HW4-Auto

2024-04-03

What is 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?

```
data_train <- read.csv('https://raw.githubusercontent.com/LeJQC/DATA-621-Group-2/main/HW4/insurance_tra
data_eval <- read.csv('https://raw.githubusercontent.com/LeJQC/DATA-621-Group-2/main/HW4/insurance-evalu
```

```
head(data_train)
```

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

Data Exploration

```
skim(data_train)
```

Table 1: Data summary

Name	data_train
Number of rows	8161

Number of columns	26
Column type frequency:	
character	14
numeric	12
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
INCOME	0	1	0	8	445	6613	0
PARENT1	0	1	2	3	0	2	0
HOME_VAL	0	1	0	8	464	5107	0
MSTATUS	0	1	3	4	0	2	0
SEX	0	1	1	3	0	2	0
EDUCATION	0	1	3	13	0	5	0
JOB	0	1	0	13	526	9	0
CAR_USE	0	1	7	10	0	2	0
BLUEBOOK	0	1	6	7	0	2789	0
CAR_TYPE	0	1	3	11	0	6	0
RED_CAR	0	1	2	3	0	2	0
OLDCLAIM	0	1	2	7	0	2857	0
REVOKED	0	1	2	3	0	2	0
URBANICITY	0	1	19	21	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INDEX	0	1.00	5151.87	2978.89	1	2559	5133	7745	10302.0	
TARGET_FLAG	0	1.00	0.26	0.44	0	0	0	1	1.0	
TARGET_AMT	0	1.00	1504.32	4704.03	0	0	0	1036	107586.1	
KIDSDRIV	0	1.00	0.17	0.51	0	0	0	0	4.0	
AGE	6	1.00	44.79	8.63	16	39	45	51	81.0	
HOMEKIDS	0	1.00	0.72	1.12	0	0	0	1	5.0	
YOJ	454	0.94	10.50	4.09	0	9	11	13	23.0	
TRAVTIME	0	1.00	33.49	15.91	5	22	33	44	142.0	
TIF	0	1.00	5.35	4.15	1	1	4	7	25.0	
CLM_FREQ	0	1.00	0.80	1.16	0	0	0	2	5.0	
MVR_PTS	0	1.00	1.70	2.15	0	0	1	3	13.0	
CAR_AGE	510	0.94	8.33	5.70	-3	1	8	12	28.0	

From the table, we can see that there are 8161 rows and 26 columns in the dataset, two of which are response variables (TARGET_FLAG and TARGET_AMT). Also, we see that there are missing values within the CAR_AGE, AGE, and YOJ columns. In addition, there are 12 columns that are numeric variables and 14 columns that are made up of strings.

Using the glimpse() function, we can see that some of the 14 columns that are made of up string are actually continuous variables like OLDCLAIM, HOME_VAL, INCOME, and BLUEBOOK. We will need to do some data cleaning later on to convert these columns to numeric variables. Also, since some of the categorical

variables have 2 characteristics, we can substitute these observations to 0 and 1 which will make it easier when we build our models. These are the categorical variables that are dichotomous: PARENT1, SEX, MSTATUS, CAR_USE, RED_CAR, REVOKED, URBANICITY.

```
glimpse(data_train)
```

```
## Rows: 8,161
## Columns: 26
## $ INDEX       <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 19, 20, 2~
## $ TARGET_FLAG <int> 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1~
## $ TARGET_AMT  <dbl> 0.000, 0.000, 0.000, 0.000, 0.000, 2946.000, 0.000, 4021.0~
## $ KIDSDRIV    <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ AGE         <int> 60, 43, 35, 51, 50, 34, 54, 37, 34, 50, 53, 43, 55, 53, 45~
## $ HOMEKIDS    <int> 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 0, 3, 0, 3, 2, 1~
## $ YOJ         <int> 11, 11, 10, 14, NA, 12, NA, NA, 10, 7, 14, 5, 11, 11, 0, 1~
## $ INCOME      <chr> "$67,349", "$91,449", "$16,039", "", "$114,986", "$125,301~
## $ PARENT1     <chr> "No", "No", "No", "No", "No", "Yes", "No", "No", "No", "No~
## $ HOME_VAL    <chr> "$0", "$257,252", "$124,191", "$306,251", "$243,925", "$0"~
## $ MSTATUS     <chr> "z_No", "z_No", "Yes", "Yes", "Yes", "z_No", "Yes", "Yes",~
## $ SEX         <chr> "M", "M", "z_F", "M", "z_F", "z_F", "z_F", "M", "z_F", "M"~
## $ EDUCATION   <chr> "PhD", "z_High School", "z_High School", "<High School", "~
## $ JOB         <chr> "Professional", "z_Blue Collar", "Clerical", "z_Blue Colla~
## $ TRAVTIME    <int> 14, 22, 5, 32, 36, 46, 33, 44, 34, 48, 15, 36, 25, 64, 48,~
## $ CAR_USE     <chr> "Private", "Commercial", "Private", "Private", "Private", ~
## $ BLUEBOOK    <chr> "$14,230", "$14,940", "$4,010", "$15,440", "$18,000", "$17~
## $ TIF         <int> 11, 1, 4, 7, 1, 1, 1, 1, 1, 7, 1, 7, 7, 6, 1, 6, 6, 7, 4, ~
## $ CAR_TYPE    <chr> "Minivan", "Minivan", "z_SUV", "Minivan", "z_SUV", "Sports~
## $ RED_CAR     <chr> "yes", "yes", "no", "yes", "no", "no", "no", "yes", "no", ~
## $ OLDCLAIM    <chr> "$4,461", "$0", "$38,690", "$0", "$19,217", "$0", "$0", "$~
## $ CLM_FREQ    <int> 2, 0, 2, 0, 2, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2~
## $ REVOKED     <chr> "No", "No", "No", "No", "Yes", "No", "No", "Yes", "No", "N~
## $ MVR_PTS     <int> 3, 0, 3, 0, 3, 0, 0, 10, 0, 1, 0, 0, 3, 3, 3, 0, 0, 0, 0, ~
## $ CAR_AGE     <int> 18, 1, 10, 6, 17, 7, 1, 7, 1, 17, 11, 1, 9, 10, 5, 13, 16,~
## $ URBANICITY  <chr> "Highly Urban/ Urban", "Highly Urban/ Urban", "Highly Urba~
```

Distribution Plot Let's look at the distribution of all the numeric variables.

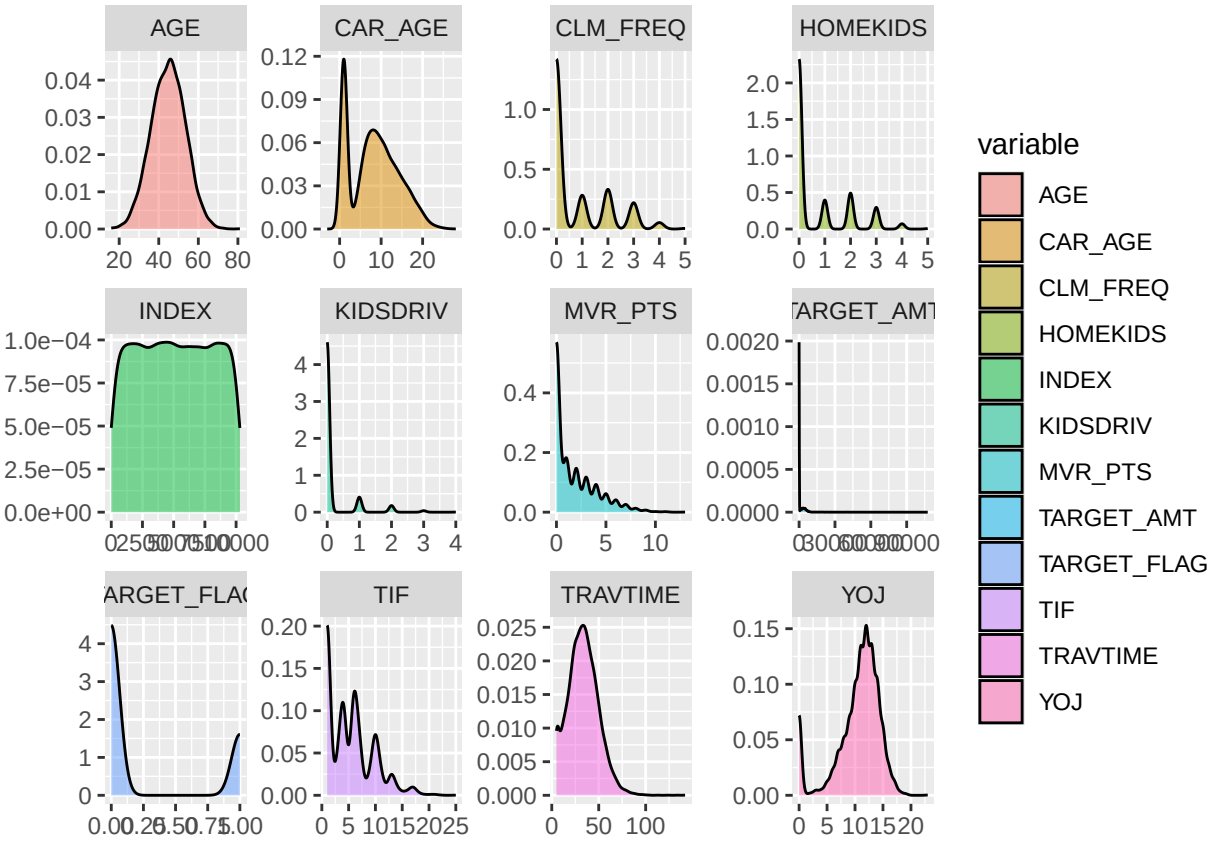
```
numeric_cols <- sapply(data_train, is.numeric)
data_train_numeric <- data_train[, numeric_cols]

non_numeric_cols <- names(data_train)[!numeric_cols]

df_long <- data_train_numeric %>%
  pivot_longer(
    cols = -one_of(non_numeric_cols),
    names_to = "variable",
    values_to = "value"
  )
```

```
## Warning: Unknown columns: `INCOME`, `PARENT1`, `HOME_VAL`, `MSTATUS`, `SEX`,
## `EDUCATION`, `JOB`, `CAR_USE`, `BLUEBOOK`, `CAR_TYPE`, `RED_CAR`, `OLDCLAIM`,
## `REVOKED`, `URBANICITY`
```

```
ggplot(df_long, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~variable, scales ="free", ncol = 4) +
```

```
  labs(x = NULL, y = NULL)
```

```
## Warning: Removed 970 rows containing non-finite outside the scale range
## (`stat_density()`).
```



Looking at the Target_Flag density plot, we see that the TARGET_FLAG plot is skewed to the right, indicating that there are more instances where cars weren't in a crash compared to those where they were. This imbalance can lead to overfitting, where the model may become overly biased towards predicting the majority class (no crash).

To confirm lets see the proportion of 0's to 1's in the TARGET_FLAG column

```
proportion_zeros <- sum(data_train$TARGET_FLAG == 0, na.rm = TRUE) / sum(!is.na(data_train$TARGET_FLAG))
proportion_ones <- 100 - proportion_zeros

cat("Proportion of 0s:", proportion_zeros, "%\n")
```

```
## Proportion of 0s: 73.61843 %
```

```
cat("Proportion of 1s:", proportion_ones, "%\n")
```

```
## Proportion of 1s: 26.38157 %
```

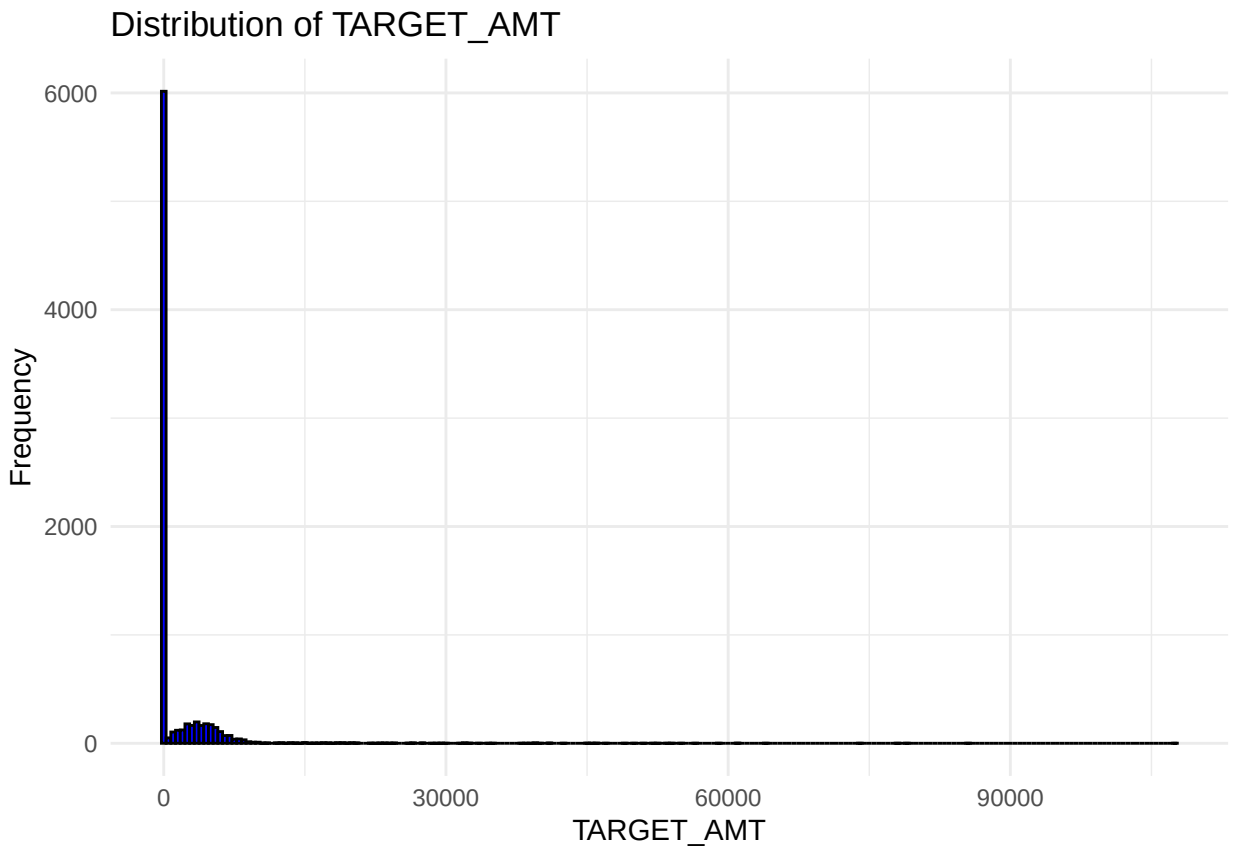
This indicates that 73.6% or 6008 cars were not involved in a car crash while 26.4% or 2153 cars were. Since we have to predict the cost of the car crash let's take a closer look at the TARGET_AMT variable. Let's take out all the 6008 cars that were not involved in a crash and use the describe() from the psych package to look at the summary statistics of the cost.

```
df_filtered <- data_train[data_train$TARGET_AMT != 0,]
describe(df_filtered$TARGET_AMT)
```

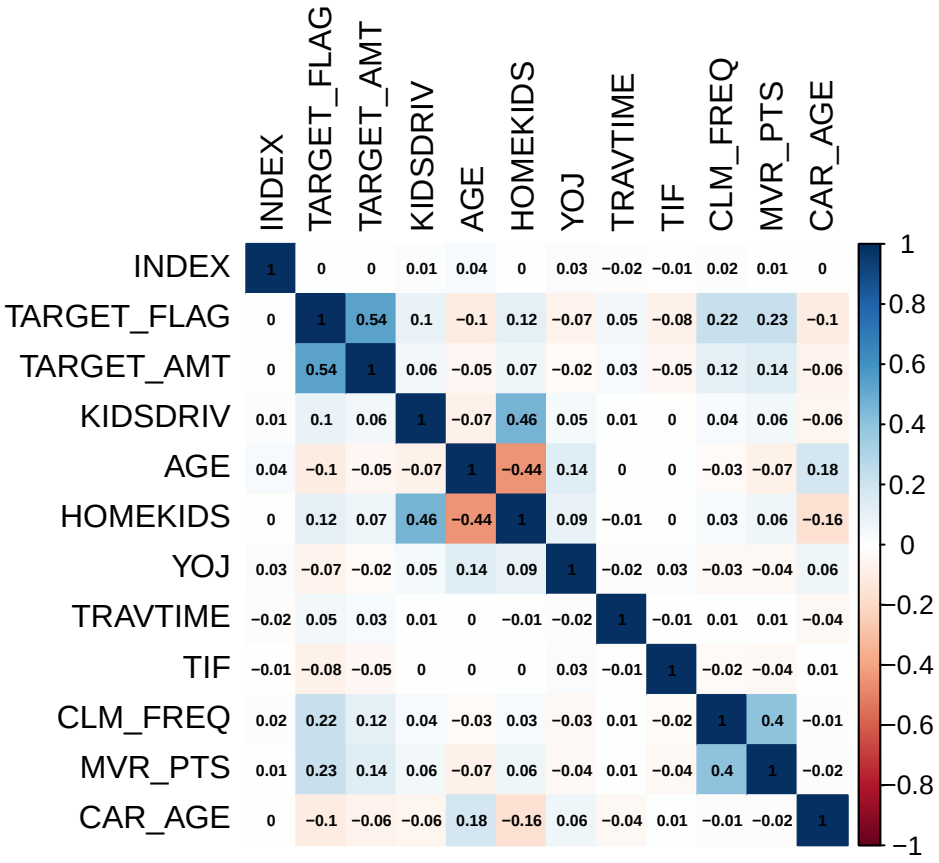
```
##    vars    n    mean      sd median trimmed    mad   min      max    range skew
## X1    1 2153 5702.18 7743.18   4104 4250.81 2317.3 30.28 107586.1 107555.9 5.63
##    kurtosis     se
## X1    42.45 166.88
```

There were 2153 cars involved in a crash. The mean cost of a car crash is $5702.18, the median is $4104 and the standard deviation is $7743.18. The cost ranged from $30.28 to $107586.10. This gives us a better sense of the distribution of the cost of car crashes. Since the mean is greater than the median, we can expect the distribution to be skewed to the right.

```
ggplot(data_train, aes(x=TARGET_AMT)) +
  geom_histogram(binwidth=500, fill="blue", color="black") +
  theme_minimal() +
  labs(title="Distribution of TARGET_AMT", x="TARGET_AMT", y="Frequency")
```




```
numeric_data <- data_train %>%
  select_if(is.numeric)

correlation_matrix <- cor(numeric_data, use = "complete.obs")

corrplot(correlation_matrix, method = "color", tl.col = "black", addCoef.col = "black", number.cex = 0.
```



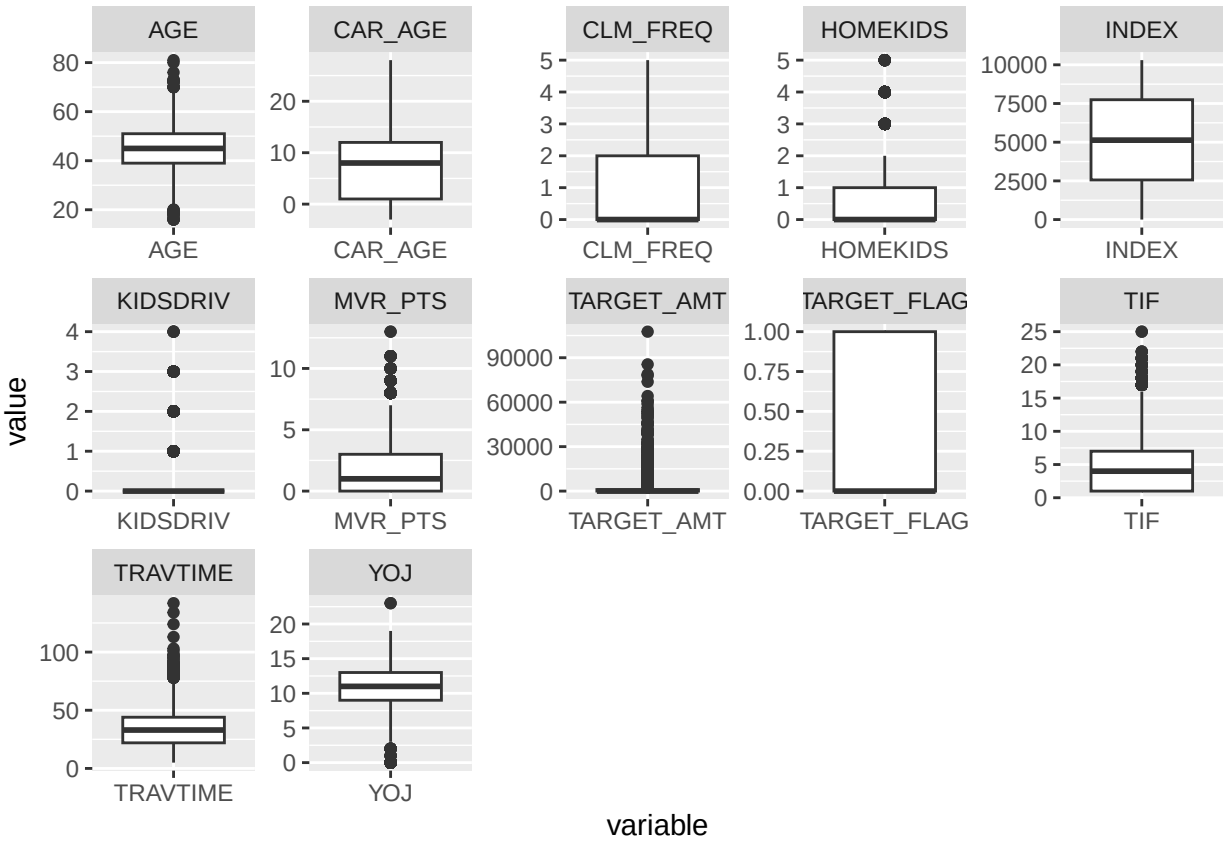
Correlation plot

To identify the correlation between each variable we can create a correlation plot by using the corrplot library. The correlation analysis indicates positive relationships with the TARGET_FLAG variable for the following variables: a moderate positive correlation with TARGET_AMT (0.53) and weak positive correlations with KIDSDRIV (0.10), CLM_FREQ (0.22), and MVR_PTS (0.22).

Box Plot The box plot for the TARGET_AMT column also indicates a highly right-skewed distribution, despite the values ranging from 30 to 90 thousand. This skewness is evident from the compressed appearance of the box plot towards the lower end, suggesting the presence of outliers with exceptionally high values.

```
df_long %>%
  ggplot(aes(variable, value)) +
  geom_boxplot() +
  facet_wrap(~variable, scales='free', ncol=5)
```

```
## Warning: Removed 970 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Data Preperation

Out of the 8161 rows in the dataset, there were 6 rows in the AGE column where there was a missing value. Let's remove rows where the AGE column is NA and also drop index column. The columns CAR_AGE and YOJ had NA values as well but we decided not to remove these rows as the NA values in these columns represented about 6% of the total rows. Removing 6% of the dataset would significantly impact the distribution of the data and our models.

```
missing_values <- data_train %>%
  summarise_all(function(x) sum(is.na(x)))

print(missing_values)
```

```
##   INDEX TARGET_FLAG TARGET_AMT KIDSDRIV AGE HOMEKIDS YOJ INCOME PARENT1
## 1     0           0          0        0   6        0 454      0       0
##   HOME_VAL MSTATUS SEX EDUCATION JOB TRAVTIME CAR_USE BLUEBOOK TIF CAR_TYPE
## 1        0       0   0         0   0        0       0        0   0        0
##   RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS CAR_AGE URBANICITY
## 1       0        0        0       0       0     510          0
```

```
data_train <- data_train %>%
  filter(!is.na(AGE))

data_train <- subset(data_train, select = -INDEX)
```

Next, we removed the dollar sign from the OLDCLAIM, HOME_VAL, INCOME, and BLUEBOOK column. These are numeric values but were represented as strings in the dataset. These continuous variables may be important to the determining whether a car crashes and how much it cost.

```
data_train$OLDCLAIM <- gsub("\\$|,", "", data_train$OLDCLAIM)
data_train$HOME_VAL <- gsub("\\$|,", "", data_train$HOME_VAL)
data_train$INCOME <- gsub("\\$|,", "", data_train$INCOME)
data_train$BLUEBOOK <- gsub("\\$|,", "", data_train$BLUEBOOK)
```

Converting the specified columns to numeric

```
data_train$OLDCLAIM <- as.numeric(data_train$OLDCLAIM)
data_train$HOME_VAL <- as.numeric(data_train$HOME_VAL)
data_train$INCOME <- as.numeric(data_train$INCOME)
data_train$BLUEBOOK <- as.numeric(data_train$BLUEBOOK)

selected_vars <- data_train %>%
  dplyr::select(OLDCLAIM, HOME_VAL, INCOME, BLUEBOOK)

glimpse(selected_vars)
```

```
## Rows: 8,155
## Columns: 4
## $ OLDCLAIM <dbl> 4461, 0, 38690, 0, 19217, 0, 0, 2374, 0, 0, 0, 0, 5028, 0, 0,~
## $ HOME_VAL <dbl> 0, 257252, 124191, 306251, 243925, 0, NA, 333680, 0, 0, 0, 20~
## $ INCOME   <dbl> 67349, 91449, 16039, NA, 114986, 125301, 18755, 107961, 62978~
## $ BLUEBOOK <dbl> 14230, 14940, 4010, 15440, 18000, 17430, 8780, 16970, 11200, ~
```

Some values seem to have invalid values, such as negative numbers, when that is so, we will replace with it with the median values. This will help maintain the distribution of the data.

```
fix_missing <- function(df) {
  df %>%
    mutate_at(vars(c("CAR_AGE", "YOJ", "AGE", "INCOME", "HOME_VAL")), ~ifelse(. < 0, median(., na.rm = T
}

data_train <- fix_missing(data_train)
```

As mentioned before, let's convert these categorical variables PARENT1, SEX, MSTATUS, CAR_USE, RED_CAR, REVOKED, URBANICITY into binary variables.

- `PARENT1` will be 1 if the observation was "YES"
- `SEX` will be 1 if the observation is "M"
- `STATUS` will be 1 if the observation is "Yes"
- `CAR_USE` will be 1 if the observation is "Private"
- `RED_CAR` will be 1 if the observation is "Yes
- `REVOKED` will be 1 if the observation is "Yes"
- `URBANICITY` will be 1 if the observation is "Urban"

```
data_train <- data_train %>%
  mutate(PARENT1 = ifelse(PARENT1 == "Yes", 1, 0),
         SEX = ifelse(SEX == "M", 1, 0),
         MSTATUS = ifelse(MSTATUS == "Yes", 1, 0),
         CAR_USE = ifelse(CAR_USE == "Private", 1, 0),
         RED_CAR = ifelse(RED_CAR == "Yes", 1, 0),
         REVOKED = ifelse(REVOKED == "Yes", 1, 0),
         URBANICITY = ifelse(URBANICITY == "Urban", 1, 0))

print(colnames(data_train))
```

```
##  [1] "TARGET_FLAG" "TARGET_AMT"  "KIDSDRIV"    "AGE"         "HOMEKIDS"
```

```
##  [6] "YOJ"         "INCOME"      "PARENT1"     "HOME_VAL"    "MSTATUS"
## [11] "SEX"         "EDUCATION"   "JOB"         "TRAVTIME"    "CAR_USE"
## [16] "BLUEBOOK"    "TIF"         "CAR_TYPE"    "RED_CAR"     "OLDCLAIM"
## [21] "CLM_FREQ"    "REVOKED"     "MVR_PTS"     "CAR_AGE"     "URBANICITY"
```

```
cate_var <- data_train %>%
  dplyr::select(PARENT1, SEX, MSTATUS, CAR_USE, RED_CAR, REVOKED, URBANICITY)

glimpse(cate_var)
```

```
## Rows: 8,155
## Columns: 7
## $ PARENT1    <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,~
## $ SEX        <dbl> 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0,~
## $ MSTATUS    <dbl> 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,~
## $ CAR_USE    <dbl> 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,~
## $ RED_CAR    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ REVOKED    <dbl> 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
## $ URBANICITY <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

Since we are dealing with binary variables, using the pearson correlation would not be appropriate here as it only measures the linear relationship between two continuous variables. Instead we can use a chi-squared test to determine the association between two categorical variables. The table below measures the association of the variables we just converted to the binary response variable, TARGET_FLAG.

```
variables <- c("PARENT1", "SEX", "MSTATUS", "CAR_USE", "RED_CAR", "REVOKED", "URBANICITY")

results <- data.frame(Variable = character(),
                      Chi_Squared = numeric(),
                      DF = integer(),
                      P_Value = numeric(),
                      stringsAsFactors = FALSE)

# Perform Chi-square test
for (var in variables) {
  contingency_table <- table(data_train[[var]], data_train$TARGET_FLAG)
  test_result <- chisq.test(contingency_table)

  results <- rbind(results, data.frame(Variable = var,
                                       Chi_Squared = test_result$statistic,
                                       DF = test_result$parameter,
                                       P_Value = test_result$p.value))
}

print(results)
```

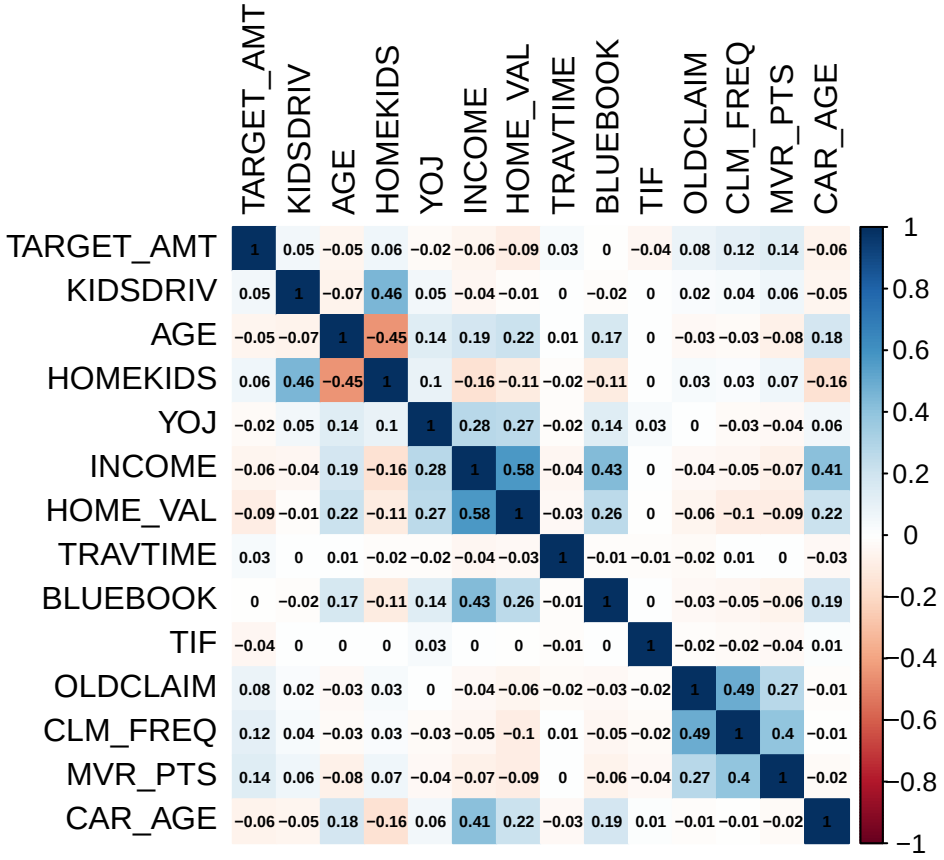
```
##              Variable Chi_Squared DF      P_Value
## X-squared     PARENT1  197.321732  1 8.022490e-45
## X-squared1        SEX    3.515379  1 6.080175e-02
## X-squared2    MSTATUS  146.701710  1 9.118832e-34
## X-squared3    CAR_USE  166.431629  1 4.452272e-38
## X-squared4    RED_CAR 1826.104353  1 0.000000e+00
## X-squared5    REVOKED  187.056050  1 1.396232e-42
## X-squared6 URBANICITY 1826.104353  1 0.000000e+00
```

The table above shows that PARENT1, MSTATUS, CAR_USE, RED_CAR, REVOKED, and URBANIC-

ITY are statistically significant to TARGET_FLAG. SEX has a p-value of 0.0608 suggesting there is not a significant association between SEX and TARGET_FLAG.

Since we changed some columns to continuous variables, let's do another correlation plot but with just the continuous variables.

```
# Identify continuous and binary variables
continuous_vars <- data_train %>%
  select_if(is.numeric) %>%
  select_if(~sum(. %in% 0:1) != length(.))

binary_vars <- data_train %>%
  select_if(is.numeric) %>%
  select_if(~sum(. %in% 0:1) == length(.))

# Compute correlation matrix for continuous variables
correlation_matrix <- cor(continuous_vars, use = "complete.obs")

# Create correlation plot
corrplot(correlation_matrix, method = "color", tl.col = "black", addCoef.col = "black", number.cex = 0.5
```



There does not seem to be any predictors that have a strong linear correlation with TARGET_AMT. MVR_PTS (0.14), CLM_FREQ(0.12), and OLDCLAIM(0.08) seem to have the highest correlation among the continuous predictors.

BUILD MODELS

Model 1 Let's build a model using the variables CLM_FREQ, KIDSDRIV, CAR_AGE, and HOMEKIDS, as they showed higher correlations with the target variable during the correlation analysis.

```
lm_model <- lm(TARGET_AMT ~ CLM_FREQ + KIDSDRIV + CAR_AGE + HOMEKIDS + MVR_PTS + AGE, data = data_train

summary(lm_model)
```

```
##
## Call:
## lm(formula = TARGET_AMT ~ CLM_FREQ + KIDSDRIV + CAR_AGE + HOMEKIDS +
##     MVR_PTS + AGE, data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4666  -1531   -967   -347 104154
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1323.146    346.233   3.822 0.000134 ***
## CLM_FREQ     305.630     50.522   6.049 1.52e-09 ***
## KIDSDRIV     272.054    121.071   2.247 0.024665 *
## CAR_AGE      -40.342      9.594  -4.205 2.64e-05 ***
## HOMEKIDS     123.105     61.243   2.010 0.044456 *
## MVR_PTS      228.105     27.337   8.344  < 2e-16 ***
## AGE           -5.563      7.082  -0.785 0.432198
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4689 on 7638 degrees of freedom
##   (510 observations deleted due to missingness)
## Multiple R-squared:  0.02962,    Adjusted R-squared:  0.02885
## F-statistic: 38.85 on 6 and 7638 DF,  p-value: < 2.2e-16
```

Now let's do a stepwise model where the algorithm chooses the best subset of predictors based on a specified criterion, such as the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC).

```
stepwise_model <- step(lm_model, direction = "both", trace = 0)

summary(stepwise_model)
```

```
##
## Call:
## lm(formula = TARGET_AMT ~ CLM_FREQ + KIDSDRIV + CAR_AGE + HOMEKIDS +
##     MVR_PTS, data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4792  -1530   -966   -365 104122
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1066.845    115.778   9.215  < 2e-16 ***
## CLM_FREQ     305.445     50.521   6.046 1.56e-09 ***
## KIDSDRIV     256.425    119.422   2.147  0.03181 *
```

```
## CAR_AGE      -41.256      9.523  -4.332 1.49e-05 ***
## HOMEKIDS     144.599     54.788   2.639  0.00833 **
## MVR_PTS      229.269     27.296   8.399  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4689 on 7639 degrees of freedom
##   (510 observations deleted due to missingness)
## Multiple R-squared:  0.02954,    Adjusted R-squared:  0.0289
## F-statistic:  46.5 on 5 and 7639 DF,  p-value: < 2.2e-16
```

lm_model (with AGE, Model 1):

Coefficients for CLM_FREQ, KIDSDRIV, CAR_AGE, HOMEKIDS, and MVR_PTS are all significant ($p < 0.05$), impacting TARGET_AMT significantly. AGE coefficient is not significant ($p = 0.432$), suggesting no linear relationship with TARGET_AMT. Adjusted R-squared: 0.02885, explaining about 2.9% of variance in TARGET_AMT. stepwise_model (without AGE, Model 2):

Coefficients for CLM_FREQ, KIDSDRIV, CAR_AGE, HOMEKIDS, and MVR_PTS remain significant. Removing AGE didn't affect other coefficients significantly. Adjusted R-squared: 0.0289, slightly lower than Model 1. Comparing both, they share significant predictors, and removing AGE didn't alter coefficients much. Hence, Model 2, simpler yet potent, might be preferable. Regarding coefficients, CLM_FREQ, KIDSDRIV, CAR_AGE, HOMEKIDS, and MVR_PTS positively correlate with TARGET_AMT, aligning with expectations.

Model 2 First, we are going to create a binary logistic regression model to predict the probability that a person will crash their car. Based on the model 1, we can see that CLM_FREQ, MVR_PTS, and KIDSDRIV have the lowest p-value and are significant predictors of TARGET_FLAG. Also, we will use MSTATUS, CAR_USE, and REVOKED since they were all showed statistical significance with TARGET_FLAG in the chi-squared test. And, intuitively, these binary variables increase the probability of a car crash.

```
model2 <- glm(TARGET_FLAG ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS + CAR_USE  + REVOKED,
              data = data_train, family = binomial)

summary(model2)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS +
##     CAR_USE + REVOKED, family = binomial, data = data_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.06125    0.05928 -17.902   <2e-16 ***
## CLM_FREQ     0.26489    0.02336  11.341   <2e-16 ***
## MVR_PTS      0.14644    0.01260  11.624   <2e-16 ***
## KIDSDRIV     0.40433    0.04795   8.432   <2e-16 ***
## MSTATUS     -0.58283    0.05401 -10.790   <2e-16 ***
## CAR_USE     -0.60461    0.05420 -11.155   <2e-16 ***
## REVOKED      0.85418    0.07392  11.555   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9404.0  on 8154  degrees of freedom
```

```
## Residual deviance: 8432.7  on 8148  degrees of freedom
## AIC: 8446.7
##
## Number of Fisher Scoring iterations: 4
```

This model shows that all the predictor variables are statistically significant. The AIC for the model is 8446.7. Next, let's use the stepAIC function to improve this model.

```
step_model <- stepAIC(model2, direction = "both", trace = FALSE)
summary(step_model)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS +
##     CAR_USE + REVOKED, family = binomial, data = data_train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.06125    0.05928 -17.902   <2e-16 ***
## CLM_FREQ     0.26489    0.02336  11.341   <2e-16 ***
## MVR_PTS      0.14644    0.01260  11.624   <2e-16 ***
## KIDSDRIV     0.40433    0.04795   8.432   <2e-16 ***
## MSTATUS     -0.58283    0.05401 -10.790   <2e-16 ***
## CAR_USE     -0.60461    0.05420 -11.155   <2e-16 ***
## REVOKED      0.85418    0.07392  11.555   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9404.0  on 8154  degrees of freedom
## Residual deviance: 8432.7  on 8148  degrees of freedom
## AIC: 8446.7
##
## Number of Fisher Scoring iterations: 4
```

After using a stepwise selection process on model2, the results, specifically the AIC, are identical. This indicates that adding or removing predictors did not improve the original model. Again, all of the predictors have a p-value less than, 0.05 indicating they are statistically significant in predicting TARGET_FLAG. The intercept when the predictors are zero is -1.644. For the coefficients, a one unit increase in CLM_FREQ or claims processed corresponds to an increases of 0.264 in the log-odds of car crashes. All the coefficients seem to cause a positive increase in log-odds of car crashes except for CAR_USE. When CAR_USE is 1 (private vehicle), the log odds of a car crash is expected to decrease by 0.60461 compared to commercial car use. Intuitively, this makes sense as one would expect commercial vehicles to be involved more car crashes than private vehicles. The predictor REVOKED seems to cause the highest increase in log odds of car crash at 0.85418. This makes sense as you would expect someone who has had their license revoked be in a car crash. One coefficient estimate that does not seem to make sense is MSTATUS as one would expect married people to be safer drivers and be involved in less crashes. However, the model predicts that individuals who are not married are expected to increase log odds of a car crash by 0.582 compared to individuals who are married.

Next, let's create a multiple linear regression model to predict the amount of money it will cost if the person does crash their car. We will use the same predictors as before with the binary model since they were all statistically significant in predicting TARGET_FLAG.

```r
model2_linear <- lm(TARGET_AMT ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS + CAR_USE + REVOKED, data = da

summary(model2_linear)
```

```
##
## Call:
## lm(formula = TARGET_AMT ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS +
##     CAR_USE + REVOKED, data = data_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5082  -1657   -917   -179 104324
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1740.83     119.76  14.536  < 2e-16 ***
## CLM_FREQ      246.90      48.21   5.122 3.10e-07 ***
## MVR_PTS       216.90      25.98   8.350  < 2e-16 ***
## KIDSDRIV      451.31     100.09   4.509 6.60e-06 ***
## MSTATUS      -737.74     104.65  -7.049 1.94e-12 ***
## CAR_USE      -824.24     106.08  -7.770 8.80e-15 ***
## REVOKED       660.78     156.23   4.229 2.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4609 on 8148 degrees of freedom
## Multiple R-squared:  0.04134,    Adjusted R-squared:  0.04064
## F-statistic: 58.56 on 6 and 8148 DF,  p-value: < 2.2e-16
```

Like the binary model, all of the coefficients in this multiple linear regression model have a p-value less than 0.05, indicating they are statistically significant. The intercept is 1740.8, which means that when all predictor variables are zero the expect cost of a car crash is $1740.8. When MSTATUS and CAR_USE changes from 0 to 1, this causes a decrease of 737.74 and 824.24 in cost of the car crash. This makes sense as married people generally drive safer and private vehicles are not driven as often as commercial vehicles. All of the other coefficients drive the cost of car crashes up with a one unit increase.

Overall, the p-value for the model is less than 0.05, indicating it is statistically significant in predicting TARGET_AMT. However, the R-squared for this model is 0.04134, suggesting that only 4% of the variability in TARGET_AMT can be explained by the model. Although this is very low, it is higher than the R-squared of the first model(0.02). Therefore, it may be advisable to retain the model for further analysis and refinement.

```r
# Fit weighted least squares to address heteroscedasticity
wls_model <- lm(TARGET_AMT ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS + CAR_USE, data = data_train, weigh
summary(wls_model)
```

```
##
## Call:
## lm(formula = TARGET_AMT ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS +
##     CAR_USE, data = data_train, weights = 1/abs(resid(model2_linear))^2)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -20.679  -1.002  -0.874  -0.493  47.425
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1743.759      7.115  245.08   <2e-16 ***
## CLM_FREQ     237.678      3.230   73.59   <2e-16 ***
## MVR_PTS      239.595      1.474  162.56   <2e-16 ***
## KIDSDRIV     474.660      6.113   77.65   <2e-16 ***
## MSTATUS     -867.017      4.859 -178.45   <2e-16 ***
## CAR_USE     -788.574      6.554 -120.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.32 on 8149 degrees of freedom
## Multiple R-squared:  0.9558, Adjusted R-squared:  0.9557
## F-statistic: 3.521e+04 on 5 and 8149 DF,  p-value: < 2.2e-16
```

LOGISTIC REGRESSIONS MODEL 2 manually selected variables based on domain knowledge, focusing on factors expected to have a significant impact on the likelihood of car crashes. Also, incorporated a weighted loss function to address class imbalance. By assigning a higher weight to the minority class (TARGET_FLAG = 1), I give more importance to correctly predicting instances of car crashes, thereby mitigating the impact of class imbalance.

The negative coefficient associated with age suggests that older drivers are less likely to be involved in car crashes, aligning with the common understanding that age correlates with driving experience and caution. Surprisingly, commercial vehicle use is associated with a lower probability of car crashes, contrary to expectations. However, this could be attributed to differences in driving behavior or regulations for commercial vehicles. The positive coefficients for variables such as CLM_FREQ and MVR_PTS indicate that individuals with a history of claims and traffic violations are more likely to be involved in car crashes, reflecting risky driving behavior. Interestingly, the coefficient for HOME_VAL, while statistically significant due to the large sample size, has a negligible practical significance. Overall, the model captures meaningful relationships between variables and the likelihood of car crashes, despite some counterintuitive coefficients.

```r
# Model 2: Manual Variable Selection with Weighted Loss Function
model_glm1 <- glm(TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + MVR_PTS + HOME_VAL, data = data_train, family
summary(model_glm1)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ AGE + CAR_USE + CLM_FREQ + MVR_PTS +
##     HOME_VAL, family = "binomial", data = data_train, weights = ifelse(data_train$TARGET_FLAG ==
##     1, 3, 1))
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.072e+00  1.053e-01  10.181  < 2e-16 ***
## AGE         -1.612e-02  2.273e-03  -7.092 1.32e-12 ***
## CAR_USE     -6.134e-01  4.066e-02 -15.087  < 2e-16 ***
## CLM_FREQ     2.918e-01  1.798e-02  16.226  < 2e-16 ***
## MVR_PTS      1.351e-01  9.721e-03  13.898  < 2e-16 ***
## HOME_VAL    -3.124e-06  1.656e-07 -18.867  < 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: 16273  on 7692  degrees of freedom
## Residual deviance: 14627  on 7687  degrees of freedom
```

```
##   (462 observations deleted due to missingness)
## AIC: 14639
##
## Number of Fisher Scoring iterations: 5
```

MODEL 3 I chose a subset of variables based on expert knowledge and relevance to the task. This approach allowed to include only the most important predictors while reducing the risk of overfitting. Class imbalance was handled by undersampling the majority class (TARGET_FLAG = 0). This technique reduces the dominance of the majority class in the training data, making the model less biased towards predicting the majority class and improving its ability to capture patterns in the minority class.

he negative coefficient for CAR_AGE suggests that older vehicles are associated with a lower probability of accidents, possibly due to better safety features and maintenance. Similar to Model 2, commercial vehicle use is linked to a reduced likelihood of car crashes, which may be attributed to differences in driving behavior or regulations. Consistent with Model 2, variables such as CLM_FREQ and MVR_PTS exhibit positive coefficients, indicating that a history of claims and traffic violations increases the probability of car crashes. Additionally, the coefficients for variables such as SEX and REVOKED provide insights into demographic and regulatory factors influencing driving behavior and accident risk. Despite potential counterintuitive findings, the model retains its predictive power and relevance for the task at hand.

```r
# Undersample the majority class
set.seed(12)
train_undersampled <- ovun.sample(TARGET_FLAG ~ ., data = data_train, method = "under", N = 3500)$data

# Model 3: Expert-Driven Selection with Undersampling
model_glm3 <- glm(TARGET_FLAG ~ CAR_AGE + CAR_USE + CLM_FREQ + MVR_PTS + SEX + REVOKED, data = train_un
summary(model_glm3)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ CAR_AGE + CAR_USE + CLM_FREQ + MVR_PTS +
##     SEX + REVOKED, family = "binomial", data = train_undersampled)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.11194    0.09642   1.161    0.246
## CAR_AGE     -0.04371    0.00644  -6.788 1.14e-11 ***
## CAR_USE     -0.61162    0.07747  -7.895 2.91e-15 ***
## CLM_FREQ     0.29133    0.03236   9.002  < 2e-16 ***
## MVR_PTS      0.14934    0.01759   8.491  < 2e-16 ***
## SEX         -0.30773    0.07658  -4.018 5.86e-05 ***
## REVOKED      0.88955    0.10689   8.322  < 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: 4849.5  on 3499  degrees of freedom
## Residual deviance: 4371.1  on 3493  degrees of freedom
## AIC: 4385.1
##
## Number of Fisher Scoring iterations: 4
```

PART 4. SELECTING MODEL Preprocessing step used in training data for eval data

```r
data_eval$OLDCLAIM <- gsub("\\$|,", "", data_eval$OLDCLAIM)
data_eval$HOME_VAL <- gsub("\\$|,", "", data_eval$HOME_VAL)
data_eval$INCOME <- gsub("\\$|,", "", data_eval$INCOME)
data_eval$BLUEBOOK <- gsub("\\$|,", "", data_eval$BLUEBOOK)

#Converting the specified columns to numeric

data_eval$OLDCLAIM <- as.numeric(data_eval$OLDCLAIM)
data_eval$HOME_VAL <- as.numeric(data_eval$HOME_VAL)
data_eval$INCOME <- as.numeric(data_eval$INCOME)
data_eval$BLUEBOOK <- as.numeric(data_eval$BLUEBOOK)
```

```r
fix_missing <- function(df) {
  df %>%
    mutate_at(vars(c("CAR_AGE", "YOJ", "AGE", "INCOME", "HOME_VAL")), ~ifelse(. < 0, median(., na.rm = T
}

data_eval <- fix_missing(data_eval)
```

```r
data_eval <- data_eval %>%
  mutate(PARENT1 = ifelse(PARENT1 == "Yes", 1, 0),
         SEX = ifelse(SEX == "M", 1, 0),
         MSTATUS = ifelse(MSTATUS == "Yes", 1, 0),
         CAR_USE = ifelse(CAR_USE == "Private", 1, 0),
         RED_CAR = ifelse(RED_CAR == "Yes", 1, 0),
         REVOKED = ifelse(REVOKED == "Yes", 1, 0),
         URBANICITY = ifelse(URBANICITY == "Urban", 1, 0))
```

#MULTIPLE LINEAR REGRESSION

```r
# Predicting the target variable using Model 1
predictions_model1 <- predict(lm_model, newdata = data_train)

# Calculating Mean Squared Error (MSE) for Model 1
mse_model1 <- mean((data_train$TARGET_AMT - predictions_model1)^2)
print(paste("Multiple linear 1 - Mean Squared Error (MSE):", mse_model1))
```

```
## [1] "Multiple linear 1 - Mean Squared Error (MSE): NA"
```

```r
# Calculating R-squared (R2) for Model 1
r_squared_model1 <- summary(lm_model)$r.squared
print(paste("Multiple linear 1 - R-squared (R2):", r_squared_model1))
```

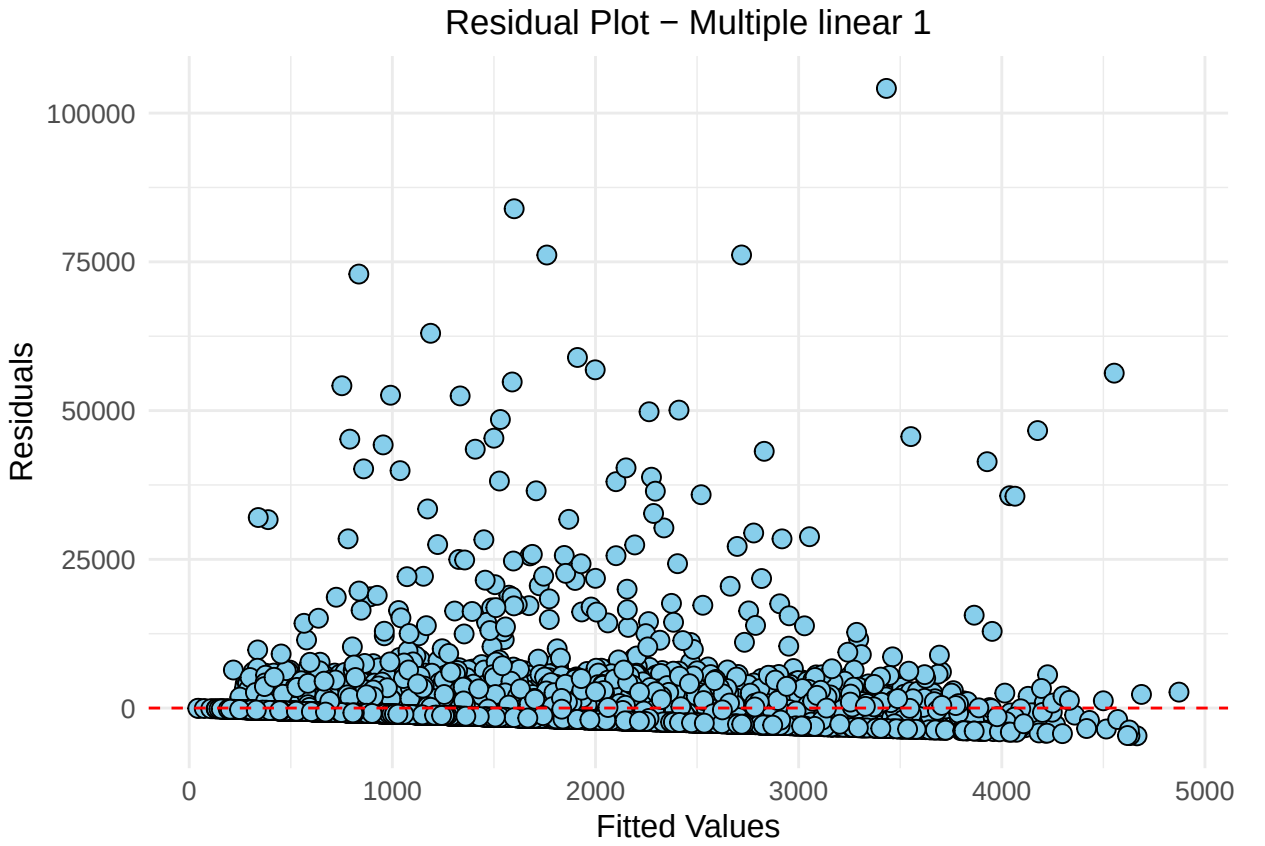
```
## [1] "Multiple linear 1 - R-squared (R2): 0.0296153596654112"
```

```r
# Extracting F-statistic for Model 1
f_statistic_model1 <- summary(lm_model)$fstatistic[1]
print(paste("Multiple linear 1 - F-statistic:", f_statistic_model1))
```

```
## [1] "Multiple linear 1 - F-statistic: 38.8509373366312"
```

```r
# Calculate residuals for Model 1
residuals_model1 <- residuals(lm_model)
fitted_values_model1 <- fitted(lm_model)

# Create a data frame for plotting
residuals_df <- data.frame(Residuals = residuals_model1, Fitted_Values = fitted_values_model1)
```

```r
# Create the residual plot
residual_plot_model1 <- ggplot(residuals_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point(shape = 21, size = 3, fill = "skyblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot - Multiple linear 1", x = "Fitted Values", y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

# Display the residual plot for Model 1
print(residual_plot_model1)
```




```r
# Predicting the target variable using Model 2 (Stepwise Model)
predictions_model2 <- predict(stepwise_model, newdata = data_train)

# Calculating Mean Squared Error (MSE) for Model 2 (Stepwise Model)
mse_model2 <- mean((data_train$TARGET_AMT - predictions_model2)^2)
print(paste("Multiple linear 2 - Mean Squared Error (MSE):", mse_model2))
```

```
## [1] "Multiple linear 2 - Mean Squared Error (MSE): NA"
```

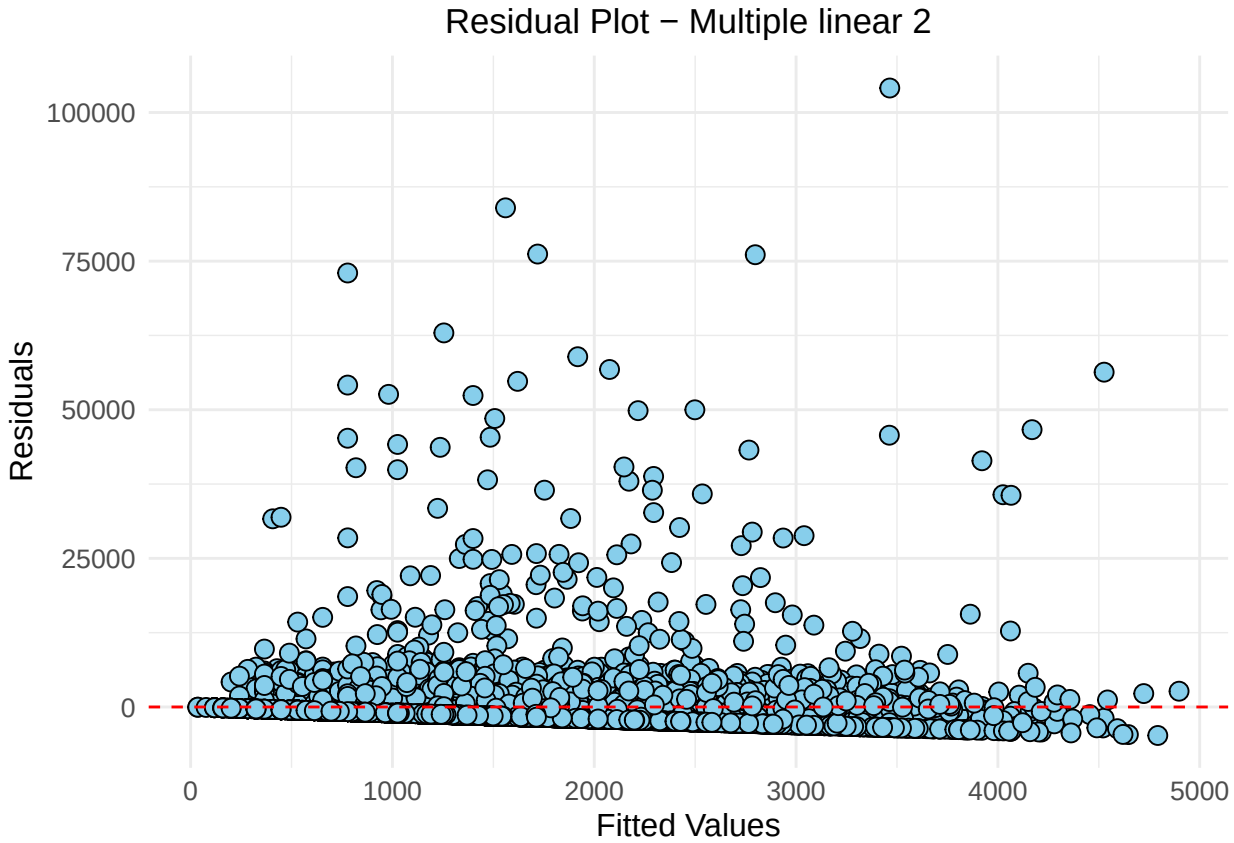
```r
# Calculating R-squared (R2) for Model 2 (Stepwise Model)
r_squared_model2 <- summary(stepwise_model)$r.squared
print(paste("Multiple linear 2 - R-squared (R2):", r_squared_model2))
```

```
## [1] "Multiple linear 2 - R-squared (R2): 0.0295369749014077"
```

```r
# Extracting F-statistic for Model 2 (Stepwise Model)
f_statistic_model2 <- summary(stepwise_model)$fstatistic[1]
print(paste("Multiple linear 2 - F-statistic:", f_statistic_model2))
```

```
## [1] "Multiple linear 2 - F-statistic: 46.5000614008825"
```

```r
# Calculate residuals for Model 2
residuals_model2 <- residuals(stepwise_model)
fitted_values_model2 <- fitted(stepwise_model)

# Create a data frame for plotting
residuals_df <- data.frame(Residuals = residuals_model2, Fitted_Values = fitted_values_model2)

# Create the residual plot
residual_plot_model2 <- ggplot(residuals_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point(shape = 21, size = 3, fill = "skyblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot - Multiple linear 2", x = "Fitted Values", y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

# Display the residual plot for Model 1
print(residual_plot_model2)
```



Among the multiple linear regression models, Multiple linear 3 has the highest R-squared value (0.0413),

indicating that it explains a slightly larger proportion of the variance in the dependent variable compared to the other models. Additionally, it has the highest F-statistic (58.563), suggesting that the overall model fit is better than the other models. Therefore, Multiple linear 3 would be the preferred linear regression model based on these metrics. However, the R-squared value of 0.0413 for Multiple linear 3 indicates that approximately 4.13% of the total variance in the dependent variable (TARGET_AMT) is explained by the independent variables included in the model which is very low. Hence, I decided to validate maodel assumptions in order to arrive to better results.

```
# Predicting the target variable using Model 3
predictions_model3 <- predict(model2_linear, newdata = data_train)

# Calculating Mean Squared Error (MSE) for Model 2
mse_model3 <- mean((data_train$TARGET_AMT - predictions_model3)^2)
print(paste("Multiple linear 3 - Mean Squared Error (MSE):", mse_model3))
```

```
## [1] "Multiple linear 3 - Mean Squared Error (MSE): 21220398.4745726"
```

```
# Calculating R-squared (R2) for Model 2
r_squared_model3 <- summary(model2_linear)$r.squared
print(paste("Multiple linear 3 - R-squared (R2):", r_squared_model3))
```

```
## [1] "Multiple linear 3 - R-squared (R2): 0.041341760187715"
```

```
# Extracting F-statistic for Model 2
f_statistic_model3 <- summary(model2_linear)$fstatistic[1]
print(paste("Multiple linear 3 - F-statistic:", f_statistic_model3))
```

```
## [1] "Multiple linear 3 - F-statistic: 58.5632167996702"
```

```
# Calculate residuals for Model 3
residuals_model3 <- residuals(model2_linear)
fitted_values_model3 <- fitted(model2_linear)

# Create a data frame for plotting
residuals_df <- data.frame(Residuals = residuals_model3, Fitted_Values = fitted_values_model3)

# Create the residual plot
residual_plot_model3 <- ggplot(residuals_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point(shape = 21, size = 3, fill = "skyblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot - Multiple linear 3", x = "Fitted Values", y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

# Display the residual plot for Model 1
print(residual_plot_model3)
```

By thoroughly validating the assumptions of the linear regression model and addressing any detected issues, such as multicollinearity, non-linearity, heteroscedasticity, and non-normality of residuals, the resulting model is likely to yield better and more reliable results. By ensuring that the model conforms to the underlying assumptions of linear regression, we increase our confidence in its predictive capabilities and the validity of the estimated coefficients. Addressing issues like multicollinearity helps to stabilize parameter estimates, while ensuring linearity and homoscedasticity enhances the model's ability to accurately capture the relationships between the predictor variables and the target variable. Furthermore, employing weighted least squares (WLS) to account for non-constant variance of residuals can improve the precision of parameter estimates and mitigate the impact of outliers.

With all VIF values well below 10, it appears that multicollinearity is not a major issue in this model. Therefore, the predictor variables are likely contributing unique information to the regression model without substantial redundancy or overlap.

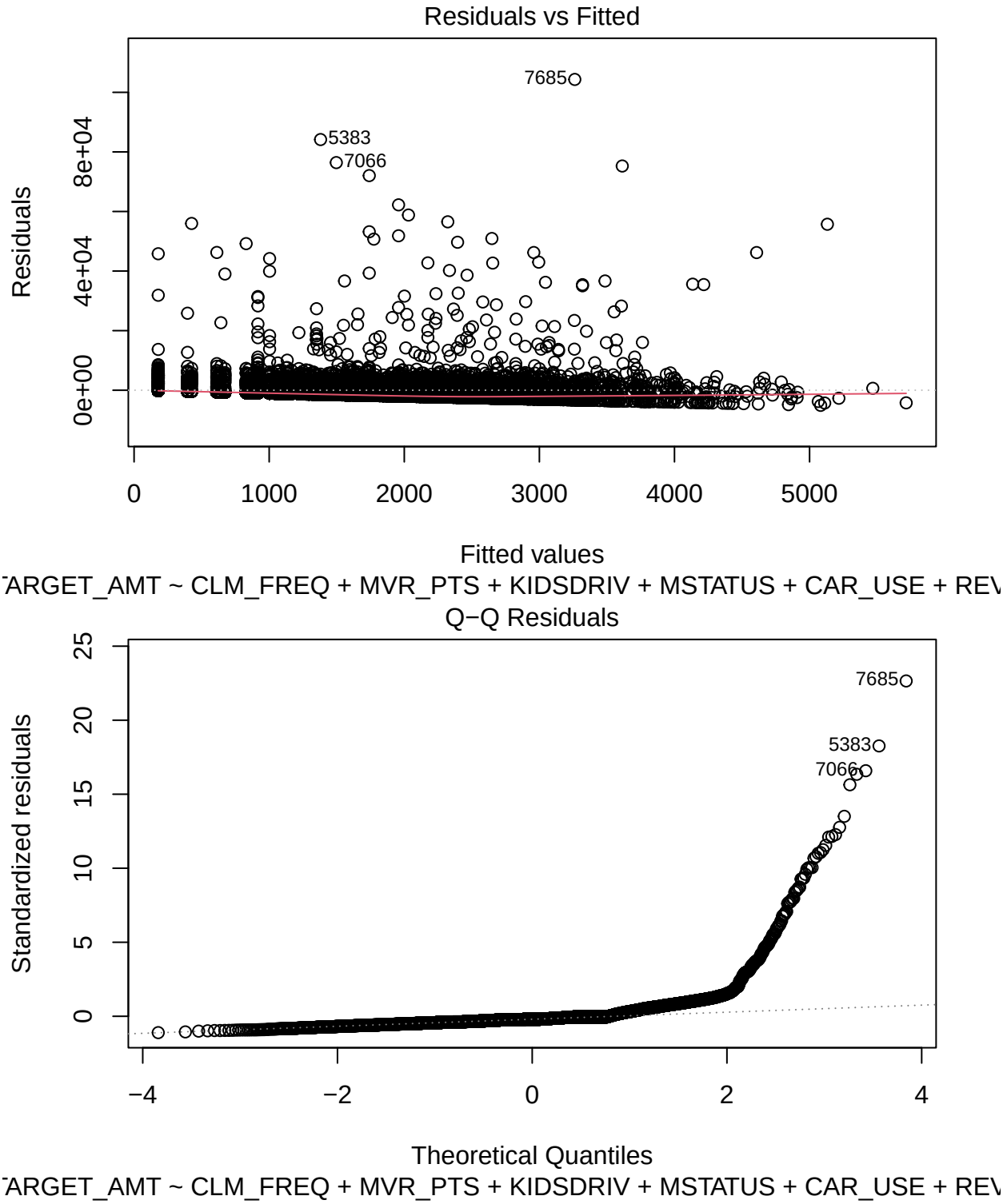
```
# Calculate VIFs
vif_values <- car::vif(model2_linear)

# Check VIF values
vif_values
```

```
## CLM_FREQ  MVR_PTS KIDSDRIV  MSTATUS  CAR_USE  REVOKED
## 1.196730 1.193183 1.007091 1.009363 1.008609 1.007475
```

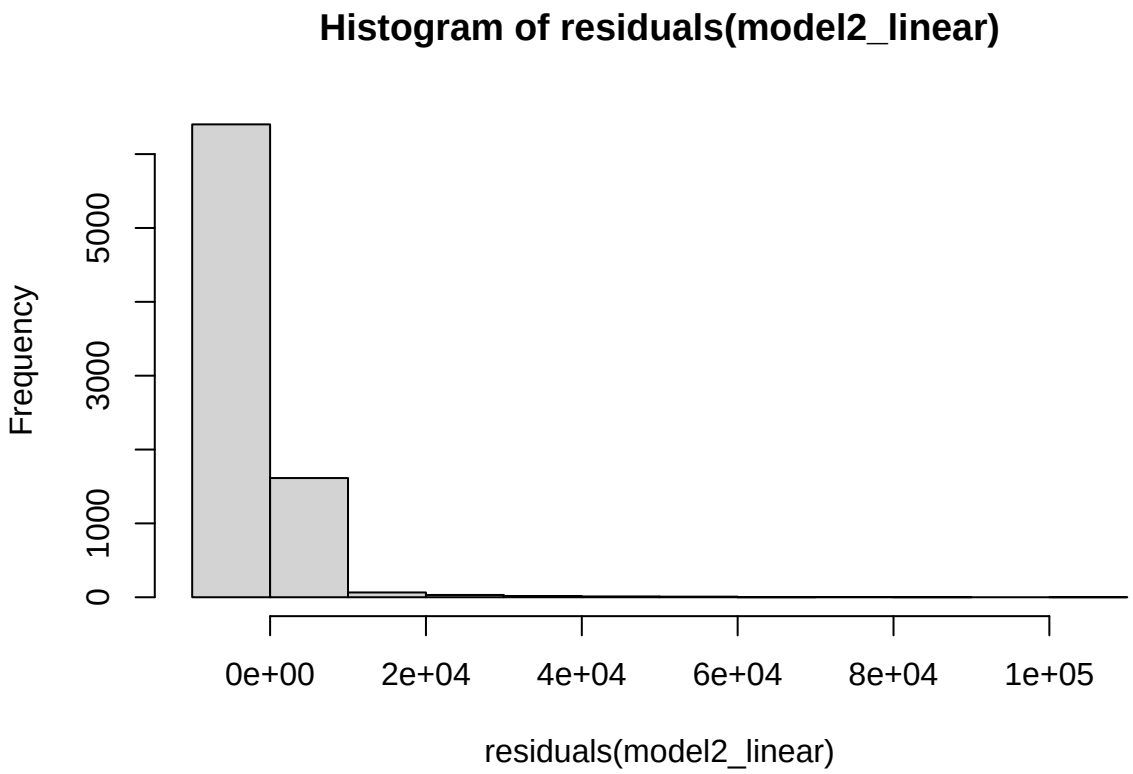
The residuals mostly follow the diagonal line, suggesting that the normality assumption is reasonably met, except for a few large positive residuals at the upper end showing potential outliers and a heavy-tailed residual distribution. The histogram distribution is skewed to the right, with most residuals concentrated around zero but a few very large positive residuals. This confirms the presence of potential outliers and heavy tails in the residual distribution, deviating from the normality assumption. The residuals are randomly scattered around zero for most of the fitted values, but there are a few very large positive residuals at the upper end

of the fitted values, hence, heteroscedasticity issues for those high fitted value ranges, violating the linearity and homoscedasticity assumptions.

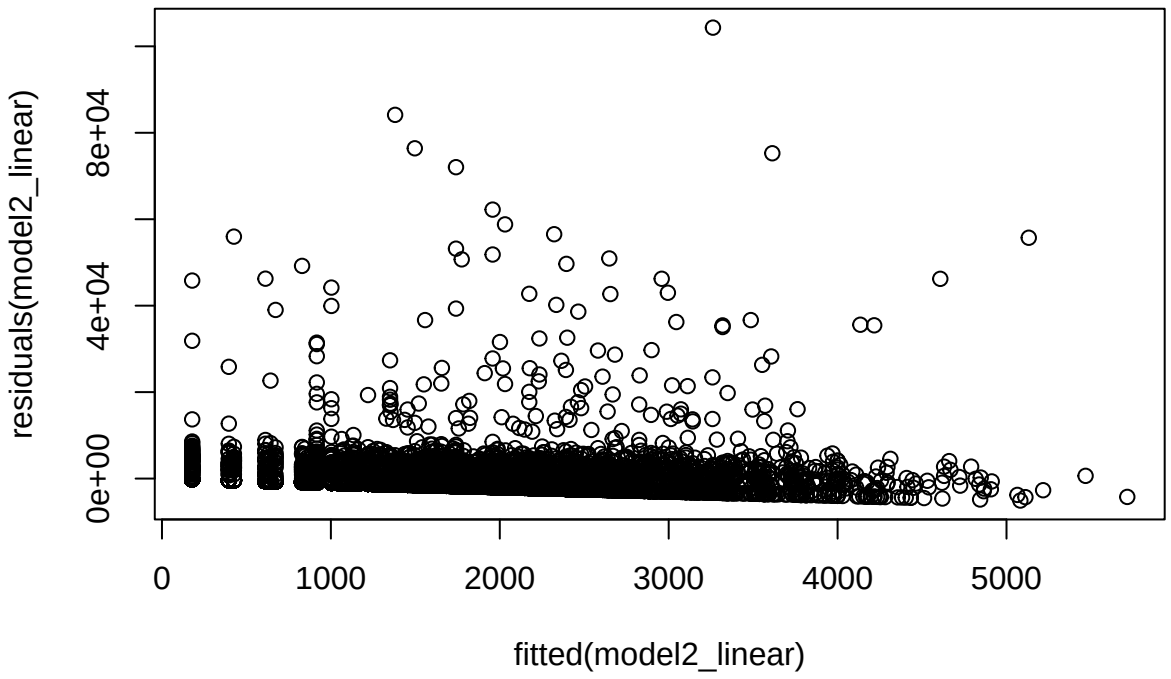
```
# Check for linearity
plot(model2_linear, which = c(1, 2))
```



```r
# Check for normality of residuals
hist(residuals(model2_linear))
```




```r
# Check for homoscedasticity
plot(fitted(model2_linear), residuals(model2_linear))
```



IMPROVEMENT handling heteroscedasticity issues

```r
# Fit weighted least squares to address heteroscedasticity
wls_model <- lm(TARGET_AMT ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS + CAR_USE,
                data = data_train, weights = 1/abs(resid(model2_linear))^2)

summary(wls_model)
```

```
##
## Call:
## lm(formula = TARGET_AMT ~ CLM_FREQ + MVR_PTS + KIDSDRIV + MSTATUS +
##     CAR_USE, data = data_train, weights = 1/abs(resid(model2_linear))^2)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -20.679  -1.002  -0.874  -0.493  47.425
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1743.759      7.115  245.08   <2e-16 ***
## CLM_FREQ     237.678      3.230   73.59   <2e-16 ***
## MVR_PTS      239.595      1.474  162.56   <2e-16 ***
## KIDSDRIV     474.660      6.113   77.65   <2e-16 ***
## MSTATUS     -867.017      4.859 -178.45   <2e-16 ***
## CAR_USE     -788.574      6.554 -120.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.32 on 8149 degrees of freedom
## Multiple R-squared:  0.9558, Adjusted R-squared:  0.9557
## F-statistic: 3.521e+04 on 5 and 8149 DF,  p-value: < 2.2e-16
```

IMPROVED MODEL RESULTS After fitting the WLS model, significant improvements were observed in various performance metrics. The R-squared value increased substantially to 0.9558, indicating that approximately 95.58% of the total variance in the target variable (TARGET_AMT) is explained by the independent variables in the model. Additionally, both the Mean Squared Error (MSE) and the F-statistic showed improvements, further confirming the enhanced predictive accuracy and goodness of fit of the WLS model compared to the original linear regression model.

```r
# Predicting the target variable using Model 4
predictions_model4 <- predict(wls_model, newdata = data_train)

# Calculating Mean Squared Error (MSE) for Model 4
mse_model4 <- mean((data_train$TARGET_AMT - predictions_model4)^2)
print(paste("Multiple linear 4 - Mean Squared Error (MSE):", mse_model4))
```

```
## [1] "Multiple linear 4 - Mean Squared Error (MSE): 21281370.5555677"
```

```r
# Calculating R-squared (R2) for Model 2
r_squared_model4 <- summary(wls_model)$r.squared
print(paste("Multiple linear 4 - R-squared (R2):", r_squared_model4))
```

```
## [1] "Multiple linear 4 - R-squared (R2): 0.955754824373881"
```

```r
# Extracting F-statistic for Model 2
f_statistic_model4 <- summary(wls_model)$fstatistic[1]
print(paste("Multiple linear 4 - F-statistic:", f_statistic_model4))
```

```
## [1] "Multiple linear 4 - F-statistic: 35205.8544399809"
```

```r
# Calculate residuals for Model 3
residuals_model4 <- residuals(wls_model)
fitted_values_model4 <- fitted(wls_model)

# Create a data frame for plotting
residuals_df <- data.frame(Residuals = residuals_model3, Fitted_Values = fitted_values_model4)

# Create the residual plot
residual_plot_model3 <- ggplot(residuals_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point(shape = 21, size = 3, fill = "skyblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot - Multiple linear 4", x = "Fitted Values", y = "Residuals") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

# Display the residual plot for Model 1
print(residual_plot_model3)
```

LOGISTIC REGRESSION

```r
#LOGISTIC MODEL 1
# Predicting the target variable using the binary logistic regression model
logistic_predictions <- predict(step_model, newdata = data_train, type = "response")

# Converting probabilities to binary predictions
```

```r
binary_predictions <- ifelse(logistic_predictions > 0.5, 1, 0)

# Confusion matrix
conf_matrix <- table(data_train$TARGET_FLAG, binary_predictions)

# Displaying Confusion Matrix
print("Logistic Model 1 Confusion Matrix:")
```

```
## [1] "Logistic Model 1 Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
##    binary_predictions
##        0    1
##   0 5690  317
##   1 1678  470
```

```r
# Accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Logistic Model 1 Accuracy:", accuracy))
```

```
## [1] "Logistic Model 1 Accuracy: 0.755364806866953"
```

```r
# Classification Error Rate
error_rate <- 1 - accuracy
print(paste("Logistic Model 1 Classification Error Rate:", error_rate))
```

```
## [1] "Logistic Model 1 Classification Error Rate: 0.244635193133047"
```

```r
# Confusion Matrix
true_negatives <- 5690
false_positives <- 1678
false_negatives <- 317
true_positives <- 470

# Precision
precision <- true_positives / (true_positives + false_positives)
print(paste("Logistic Model 1 Precision:", precision))
```

```
## [1] "Logistic Model 1 Precision: 0.218808193668529"
```

```r
# Sensitivity (True Positive Rate)
sensitivity <- true_positives / (true_positives + false_negatives)
print(paste("Logistic Model 1 Sensitivity (True Positive Rate):", sensitivity))
```

```
## [1] "Logistic Model 1 Sensitivity (True Positive Rate): 0.59720457433291"
```

```r
# Specificity (True Negative Rate)
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
print(paste("Logistic Model 1 Specificity (True Negative Rate):", specificity))
```

```
## [1] "Logistic Model 1 Specificity (True Negative Rate): 0.947228233727318"
```

```r
# F1 Score
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)
print(paste("Logistic Model 1 F1 Score:", f1_score))
```

```
## [1] "Logistic Model 1 F1 Score: 0.320272572402044"
```

```r
# Calculating AUC (Area Under the ROC Curve)
roc_curve <- roc(data_train$TARGET_FLAG, logistic_predictions)
```

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

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

```r
auc <- auc(roc_curve)
print(paste("Logistic Model 1 AUC (Area Under the ROC Curve):", auc))
```

```
## [1] "Logistic Model 1 AUC (Area Under the ROC Curve): 0.716757823507584"
```

```r
#LOGISTIC MODEL 2
# Predicting the target variable using the binary logistic regression model
logistic_predictions <- predict(model_glm1, newdata = data_train, type = "response")

# Converting probabilities to binary predictions
binary_predictions <- ifelse(logistic_predictions > 0.5, 1, 0)

# Confusion matrix
conf_matrix <- table(data_train$TARGET_FLAG, binary_predictions)

# Displaying Confusion Matrix
print("Logistic Model 2 Confusion Matrix:")
```

```
## [1] "Logistic Model 2 Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
##    binary_predictions
##        0    1
##   0 3724 1941
##   1  670 1358
```

```r
# Accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Logistic Model 2 Accuracy:", accuracy))
```

```
## [1] "Logistic Model 2 Accuracy: 0.660600545950864"
```

```r
# Classification Error Rate
error_rate <- 1 - accuracy
print(paste("Logistic Model 2 Classification Error Rate:", error_rate))
```

```
## [1] "Logistic Model 2 Classification Error Rate: 0.339399454049136"
```

```r
# Confusion Matrix
true_negatives <- 3724
false_positives <- 1941
false_negatives <- 670
true_positives <- 1358

# Precision
precision <- true_positives / (true_positives + false_positives)
print(paste("Logistic Model 2 Precision:", precision))
```

```
## [1] "Logistic Model 2 Precision: 0.411639890876023"
```

```r
# Sensitivity (True Positive Rate)
sensitivity <- true_positives / (true_positives + false_negatives)
print(paste("Logistic Model 2 Sensitivity (True Positive Rate):", sensitivity))
```

```
## [1] "Logistic Model 2 Sensitivity (True Positive Rate): 0.669625246548323"
```

```r
# Specificity (True Negative Rate)
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
print(paste("Logistic Model 2 Specificity (True Negative Rate):", specificity))
```

```
## [1] "Logistic Model 2 Specificity (True Negative Rate): 0.657369814651368"
```

```r
# F1 Score
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)
print(paste("Logistic Model 2 F1 Score:", f1_score))
```

```
## [1] "Logistic Model 2 F1 Score: 0.509855453350854"
```

```r
# Calculating AUC (Area Under the ROC Curve)
roc_curve <- roc(data_train$TARGET_FLAG, logistic_predictions)
```

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

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

```r
auc <- auc(roc_curve)
print(paste("Logistic Model 2 AUC (Area Under the ROC Curve):", auc))
```

```
## [1] "Logistic Model 2 AUC (Area Under the ROC Curve): 0.711360676913328"
```

Given the class imbalance issue, where the majority class (0) significantly outweighs the minority class (1), Logistic Model 1 may have the highest accuracy, it might not be the best choice due to its low sensitivity (True Positive Rate) and F1 Score. Logistic Model 1 has a sensitivity of only 0.597, indicating that it struggles to correctly classify instances of car crashes. This lower sensitivity suggests that Model 1 is biased towards predicting the majority class, leading to a higher number of false negatives the actual crashes incorrectly classified as non-crashes. On the other hand, Logistic Model 3 exhibits a higher sensitivity of 0.429, indicating a better ability to identify true positive cases of car crashes despite the class imbalance. Additionally, Model 3 demonstrates a higher F1 Score, which balances precision and sensitivity, making it more suitable for imbalanced this dataset.

```r
#LOGISTIC MODEL 3
# Predicting the target variable using the binary logistic regression model
logistic_predictions <- predict(model_glm3, newdata = data_train, type = "response")

# Converting probabilities to binary predictions
binary_predictions <- ifelse(logistic_predictions > 0.5, 1, 0)

# Confusion matrix
conf_matrix <- table(data_train$TARGET_FLAG, binary_predictions)

# Displaying Confusion Matrix
print("Logistic Model 3 Confusion Matrix:")
```

```
## [1] "Logistic Model 3 Confusion Matrix:"
```

```r
print(conf_matrix)
```

```
##    binary_predictions
##        0    1
```

```
##   0 4039 1600
##   1  803 1203
```

```
# Accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
print(paste("Logistic Model 3 Accuracy:", accuracy))
```

```
## [1] "Logistic Model 3 Accuracy: 0.685676913015043"
```

```
# Classification Error Rate
error_rate <- 1 - accuracy
print(paste("Logistic Model 3 Classification Error Rate:", error_rate))
```

```
## [1] "Logistic Model 3 Classification Error Rate: 0.314323086984957"
```

```
# Confusion Matrix
true_negatives <- 4039
false_positives <- 803
false_negatives <- 1600
true_positives <- 1203

# Precision
precision <- true_positives / (true_positives + false_positives)
print(paste("Logistic Model 3 Precision:", precision))
```

```
## [1] "Logistic Model 3 Precision: 0.599700897308076"
```

```
# Sensitivity (True Positive Rate)
sensitivity <- true_positives / (true_positives + false_negatives)
print(paste("Logistic Model 3 Sensitivity (True Positive Rate):", sensitivity))
```

```
## [1] "Logistic Model 3 Sensitivity (True Positive Rate): 0.429183018194791"
```

```
# Specificity (True Negative Rate)
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
print(paste("Logistic Model 3 Specificity (True Negative Rate):", specificity))
```

```
## [1] "Logistic Model 3 Specificity (True Negative Rate): 0.716261748536975"
```

```
# F1 Score
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)
print(paste("Logistic Model 3 F1 Score:", f1_score))
```

```
## [1] "Logistic Model 3 F1 Score: 0.500311915159077"
```

```
# Calculating AUC (Area Under the ROC Curve)
roc_curve <- roc(data_train$TARGET_FLAG, logistic_predictions)
```

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

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

```
auc <- auc(roc_curve)
print(paste("Logistic Model 3 AUC (Area Under the ROC Curve):", auc))
```

```
## [1] "Logistic Model 3 AUC (Area Under the ROC Curve): 0.710972729974644"
```

PREDICTIONS USING THE EVALUATION DATASET

```
# Predict using Multiple Linear 4
predictions_eval <- predict(wls_model, newdata = data_eval)
```

```r
# Display the first few predicted values
head(predictions_eval)
```

```
##         1         2         3         4         5         6
## 1434.3747 2146.7129 1743.7588  955.1852 2388.9203 1593.6098
```

```r
# Predict using Logistic Model 3
predictions_eval <- predict(model_glm3, newdata = data_eval, type = "response")

# Convert predicted probabilities to binary predictions
binary_predictions_eval <- ifelse(predictions_eval > 0.5, 1, 0)

# Create a data frame to store the results
evaluation_results <- data.frame(Actual = data_eval$TARGET_FLAG, Predicted = binary_predictions_eval)

# Display the first few rows of the evaluation results
head(evaluation_results)
```

```
##   Actual Predicted
## 1     NA         0
## 2     NA         0
## 3     NA         0
## 4     NA         0
## 5     NA         1
## 6     NA         0
```