Homework 4 - DATA 621

Business Analytics & Data Mining

Ohannes (Hovig) Ohannessian
ohannes.ohannessian16@spsmail.cuny.edu

7/5/2018

Contents

1. Introduction

In this assignment, the dataset contains information about customers of an auto insurance company. It consists of 8161 customers with 23 potential predictor variables and 2 response (target) variables. The purpose of this assignment is to build a series two different models.

1- The first model will predict whether a person will get into a car crash
2- The second model will be used to predict the amount as to which the crash will cost

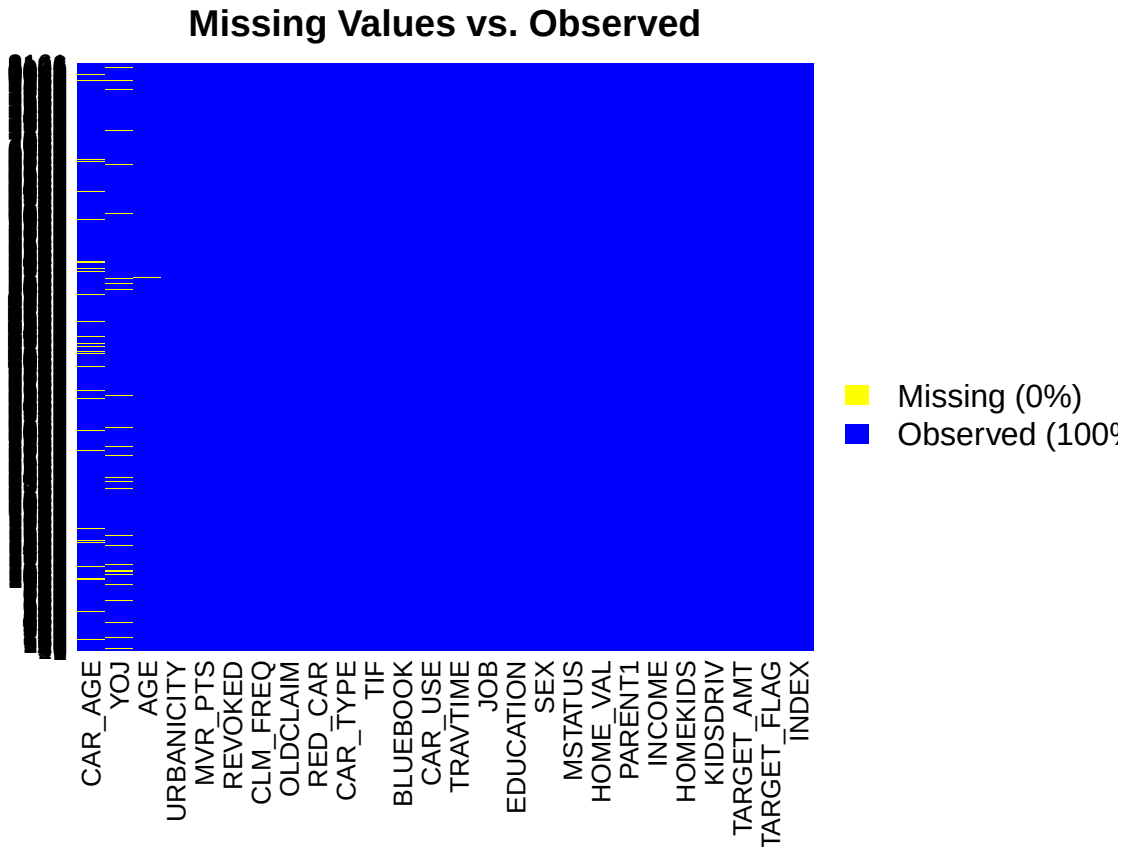
These variables are:

- `TARGET_FLAG`: This flag will take a value of 0 or 1. One means that the person was in a car crash and 0 means that they were not.
- `TARGET_AMT`: This is the amount associated with the car crash. If the `TARGET_FLAG` field is 0, then there will be no amount, because the person did not get into a crash. If the `TARGET_FLAG` is a 1, then there will be a value in this field, because there was a crash.

2. Data Exploration

The first thing that has to be taken care of is the missing (NA) values in the data. We can see that in Figure 1.1, there are a few fields that have some missing values. We can see that the fields JOB, CAR_AGE, HOME_VAL, YOJ, AGE and INCOME have missing values. These need to either be removed or imputed to continue on with the analysis.

Figure 1.1



We can also see, in the summary statistics (Figure 1.2) some of the data is incorrect. The CAR_AGE filed is showing that the minimum car age is -3. That is not a possible value for that field. That field will have to be looked at more closely to see what other values are invalid. They will have to be re-imputed or removed before the final models are created.

The data also contains 10 categorical variables and 16 numeric variables. The categorical data will need to be converted into a numerical field in order to be plugged into the model and get a value for the two target variables.

Figure 1.2

```
##      INDEX        TARGET_FLAG       TARGET_AMT        KIDSDRIV
##  Min.   :    1   Min.   :0.0000   Min.   :     0   Min.   :0.0000
##  1st Qu.: 2559   1st Qu.:0.0000   1st Qu.:     0   1st Qu.:0.0000
##  Median : 5133   Median :0.0000   Median :     0   Median :0.0000
##  Mean   : 5152   Mean   :0.2638   Mean   :  1504   Mean   :0.1711
##  3rd Qu.: 7745   3rd Qu.:1.0000   3rd Qu.:  1036   3rd Qu.:0.0000
##  Max.   :10302   Max.   :1.0000   Max.   :107586   Max.   :4.0000
##
##       AGE           HOMEKIDS           YOJ            INCOME
##  Min.   :16.00   Min.   :0.0000   Min.   : 0.0   $0      : 615
##  1st Qu.:39.00   1st Qu.:0.0000   1st Qu.: 9.0           : 445
##  Median :45.00   Median :0.0000   Median :11.0   $26,840 :   4
##  Mean   :44.79   Mean   :0.7212   Mean   :10.5   $48,509 :   4
##  3rd Qu.:51.00   3rd Qu.:1.0000   3rd Qu.:13.0   $61,790 :   4
##  Max.   :81.00   Max.   :5.0000   Max.   :23.0   $107,375:   3
##  NA's   :6                        NA's   :454    (Other) :7086
##  PARENT1        HOME_VAL    MSTATUS      SEX               EDUCATION
##  No :7084   $0      :2294   Yes :4894   M  :3786   <High School :1203
##  Yes:1077           : 464   z_No:3267   z_F:4375   Bachelors    :2242
##             $111,129:   3                          Masters      :1658
##             $115,249:   3                          PhD          : 728
##             $123,109:   3                          z_High School:2330
##             $153,061:   3
##             (Other) :5391
##             JOB          TRAVTIME           CAR_USE        BLUEBOOK
##  z_Blue Collar:1825   Min.   :  5.00   Commercial:3029   $1,500 : 157
##  Clerical     :1271   1st Qu.: 22.00   Private   :5132   $6,000 :  34
##  Professional :1117   Median : 33.00                     $5,800 :  33
##  Manager      : 988   Mean   : 33.49                     $6,200 :  33
##  Lawyer       : 835   3rd Qu.: 44.00                     $6,400 :  31
##  Student      : 712   Max.   :142.00                     $5,900 :  30
##  (Other)      :1413                                      (Other):7843
##       TIF               CAR_TYPE    RED_CAR       OLDCLAIM
##  Min.   : 1.000   Minivan    :2145   no :5783   $0     :5009
##  1st Qu.: 1.000   Panel Truck: 676   yes:2378   $1,310 :   4
##  Median : 4.000   Pickup     :1389              $1,391 :   4
##  Mean   : 5.351   Sports Car : 907              $4,263 :   4
##  3rd Qu.: 7.000   Van        : 750              $1,105 :   3
##  Max.   :25.000   z_SUV      :2294              $1,332 :   3
##                                                 (Other):3134
##     CLM_FREQ      REVOKED       MVR_PTS          CAR_AGE
##  Min.   :0.0000   No :7161   Min.   : 0.000   Min.   :-3.000
##  1st Qu.:0.0000   Yes:1000   1st Qu.: 0.000   1st Qu.: 1.000
##  Median :0.0000              Median : 1.000   Median : 8.000
##  Mean   :0.7986              Mean   : 1.696   Mean   : 8.328
##  3rd Qu.:2.0000              3rd Qu.: 3.000   3rd Qu.:12.000
##  Max.   :5.0000              Max.   :13.000   Max.   :28.000
##                                               NA's   :510
##                  URBANICITY
##  Highly Urban/ Urban  :6492
##  z_Highly Rural/ Rural:1669
##
##
##
##
##
```

In Figure 1.3 we take a look at the first target variable (`TARGET_FLAG`). The summary stats are saying that the mean is less than .5 which means that the data is not evenly disbursed between 0 and 1. The mean would be .5 if that were the case. We can see by the histogram that there are about 3 times as many 0's (no crashes) then there are crashes in this data set. In Figure 1.4, we see that the amount that the customer paid is also around 0. That is agreeing with the high amount of no crashes that we saw in Figure 1.3. This means we can procede with our analysis.

Figure 1.3

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.2638  1.0000  1.0000
```

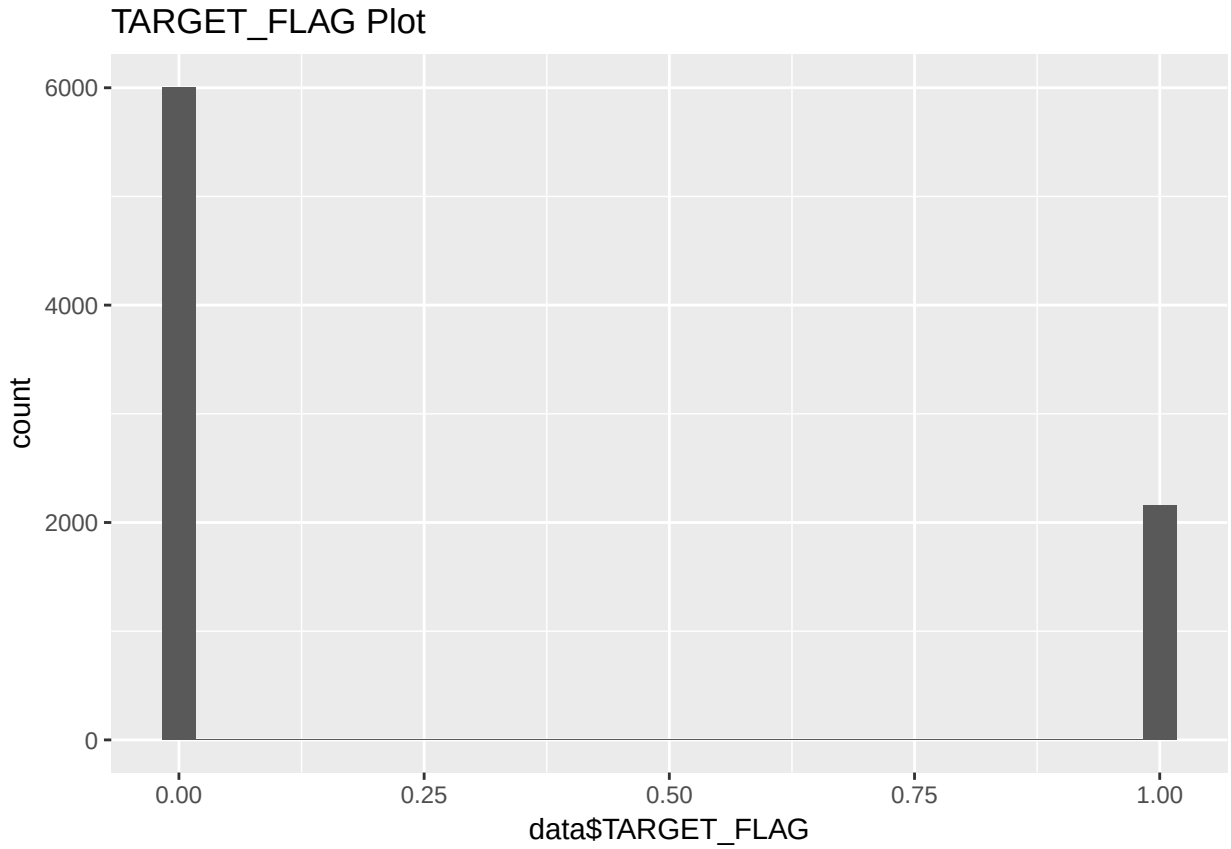
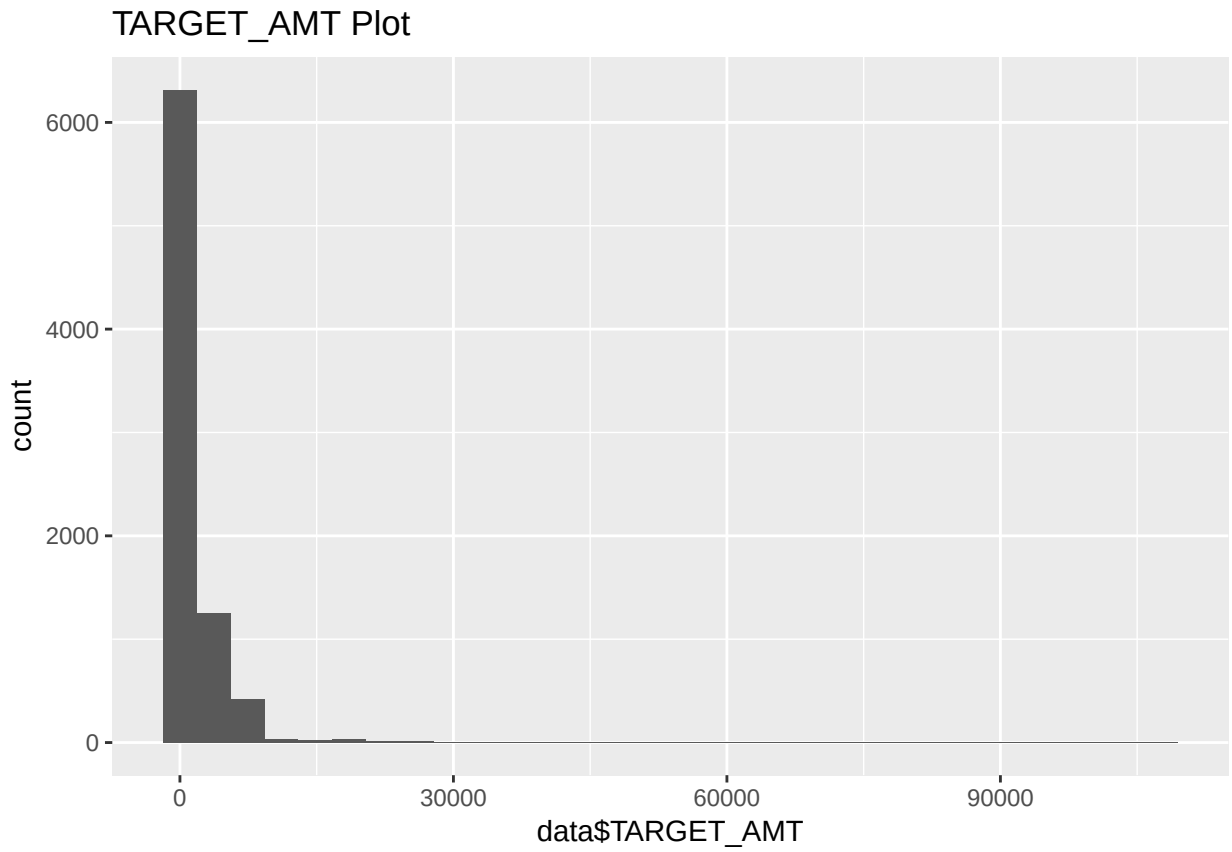


Figure 1.4

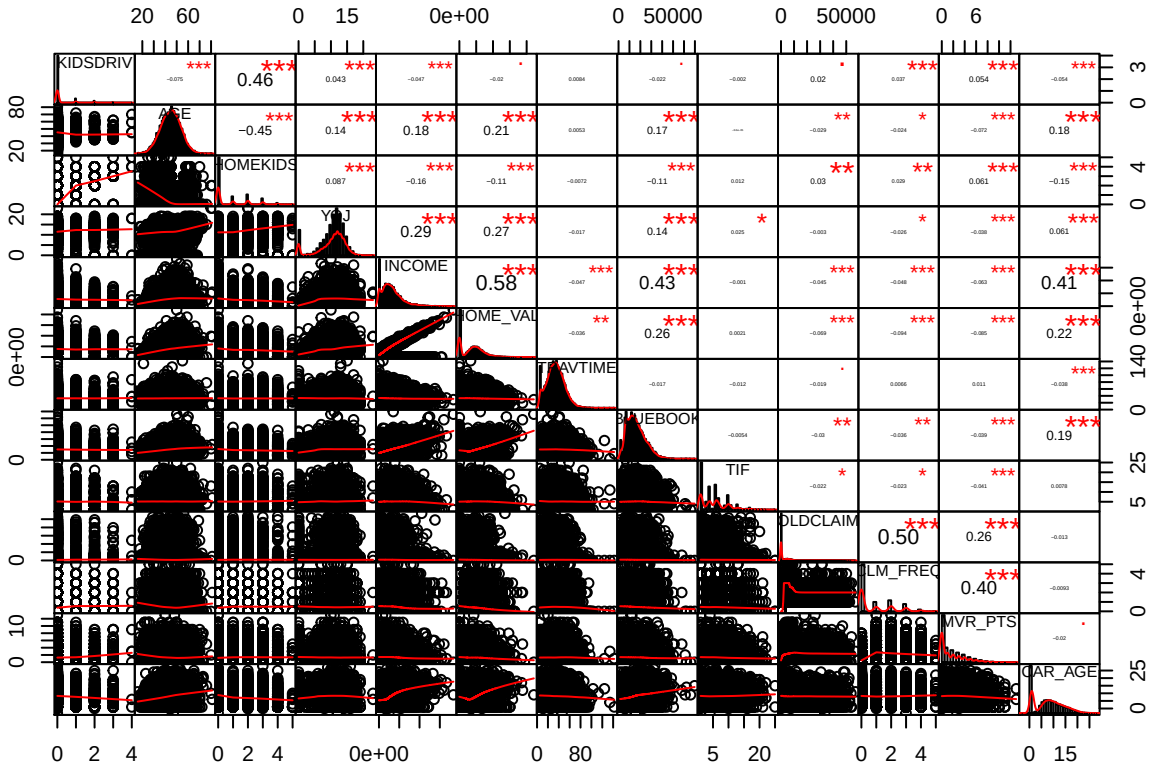
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1504    1036  107586
```



Finally, we want to look at the variables and see how they correlate with each other. In Figure 1.5 we see a grid that plots each field in the data set against each other. It also shows how strong of a correlation there is between each variable through the correlation value and the amount of starts in each box (correlation significance), 3 stars means that the variables have a very strong correlation and the strength drops as the number of stars drop.

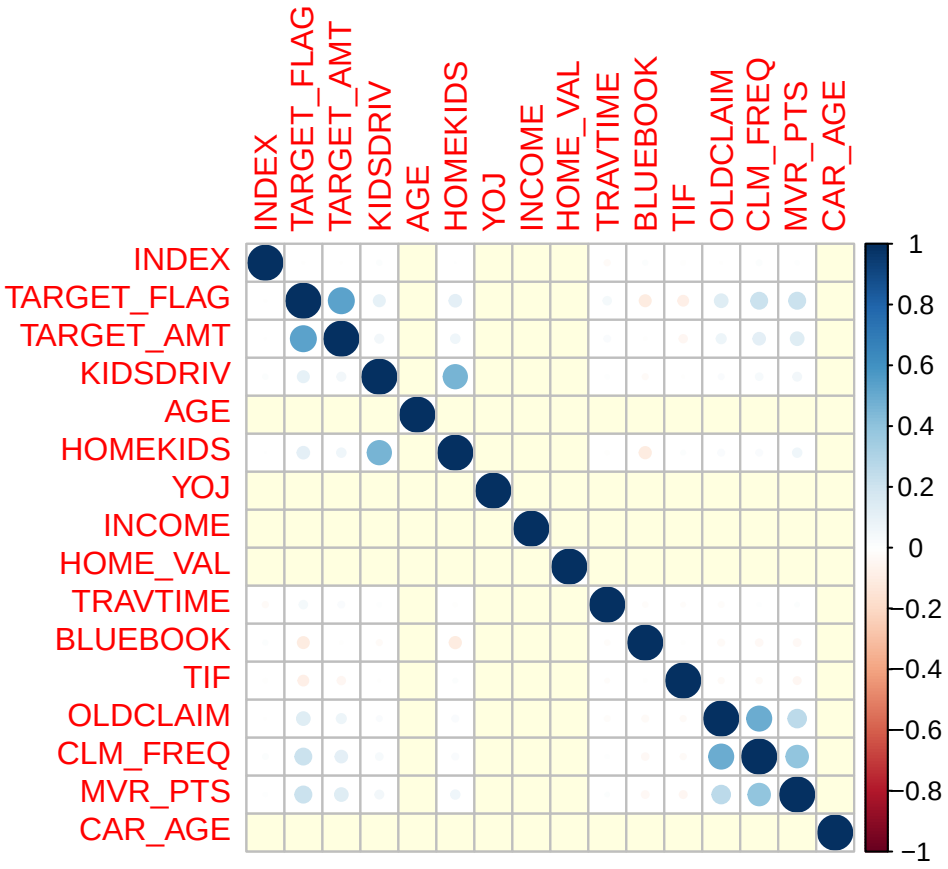
We can see from the grid, that not all of the variables correlate with each other. The variables that appear to not make good predictors would be, MVR_PTS, TRAVTIME, and TIF. These variables have a low correlation signifiance. This means that they may not be good choices to include in our final model for any of the target variables. All of the other variables appear to have a relatively strong correlation between each other and would be good choices for the overall model.

Figure 1.5



Firgue 1.6 is another way to show the correlation between variables, just like in Figure 1.5. This grid has different color/size circles within each variable intersection in the grid. The bigger the circle the better the correlation. We can see that the fields TIF, TRAVTIME, and MVR_PTS are still relatively bad choices for variables, just like in Figure 1.5.

Figure 1.6



3. Data Preparation

3.1 - Cleaning NA's & Invalid Values

The first thing that we want to take care of is the NA's (missing values) in the data. We decided that the best course of action would be to remove any of the records that were not complete. Also, we will remove any of the data that is invalid (like a car age below 0). With the removal of the NA's and the invalid values, the total observations for the data is reduced from 8161 to 6044 total obsverations. That is still plenty of observations to do analysis and modeling on.

3.2 - Categorical Data

Within our data, there are 10 fields that are categorical. These variables group the data into different sections. The best way to deal with this data is to make dummy variables. Most of the data contains 1 of 2 posibilities (YES/NO etc). This allows me to assign a value of 0 for one possibility and a 1 for another. It changes the factor data into a number so it can be used in the regression analysis. The education and jobs fields were a little different. They have more than 2 possibilities. I grouped the possibilities into logical groups. The eductaion was grouped by level of academic achievement (Masters and above = 1, below is a 0). The education was grouped along the same way, a college education/advanced education was grouped into one (Lawyer, Professional, Manager = 1), everybody else gets a 0. The field of CAR_TYPE is dealt the same way as well. Those who drive a Panel Truck, Pickup or Sports car will be labeled with a 1 and everything else will be labeled with a 0.

3.3 - Accounting Data Conversion

The last thing that needed to be done with the data is to convert the accounting data, into numeric fields. The accounting data (dollar amount data) has dollar signs within the data. That means that the data will be treated as a character set. The dollar sign needs to be removed and the data needs to be changed into a number, so it can be used in the analysis later on.

4. Build Models

4.1 - TARGET_FLAG

This is the target variable that will tell us there was a crash or not for the given customer. If the field takes a value of 0, that means that the customer was not in an accident, or the accident was not their fault. If the field takes a value of 1, that means the customer has been in an accident, or the accident was their fault.

The first thing that we need to do is split the data up into a training set and a test set. We will be taking the data and separating it up so 70% of the data is the training set, and 30% of the data will be the testing set.

Model 1

In this model we are doing a type of stepwise function. We are taking the training set and removing some of the variables that we feel are not good predictors. We run the model against the remaining variables and look at the output. Then, we go back through and remove anymore variables that we do not feel are good predictors. That will leave us with a final function that has all the variables that we fell will make the best predicting function.

The output for this function is as follows:

```
##
## Call:
## lm(formula = TARGET_FLAG ~ TARGET_AMT + PARENT1 + HOME_VAL +
##     MSTATUS + TRAVTIME + CAR_USE + REVOKED + CAR_AGE + URBANICITY,
##     data = training_2a)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3927 -0.2157 -0.1140  0.0936  1.0080
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.291e-01  2.654e-02   8.635  < 2e-16 ***
## TARGET_AMT   4.788e-05  1.200e-06  39.892  < 2e-16 ***
## PARENT1     -1.110e-01  1.732e-02  -6.406 1.65e-10 ***
## HOME_VAL    -4.018e-07  4.706e-08  -8.537  < 2e-16 ***
## MSTATUS     -1.536e-02  1.323e-02  -1.160    0.246
## TRAVTIME     1.577e-03  3.305e-04   4.772 1.88e-06 ***
## CAR_USE     -7.831e-02  1.077e-02  -7.270 4.21e-13 ***
## REVOKED      1.112e-01  1.591e-02   6.989 3.16e-12 ***
## CAR_AGE     -5.889e-03  9.527e-04  -6.181 6.93e-10 ***
## URBANICITY   1.960e-01  1.340e-02  14.627  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3469 on 4503 degrees of freedom
## Multiple R-squared:  0.3708, Adjusted R-squared:  0.3695
## F-statistic: 294.9 on 9 and 4503 DF,  p-value: < 2.2e-16
```

Model 2

In this model, we are untilizing a backwards approach into solving for the overall model. The backwards approach to variable selection starts off withh all variables in the model. I then starts to remove fields, until it gets to a point where removing anymore fields will not be beneficial to the model. That is the point when the final model is found.

To solve for the `TARGET_FLAG` field, I will be using a probit function. This function is very useful when there are only two possible outcomes for the field that you are trying to predict. This model utilizes backwards selection when picking the variables for the model. It starts out all of the variables that are possible. It then starts to remove variables until it reaches the optimal solution for the function. The output from the model is as follows:

```
##
## Call:
## glm(formula = TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 + HOME_VAL +
##     MSTATUS + SEX + JOB + TRAVTIME + CAR_USE + BLUEBOOK + TIF +
##     CAR_TYPE + OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS + CAR_AGE +
##     URBANICITY, family = binomial(link = "probit"), data = train1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3784  -0.7447  -0.4261   0.6173   3.2243
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.078e+00  1.375e-01  -7.844 4.37e-15 ***
## KIDSDRIV     1.999e-01  4.345e-02   4.602 4.19e-06 ***
## INCOME      -3.507e-06  7.344e-07  -4.775 1.80e-06 ***
## PARENT1     -2.964e-01  7.238e-02  -4.095 4.23e-05 ***
## HOME_VAL    -6.489e-07  2.617e-07  -2.479 0.013164 *
## MSTATUS      2.187e-01  6.176e-02   3.542 0.000398 ***
## SEX          1.754e-01  4.777e-02   3.672 0.000241 ***
## JOB         -1.601e-01  4.833e-02  -3.312 0.000926 ***
## TRAVTIME     7.918e-03  1.429e-03   5.542 3.00e-08 ***
## CAR_USE     -4.481e-01  5.252e-02  -8.530  < 2e-16 ***
## BLUEBOOK    -1.668e-05  3.063e-06  -5.444 5.20e-08 ***
## TIF         -2.748e-02  5.553e-03  -4.950 7.43e-07 ***
## CAR_TYPE     1.324e-01  4.820e-02   2.747 0.006016 **
## OLDCLAIM    -7.015e-06  3.030e-06  -2.315 0.020609 *
## CLM_FREQ     1.271e-01  2.239e-02   5.677 1.37e-08 ***
## REVOKED      5.036e-01  7.088e-02   7.105 1.20e-12 ***
## MVR_PTS      7.526e-02  1.061e-02   7.094 1.30e-12 ***
## CAR_AGE     -1.339e-02  4.458e-03  -3.003 0.002670 **
## URBANICITY   1.146e+00  7.522e-02  15.238  < 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: 5142.1  on 4512  degrees of freedom
## Residual deviance: 4104.8  on 4494  degrees of freedom
## AIC: 4142.8
##
## Number of Fisher Scoring iterations: 5
```

Model 3

The model is a little interesting in the fact that, the original analysis of the data showed that the fields TIF, TRAVTIME and MVR_PTS were not good predictors. They all had very few starts when being correlated with the other variables. They all showed up in the model. This could be due to the fact that the original analysis was based on a linear comparision. This equation is a log function. That shows that those variables may have a horrible linear relationship with the rest of the variables but a very stong logistic relationship.

To solve for the `TARGET_FLAG` field, I will be using a probit function, just like during the backwards function. This function goes the opposite way as the backward function. It starts with a plain function and adds variables until it gets to the optimal solution. Once it cannot add variables to make the equation better, it stops and that is the final function. The output for the forward stepping function is as follows:

```
##
## Call:
## glm(formula = TARGET_FLAG ~ URBANICITY + HOME_VAL + MVR_PTS +
##     CAR_USE + BLUEBOOK + PARENT1 + REVOKED + INCOME + TRAVTIME +
```

```
##     CLM_FREQ + TIF + KIDSDRIV + JOB + SEX + MSTATUS + CAR_AGE +
##     CAR_TYPE + OLDCLAIM, family = binomial(link = "probit"),
##     data = train1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3784  -0.7447  -0.4261   0.6173   3.2243
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.078e+00  1.375e-01  -7.844 4.37e-15 ***
## URBANICITY   1.146e+00  7.522e-02  15.238  < 2e-16 ***
## HOME_VAL    -6.489e-07  2.617e-07  -2.479 0.013164 *
## MVR_PTS      7.526e-02  1.061e-02   7.094 1.30e-12 ***
## CAR_USE     -4.481e-01  5.252e-02  -8.530  < 2e-16 ***
## BLUEBOOK    -1.668e-05  3.063e-06  -5.444 5.20e-08 ***
## PARENT1     -2.964e-01  7.238e-02  -4.095 4.23e-05 ***
## REVOKED      5.036e-01  7.088e-02   7.105 1.20e-12 ***
## INCOME      -3.507e-06  7.344e-07  -4.775 1.80e-06 ***
## TRAVTIME     7.918e-03  1.429e-03   5.542 3.00e-08 ***
## CLM_FREQ     1.271e-01  2.239e-02   5.677 1.37e-08 ***
## TIF         -2.748e-02  5.553e-03  -4.950 7.43e-07 ***
## KIDSDRIV     1.999e-01  4.345e-02   4.602 4.19e-06 ***
## JOB         -1.601e-01  4.833e-02  -3.312 0.000926 ***
## SEX          1.754e-01  4.777e-02   3.672 0.000241 ***
## MSTATUS      2.187e-01  6.176e-02   3.542 0.000398 ***
## CAR_AGE     -1.339e-02  4.458e-03  -3.003 0.002670 **
## CAR_TYPE     1.324e-01  4.820e-02   2.747 0.006016 **
## OLDCLAIM    -7.015e-06  3.030e-06  -2.315 0.020609 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5142.1  on 4512  degrees of freedom
## Residual deviance: 4104.8  on 4494  degrees of freedom
## AIC: 4142.8
##
## Number of Fisher Scoring iterations: 5
```

This functions is very interesting. In the backwards function, stated above (Model 1), the MSTATUS (Marital status) field is positive, but it is negative in the forwards function. The sign switched depending on which way you come at the optimal function. Since the model also contains the field of KIDSDRIV and PARENT1, it is possible that multicolinearity could be playing a factor. It is also very interesting that the TRAVTIME, TIF and MVR_PTS fields are also included in this model just like in the backwards model.

Model 4 (Remove Redundancy, Calculate All Combinations)

Because the data contains such a large number of categorical fields, it is impossible (with our machine power + time constraints) to calculate and test the model for all possible combinations of the categorical's inclusion and/or values. However, what if some of the fields could ***simply be removed***? Let's take a look at the ***correlations*** between some of the dependent attributes in our data.

col_names	accuracy_percent
CAR_TYPE	74.313
CAR_USE	74.313
EDUCATION	74.313
JOB	74.313
MSTATUS	74.313
PARENT1	74.313

col_names	accuracy_percent
RED_CAR	74.313
REVOKED	74.313
SEX	74.313
URBANICITY	74.313

Those categorical variables have the exact same calculated accuracties.

Could this be coincidence? Seems those columns could be duplicates, let's run Chi-Square correlation tests to confirm:

all_chi_sq_labels	all_chi_sq_results
car_type_to_car_use	0.000
car_type_to_education	0.883
car_type_to_job	0.004
car_type_to_mstatus	0.850
car_type_to_parent1	0.969
car_type_to_red_car	0.000
car_type_to_revoked	0.848
car_type_to_sex	0.000
car_type_to_urbancity	0.907

Chi-Squared says some columns are exact duplicates: Remove Them! (We'll keep CARTYPE):

Why can we simply remove fields? Imagine there is a categorical called letters with "a","b","c","d" values. And there is a categorical called numbers with "one","two","three","four". If for EVERY occurence:

- "a" -> "one"
- "b" -> "two"
- "c" -> "three"
- "d" -> "four"

Then would it make sense to calculate a model for all 16 letter-to-number commbinations? Of course not, they are 100% correlated. We can just keep one of the fields.

```
drops <- c("CAR_USE","EDUCATION","JOB","PARENT1","RED_CAR","SEX","URBANICITY")
ins.train <- ins.train[ , !(names(ins.train) %in% drops)]
ins.test <- ins.test[ , !(names(ins.test) %in% drops)]
```

Also, ***car_type_to_mstatus*** and ***car_type_to_revoked*** were close, lets compare mstatus to revoked:

```
mstatus_to_revoked <- chisq.test(table(ins.train$MSTATUS, ins.train$REVOKED))
round(mstatus_to_revoked$p.value, digits = 3)
```

```
## [1] 0.037
```

So those are ALSO ***DUPLICATES***, remove REVOKED

```
drops <- c("REVOKED")
ins.train <- ins.train[ , !(names(ins.train) %in% drops)]
ins.test <- ins.test[ , !(names(ins.test) %in% drops)]
```

So now that we have lowered our number of categorical variables, and thus lowered the total number of possible categorical combinations to calculate, we can use a tool "grind out" every combination and evaluate based on whatever criteria we wish:

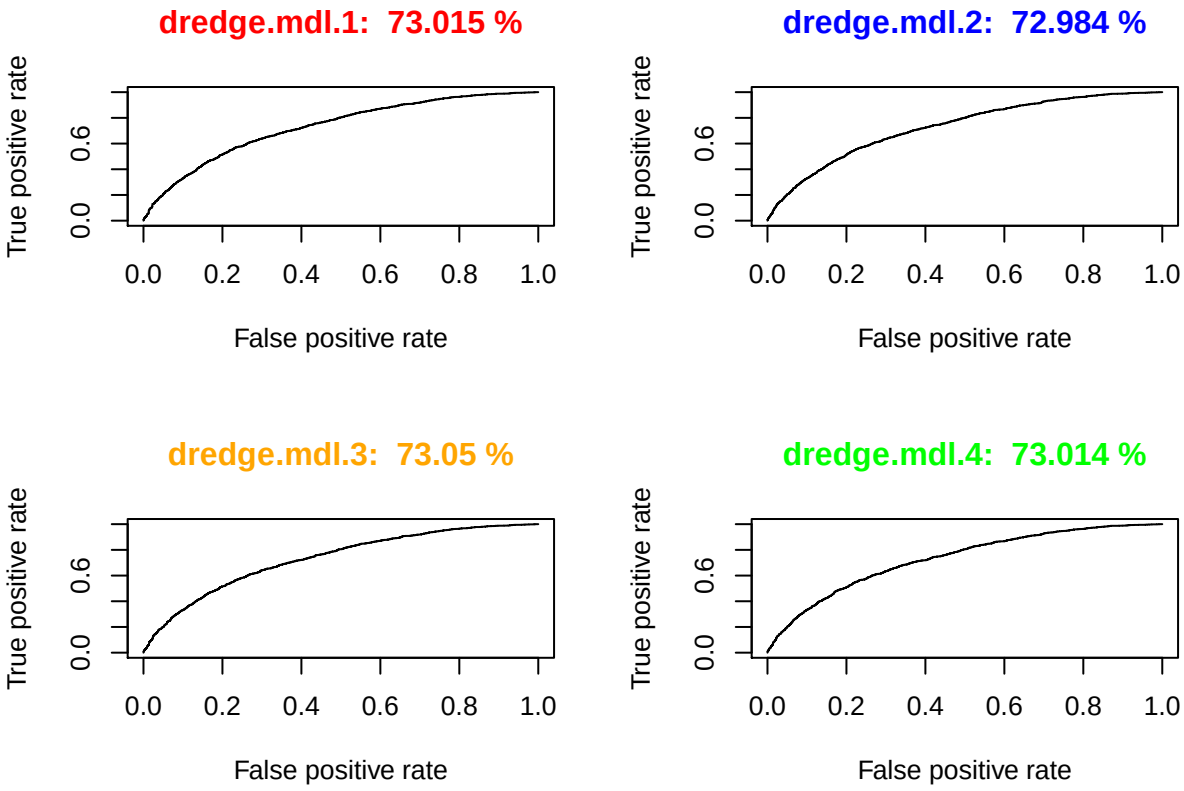
Try out all categorical combinations with a tools, in this case - MuMIn DREDGE

```
## Fixed term is "(Intercept)"
```

	(Intercept)	AGE	BLUEBOOK	CAR_AGE	CAR_TYPE	CLM_FREQ	HOME_VAL	HOMEKIDS	IN
14336	0.2687567	-0.0012601	-2.6e-06	-0.0036302	0.0578763	0.0529060	-1e-07	0.0167771	
14335	0.2136635	NA	-2.7e-06	-0.0037902	0.0579503	0.0527074	-1e-07	0.0215575	
16384	0.2678297	-0.0012537	-2.6e-06	-0.0036491	0.0579374	0.0501092	-1e-07	0.0166929	

	(Intercept)	AGE	BLUEBOOK	CAR_AGE	CAR_TYPE	CLM_FREQ	HOME_VAL	HOMEKIDS	IN
16383	0.2130082	NA	-2.7e-06	-0.0038086	0.0580118	0.0498788	-1e-07	0.0214479	

```
## NULL
## NULL
## NULL
```




```
## NULL
```

All in all, this approach yielded models with AUC's around 0.735, slightly higher than the basic step reduction.

4.2 - TARGET_AMT

This is the target field that says wether the customer had to pay some amount after an accident. This field will only have a value if the `TARGET_FLAG` field has a 1. If the `TARGET_FLAG` field is a 0, then this field will be 0 as well. The first thing that we have to do is re-pick the training set. We do not want to use the exact same training set as before, because it is the same data and we really are not changing anything from the first models. We will be using a 70/30 split just like before.

Model 1

The first model that will be used is a form of a stepwise function. Bascially, we select a set of fields from the training set and use that as a base model. We take a look at that model and see what fields should be kept and what fields whould be removed. We remove the fields that we feel are not well correlated and get the final model.

The output for the model is as follows:

```
##
## Call:
## lm(formula = TARGET_AMT ~ HOME_VAL + MSTATUS + CAR_USE + URBANICITY,
##     data = training_2a)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6438  -2892  -1385    272  67739
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.051e+03  9.854e+02   4.112  4.2e-05 ***
## HOME_VAL     3.265e-03  1.993e-03   1.638   0.1017
## MSTATUS      1.257e+03  4.755e+02   2.644   0.0083 **
## CAR_USE     -4.213e+02  4.149e+02  -1.015   0.3101
## URBANICITY   7.356e+02  9.003e+02   0.817   0.4141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7135 on 1186 degrees of freedom
## Multiple R-squared:  0.007494,   Adjusted R-squared:  0.004147
## F-statistic: 2.239 on 4 and 1186 DF,  p-value: 0.06289
```

Model 2

The second model that we will be using is a forward selection, like used in the `TARGET_FLAG` prediction section above. This model takes a "blank" equation and starts to add variables until it finds the optimal solution for the model. It is a very similar process to the first model.

The outout for the model is as follows:

```
##
## Call:
## lm(formula = TARGET_AMT ~ BLUEBOOK + MVR_PTS + CAR_AGE + MSTATUS +
##     YOJ, data = train2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8786  -2993  -1374    492  67603
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2893.4153   701.0309   4.127 3.93e-05 ***
## BLUEBOOK       0.1151     0.0259   4.446 9.59e-06 ***
## MVR_PTS      186.4368    78.1629   2.385   0.0172 *
## CAR_AGE      -82.8891    38.2707  -2.166   0.0305 *
## MSTATUS      925.5317   415.0766   2.230   0.0259 *
## YOJ           67.2639    46.6534   1.442   0.1496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7063 on 1185 degrees of freedom
## Multiple R-squared:  0.02825,    Adjusted R-squared:  0.02415
## F-statistic:  6.89 on 5 and 1185 DF,  p-value: 2.38e-06
```

It is very interesting to conpare the two models in this section. We can see by the summary statistics of both models, that Model 1'2 coefficients are are statistically significant (below .05 confidence), while model 1 seems to have a few variables that are not significant at all, but the R squared value for model 1 is lower than the R squared of the second model. It will be interesting to do some further analysis and see which model is actuall better than the other.

5. Select Models

Now that all of the models have been created and predicted, it is time to pick and choose which are the best. We will pick the best model for `TARGET_FLAG` (probit model) and the best model for the `TARGET_AMT` field (linear regression).

5.1 - TARGET_FLAG

F1 Score

```
##        Predicted
## Actual    0    1
##      0 3349    5
##      1  662  497
```

```
##        Predicted
## Actual    0    1
##      0 3139  215
##      1  745  414
```

```
##        Predicted
## Actual    0    1
##      0 3139  215
##      1  745  414
```

We can take a look at how well the models compare to each other. One way to do that is to look at the F1 score. This score takes the precision and recall of the model and lets us know how the model fairs. The equation for the F1 score is as follows:

$$F1Score = \frac{2 * precision * recall}{precision + recall}$$

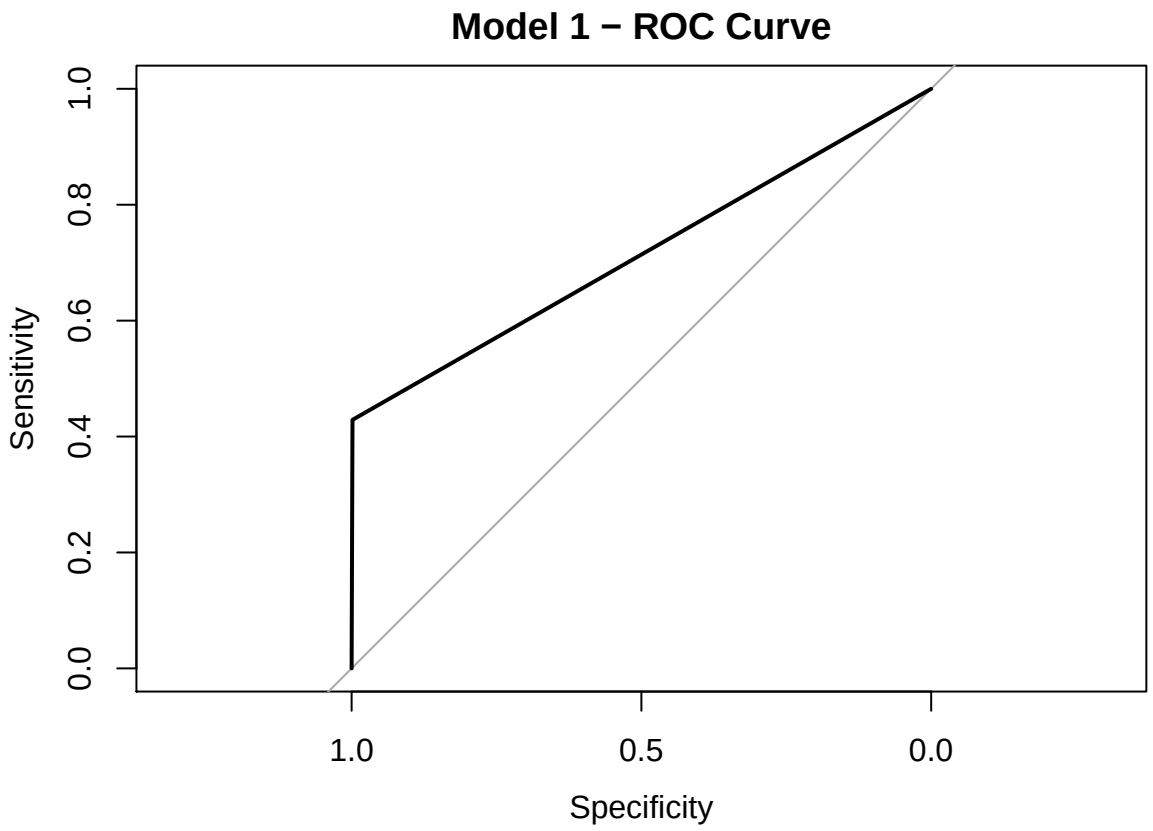
We can see that all of the models have pretty high F1 scores. The best most is model 1 by just a little bit.

model	precision	recall	F1 Score
model 1	.8563123	.9974202	.9214956
model 2	.8722322	.9316075	.9009427
model 3	.8722322	.9316075	.9009427

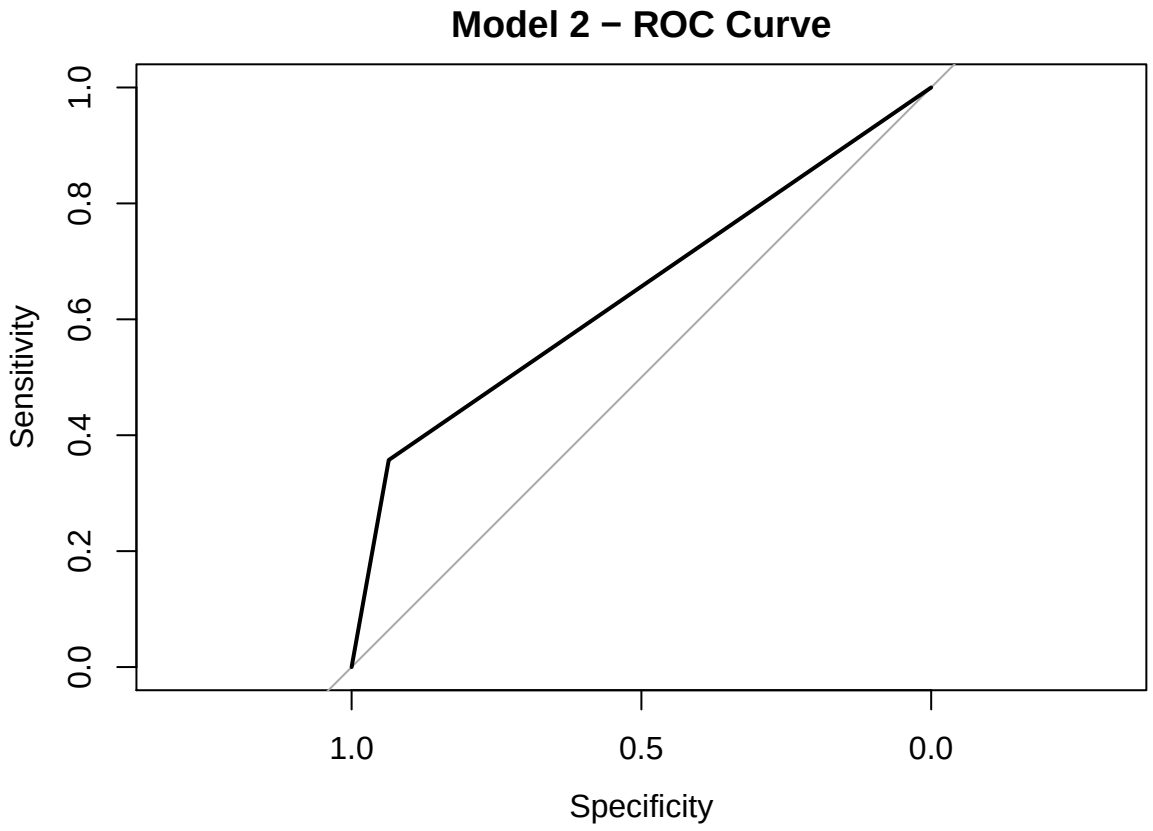
ROC Curve

Another way to look at the models is through the ROC curve. The ROC curve comapres the sensitivity of the model with the specificity of the model. It bascially give the performance of the model. With that curve, we can cacluate the AUC (Area Under the Curve). The higher this number is, the better the performance of the model. We can see that model 1 is still the best choice from all three models.

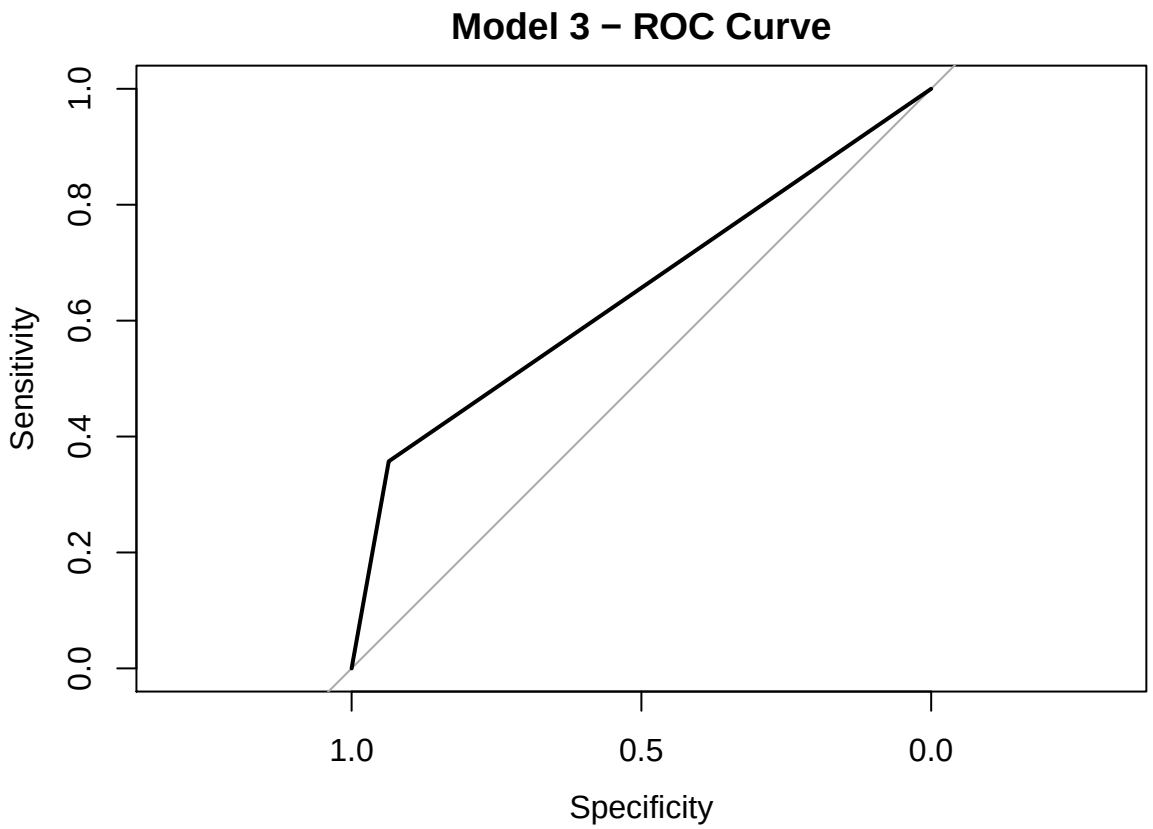
```
train3 <- cbind(train1 , answer1a[1:4513], answer, answer2)

rc1 <- roc(factor(TARGET_FLAG) ~ answer1a[1:4513], data=train3)
rc2 <- roc(factor(TARGET_FLAG) ~ answer, data=train3)
rc3 <- roc(factor(TARGET_FLAG) ~ answer2, data=train3)

plot(rc1,main='Model 1 - ROC Curve')
```




```
plot(rc2,main='Model 2 - ROC Curve')
```




```
plot(rc3,main='Model 3 - ROC Curve')
```




```
model <- c('Model 1', 'Model 2', 'Model 3')
area <- c(auc(train1$TARGET_FLAG, answer1a),auc(train1$TARGET_FLAG, answer),auc(train1$TARGET_FLAG, answer2))
df <- data.frame(Model=model,AUC=area)
df
```

```
##     Model       AUC
## 1 Model 1 0.7136636
## 2 Model 2 0.6465510
## 3 Model 3 0.6465510
```

model	AUC (Area Under Curve)
model 1	.7283561
model 2	.6572123
model 3	.6572123

AIC/BIC/Log-Likelihood

The Akaike information criterion (AIC) is a measure of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Hence, AIC provides a means for model selection. The lower the AIC the better the model.

Bayesian information criterion (BIC), is a criterion for model selection among a finite set of models. The lower the BIC, the better the model. Ghand and hand with AIC.

We can see that the smallest AIC and BIC is still model number 1. All of the criteria is pointiong towards model number 1. That means that the best model, out of these three would be model number 1 to continue on for the evaluation set.

model	AIC	BIC
model 1	3188.62	3258.47
model 2	2900.54	4021.20
model 3	82733.65	82835.25

5.2 - TARGET_AMT

We first check the summary stats with the two models. The first this we check is the MSE (Mean Squared Error). This is the mean of the residuals (actual - predicted) squared. It is a good way to see how accurate your model is. A smaller MSE is always good. The next things is the R squared. This is usually called the goodness of fit. The higher the R squared value the better the model is. The last thing is the F-Stat. It is most often used when comparing statistical models that have been fitted to a data set, in order to identify the model that best fits the population from which the data were sampled.

All three of these parameters are showing that the best model to use would be model 1. It has the lowest MSE, the highest R squared and the highest F-stat (which means a lower alpha value and most statistically relevant).

model	MSE (Mean Squared Error)	R Squared	F-Stat
model 1	18051149	.06530975	86.13564
model 2	18329321	.05090597	40.27218

Now we can look at the residuals and see how well they fit to the regression line. In figure 4.1, we look at the first model. Overall, the data does not fit a linear model at all. The histograms of the residual (lower left have corner) are heavily skewed to the right. Almost all of the data is clustered around 0. Next, the normal plot (top left corner) has a pretty obvious bend to it. The line is supposed to be pretty straing. Finally the residual plot (top right corner) show a pretty obvious patter amoungst the points. It is supposed to be random. This is showing that the data is really not lending itself to a linear model. That means that a different model may be a better choice. In figure 4.2, we see the exact same patterns as in Figure 4.1. This data does not lend itself to a linear model.

Figure 4.1

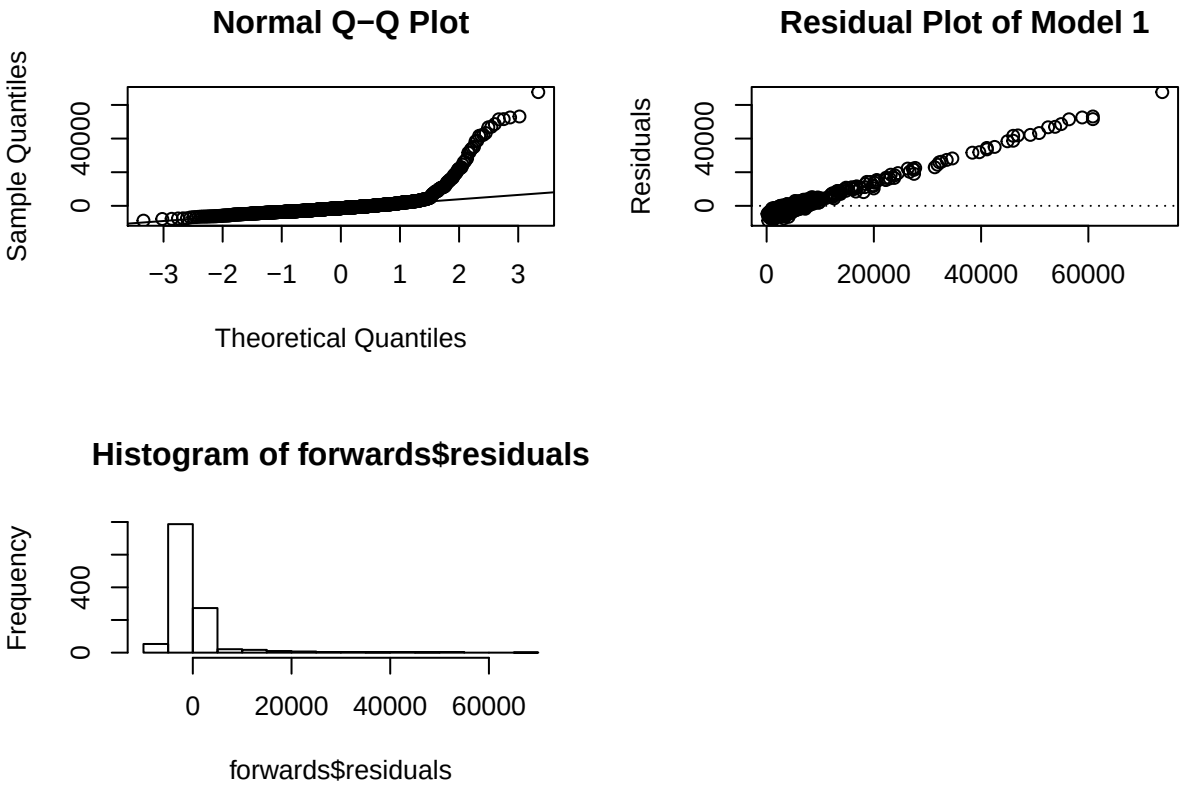
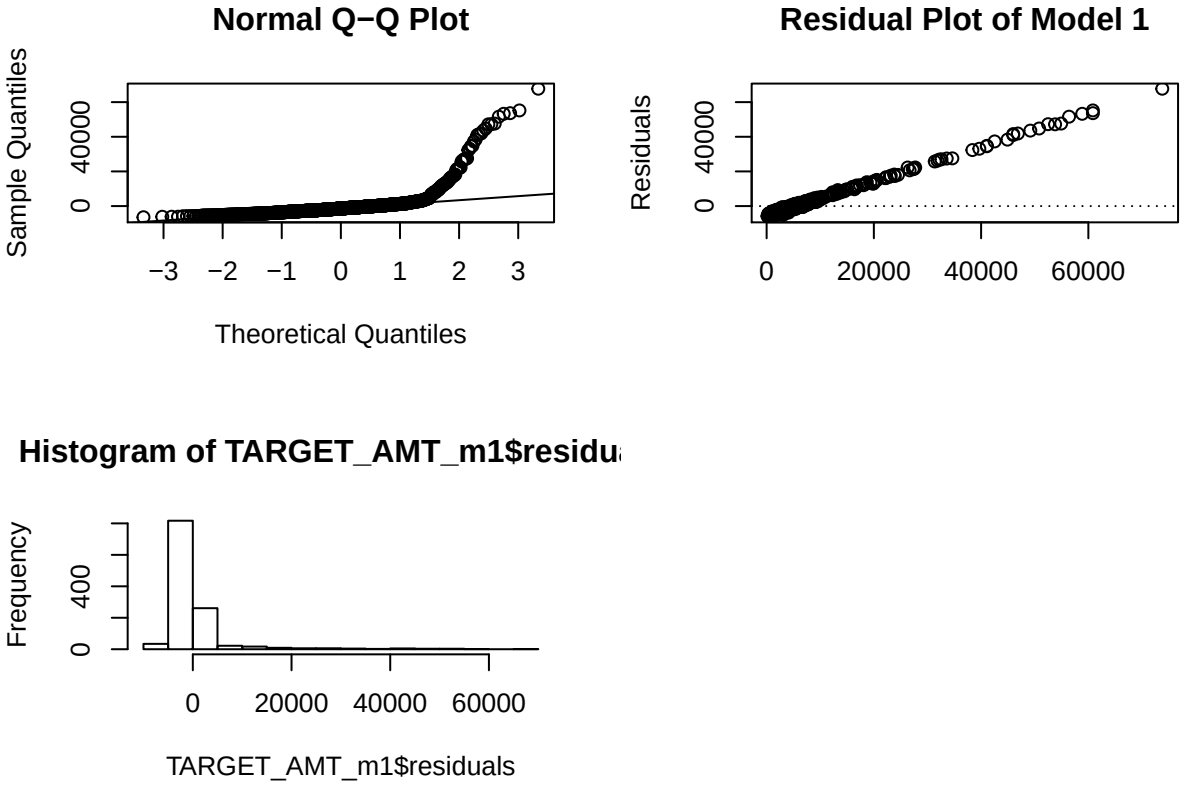


Figure 4.2



- Overal, model 1 is the best choice. The data may not lend itself to a linear model, but it is a better choice then model 2.
- That means, to predict the `TARGET_FLAG`, we will be using model 1 and to predict the `TARGET_AMT` we will also be using model 1.

6. Model Evaluation

The final flag values and the amounts can be seen in the final_result.csv

```
## final_answer
##    0    1
## 1706    9
```

7. Appendix A

```
library(ggplot2)
library(gridExtra)
library(knitr)
library(stringr)
library(dplyr)
library(lattice)
library(tidyr)
library(corrplot)
library(pROC)
library(aod)
library(Rcpp)
library(Amelia)
library(gam)
library(pscl)
library(ROCR)
library(gmodels)
library(PerformanceAnalytics)
library(rpart)
library(MuMIn)
library(Metrics)

train_url<-"https://raw.githubusercontent.com/hovig/MSDS_CUNY/master/DATA621/hw4/insurance_training_data.csv"
data<-read.csv(train_url)
eval_url<-"https://raw.githubusercontent.com/hovig/MSDS_CUNY/master/DATA621/hw4/insurance-evaluation-data.csv"
dataeval<-read.csv(eval_url)

missmap(data, legend = TRUE, main = "Missing Values vs Observed", col =  c("white", "black"))
summary(data)
summary(data$TARGET_FLAG)
hist(data$TARGET_FLAG)
summary(data$TARGET_AMT)
hist(data$TARGET_AMT)

blue_book <- unname(sapply(data$BLUEBOOK, str_replace_all, '[,$]', ''))
blue_book <- as.numeric(blue_book)

income <- unname(sapply(data$INCOME, str_replace_all, '[,$]', ''))
income <- as.numeric(income)

home_val <- unname(sapply(data$HOME_VAL, str_replace_all, '[,$]', ''))
home_val <- as.numeric(home_val)

old_claim <- unname(sapply(data$OLDCLAIM, str_replace_all, '[,$]', ''))
old_claim <- as.numeric(old_claim)

data$BLUEBOOK <- blue_book
data$INCOME <- income
data$HOME_VAL <- home_val
data$OLDCLAIM <- old_claim

data2 <- data[,-c(1,2,3,9,11,12,13,14,16,19,20,23,26)]
chart.Correlation(data2)

the_cor <- cor(data[sapply(data, is.numeric)])
corrplot(the_cor, method = "circle")

data <- data[complete.cases(data),]
```

```r
data <- data[data$CAR_AGE >= 0,]

data$PARENT1 <- ifelse(data$PARENT1 == "No", 1, 0)
data$SEX <- ifelse(data$SEX == 'M', 0, 1)
data$CAR_USE <- ifelse(data$CAR_USE == 'Commercial', 0, 1)
data$MSTATUS <- ifelse(data$MSTATUS == 'Yes', 0, 1)
data$RED_CAR <- ifelse(data$RED_CAR == "no", 0, 1)
data$EDUCATION <- ifelse(data$EDUCATION %in% c('PhD', "Masters"),0, 1)
data$REVOKED <- ifelse(data$REVOKED == "No", 0, 1)
data$URBANICITY <- ifelse(data$URBANICITY == "Highly Urban/ Urban", 1, 0)
data$JOB <- ifelse(data$JOB %in% c('Professional', 'Manager', 'Student', 'Lawyer'), 1, 0)
data$CAR_TYPE <- ifelse(data$CAR_TYPE %in% c('Panel Truck', "Pickup", "Sports Car"), 1, 0)

data <- data[,-1]
data <- data[sample(nrow(data)),]
top <- round(.70 * NROW(data))

train1 <- data[1:top,]
test1 <- data[(top + 1):NROW(data),]

training_2a <- dplyr::select(train1, -c(KIDSDRIV, HOMEKIDS, EDUCATION, JOB, TIF, CAR_TYPE, OLDCLAIM, CLM_FREQ,
M11 <- lm( TARGET_FLAG~ .-TARGET_FLAG, data=training_2a)
M12 <- update(M11,.~.-AGE-HOMEKIDS-YOJ-INCOME-SEX-EDUCATION-BLUEBOOK-RED_CAR-OLDCLAIM-CLM_FREQ)
TARGET_FLAG_m1 <- M12
summary(TARGET_FLAG_m1)

answer1a <- predict(TARGET_FLAG_m1, type = "response")
answer1a <- ifelse(answer1a <.5, 0, 1)

fullmod <- glm(TARGET_FLAG ~ KIDSDRIV + AGE + HOMEKIDS + YOJ + INCOME + PARENT1 + HOME_VAL + MSTATUS + SEX + E

backwards <- step(fullmod, trace = 0)
prediction <- round(predict(backwards, type = 'response'), 4)

answer <- ifelse(prediction < .5, 0 ,1)
summary(backwards)

nothing <- glm(TARGET_FLAG ~ 1, data = train1, family = binomial(link = 'probit'))
forwards <- step(nothing, scope = list(lower=formula(nothing), upper=formula(fullmod)), direction = "forward",

pred <- round(predict(forwards, type = 'response'), 4)
answer2 <- ifelse(pred < .5, 0 ,1)
summary(forwards)

data <- data[sample(nrow(data)),]
top <- round(.70 * NROW(data))

train2 <- data[1:top,]
test2 <- data[(top + 1):NROW(data),]

training_2a <- dplyr::select(train2, -c(KIDSDRIV, HOMEKIDS, EDUCATION, JOB, TIF, CAR_TYPE, OLDCLAIM, CLM_FREQ,
M11 <- lm(TARGET_AMT~ .-TARGET_FLAG, data=training_2a)
M12 <- update(M11,.~.-AGE-HOMEKIDS-YOJ-INCOME-SEX-EDUCATION-BLUEBOOK-RED_CAR-OLDCLAIM-CLM_FREQ)
TARGET_AMT_m1 <- M12

pred5 <- predict(TARGET_AMT_m1)
summary(TARGET_AMT_m1)

nothing <- lm(TARGET_AMT ~ 1, data = train2)
```

```r
forwards <- step(nothing, scope = list(lower=formula(nothing), upper=formula(fullmod)), direction = "forward",

pred4 <- predict(forwards)
summary(forwards)

t1 <- table(Actual = train1$TARGET_FLAG, Predicted = answer1a);t1
t2 <- table(Actual = train1$TARGET_FLAG, Predicted = answer);t2
t3 <- table(Actual = train1$TARGET_FLAG, Predicted = answer2);t3

fscore <- function(x)
{
  precision <- x[1]/(x[1] + x[4])
  recall <- x[1] / (x[1] + x[3])
  score <- (2*precision*recall)/(precision+recall)

  return(c(precision, recall, score))
}

model1 <- fscore(t1)
model2 <- fscore(t2)
model3 <- fscore(t3)

train3 <- cbind(train1 , answer1a[1:4231], answer, answer2)

rc1 <- roc(factor(TARGET_FLAG) ~ answer1a[1:4231], data=train3)
rc2 <- roc(factor(TARGET_FLAG) ~ answer, data=train3)
rc3 <- roc(factor(TARGET_FLAG) ~ answer2, data=train3)

plot(rc1,main='Model 1 - ROC Curve')
plot(rc2,main='Model 2 - ROC Curve')
plot(rc3,main='Model 3 - ROC Curve')

model <- c('Model 1', 'Model 2', 'Model 3')
area <- c(auc(train1$TARGET_FLAG, answer1a),auc(train1$TARGET_FLAG, answer),auc(train1$TARGET_FLAG, answer2))
df <- data.frame(Model=model,AUC=area)
df

# AIC = Akaike Information Criterion
AIC.1 <- AIC(TARGET_FLAG_m1)
AIC.2 <- AIC(backwards)
AIC.3 <- AIC(forwards)
AIC <- rbind(AIC.1, AIC.2, AIC.3) %>% round(2)

# BIC = Bayesian information criterion
BIC.1 <- BIC(TARGET_FLAG_m1)
BIC.2 <- BIC(backwards)
BIC.3 <- BIC(forwards)
BIC <- rbind(BIC.1, BIC.2, BIC.3) %>% round(2)

model1MSE <- Metrics::mse(train2$TARGET_AMT, pred4)
model2MSE <- Metrics::mse(train2$TARGET_AMT, pred5)

col_mse <- c(model1MSE,model2MSE)
col_r_sq <- c(summary(forwards)$r.squared, summary(TARGET_AMT_m1)$r.squared)
col_f_stat <- c(summary(aov(forwards))[[1]]$F[1], summary(aov(TARGET_AMT_m1))[[1]]$F[1])

par(mfrow = c(2,2))
g1<- qqnorm(forwards$residuals)
g2 <- qqline(forwards$residuals)
```

```
g3 <-plot(forwards$residuals ~ train2$TARGET_AMT,
     xlab='',
     ylab='Residuals',
     main='Residual Plot of Model 1')
abline(h=0,lty=3)
g4 <- hist(forwards$residuals)

par(mfrow = c(2,2))
g1<- qqnorm(TARGET_AMT_m1$residuals)
g2 <- qqline(TARGET_AMT_m1$residuals)
g3 <-plot(TARGET_AMT_m1$residuals ~ train2$TARGET_AMT,
     xlab='',
     ylab='Residuals',
     main='Residual Plot of Model 1')
abline(h=0,lty=3)
g4 <- hist(TARGET_AMT_m1$residuals)

dataeval <- dataeval[,-c(1)]

blue_book <- unname(sapply(dataeval$BLUEBOOK, str_replace_all, '[,$]', ''))
blue_book <- as.numeric(blue_book)

income <- unname(sapply(dataeval$INCOME, str_replace_all, '[,$]', ''))
income <- as.numeric(income)

home_val <- unname(sapply(dataeval$HOME_VAL, str_replace_all, '[,$]', ''))
home_val <- as.numeric(home_val)

old_claim <- unname(sapply(dataeval$OLDCLAIM, str_replace_all, '[,$]', ''))
old_claim <- as.numeric(old_claim)

dataeval$BLUEBOOK <- blue_book
dataeval$INCOME <- income
dataeval$HOME_VAL <- home_val
dataeval$OLDCLAIM <- old_claim
dataeval$TARGET_FLAG <- rep(0, NROW(dataeval))
dataeval$TARGET_AMT <- rep(0, NROW(dataeval))
dataeval <- dataeval[complete.cases(dataeval),]
dataeval <- dataeval[dataeval$CAR_AGE >= 0,]
dataeval$PARENT1 <- ifelse(dataeval$PARENT1 == "No", 1, 0)
dataeval$SEX <- ifelse(dataeval$SEX == 'M', 0, 1)
dataeval$CAR_USE <- ifelse(dataeval$CAR_USE == 'Commercial', 0, 1)
dataeval$MSTATUS <- ifelse(dataeval$MSTATUS == 'Yes', 0, 1)
dataeval$RED_CAR <- ifelse(dataeval$RED_CAR == "no", 0, 1)
dataeval$EDUCATION <- ifelse(dataeval$EDUCATION %in% c('PhD', "Masters"),0, 1)
dataeval$REVOKED <- ifelse(dataeval$REVOKED == "No", 0, 1)
dataeval$URBANICITY <- ifelse(dataeval$URBANICITY == "Highly Urban/ Urban", 1, 0)
dataeval$JOB <- ifelse(dataeval$JOB %in% c('Professional', 'Manager', 'Student', 'Lawyer'), 1, 0)
dataeval$CAR_TYPE <- ifelse(dataeval$CAR_TYPE %in% c('Panel Truck', "Pickup", "Sports Car"), 1, 0)

predict_eval_target_flag <- predict(TARGET_FLAG_m1, newdata = dataeval, type = 'response')
final_answer <- ifelse(predict_eval_target_flag <.5, 0, 1)

predict_eval_taget_amt <- predict(TARGET_AMT_m1, newdata = dataeval)

everything <- cbind(Flag = final_answer, Amount = predict_eval_taget_amt)

table(final_answer)
write.csv(everything, file = "results.csv")
```