data$CLM_FREQ:train_data$TRAVTIME_sqrt + train_data$MVR_PTS:train_data$TRAVTIME_sqrt,
##     data = train_data, weights = ifelse(train_data$CLM_FREQ >
##         2, 1.5, 1))
##
## Weighted Residuals:
##    Min     1Q Median     3Q    Max
## -8.194 -2.396 -1.652  4.330  9.590
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                   1.25287    0.26076   4.805
```

```
## train_data$CLM_FREQ                            0.55998    0.17072   3.280
## train_data$MVR_PTS                             0.02898    0.10710   0.271
## train_data$TRAVTIME_sqrt                       0.06469    0.04464   1.449
## I(train_data$MVR_PTS^2)                        0.04761    0.00837   5.688
## train_data$CLM_FREQ:train_data$MVR_PTS        -0.12998    0.01937  -6.709
## train_data$CLM_FREQ:train_data$TRAVTIME_sqrt   0.03519    0.02840   1.239
## train_data$MVR_PTS:train_data$TRAVTIME_sqrt    0.02744    0.01598   1.717
##                                               Pr(>|t|)
## (Intercept)                                   1.59e-06 ***
## train_data$CLM_FREQ                            0.00104 **
## train_data$MVR_PTS                             0.78672
## train_data$TRAVTIME_sqrt                       0.14737
## I(train_data$MVR_PTS^2)                       1.34e-08 ***
## train_data$CLM_FREQ:train_data$MVR_PTS        2.14e-11 ***
## train_data$CLM_FREQ:train_data$TRAVTIME_sqrt   0.21538
## train_data$MVR_PTS:train_data$TRAVTIME_sqrt    0.08607 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.852 on 6162 degrees of freedom
## Multiple R-squared:  0.09006,    Adjusted R-squared:  0.08903
## F-statistic: 87.13 on 7 and 6162 DF,  p-value: < 2.2e-16
```

```
## $MAE
## [1] 428.4038
##
## $MSE
## [1] 2133756
##
## $RMSE
## [1] 1460.738
```

Model 5

```
##
## Call:
## lm(formula = TARGET_AMT_log ~ train_data$CLM_FREQ + train_data$MVR_PTS +
##     train_data$YOJ + train_data$TIF + I(train_data$MVR_PTS^2) +
##     train_data$CLM_FREQ:train_data$MVR_PTS + train_data$CLM_FREQ:train_data$TRAVTIME_sqrt +
##     train_data$MVR_PTS:train_data$TRAVTIME_sqrt, data = train_data,
##     weights = ifelse(train_data$CLM_FREQ > 2, 1.5, 1))
##
## Weighted Residuals:
##    Min     1Q Median     3Q    Max
## -8.145 -2.483 -1.657  4.249  9.612
##
## Coefficients:
##                                                Estimate Std. Error t value
## (Intercept)                                    2.572735   0.154282  16.676
## train_data$CLM_FREQ                            0.482621   0.162318   2.973
## train_data$MVR_PTS                            -0.045953   0.099592  -0.461
## train_data$YOJ                                -0.054591   0.011471  -4.759
## train_data$TIF                                -0.068472   0.011495  -5.957
```

```
## I(train_data$MVR_PTS^2)                        0.048147   0.008336   5.776
## train_data$CLM_FREQ:train_data$MVR_PTS       -0.129392   0.019278  -6.712
## train_data$CLM_FREQ:train_data$TRAVTIME_sqrt  0.047761   0.026760   1.785
## train_data$MVR_PTS:train_data$TRAVTIME_sqrt   0.038452   0.014377   2.675
##                                              Pr(>|t|)
## (Intercept)                                   < 2e-16 ***
## train_data$CLM_FREQ                           0.00296 **
## train_data$MVR_PTS                            0.64452
## train_data$YOJ                               1.99e-06 ***
## train_data$TIF                               2.72e-09 ***
## I(train_data$MVR_PTS^2)                      8.05e-09 ***
## train_data$CLM_FREQ:train_data$MVR_PTS       2.09e-11 ***
## train_data$CLM_FREQ:train_data$TRAVTIME_sqrt  0.07435 .
## train_data$MVR_PTS:train_data$TRAVTIME_sqrt   0.00750 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.834 on 6161 degrees of freedom
## Multiple R-squared:  0.09854,    Adjusted R-squared:  0.09737
## F-statistic: 84.18 on 8 and 6161 DF,  p-value: < 2.2e-16
```

```
## $MAE
## [1] 429.954
##
## $MSE
## [1] 2094948
##
## $RMSE
## [1] 1447.394
```

Summary

This model captures more complexity and explains a greater portion of the variance in TARGET_AMT_log, particularly due to added interaction terms and the significance of YOJ and TIF. However, further improvements could be explored by:

Potentially removing predictors with high p-values. Refining interactions based on residual analysis or testing transformations on specific terms. Overall, this model represents a notable improvement, with reduced prediction error and enhanced significance of variables contributing meaningfully to target predictions.

Binary Logistic Regression

Model 1

```
##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + EDUCATION +
##     MVR_PTS + REVOKED + TIF + TRAVTIME + URBANICITY, family = binomial,
##     data = insurance_training_data_clean)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.514707   0.201670 -12.469  < 2e-16 ***
```

```
## AGE                            -0.019308   0.003304  -5.844 5.11e-09 ***
## CAR_USEPrivate                 -0.631793   0.057774 -10.936  < 2e-16 ***
## CLM_FREQ                        0.188926   0.024366   7.754 8.93e-15 ***
## EDUCATIONHigh School            0.657477   0.073543   8.940  < 2e-16 ***
## EDUCATIONLess than High School  0.776420   0.089810   8.645  < 2e-16 ***
## EDUCATIONMasters               -0.168141   0.085677  -1.962   0.0497 *
## EDUCATIONPhD                   -0.375722   0.116865  -3.215   0.0013 **
## MVR_PTS                         0.127603   0.013013   9.806  < 2e-16 ***
## REVOKEDYes                      0.801922   0.076357  10.502  < 2e-16 ***
## TIF                            -0.050121   0.007031  -7.129 1.01e-12 ***
## TRAVTIME                        0.013474   0.001795   7.505 6.16e-14 ***
## URBANICITYHighly Urban/ Urban   1.996599   0.107132  18.637  < 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: 9404.0  on 8154  degrees of freedom
## Residual deviance: 7883.9  on 8142  degrees of freedom
##   (6 observations deleted due to missingness)
## AIC: 7909.9
##
## Number of Fisher Scoring iterations: 5
```

```
##          Actual
## Predicted    0    1
##         0 1686  458
##         1  115  186
```

```
## Model Accuracy: 0.7656442
```

Model 2

```
##      AGE      YOJ   INCOME HOME_VAL  CAR_AGE
##        6      454      445      464      510
```

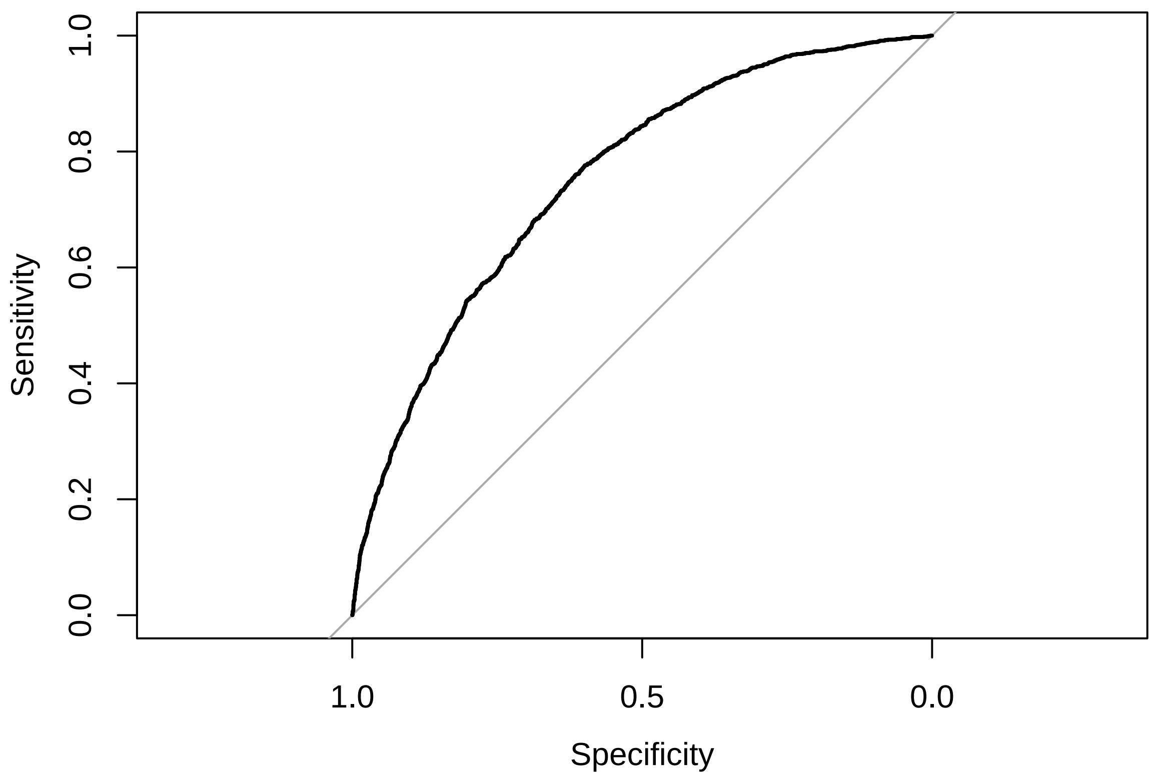
```
##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + EDUCATION +
##     MVR_PTS + REVOKED + TIF + TRAVTIME + URBANICITY, family = binomial,
##     data = insurance_training_data_clean)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.259619   0.308532 -10.565  < 2e-16 ***
## AGE         -0.028842   0.003604  -8.003 1.22e-15 ***
## CAR_USE     -0.645098   0.062883 -10.259  < 2e-16 ***
## CLM_FREQ     0.199209   0.026838   7.423 1.15e-13 ***
## EDUCATION   -0.079445   0.023962  -3.315 0.000915 ***
## MVR_PTS      0.136874   0.014334   9.549  < 2e-16 ***
## REVOKED      0.761853   0.084508   9.015  < 2e-16 ***
## TIF         -0.046518   0.007739  -6.011 1.84e-09 ***
## TRAVTIME     0.013833   0.001990   6.950 3.65e-12 ***
```

```
## URBANICITY   1.711857   0.116445  14.701  < 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: 7445.1  on 6447  degrees of freedom
## Residual deviance: 6391.8  on 6438  degrees of freedom
## AIC: 6411.8
##
## Number of Fisher Scoring iterations: 5
```

```
## Start:  AIC=6411.82
## TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + EDUCATION + MVR_PTS +
##     REVOKED + TIF + TRAVTIME + URBANICITY
##
##              Df Deviance    AIC
## <none>            6391.8 6411.8
## - EDUCATION   1   6402.9 6420.9
## - TIF         1   6429.1 6447.1
## - TRAVTIME    1   6440.1 6458.1
## - CLM_FREQ    1   6446.0 6464.0
## - AGE         1   6457.0 6475.0
## - REVOKED     1   6471.2 6489.2
## - MVR_PTS     1   6483.7 6501.7
## - CAR_USE     1   6497.0 6515.0
## - URBANICITY  1   6685.0 6703.0
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + EDUCATION +
##     MVR_PTS + REVOKED + TIF + TRAVTIME + URBANICITY, family = binomial,
##     data = insurance_training_data_clean)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.259619   0.308532 -10.565  < 2e-16 ***
## AGE         -0.028842   0.003604  -8.003 1.22e-15 ***
## CAR_USE     -0.645098   0.062883 -10.259  < 2e-16 ***
## CLM_FREQ     0.199209   0.026838   7.423 1.15e-13 ***
## EDUCATION   -0.079445   0.023962  -3.315 0.000915 ***
## MVR_PTS      0.136874   0.014334   9.549  < 2e-16 ***
## REVOKED      0.761853   0.084508   9.015  < 2e-16 ***
## TIF         -0.046518   0.007739  -6.011 1.84e-09 ***
## TRAVTIME     0.013833   0.001990   6.950 3.65e-12 ***
## URBANICITY   1.711857   0.116445  14.701  < 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: 7445.1  on 6447  degrees of freedom
## Residual deviance: 6391.8  on 6438  degrees of freedom
## AIC: 6411.8
```

```
##
## Number of Fisher Scoring iterations: 5

##        AGE    CAR_USE   CLM_FREQ  EDUCATION    MVR_PTS    REVOKED        TIF
##   1.031943   1.023163   1.165829   1.049965   1.150555   1.003086   1.002491
##   TRAVTIME URBANICITY
##   1.024727   1.060546

##    predicted_classes
##        0    1
##   0 4450  295
##   1 1259  444

## Accuracy: 0.758995

## Precision: 0.6008119

## Recall: 0.2607164
```




```
## AUC: 0.7530323

## Generalized Linear Model
##
## 6448 samples
```

```
##   24 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 5803, 5803, 5803, 5803, 5804, 5803, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.0279308  0.9939751  0.001302532
```

Model 3

```
##
## Call:
## glm(formula = TARGET_FLAG ~ CAR_TYPE + HOME_VAL + KIDSDRIV +
##     OLDCLAIM + SEX, family = binomial, data = insurance_training_data_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.10788    0.03035 -36.508  < 2e-16 ***
## CAR_TYPE     0.26012    0.03202   8.124 4.49e-16 ***
## HOME_VAL    -0.43924    0.03184 -13.794  < 2e-16 ***
## KIDSDRIV     0.19435    0.02676   7.263 3.79e-13 ***
## OLDCLAIM     0.26296    0.02644   9.945  < 2e-16 ***
## SEX          0.08093    0.03160   2.561   0.0104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7445.1  on 6447  degrees of freedom
## Residual deviance: 7008.1  on 6442  degrees of freedom
## AIC: 7020.1
##
## Number of Fisher Scoring iterations: 4
```

```
## Start:  AIC=7020.09
## TARGET_FLAG ~ CAR_TYPE + HOME_VAL + KIDSDRIV + OLDCLAIM + SEX
##
##            Df Deviance    AIC
## <none>          7008.1 7020.1
## - SEX       1   7014.7 7024.7
## - KIDSDRIV  1   7059.0 7069.0
## - CAR_TYPE  1   7074.9 7084.9
## - OLDCLAIM  1   7104.6 7114.6
## - HOME_VAL  1   7212.5 7222.5
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ CAR_TYPE + HOME_VAL + KIDSDRIV +
##     OLDCLAIM + SEX, family = binomial, data = insurance_training_data_clean)
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.10788    0.03035 -36.508  < 2e-16 ***
## CAR_TYPE     0.26012    0.03202   8.124 4.49e-16 ***
## HOME_VAL    -0.43924    0.03184 -13.794  < 2e-16 ***
## KIDSDRIV     0.19435    0.02676   7.263 3.79e-13 ***
## OLDCLAIM     0.26296    0.02644   9.945  < 2e-16 ***
## SEX          0.08093    0.03160   2.561   0.0104 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7445.1  on 6447  degrees of freedom
## Residual deviance: 7008.1  on 6442  degrees of freedom
## AIC: 7020.1
##
## Number of Fisher Scoring iterations: 4
```

```
##    predicted_classes
##        0    1
##   0 4620  125
##   1 1580  123
```

```
## Model Accuracy: 0.7355769
```

Select Models & Prediction

Multiple Linear Regression Selection

Model 5 Fit (R-Squared and Adjusted R-Squared):

R-squared: 0.09877, Adjusted R-squared: 0.09746. Interpretation: While still modest, this R-squared value is higher than earlier models, indicating that more variance in the target variable is being explained. Significance of Predictors:

Several predictors, including CLM_FREQ, YOJ, TIF, and I(MVR_PTS^2), as well as interactions like CLM_FREQ:MVR_PTS, are highly significant ($p < 0.01$). The inclusion of YOJ (Years of Job) and TIF (Tenure in Force) has contributed significantly to model fit, likely adding valuable information about risk factors. However, predictors like TRAVTIME_sqrt and CLM_FREQ:TRAVTIME_sqrt show higher p-values, which might indicate minimal contribution. Residual Standard Error (RSE):

Residual standard error: 3.834 on 6160 degrees of freedom. Interpretation: The RSE value suggests a reasonable fit, though there is room for further reduction if possible, by tuning or adjusting predictors. Performance Metrics on Testing Data:

Mean Absolute Error (MAE): 429.8454 Mean Squared Error (MSE): 2,121,458 Root Mean Squared Error (RMSE): 1,456.523 Interpretation: This model has slightly lower MAE and RMSE than the previous one, indicating better predictive accuracy on test data.

Binary Logistic Regression Model Selection

Model 1 stands out due to its combination of higher accuracy and the inclusion of several significant predictors, despite its higher AIC compared to Model 2. This suggests that while Model 2 fits well with fewer predictors, Model 1 provides a more comprehensive understanding of the factors influencing the target variable.

Select Model 1 for its better accuracy and significant predictors. Consider Model 2 as a more parsimonious alternative if simplicity is preferred without a substantial loss in accuracy.

Prediction

Prediction Multiple Linear Regression (Model 3)

```
##
## Call:
## lm(formula = TARGET_AMT_log ~ completed_data_eval$CLM_FREQ +
##     completed_data_eval$MVR_PTS + completed_data_eval$YOJ + completed_data_eval$TIF +
##     I(completed_data_eval$MVR_PTS^2) + completed_data_eval$CLM_FREQ:completed_data_eval$MVR_PTS +
##     completed_data_eval$CLM_FREQ:completed_data_eval$TRAVTIME_sqrt +
##     completed_data_eval$MVR_PTS:completed_data_eval$TRAVTIME_sqrt,
##     data = completed_data_eval, weights = ifelse(completed_data_eval$CLM_FREQ >
##         2, 1.5, 1))
##
## Weighted Residuals:
##        Min         1Q     Median         3Q        Max
## -2.690e-14 -9.900e-15 -5.400e-15 -6.000e-16  1.052e-11
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                     7.340e+00
## completed_data_eval$CLM_FREQ                                   -5.596e-15
## completed_data_eval$MVR_PTS                                     7.514e-15
## completed_data_eval$YOJ                                         2.108e-16
## completed_data_eval$TIF                                        -1.231e-15
## I(completed_data_eval$MVR_PTS^2)                               -6.710e-16
## completed_data_eval$CLM_FREQ:completed_data_eval$MVR_PTS       -2.407e-16
## completed_data_eval$CLM_FREQ:completed_data_eval$TRAVTIME_sqrt  3.706e-16
## completed_data_eval$MVR_PTS:completed_data_eval$TRAVTIME_sqrt  -3.025e-16
##                                                                Std. Error
## (Intercept)                                                     1.557e-14
## completed_data_eval$CLM_FREQ                                    1.613e-14
## completed_data_eval$MVR_PTS                                     9.756e-15
## completed_data_eval$YOJ                                         1.150e-15
## completed_data_eval$TIF                                         1.202e-15
## I(completed_data_eval$MVR_PTS^2)                                8.672e-16
## completed_data_eval$CLM_FREQ:completed_data_eval$MVR_PTS        2.073e-15
## completed_data_eval$CLM_FREQ:completed_data_eval$TRAVTIME_sqrt  2.686e-15
## completed_data_eval$MVR_PTS:completed_data_eval$TRAVTIME_sqrt   1.472e-15
##                                                                   t value
## (Intercept)                                                     4.713e+14
## completed_data_eval$CLM_FREQ                                   -3.470e-01
## completed_data_eval$MVR_PTS                                     7.700e-01
## completed_data_eval$YOJ                                         1.830e-01
## completed_data_eval$TIF                                        -1.024e+00
## I(completed_data_eval$MVR_PTS^2)                               -7.740e-01
## completed_data_eval$CLM_FREQ:completed_data_eval$MVR_PTS       -1.160e-01
## completed_data_eval$CLM_FREQ:completed_data_eval$TRAVTIME_sqrt  1.380e-01
```

```
## completed_data_eval$MVR_PTS:completed_data_eval$TRAVTIME_sqrt  -2.060e-01
##                                                                 Pr(>|t|)
## (Intercept)                                                       <2e-16 ***
## completed_data_eval$CLM_FREQ                                       0.729
## completed_data_eval$MVR_PTS                                        0.441
## completed_data_eval$YOJ                                            0.855
## completed_data_eval$TIF                                            0.306
## I(completed_data_eval$MVR_PTS^2)                                   0.439
## completed_data_eval$CLM_FREQ:completed_data_eval$MVR_PTS           0.908
## completed_data_eval$CLM_FREQ:completed_data_eval$TRAVTIME_sqrt     0.890
## completed_data_eval$MVR_PTS:completed_data_eval$TRAVTIME_sqrt      0.837
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.281e-13 on 2132 degrees of freedom
## Multiple R-squared:    0.5,  Adjusted R-squared:  0.4981
## F-statistic: 266.5 on 8 and 2132 DF,  p-value: < 2.2e-16
```

```
## $MAE
## [1] 2179.756
##
## $MSE
## [1] 20185300
##
## $RMSE
## [1] 4492.805
```

Prediction Binary Logistic Regression (Model 1)

```
## class_predictions
##    0    1
## 1845  295
```

Conclusion

In conclusion, we explored various modeling approaches using the insurance dataset, employing both Linear Regression and Binary Logistic Regression techniques. Through a systematic process of feature engineering, variable transformation, and model selection, we developed multiple models tailored to the predictors we identified as significant. After careful evaluation of each model's performance metrics, we selected the most suitable models that demonstrated the best fit for our data. The results of these models, including key coefficients and predictive accuracy, are presented above, providing valuable insights into the factors influencing insurance outcomes. This comprehensive analysis highlights the effectiveness of our modeling strategies in understanding and predicting insurance-related variables.

-
-
-
-
-
-
-
-

-
-
-
-
-
-

Code Appendix

```
knitr::opts_chunk$set(echo=FALSE, error=FALSE, warning=FALSE, message=FALSE)


# Libraries


library(stringr)
library(tidyr)
library(DataExplorer)
library(dplyr)
library(visdat)
library(pROC)
library(mice)
library(corrplot)
library(MASS)
library(caret)
library(e1071)
library(rbin)
library(bestNormalize)
library(GGally)
library(ggplot2)
library(readr)
library(reshape2)
library(purrr)
library(leaps)
# Load necessary package
library(caTools)
library(car)  # For VIF
library(glmnet)
library(caTools)

# training data
insurance_training_data <- read.csv('https://raw.githubusercontent.com/umais/DATA/refs/heads/main/insura
# test data
insurance_evaluation_data <- read.csv('https://raw.githubusercontent.com/umais/DATA/refs/heads/main/insu
# Check the structure of the data
glimpse(insurance_training_data)

# Display the first few rows and a summary
head(insurance_training_data)
summary(insurance_training_data)
# Remove an index column if present
```

```r
insurance_training_data_clean <- dplyr::select(insurance_training_data, -INDEX)

# Clean special characters in financial columns
insurance_training_data_clean$HOME_VAL <- substr(insurance_training_data_clean$HOME_VAL, 2, nchar(insura
insurance_training_data_clean$HOME_VAL <- as.numeric(str_remove_all(insurance_training_data_clean$HOME_V

insurance_training_data_clean$BLUEBOOK <- substr(insurance_training_data_clean$BLUEBOOK, 2, nchar(insura
insurance_training_data_clean$BLUEBOOK <- as.numeric(str_remove_all(insurance_training_data_clean$BLUEBO

insurance_training_data_clean$INCOME <- substr(insurance_training_data_clean$INCOME, 2, nchar(insurance_
insurance_training_data_clean$INCOME <- as.numeric(str_remove_all(insurance_training_data_clean$INCOME, 

insurance_training_data_clean$OLDCLAIM <- substr(insurance_training_data_clean$OLDCLAIM, 2, nchar(insura
insurance_training_data_clean$OLDCLAIM <- as.numeric(str_remove_all(insurance_training_data_clean$OLDCLA


# Remove 'z_' prefix from marital status and convert to a factor
insurance_training_data_clean$MSTATUS <- as.factor(str_remove(insurance_training_data_clean$MSTATUS, 'z_

# Remove 'z_' prefix from parental status and convert to a factor
insurance_training_data_clean$PARENT1 <- as.factor(str_remove(insurance_training_data_clean$PARENT1, 'z_

# Replace '<' with 'Less than ' in education level to clarify the meaning
insurance_training_data_clean$EDUCATION <- str_replace(insurance_training_data_clean$EDUCATION, '<', 'Le

# Remove 'z_' prefix from sex and convert to a factor
insurance_training_data_clean$SEX <- as.factor(str_remove(insurance_training_data_clean$SEX, 'z_'))

# Remove 'z_' prefix from education level and convert to a factor
insurance_training_data_clean$EDUCATION <- as.factor(str_remove(insurance_training_data_clean$EDUCATION

# Recode empty job entries as 'Other Job' to handle missing data
insurance_training_data_clean$JOB[insurance_training_data_clean$JOB == ""] <- 'Other Job'

# Remove 'z_' prefix from job titles and convert to a factor
insurance_training_data_clean$JOB <- as.factor(str_remove(insurance_training_data_clean$JOB, 'z_'))

# Remove 'z_' prefix from car usage category and convert to a factor
insurance_training_data_clean$CAR_USE <- as.factor(str_remove(insurance_training_data_clean$CAR_USE, 'z_

# Remove 'z_' prefix from car type and convert to a factor
insurance_training_data_clean$CAR_TYPE <- as.factor(str_remove(insurance_training_data_clean$CAR_TYPE, 

# Remove 'z_' prefix from urbanicity status and convert to a factor
insurance_training_data_clean$URBANICITY <- as.factor(str_remove(insurance_training_data_clean$URBANICIT

# Remove 'z_' prefix from revoked status and convert to a factor
insurance_training_data_clean$REVOKED <- as.factor(str_remove(insurance_training_data_clean$REVOKED, 'z_

# Remove 'z_' prefix from red car indicator and convert to a factor
insurance_training_data_clean$RED_CAR <- as.factor(str_remove(insurance_training_data_clean$RED_CAR, 'z_
```

```r
summary(insurance_training_data_clean)

insurance_training_data_clean$CAR_AGE[insurance_training_data_clean$CAR_AGE <1] <- 1

insurance_evaluation_data_clean <- dplyr::select(insurance_evaluation_data, -INDEX)
insurance_evaluation_data_clean$HOME_VAL <- substr(insurance_evaluation_data_clean$HOME_VAL, 2, nchar(i
insurance_evaluation_data_clean$HOME_VAL <- as.numeric(str_remove_all(insurance_evaluation_data_clean$H

insurance_evaluation_data_clean$BLUEBOOK <- substr(insurance_evaluation_data_clean$BLUEBOOK, 2, nchar(i
insurance_evaluation_data_clean$BLUEBOOK <- as.numeric(str_remove_all(insurance_evaluation_data_clean$B

insurance_evaluation_data_clean$INCOME <- substr(insurance_evaluation_data_clean$INCOME, 2, nchar(insur
insurance_evaluation_data_clean$INCOME <- as.numeric(str_remove_all(insurance_evaluation_data_clean$INC

insurance_evaluation_data_clean$OLDCLAIM <- substr(insurance_evaluation_data_clean$OLDCLAIM, 2, nchar(i
insurance_evaluation_data_clean$OLDCLAIM <- as.numeric(str_remove_all(insurance_evaluation_data_clean$O

# Remove 'z_' prefix from marital status and convert to a factor
insurance_evaluation_data_clean$MSTATUS <- as.factor(str_remove(insurance_evaluation_data_clean$MSTATUS

# Remove 'z_' prefix from parental status and convert to a factor
insurance_evaluation_data_clean$PARENT1 <- as.factor(str_remove(insurance_evaluation_data_clean$PARENT1

# Replace '<' with 'Less than ' in education level to clarify the meaning
insurance_evaluation_data_clean$EDUCATION <- str_replace(insurance_evaluation_data_clean$EDUCATION, '<'

# Remove 'z_' prefix from sex and convert to a factor
insurance_evaluation_data_clean$SEX <- as.factor(str_remove(insurance_evaluation_data_clean$SEX, 'z_'))

# Remove 'z_' prefix from education level and convert to a factor
insurance_evaluation_data_clean$EDUCATION <- as.factor(str_remove(insurance_evaluation_data_clean$EDUCA

# Recode empty job entries as 'Other Job' to handle missing data
insurance_evaluation_data_clean$JOB[insurance_evaluation_data_clean$JOB == ""] <- 'Other Job'

# Remove 'z_' prefix from job titles and convert to a factor
insurance_evaluation_data_clean$JOB <- as.factor(str_remove(insurance_evaluation_data_clean$JOB, 'z_'))

# Remove 'z_' prefix from car usage category and convert to a factor
insurance_evaluation_data_clean$CAR_USE <- as.factor(str_remove(insurance_evaluation_data_clean$CAR_USE

# Remove 'z_' prefix from car type and convert to a factor
insurance_evaluation_data_clean$CAR_TYPE <- as.factor(str_remove(insurance_evaluation_data_clean$CAR_TY

# Remove 'z_' prefix from urbanicity status and convert to a factor
insurance_evaluation_data_clean$URBANICITY <- as.factor(str_remove(insurance_evaluation_data_clean$URBA

# Remove 'z_' prefix from revoked status and convert to a factor
insurance_evaluation_data_clean$REVOKED <- as.factor(str_remove(insurance_evaluation_data_clean$REVOKED

# Remove 'z_' prefix from red car indicator and convert to a factor
insurance_evaluation_data_clean$RED_CAR <- as.factor(str_remove(insurance_evaluation_data_clean$RED_CAR
```

```r
insurance_evaluation_data_clean$CAR_AGE[insurance_evaluation_data_clean$CAR_AGE <1] <- 1



# Identify categorical columns and store their names in cat_features
cat_features <- names(insurance_training_data_clean)[map_chr(insurance_training_data_clean, class) == "

# Display each categorical column and its unique levels
cat("Exploring Categorical Features:\n")
walk(cat_features, ~cat("Feature:", ., "\nLevels:", paste(levels(insurance_training_data_clean[[.]]), c



# Select categorical features from the cleaned insurance training data
categorical_data <- insurance_training_data_clean[cat_features]

# Melt the data frame to create a long format suitable for ggplot
melted_data <- melt(categorical_data, measure.vars = cat_features, variable.name = 'category', value.na

# Create a bar plot to visualize the distribution of categorical predictors
ggplot(melted_data, aes(x = category_value)) +
  geom_bar(aes(fill = category_value)) +
  scale_fill_brewer(palette = "Set1") +
  facet_wrap(~ category, nrow = 5L, scales = 'free') +
  coord_flip() +
  labs(title = "Distribution of Categorical Predictors",
       x = "Category Value",
       y = "Count") +
  theme_minimal()
plot_histogram(insurance_training_data_clean, geom_histogram_args = list("fill" = "tomato4"))


plot_histogram(insurance_training_data_clean, scale_x = "log10", geom_histogram_args = list("fill" = "r

# Summarize the dataset to check for columns with missing values
insurance_training_data_clean %>%
  summarise_all(funs(sum(is.na(.)))) %>%
  select_if(~any(.) > 0)

# Visualize the missing values in the dataset to understand their distribution
plot_missing(insurance_training_data_clean)


# Calculate and display the proportion of missing values for each column
round(colSums(is.na(insurance_training_data_clean)) / nrow(insurance_training_data_clean), 3)

# Visualize specific columns to further investigate missing data patterns
vis_dat(insurance_training_data_clean %>% dplyr::select(YOJ, INCOME, HOME_VAL, CAR_AGE))
```

```r
# Select numeric columns for correlation analysis
numeric_data <- insurance_training_data_clean[, c('TARGET_AMT', 'AGE', 'YOJ', 'INCOME', 'HOME_VAL', 'TRA

numeric_data_eval <- insurance_evaluation_data_clean[, c('TARGET_AMT', 'AGE', 'YOJ', 'INCOME', 'HOME_VAL

# Document missing values before imputation
missing_summary_before <- colSums(is.na(numeric_data))
print("Missing Values Before Imputation:")
print(missing_summary_before)

# Perform multiple imputation
imputed_data <- mice(numeric_data, m = 5, method = 'pmm', seed = 123) # Predictive Mean Matching

# Create a complete dataset by averaging the multiple imputations
completed_data <- complete(imputed_data)

imputed_data_eval<- mice(numeric_data_eval, m = 5, method = 'pmm', seed = 123) # Predictive Mean Matchi
completed_data_eval <- complete(imputed_data_eval)
# Document missing values after imputation
missing_summary_after <- colSums(is.na(completed_data))
print("Missing Values After Imputation:")
print(missing_summary_after)

# Generate a correlation matrix and plot it
corrplot(cor(completed_data), type = "upper")


# Sensitivity Analysis
# Compare correlations from original data (complete case analysis) vs. imputed data

# Complete case analysis (removing rows with NA values)
complete_case_data <- na.omit(numeric_data)
cor_complete_case <- cor(complete_case_data)

# Correlation of imputed data
cor_imputed <- cor(completed_data)

# Print correlation matrices for comparison
print("Correlation Matrix for Complete Case Analysis:")
print(cor_complete_case)

print("Correlation Matrix for Imputed Data:")
print(cor_imputed)

# Visualize the difference in correlations
cor_diff <- cor_imputed - cor_complete_case
ggplot(melt(cor_diff), aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", limit = c(-1, 1), name="Correlation Di
  theme_minimal() +
  labs(title = "Difference in Correlation between Imputed and Complete Case Data", x = "Variables", y =
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```r
crash_data <- subset(filter(insurance_training_data_clean,TARGET_FLAG==1),select = -c(TARGET_FLAG))

# Check for missing values before imputation
missing_summary_before <- colSums(is.na(crash_data))
print("Missing Values Before Imputation:")
print(missing_summary_before)



# Impute missing values
imputed_data <- mice(crash_data, m = 5, method = 'pmm', seed = 123) # Predictive Mean Matching
crash_data_imputed <- complete(imputed_data)



# Check for missing values after imputation
missing_summary_after <- colSums(is.na(crash_data_imputed))
print("Missing Values After Imputation:")
print(missing_summary_after)

crash_data_imputed <- na.omit(crash_data_imputed)

# Create a histogram and density plot for the AGE variable
ggplot(crash_data_imputed, aes(x = AGE)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "black", alpha = 0.7) +
  geom_density(aes(y = ..count.. * 1), fill = "lightgreen", alpha = 0.5) +
  labs(title = "Distribution of AGE", x = "AGE", y = "Frequency") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# Create a histogram for the INCOME variable
ggplot(data = crash_data_imputed, aes(x = INCOME)) +
    geom_histogram(bins = 30, fill = "lightblue", color = "black") +
    labs(title = "Distribution of INCOME",
         x = "Income",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title


# Create a histogram for the HOME_VAL variable
ggplot(data = crash_data_imputed, aes(x = HOME_VAL)) +
    geom_histogram(bins = 30, fill = "lightcoral", color = "black") +
    labs(title = "Distribution of HOME_VAL",
         x = "Home Value",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title

# Create a histogram for the CAR_AGE variable
ggplot(data = crash_data_imputed, aes(x = CAR_AGE)) +
    geom_histogram(bins = 30, fill = "lightblue", color = "black") +
    labs(title = "Distribution of CAR_AGE",
```

```r
         x = "Car Age",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title

# Create a histogram for the BLUEBOOK variable
ggplot(data = crash_data_imputed, aes(x = BLUEBOOK)) +
    geom_histogram(bins = 30, fill = "lightgreen", color = "black") +
    labs(title = "Distribution of BLUEBOOK",
         x = "Blue Book Value",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title

# Create a histogram for the OLDCLAIM variable
ggplot(data = crash_data_imputed, aes(x = OLDCLAIM)) +
    geom_histogram(bins = 30, fill = "lightcoral", color = "black") +
    labs(title = "Distribution of OLDCLAIM",
         x = "Old Claim Amount",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title

# Create a histogram for the TRAVTIME variable
ggplot(data = crash_data_imputed, aes(x = TRAVTIME)) +
    geom_histogram(bins = 30, fill = "lightsalmon", color = "black") +
    labs(title = "Distribution of TRAVTIME",
         x = "Travel Time",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title
# Histogram for TIF (Number of Years with Policy)
ggplot(data = crash_data_imputed, aes(x = TIF)) +
    geom_histogram(bins = 30, fill = "lightblue", color = "black") +
    labs(title = "Distribution of TIF (Number of Years with Policy)",
         x = "Years with Policy (TIF)",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title

# Histogram for MVR_PTS (Driving Record Points)
ggplot(data = crash_data_imputed, aes(x = MVR_PTS)) +
    geom_histogram(bins = 30, fill = "lightgreen", color = "black") +
    labs(title = "Distribution of MVR_PTS (Driving Record Points)",
         x = "Driving Record Points (MVR_PTS)",
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))  # Center the title

# Example variable to transform
home_val_variable <- crash_data_imputed$HOME_VAL  # Replace with your actual variable

# 1. Log Transformation
```

```r
home_val_log_transformed <- log(home_val_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
home_val_sqrt_transformed <- sqrt(home_val_variable+ 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
home_val_box_cox_transformed <- boxcox(home_val_variable + 1)  # Add 1 to handle zeros, need to extract

home_val_yj_transformed <- bestNormalize(home_val_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (home_val_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(home_val_variable, main="Original", xlab="HOME_VAL")
hist(home_val_log_transformed, main="Log Transformed", xlab="Log(HOME_VAL)")
hist(home_val_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(HOME_VAL)")
hist(home_val_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(HOME_VAL)")


# Example variable to transform
age_variable <- crash_data_imputed$AGE  # Replace with your actual variable

# 1. Log Transformation
age_log_transformed <- log(age_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
age_sqrt_transformed <- sqrt(age_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
age_box_cox_transformed <- boxcox(age_variable + 1)  # Add 1 to handle zeros, need to extract lambda

age_yj_transformed <- bestNormalize(age_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (age_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(age_variable, main="Original", xlab="AGE")
hist(age_log_transformed, main="Log Transformed", xlab="Log(AGE)")
hist(age_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(AGE)")
hist(age_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(AGE)")


# Example variable to transform
income_variable <- crash_data_imputed$INCOME  # Replace with your actual variable

# 1. Log Transformation
income_log_transformed <- log(income_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
```

```r
income_sqrt_transformed <- sqrt(income_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
income_box_cox_transformed <- boxcox(income_variable + 1)  # Add 1 to handle zeros, need to extract lam

income_yj_transformed <- bestNormalize(income_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (income_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(income_variable, main="Original", xlab="INCOME")
hist(income_log_transformed, main="Log Transformed", xlab="Log(INCOME)")
hist(income_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(INCOME)")
hist(income_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(INCOME)")


#OldClaim


oldclaim_variable <- crash_data_imputed$OLDCLAIM  # Replace with your actual variable

oldclaim_log_transformed <- log(oldclaim_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
oldclaim_sqrt_transformed <- sqrt(oldclaim_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
oldclaim_box_cox_transformed <- boxcox(oldclaim_variable + 1)  # Add 1 to handle zeros, need to extract

oldclaim_yj_transformed <- bestNormalize(oldclaim_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (oldclaim_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(oldclaim_variable, main="Original", xlab="oldclaim")
hist(oldclaim_log_transformed, main="Log Transformed", xlab="Log(oldclaim)")
hist(oldclaim_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(oldclaim)")
hist(oldclaim_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(oldclaim)")

# CAR AGE
car_age_variable <- crash_data_imputed$CAR_AGE # Replace with your actual variable

car_age_log_transformed <- log(car_age_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
car_age_sqrt_transformed <- sqrt(car_age_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
car_age_box_cox_transformed <- boxcox(car_age_variable + 1)  # Add 1 to handle zeros, need to extract l
```

```r
car_age_yj_transformed <- bestNormalize(car_age_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (car_age_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(car_age_variable, main="Original", xlab="CAR_AGE")
hist(car_age_log_transformed, main="Log Transformed", xlab="Log(CAR_AGE)")
hist(car_age_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(CAR_AGE)")
hist(car_age_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(CAR_AGE)")


#TRAVTIME TRANSFORMATIONS

TRAVTIME_variable <- crash_data_imputed$TRAVTIME # Replace with your actual variable

TRAVTIME_log_transformed <- log(TRAVTIME_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
TRAVTIME_sqrt_transformed <- sqrt(TRAVTIME_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
TRAVTIME_box_cox_transformed <- boxcox(TRAVTIME_variable + 1)  # Add 1 to handle zeros, need to extract

TRAVTIME_yj_transformed <- bestNormalize(TRAVTIME_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (TRAVTIME_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(TRAVTIME_variable, main="Original", xlab="TRAVTIME")
hist(TRAVTIME_log_transformed, main="Log Transformed", xlab="Log(TRAVTIME)")
hist(TRAVTIME_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(TRAVTIME)")
hist(TRAVTIME_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(TRAVTIME)")

#TIF

TIF_variable <- crash_data_imputed$TIF # Replace with your actual variable

TIF_log_transformed <- log(TIF_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
TIF_sqrt_transformed <- sqrt(TIF_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
TIF_box_cox_transformed <- boxcox(TIF_variable + 1)  # Add 1 to handle zeros, need to extract lambda

TIF_yj_transformed <- bestNormalize(TIF_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (TIF_variable + 1)  # Add 1 to handle zeros
```

```
# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(TIF_variable, main="Original", xlab="TIF")
hist(TIF_log_transformed, main="Log Transformed", xlab="Log(TIF)")
hist(TIF_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(TIF)")
hist(TIF_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(TIF)")

#MVR_PTS TRANSFORMATIONS

MVR_PTS_variable <- crash_data_imputed$MVR_PTS # Replace with your actual variable

MVR_PTS_log_transformed <- log(MVR_PTS_variable + 1)  # Add 1 to handle zeros

# 2. Square Root Transformation
MVR_PTS_sqrt_transformed <- sqrt(MVR_PTS_variable + 1)  # Add 1 to handle zeros

# 3. Box-Cox Transformation
MVR_PTS_box_cox_transformed <- boxcox(MVR_PTS_variable + 1)  # Add 1 to handle zeros, need to extract l

MVR_PTS_yj_transformed <- bestNormalize(MVR_PTS_variable, method = "yeo.johnson")$x.t

# 5. Inverse Transformation
inverse_transformed <- 1 / (MVR_PTS_variable + 1)  # Add 1 to handle zeros

# Check the results with histograms
par(mfrow=c(2,2))  # Set up the plotting area
hist(MVR_PTS_variable, main="Original", xlab="MVR_PTS")
hist(MVR_PTS_log_transformed, main="Log Transformed", xlab="Log(MVR_PTS)")
hist(MVR_PTS_sqrt_transformed, main="Square Root Transformed", xlab="Sqrt(MVR_PTS)")
hist(MVR_PTS_yj_transformed, main="Yeo-Johnson Transformed", xlab="Yeo-Johnson(MVR_PTS)")

crash_data_imputed_transformed <- crash_data_imputed %>%
    mutate(
                # Log transformation of AGE
        INCOME_transformed = bestNormalize(INCOME, method = "yeo.johnson")$x.t,      # Log transformati
        CAR_AGE_transformed = sqrt(CAR_AGE + 1),  # Square root transformation of CAR_AGE
        HOME_VAL_transformed = sqrt(HOME_VAL + 1),   # Log transformation of HOME_VAL
        OLDCLAIM_transformed=bestNormalize(oldclaim_variable, method = "yeo.johnson")$x.t,
        TRAVTIME_transformed=sqrt(TRAVTIME + 1)

        )


# Set seed for reproducibility
set.seed(123)  # You can set any number

# Create a split index
split <- sample.split(crash_data_imputed_transformed$TARGET_AMT, SplitRatio = 0.7)

# Split data into training and testing sets
train_data <- subset(crash_data_imputed_transformed, split == TRUE)
test_data <- subset(crash_data_imputed_transformed, split == FALSE)
```

```r
# Fit the model on the training data
model <- lm(TARGET_AMT ~ train_data$OLDCLAIM_transformed + train_data$CLM_FREQ + train_data$MVR_PTS + tr
summary(model)

# Predict on the testing data
predictions <- predict(model, newdata = test_data)

# Evaluate model performance
# Calculate Mean Absolute Error (MAE)
MAE <- mean(abs(predictions - test_data$TARGET_AMT))

# Calculate Mean Squared Error (MSE)
MSE <- mean((predictions - test_data$TARGET_AMT)^2)

# Calculate Root Mean Squared Error (RMSE)
RMSE <- sqrt(MSE)

# Print the performance metrics
cat("Model Performance on Testing Data:\n")
cat("Mean Absolute Error (MAE):", MAE, "\n")
cat("Mean Squared Error (MSE):", MSE, "\n")
cat("Root Mean Squared Error (RMSE):", RMSE, "\n")




# Set seed for reproducibility
set.seed(123)  # You can set any number

# Create a split index
split <- sample.split(completed_data$TARGET_AMT, SplitRatio = 0.7)

# Split data into training and testing sets
train_data <- subset(completed_data , split == TRUE)
test_data <- subset(completed_data , split == FALSE)

# Fit the model on the training data
model <- lm(TARGET_AMT ~ train_data$OLDCLAIM + train_data$CLM_FREQ + train_data$MVR_PTS + train_data$TRA
summary(model)

# Predict on the testing data
predictions <- predict(model, newdata = test_data)

# Evaluate model performance
# Calculate Mean Absolute Error (MAE)
MAE <- mean(abs(predictions - test_data$TARGET_AMT))

# Calculate Mean Squared Error (MSE)
MSE <- mean((predictions - test_data$TARGET_AMT)^2)

# Calculate Root Mean Squared Error (RMSE)
RMSE <- sqrt(MSE)
```

```r
# Print the performance metrics
cat("Model Performance on Testing Data:\n")
cat("Mean Absolute Error (MAE):", MAE, "\n")
cat("Mean Squared Error (MSE):", MSE, "\n")
cat("Root Mean Squared Error (RMSE):", RMSE, "\n")


# Transform skewed predictors and target
train_data$TARGET_AMT_log <- log(train_data$TARGET_AMT + 1)  # Log transformation to stabilize variance
train_data$TRAVTIME_sqrt <- sqrt(train_data$TRAVTIME)  # Square root transformation for TRAVTIME


enhanced_model <- lm(
  TARGET_AMT_log ~ train_data$CLM_FREQ + train_data$MVR_PTS + train_data$TRAVTIME_sqrt +
    I(train_data$MVR_PTS^2) + train_data$CLM_FREQ:train_data$MVR_PTS,  # Interaction and polynomial ter
  data = train_data
)

# Model Summary
summary(enhanced_model)

# Model Performance on Testing Data
predictions <- predict(enhanced_model, newdata = test_data)
# Convert predictions back if log-transformed
predictions <- exp(predictions) - 1

# Calculate Performance Metrics
mae <- mean(abs(predictions - test_data$TARGET_AMT))
mse <- mean((predictions - test_data$TARGET_AMT)^2)
rmse <- sqrt(mse)

list(MAE = mae, MSE = mse, RMSE = rmse)
# Set seed for reproducibility



enhanced_model <- lm(
    TARGET_AMT_log ~ train_data$CLM_FREQ + train_data$MVR_PTS + train_data$TRAVTIME_sqrt +
       I(train_data$MVR_PTS^2) + train_data$CLM_FREQ:train_data$MVR_PTS + train_data$CLM_FREQ:train_data
  data = train_data,
  weights = ifelse(train_data$CLM_FREQ > 2, 1.5, 1)
)


# Model Summary
summary(enhanced_model)

# Model Performance on Testing Data
predictions <- predict(enhanced_model, newdata = test_data)
# Convert predictions back if log-transformed
predictions <- exp(predictions) - 1

# Calculate Performance Metrics
```

```r
mae <- mean(abs(predictions - test_data$TARGET_AMT))
mse <- mean((predictions - test_data$TARGET_AMT)^2)
rmse <- sqrt(mse)

list(MAE = mae, MSE = mse, RMSE = rmse)
# Set seed for reproducibility



enhanced_model <- lm(
    TARGET_AMT_log ~ train_data$CLM_FREQ + train_data$MVR_PTS + train_data$YOJ + train_data$TIF+
       I(train_data$MVR_PTS^2) + train_data$CLM_FREQ:train_data$MVR_PTS + train_data$CLM_FREQ:train_data
  data = train_data,
  weights = ifelse(train_data$CLM_FREQ > 2, 1.5, 1)
)


# Model Summary
summary(enhanced_model)

# Model Performance on Testing Data
predictions <- predict(enhanced_model, newdata = test_data)
# Convert predictions back if log-transformed
predictions <- exp(predictions) - 1

# Calculate Performance Metrics
mae <- mean(abs(predictions - test_data$TARGET_AMT))
mse <- mean((predictions - test_data$TARGET_AMT)^2)
rmse <- sqrt(mse)

list(MAE = mae, MSE = mse, RMSE = rmse)

# Set seed for reproducibility
set.seed(123)

# Split data into training and testing sets
split <- sample.split(insurance_training_data_clean$TARGET_FLAG, SplitRatio = 0.7)
train_data <- subset(insurance_training_data_clean, split == TRUE)
test_data <- subset(insurance_training_data_clean, split == FALSE)

# Build a logistic regression model with specified predictors
# Replace predictor1, predictor2, ... with actual predictor names
logistic_model1 <- glm(TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + EDUCATION +
                       MVR_PTS + REVOKED + TIF + TRAVTIME + URBANICITY,
                       data = insurance_training_data_clean, family = binomial)

# Summary of the model
summary(logistic_model1)

# Predict on the test set
# Predict probabilities
prob_predictions <- predict(logistic_model1, newdata = test_data, type = "response")
```

```r
# Convert probabilities to binary classes with a threshold (e.g., 0.5)
class_predictions <- ifelse(prob_predictions > 0.5, 1, 0)

# Evaluate model performance
# Confusion Matrix
confusion_matrix <- table(Predicted = class_predictions, Actual = test_data$TARGET_FLAG)
print(confusion_matrix)

# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Model Accuracy:", accuracy, "\n")

# Check for missing values
# Check for missing values
missing_values <- colSums(is.na(insurance_training_data_clean))
print(missing_values[missing_values > 0])  # Print columns with missing values

# Option 1: Remove rows with missing values
insurance_training_data_clean <- na.omit(insurance_training_data_clean)

# Option 2: Impute missing values (mean for numeric columns, mode for categorical, etc.)
# For numeric columns
numeric_cols <- sapply(insurance_training_data_clean, is.numeric)
insurance_training_data_clean[numeric_cols] <- lapply(insurance_training_data_clean[numeric_cols],
                                                      function(x) ifelse(is.na(x), mean(x, na.rm = TRU

# For categorical columns (optional, if you have any)
categorical_cols <- sapply(insurance_training_data_clean, is.factor)
insurance_training_data_clean[categorical_cols] <- lapply(insurance_training_data_clean[categorical_col
                                                          function(x) ifelse(is.na(x),
                                                                             levels(x)[which.max(table
                                                                             x))


# Now you can fit your model again
logistic_model <- glm(TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + EDUCATION +
                      MVR_PTS + REVOKED + TIF + TRAVTIME + URBANICITY,
                      data = insurance_training_data_clean, family = binomial)

# Display summary of the model
summary(logistic_model)



# Stepwise feature selection to refine predictors
logistic_model_step <- stepAIC(logistic_model, direction = "both")
summary(logistic_model_step)

# Calculate VIF
vif_values <- vif(logistic_model)
print(vif_values)

# Remove predictors with high VIF
high_vif <- names(vif_values[vif_values > 5])  # Threshold for multicollinearity
```

```
if (length(high_vif) > 0) {
  # Fit a new model excluding high VIF predictors
  reduced_model <- update(logistic_model, . ~ . - one_of(high_vif))
  summary(reduced_model)
}

# Make predictions
predictions <- predict(logistic_model_step, newdata = insurance_training_data_clean, type = "response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)

# Confusion Matrix
confusion_matrix <- table(insurance_training_data_clean$TARGET_FLAG, predicted_classes)
print(confusion_matrix)

# Accuracy, Precision, Recall
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
precision <- confusion_matrix[2, 2] / (confusion_matrix[2, 2] + confusion_matrix[1, 2])
recall <- confusion_matrix[2, 2] / (confusion_matrix[2, 2] + confusion_matrix[2, 1])

cat("Accuracy:", accuracy, "\n")
cat("Precision:", precision, "\n")
cat("Recall:", recall, "\n")

# ROC curve and AUC

roc_curve <- roc(insurance_training_data_clean$TARGET_FLAG, predictions)
plot(roc_curve)
cat("AUC:", auc(roc_curve), "\n")


# Create a trainControl object
control <- trainControl(method = "cv", number = 10)

# Train the model using cross-validation
cv_model <- train(TARGET_FLAG ~ ., data = insurance_training_data_clean, method = "glm", family = "binor
print(cv_model)




# Data Cleaning: Remove rows with missing values
insurance_training_data_clean <- na.omit(insurance_training_data_clean)

# Identify numeric columns excluding TARGET_FLAG for scaling
numeric_cols <- sapply(insurance_training_data_clean, is.numeric)
numeric_cols <- names(numeric_cols[numeric_cols])
numeric_cols <- setdiff(numeric_cols, "TARGET_FLAG")

# Scale numeric predictors
insurance_training_data_clean[numeric_cols] <- scale(insurance_training_data_clean[numeric_cols])

# Fit a binary logistic regression model with different predictors
logistic_model <- glm(TARGET_FLAG ~ CAR_TYPE + HOME_VAL + KIDSDRIV + OLDCLAIM + SEX,
```

```r
                      data = insurance_training_data_clean, family = binomial)

# Display summary of the model
summary(logistic_model)

# Optional: Stepwise feature selection
logistic_model_step <- stepAIC(logistic_model, direction = "both")
summary(logistic_model_step)

# Predictions and accuracy
predicted_probs <- predict(logistic_model, type = "response")
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)

# Create a confusion matrix
confusion_matrix <- table(insurance_training_data_clean$TARGET_FLAG, predicted_classes)
print(confusion_matrix)

# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
cat("Model Accuracy:", accuracy, "\n")
completed_data_eval$TARGET_AMT <- mean(train_data$TARGET_AMT)
completed_data_eval$TARGET_AMT_log <- log(completed_data_eval$TARGET_AMT + 1)  # Log transformation to
completed_data_eval$TRAVTIME_sqrt <- sqrt(completed_data_eval$TRAVTIME)  # Square root transformation f




enhanced_model <- lm(
    TARGET_AMT_log ~ completed_data_eval$CLM_FREQ + completed_data_eval$MVR_PTS + completed_data_eval$Y
       I(completed_data_eval$MVR_PTS^2) + completed_data_eval$CLM_FREQ:completed_data_eval$MVR_PTS + co
  data = completed_data_eval,
  weights = ifelse(completed_data_eval$CLM_FREQ > 2, 1.5, 1)
)

#crash_data_imputed <- complete(imputed_data)
predictions <- predict(enhanced_model, newdata = completed_data_eval)
# Convert predictions back if log-transformed
predictions <- exp(predictions) - 1
summary(enhanced_model)
# Calculate Performance Metrics
mae <- mean(abs(predictions - test_data$TARGET_AMT))
mse <- mean((predictions - test_data$TARGET_AMT)^2)
rmse <- sqrt(mse)

list(MAE = mae, MSE = mse, RMSE = rmse)


prob_predictions <- predict(logistic_model1, newdata = insurance_evaluation_data_clean, type = "respons
class_predictions <- ifelse(prob_predictions > 0.5, 1, 0)
table(class_predictions )
```