DATA 621 - HW4:Auto Insurance Claims

2025-04-17

Required Libraries

```r
library(janitor)
library(kableExtra)
library(latex2exp)
library(psych)
library(scales)
library(stringr)
library(ggcorrplot)
library(tidyverse)
library(mice)
library(ggmice)
library(caret)
library(bestNormalize)
library(e1071)
library(car)
library(glmnet)
library(pROC)
library(Metrics)
```

Overview

In this homework assignment, you will explore, analyze and model a data set containing approximately 8000 records representing a customer at an auto insurance company. Each record has two response variables. The first response variable, TARGET_FLAG, is a 1 or a 0. A "1" means that the person was in a car crash. A zero means that the person was not in a car crash. The second response variable is TARGET_AMT. This value is zero if the person did not crash their car. But if they did crash their car, this number will be a value greater than zero. Your objective is to build multiple linear regression and binary logistic regression models on the training data to predict the probability that a person will crash their car and also the amount of money it will cost if the person does crash their car. You can only use the variables given to you (or variables that you derive from the variables provided).

Introduction

In this project, we analyze a dataset comprising around 8,000 entries, each representing a customer from an auto insurance provider. Each entry includes two key outcome variables. The first, **TARGET_FLAG**, is a binary indicator: a value of 1 indicates that the individual was involved in an automobile accident, while a value of 0 signifies no accident. The second variable, **TARGET_AMT**, represents the monetary cost associated with the accident. If no accident occurred, this amount is recorded as zero; otherwise, it reflects a positive dollar value.

Our analysis begins with an in-depth examination of the dataset, focusing on its structure and the nature of its variables. This includes identifying any missing data, assessing skewness in numerical features, and analyzing correlations among variables. Following this exploratory phase, we proceed to data preprocessing—transforming and preparing the dataset to resolve issues identified earlier.

Next, we develop two types of predictive models. First, we construct **logistic regression models** to estimate the likelihood of a customer being involved in a crash. Then, for those predicted to crash, we use **multiple linear regression models** to forecast the potential cost of the crash. To wrap up the project, we assess the performance of each model, determine which ones perform best, and use them to generate predictions on a validation dataset.

VARIABLE	**DEFINITION**	**THEORETICAL EFFECT**
`INDEX`	Identification Variable (do not use)	None
`TARGET_FLAG`	Was Car in a crash? 1=YES 0=NO	None
`TARGET_AMT`	If car was in a crash, what was the cost	None
`AGE`	Age of Driver	Very young people tend to be risky. Maybe very old people also.
`BLUEBOOK`	Value of Vehicle	Unknown effect on probability of collision, but probably effect the payout if there is a crash
`CAR_AGE`	Vehicle Age	Unknown effect on probability of collision, but probably effect the payout if there is a crash
`CAR_TYPE`	Type of Car	Unknown effect on probability of collision, but probably effect the payout if there is a crash
`CAR_USE`	Vehicle Use	Commercial vehicles are driven more, so might increase probability of collision
`CLM_FREQ`	# Claims (Past 5 Years)	The more claims you filed in the past, the more you are likely to file in the future
`EDUCATION`	Max Education Level	Unknown effect, but in theory more educated people tend to drive more safely
`HOMEKIDS`	# Children at Home	Unknown effect
`HOME_VAL`	Home Value	In theory, home owners tend to drive more responsibly
`INCOME`	Income	In theory, rich people tend to get into fewer crashes
`JOB`	Job Category	In theory, white collar jobs tend to be safer
`KIDSDRIV`	# Driving Children	When teenagers drive your car, you are more likely to get into crashes
`MSTATUS`	Marital Status	In theory, married people drive more safely
`MVR_PTS`	Motor Vehicle Record Points	If you get lots of traffic tickets, you tend to get into more crashes
`OLDCLAIM`	Total Claims (Past 5 Years)	If your total payout over the past five years was high, this suggests future payouts will be high
`PARENT1`	Single Parent	Unknown effect
`RED_CAR`	A Red Car	Urban legend says that red cars (especially red sports cars) are more risky. Is that true?
`REVOKED`	License Revoked (Past 7 Years)	If your license was revoked in the past 7 years, you probably are a more risky driver.
`SEX`	Gender	Urban legend says that women have less crashes then men. Is that true?
`TIF`	Time in Force	People who have been customers for a long time are usually more safe.
`TRAVTIME`	Distance to Work	Long drives to work usually suggest greater risk
`URBANICITY`	Home/Work Area	Unknown
`YOJ`	Years on Job	People who stay at a job for a long time are usually more safe

Data Exploration

Import Data

Upon importing the training and evaluation datasets, we find that there are 26 columns, each corresponding to one of the variables described earlier. The training dataset contains 8,161 observations, while the evaluation dataset includes 2,141 entries. A preliminary review of the columns reveals that some data cleaning and preprocessing will be necessary before we can accurately compute any summary statistics.

Table 2: Training Set

INDEX	TARGET_FLAG	TARGET_AMT	KIDSDRIV	AGE	HOMEKIDS	YOJ	INCOME	PARENT1	HOME_VAL	MSTATUS	SEX	EDU
1	0	0	0	60	0	11	$67,349	No	$0	z_No	M	PhD
2	0	0	0	43	0	11	$91,449	No	$257,252	z_No	M	z_H
4	0	0	0	35	1	10	$16,039	No	$124,191	Yes	z_F	z_H
5	0	0	0	51	0	14		No	$306,251	Yes	M	<Hi
6	0	0	0	50	0	NA	$114,986	No	$243,925	Yes	z_F	PhD
7	1	2946	0	34	1	12	$125,301	Yes	$0	z_No	z_F	Bacl

Dimensions:

8161 x 26

Table 3: Evaluation Set

INDEX	TARGET_FLAG	TARGET_AMT	KIDSDRIV	AGE	HOMEKIDS	YOJ	INCOME	PARENT1	HOME_VAL	MSTATUS	SEX	EDU
3	NA	NA	0	48	0	11	$52,881	No	$0	z_No	M	Bac
9	NA	NA	1	40	1	11	$50,815	Yes	$0	z_No	M	z_H
10	NA	NA	0	44	2	12	$43,486	Yes	$0	z_No	z_F	z_H
18	NA	NA	0	35	2	NA	$21,204	Yes	$0	z_No	M	z_H
21	NA	NA	0	59	0	12	$87,460	No	$0	z_No	M	z_H
30	NA	NA	0	46	0	14		No	$207,519	Yes	M	Bac

Dimensions:

2141 x 26

Data Wrangling

In this section, we begin making initial modifications to the training dataset. Unless stated otherwise, all adjustments made here will also be applied to the evaluation dataset to maintain consistency.

To start, we remove the **INDEX** column, as it does not contribute any meaningful information to our analysis.

Table 4: Training Set

TARGET_FLAG	TARGET_AMT	KIDSDRIV	AGE	HOMEKIDS	YOJ	INCOME	PARENT1	HOME_VAL	MSTATUS	SEX	EDUCATION
0	0	0	60	0	11	$67,349	No	$0	z_No	M	PhD
0	0	0	43	0	11	$91,449	No	$257,252	z_No	M	z_High School
0	0	0	35	1	10	$16,039	No	$124,191	Yes	z_F	z_High School
0	0	0	51	0	14		No	$306,251	Yes	M	<High School
0	0	0	50	0	NA	$114,986	No	$243,925	Yes	z_F	PhD
1	2946	0	34	1	12	$125,301	Yes	$0	z_No	z_F	Bachelors

Note:

Dropped ‘INDEX‘ column:

Next, we observe that the **INCOME**, **HOME_VAL**, **BLUEBOOK**, and **OLDCLAIM** columns are currently formatted as currency strings. To enable proper analysis, these values must be converted into a numeric format.

Table 5: Training Set: Before

INCOME	HOME_VAL	BLUEBOOK	OLDCLAIM
$67,349	$0	$14,230	$4,461
$91,449	$257,252	$14,940	$0
$16,039	$124,191	$4,010	$38,690
	$306,251	$15,440	$0
$114,986	$243,925	$18,000	$19,217
$125,301	$0	$17,430	$0

Table 6: Training Set: After

INCOME	HOME_VAL	BLUEBOOK	OLDCLAIM
67349	0	14230	4461
91449	257252	14940	0
16039	124191	4010	38690
NA	306251	15440	0
114986	243925	18000	19217
125301	0	17430	0

We also notice that several columns—**MSTATUS**, **SEX**, **EDUCATION**, **JOB**, **CAR_TYPE**, and **URBANICITY**—contain values with the prefix “z_” that should be removed for consistency and clarity.

Table 7: Training Set: Before

MSTATUS	SEX	EDUCATION	JOB	CAR_TYPE	URBANICITY
z_No	M	PhD	Professional	Minivan	Highly Urban/ Urban
z_No	M	z_High School	z_Blue Collar	Minivan	Highly Urban/ Urban
Yes	z_F	z_High School	Clerical	z_SUV	Highly Urban/ Urban
Yes	M	<High School	z_Blue Collar	Minivan	Highly Urban/ Urban
Yes	z_F	PhD	Doctor	z_SUV	Highly Urban/ Urban
z_No	z_F	Bachelors	z_Blue Collar	Sports Car	Highly Urban/ Urban

Table 8: Training Set: After

MSTATUS	SEX	EDUCATION	JOB	CAR_TYPE	URBANICITY
No	M	PhD	Professional	Minivan	Highly Urban/ Urban
No	M	High School	Blue Collar	Minivan	Highly Urban/ Urban
Yes	F	High School	Clerical	SUV	Highly Urban/ Urban
Yes	M	<High School	Blue Collar	Minivan	Highly Urban/ Urban
Yes	F	PhD	Doctor	SUV	Highly Urban/ Urban
No	F	Bachelors	Blue Collar	Sports Car	Highly Urban/ Urban

With the data values now cleaned, the next step is to verify that each variable has the appropriate data type.

In particular, we'll convert certain variables into categorical types (factors), as they represent distinct groups or categories. The specific variables to be converted are:

- `PARENT1`: Yes/No
- `MSTATUS`: Yes/No
- `SEX`: M/F
- `RED_CAR`: Yes/No (Fix capital punctuation of these values)
- `REVOKED`: Yes/No
- `EDUCATION`: High School, Bachelors, Masters, PhD (Ordered Factor as each level has an ordered precedence of completing it.)

Table 9: Training Set: Before

PARENT1	MSTATUS	SEX	RED_CAR	REVOKED	EDUCATION
No	No	M	yes	No	PhD
No	No	M	yes	No	High School
No	Yes	F	no	No	High School
No	Yes	M	yes	No	<High School
No	Yes	F	no	Yes	PhD
Yes	No	F	no	No	Bachelors

Table 10: Training Set: After

PARENT1	MSTATUS	SEX	RED_CAR	REVOKED	EDUCATION
No	No	M	Yes	No	PhD
No	No	M	Yes	No	High School
No	Yes	F	No	No	High School
No	Yes	M	Yes	No	<High School
No	Yes	F	No	Yes	PhD
Yes	No	F	No	No	Bachelors

Summary Statistics

With the dataset in good shape, we are now ready to take a deeper look at the data within.

Table 11: Summary Statistics

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
TARGET_FLAG	1	8161	0.26	0.44	0	0.20	0.00	0	1.0	1.0	1.07	-0.85	0.00
TARGET_AMT	2	8161	1504.32	4704.03	0	593.71	0.00	0	107586.1	107586.1	8.71	112.29	52.07
KIDSDRIV	3	8161	0.17	0.51	0	0.03	0.00	0	4.0	4.0	3.35	11.78	0.01
AGE	4	8155	44.79	8.63	45	44.83	8.90	16	81.0	65.0	-0.03	-0.06	0.10
HOMEKIDS	5	8161	0.72	1.12	0	0.50	0.00	0	5.0	5.0	1.34	0.65	0.01
YOJ	6	7707	10.50	4.09	11	11.07	2.97	0	23.0	23.0	-1.20	1.18	0.05
INCOME	7	7716	61898.09	47572.68	54028	56840.98	41792.27	0	367030.0	367030.0	1.19	2.13	541.58
HOME_VAL	9	7697	154867.29	129123.77	161160	144032.07	147867.11	0	885282.0	885282.0	0.49	-0.02	1471.79
TRAVTIME	14	8161	33.49	15.91	33	33.00	16.31	5	142.0	137.0	0.45	0.66	0.18
BLUEBOOK	16	8161	15709.90	8419.73	14440	15036.89	8450.82	1500	69740.0	68240.0	0.79	0.79	93.20
TIF	17	8161	5.35	4.15	4	4.84	4.45	1	25.0	24.0	0.89	0.42	0.05
OLDCLAIM	20	8161	4037.08	8777.14	0	1719.29	0.00	0	57037.0	57037.0	3.12	9.86	97.16
CLM_FREQ	21	8161	0.80	1.16	0	0.59	0.00	0	5.0	5.0	1.21	0.28	0.01
MVR_PTS	23	8161	1.70	2.15	1	1.31	1.48	0	13.0	13.0	1.35	1.38	0.02
CAR_AGE	24	7651	8.33	5.70	8	7.96	7.41	-3	28.0	31.0	0.28	-0.75	0.07

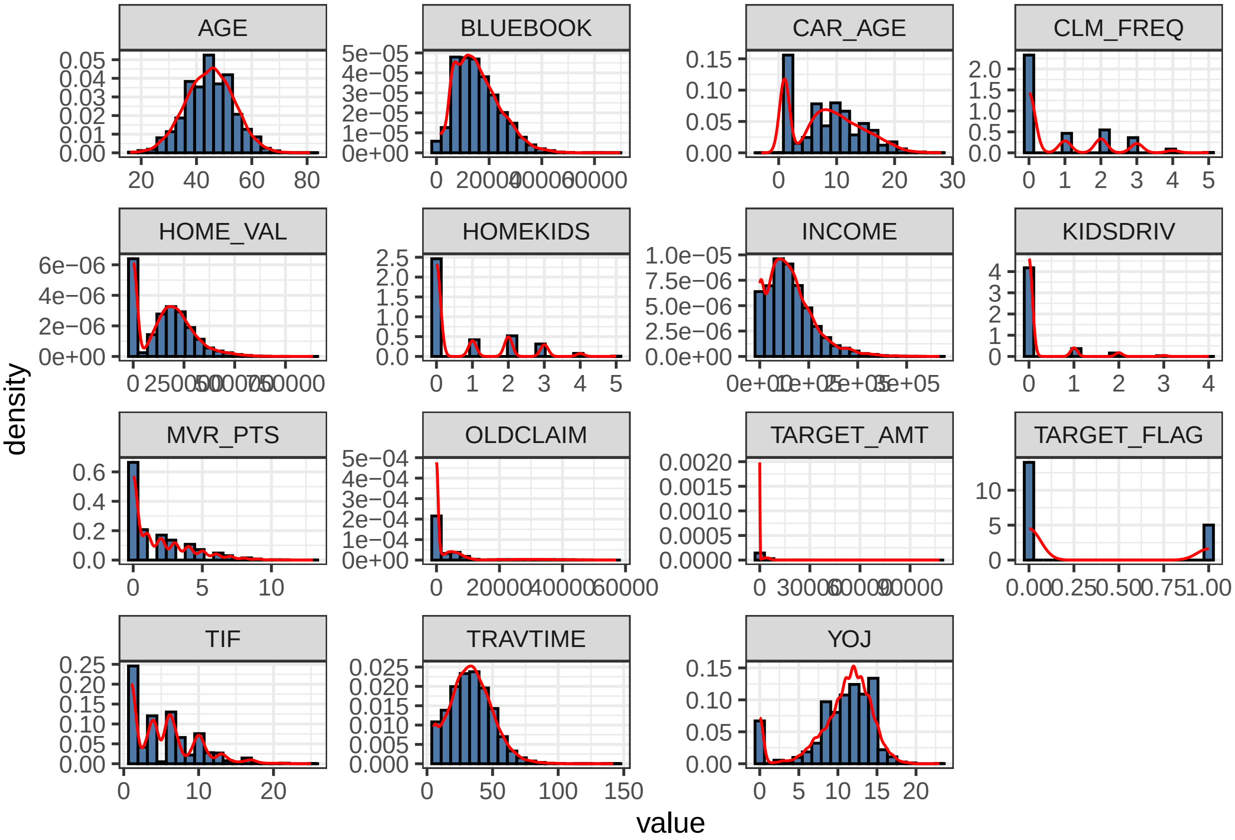
The typical customer in our dataset is about 44.79 years old. On average, they earn nearly $62,000 annually, and their homes are valued around $155,000. For those involved in accidents, the average claim amount is approximately $1,500.

Visualizations

Next, we turn our attention to visualizing the data. Since the dataset includes both continuous and categorical variables, we will use different visualization techniques tailored to each type.

Density

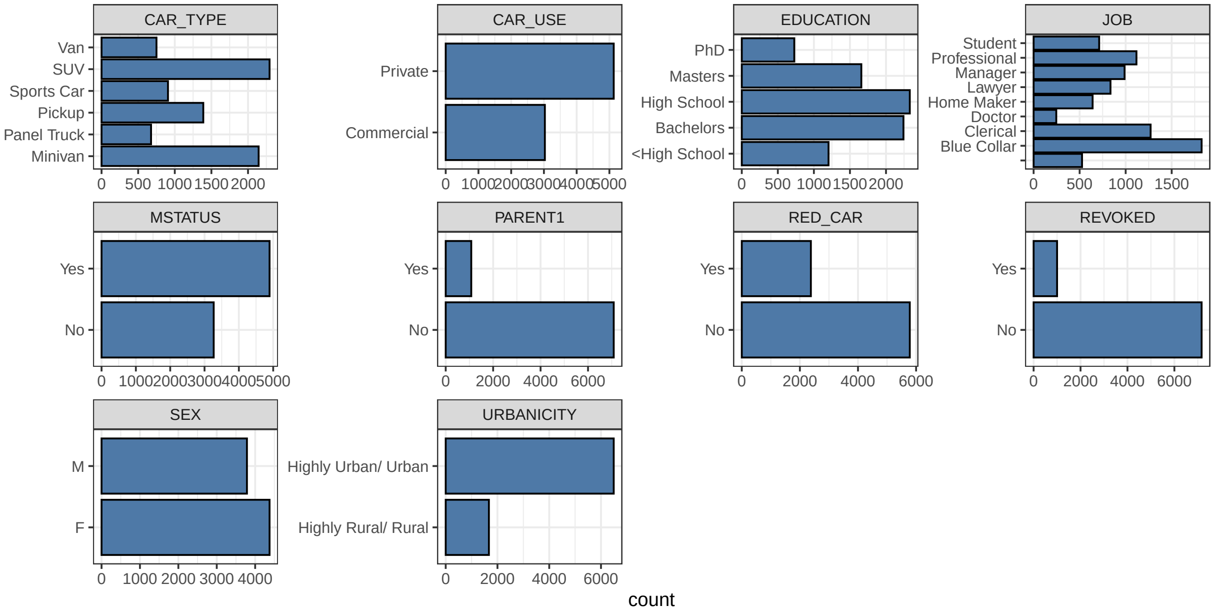
We can get a better idea of the distributions and skewness by plotting our continuous variables:



The variable AGE appears to follow a normal distribution. When examining our first response variable, TARGET_FLAG, its distribution aligns with the expected shape of a logit function, ranging between 0 and 1. Several other variables—such as BLUEBOOK, INCOME, MVR_PTS, OLDCLAIM, TARGET_AMT, TIF, and TRAVTIME—exhibit noticeable right skewness. This is reasonable, as these values are inherently non-negative and only restricted on the lower end. Additionally, variables like CAR_AGE, HOME_VAL, and YOJ display bimodal distributions. These patterns suggest that some transformations may be necessary, and we might also explore grouping strategies for the bimodal variables.

Bar Plots

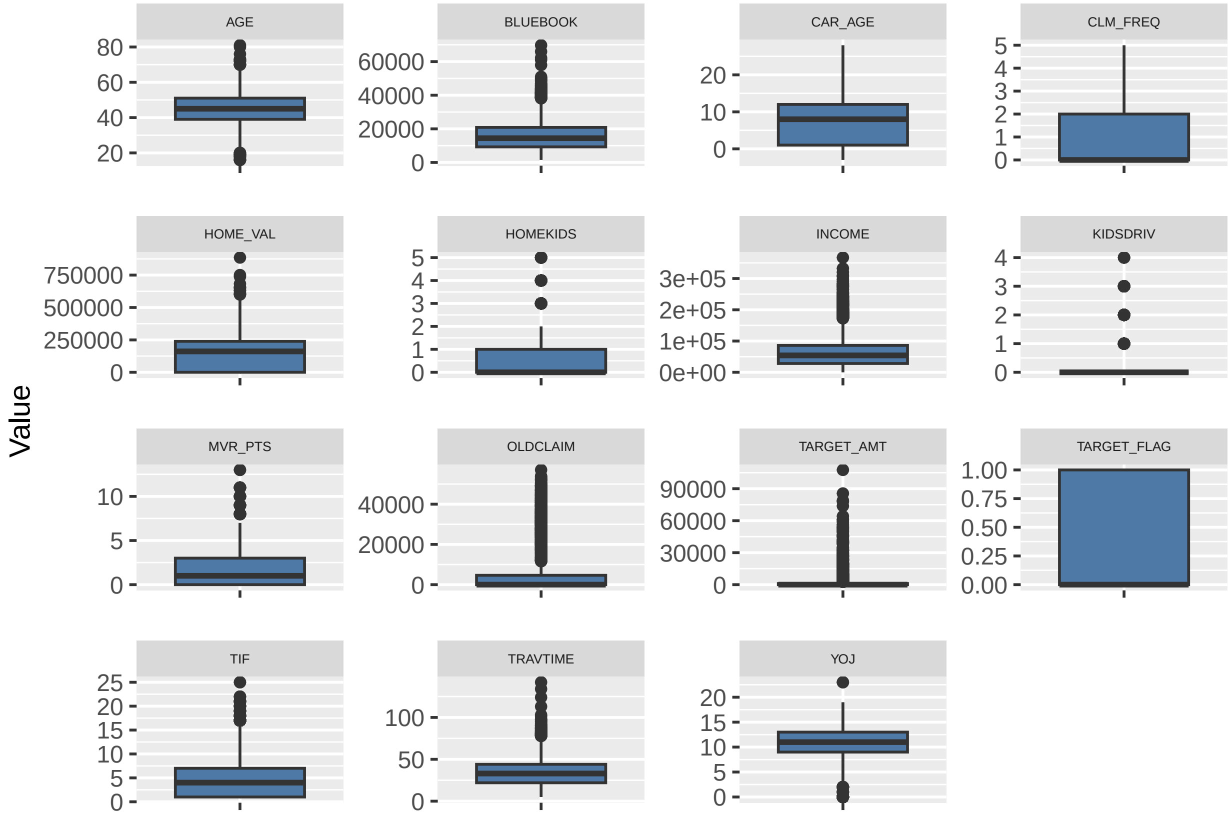
Our bar plots show us how our categorical data is divided up.



Additional insights from the data reveal that the majority of vehicles are either SUVs or Minivans. In terms of education, most drivers have attained either a High School diploma or a Bachelor's degree. The dataset also shows that most individuals reside or work in Highly Urban or Urban environments. Furthermore, vehicle usage is primarily for personal rather than commercial purposes.

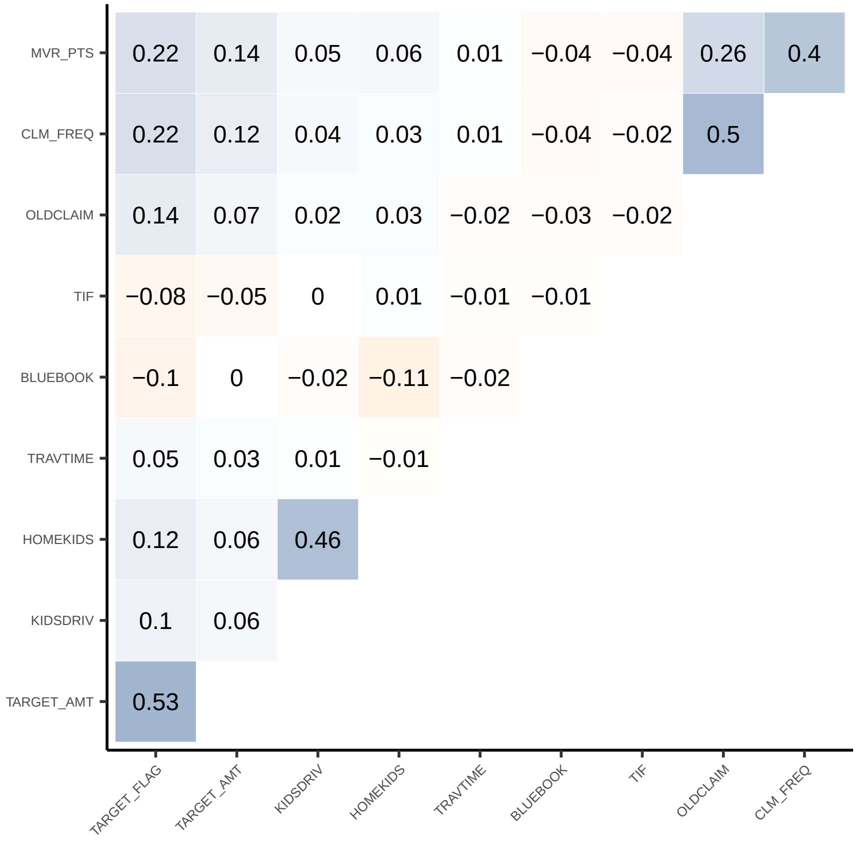
Box Plots

Visualizations can also display the presence of outliers. We expect quite a few outliers, especially when it comes to the value of cars, income of drivers, and home values.



As indicated by the box plots, there are several outliers that need to be addressed. For instance, BLUEBOOK values show that some insured vehicles are significantly more expensive than others. Similarly, HOMEKIDS and KIDSDRIV contain outliers as well—many drivers have no children, and even fewer have children who drive. Interestingly, the interquartile range for HOMEKIDS is noticeably higher than that for KIDSDRIV, which makes sense, as only a portion of the children in a household are of driving age. Given this relationship, it's reasonable to expect some correlation between these two variables—an idea we explore further in the next section.

Correlation Matrix



As expected, we have some moderately strong correlations between some of our variables. This will have to be addressed with when we build our models.

- `KIDSDRIV` and `HOMEKIDS`: As discussed, we expect multicollinearity as if you have children, they may be of age to drive already
- `MVR_PTS` and `CLM_FREQ`: The multicollinearity is intuitive as, if you have higher motor vehicle points accumulated from negative driving habits, you may be more likely to have accidents and require to file more claims than the average driver.
- `CLM_FREQ` and `OLDCLAIM`: There would be some multicollinearity since those that file more claims are likely to have a higher total claim value over the past 5 years.
- `TARGET_AMT` and `TARGET_FLAG`: Perhaps most obviously of all, since we expect TARGET_AMT to be zero if the person did not crash their car, but greater than zero if they did crash their car.

Missing Values

Table 12: Missing Values Count

AGE	YOJ	INCOME	HOME_VAL	CAR_AGE
6	454	445	464	510

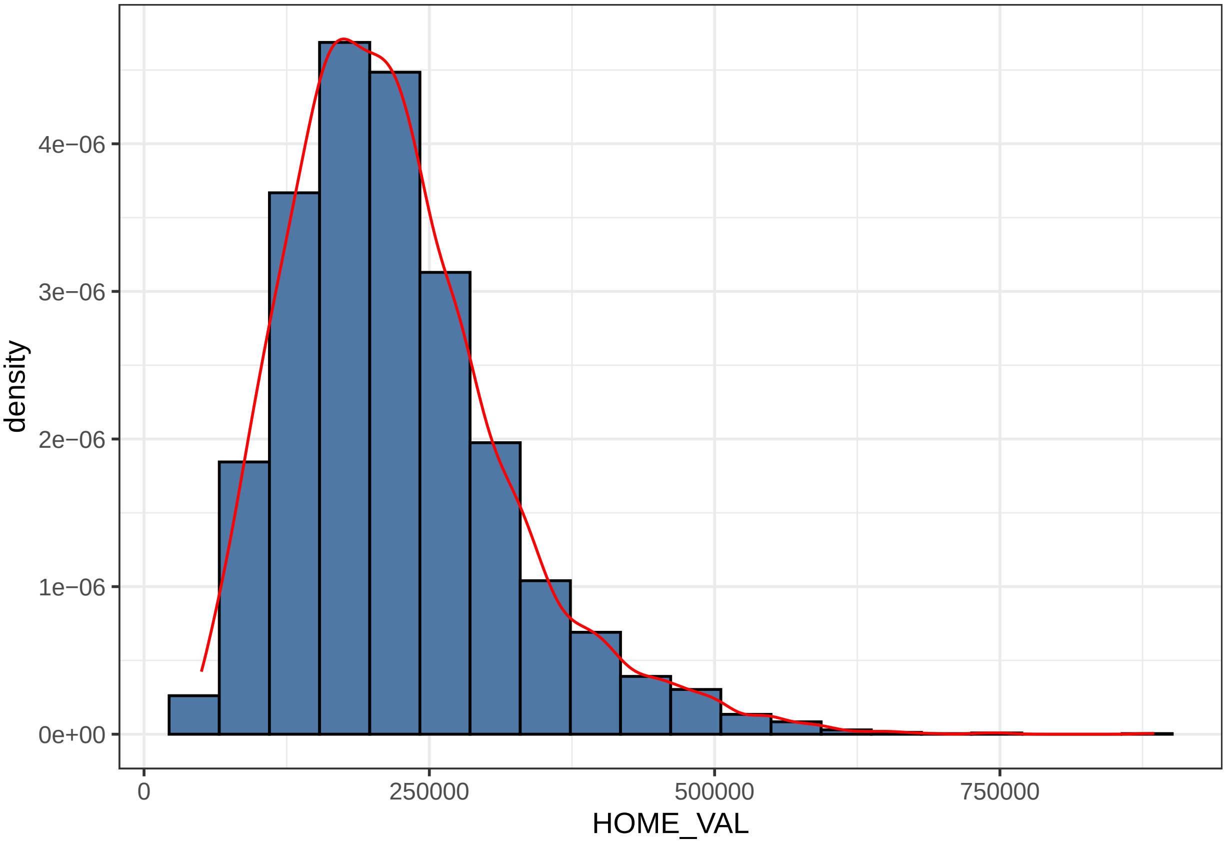
We can see we have some columns missing values.

- `AGE`: This column is only missing a few values and, given that it is a normally distributed variable, we have many options to impute them
- `YOJ`: We are missing a lot of values for how many year people have been at their job

- INCOME: We don't have how much money they are making in a year. It could be that they are not working.
- HOME_VAL: These missing values may be under the assumption they don't own a home and possibly renting. We return to this point in a moment.
- CAR_AGE: The highest amount of values we don't have is how old the car is.

There is a nuanced point about HOME_VAL. As aforementioned, these plausibly represent rentals. However, recall the density plots earlier; there were many 0s for HOME_VAL. There cannot realistically be that many houses actually valued at $0. It is possible, then, that the 0s *also* represent rentals. In that case, we should convert the 0s to missing values, and impute them as we will the other missing values for this column.

Plotting the HOME_VAL data without the 0s, we get:



And we can see a much more normal distribution than before.

We now check if there are any other suspect 0s:

Table 13: Zero Counts in Training Dataset

	Zero.Count
TARGET_FLAG	6008
TARGET_AMT	6008
KIDSDRIV	7180
AGE	0
HOMEKIDS	5289
YOJ	625
INCOME	615
PARENT1	0
HOME_VAL	0
MSTATUS	0
SEX	0
EDUCATION	0
JOB	0
TRAVTIME	0
CAR_USE	0
BLUEBOOK	0
TIF	0
CAR_TYPE	0
RED_CAR	0
OLDCLAIM	5009
CLM_FREQ	5009
REVOKED	0
MVR_PTS	3712
CAR_AGE	3
URBANICITY	0

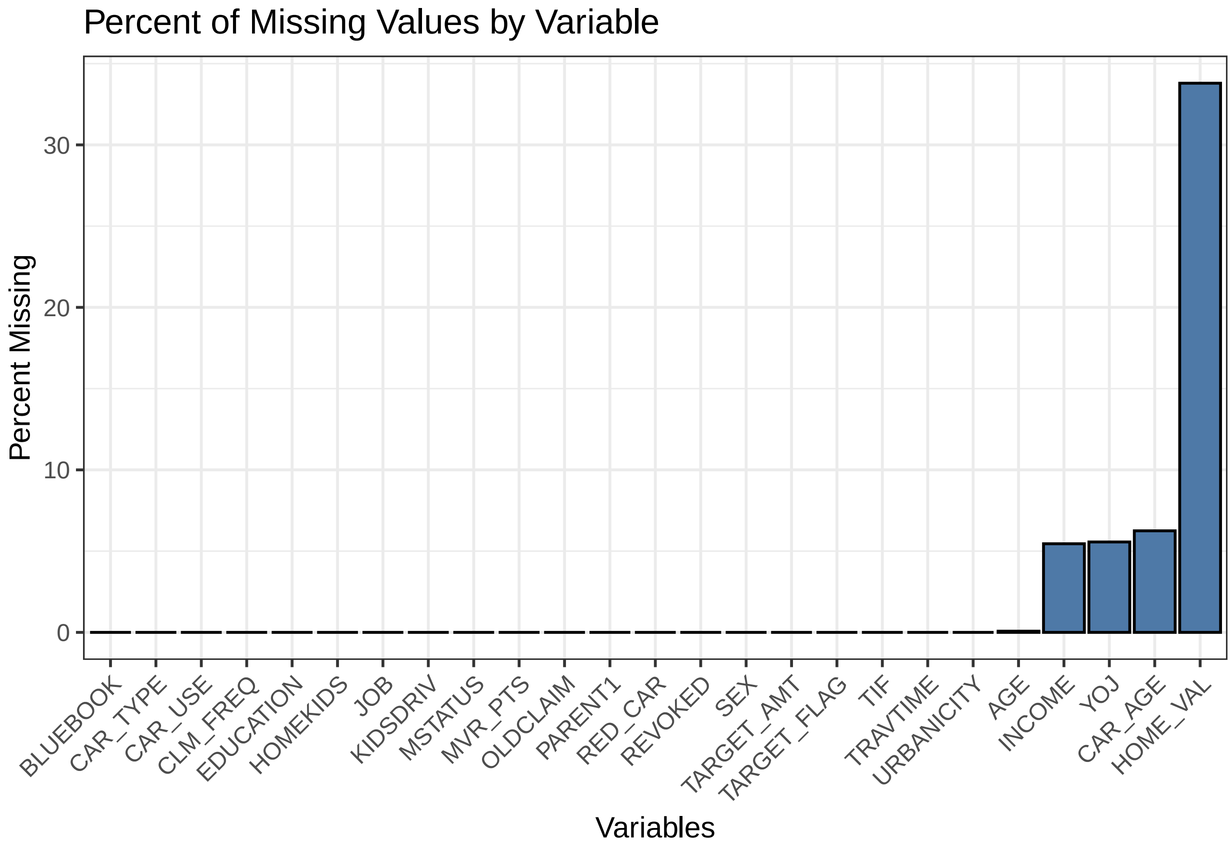
There are no other columns containing suspect 0 values. This concludes the largely exploratory phase of our analysis; we move now to consider broader transformations of the data.

Data Preparation

Although we've already carried out some initial data cleaning, this section focuses on more thorough data transformation to optimize the performance of both our multiple linear regression and logistic regression models.

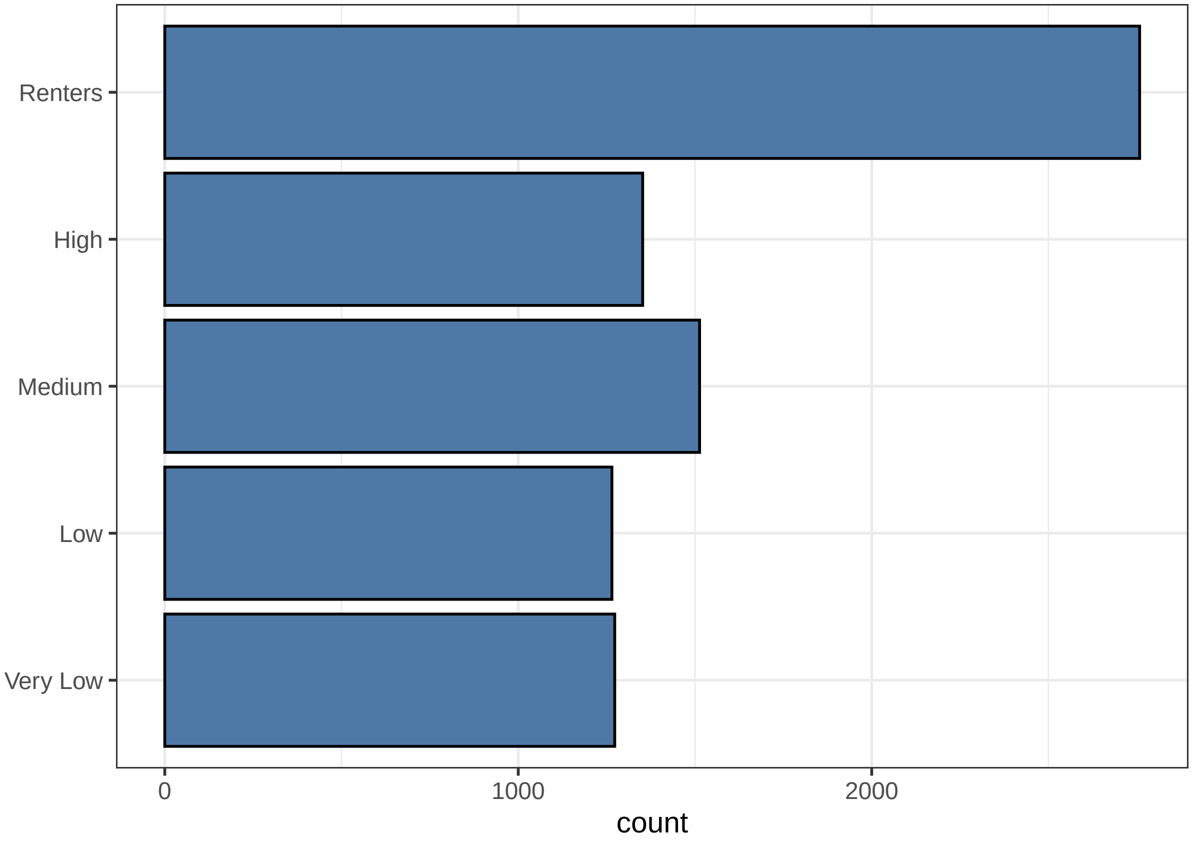
Missing Values

We pick up from the previous section by addressing missing values. As previously mentioned, the variables with missing data include **CAR_AGE**, **HOME_VAL**, **YOJ**, **INCOME**, and **AGE**. It's important to note that we recently increased the number of missing entries in **HOME_VAL** by treating zero values as **NAs**. The extent of missing data across these columns is illustrated below:



With the exception of HOME_VAL, the remaining variables have relatively few missing values—each with less than 6% missing. We'll soon move on to imputing those. However, it's important to take a closer look at HOME_VAL again. We've assumed that missing values in this column likely correspond to renters, which is why we previously replaced zeros with NAs. Given that the absence of a value here is likely informative, it would be inappropriate to impute it. Instead, we'll treat HOME_VAL as a categorical variable and retain the missing entries as their own category.

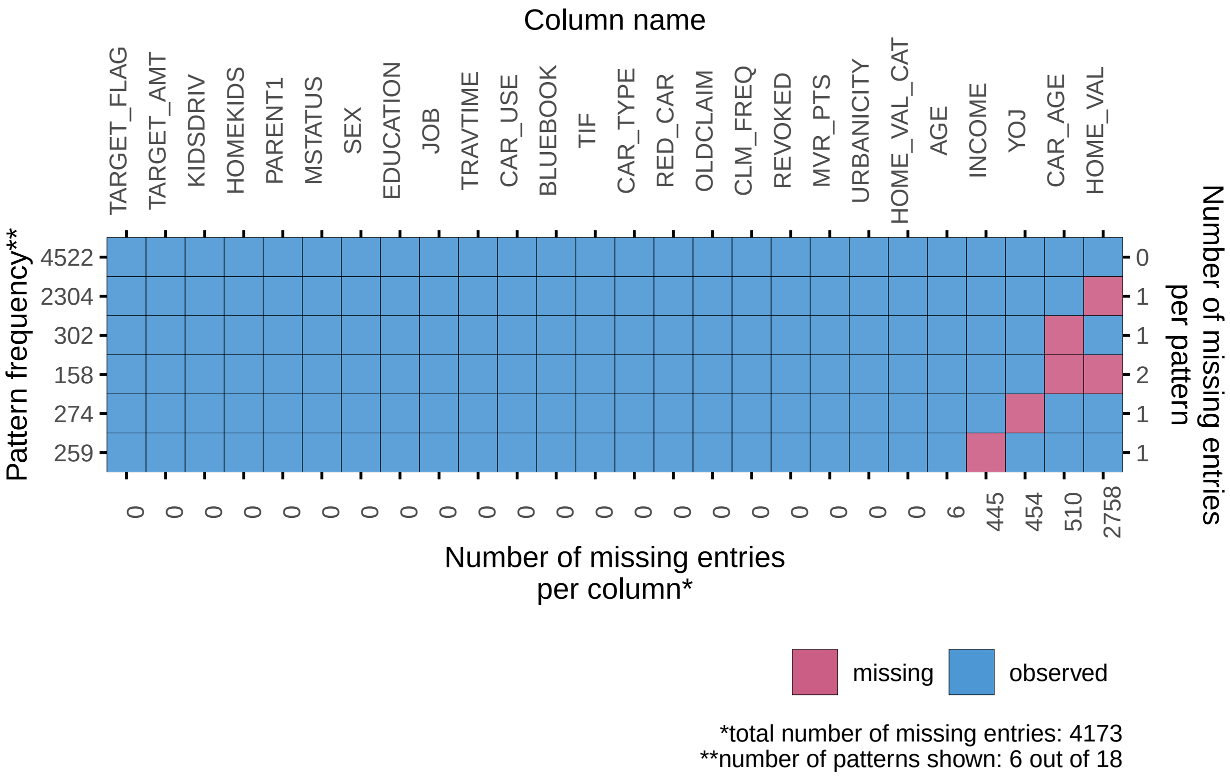
After converting, we can do another bar plot:



And we see that the data is divided fairly evenly, although “Renters” is the largest category.

Let’s investigate patterns potentially underlying the missing values:

```r
plot_pattern(train, square = TRUE, rotate = TRUE, npat = 6)
```



To begin, we observe that the pattern of missing data is largely concentrated in **HOME_VAL**, which reinforces our earlier assumption that these missing values likely correspond to renters. This lends support to the decision to treat HOME_VAL as a categorical variable. As a result, we can confidently discard the original continuous version of HOME_VAL from our analysis.

Secondly, and on a broader level, the observed missing data patterns support the use of **MICE** (Multiple Imputation by Chained Equations) for handling missing values. Some variables show co-occurring missingness, indicating possible inter-variable relationships, while others have missing data in isolation. Given this variability, a uniform imputation method would be insufficient. MICE is preferred because it tailors the imputation process to each variable's specific pattern of missingness.

Imputations

Before we can impute missing values, we perform the train-test split to avoid data leakage:

We then use MICE to impute. Critically, we ignore the test values when imputing for both sets, to avoid data leakage.

```
## Warning: Number of logged events: 4
## Warning: Number of logged events: 4
```

Table 14: Summary Statistics Comparison Across Datasets

Variable_Stat	Dataset (Pre-Imputations)	Train Imputed	Test Imputed
CAR_AGE_min	-3.000000	-3.000000	0.00000
CAR_AGE_q1	1.000000	1.000000	1.00000
CAR_AGE_median	8.000000	8.000000	8.00000
CAR_AGE_mean	8.347639	8.350884	8.29469
CAR_AGE_q3	12.000000	12.000000	12.00000
CAR_AGE_max	27.000000	27.000000	28.00000
YOJ_min	0.000000	0.000000	0.00000
YOJ_q1	9.000000	9.000000	9.00000
YOJ_median	11.000000	11.000000	11.00000
YOJ_mean	10.499536	10.510100	10.48987
YOJ_q3	13.000000	13.000000	13.00000
YOJ_max	23.000000	23.000000	19.00000
INCOME_min	0.000000	0.000000	0.00000
INCOME_q1	28127.500000	27907.000000	27684.00000
INCOME_median	54007.000000	53660.000000	53770.00000
INCOME_mean	62215.300443	62052.339331	60664.00613
INCOME_q3	85865.500000	85734.000000	85384.00000
INCOME_max	332339.000000	332339.000000	367030.00000
AGE_min	16.000000	16.000000	17.00000
AGE_q1	39.000000	39.000000	39.00000
AGE_median	45.000000	45.000000	45.00000
AGE_mean	44.781573	44.773744	44.80466
AGE_q3	51.000000	51.000000	51.00000
AGE_max	81.000000	81.000000	76.00000

The summary statistics are quite promising. For the most part, the values remain consistent across all three datasets, suggesting that the distributions in the imputed datasets closely reflect those of the original. This holds true for both the means and medians. Additionally, it appears that outliers and edge cases have been managed appropriately.

Outliers and Transformations

During the data exploration phase, we conducted a preliminary review of outliers. While the outlier values appeared to be valid, they do contribute to increased skewness in the data, which can negatively affect both logistic and linear regression models. To address this, we applied various data transformations. In evaluating skewness, we consider values with an absolute skewness above 1 to be **heavily skewed**, those between ±0.5 and ±1 to be **moderately skewed**, and values between 0 and ±0.5 to be **lightly skewed**.

We can assess the most appropriate transformations for each variable using the bestNormalize function.

Table 15: Best Transformations

Variable	Transformation
BLUEBOOK	orderNorm
INCOME	orderNorm
MVR_PTS	sqrt_x
OLDCLAIM	center_scale
TIF	yeojohnson
TRAVTIME	boxcox
YOJ	sqrt_x
CLM_FREQ	sqrt_x
CAR_AGE	yeojohnson

Once again, it's crucial to avoid data leakage by ensuring that all transformation parameters are derived solely from the training set. These same parameters should then be applied consistently to the evaluation and test sets.

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(...)`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

Table 16: Pre and Post Transformation Skewness Comparison

Variable	Pre-Transformation Skew	Post-Transformation Skew
BLUEBOOK	0.791	0.042
INCOME	1.188	0.146
MVR_PTS	1.337	0.393
OLDCLAIM	3.190	3.190
TIF	0.890	-0.034
TRAVTIME	0.471	-0.043
YOJ	-1.205	-2.254
CLM_FREQ	1.217	0.711
CAR_AGE	0.272	-0.187

In general, the applied transformations successfully reduced skewness to more acceptable levels across most variables. However, OLDCLAIM—the most skewed variable—showed little to no improvement. Despite this, the overall distribution of the dataset is now significantly closer to normality, which should enhance the performance of our models moving forward.

Outliers

Despite applying transformations, some outliers still remain. To address this, we will use outlier replacement techniques. As before, it's essential to calculate the replacement thresholds based solely on the training set and apply them consistently across all datasets. This approach ensures we prevent any risk of data leakage.

Encoding, Center/Scale/NearZeroVariance

The last step in our data preparation process involves encoding categorical variables using one-hot encoding (OHC). Additionally, we center and scale (CS) all continuous variables to prevent extreme values from skewing the model—using statistics derived solely from the training data to avoid data leakage. We also check continuous variables for near-zero variance (NZV), as such features provide little predictive power. For ordinal variables, we treat them as continuous since the spacing between values is both consistent and meaningful. Both centering/scaling and NZV filtering can be streamlined through a single step in a preprocessing pipeline.

The dataframes are now fully processed. We are ready to move on to the modeling phase.

Modeling

With preprocessing complete, we're now ready to begin modeling. In the first phase, we'll develop **logistic regression models** to predict the likelihood of a person being involved in a car accident. In the second phase, we'll construct **multiple linear regression models** to estimate the payout amount in cases where a crash has occurred.

Logistic regression

As a reminder, our initial objective is to predict whether a driver was involved in a crash. Since our dataset includes both original and transformed versions of some variables, it's important to ensure we don't include duplicate information in the model. To maintain clarity and prevent redundancy, we will explicitly categorize the variables being used.

Table 17: Variables Summary

Category	Variables
Encoded	SEX.F, SEX.M, EDUCATION.L, EDUCATION.Q, EDUCATION.C, JOBBlue Collar, JOBClerical, JOBDoctor, JOBHome Maker, JOBLawyer, JC
Original	KIDSDRIV, AGE, HOMEKIDS, YOJ, INCOME, PARENT1, MSTATUS, SEX, EDUCATION, JOB, TRAVTIME, CAR_USE, BLUEBOOK, TIF,
Transformed	BLUEBOOK_transformed, INCOME_transformed, MVR_PTS_transformed, OLDCLAIM_transformed, TIF_transformed, TRAVTIME_transfo

We must also proceed with caution due to the **One-Hot Encoding (OHC)** applied to our categorical variables. This encoding can introduce **multicollinearity**, especially among the resulting dummy variables. For binary categorical variables, we can safely drop one of the two encoded columns, as the remaining one still captures all necessary information. However, for variables with more than two categories, multicollinearity becomes more complex and may require more advanced handling strategies—which we'll address shortly.

Model 1 To begin, we'll construct a straightforward model to establish a baseline understanding of our data. We'll focus on using the **transformed variables**, as they are all continuous and generally more manageable. Additionally, we expect these transformed features to outperform their original versions in terms of model effectiveness.

```
simple_model1 <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_columns, "TAR
summary(simple_model1)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(transformed_columns, encoded_columns, "TARGET_FLAG")])
##
## Coefficients: (9 not defined because of singularities)
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.01095    0.13811   0.079 0.936788
## BLUEBOOK_transformed            -0.21614    0.02211  -9.774  < 2e-16 ***
## INCOME_transformed              -0.23605    0.03342  -7.063 1.63e-12 ***
## MVR_PTS_transformed              0.24153    0.01780  13.567  < 2e-16 ***
## OLDCLAIM_transformed            -0.12031    0.01905  -6.314 2.71e-10 ***
## TIF_transformed                 -0.23223    0.01574 -14.758  < 2e-16 ***
## TRAVTIME_transformed             0.26365    0.01626  16.219  < 2e-16 ***
## YOJ_transformed                 -0.06911    0.01941  -3.560 0.000371 ***
## CLM_FREQ_transformed             0.36465    0.02707  13.470  < 2e-16 ***
## CAR_AGE_transformed             -0.05784    0.02168  -2.667 0.007642 **
## SEX.F                            0.03393    0.05864   0.579 0.562906
## SEX.M                                 NA         NA      NA       NA
## EDUCATION.L                     -0.19862    0.08600  -2.309 0.020918 *
## EDUCATION.Q                      0.27364    0.04913   5.569 2.56e-08 ***
## EDUCATION.C                      0.14623    0.04168   3.508 0.000451 ***
## `JOBBlue Collar`                 0.48362    0.09377   5.157 2.50e-07 ***
## JOBClerical                      0.55199    0.10098   5.466 4.59e-08 ***
## JOBDoctor                       -0.18705    0.13395  -1.396 0.162582
## `JOBHome Maker`                  0.28367    0.11372   2.495 0.012612 *
## JOBLawyer                        0.39836    0.09065   4.395 1.11e-05 ***
## JOBManager                      -0.35241    0.08921  -3.950 7.80e-05 ***
## JOBProfessional                  0.29475    0.09079   3.247 0.001168 **
## JOBStudent                       0.06732    0.11540   0.583 0.559666
## CAR_USECommercial                0.80234    0.04922  16.301  < 2e-16 ***
## CAR_USEPrivate                        NA         NA      NA       NA
## CAR_TYPEMinivan                 -0.66738    0.06764  -9.866  < 2e-16 ***
## `CAR_TYPEPanel Truck`           -0.09701    0.07717  -1.257 0.208745
## CAR_TYPEPickup                  -0.14514    0.06721  -2.159 0.030820 *
## `CAR_TYPESports Car`             0.23258    0.09199   2.528 0.011460 *
## CAR_TYPESUV                      0.05097    0.08380   0.608 0.543067
## CAR_TYPEVan                           NA         NA      NA       NA
## `URBANICITYHighly Rural/ Rural` -2.48122    0.06143 -40.391  < 2e-16 ***
## `URBANICITYHighly Urban/ Urban`       NA         NA      NA       NA
## `HOME_VAL_CAT.Very Low`         -0.40163    0.05783  -6.945 3.78e-12 ***
## HOME_VAL_CAT.Low              -0.38302    0.05654  -6.774 1.25e-11 ***
## HOME_VAL_CAT.Medium           -0.35564    0.05319  -6.686 2.29e-11 ***
## HOME_VAL_CAT.High             -0.37938    0.06557  -5.786 7.23e-09 ***
## HOME_VAL_CAT.Renters                NA         NA      NA       NA    
## RED_CAR.No                    -0.01641    0.04639  -0.354 0.723475    
## RED_CAR.Yes                         NA         NA      NA       NA    
## REVOKED.No                    -0.88415    0.04934 -17.919  < 2e-16 ***
## REVOKED.Yes                         NA         NA      NA       NA    
## PARENT1.No                    -0.71334    0.04898 -14.564  < 2e-16 ***
## PARENT1.Yes                         NA         NA      NA       NA    
## MSTATUS.No                     0.35475    0.04281   8.286  < 2e-16 ***
## MSTATUS.Yes                         NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 33055  on 28564  degrees of freedom
## Residual deviance: 25420  on 28528  degrees of freedom
## AIC: 25494
## 
## Number of Fisher Scoring iterations: 5
```

Given the results, we'll proceed by eliminating variables that contribute to multicollinearity. For binary variables, we can remove either one of the two encoded columns, as they carry equivalent information. For variables with more than two categories, we'll drop the one that shows the weakest correlation with the target variable, under the assumption that it will have the least influence on the model's performance.

Table 18: Correlations with TARGET FLAG

	x
SEX.F	0.0318414
SEX.M	-0.0318414
EDUCATION.L	-0.1316931
EDUCATION.Q	0.0172592
EDUCATION.C	0.0841712
JOBBlue Collar	0.1043799
JOBClerical	0.0261655
JOBDoctor	-0.0530153
JOBHome Maker	0.0123769
JOBLawyer	-0.0629416
JOBManager	-0.1006736
JOBProfessional	-0.0337543
JOBStudent	0.0734056
CAR_USECommercial	0.1427534
CAR_USEPrivate	-0.1427534
CAR_TYPEMinivan	-0.1339852
CAR_TYPEPanel Truck	-0.0020186
CAR_TYPEPickup	0.0481774
CAR_TYPESports Car	0.0574468
CAR_TYPESUV	0.0515462
CAR_TYPEVan	0.0018546
URBANICITYHighly Rural/ Rural	-0.2342684
URBANICITYHighly Urban/ Urban	0.2342684
HOME_VAL_CAT.Very Low	0.0047065
HOME_VAL_CAT.Low	-0.0167182
HOME_VAL_CAT.Medium	-0.0648757
HOME_VAL_CAT.High	-0.1241068
HOME_VAL_CAT.Renters	0.1585339
RED_CAR.No	0.0093489
RED_CAR.Yes	-0.0093489
REVOKED.No	-0.1433965
REVOKED.Yes	0.1433965
PARENT1.No	-0.1697927
PARENT1.Yes	0.1697927
MSTATUS.No	0.1473048
MSTATUS.Yes	-0.1473048
TARGET_FLAG	1.0000000

Again, with the binary columns, we can just remove any one them. Thus, we'll remove:

- SEX.M
- CAR_USEPrivate
- `URBANICITYHighly Urban/ Urban`
- RED_CAR.Yes
- REVOKED.Yes
- PARENT1.Yes
- MSTATUS.Yes

Of the categorical, but not binary, variables we will remove the least informative:

- EDUCATION.Q
- JOBHome Maker
- CAR_TYPEVan
- HOME_VAL_CAT.Low

We now rerun the simple model without these columns to get a look at some of the coefficients.

```r
simple_model2 <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_columns_filte
summary(simple_model2)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(transformed_columns, encoded_columns_filtered, "TARGET_FLAG")])
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -0.18873    0.12457  -1.515  0.12975
## BLUEBOOK_transformed              -0.21500    0.02211  -9.724  < 2e-16 ***
## INCOME_transformed                -0.23887    0.03188  -7.492 6.78e-14 ***
## MVR_PTS_transformed                0.24278    0.01779  13.646  < 2e-16 ***
## OLDCLAIM_transformed              -0.12386    0.01903  -6.510 7.53e-11 ***
## TIF_transformed                   -0.23209    0.01571 -14.771  < 2e-16 ***
## TRAVTIME_transformed               0.26202    0.01625  16.128  < 2e-16 ***
## YOJ_transformed                   -0.07759    0.01910  -4.062 4.86e-05 ***
## CLM_FREQ_transformed               0.36438    0.02706  13.466  < 2e-16 ***
## CAR_AGE_transformed               -0.06327    0.02161  -2.928  0.00342 **
## SEX.F                              0.04894    0.05843   0.838  0.40224
## EDUCATION.L                       -0.36111    0.08216  -4.395 1.11e-05 ***
## EDUCATION.C                        0.11193    0.04024   2.781  0.00541 **
## `JOBBlue Collar`                   0.27853    0.07089   3.929 8.54e-05 ***
## JOBClerical                        0.32787    0.07039   4.658 3.19e-06 ***
## JOBDoctor                         -0.08225    0.12332  -0.667  0.50479
## JOBLawyer                          0.22851    0.07555   3.024  0.00249 **
## JOBManager                        -0.55231    0.07098  -7.781 7.18e-15 ***
## JOBProfessional                    0.03946    0.06671   0.592  0.55417
## JOBStudent                        -0.16329    0.08025  -2.035  0.04187 *
## CAR_USECommercial                  0.74433    0.04784  15.559  < 2e-16 ***
## CAR_TYPEMinivan                   -0.68481    0.06747 -10.150  < 2e-16 ***
## `CAR_TYPEPanel Truck`             -0.08222    0.07696  -1.068  0.28535
## CAR_TYPEPickup                    -0.14488    0.06716  -2.157  0.03099 *
## `CAR_TYPESports Car`               0.20891    0.09179   2.276  0.02285 *
## CAR_TYPESUV                        0.02260    0.08356   0.270  0.78683
## `URBANICITYHighly Rural/ Rural` -2.47184    0.06132 -40.310  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`            0.02789    0.05945   0.469  0.63893
## HOME_VAL_CAT.Medium                0.01847    0.05768   0.320  0.74885
## HOME_VAL_CAT.High                  0.02404    0.07368   0.326  0.74422
## HOME_VAL_CAT.Renters               0.40193    0.05632   7.136 9.58e-13 ***
## RED_CAR.No                        -0.01468    0.04631  -0.317  0.75120
## REVOKED.No                        -0.88513    0.04926 -17.967  < 2e-16 ***
## PARENT1.No                        -0.71023    0.04892 -14.518  < 2e-16 ***
## MSTATUS.No                         0.34740    0.04276   8.125 4.46e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33055  on 28564  degrees of freedom
## Residual deviance: 25452  on 28530  degrees of freedom
## AIC: 25522
##
## Number of Fisher Scoring iterations: 5
```

At this point, we've eliminated all singularities from the model. However, since the **Deviance** and **AIC** remain unchanged, it's clear that the model itself hasn't improved—only the **stability and interpretability** of the coefficients has, thanks to the removal of perfect multicollinearity. That said, several variables remain statistically insignificant and may be detracting from model performance. Additionally, although perfect multicollinearity has been addressed, some degree of **collinearity** still exists, particularly among the categorical predictors. To further refine the model, we'll use a combination of the **`vif()` function** and the previously generated **correlation matrix**.

Table 19: VIF Values simple model 2

	x
BLUEBOOK_transformed	1.994699
INCOME_transformed	3.768944
MVR_PTS_transformed	1.195063
OLDCLAIM_transformed	1.772401
TIF_transformed	1.011992
TRAVTIME_transformed	1.037126
YOJ_transformed	1.650938
CLM_FREQ_transformed	1.667716
CAR_AGE_transformed	1.924926
SEX.F	3.509376
EDUCATION.L	3.760556
EDUCATION.C	1.456489
‘JOBBlue Collar‘	3.967767
JOBClerical	2.760673
JOBDoctor	1.377429
JOBLawyer	1.992265
JOBManager	1.701806
JOBProfessional	2.081311
JOBStudent	2.417376
CAR_USECommercial	2.319161
CAR_TYPEMinivan	3.059621
‘CAR_TYPEPanel Truck‘	1.972728
CAR_TYPEPickup	2.855013
‘CAR_TYPESports Car‘	3.726209
CAR_TYPESUV	6.081267
‘URBANICITYHighly Rural/ Rural‘	1.132941
‘HOME_VAL_CAT.Very Low‘	1.958913
HOME_VAL_CAT.Medium	1.981314
HOME_VAL_CAT.High	2.337253
HOME_VAL_CAT.Renters	3.149109
RED_CAR.No	1.832978
REVOKED.No	1.328309
PARENT1.No	1.367325
MSTATUS.No	1.878276

Given this information, we opt to remove CAR_TYPESUV which has a VIF value of over 6. We will then fit a new model.

```
encoded_columns_filtered2 <- setdiff(encoded_columns_filtered, "CAR_TYPESUV")

transformed_model <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_columns_fi

summary(transformed_model)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(transformed_columns, encoded_columns_filtered2, "TARGET_FLAG")])
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -0.18005    0.12035  -1.496 0.134635
## BLUEBOOK_transformed              -0.21764    0.01982 -10.979  < 2e-16 ***
## INCOME_transformed                -0.23931    0.03184  -7.516 5.65e-14 ***
## MVR_PTS_transformed                0.24268    0.01779  13.643  < 2e-16 ***
## OLDCLAIM_transformed              -0.12394    0.01902  -6.514 7.30e-11 ***
## TIF_transformed                   -0.23203    0.01571 -14.769  < 2e-16 ***
## TRAVTIME_transformed               0.26213    0.01624  16.139  < 2e-16 ***
## YOJ_transformed                   -0.07744    0.01909  -4.056 4.98e-05 ***
## CLM_FREQ_transformed               0.36456    0.02705  13.476  < 2e-16 ***
## CAR_AGE_transformed               -0.06317    0.02161  -2.923 0.003462 **
## SEX.F                              0.05773    0.04855   1.189 0.234444
## EDUCATION.L                       -0.36003    0.08206  -4.387 1.15e-05 ***
## EDUCATION.C                        0.11249    0.04019   2.799 0.005125 **
## `JOBBlue Collar`                   0.28145    0.07006   4.017 5.88e-05 ***
## JOBClerical                        0.32899    0.07026   4.682 2.84e-06 ***
## JOBDoctor                         -0.08033    0.12312  -0.652 0.514107
## JOBLawyer                          0.23019    0.07530   3.057 0.002235 **
## JOBManager                        -0.55111    0.07084  -7.780 7.28e-15 ***
## JOBProfessional                    0.04062    0.06657   0.610 0.541738
## JOBStudent                        -0.16164    0.08001  -2.020 0.043362 *
## CAR_USECommercial                  0.74180    0.04691  15.815  < 2e-16 ***
## CAR_TYPEMinivan                   -0.69801    0.04656 -14.993  < 2e-16 ***
## `CAR_TYPEPanel Truck`             -0.08647    0.07531  -1.148 0.250929
## CAR_TYPEPickup                    -0.15743    0.04851  -3.245 0.001174 **
## `CAR_TYPESports Car`               0.18854    0.05243   3.596 0.000323 ***
## `URBANICITYHighly Rural/ Rural` -2.47185    0.06132 -40.309  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`            0.02773    0.05945   0.466 0.640878
## HOME_VAL_CAT.Medium                0.01831    0.05767   0.318 0.750863
## HOME_VAL_CAT.High                  0.02359    0.07366   0.320 0.748791
## HOME_VAL_CAT.Renters               0.40144    0.05629   7.132 9.92e-13 ***
## RED_CAR.No                        -0.01423    0.04628  -0.307 0.758553
## REVOKED.No                        -0.88531    0.04926 -17.973  < 2e-16 ***
## PARENT1.No                        -0.70998    0.04891 -14.516  < 2e-16 ***
## MSTATUS.No                         0.34759    0.04275   8.131 4.27e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 33055  on 28564  degrees of freedom
## Residual deviance: 25452  on 28531  degrees of freedom
## AIC: 25520
##
## Number of Fisher Scoring iterations: 5
```

Now let’s remove SEX and RED_CAR, both of which are not significant predictors.

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(transformed_columns, encoded_columns_filtered3, "TARGET_FLAG")])
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -0.13728    0.11203  -1.225 0.220423
## BLUEBOOK_transformed           -0.21369    0.01955 -10.933  < 2e-16 ***
## INCOME_transformed             -0.24035    0.03183  -7.552 4.29e-14 ***
## MVR_PTS_transformed             0.24287    0.01779  13.656  < 2e-16 ***
## OLDCLAIM_transformed           -0.12410    0.01902  -6.524 6.85e-11 ***
## TIF_transformed                -0.23201    0.01571 -14.772  < 2e-16 ***
## TRAVTIME_transformed            0.26224    0.01624  16.148  < 2e-16 ***
## YOJ_transformed                -0.07777    0.01909  -4.074 4.61e-05 ***
## CLM_FREQ_transformed            0.36460    0.02705  13.478  < 2e-16 ***
## CAR_AGE_transformed            -0.06236    0.02160  -2.887 0.003886 **
## EDUCATION.L                    -0.35950    0.08206  -4.381 1.18e-05 ***
## EDUCATION.C                     0.11662    0.04004   2.912 0.003590 **
## `JOBBlue Collar`                0.28242    0.07004   4.032 5.53e-05 ***
## JOBClerical                     0.32340    0.07011   4.613 3.98e-06 ***
## JOBDoctor                      -0.08828    0.12289  -0.718 0.472529
## JOBLawyer                       0.22609    0.07521   3.006 0.002644 **
## JOBManager                     -0.55549    0.07064  -7.864 3.73e-15 ***
## JOBProfessional                 0.03572    0.06642   0.538 0.590718
## JOBStudent                     -0.16340    0.07993  -2.044 0.040930 *
## CAR_USECommercial               0.73081    0.04600  15.888  < 2e-16 ***
## CAR_TYPEMinivan                -0.71996    0.04292 -16.773  < 2e-16 ***
## `CAR_TYPEPanel Truck`          -0.11877    0.07049  -1.685 0.092023 .
## CAR_TYPEPickup                 -0.17573    0.04604  -3.816 0.000135 ***
## `CAR_TYPESports Car`            0.19944    0.05170   3.858 0.000114 ***
## `URBANICITYHighly Rural/ Rural` -2.47017    0.06130 -40.299  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`         0.02973    0.05941   0.500 0.616793
## HOME_VAL_CAT.Medium             0.01830    0.05763   0.318 0.750820
## HOME_VAL_CAT.High               0.02167    0.07364   0.294 0.768577
## HOME_VAL_CAT.Renters            0.40243    0.05627   7.152 8.55e-13 ***
## REVOKED.No                     -0.88597    0.04925 -17.988  < 2e-16 ***
## PARENT1.No                     -0.71441    0.04877 -14.648  < 2e-16 ***
## MSTATUS.No                      0.34634    0.04272   8.107 5.17e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33055  on 28564  degrees of freedom
## Residual deviance: 25454  on 28533  degrees of freedom
## AIC: 25518
```

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

At this stage, all predictors in our model are statistically significant, and collinearity is minimal. However, because this model was built using transformed variables, interpretation becomes more complex, and applying it to new data requires replicating the exact same transformation steps. This limitation leads us to the development of our next model.

Model 2 Now that we have found the predictors we would like to include in our model, we can try using the non-transformed versions of the variables and recreating the model.

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(original_non_cat, encoded_columns_filtered3, "TARGET_FLAG")])
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -0.210557   0.098695  -2.133 0.032891 *
## KIDSDRIV                         0.235158   0.017080  13.768  < 2e-16 ***
## AGE                             -0.067395   0.018871  -3.571 0.000355 ***
## HOMEKIDS                        -0.001726   0.023160  -0.075 0.940605
## YOJ                              0.033828   0.017102   1.978 0.047924 *
## INCOME                          -0.240753   0.026634  -9.039  < 2e-16 ***
## TRAVTIME                         0.275518   0.015950  17.274  < 2e-16 ***
## BLUEBOOK                        -0.213857   0.019439 -11.001  < 2e-16 ***
## TIF                             -0.212366   0.016068 -13.216  < 2e-16 ***
## OLDCLAIM                         0.195772   0.018521  10.570  < 2e-16 ***
## CLM_FREQ                         0.092852   0.018997   4.888 1.02e-06 ***
## MVR_PTS                          0.112000   0.015697   7.135 9.65e-13 ***
## CAR_AGE                         -0.031830   0.022788  -1.397 0.162482
## EDUCATION.L                     -0.491294   0.081060  -6.061 1.35e-09 ***
## EDUCATION.C                      0.112157   0.040248   2.787 0.005326 **
## `JOBBlue Collar`                 0.146300   0.068558   2.134 0.032845 *
## JOBClerical                      0.193101   0.067855   2.846 0.004430 **
## JOBDoctor                       -0.117233   0.122039  -0.961 0.336745
## JOBLawyer                        0.217822   0.074962   2.906 0.003663 **
## JOBManager                      -0.646056   0.070073  -9.220  < 2e-16 ***
## JOBProfessional                 -0.008196   0.065776  -0.125 0.900840
## JOBStudent                      -0.165872   0.079916  -2.076 0.037932 *
## CAR_USECommercial                0.794341   0.046368  17.131  < 2e-16 ***
## CAR_TYPEMinivan                 -0.708862   0.043006 -16.483  < 2e-16 ***
## `CAR_TYPEPanel Truck`           -0.183828   0.070605  -2.604 0.009224 **
## CAR_TYPEPickup                  -0.185582   0.046133  -4.023 5.75e-05 ***
## `CAR_TYPESports Car`             0.250710   0.051737   4.846 1.26e-06 ***
## `URBANICITYHighly Rural/ Rural` -2.516532   0.061696 -40.789  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`          0.044575   0.059506   0.749 0.453808
## HOME_VAL_CAT.Medium              0.047490   0.057792   0.822 0.411226
## HOME_VAL_CAT.High                0.002938   0.071964   0.041 0.967438
## HOME_VAL_CAT.Renters             0.430932   0.056271   7.658 1.89e-14 ***
## REVOKED.No                      -0.857618   0.045381 -18.898  < 2e-16 ***
## PARENT1.No                      -0.485706   0.058701  -8.274  < 2e-16 ***
## MSTATUS.No                       0.471025   0.045248  10.410  < 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: 33055  on 28564  degrees of freedom
## Residual deviance: 25390  on 28530  degrees of freedom
## AIC: 25460
##
## Number of Fisher Scoring iterations: 5
```

HOMEKIDS does not seem very predictive at all; we remove it:

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(non_cat2, encoded_columns_filtered3, "TARGET_FLAG")])
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -0.212650   0.094613  -2.248 0.024604 *
## KIDSDRIV                         0.234600   0.015351  15.283  < 2e-16 ***
## AGE                             -0.066842   0.017350  -3.853 0.000117 ***
## YOJ                              0.033457   0.016361   2.045 0.040861 *
## INCOME                          -0.240792   0.026629  -9.043  < 2e-16 ***
## TRAVTIME                         0.275536   0.015948  17.277  < 2e-16 ***
## BLUEBOOK                        -0.213855   0.019439 -11.001  < 2e-16 ***
## TIF                             -0.212404   0.016060 -13.225  < 2e-16 ***
## OLDCLAIM                         0.195761   0.018521  10.570  < 2e-16 ***
## CLM_FREQ                         0.092848   0.018997   4.888 1.02e-06 ***
## MVR_PTS                          0.111992   0.015696   7.135 9.68e-13 ***
## CAR_AGE                         -0.031824   0.022788  -1.397 0.162554
## EDUCATION.L                     -0.491147   0.081035  -6.061 1.35e-09 ***
## EDUCATION.C                      0.112105   0.040242   2.786 0.005340 **
## `JOBBlue Collar`                 0.146338   0.068556   2.135 0.032795 *
## JOBClerical                      0.193067   0.067853   2.845 0.004436 **
## JOBDoctor                       -0.117270   0.122039  -0.961 0.336589
## JOBLawyer                        0.217866   0.074960   2.906 0.003656 **
## JOBManager                      -0.645927   0.070052  -9.221  < 2e-16 ***
## JOBProfessional                 -0.008071   0.065755  -0.123 0.902307
## JOBStudent                      -0.166188   0.079802  -2.083 0.037297 *
## CAR_USECommercial                0.794347   0.046368  17.131  < 2e-16 ***
## CAR_TYPEMinivan                 -0.708749   0.042979 -16.490  < 2e-16 ***
## `CAR_TYPEPanel Truck`           -0.183757   0.070599  -2.603 0.009246 **
## CAR_TYPEPickup                  -0.185423   0.046083  -4.024 5.73e-05 ***
## `CAR_TYPESports Car`             0.250652   0.051731   4.845 1.26e-06 ***
## `URBANICITYHighly Rural/ Rural` -2.516565   0.061694 -40.791  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`          0.044460   0.059487   0.747 0.454828
## HOME_VAL_CAT.Medium              0.047579   0.057781   0.823 0.410251
## HOME_VAL_CAT.High                0.002965   0.071963   0.041 0.967130
## HOME_VAL_CAT.Renters             0.430865   0.056263   7.658 1.89e-14 ***
## REVOKED.No                      -0.857529   0.045366 -18.903  < 2e-16 ***
## PARENT1.No                      -0.483778   0.052687  -9.182  < 2e-16 ***
## MSTATUS.No                       0.471788   0.044075  10.704  < 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: 33055  on 28564  degrees of freedom
## Residual deviance: 25390  on 28531  degrees of freedom
## AIC: 25458
##
## Number of Fisher Scoring iterations: 5
```

And we can safely remove CAR_AGE as well:

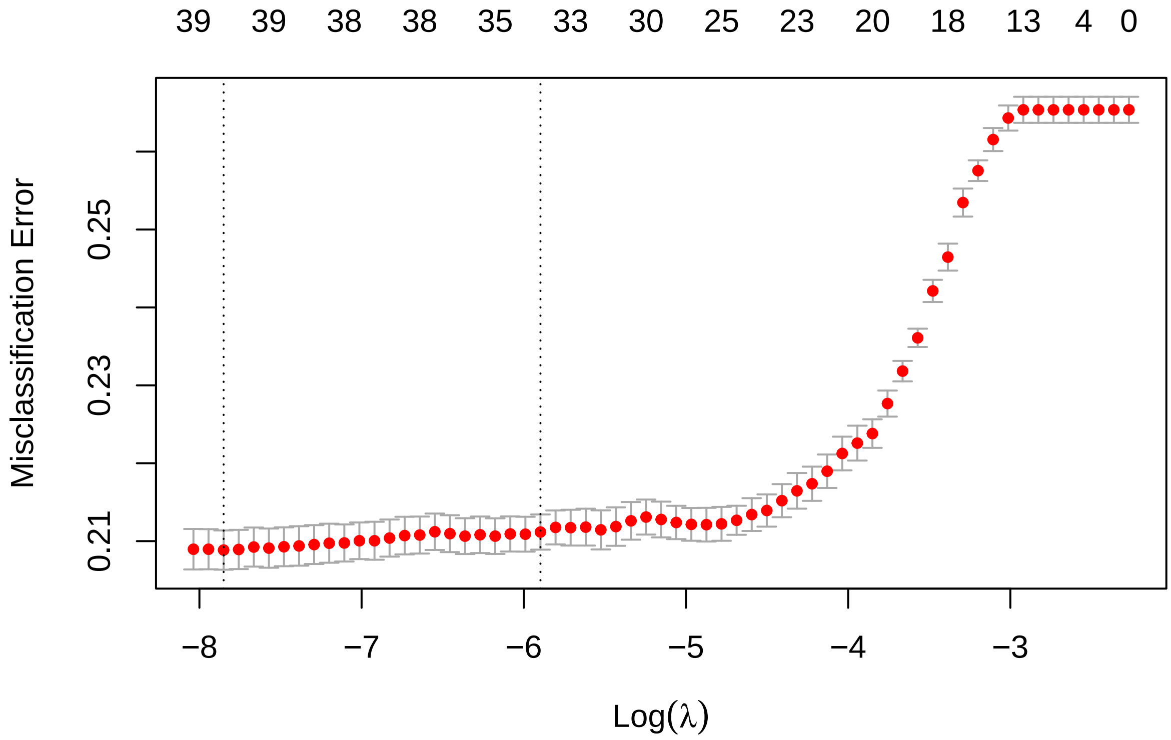
```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = "binomial", data = train_final[,
##     c(non_cat3, encoded_columns_filtered3, "TARGET_FLAG")])
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -0.216392   0.094577  -2.288 0.022138 *
## KIDSDRIV                         0.234694   0.015348  15.291  < 2e-16 ***
## AGE                             -0.066777   0.017349  -3.849 0.000119 ***
## YOJ                              0.033613   0.016360   2.055 0.039924 *
## INCOME                          -0.242831   0.026591  -9.132  < 2e-16 ***
## TRAVTIME                         0.275722   0.015948  17.288  < 2e-16 ***
## BLUEBOOK                        -0.213440   0.019437 -10.981  < 2e-16 ***
## TIF                             -0.212843   0.016054 -13.258  < 2e-16 ***
## OLDCLAIM                         0.196123   0.018524  10.587  < 2e-16 ***
## CLM_FREQ                         0.092711   0.019000   4.880 1.06e-06 ***
## MVR_PTS                          0.111357   0.015690   7.097 1.27e-12 ***
## EDUCATION.L                     -0.544136   0.071659  -7.593 3.12e-14 ***
## EDUCATION.C                      0.126288   0.038937   3.243 0.001181 **
## `JOBBlue Collar`                 0.146452   0.068553   2.136 0.032652 *
## JOBClerical                      0.194275   0.067843   2.864 0.004189 **
## JOBDoctor                       -0.115999   0.122048  -0.950 0.341890
## JOBLawyer                        0.216229   0.074939   2.885 0.003909 **
## JOBManager                      -0.646691   0.070055  -9.231  < 2e-16 ***
## JOBProfessional                 -0.007363   0.065745  -0.112 0.910829
## JOBStudent                      -0.168731   0.079787  -2.115 0.034450 *
## CAR_USECommercial                0.793767   0.046372  17.118  < 2e-16 ***
## CAR_TYPEMinivan                 -0.708979   0.042975 -16.497  < 2e-16 ***
## `CAR_TYPEPanel Truck`           -0.184375   0.070598  -2.612 0.009011 **
## CAR_TYPEPickup                  -0.185678   0.046086  -4.029 5.60e-05 ***
## `CAR_TYPESports Car`             0.250228   0.051733   4.837 1.32e-06 ***
## `URBANICITYHighly Rural/ Rural` -2.516279   0.061684 -40.793  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`          0.045214   0.059480   0.760 0.447163
## HOME_VAL_CAT.Medium              0.050753   0.057736   0.879 0.379373
## HOME_VAL_CAT.High                0.008984   0.071833   0.125 0.900465
## HOME_VAL_CAT.Renters             0.432218   0.056255   7.683 1.55e-14 ***
## REVOKED.No                      -0.857737   0.045367 -18.907  < 2e-16 ***
## PARENT1.No                      -0.484592   0.052684  -9.198  < 2e-16 ***
## MSTATUS.No                       0.471284   0.044071  10.694  < 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: 33055  on 28564  degrees of freedom
## Residual deviance: 25392  on 28532  degrees of freedom
## AIC: 25458
##
## Number of Fisher Scoring iterations: 5
```

On initial evaluation, the model using non-transformed data performs comparably to the one using transformed variables. We'll revisit this comparison in more detail later.

Up to this point, we've built two models with largely overlapping predictors, relying heavily on manual judgment to decide which variables to include. Moving forward, we'll implement an **automated variable selection** technique to guide this process more systematically.

Model 3 To further streamline the variable selection process, we'll now turn to an automated approach using Lasso regression. This technique not only performs feature selection by shrinking some coefficients to zero but also helps prevent overfitting, making it well-suited for refining our model.



```
## 46 x 1 sparse Matrix of class "dgCMatrix"
##                                         s1
## (Intercept)                  -4.038948e-01
## BLUEBOOK_transformed         -2.124878e-01
## INCOME_transformed           -2.338356e-01
## MVR_PTS_transformed           2.402444e-01
## OLDCLAIM_transformed         -1.130455e-01
## TIF_transformed              -2.291182e-01
## TRAVTIME_transformed          2.592883e-01
## YOJ_transformed              -6.752574e-02
## CLM_FREQ_transformed           3.571027e-01
## CAR_AGE_transformed           -5.649233e-02
## SEX.F                          2.374819e-02
## SEX.M                         -7.408787e-13
## EDUCATION.L                   -2.285970e-01
## EDUCATION.Q                    2.385803e-01
## EDUCATION.C                    1.415082e-01
## JOBBlue Collar                 4.071463e-01
## JOBClerical                    4.556895e-01
## JOBDoctor                     -2.055641e-01
## JOBHome Maker                  1.895722e-01
## JOBLawyer                      3.128729e-01
## JOBManager                    -4.163305e-01
## JOBProfessional                2.060128e-01
## JOBStudent                     .
## CAR_USECommercial              7.734120e-01
## CAR_USEPrivate                -3.580514e-12
## CAR_TYPEMinivan               -5.676041e-01
## CAR_TYPEPanel Truck            .
## CAR_TYPEPickup                -4.336778e-02
## CAR_TYPESports Car             3.179114e-01
## CAR_TYPESUV                    1.396060e-01
## CAR_TYPEVan                    8.360436e-02
## URBANICITYHighly Rural/ Rural -2.453003e+00
## URBANICITYHighly Urban/ Urban  1.994636e-12
## HOME_VAL_CAT.Very Low         -7.519304e-03
## HOME_VAL_CAT.Low               .
## HOME_VAL_CAT.Medium            1.218348e-02
## HOME_VAL_CAT.High             -1.407696e-03
## HOME_VAL_CAT.Renters           3.752189e-01
## RED_CAR.No                     .
## RED_CAR.Yes                    .
## REVOKED.No                    -8.674595e-01
## REVOKED.Yes                    9.058198e-13
## PARENT1.No                    -7.082687e-01
## PARENT1.Yes                    .
## MSTATUS.No                     3.506286e-01
## MSTATUS.Yes                   -3.947048e-12
```

```
## [1] "Best Lambda:  0.000389427129306494"
```

The results from the model yield some notable insights. For example, the variable CAR_USECommercial shows a strong positive association with crash likelihood—an intuitive finding, as commercial drivers may be less cautious when the vehicle isn't personally owned, and they may also spend more time on the road. Similarly, URBANICITYHighly Rural/Rural has a strong negative correlation with crashes, which makes sense given the lower traffic density in rural areas, reducing the chance of collisions. As expected, REVOKED.No (i.e., not having a revoked license) is linked to a reduced probability of being involved in a crash.

Model Comparison We now run our three models on the test data and compare the results.

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
```

```
## Warning in roc.default(test_final$TARGET_FLAG, lasso_model_predictions):
## Deprecated use a matrix as predictor. Unexpected results may be produced,
## please pass a numeric vector.
```

```
## Setting direction: controls < cases
```

Table 20: Comparison of Logistic Models

Model	RMSE	AIC	Accuracy	Precision	Recall	F1_Score	ROC_AUC
Model_Transformed	0.4645962	25517.99	0.7841503	0.3918367	0.6390169	0.4857921	0.7945324
Model_Untransformed	0.4625695	25457.75	0.7860294	0.4156986	0.6359270	0.5027530	0.7984920
Lasso Model	0.4626579	NA	0.7859477	0.3974882	0.6436197	0.4914596	0.7932680

It’s important to note that we haven’t reported the AIC for the Lasso model, as its use of regularization complicates the assumptions underlying AIC calculations. That said, one of the most striking observations is how similar the performance metrics are across all three models.

The untransformed model shows the lowest RMSE, suggesting it yields the smallest prediction error. It also has the lowest AIC, meaning it performs best in terms of balancing model fit and complexity. When it comes to classification metrics—accuracy, precision, and recall—the untransformed model holds a slight lead in accuracy and precision, while the Lasso model edges out in recall. This could be significant in contexts where identifying true positives is especially important.

Interestingly, the second model (non-transformed, manually refined) leads in both F1 score and ROC-AUC, indicating strong overall performance and good class discrimination across various thresholds.

While the differences among the models are relatively small, the second model stands out for its consistency across nearly all metrics. Although it doesn’t have the highest recall, its shortfall compared to the Lasso model is minimal (only 0.0072004), and thus not particularly concerning. Most importantly, the second model is **highly interpretable**, making it the preferred choice—especially in scenarios where understanding the model’s decision-making process is critical.

Therefore, we select the second model—“final_non-transformed_model”—as our final choice.

Multiple Linear Regression

Up to this point, we’ve focused on modeling the likelihood of a car crash and have chosen the most effective model for that task. We now shift our attention to predicting the payout amount in cases where a crash has occurred. To start, we’ll manually select predictors based on their correlation with the response variable and domain knowledge. We’ll then refine this initial model using a stepwise regression approach. Finally, we’ll compare the performance of both models—manual and stepwise—using key evaluation metrics such as RMSE and R-squared to determine which model more accurately predicts TARGET_AMT.

Model 1 When selecting variables, it’s reasonable to assume that predictors which were significant in determining whether a crash occurred may also be relevant in estimating the payout following a crash. We’ll now put that intuition to the test by evaluating their effectiveness in this new context.

```r
#have to add back ticks or it just won't work
predictors <- c("KIDSDRIV", "AGE", "YOJ", "INCOME", "TRAVTIME", "BLUEBOOK", "TIF",
                "OLDCLAIM", "CLM_FREQ", "MVR_PTS", "EDUCATION.L", "EDUCATION.C",
                "`JOBBlue Collar`", "JOBClerical", "JOBDoctor", "JOBLawyer",
                "JOBManager", "JOBProfessional", "JOBStudent", "CAR_USECommercial",
                "CAR_TYPEMinivan", "`CAR_TYPEPanel Truck`", "CAR_TYPEPickup",
                "`CAR_TYPESports Car`", "`URBANICITYHighly Rural/ Rural`",
                "`HOME_VAL_CAT.Very Low`", "HOME_VAL_CAT.Medium", "HOME_VAL_CAT.High",
                "HOME_VAL_CAT.Renters", "REVOKED.No", "PARENT1.No", "MSTATUS.No")

formula <- as.formula(paste("TARGET_AMT ~", paste(predictors, collapse=" + ")))
mlr1 <- lm(formula, data = train_final)
summary(mlr1)
```

```
## 
## Call:
## lm(formula = formula, data = train_final)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -5651  -1778   -785    384 103969 
## 
## Coefficients:
```

```
##                                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                      2822.36     177.38  15.911  < 2e-16 ***
## KIDSDRIV                          216.94      29.01   7.478 7.76e-14 ***
## AGE                                34.78      31.20   1.115 0.264961    
## YOJ                                12.29      29.09   0.423 0.672613    
## INCOME                           -162.26      46.46  -3.493 0.000479 ***
## TRAVTIME                          234.75      28.27   8.304  < 2e-16 ***
## BLUEBOOK                           25.16      33.86   0.743 0.457381    
## TIF                              -215.05      27.99  -7.683 1.61e-14 ***
## OLDCLAIM                           43.86      37.78   1.161 0.245760    
## CLM_FREQ                          136.03      38.11   3.569 0.000359 ***
## MVR_PTS                           208.20      30.58   6.809 1.00e-11 ***
## EDUCATION.L                      -596.20     126.66  -4.707 2.52e-06 ***
## EDUCATION.C                        43.70      70.61   0.619 0.536023    
## `JOBBlue Collar`                  -43.95     125.19  -0.351 0.725506    
## JOBClerical                        40.64     122.43   0.332 0.739924    
## JOBDoctor                        -383.95     196.18  -1.957 0.050338 .  
## JOBLawyer                         100.99     131.85   0.766 0.443695    
## JOBManager                       -936.55     119.71  -7.823 5.32e-15 ***
## JOBProfessional                    99.27     117.84   0.842 0.399595    
## JOBStudent                       -230.33     147.49  -1.562 0.118370    
## CAR_USECommercial                 952.74      85.35  11.163  < 2e-16 ***
## CAR_TYPEMinivan                  -578.39      72.46  -7.982 1.49e-15 ***
## `CAR_TYPEPanel Truck`            -508.79     132.32  -3.845 0.000121 ***
## CAR_TYPEPickup                   -209.38      85.32  -2.454 0.014135 *  
## `CAR_TYPESports Car`              114.91      97.39   1.180 0.238049    
## `URBANICITYHighly Rural/ Rural` -1880.31      76.20 -24.677  < 2e-16 ***
## `HOME_VAL_CAT.Very Low`          -262.88     107.56  -2.444 0.014529 *  
## HOME_VAL_CAT.Medium              -243.67     102.31  -2.382 0.017240 *  
## HOME_VAL_CAT.High                -219.18     123.36  -1.777 0.075626 .  
## HOME_VAL_CAT.Renters              110.51     103.70   1.066 0.286600    
## REVOKED.No                       -312.04      89.60  -3.483 0.000497 ***
## PARENT1.No                       -837.60     101.29  -8.269  < 2e-16 ***
## MSTATUS.No                        407.44      79.40   5.132 2.89e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4717 on 28532 degrees of freedom
## Multiple R-squared:  0.06821,    Adjusted R-squared:  0.06717 
## F-statistic: 65.27 on 32 and 28532 DF,  p-value: < 2.2e-16
```

While the overall model is statistically significant, the Adjusted R-squared remains quite low, indicating limited explanatory power. However, it's important to emphasize that our primary interest lies in predicting payouts only when a crash has actually occurred—that is, when **TARGET_FLAG =1**. In all other cases, the payout is simply zero. With this in mind, let's assess model performance by multiplying the predicted values by the TARGET_FLAG, effectively filtering predictions to only those instances where a payout would apply.

```
## 
## Call:
## lm(formula = formula, data = train_final)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -9665   -411    -49    183  98279 
```

```
##
## Coefficients:
##                                                Estimate Std. Error t value
## (Intercept)                                   1.991e+03  1.580e+02  12.606
## KIDSDRIV:TARGET_FLAG0                        -2.883e+01  3.257e+01  -0.885
## KIDSDRIV:TARGET_FLAG1                         1.429e+02  4.083e+01   3.500
## TARGET_FLAG0:AGE                              5.163e+01  3.277e+01   1.575
## TARGET_FLAG1:AGE                              1.737e+02  4.867e+01   3.568
## TARGET_FLAG0:YOJ                             -2.052e+01  2.910e+01  -0.705
## TARGET_FLAG1:YOJ                              1.156e+02  5.202e+01   2.221
## TARGET_FLAG0:INCOME                           8.715e+01  4.639e+01   1.879
## TARGET_FLAG1:INCOME                          -1.829e+02  8.229e+01  -2.222
## TARGET_FLAG0:TRAVTIME                         1.417e+01  2.866e+01   0.494
## TARGET_FLAG1:TRAVTIME                         6.869e+01  4.935e+01   1.392
## TARGET_FLAG0:BLUEBOOK                         3.628e-02  3.402e+01   0.001
## TARGET_FLAG1:BLUEBOOK                         7.575e+02  6.030e+01  12.562
## TARGET_FLAG0:TIF                             -2.251e+00  2.838e+01  -0.079
## TARGET_FLAG1:TIF                             -2.107e+02  4.863e+01  -4.333
## TARGET_FLAG0:OLDCLAIM                         3.468e+01  4.245e+01   0.817
## TARGET_FLAG1:OLDCLAIM                        -6.101e+02  5.324e+01 -11.459
## TARGET_FLAG0:CLM_FREQ                        -2.454e+01  4.234e+01  -0.580
## TARGET_FLAG1:CLM_FREQ                         3.231e+02  5.461e+01   5.916
## TARGET_FLAG0:MVR_PTS                         -1.104e+01  3.328e+01  -0.332
## TARGET_FLAG1:MVR_PTS                          2.324e+02  4.519e+01   5.142
## TARGET_FLAG0:EDUCATION.L                     -1.166e+02  1.290e+02  -0.903
## TARGET_FLAG1:EDUCATION.L                     -4.027e+01  2.155e+02  -0.187
## TARGET_FLAG0:EDUCATION.C                      1.067e+00  7.248e+01   0.015
## TARGET_FLAG1:EDUCATION.C                     -3.404e+02  1.183e+02  -2.877
## TARGET_FLAG0:`JOBBlue Collar`                -4.235e+02  1.267e+02  -3.343
## TARGET_FLAG1:`JOBBlue Collar`                 4.258e+02  1.991e+02   2.138
## TARGET_FLAG0:JOBClerical                     -5.008e+02  1.229e+02  -4.076
## TARGET_FLAG1:JOBClerical                      8.224e+02  1.907e+02   4.313
## TARGET_FLAG0:JOBDoctor                       -3.709e+02  1.876e+02  -1.977
## TARGET_FLAG1:JOBDoctor                       -8.727e+02  4.288e+02  -2.035
## TARGET_FLAG0:JOBLawyer                       -4.400e+02  1.288e+02  -3.417
## TARGET_FLAG1:JOBLawyer                        1.144e+03  2.369e+02   4.830
## TARGET_FLAG0:JOBManager                      -4.596e+02  1.151e+02  -3.993
## TARGET_FLAG1:JOBManager                      -1.004e+03  2.295e+02  -4.377
## TARGET_FLAG0:JOBProfessional                 -4.557e+02  1.160e+02  -3.927
## TARGET_FLAG1:JOBProfessional                  1.704e+03  1.961e+02   8.689
## TARGET_FLAG0:JOBStudent                      -3.185e+02  1.555e+02  -2.049
## TARGET_FLAG1:JOBStudent                       5.611e+02  2.263e+02   2.479
## TARGET_FLAG0:CAR_USECommercial               -1.013e+02  8.890e+01  -1.139
## TARGET_FLAG1:CAR_USECommercial                1.242e+03  1.393e+02   8.921
## TARGET_FLAG0:CAR_TYPEMinivan                 -8.620e+01  7.170e+01  -1.202
## TARGET_FLAG1:CAR_TYPEMinivan                  2.625e+02  1.387e+02   1.892
## TARGET_FLAG0:`CAR_TYPEPanel Truck`           -1.115e+02  1.365e+02  -0.816
## TARGET_FLAG1:`CAR_TYPEPanel Truck`           -7.833e+02  2.179e+02  -3.595
## TARGET_FLAG0:CAR_TYPEPickup                  -9.455e+01  8.852e+01  -1.068
## TARGET_FLAG1:CAR_TYPEPickup                  -6.070e+01  1.381e+02  -0.440
## TARGET_FLAG0:`CAR_TYPESports Car`            -1.264e+02  1.034e+02  -1.223
## TARGET_FLAG1:`CAR_TYPESports Car`             8.150e+01  1.477e+02   0.552
## TARGET_FLAG0:`URBANICITYHighly Rural/ Rural` -7.518e+01  7.474e+01  -1.006
## TARGET_FLAG1:`URBANICITYHighly Rural/ Rural` -7.315e+02  2.204e+02  -3.319
## TARGET_FLAG0:`HOME_VAL_CAT.Very Low`         -3.587e+02  1.071e+02  -3.350
## TARGET_FLAG1:`HOME_VAL_CAT.Very Low`         -2.385e+02  1.768e+02  -1.349
## TARGET_FLAG0:HOME_VAL_CAT.Medium             -3.454e+02  1.006e+02  -3.434
## TARGET_FLAG1:HOME_VAL_CAT.Medium             -5.665e+01  1.773e+02  -0.320
## TARGET_FLAG0:HOME_VAL_CAT.High               -4.521e+02  1.199e+02  -3.772
## TARGET_FLAG1:HOME_VAL_CAT.High                6.290e+02  2.312e+02   2.720
## TARGET_FLAG0:HOME_VAL_CAT.Renters            -3.329e+02  1.049e+02  -3.174
## TARGET_FLAG1:HOME_VAL_CAT.Renters            -2.614e+02  1.662e+02  -1.573
## TARGET_FLAG0:REVOKED.No                      -4.890e+02  9.723e+01  -5.030
## TARGET_FLAG1:REVOKED.No                       2.421e+03  1.224e+02  19.777
## TARGET_FLAG0:PARENT1.No                      -6.918e+02  1.067e+02  -6.485
## TARGET_FLAG1:PARENT1.No                       4.831e+02  1.321e+02   3.656
## TARGET_FLAG0:MSTATUS.No                      -2.569e+02  7.898e+01  -3.253
## TARGET_FLAG1:MSTATUS.No                       1.135e+03  1.337e+02   8.491
##                                              Pr(>|t|)
## (Intercept)                                   < 2e-16 ***
## KIDSDRIV:TARGET_FLAG0                        0.376081
## KIDSDRIV:TARGET_FLAG1                        0.000466 ***
## TARGET_FLAG0:AGE                             0.115157
## TARGET_FLAG1:AGE                             0.000360 ***
## TARGET_FLAG0:YOJ                             0.480757
## TARGET_FLAG1:YOJ                             0.026326 *
## TARGET_FLAG0:INCOME                          0.060306 .
## TARGET_FLAG1:INCOME                          0.026274 *
## TARGET_FLAG0:TRAVTIME                        0.621062
## TARGET_FLAG1:TRAVTIME                        0.163988
## TARGET_FLAG0:BLUEBOOK                        0.999149
## TARGET_FLAG1:BLUEBOOK                         < 2e-16 ***
## TARGET_FLAG0:TIF                             0.936772
## TARGET_FLAG1:TIF                             1.48e-05 ***
## TARGET_FLAG0:OLDCLAIM                        0.413887
## TARGET_FLAG1:OLDCLAIM                         < 2e-16 ***
## TARGET_FLAG0:CLM_FREQ                        0.562203
## TARGET_FLAG1:CLM_FREQ                        3.34e-09 ***
## TARGET_FLAG0:MVR_PTS                         0.740029
## TARGET_FLAG1:MVR_PTS                         2.74e-07 ***
## TARGET_FLAG0:EDUCATION.L                     0.366274
## TARGET_FLAG1:EDUCATION.L                     0.851784
## TARGET_FLAG0:EDUCATION.C                     0.988259
## TARGET_FLAG1:EDUCATION.C                     0.004023 **
## TARGET_FLAG0:`JOBBlue Collar`                0.000830 ***
## TARGET_FLAG1:`JOBBlue Collar`                0.032529 *
## TARGET_FLAG0:JOBClerical                     4.59e-05 ***
## TARGET_FLAG1:JOBClerical                     1.62e-05 ***
## TARGET_FLAG0:JOBDoctor                       0.048078 *
## TARGET_FLAG1:JOBDoctor                       0.041853 *
## TARGET_FLAG0:JOBLawyer                       0.000634 ***
## TARGET_FLAG1:JOBLawyer                       1.37e-06 ***
## TARGET_FLAG0:JOBManager                      6.54e-05 ***
## TARGET_FLAG1:JOBManager                      1.21e-05 ***
## TARGET_FLAG0:JOBProfessional                 8.61e-05 ***
## TARGET_FLAG1:JOBProfessional                  < 2e-16 ***
## TARGET_FLAG0:JOBStudent                      0.040512 *
## TARGET_FLAG1:JOBStudent                      0.013175 *
## TARGET_FLAG0:CAR_USECommercial                0.254625
## TARGET_FLAG1:CAR_USECommercial                 < 2e-16 ***
## TARGET_FLAG0:CAR_TYPEMinivan                   0.229279
## TARGET_FLAG1:CAR_TYPEMinivan                   0.058437 .
## TARGET_FLAG0:`CAR_TYPEPanel Truck`             0.414271
## TARGET_FLAG1:`CAR_TYPEPanel Truck`             0.000325 ***
## TARGET_FLAG0:CAR_TYPEPickup                    0.285454
## TARGET_FLAG1:CAR_TYPEPickup                    0.660258
## TARGET_FLAG0:`CAR_TYPESports Car`              0.221391
## TARGET_FLAG1:`CAR_TYPESports Car`              0.581042
## TARGET_FLAG0:`URBANICITYHighly Rural/ Rural` 0.314522
## TARGET_FLAG1:`URBANICITYHighly Rural/ Rural` 0.000903 ***
## TARGET_FLAG0:`HOME_VAL_CAT.Very Low`           0.000809 ***
## TARGET_FLAG1:`HOME_VAL_CAT.Very Low`           0.177432
## TARGET_FLAG0:HOME_VAL_CAT.Medium               0.000596 ***
## TARGET_FLAG1:HOME_VAL_CAT.Medium               0.749319
## TARGET_FLAG0:HOME_VAL_CAT.High                 0.000162 ***
## TARGET_FLAG1:HOME_VAL_CAT.High                 0.006530 **
## TARGET_FLAG0:HOME_VAL_CAT.Renters              0.001505 **
## TARGET_FLAG1:HOME_VAL_CAT.Renters              0.115787
## TARGET_FLAG0:REVOKED.No                        4.95e-07 ***
## TARGET_FLAG1:REVOKED.No                         < 2e-16 ***
## TARGET_FLAG0:PARENT1.No                        9.04e-11 ***
## TARGET_FLAG1:PARENT1.No                        0.000256 ***
## TARGET_FLAG0:MSTATUS.No                        0.001144 **
## TARGET_FLAG1:MSTATUS.No                         < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4108 on 28500 degrees of freedom
## Multiple R-squared:  0.2944, Adjusted R-squared:  0.2928
## F-statistic: 185.8 on 64 and 28500 DF,  p-value: < 2.2e-16
```

This adjusted model shows a notable improvement and aligns well with intuition—it ensures that only cases predicted as crashes yield a positive **TARGET_AMT**. This logic adds a layer of practical accuracy to our predictions. That said, we'll now explore whether we can retain this intuitive structure while enhancing the model's performance by refining our set of predictors.

Model 2 Once again, we plan to incorporate TARGET_FLAG into our modeling strategy. However, this time we'll apply an automated stepwise regression approach. This iterative process evaluates predictors based on their statistical significance, systematically adding or removing variables to arrive at the most optimal model fit.

From the stepwise process, we learn that our best model is:

TARGET_AMT ~ AGE:TARGET_FLAG + TARGET_FLAG:INCOME + TARGET_FLAG:TIF + TARGET_FLAG:YOJ + TARGET_FLAG:HOMEKIDS + TARGET_FLAG:CLM_FREQ + TARGET_FLAG:CAR_AGE + TARGET_FLAG:OLDCLAIM + TARGET_FLAG:BLUEBOOK_transformed + TARGET_FLAG:INCOME_transformed + TARGET_FLAG:MVR_PTS_transformed + TARGET_FLAG:TIF_transformed + TARGET_FLAG:YOJ_transformed + TARGET_FLAG:CLM_FREQ_transformed + TARGET_FLAG:CAR_AGE_transformed + TARGET_FLAG:SEX.F + TARGET_FLAG:SEX.M + TARGET_FLAG:EDUCATION.L + TARGET_FLAG:EDUCATION.Q + TARGET_FLAG:`EDUCATION^4` + TARGET_FLAG:`JOBBlue Collar` + TARGET_FLAG:JOBClerical + TARGET_FLAG:JOBDoctor + TARGET_FLAG:`JOBHome Maker` + TARGET_FLAG:JOBLawyer + TARGET_FLAG:JOBManager + TARGET_FLAG:JOBStudent + TARGET_FLAG:CAR_USECommercial + TARGET_FLAG:`CAR_TYPEPanel`

Truck + TARGET_FLAG:CAR_TYPEPickup + TARGET_FLAG:CAR_TYPESports Car + TARGET_FLAG:CAR_TYPESUV + TARGET_FLAG:URBANICITYHighly Rural/ Rural + TARGET_FLAG:HOME_VAL_CAT.Low + TARGET_FLAG:HOME_VAL_CAT.Medium + TARGET_FLAG:HOME_VAL_CAT.High + TARGET_FLAG:REVOKED.No + TARGET_FLAG:PARENT1.No + TARGET_FLAG:MSTATUS.No

Now, this seems quite involved. But it's worth repeating–the constant presence of TARGET_FLAG is simply to guarantee that only observations with a TARGET_FLAG of 1 will have a non-zero value for TARGET_AMT. Let us now print the summary:

```
##
## Call:
## lm(formula = best_stepwise_formula, data = train_final)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9849      0      0      0  98522
##
## Coefficients: (1 not defined because of singularities)
##                                               Estimate Std. Error t value
## (Intercept)                                  4.146e+03  3.983e+02  10.409
## AGE:TARGET_FLAG0                            -4.906e-11  3.535e+01   0.000
## AGE:TARGET_FLAG1                             2.880e+02  5.162e+01   5.578
## TARGET_FLAG0:INCOME                          2.184e-12  5.750e+01   0.000
## TARGET_FLAG1:INCOME                          5.720e+02  1.250e+02   4.576
## TARGET_FLAG0:TIF                            -8.206e-12  6.672e+01   0.000
## TARGET_FLAG1:TIF                            -6.339e+02  1.329e+02  -4.771
## TARGET_FLAG0:YOJ                             9.931e-12  3.499e+01   0.000
## TARGET_FLAG1:YOJ                            -1.588e+02  5.986e+01  -2.654
## TARGET_FLAG0:HOMEKIDS                       -1.058e-11  3.866e+01   0.000
## TARGET_FLAG1:HOMEKIDS                        3.008e+02  6.204e+01   4.848
## TARGET_FLAG0:CLM_FREQ                        3.974e-13  1.389e+02   0.000
## TARGET_FLAG1:CLM_FREQ                       -6.836e+02  1.736e+02  -3.939
## TARGET_FLAG0:CAR_AGE                        -1.285e-11  2.026e+02   0.000
## TARGET_FLAG1:CAR_AGE                        -3.413e+03  3.566e+02  -9.570
## TARGET_FLAG0:OLDCLAIM                       -3.210e-12  4.604e+01   0.000
## TARGET_FLAG1:OLDCLAIM                       -6.400e+02  5.971e+01 -10.719
## TARGET_FLAG0:BLUEBOOK_transformed            8.370e-13  3.845e+01   0.000
## TARGET_FLAG1:BLUEBOOK_transformed            9.650e+02  6.434e+01  14.999
## TARGET_FLAG0:INCOME_transformed              1.615e-11  7.222e+01   0.000
## TARGET_FLAG1:INCOME_transformed             -9.992e+02  1.503e+02  -6.646
## TARGET_FLAG0:MVR_PTS_transformed             2.120e-12  3.569e+01   0.000
## TARGET_FLAG1:MVR_PTS_transformed             2.376e+02  5.061e+01   4.696
## TARGET_FLAG0:TIF_transformed                 4.705e-12  6.719e+01   0.000
## TARGET_FLAG1:TIF_transformed                 4.436e+02  1.310e+02   3.386
## TARGET_FLAG0:YOJ_transformed                -1.818e-11  4.379e+01   0.000
## TARGET_FLAG1:YOJ_transformed                 2.356e+02  6.599e+01   3.570
## TARGET_FLAG0:CLM_FREQ_transformed           -6.965e-12  2.115e+02   0.000
## TARGET_FLAG1:CLM_FREQ_transformed            1.544e+03  2.761e+02   5.591
## TARGET_FLAG0:CAR_AGE_transformed             8.029e-12  1.993e+02   0.000
## TARGET_FLAG1:CAR_AGE_transformed             3.132e+03  3.321e+02   9.429
## TARGET_FLAG0:SEX.F                          -4.146e+03  4.792e+02  -8.652
## TARGET_FLAG1:SEX.F                          -1.107e+03  1.617e+02  -6.850
## TARGET_FLAG0:SEX.M                          -4.146e+03  4.758e+02  -8.714
## TARGET_FLAG1:SEX.M                                  NA         NA      NA
## TARGET_FLAG0:EDUCATION.L                    -4.925e-11  1.465e+02   0.000
## TARGET_FLAG1:EDUCATION.L                     1.416e+03  2.660e+02   5.324
## TARGET_FLAG0:EDUCATION.Q                    -2.748e-11  8.261e+01   0.000
## TARGET_FLAG1:EDUCATION.Q                     1.708e+03  1.430e+02  11.949
## TARGET_FLAG0:`EDUCATION^4`                  -2.494e-12  6.219e+01   0.000
## TARGET_FLAG1:`EDUCATION^4`                  -4.354e+02  1.037e+02  -4.200
## TARGET_FLAG0:`JOBBlue Collar`               -5.367e-13  1.133e+02   0.000
## TARGET_FLAG1:`JOBBlue Collar`               -1.584e+03  1.808e+02  -8.758
## TARGET_FLAG0:JOBClerical                     1.114e-11  1.201e+02   0.000
## TARGET_FLAG1:JOBClerical                    -1.648e+03  1.989e+02  -8.285
## TARGET_FLAG0:JOBDoctor                       6.906e-11  1.893e+02   0.000
## TARGET_FLAG1:JOBDoctor                      -3.873e+03  4.419e+02  -8.764
## TARGET_FLAG0:`JOBHome Maker`                 2.618e-11  1.536e+02   0.000
## TARGET_FLAG1:`JOBHome Maker`                -1.711e+03  2.785e+02  -6.145
## TARGET_FLAG0:JOBLawyer                       4.124e-11  1.194e+02   0.000
## TARGET_FLAG1:JOBLawyer                      -1.078e+03  2.444e+02  -4.412
## TARGET_FLAG0:JOBManager                      2.289e-11  9.927e+01   0.000
## TARGET_FLAG1:JOBManager                     -2.738e+03  2.216e+02 -12.355
## TARGET_FLAG0:JOBStudent                      1.348e-11  1.608e+02   0.000
## TARGET_FLAG1:JOBStudent                     -1.449e+03  2.568e+02  -5.642
## TARGET_FLAG0:CAR_USECommercial               1.611e-11  8.832e+01   0.000
## TARGET_FLAG1:CAR_USECommercial               7.948e+02  1.418e+02   5.605
## TARGET_FLAG0:`CAR_TYPEPanel Truck`           1.581e-11  1.367e+02   0.000
## TARGET_FLAG1:`CAR_TYPEPanel Truck`          -1.691e+03  2.226e+02  -7.597
## TARGET_FLAG0:CAR_TYPEPickup                  9.718e-12  8.912e+01   0.000
## TARGET_FLAG1:CAR_TYPEPickup                 -3.049e+02  1.501e+02  -2.031
## TARGET_FLAG0:`CAR_TYPESports Car`            7.493e-12  1.221e+02   0.000
## TARGET_FLAG1:`CAR_TYPESports Car`            5.474e+02  2.119e+02   2.583
## TARGET_FLAG0:CAR_TYPESUV                     6.658e-12  9.600e+01   0.000
## TARGET_FLAG1:CAR_TYPESUV                     7.021e+02  1.846e+02   3.803
## TARGET_FLAG0:`URBANICITYHighly Rural/ Rural` -4.324e-12  7.441e+01   0.000
## TARGET_FLAG1:`URBANICITYHighly Rural/ Rural` -7.258e+02  2.170e+02  -3.344
## TARGET_FLAG0:HOME_VAL_CAT.Low                7.274e-12  9.211e+01   0.000
## TARGET_FLAG1:HOME_VAL_CAT.Low                1.309e+03  1.587e+02   8.249
## TARGET_FLAG0:HOME_VAL_CAT.Medium             1.054e-11  9.030e+01   0.000
## TARGET_FLAG1:HOME_VAL_CAT.Medium             4.499e+02  1.588e+02   2.834
## TARGET_FLAG0:HOME_VAL_CAT.High               1.394e-11  1.105e+02   0.000
## TARGET_FLAG1:HOME_VAL_CAT.High               7.079e+02  2.187e+02   3.238
## TARGET_FLAG0:REVOKED.No                     -5.681e-12  9.996e+01   0.000
## TARGET_FLAG1:REVOKED.No                      1.892e+03  1.305e+02  14.502
## TARGET_FLAG0:PARENT1.No                     -1.414e-11  1.219e+02   0.000
## TARGET_FLAG1:PARENT1.No                     -4.087e+02  1.663e+02  -2.457
## TARGET_FLAG0:MSTATUS.No                     -1.054e-11  7.713e+01   0.000
## TARGET_FLAG1:MSTATUS.No                      7.675e+02  1.362e+02   5.634
##                                              Pr(>|t|)    
## (Intercept)                                   < 2e-16 ***
## AGE:TARGET_FLAG0                             1.000000    
## AGE:TARGET_FLAG1                             2.45e-08 ***
## TARGET_FLAG0:INCOME                          1.000000    
## TARGET_FLAG1:INCOME                          4.75e-06 ***
## TARGET_FLAG0:TIF                             1.000000    
## TARGET_FLAG1:TIF                             1.84e-06 ***
## TARGET_FLAG0:YOJ                             1.000000    
## TARGET_FLAG1:YOJ                             0.007964 ** 
## TARGET_FLAG0:HOMEKIDS                        1.000000    
## TARGET_FLAG1:HOMEKIDS                        1.25e-06 ***
## TARGET_FLAG0:CLM_FREQ                        1.000000    
## TARGET_FLAG1:CLM_FREQ                        8.22e-05 ***
## TARGET_FLAG0:CAR_AGE                         1.000000    
## TARGET_FLAG1:CAR_AGE                          < 2e-16 ***
## TARGET_FLAG0:OLDCLAIM                        1.000000    
## TARGET_FLAG1:OLDCLAIM                         < 2e-16 ***
## TARGET_FLAG0:BLUEBOOK_transformed            1.000000    
## TARGET_FLAG1:BLUEBOOK_transformed             < 2e-16 ***
## TARGET_FLAG0:INCOME_transformed              1.000000    
## TARGET_FLAG1:INCOME_transformed              3.06e-11 ***
## TARGET_FLAG0:MVR_PTS_transformed             1.000000    
## TARGET_FLAG1:MVR_PTS_transformed             2.67e-06 ***
## TARGET_FLAG0:TIF_transformed                 1.000000    
## TARGET_FLAG1:TIF_transformed                 0.000709 ***
## TARGET_FLAG0:YOJ_transformed                 1.000000    
## TARGET_FLAG1:YOJ_transformed                 0.000357 ***
## TARGET_FLAG0:CLM_FREQ_transformed            1.000000    
## TARGET_FLAG1:CLM_FREQ_transformed            2.28e-08 ***
## TARGET_FLAG0:CAR_AGE_transformed             1.000000    
## TARGET_FLAG1:CAR_AGE_transformed              < 2e-16 ***
## TARGET_FLAG0:SEX.F                            < 2e-16 ***
## TARGET_FLAG1:SEX.F                           7.52e-12 ***
## TARGET_FLAG0:SEX.M                            < 2e-16 ***
## TARGET_FLAG1:SEX.M                                 NA    
## TARGET_FLAG0:EDUCATION.L                     1.000000    
## TARGET_FLAG1:EDUCATION.L                     1.02e-07 ***
## TARGET_FLAG0:EDUCATION.Q                     1.000000    
## TARGET_FLAG1:EDUCATION.Q                      < 2e-16 ***
## TARGET_FLAG0:`EDUCATION^4`                   1.000000    
## TARGET_FLAG1:`EDUCATION^4`                   2.68e-05 ***
## TARGET_FLAG0:`JOBBlue Collar`                1.000000    
## TARGET_FLAG1:`JOBBlue Collar`                 < 2e-16 ***
## TARGET_FLAG0:JOBClerical                     1.000000    
## TARGET_FLAG1:JOBClerical                      < 2e-16 ***
## TARGET_FLAG0:JOBDoctor                       1.000000    
## TARGET_FLAG1:JOBDoctor                        < 2e-16 ***
## TARGET_FLAG0:`JOBHome Maker`                 1.000000    
## TARGET_FLAG1:`JOBHome Maker`                 8.10e-10 ***
## TARGET_FLAG0:JOBLawyer                       1.000000    
## TARGET_FLAG1:JOBLawyer                       1.03e-05 ***
## TARGET_FLAG0:JOBManager                      1.000000    
## TARGET_FLAG1:JOBManager                       < 2e-16 ***
## TARGET_FLAG0:JOBStudent                      1.000000    
## TARGET_FLAG1:JOBStudent                      1.70e-08 ***
## TARGET_FLAG0:CAR_USECommercial               1.000000    
## TARGET_FLAG1:CAR_USECommercial               2.11e-08 ***
## TARGET_FLAG0:`CAR_TYPEPanel Truck`           1.000000    
## TARGET_FLAG1:`CAR_TYPEPanel Truck`           3.13e-14 ***
## TARGET_FLAG0:CAR_TYPEPickup                  1.000000    
## TARGET_FLAG1:CAR_TYPEPickup                  0.042250 *  
## TARGET_FLAG0:`CAR_TYPESports Car`            1.000000    
## TARGET_FLAG1:`CAR_TYPESports Car`            0.009796 ** 
## TARGET_FLAG0:CAR_TYPESUV                    1.000000
## TARGET_FLAG1:CAR_TYPESUV                    0.000143 ***
## TARGET_FLAG0:`URBANICITYHighly Rural/ Rural` 1.000000
## TARGET_FLAG1:`URBANICITYHighly Rural/ Rural` 0.000826 ***
## TARGET_FLAG0:HOME_VAL_CAT.Low               1.000000
## TARGET_FLAG1:HOME_VAL_CAT.Low                < 2e-16 ***
## TARGET_FLAG0:HOME_VAL_CAT.Medium            1.000000
## TARGET_FLAG1:HOME_VAL_CAT.Medium            0.004607 **
## TARGET_FLAG0:HOME_VAL_CAT.High              1.000000
## TARGET_FLAG1:HOME_VAL_CAT.High              0.001207 **
## TARGET_FLAG0:REVOKED.No                     1.000000
## TARGET_FLAG1:REVOKED.No                      < 2e-16 ***
## TARGET_FLAG0:PARENT1.No                     1.000000
## TARGET_FLAG1:PARENT1.No                     0.014013 *
## TARGET_FLAG0:MSTATUS.No                     1.000000
## TARGET_FLAG1:MSTATUS.No                     1.77e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4053 on 28487 degrees of freedom
## Multiple R-squared:  0.3133, Adjusted R-squared:  0.3114
## F-statistic: 168.8 on 77 and 28487 DF,  p-value: < 2.2e-16
```

Model Comparison We now have two models that share a key similarity but differ in how they were built. Both models incorporate **TARGET_FLAG** to guide predictions of **TARGET_AMT**, a strategy we've intentionally preserved due to its clear relevance. The primary distinction lies in the variable selection process: the first model was manually curated using insights from our best logistic regression model, while the second was generated through an automated stepwise regression approach. Naturally, it's essential to compare their performance to determine which model is more effective.

We summarize key metrics below:

Table 21: Multiple Linear Regression Models Comparison

Model	RMSE	MAE	R_squared
Best Stepwise	3636.917	987.6474	0.2676990
MLR2	3599.482	1131.3922	0.2826969

The comparison table offers valuable insight for selecting our final model. While MLR2 has a marginally lower RMSE and a slightly higher R-squared, it's important to emphasize that in this context, RMSE and especially MAE carry more weight than R-squared. Since our primary goal is to minimize financial prediction errors, lower RMSE and MAE values indicate more reliable payout estimates.

Although Best Stepwise slightly underperforms on RMSE and R-squared, it delivers a significantly lower MAE, which reflects the average prediction error. This makes it particularly relevant for applications where accuracy in dollar amounts is crucial. Furthermore, in this case, predictive accuracy outweighs model simplicity, meaning interpretability is not the top priority.

Given these factors, despite MLR2's slight edge in two metrics, the much stronger MAE performance of Best Stepwise justifies its selection as the final model.

Predictions

With our final models selected, we're now prepared to generate predictions using the evaluation dataset. However, this process is slightly more involved, as our second model's predictions depend on the output of the first. The complete prediction workflow is outlined below:

Table 22: Sample 10 Predictions for Evaluation Dataset

	Index	TARGET_FLAG	TARGET_AMT
11	11	0	0
12	12	0	0
13	13	0	0
14	14	0	0
15	15	0	0
16	16	0	0
17	17	0	0
18	18	0	0
19	19	0	0
20	20	0	0

Conclusion

In this project, our objective was twofold: to predict whether a car crash would occur and, if so, to estimate the resulting payout. Achieving this required extensive data exploration, cleaning, and transformation. To predict crash occurrences, we built several binary logistic regression models and selected the most effective one. We then incorporated these predictions into our payout estimation by using interaction terms, ensuring payouts were only predicted when a crash was likely. For improved accuracy, we applied stepwise regression to develop our highest-performing payout model. Finally, we generated predictions using both models sequentially. These models provide valuable insights that could help insurance companies design policies that are not only equitable, but also financially sound and risk-conscious.

Appendix: Report Code

Below is the code for this report to generate the models and charts above.

```
knitr::opts_chunk$set(echo = FALSE)


library(janitor)
library(kableExtra)
library(latex2exp)
library(psych)
library(scales)
library(stringr)
library(ggcorrplot)
library(tidyverse)
library(mice)
library(ggmice)
library(caret)
library(bestNormalize)
library(e1071)
library(car)
library(glmnet)
library(pROC)
library(Metrics)
table_def <- "
| **VARIABLE**     | **DEFINITION**                           | **THEORETICAL EFFECT**
|:------------------|:-----------------------------------------|:---------------------------------------
```

```
| `INDEX`          | Identification Variable (do not use)     | None
| `TARGET_FLAG`    | Was Car in a crash? 1=YES 0=NO           | None
| `TARGET_AMT`     | If car was in a crash, what was the cost | None
| `AGE`            | Age of Driver                            | Very young people tend to be risky. May
| `BLUEBOOK`       | Value of Vehicle                         | Unknown effect on probability of collis
| `CAR_AGE`        | Vehicle Age                              | Unknown effect on probability of collis
| `CAR_TYPE`       | Type of Car                              | Unknown effect on probability of collis
| `CAR_USE`        | Vehicle Use                              | Commercial vehicles are driven more, so
| `CLM_FREQ`       | # Claims (Past 5 Years)                  | The more claims you filed in the past,
| `EDUCATION`      | Max Education Level                      | Unknown effect, but in theory more educa
| `HOMEKIDS`       | # Children at Home                       | Unknown effect
| `HOME_VAL`       | Home Value                               | In theory, home owners tend to drive mo
| `INCOME`         | Income                                   | In theory, rich people tend to get into
| `JOB`            | Job Category                             | In theory, white collar jobs tend to be
| `KIDSDRIV`       | # Driving Children                       | When teenagers drive your car, you are
| `MSTATUS`        | Marital Status                           | In theory, married people drive more sa
| `MVR_PTS`        | Motor Vehicle Record Points              | If you get lots of traffic tickets, you
| `OLDCLAIM`       | Total Claims (Past 5 Years)              | If your total payout over the past five
| `PARENT1`        | Single Parent                            | Unknown effect
| `RED_CAR`        | A Red Car                                | Urban legend says that red cars (especia
| `REVOKED`        | License Revoked (Past 7 Years)           | If your license was revoked in the past
| `SEX`            | Gender                                   | Urban legend says that women have less
| `TIF`            | Time in Force                            | People who have been customers for a lo
| `TRAVTIME`       | Distance to Work                         | Long drives to work usually suggest grea
| `URBANICITY`     | Home/Work Area                           | Unknown
| `YOJ`            | Years on Job                             | People who stay at a job for a long tim
"
cat(table_def)
url <- "https://raw.githubusercontent.com/Shriyanshh/DATA-621/refs/heads/main/insurance_training_data.c
eval_url <- "https://raw.githubusercontent.com/Shriyanshh/DATA-621/refs/heads/main/insurance-evaluation

train <- read.csv(url)
eval <- read.csv(eval_url)
kbl(head(train), caption = "Training Set") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  footnote(general_title = "Dimensions: ",
          TeX(paste0(nrow(train), " x ", ncol(train)))) %>%
  kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::landscape()
kbl(head(eval), caption = "Evaluation Set") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  footnote(general_title = "Dimensions: ",
          TeX(paste0(nrow(eval), " x ", ncol(eval)))) %>%
  kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::landscape()
train <-
  train |>
  select(-INDEX)

eval <-
  eval |>
  select(-INDEX)
```

```
kbl(head(train), caption = "Training Set") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  footnote("Dropped `INDEX` column:") %>%
  kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::landscape()
preview <-
  train |>
  select(INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM)

kbl(head(preview), caption = "Training Set: Before") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")

train <-
  train |>
  mutate(INCOME = as.numeric(gsub("[^\\d]", "", train$INCOME, perl = TRUE)),
         HOME_VAL = as.numeric(gsub("[^\\d]", "", train$HOME_VAL, perl = TRUE)),
         BLUEBOOK = as.numeric(gsub("[^\\d]", "", train$BLUEBOOK, perl = TRUE)),
         OLDCLAIM = as.numeric(gsub("[^\\d]", "", train$OLDCLAIM, perl = TRUE)))

eval <-
  eval |>
  mutate(INCOME = as.numeric(gsub("[^\\d]", "", eval$INCOME, perl = TRUE)),
         HOME_VAL = as.numeric(gsub("[^\\d]", "", eval$HOME_VAL, perl = TRUE)),
         BLUEBOOK = as.numeric(gsub("[^\\d]", "", eval$BLUEBOOK, perl = TRUE)),
         OLDCLAIM = as.numeric(gsub("[^\\d]", "", eval$OLDCLAIM, perl = TRUE)))

preview <-
  train |>
  select(INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM)

kbl(head(preview), caption = "Training Set: After") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")
preview <-
  train |>
  select(MSTATUS, SEX, EDUCATION, JOB, CAR_TYPE, URBANICITY)

kbl(head(preview), caption = "Training Set: Before") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")

train <-
  train |>
  mutate(MSTATUS = str_remove(MSTATUS, "^z_"),
         SEX = str_remove(SEX, "^z_"),
         EDUCATION = str_remove(EDUCATION, "^z_"),
         JOB = str_remove(JOB, "^z_"),
         CAR_TYPE = str_remove(CAR_TYPE, "^z_"),
         URBANICITY = str_remove(URBANICITY, "^z_"))

eval <-
  eval |>
  mutate(MSTATUS = str_remove(MSTATUS, "^z_"),
         SEX = str_remove(SEX, "^z_"),
         EDUCATION = str_remove(EDUCATION, "^z_"),
         JOB = str_remove(JOB, "^z_"),
         CAR_TYPE = str_remove(CAR_TYPE, "^z_"),
         URBANICITY = str_remove(URBANICITY, "^z_"))

preview <-
  train |>
  select(MSTATUS, SEX, EDUCATION, JOB, CAR_TYPE, URBANICITY)

kbl(head(preview), caption = "Training Set: After") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")
preview <-
  train |>
  select(PARENT1, MSTATUS, SEX, RED_CAR, REVOKED, EDUCATION)

kbl(head(preview), caption = "Training Set: Before") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")

train <-
  train |>
  mutate(PARENT1 = as.factor(PARENT1),
         MSTATUS = as.factor(MSTATUS),
         SEX = as.factor(SEX),
         RED_CAR = as.factor(str_to_title(RED_CAR)),
         REVOKED = as.factor(REVOKED),
         EDUCATION = ordered(as.factor(EDUCATION), levels=c("<High School", "High School", "Bachelors",

eval <-
  eval |>
  mutate(PARENT1 = as.factor(PARENT1),
         MSTATUS = as.factor(MSTATUS),
         SEX = as.factor(SEX),
         RED_CAR = as.factor(str_to_title(RED_CAR)),
         REVOKED = as.factor(REVOKED),
         EDUCATION = ordered(as.factor(EDUCATION), levels=c("<High School", "High School", "Bachelors",

preview <-
  train |>
  select(PARENT1, MSTATUS, SEX, RED_CAR, REVOKED, EDUCATION)

kbl(head(preview), caption = "Training Set: After") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")
desc_train <- describe(train, omit = TRUE)

kbl(desc_train, digits=2, caption = "Summary Statistics") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")  %>%
  kableExtra::landscape()
```

```r
## Split dataset into categorical and continuous variables
train_cont <-
  train |>
  select(rownames(desc_train))

train_cat <-
  train |>
  select(-rownames(desc_train))
train_cont |>
  gather(key = "variable", value = "value") |>
  ggplot(aes(x = value)) +
  geom_histogram(aes(y = after_stat(density)), bins = 20, fill = '#4E79A7', color = 'black') +
  stat_density(geom = "line", color = "red") +
  facet_wrap(~ variable, scales = 'free') +
  theme(strip.text = element_text(size = 5)) +
  theme_bw()
train_cat |>
  gather(key = "variable", value = "value") |>
  ggplot(aes(y = value)) +
  geom_bar(aes(x = after_stat(count)), bins = 20, fill = '#4E79A7', color = 'black') +
  facet_wrap(~ variable, scales = 'free') +
  theme(strip.text = element_text(size = 5)) +
  theme_bw() +
  labs(y = "")
train_cont %>%
  gather(key = "Variable", value = "Value") |>
  ggplot(aes(x = "", y = Value)) +
  geom_boxplot(fill = "#4E79A7") +
  facet_wrap(~ Variable, scales = "free") +
  labs(x = NULL, y = "Value") +
  theme(strip.text = element_text(size = 5))
q <- cor(train_cont)

ggcorrplot(q, type = "upper", outline.color = "white",
           ggtheme = theme_classic,
           colors = c("#F28E2B", "white", "#4E79A7"),
           lab = TRUE, show.legend = F, tl.cex = 5, lab_size = 3)
missing_val <-
  train %>%
  summarise(across(everything(), ~ sum(is.na(.x)))) %>%
  select_if(function(.) last(.) != 0)

kbl(missing_val, caption = "Missing Values Count") |>
    kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position")
train$HOME_VAL[train$HOME_VAL == 0] <- NA
eval$HOME_VAL[eval$HOME_VAL == 0] <- NA
train %>%
  ggplot(aes(x = HOME_VAL)) +
  geom_histogram(aes(y = after_stat(density)), bins = 20, fill = '#4E79A7', color = 'black') +
  stat_density(geom = "line", color = "red") +
  theme_bw()
count_zeros <- function(column) {
  zero_count <- sum(column == 0, na.rm = TRUE)
  return(zero_count)
}

zero_counts_train <- sapply(train, count_zeros)
zero_counts_df <- data.frame("Zero Count" = zero_counts_train)

kbl(zero_counts_df, caption = "Zero Counts in Training Dataset") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")
percentMiss <- function(x){sum(is.na(x))/length(x)*100} # Creates percentage of missing values

# Cut offs for variable dropping was 25% of values missing - none were dropped
# Cut offs for sample dropping was 50% of values missing - none were dropped

variable_pMiss <- apply(train,2,percentMiss) # 2 = runs on columns
sample_pMiss <- apply(train,1,percentMiss) # 1 = runs on rows

#sum(sample_pMiss > 50)

pMiss <- data.frame(variables = names(variable_pMiss),pMiss = (variable_pMiss), row.names = NULL)
pMiss <- pMiss %>% arrange(desc(pMiss))


pMiss |>
  ggplot(aes(x = reorder(variables,pMiss), y = pMiss)) +
  geom_bar(stat = 'identity', fill = '#4E79A7', color = 'black') +
  theme(strip.text = element_text(size = 5)) +
  theme_bw() +
  scale_x_discrete(guide = guide_axis(angle = 45))+
  labs(x = 'Variables',y = 'Percent Missing',title = 'Percent of Missing Values by Variable')

train$HOME_VAL_CAT <- cut(
  train$HOME_VAL,
  breaks = c(0, 150000, 200000, 270000, Inf),
  labels = c("Very Low", "Low", "Medium", "High"),
  right = FALSE,
  include.lowest = TRUE
)

train$HOME_VAL_CAT <- factor(train$HOME_VAL_CAT, levels = c("Very Low", "Low", "Medium", "High", "Rente
train$HOME_VAL_CAT[is.na(train$HOME_VAL)] <- "Renters"

eval$HOME_VAL_CAT <- cut(
  eval$HOME_VAL,
  breaks = c(0, 150000, 200000, 270000, Inf),
  labels = c("Very Low", "Low", "Medium", "High"),
  right = FALSE,
  include.lowest = TRUE
)

eval$HOME_VAL_CAT <- factor(eval$HOME_VAL_CAT, levels = c("Very Low", "Low", "Medium", "High", "Renters
eval$HOME_VAL_CAT[is.na(eval$HOME_VAL)] <- "Renters"
```

```
train %>%
ggplot(aes(y = HOME_VAL_CAT)) +
  geom_bar(fill = '#4E79A7', color = 'black') +
  theme_bw() +
  labs(y = "")
plot_pattern(train, square = TRUE, rotate = TRUE, npat = 6)


train <- train %>%
  select(-HOME_VAL)

eval <- eval %>%
  select(-HOME_VAL)
set.seed(123)

trainIndex <- createDataPartition(y = train$TARGET_FLAG, p = 0.7, list = FALSE, times = 1)

train_data <- train[trainIndex,]
test_data <- train[-trainIndex,]
train_data_no_targets <- train_data[, !colnames(train_data) %in% c("TARGET_FLAG", "TARGET_AMT")]
test_data_no_targets <- test_data[, !colnames(test_data) %in% c("TARGET_FLAG", "TARGET_AMT")]
eval_data_no_targets <- eval[, !colnames(eval) %in% c("TARGET_FLAG", "TARGET_AMT")]

combined_data <- rbind(train_data_no_targets, test_data_no_targets, eval_data_no_targets)

data_type <- c(rep("train", nrow(train_data)),
               rep("test", nrow(test_data)),
               rep("eval", nrow(eval)))

impute_func <- function(data, data_type) {
    ini <- mice(data, maxit = 0, ignore = data_type != "train")
    meth <- ini$meth
    imputed_object <- mice(data, method = meth, m = 5, maxit = 30, seed = 500, print = FALSE)
    imputed_data <- complete(imputed_object, "long")

    return(list(imputed_object = imputed_object, imputed_data = imputed_data))
}

results <- impute_func(combined_data, data_type)

reintegrate_targets <- function(imputed_data, original_data, target_vars) {
    if (!all(target_vars %in% colnames(original_data))) {
        stop("Target variables not found in the original data")
    }
    target_data <- original_data[target_vars]
    imputed_data_with_targets <- cbind(imputed_data, target_data)
    return(imputed_data_with_targets)
}

full_combined_data <- rbind(train_data, test_data, eval)

imputed_data_with_targets <- reintegrate_targets(results$imputed_data, full_combined_data, c("TARGET_FL
```

```r
train_data_imputed <- imputed_data_with_targets[data_type == "train", ]
test_data_imputed <- imputed_data_with_targets[data_type == "test", ]
eval_data_imputed <- imputed_data_with_targets[data_type == "eval", ]

train_data_imputed <- train_data_imputed[, !colnames(train_data_imputed) %in% c(".imp", ".id")]
test_data_imputed <- test_data_imputed[, !colnames(test_data_imputed) %in% c(".imp", ".id")]
tracking_df <- data.frame(eval_data_imputed)
eval_data_imputed <- eval_data_imputed[, !colnames(eval_data_imputed) %in% c(".imp", ".id")]

generate_summary <- function(data, vars, dataset_name) {
    summary_stats <- data %>%
        select(all_of(vars)) %>%
        summarise(across(everything(), list(
            min = ~min(., na.rm = TRUE),
            q1 = ~quantile(., probs = 0.25, na.rm = TRUE),
            median = ~median(., na.rm = TRUE),
            mean = ~mean(., na.rm = TRUE),
            q3 = ~quantile(., probs = 0.75, na.rm = TRUE),
            max = ~max(., na.rm = TRUE)
        ))) %>%
        pivot_longer(cols = everything(), names_to = "Variable_Stat", values_to = "Value") %>%
        mutate(Dataset = dataset_name)
    return(summary_stats)
}

variables <- c("CAR_AGE", "YOJ", "INCOME", "AGE")

summary_full_train <- generate_summary(train_data, variables, "Dataset (Pre-Imputations)")
summary_train_imputed <- generate_summary(train_data_imputed, variables, "Train Imputed")
summary_test_imputed <- generate_summary(test_data_imputed, variables, "Test Imputed")

combined_summary <- bind_rows(summary_full_train, summary_train_imputed, summary_test_imputed)

# pivoting wide so it's easier to compare
final_summary <- combined_summary %>%
    pivot_wider(names_from = Dataset, values_from = Value) %>%
    mutate(across(where(is.numeric), ~format(., scientific = FALSE)))

kbl(final_summary, caption = "Summary Statistics Comparison Across Datasets") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position")

variables <- c("BLUEBOOK", "INCOME", "MVR_PTS", "OLDCLAIM", "TIF", "TRAVTIME", "YOJ", "CLM_FREQ", "CAR_A

apply_best_normalize <- function(data, variables) {
  results <- data.frame(Variable = character(), Transformation = character(), stringsAsFactors = FALSE)

  for (var in variables) {
    has_negatives <- any(data[[var]] < 0, na.rm = TRUE)
    BN_object <- bestNormalize(data[[var]], allow.negative = has_negatives)
    #print(list(BN_object))

    if (is.list(BN_object$chosen_transform)) {
      best_method <- attr(BN_object$chosen_transform, "class")[1]
    } else {
      best_method <- "Check Structure"  #In case structure is unexpected
    }

    results <- rbind(results, data.frame(Variable = var, Transformation = best_method))
   # cat("Best transformation for", var, ":", best_method, "\n")
  }

  return(results)
}

BN_results_train <- apply_best_normalize(train_data_imputed, variables)

kbl(BN_results_train, caption = "Best Transformations") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position")


calculate_transformations <- function(data) {
  list(
    BLUEBOOK_bn = orderNorm(data$BLUEBOOK),
    INCOME_bn = orderNorm(data$INCOME),
    MVR_PTS_sqrt = function(x) sqrt(x + 0),
    OLDCLAIM_cs = function(x) scale(x),
    TIF_yj = yeojohnson(data$TIF),
    TRAVTIME_bc = boxcox(data$TRAVTIME),
    YOJ_sqrt = function(x) sqrt(x + 0),
    CLM_FREQ_sqrt = function(x) sqrt(x + 0),
    CAR_AGE_yj = yeojohnson(data$CAR_AGE)
  )
}

apply_pre_calculated_transformations <- function(data, transforms) {
  data %>%
    mutate(
      BLUEBOOK_transformed = predict(transforms$BLUEBOOK_bn, newdata = BLUEBOOK),
      INCOME_transformed = predict(transforms$INCOME_bn, newdata = INCOME),
      MVR_PTS_transformed = transforms$MVR_PTS_sqrt(MVR_PTS),
      OLDCLAIM_transformed = transforms$OLDCLAIM_cs(OLDCLAIM)[, 1],
      OLDCLAIM = OLDCLAIM,
      TIF_transformed = predict(transforms$TIF_yj, newdata = TIF),
      TRAVTIME_transformed = predict(transforms$TRAVTIME_bc, newdata = TRAVTIME),
      YOJ_transformed = transforms$YOJ_sqrt(YOJ),
      CLM_FREQ_transformed = transforms$CLM_FREQ_sqrt(CLM_FREQ),
      CAR_AGE_transformed = predict(transforms$CAR_AGE_yj, newdata = CAR_AGE)
    )
}

transform_params <- calculate_transformations(train_data_imputed)

train_data_transformed <- apply_pre_calculated_transformations(train_data_imputed, transform_params)
test_data_transformed <- apply_pre_calculated_transformations(test_data_imputed, transform_params)
```

```r
eval_data_transformed <- apply_pre_calculated_transformations(eval_data_imputed, transform_params)


pre_trans_skew <- summarise(train_data_imputed,
                            across(c(BLUEBOOK, INCOME, MVR_PTS, OLDCLAIM, TIF, TRAVTIME, YOJ, CLM_FREQ,
                                   skewness,
                                   na.rm = TRUE)) %>%
                            pivot_longer(everything(), names_to = "Variable", values_to = "Pre-Transform

post_trans_skew <- summarise(train_data_transformed,
                             across(c(BLUEBOOK_transformed, INCOME_transformed, MVR_PTS_transformed, OL
                                    skewness,
                                    na.rm = TRUE)) %>%
                             pivot_longer(everything(), names_to = "Variable", values_to = "Post-Transf

post_trans_skew$Variable <- sub("_transformed", "", post_trans_skew$Variable)

skewness_comparison <- left_join(pre_trans_skew, post_trans_skew, by = "Variable")

kbl(skewness_comparison, caption = "Pre and Post Transformation Skewness Comparison", digits = 3) %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position")



calc_outliers <- function(data, columns) {
  sapply(columns, function(column) {
    Q1 <- quantile(data[[column]], 0.25, na.rm = TRUE)
    Q3 <- quantile(data[[column]], 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    list(lower = Q1 - 1.5 * IQR, upper = Q3 + 1.5 * IQR)
  }, simplify = FALSE)
}

#continuous
columns_to_check <- c("AGE", "BLUEBOOK", "INCOME", "MVR_PTS", "OLDCLAIM", "TIF", "TRAVTIME", "YOJ", "CL
limits <- calc_outliers(train_data_transformed, columns_to_check)

#replace outliers with median
replace_outliers <- function(data, columns, limits) {
  for(column in columns) {
    median_value <- median(data[[column]], na.rm = TRUE)
    lower_limit <- limits[[column]]$lower
    upper_limit <- limits[[column]]$upper

    data[[column]] <- ifelse(data[[column]] < lower_limit | data[[column]] > upper_limit,
                             median_value,
                             data[[column]])
  }
  return(data)
}

train_data_cleaned <- replace_outliers(train_data_transformed, columns_to_check, limits)
```

```
test_data_cleaned <- replace_outliers(test_data_transformed, columns_to_check, limits)
eval_data_cleaned <- replace_outliers(eval_data_transformed, columns_to_check, limits)


cont_cols <- c('AGE', 'BLUEBOOK', 'INCOME', 'MVR_PTS', 'OLDCLAIM', 'TIF', 'TRAVTIME', 'YOJ', 'KIDSDRIV'
cat_cols <- c('SEX', 'EDUCATION', 'JOB', 'CAR_USE', 'CAR_TYPE', 'URBANICITY', 'HOME_VAL_CAT', 'RED_CAR'

preprocess_params <- preProcess(train_data_cleaned[cont_cols], method = c("center", "scale"))

train_data_processed <- predict(preprocess_params, train_data_cleaned)
test_data_processed <- predict(preprocess_params, test_data_cleaned)
eval_data_processed <- predict(preprocess_params, eval_data_cleaned)

# exclude OLDCLAIM!
nzv_params <- preProcess(train_data_processed[, setdiff(cont_cols, "OLDCLAIM")], method = "nzv")
nzv_features <- predict(nzv_params, train_data_processed[, setdiff(cont_cols, "OLDCLAIM")])

train_data_processed <- cbind(nzv_features, OLDCLAIM = train_data_processed$OLDCLAIM)
test_data_processed <- cbind(predict(nzv_params, test_data_processed[, setdiff(cont_cols, "OLDCLAIM")])
eval_data_processed <- cbind(predict(nzv_params, eval_data_processed[, setdiff(cont_cols, "OLDCLAIM")])

transformed_columns <- c("BLUEBOOK_transformed", "INCOME_transformed", "MVR_PTS_transformed",
                         "OLDCLAIM_transformed", "TIF_transformed", "TRAVTIME_transformed",
                         "YOJ_transformed", "CLM_FREQ_transformed", "CAR_AGE_transformed")
encoded_columns <- c("SEX.F", "SEX.M", "EDUCATION.L", "EDUCATION.Q", "EDUCATION.C",
                     "JOBBlue Collar", "JOBClerical", "JOBDoctor", "JOBHome Maker", "JOBLawyer",
                     "JOBManager", "JOBProfessional", "JOBStudent", "CAR_USECommercial",
                     "CAR_USEPrivate", "CAR_TYPEMinivan", "CAR_TYPEPanel Truck", "CAR_TYPEPickup",
                     "CAR_TYPESports Car", "CAR_TYPESUV", "CAR_TYPEVan", "URBANICITYHighly Rural/ Rural
                     "URBANICITYHighly Urban/ Urban", "HOME_VAL_CAT.Very Low", "HOME_VAL_CAT.Low",
                     "HOME_VAL_CAT.Medium", "HOME_VAL_CAT.High", "HOME_VAL_CAT.Renters", "RED_CAR.No",
                     "RED_CAR.Yes", "REVOKED.No", "REVOKED.Yes", "PARENT1.No", "PARENT1.Yes",
                     "MSTATUS.No", "MSTATUS.Yes")


dummyVars_obj <- dummyVars(~ ., data = train_data_cleaned[cat_cols], levelsOnly = FALSE)
train_data_encoded <- predict(dummyVars_obj, newdata = train_data_cleaned[cat_cols])
test_data_encoded <- predict(dummyVars_obj, newdata = test_data_cleaned[cat_cols])
eval_data_encoded <- predict(dummyVars_obj, newdata = eval_data_cleaned[cat_cols])


train_data_encoded_df <- as.data.frame(train_data_encoded)
colnames(train_data_encoded_df) <- attr(train_data_encoded, "dimnames")[[2]]

test_data_encoded_df <- as.data.frame(test_data_encoded)
colnames(test_data_encoded_df) <- attr(test_data_encoded, "dimnames")[[2]]

eval_data_encoded_df <- as.data.frame(eval_data_encoded)
colnames(eval_data_encoded_df) <- attr(eval_data_encoded, "dimnames")[[2]]



train_final <- cbind(train_data_processed, train_data_cleaned[transformed_columns], train_data_encoded)
```

```r
test_final <- cbind(test_data_processed, test_data_cleaned[transformed_columns], test_data_encoded)
eval_final <- cbind(eval_data_processed, eval_data_cleaned[transformed_columns], eval_data_encoded)

# Combine
train_final <- cbind(train_data_processed, train_data_cleaned[transformed_columns], train_data_encoded_d
                     TARGET_FLAG = train_data_cleaned$TARGET_FLAG, TARGET_AMT = train_data_cleaned$TARGE

test_final <- cbind(test_data_processed, test_data_cleaned[transformed_columns], test_data_encoded_df,
                    TARGET_FLAG = test_data_cleaned$TARGET_FLAG, TARGET_AMT = test_data_cleaned$TARGET_A

eval_final <- cbind(eval_data_processed, eval_data_cleaned[transformed_columns], eval_data_encoded_df,
                    TARGET_FLAG = eval_data_cleaned$TARGET_FLAG, TARGET_AMT = eval_data_cleaned$TARGET_A

#write_csv(train_final,"data\\train_final.csv")
#write_csv(test_final,"data\\test_final.csv")
#write_csv(eval_final,"data\\eval_final.csv")

target_columns <- c("TARGET_FLAG", "TARGET_AMT")

original_columns <- c("KIDSDRIV", "AGE", "HOMEKIDS", "YOJ", "INCOME", "PARENT1", "MSTATUS",
                      "SEX", "EDUCATION", "JOB", "TRAVTIME", "CAR_USE", "BLUEBOOK", "TIF",
                      "CAR_TYPE", "RED_CAR", "OLDCLAIM", "CLM_FREQ", "REVOKED", "MVR_PTS",
                      "CAR_AGE", "URBANICITY", "HOME_VAL_CAT")

# other categories defined earlier....

# fyi, making df to show in a table

variable_process_df <- data.frame(
  Variable = c(original_columns, transformed_columns, encoded_columns),
  Category = c(rep("Original", length(original_columns)),
             rep("Transformed", length(transformed_columns)),
             rep("Encoded", length(encoded_columns)))
)

summary_table <- variable_process_df %>%
  group_by(Category) %>%
  summarize(Variables = paste(collapse = ", ", Variable))

kbl(summary_table, caption = "Variables Summary") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::landscape()
simple_model1 <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_columns, "TARG
summary(simple_model1)
cor_mat <- cor(train_final[, c(encoded_columns, "TARGET_FLAG")])
#kbl(cor_mat[,"TARGET_FLAG"])


kbl(cor_mat[,"TARGET_FLAG"], caption = "Correlations with TARGET FLAG") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position")
columns_causing_multicollinearity <- c("SEX.M", "CAR_USEPrivate", "URBANICITYHighly Urban/ Urban",
                                       "RED_CAR.Yes", "REVOKED.Yes", "PARENT1.Yes", "MSTATUS.Yes",
                                       "EDUCATION.Q", "JOBHome Maker", "CAR_TYPEVan", "HOME_VAL_CAT.Low

encoded_columns_filtered <- setdiff(encoded_columns, columns_causing_multicollinearity)
simple_model2 <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_columns_filte
summary(simple_model2)
kbl(vif(simple_model2), caption = "VIF Values simple model 2") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position")
encoded_columns_filtered2 <- setdiff(encoded_columns_filtered, "CAR_TYPESUV")

transformed_model <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_columns_fi

summary(transformed_model)

encoded_columns_filtered3 <- setdiff(encoded_columns_filtered2, c("SEX.F", "RED_CAR.No"))

final_transformed_model <- glm(TARGET_FLAG ~ ., data = train_final[, c(transformed_columns, encoded_colu

summary(final_transformed_model)

original_non_cat = c("KIDSDRIV", "AGE", "HOMEKIDS", "YOJ", "INCOME", "TRAVTIME", "BLUEBOOK", "TIF",
                       "OLDCLAIM", "CLM_FREQ", "MVR_PTS",
                      "CAR_AGE")

nontransformed_model1 <- glm(TARGET_FLAG ~ ., data = train_final[, c(original_non_cat, encoded_columns_
summary(nontransformed_model1)
non_cat2 = c("KIDSDRIV", "AGE", "YOJ", "INCOME", "TRAVTIME", "BLUEBOOK", "TIF",
                       "OLDCLAIM", "CLM_FREQ", "MVR_PTS",
                      "CAR_AGE")

nontransformed_model2 <- glm(TARGET_FLAG ~ ., data = train_final[, c(non_cat2, encoded_columns_filtered
summary(nontransformed_model2)
non_cat3 = c("KIDSDRIV", "AGE", "YOJ", "INCOME", "TRAVTIME", "BLUEBOOK", "TIF",
                       "OLDCLAIM", "CLM_FREQ", "MVR_PTS")

final_nontransformed_model <- glm(TARGET_FLAG ~ ., data = train_final[, c(non_cat3, encoded_columns_fil
summary(final_nontransformed_model)
predictors <- as.matrix(train_final[, c(transformed_columns, encoded_columns)]) #has to be matrix
train_final$TARGET_FLAG <- as.factor(train_final$TARGET_FLAG)
response <- train_final$TARGET_FLAG

lasso_model <- glmnet(predictors, response, family = "binomial", alpha = 1)

# We'll use cv.glmnet to perform cross-validation to select lambda (regularization parameter)
cv_lasso <- cv.glmnet(predictors, response, family = "binomial", alpha = 1, type.measure = "class")

plot(cv_lasso)

# Coefficients at best lambda
best_lambda <- cv_lasso$lambda.min
coef(cv_lasso, s = "lambda.min")
```

```r
print(paste("Best Lambda: ", best_lambda))

# Model1
model_transformed_predictions <- predict(final_transformed_model, newdata = test_final, type = "respons
model_transformed_binary_predictions <- ifelse(model_transformed_predictions >= 0.5, 1, 0)

model_transformed_rmse <- rmse(test_final$TARGET_FLAG, model_transformed_binary_predictions)

# Model2
model2_predictions <- predict(final_nontransformed_model, newdata = test_final, type = "response")
model2_binary_predictions <- ifelse(model2_predictions >= 0.5, 1, 0)

model2_rmse <- rmse(test_final$TARGET_FLAG, model2_binary_predictions)

# Lasso Model
predictors <- test_final[, c(transformed_columns, encoded_columns)]
predictors_matrix <- as.matrix(predictors)

# predictions using the best lambda
lasso_model_predictions <- predict(lasso_model, newx = predictors_matrix, s = best_lambda, type = "resp
lasso_model_binary_predictions <- ifelse(lasso_model_predictions >= 0.5, 1, 0)

# RMSE for Lasso model predictions
lasso_model_rmse <- rmse(test_final$TARGET_FLAG, lasso_model_binary_predictions)

# Create a dataframe to store the results
results <- data.frame(Model = c("Model_Transformed", "Model_Untransformed", "Lasso Model"),
                      RMSE = c(model_transformed_rmse, model2_rmse, lasso_model_rmse))

# AIC
model_transformed_AIC <- AIC(final_transformed_model)
model2_AIC <- AIC(final_nontransformed_model)
lasso_model_AIC <- NA

results$AIC <- c(model_transformed_AIC, model2_AIC, lasso_model_AIC)

# Deviance
model_transformed_deviance <- deviance(final_transformed_model)
model2_deviance <- deviance(final_nontransformed_model)
lasso_model_deviance <- cv_lasso$cvm[cv_lasso$lambda == best_lambda]

# Add additional evaluation metrics to the results dataframe
conf_matrix <- table(model_transformed_binary_predictions, test_final$TARGET_FLAG)
TP <- conf_matrix[2, 2]
FP <- conf_matrix[1, 2]
FN <- conf_matrix[2, 1]
model_transformed_accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
model_transformed_precision <- TP / (TP + FP)
model_transformed_recall <- TP / (TP + FN)
model_transformed_f1_score <- 2 * (model_transformed_precision * model_transformed_recall) / (model_tra
model_transformed_roc_auc <- roc(test_final$TARGET_FLAG, model_transformed_predictions)$auc

conf_matrix <- table(model2_binary_predictions, test_final$TARGET_FLAG)
```

```r
TP <- conf_matrix[2, 2]
FP <- conf_matrix[1, 2]
FN <- conf_matrix[2, 1]
model2_accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
model2_precision <- TP / (TP + FP)
model2_recall <- TP / (TP + FN)
model2_f1_score <- 2 * (model2_precision * model2_recall) / (model2_precision + model2_recall)
model2_roc_auc <- roc(test_final$TARGET_FLAG, model2_predictions)$auc

conf_matrix <- table(lasso_model_binary_predictions, test_final$TARGET_FLAG)
TP <- conf_matrix[2, 2]
FP <- conf_matrix[1, 2]
FN <- conf_matrix[2, 1]
lasso_model_accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
lasso_model_precision <- TP / (TP + FP)
lasso_model_recall <- TP / (TP + FN)
lasso_model_f1_score <- 2 * (lasso_model_precision * lasso_model_recall) / (lasso_model_precision + lass
lasso_model_roc_auc <- roc(test_final$TARGET_FLAG, lasso_model_predictions)$auc



results$Accuracy <- c(model_transformed_accuracy, model2_accuracy, lasso_model_accuracy)
results$Precision <- c(model_transformed_precision, model2_precision, lasso_model_precision)
results$Recall <- c(model_transformed_recall, model2_recall, lasso_model_recall)
results$F1_Score <- c(model_transformed_f1_score, model2_f1_score, lasso_model_f1_score)
results$ROC_AUC <- c(model_transformed_roc_auc, model2_roc_auc, lasso_model_roc_auc)


kbl(results, caption = "Comparison of Logistic Models") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(latex_options = "HOLD_position") %>%
  kableExtra::landscape()
#have to add back ticks or it just won't work
predictors <- c("KIDSDRIV", "AGE", "YOJ", "INCOME", "TRAVTIME", "BLUEBOOK", "TIF",
                "OLDCLAIM", "CLM_FREQ", "MVR_PTS", "EDUCATION.L", "EDUCATION.C",
                "`JOBBlue Collar`", "JOBClerical", "JOBDoctor", "JOBLawyer",
                "JOBManager", "JOBProfessional", "JOBStudent", "CAR_USECommercial",
                "CAR_TYPEMinivan", "`CAR_TYPEPanel Truck`", "CAR_TYPEPickup",
                "`CAR_TYPESports Car`", "`URBANICITYHighly Rural/ Rural`",
                "`HOME_VAL_CAT.Very Low`", "HOME_VAL_CAT.Medium", "HOME_VAL_CAT.High",
                "HOME_VAL_CAT.Renters", "REVOKED.No", "PARENT1.No", "MSTATUS.No")

formula <- as.formula(paste("TARGET_AMT ~", paste(predictors, collapse=" + ")))
mlr1 <- lm(formula, data = train_final)
summary(mlr1)


interaction_terms <- paste(predictors, ":TARGET_FLAG", sep="")

formula <- as.formula(paste("TARGET_AMT ~", paste(interaction_terms, collapse=" + ")))

mlr2 <- lm(formula, data = train_final)
summary(mlr2)
```

```r
stepwise_regression <- function(data, response_var) {

  predictors_for_stepwise <- c("`AGE`", "`BLUEBOOK`", "`INCOME`",
                  "`MVR_PTS`", "`TIF`", "`TRAVTIME`",
                  "`YOJ`", "`KIDSDRIV`", "`HOMEKIDS`",
                  "`CLM_FREQ`", "`CAR_AGE`", "`OLDCLAIM`",
                  "`BLUEBOOK_transformed`", "`INCOME_transformed`", "`MVR_PTS_transformed`",
                  "`OLDCLAIM_transformed`", "`TIF_transformed`", "`TRAVTIME_transformed`",
                  "`YOJ_transformed`", "`CLM_FREQ_transformed`", "`CAR_AGE_transformed`",
                  "`SEX.F`", "`SEX.M`", "`EDUCATION.L`",
                  "`EDUCATION.Q`", "`EDUCATION.C`", "`EDUCATION^4`",
                  "`JOBBlue Collar`", "`JOBClerical`", "`JOBDoctor`",
                  "`JOBHome Maker`", "`JOBLawyer`", "`JOBManager`",
                  "`JOBProfessional`", "`JOBStudent`", "`CAR_USECommercial`",
                  "`CAR_USEPrivate`", "`CAR_TYPEMinivan`", "`CAR_TYPEPanel Truck`",
                  "`CAR_TYPEPickup`", "`CAR_TYPESports Car`", "`CAR_TYPESUV`",
                  "`CAR_TYPEVan`", "`URBANICITYHighly Rural/ Rural`", "`URBANICITYHighly Urban/ Urban`"
                  "`HOME_VAL_CAT.Very Low`", "`HOME_VAL_CAT.Low`", "`HOME_VAL_CAT.Medium`",
                  "`HOME_VAL_CAT.High`", "`HOME_VAL_CAT.Renters`", "`RED_CAR.No`",
                  "`RED_CAR.Yes`", "`REVOKED.No`", "`REVOKED.Yes`",
                  "`PARENT1.No`", "`PARENT1.Yes`", "`MSTATUS.No`", "`MSTATUS.Yes`")

  interaction_terms <- paste(predictors_for_stepwise, ":TARGET_FLAG", sep="")

  formula <- as.formula(paste(response_var, "~", paste(interaction_terms, collapse=" + ")))
  initial_model <- lm(formula, data = data)

  step_model <- step(initial_model, direction = "both")

  best_model <- step_model$call[[1]]  # final model formula
  return(best_model)
}

best_model_formula <- stepwise_regression(train_final, "TARGET_AMT")
```

```r
best_stepwise_formula <- TARGET_AMT ~ AGE:TARGET_FLAG + TARGET_FLAG:INCOME + TARGET_FLAG:TIF +
    TARGET_FLAG:YOJ + TARGET_FLAG:HOMEKIDS + TARGET_FLAG:CLM_FREQ +
    TARGET_FLAG:CAR_AGE + TARGET_FLAG:OLDCLAIM + TARGET_FLAG:BLUEBOOK_transformed +
    TARGET_FLAG:INCOME_transformed + TARGET_FLAG:MVR_PTS_transformed +
    TARGET_FLAG:TIF_transformed + TARGET_FLAG:YOJ_transformed +
    TARGET_FLAG:CLM_FREQ_transformed + TARGET_FLAG:CAR_AGE_transformed +
    TARGET_FLAG:SEX.F + TARGET_FLAG:SEX.M + TARGET_FLAG:EDUCATION.L +
    TARGET_FLAG:EDUCATION.Q + TARGET_FLAG:`EDUCATION^4` + TARGET_FLAG:`JOBBlue Collar` +
    TARGET_FLAG:JOBClerical + TARGET_FLAG:JOBDoctor + TARGET_FLAG:`JOBHome Maker` +
    TARGET_FLAG:JOBLawyer + TARGET_FLAG:JOBManager + TARGET_FLAG:JOBStudent +
    TARGET_FLAG:CAR_USECommercial + TARGET_FLAG:`CAR_TYPEPanel Truck` +
    TARGET_FLAG:CAR_TYPEPickup + TARGET_FLAG:`CAR_TYPESports Car` +
    TARGET_FLAG:CAR_TYPESUV + TARGET_FLAG:`URBANICITYHighly Rural/ Rural` +
    TARGET_FLAG:HOME_VAL_CAT.Low + TARGET_FLAG:HOME_VAL_CAT.Medium +
    TARGET_FLAG:HOME_VAL_CAT.High + TARGET_FLAG:REVOKED.No +
    TARGET_FLAG:PARENT1.No + TARGET_FLAG:MSTATUS.No

best_stepwise <- lm(formula = best_stepwise_formula, data = train_final)
summary(best_stepwise)

test_final$TARGET_FLAG <- as.factor(test_final$TARGET_FLAG)
predictions_best_stepwise <- predict(best_stepwise, newdata = test_final)
predictions_mlr2 <- predict(mlr2, newdata = test_final)
actual_values <- test_final$TARGET_AMT

# RMSE
rmse_best_stepwise <- sqrt(mean((predictions_best_stepwise - actual_values)^2))
rmse_mlr2 <- sqrt(mean((predictions_mlr2 - actual_values)^2))

# MAE
mae_best_stepwise <- mean(abs(predictions_best_stepwise - actual_values))
mae_mlr2 <- mean(abs(predictions_mlr2 - actual_values))

# R-squared
ss_total <- sum((actual_values - mean(actual_values))^2)
ss_residual_best_stepwise <- sum((actual_values - predictions_best_stepwise)^2)
ss_residual_mlr2 <- sum((actual_values - predictions_mlr2)^2)
r_squared_best_stepwise <- 1 - (ss_residual_best_stepwise / ss_total)
r_squared_mlr2 <- 1 - (ss_residual_mlr2 / ss_total)

evaluation_metrics <- data.frame(
  Model = c("Best Stepwise", "MLR2"),
  RMSE = c(rmse_best_stepwise, rmse_mlr2),
  MAE = c(mae_best_stepwise, mae_mlr2),
  R_squared = c(r_squared_best_stepwise, r_squared_mlr2)
)

kbl(evaluation_metrics, caption = "Multiple Linear Regression Models Comparison") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")

# TARGET_FLAG
eval_final$TARGET_FLAG <- predict(final_nontransformed_model, newdata = eval_final, type = "response")
eval_final$TARGET_FLAG <- ifelse(eval_final$TARGET_FLAG >= 0.5, 1, 0)
eval_final$TARGET_FLAG <- factor(eval_final$TARGET_FLAG, levels = c(0, 1))

#  TARGET_AMT
eval_final$TARGET_AMT <- predict(best_stepwise, newdata = eval_final, type = "response")
eval_final$TARGET_AMT <- ifelse(eval_final$TARGET_FLAG == "0", 0, eval_final$TARGET_AMT) # just in case

eval_final$.id <- tracking_df$.id #because we used MICE, we need to aggregate now

eval_final$TARGET_FLAG <- as.numeric(as.character(eval_final$TARGET_FLAG))
eval_final$TARGET_AMT <- as.numeric(as.character(eval_final$TARGET_AMT))

eval_aggregated <- eval_final %>%
  group_by(.id) %>%
  summarise(
    Median_TARGET_FLAG = median(TARGET_FLAG, na.rm = TRUE),
    Median_TARGET_AMT = ifelse(Median_TARGET_FLAG == 0, 0, median(TARGET_AMT, na.rm = TRUE))
  )

eval_predictions_only <- data.frame(
  Index = rownames(eval_aggregated),
  TARGET_FLAG = eval_aggregated$Median_TARGET_FLAG,
  TARGET_AMT = eval_aggregated$Median_TARGET_AMT
)

write.csv(eval_predictions_only, "Eval_Final_Predictions_Only.csv", row.names = FALSE)

kbl(eval_predictions_only[11:20, ], caption = "Sample 10 Predictions for Evaluation Dataset") |>
  kable_classic(full_width = F, html_font = "Cambria") |>
  kable_styling(latex_options = "HOLD_position")
```