# DATA 621 - Business Analytics and Data Mining

Fall 2020 - Group 2 - Homework #4

Avraham Adler, Samantha Deokinanan, Amber Ferger, John Kellogg, Bryan Persaud, Jeff Shamp

11/08/2020

## Contents

# Introduction

The assignment for HW4 is to analyze and model a dataset containing approximately 8000 records representing customers of an auto insurance company. Each record has two response variables. The first response variable, `TARGET_FLAG`, is a 1 or a 0. where a `1` means that the person was in a car and a `0` means that the person was not in a car crash. The second response variable is `TARGET_AMT`. This value is 0 if the person did not crash their car. However, if they did crash their car, this number will be a value greater than zero.

The objective of the assignment is to build multiple linear regression and binary logistic regression models on the training data to predict **both** the probability that a person will crash their car and the amount of money it will cost given a crash. Only the variables in the dataset, or variables directly derived from them, may be used.

# DATA EXPLORATION

## Variables

The data is composed of the following variables:

| VARIABLE NAME | DEFINITION | THEORETICAL EFFECT |
|---|---|---|
| INDEX | Identification Variable (do not use) | None |
| TARGET_FLAG | Was Car in a crash? 1=YES 0=NO | None |

| VARIABLE NAME | DEFINITION | THEORETICAL EFFECT |
|---|---|---|
| TARGET_AMT | If car was in a crash, what was the cost | None |
| AGE | Age of Driver | Very young people tend to be risky. Maybe very old people also. |
| BLUEBOOK | Value of Vehicle | Unknown effect on probability of collision, but probably effect the payout if there is a crash |
| CAR_AGE | Vehicle Age | Unknown effect on probability of collision, but probably effect the payout if there is a crash |
| CAR_TYPE | Type of Car | Unknown effect on probability of collision, but probably effect the payout if there is a crash |
| CAR_USE | Vehicle Use | Commercial vehicles are driven more, so might increase probability of collision |
| CLM_FREQ | # Claims (Past 5 Years) | The more claims you filed in the past, the more you are likely to file in the future |
| EDUCATION | Max Education Level | Unknown effect, but in theory more educated people tend to drive more safely |
| HOMEKIDS | # Children at Home | Unknown effect |
| HOME_VAL | Home Value | In theory, home owners tend to drive more responsibly |
| INCOME | Income | In theory, rich people tend to get into fewer crashes |
| JOB | Job Category | In theory, white collar jobs tend to be safer |
| KIDSDRIV | # Driving Children | When teenagers drive your car, you are more likely to get into crashes |
| MSTATUS | Marital Status | In theory, married people drive more safely |
| MVR_PTS | Motor Vehicle Record Points | If you get lots of traffic tickets, you tend to get into more crashes |
| OLDCLAIM | Total Claims (Past 5 Years) | If your total payout over the past five years was high, this suggests future payouts will be high |
| PARENT1 | Single Parent | Unknown effect |
| RED_CAR | A Red Car | Urban legend says that red cars (especially red sports cars) are more risky. Is that true? |
| REVOKED | License Revoked (Past 7 Years) | If your license was revoked in the past 7 years, you probably are a more risky driver. |
| SEX | Gender | Urban legend says that women have less crashes then men. Is that true? |
| TIF | Time in Force | People who have been customers for a long time are usually more safe. |
| TRAVTIME | Distance to Work | Long drives to work usually suggest greater risk |
| URBANICITY | Home/Work Area | Unknown |
| YOJ | Years on Job | People who stay at a job for a long time are usually more safe |

There are 8161 observations of 11 numeric predictor variables and 10 factor predictor variables.

## Missing Data

There are missing observations. Specifically, the following variables have missing values coded as `NA`:

Table 2: Variables with Count of Missing Values > 0

| AGE | YOJ | INCOME | HOME_VAL | CAR_AGE |
|---|---|---|---|---|
| 6 | 454 | 445 | 464 | 510 |

Another concern is that the following variables should not have significant probabilities of 0 because they represent values which *should* be positive:

- AGE
- YOJ
- INCOME
- HOME_VAL
- TRAVTIME
- BLUEBOOK
- TIF
- CAR_AGE

Table 3: Variables with Count of Missing Values > 0

| YOJ | INCOME | HOME_VAL | CAR_AGE |
|---|---|---|---|
| 625 | 615 | 2294 | 3 |

The few missing `CAR_AGE` values should be easy to impute. The other heavy-0 observations are of more concern. Over 25% of the `HOME_VAL` observations are 0! Now it may be that those with `HOME_VAL` of 0 do not own homes but rent, and it may be that those with `YOJ` and `INCOME` of 0 were unemployed at the time of the data collection. There are 581 observations with both `YOJ` and `INCOME` of 0, meaning there are a few dozen of each which have values in one class and not the other. Handling this will be addressed in the DATA PREPARATION section.

## Summary Statistics and Graphs

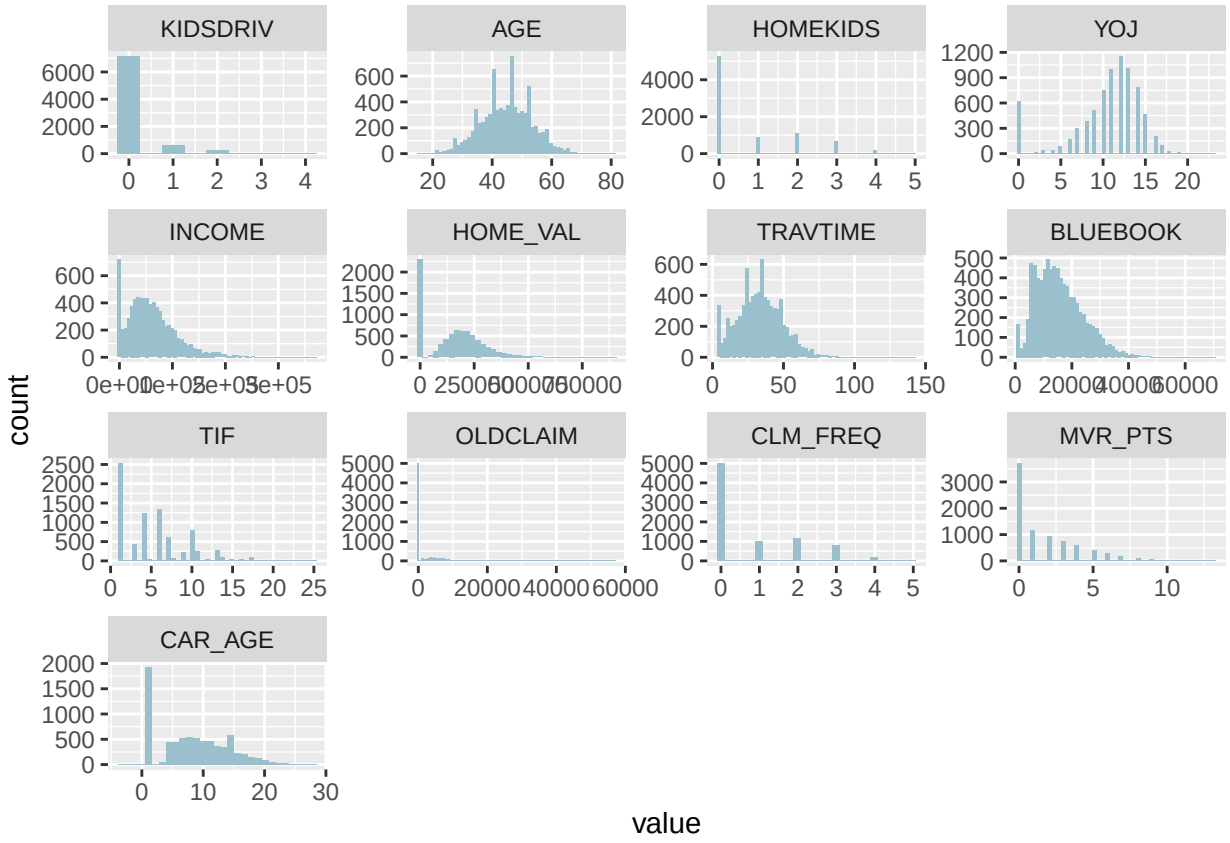
### Numeric Predictors

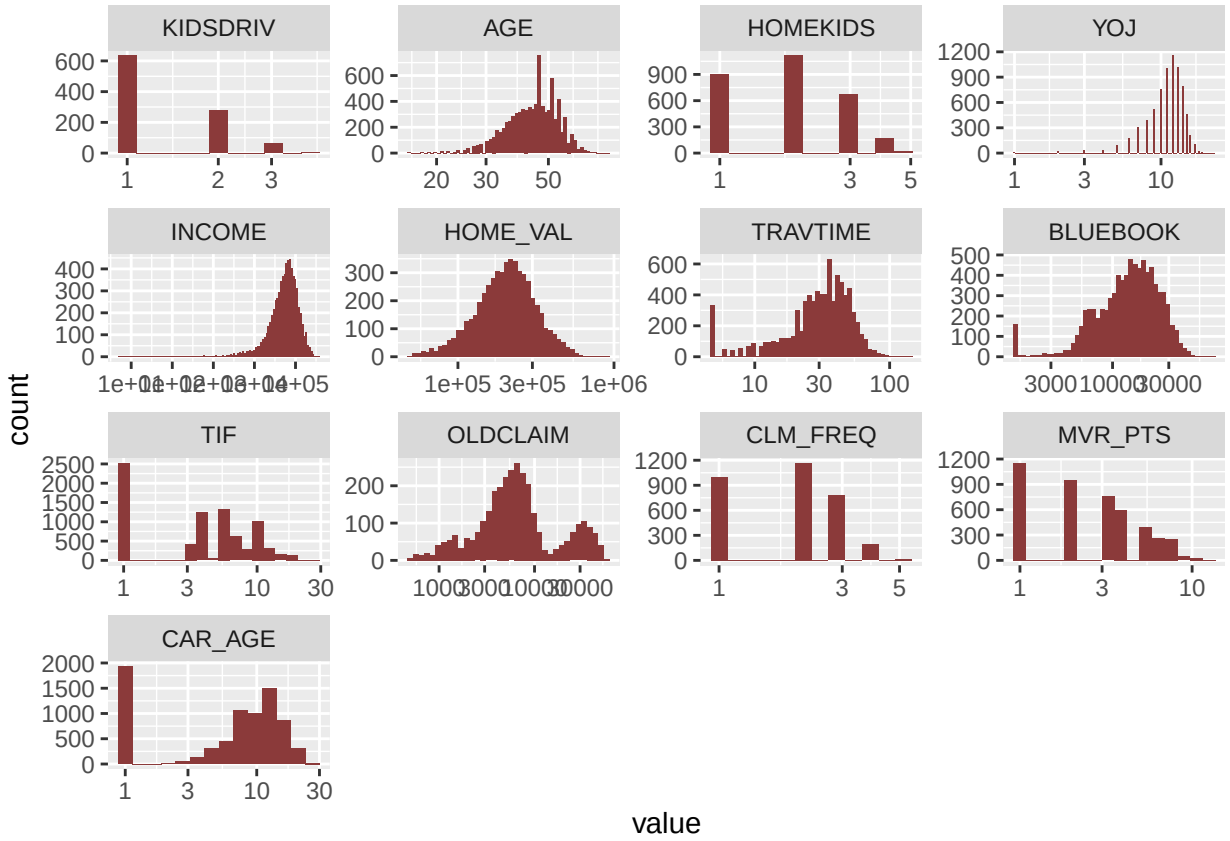
The numeric predictor variables have the following summary statistics, ignoring missing values:

Table 4: Summary Statistitics for Numeric Variables

| metric | Mean | SD | Min | Q1 | Median | Q3 | Max | IQR |
|---|---|---|---|---|---|---|---|---|
| KIDSDRIV | 0.171 | 0.512 | 0 | 0 | 0 | 0 | 4 | 0 |
| AGE | 44.790 | 8.628 | 16 | 39 | 45 | 51 | 81 | 12 |
| HOMEKIDS | 0.721 | 1.116 | 0 | 0 | 0 | 1 | 5 | 1 |
| YOJ | 10.499 | 4.092 | 0 | 9 | 11 | 13 | 23 | 4 |
| INCOME | 61898.095 | 47572.683 | 0 | 28097 | 54028 | 85986 | 367030 | 57889 |
| HOME_VAL | 154867.290 | 129123.775 | 0 | 0 | 161160 | 238724 | 885282 | 238724 |
| TRAVTIME | 33.486 | 15.908 | 5 | 22 | 33 | 44 | 142 | 22 |
| BLUEBOOK | 15709.900 | 8419.734 | 1500 | 9280 | 14440 | 20850 | 69740 | 11570 |
| TIF | 5.351 | 4.147 | 1 | 1 | 4 | 7 | 25 | 6 |
| OLDCLAIM | 4037.076 | 8777.139 | 0 | 0 | 0 | 4636 | 57037 | 4636 |
| CLM_FREQ | 0.799 | 1.158 | 0 | 0 | 0 | 2 | 5 | 2 |
| MVR_PTS | 1.696 | 2.147 | 0 | 0 | 1 | 3 | 13 | 3 |
| CAR_AGE | 8.328 | 5.701 | -3 | 1 | 8 | 12 | 28 | 11 |

A visual depiction of the distributions of the numeric predictors follows. The blue graphs are on a normal scale and the red ones are on a $\log_{10}$ scale.



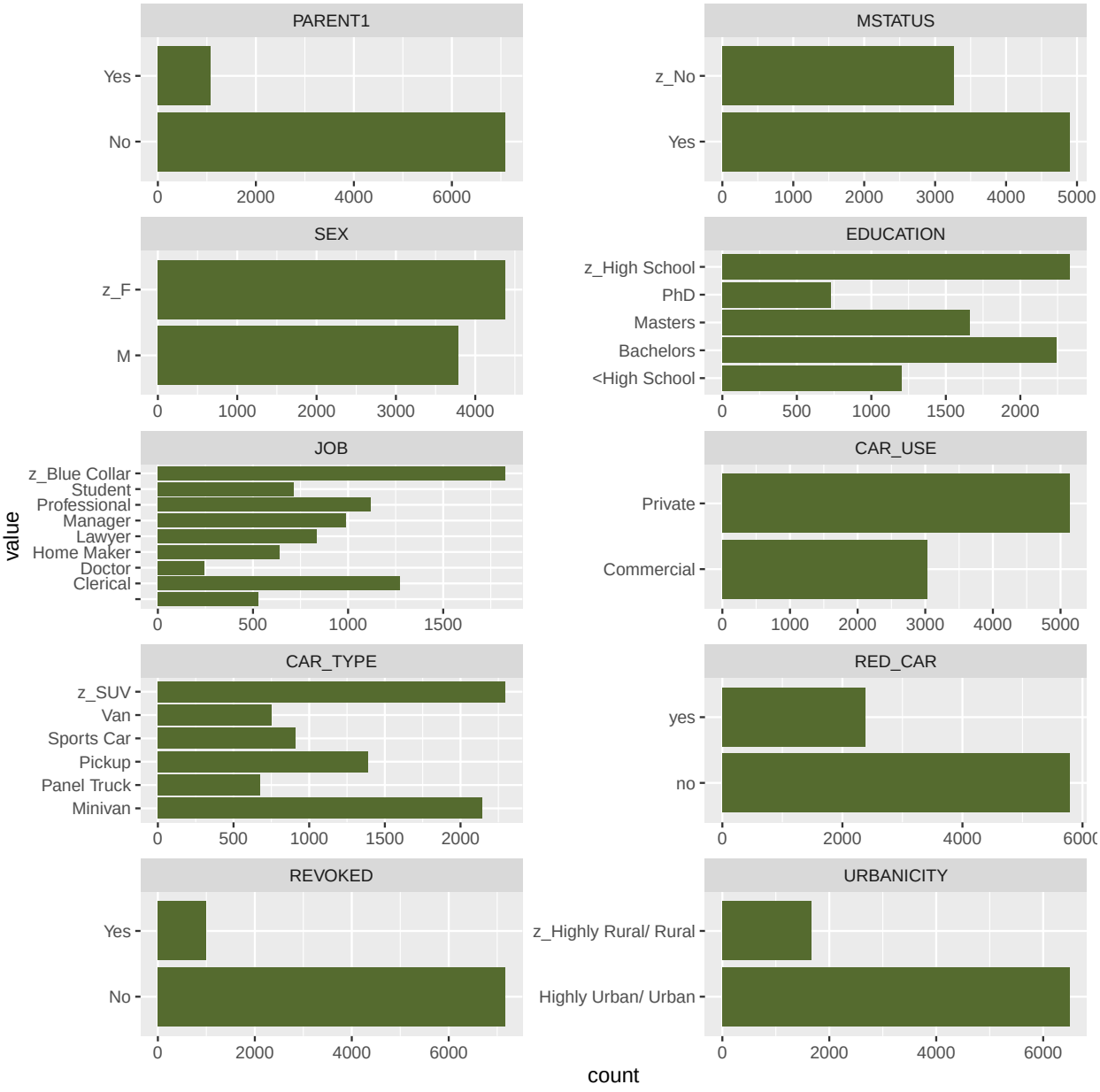



What the above visualizations show is that there are two sets of numeric variables. There are those which are clearly discrete, taking one of a few integral values, and there are those which are more continuous. Of the continuous variables, some like `AGE` look rather Gaussian at first glance, while others, such as `HOME_VAL` or `INCOME` are skewed. However, `HOME_VAL` looks lognormal, as its histogram on the log scale is near-Gaussian.

A variable such as `OLDCLAIM` is expected to have a mass point at 0. Every observation that had fewer than 2 claims clearly did not have a prior claim!

### Factor Predictors

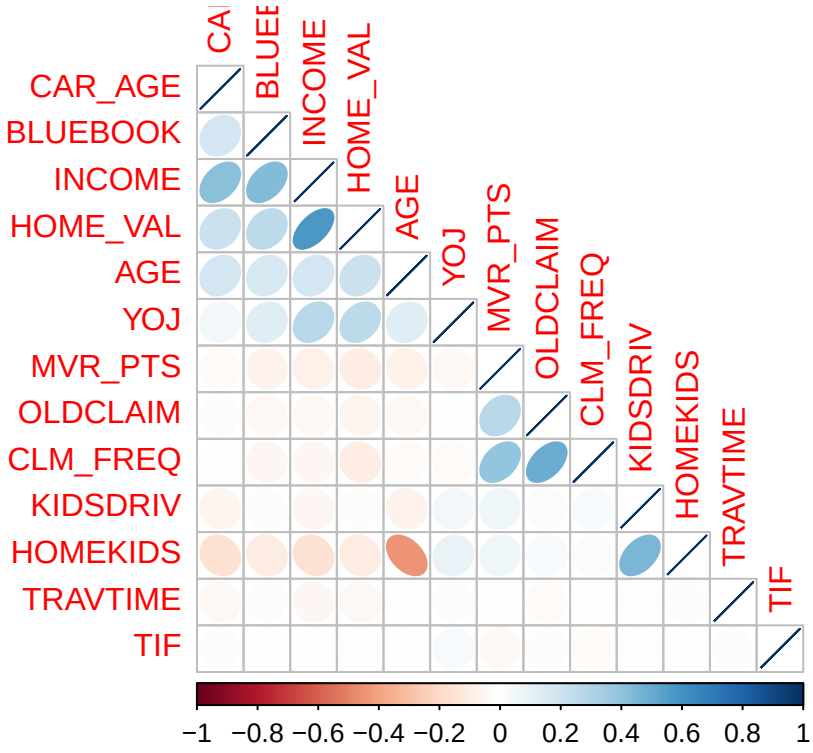
While a tabular depiction of factor variables is usually of little value, visual representation of their distributions is of use.




In order to use regression-based tools, we are going to have to convert these factors to dummy variables. Another important observation is that there are rows for which `JOB` is blank and some strange value names for some of the factors. Cleaning this up will be handled in the DATA PREPARATION section.

### Correlations

The corrgram below graphically represents the correlations between the numeric predictor variables, when ignoring the missing variables.




Most of the numeric variables are uncorrelated with one another. A few exceptions exist:

- `HOMEKIDS` with `KIDSDRIV`
  - This is self-explanatory.
- `HOME_VAL` with `INCOME`
  - Both are very strongly tied to wealth and net worth.
- `BLUEBOOK` with`INCOME`
  - People with more disposable income can afford more expensive vehicles.
- `CLM_FREQ` with `OLDCLAIM`
  - More claims implies a greater total aggregate payment.
- `CAR_AGE` with `INCOME`
  - This is interesting. It could be that people with higher disposable incomes buy more expensive cars and keep them for longer, but that's a guess.
- `MVR_PTS` is *lightly* correlated with `CLM_FREQ` and `OLDCLAIM`
  - Probably when one is in a lot of accidents, the chances of them being at some level of fault rises.
- `HOMEKIDS` **negatively** correlated with `AGE`
  - As people age, their children age as well and eventually move out, we hope!

# DATA PREPARATION

## Data Value Cleanup

The first step will be to clean up the strange values and blank `JOB` entries.

## Training & Testing Split

While the following discussion more properly occurs under BUILD MODELS, as it factors into how the data is prepared, it will happen here. All the models will be trained on the same approximately 70% of the training set, reserving 30% for validation of which model to select for the frequency and severity estimation on the supplied evaluation set.

However, there will need to be two sets of training and testing, as the severity model must perforce be trained on data for which a claim occurred. To maintain integrity, the split of all the data for frequency will be honored for severity. The 70/30 split will be honored as all claims with a `TARGET_FLAG` of 1 will have non-zero `TARGET_AMT`. However, there will be no guarantee that the *distribution* of the loss conditional on the event in the validation set will be similar to the training set as a whole.

## Model Set #1

### Missing Data

The issues with factor data were handled universally for all models through resetting the levels. What remains is handling the missing numeric values. For `CAR_AGE` simple k-means imputation will be used. The remaining three problem variables, `YOJ`, `INCOME`, and `HOME_VAL` have both `NA` and 0 issues.

For `YOJ` and `INCOME`, it is not unreasonable to consider 0 to be realistic: unemployed or employed less than half a year. As such, we will leave the 0's and impute the NAs.

We will take a different approach with `HOME_VAL`. As seen above when the 0 values are removed, `HOME_VAL` has a nice lognormal shape. It is also reasonable to consider that people responding with 0 for `HOME_VAL` are non-owners as opposed to it being missing, although we would want to investigate this further were we allowed. Therefore, we are going to add an "Own" variable which will be `Yes` for all positive `HOME_VAL` and `No` otherwise, and allow an interaction between this value and the actual value. This would be much more effective in a decision tree framework, but that is not available.

## Model Set #2

### Missing Data

Like Model Set #1, the data preparation efforts for Model Set #2 also includes the imputation of missing values for the predictor variables. However, because it combines the standard linear regression and the nearest-neighbor imputation approach, Model Set #2 elected to fill in missing values of a continuous variable by using the predictive mean matching imputation method. Moreover, there will be the derivation of three new binary factor variables.

- `AGE`
  - Since the data set is comprised of auto insurance policyholder information, the missing `AGE` values required imputation. However, given the relatively small number of missing values, imputation by predictive mean matching is conducted.
- `HOME_VAL`
  - A `HOME_VAL` response of '0' makes it understandable to assume that those missing entries could indicate that the policyholder is a renter rather than a homeowner. Therefore, missing values were replaced with zeroes. The new `OWN` variables created in Model Set #1 is kept in Model Set #2.
- `CAR_AGE`, `INCOME` & `YOJ`
  - Because there is a larger number of missing values (NA's) were identified, imputation by predictive mean matching is conducted.
  - Moreover, there is a single negative value ('-3') for `CAR_AGE` which seemed implausible and was assumed to be a typographical error. Thus, the absolute value of `CAR_AGE` is taken.

### New Variables

For Model Set #2, new factor variables ($VARIABLE.`f`) were create as a way to better capture policyholder's profile in the binary logistic regression modeling process.

- `CAR_AGE.f`
  - Because `CAR_AGE` is dominated by values of 1 (i.e. car being one year old or less), the `CAR_AGE.f` variable is derived as a factor variable to be used during the binary logistic regression modeling process. Therefore, it is interpreted as:

    * CAR_AGE.f = 0: The vehicle is 1 year old or less.
    * CAR_AGE.f = 1: The vehicle is more than a 1-year-old.
- HOMEKIDS.f & KIDSDRIV.f
  - Both the HOMEKIDS and KIDSDRIV variables indicated that large percentages of the values of each were 0. Therefore, both variables were converted to binary variables. The following is their binary interpretations:
    * HOMEKIDS.f = 0: The policyholder has no child living at home.
    * HOMEKIDS.f = 1: The policyholder has 1 or more children living at home.
    * KIDSDRIV.f = 0: No teenage drivers use the policyholder's insured vehicle.
    * KIDSDRIV.f = 1: One or more teenage drivers use the policyholder's insured vehicle.

## Model Set #3

### Missing Data

Similar to model 2, model set 3 deals with missing numerical data by imputing them using the Multivariate imputation by chained equations (MICE) method. Multiple imputation involves creating multiple predictions for each missing value, helping to account for the uncertainty in the individual imputations.

One thing to note about Model 3 is that it does *not* assume that an NA in the HOME_VAL variable is indicative of renting. This is a key difference between this model and the previous two models.

# BUILD MODELS

## Model Set #1

### Dummy Variables

As there are many factor variables, dummy variables will be created. The target variables will be removed and held to the side so as not to contaminate the fitting.

### Frequency Model

The frequency model will be a binary logistic on the prepared data. It will start with the complete model and allow for some interactions. It will use both forward and backward steps to identify which variables to include, using AIC as the test metric. Using this approach does not allow for cross-validation. The initial interactions considered will be:

- Three-way interaction between OWN, HOME_VAL and INCOME
  - Represents wealth
- Three-way interaction between CLM_FREQ, OLDCLAIM, and MVR_PTS
  - Represents a prior history
- CAR_USE with URBANICITY
  - Represents increased risk of location

**Training** At this point, the code will be set up to preprocess the data and train the model. The two pre-processing steps which will be done will be to check for near-zero variance between the predictors and then use kNN imputations for missing values. Even though the caret package allows the preprocessing to occur in the call to train, it will be separated here for clarity. The actual code may be found in the CODE APPENDIX.

Table 5: Model 1 Frequency Regression Output

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -1.422 | 0.041 | -34.265 | 0.000 |
| KIDSDRIV | 0.193 | 0.033 | 5.773 | 0.000 |

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| AGE | -0.069 | 0.038 | -1.827 | 0.068 |
| INCOME | -0.229 | 0.069 | -3.323 | 0.001 |
| PARENT1.Yes | 0.145 | 0.040 | 3.618 | 0.000 |
| MSTATUS.Yes | -0.173 | 0.047 | -3.703 | 0.000 |
| SEX.F | -0.109 | 0.059 | -1.837 | 0.066 |
| EDUCATION.Bachelors | -0.157 | 0.039 | -4.027 | 0.000 |
| JOB.Clerical | 0.261 | 0.052 | 4.999 | 0.000 |
| JOB.HomeMaker | 0.170 | 0.050 | 3.392 | 0.001 |
| JOB.Lawyer | 0.114 | 0.046 | 2.477 | 0.013 |
| JOB.Professional | 0.211 | 0.046 | 4.550 | 0.000 |
| JOB.Student | 0.153 | 0.052 | 2.932 | 0.003 |
| JOB.BlueCollar | 0.305 | 0.052 | 5.872 | 0.000 |
| TRAVTIME | 0.247 | 0.036 | 6.907 | 0.000 |
| BLUEBOOK | -0.160 | 0.052 | -3.068 | 0.002 |
| TIF | -0.220 | 0.036 | -6.051 | 0.000 |
| CAR_TYPE.PanelTruck | 0.138 | 0.052 | 2.665 | 0.008 |
| CAR_TYPE.Pickup | 0.190 | 0.045 | 4.269 | 0.000 |
| CAR_TYPE.SportsCar | 0.328 | 0.049 | 6.706 | 0.000 |
| CAR_TYPE.Van | 0.153 | 0.043 | 3.540 | 0.000 |
| CAR_TYPE.SUV | 0.360 | 0.059 | 6.089 | 0.000 |
| OLDCLAIM | 0.230 | 0.111 | 2.077 | 0.038 |
| CLM_FREQ | 0.368 | 0.060 | 6.125 | 0.000 |
| REVOKED.Yes | 0.296 | 0.036 | 8.150 | 0.000 |
| MVR_PTS | 0.407 | 0.061 | 6.695 | 0.000 |
| URBANICITY.Urban | 0.790 | 0.056 | 14.108 | 0.000 |
| `INCOME:OWNYes` | -0.606 | 0.115 | -5.248 | 0.000 |
| `INCOME:HOME_VAL` | 0.516 | 0.115 | 4.486 | 0.000 |
| `OLDCLAIM:CLM_FREQ` | -0.332 | 0.124 | -2.684 | 0.007 |
| `CLM_FREQ:MVR_PTS` | -0.217 | 0.080 | -2.709 | 0.007 |
| `OLDCLAIM:MVR_PTS` | -0.326 | 0.124 | -2.617 | 0.009 |
| `CAR_USECommercial:URBANICITYUrban` | 0.402 | 0.047 | 8.598 | 0.000 |
| `OLDCLAIM:CLM_FREQ:MVR_PTS` | 0.225 | 0.135 | 1.667 | 0.095 |

**Coefficient Discussion** It is difficult to directly map the predictors to probabilities, as all the values have been centered and scaled in order to perform the imputation. However, we can say that positive coefficients represent an **increased** risk of a claim over the baseline and negative coefficients represent a **decreased** risk under the baseline. When validating and then testing, the same imputations will be done and the results will be predicted on the normal probability scale.

Most of the coefficients are in the expected direction. For example, having one's license revoked, living in an urban area, or increased travel times are all *positive* risk factors while higher levels of wealth or being married are *negative* risk factors.

Some interesting observations. The predictors with the greatest effect on risk are `URBANICITY` and the increased factor when it is a commercial vehicle. The value of the coefficient for `SEX` implies that being female decreases the probability of a claim, as suspected, but it is only significant as a predictor at the 10% level. However, it is kept in the model as removing it increases the AIC. `BLUEBOOK`, `INCOME`, and Having a `HOME_VAL` greater than 0 (as indicated by the `OWN` variable being Yes) are all negative indicators, which supports the hypothesis that people with greater wealth have fewer accidents. Those values are tempered by some of the wealth interactions.

Every level of the interactions between `CLM_FREQ`, `OLDCLAIM`, and `MVR_PTS` are significant. What this may be telling us is that linear regression is ***not*** the proper model for this analysis and the algorithm is struggling to

add non-linearity via the interactions.

The interaction between `CAR_USE` and `URBANICITY` is also of interest. It indicates that commercial use vehicles don't really add to claim frequency in rural areas—the singleton variable is absent from the model—but they have an outsized influence on increased claim probabilities in urban areas!

Lastly, red cars do not contribute to increased claims frequency in this model.

**Variable Importance** In terms of variable importance, it can be estimated from the the last trace of the stepping procedure shown below. The further down the table the variable is, the more dramatic its inclusion or exclusion will be on the deviance and the AIC.

```
                                  Df Deviance    AIC
<none>                                 5131.6 5199.6
- `OLDCLAIM:CLM_FREQ:MVR_PTS`      1   5134.4 5200.4
+ EDUCATION.Masters                1   5130.6 5200.6
+ HOMEKIDS                         1   5130.7 5200.7
+ CAR_AGE                          1   5130.8 5200.8
- AGE                              1   5134.9 5200.9
+ JOB.Manager                      1   5131.0 5201.0
- SEX.F                            1   5135.0 5201.0
+ HOME_VAL                         1   5131.1 5201.1
+ `HOME_VAL:OWNYes`                1   5131.1 5201.1
+ EDUCATION.PhD                    1   5131.3 5201.3
+ YOJ                              1   5131.6 5201.6
+ OWN.Yes                          1   5131.6 5201.6
+ EDUCATION.HighSchool             1   5131.6 5201.6
+ RED_CAR.yes                      1   5131.6 5201.6
+ CAR_USE.Commercial               1   5131.6 5201.6
- OLDCLAIM                         1   5135.9 5201.9
- JOB.Lawyer                       1   5137.6 5203.6
- `OLDCLAIM:MVR_PTS`               1   5138.4 5204.4
- CAR_TYPE.PanelTruck              1   5138.7 5204.7
- `CLM_FREQ:MVR_PTS`               1   5138.9 5204.9
- `OLDCLAIM:CLM_FREQ`              1   5139.0 5205.0
- JOB.Student                      1   5140.2 5206.2
- BLUEBOOK                         1   5141.1 5207.1
- INCOME                           1   5143.0 5209.0
- JOB.HomeMaker                    1   5143.1 5209.1
- CAR_TYPE.Van                     1   5144.0 5210.0
- PARENT1.Yes                      1   5144.7 5210.7
- MSTATUS.Yes                      1   5145.2 5211.2
- EDUCATION.Bachelors              1   5148.1 5214.1
- `INCOME:HOME_VAL`                1   5148.6 5214.6
- CAR_TYPE.Pickup                  1   5149.8 5215.8
- JOB.Professional                 1   5152.2 5218.2
- JOB.Clerical                     1   5156.9 5222.9
- `INCOME:OWNYes`                  1   5156.9 5222.9
- KIDSDRIV                         1   5164.6 5230.6
- JOB.BlueCollar                   1   5166.6 5232.6
- CLM_FREQ                         1   5168.5 5234.5
- TIF                              1   5169.5 5235.5
- CAR_TYPE.SUV                     1   5169.8 5235.8
- MVR_PTS                          1   5176.4 5242.4
- CAR_TYPE.SportsCar               1   5176.9 5242.9
```

```
- TRAVTIME                              1   5179.5 5245.5
- REVOKED.Yes                           1   5196.8 5262.8
- `CAR_USECommercial:URBANICITYUrban`   1   5207.3 5273.3
- URBANICITY.Urban                      1   5388.1 5454.1
```

**Severity Model**

In the insurance world, severity is not necessarily fit via linear regression. Rather, it is more often fit through comparison of the maximum likelihood fits of various distributional families to the size of the claims, but the task here is to fit a linear regression model to estimate future claim size.

**Training** The observations in the training set which have claim amounts greater than 0 will be extracted and used to fit the linear regression. Similar to frequency, the model will be selected via a forward and backward stepwise AIC algorithm starting with all the predictors. Experimentation with interactions only had worse effects, so no interaction will be considered.

Table 6: Model 1 Severity Regression Output

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 5727.843 | 206.072 | 27.795 | 0.000 |
| MSTATUS.Yes | -402.470 | 252.974 | -1.591 | 0.112 |
| SEX.F | -321.509 | 206.815 | -1.555 | 0.120 |
| BLUEBOOK | 817.670 | 212.164 | 3.854 | 0.000 |
| CAR_AGE | -304.722 | 214.951 | -1.418 | 0.157 |
| OWN.Yes | 373.167 | 252.375 | 1.479 | 0.139 |

**Coefficient Disucssion** It is not surprising that this model has fewer predictors. Information which helps in identifying the *existence* of a claim often does not help in estimating the *magnitude* of the claim.

Similar to the frequency model, it is difficult to directly map the predictors to claim values, as all the values have been centered and scaled in order to perform the imputation. However, we can say that positive coefficients represent **increased** claim sizes over the baseline and negative coefficients represent **decreased** claim sizes under the baseline. When validating and then testing, the same imputations will be done and the results will be predicted on the normal dollar scale.

Unlike the frequency model, many of the resulting predictors are not significant at even the 10% level. What this may be telling us is that linear regression is ***not*** the proper model for this analysis and the algorithm is struggling to add non-linearity via the interactions. In practice, at the very least a GLM with a gamma link would be tried or, more sophisticated predictive models such as tree-based ones would be investigated. Most likely, MLE would be used to estimate the severity and a hierarchical contingent stochastic cash flow model would be built such that given a claim, a loss would be generated from a much more amenable distribution, such as a Pareto or a Burr.

**Variable Importance** In terms of variable importance, it can be estimated from the last trace of the stepping procedure shown below. The further down the list a variable preceded by a - is, the more drastic of an effect its removal would have on the model AIC.

```
                    Df   Deviance   AIC
<none>                  9.5799e+10 31328
- CAR_AGE            1 9.5927e+10 31328
- OWN.Yes            1 9.5938e+10 31329
+ JOB.Manager        1 9.5696e+10 31329
- SEX.F              1 9.5953e+10 31329
- MSTATUS.Yes        1 9.5960e+10 31329
```

```
+ MVR_PTS               1 9.5712e+10 31329
+ CAR_USE.Commercial    1 9.5729e+10 31329
+ REVOKED.Yes           1 9.5735e+10 31329
+ CAR_TYPE.Pickup       1 9.5748e+10 31330
+ JOB.Professional      1 9.5754e+10 31330
+ CAR_TYPE.Van          1 9.5757e+10 31330
+ HOMEKIDS              1 9.5763e+10 31330
+ CAR_TYPE.PanelTruck   1 9.5769e+10 31330
+ JOB.Student           1 9.5778e+10 31330
+ CLM_FREQ              1 9.5779e+10 31330
+ CAR_TYPE.SUV          1 9.5784e+10 31330
+ EDUCATION.Bachelors   1 9.5787e+10 31330
+ CAR_TYPE.SportsCar    1 9.5787e+10 31330
+ TRAVTIME              1 9.5788e+10 31330
+ PARENT1.Yes           1 9.5788e+10 31330
+ EDUCATION.HighSchool  1 9.5789e+10 31330
+ JOB.BlueCollar        1 9.5789e+10 31330
+ JOB.HomeMaker         1 9.5790e+10 31330
+ INCOME                1 9.5793e+10 31330
+ JOB.Lawyer            1 9.5793e+10 31330
+ OLDCLAIM              1 9.5794e+10 31330
+ TIF                   1 9.5795e+10 31330
+ KIDSDRIV              1 9.5796e+10 31330
+ YOJ                   1 9.5796e+10 31330
+ EDUCATION.PhD         1 9.5796e+10 31330
+ EDUCATION.Masters     1 9.5797e+10 31330
+ JOB.Clerical          1 9.5797e+10 31330
+ AGE                   1 9.5797e+10 31330
+ HOME_VAL              1 9.5797e+10 31330
+ URBANICITY.Urban      1 9.5798e+10 31330
+ RED_CAR.yes           1 9.5798e+10 31330
- BLUEBOOK              1 9.6748e+10 31341
```

## Model Set #2

### Frequency Model

The frequency model is built to determine whether a vehicle is likely to crash. In Model Set #2, to work with complete separation in the logistic regression model, a Bayesian analysis is fitted. The Bayesian statistical model returns samples of the parameters of interest (the "posterior" distribution) based on some "prior" distribution which is then updated by the data. Here, Cauchy distribution is accepted as a prior for parameters of the generalized linear model.

Table 7: Model 2 Frequency Regression Output

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -3.527 | 0.394 | -8.951 | 0.000 |
| KIDSDRIV | 0.183 | 0.146 | 1.253 | 0.210 |
| AGE | -0.003 | 0.005 | -0.624 | 0.532 |
| HOMEKIDS | -0.013 | 0.064 | -0.196 | 0.844 |
| YOJ | -0.007 | 0.010 | -0.728 | 0.467 |
| INCOME | 0.000 | 0.000 | -3.100 | 0.002 |
| PARENT1Yes | 0.264 | 0.144 | 1.841 | 0.066 |
| HOME_VAL | 0.000 | 0.000 | -0.949 | 0.342 |

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| MSTATUSYes | -0.444 | 0.103 | -4.326 | 0.000 |
| SEXF | -0.233 | 0.132 | -1.762 | 0.078 |
| EDUCATIONBachelors | -0.419 | 0.136 | -3.085 | 0.002 |
| EDUCATIONMasters | -0.295 | 0.211 | -1.397 | 0.162 |
| EDUCATIONPhD | -0.025 | 0.247 | -0.101 | 0.919 |
| EDUCATIONHighSchool | -0.057 | 0.112 | -0.507 | 0.612 |
| JOBClerical | 0.461 | 0.224 | 2.056 | 0.040 |
| JOBDoctor | -0.351 | 0.311 | -1.128 | 0.259 |
| JOBHomeMaker | 0.525 | 0.243 | 2.165 | 0.030 |
| JOBLawyer | 0.219 | 0.197 | 1.113 | 0.266 |
| JOBManager | -0.302 | 0.194 | -1.558 | 0.119 |
| JOBProfessional | 0.339 | 0.203 | 1.670 | 0.095 |
| JOBStudent | 0.296 | 0.254 | 1.166 | 0.243 |
| JOBBlueCollar | 0.418 | 0.211 | 1.978 | 0.048 |
| TRAVTIME | 0.015 | 0.002 | 6.905 | 0.000 |
| CAR_USECommercial | 0.799 | 0.109 | 7.352 | 0.000 |
| BLUEBOOK | 0.000 | 0.000 | -2.910 | 0.004 |
| TIF | -0.053 | 0.009 | -6.110 | 0.000 |
| CAR_TYPEPanelTruck | 0.533 | 0.190 | 2.807 | 0.005 |
| CAR_TYPEPickup | 0.503 | 0.118 | 4.252 | 0.000 |
| CAR_TYPESportsCar | 1.046 | 0.153 | 6.826 | 0.000 |
| CAR_TYPEVan | 0.535 | 0.150 | 3.566 | 0.000 |
| CAR_TYPESUV | 0.829 | 0.131 | 6.309 | 0.000 |
| RED_CARyes | 0.003 | 0.102 | 0.029 | 0.977 |
| OLDCLAIM | 0.000 | 0.000 | -2.839 | 0.005 |
| CLM_FREQ | 0.195 | 0.034 | 5.820 | 0.000 |
| REVOKEDYes | 0.853 | 0.110 | 7.773 | 0.000 |
| MVR_PTS | 0.104 | 0.016 | 6.384 | 0.000 |
| CAR_AGE | 0.009 | 0.013 | 0.688 | 0.492 |
| URBANICITYUrban | 2.354 | 0.133 | 17.672 | 0.000 |
| OWNYes | -0.158 | 0.180 | -0.881 | 0.378 |
| CAR_AGE.fMore than a Year | -0.159 | 0.125 | -1.265 | 0.206 |
| HOMEKIDS.fAt least 1 | 0.229 | 0.171 | 1.334 | 0.182 |
| KIDSDRIV.fAt least 1 | 0.293 | 0.237 | 1.238 | 0.216 |

**Coefficient Discussion** Contributing coefficients are those which lie within the 95% level of significance. A positive coefficient indicates that as the value of the predictor increases, the response variable also tends to increase. A negative coefficient suggests that as the predictor increases, the response variable tends to decrease.

Immediately, the model follows the suggested theories behind specific variables. For example, policyholders who are married, have a higher income and an old claim is less likely to be involved in an accident. Moreover, more educated, at least with a Bachelor's degree, policyholders drive more safely. The higher the bluebook value of the policyholder's vehicle, it decreases the probability of a claim. This makes sense naturally since more intelligent individuals would seek to protect relatively valuable assets they are responsible for. Last, the longer a policyholder has been a customer of the company, the less likely they are to be involved in an accident. Since insurance companies would typically impose larger rates or cancel the policies of careless drivers.

On the contrary, and unsurprising, policyholders who have previously had their driver's licenses revoked increases the probability of a claim. This parallels the result that the more motor vehicle record points a policyholder has gained, they are assumed to be unsafe drivers and are more likely to be involved in an accident. If a policyholder lives in an urban area, they are prone to an accident since traffic and pedestrian

congestion are more common. Another expected result, is that the longer a policyholder's commute to work is, there is an increased likelihood of a driver being involved in an accident. The last connection that can be inferred is that drivers of a larger type of vehicles are more prone to accidents than are drivers of other types of vehicles, with minivans being the least prone. The results seem to suggest that owners of sports cars and SUVs are riskier types of drivers. Interestingly, commercial usage of vehicle present critical reason for more frequent accidents to occur because of driver error or non-performance, and failure to maintain vehicle causing issues or malfunction.

### Severity Model

Similar to Model Set #1, the multivariate linear regression model used the data set's `TARGET_AMT` attribute as the response variable and the data was split to include only those records that have a claim. The elastic net is a regularized regression method that linearly combines the L1 and L2 regularization penalties. Predictors, specifically qualitative variables, were automatically transformed into dummy variables, which is important because `glmnet()` can only take numerical, quantitative inputs. Cross-validation is done on different combinations of $\lambda_1$ and $\lambda_2$ to find the best values. The hybrid elastic net regression is especially good at dealing with situations when there are correlations between parameters.

Table 8: Model 2 Severity Regression Output

| variables | coef |
|---|---|
| (Intercept) | 5728.867 |
| SEXF | 0 |
| CAR_USECommercial | 56.475 |
| BLUEBOOK | 424.059 |

**Coefficient Disucssion** Once again, contributing coefficients are those which lie within the 95% level of significance. In this case, the linear model reduced the number of predictors that can predict the cost given an accident. The model fits extremely poor and does a mediocre job at explaining the variability within the training data. If interpretations are still to be made, first, the intercept itself suggests that with no information on the policyholder, given a vehicle accident, the likely claim value to the individual is $5,728. The other predictors suggest that the higher the bluebook value of the policyholder's vehicle, the larger the insurance payout will be in the event of an accident. If they use the vehicle for commercial, they will also have a larger claim amount. While being a female resulted as a significant variable, its coefficient is near zero and will be ignored.

## Model Set #3

### Frequency Model

The frequency model is used to determine whether the vehicle is likely to crash. The base of this is a binary classifier, with the output being a probability representing the likelihood of crashing.

**Baseline Model** We will start with a simple logistic model to serve as a baseline. This includes all variables in the dataset.

Table 9: Model 3 - Base Logistic Regression Output

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -3.530 | 0.397 | -8.888 | 0.000 |
| KIDSDRIV | 0.356 | 0.072 | 4.925 | 0.000 |
| AGE | -0.005 | 0.005 | -0.947 | 0.343 |
| HOMEKIDS | 0.051 | 0.044 | 1.162 | 0.245 |

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| YOJ | -0.005 | 0.010 | -0.541 | 0.589 |
| INCOME | 0.000 | 0.000 | -2.929 | 0.003 |
| PARENT1Yes | 0.365 | 0.131 | 2.794 | 0.005 |
| HOME_VAL | 0.000 | 0.000 | -0.658 | 0.511 |
| MSTATUSYes | -0.403 | 0.103 | -3.922 | 0.000 |
| SEXF | -0.223 | 0.133 | -1.682 | 0.093 |
| EDUCATIONBachelors | -0.417 | 0.137 | -3.034 | 0.002 |
| EDUCATIONMasters | -0.227 | 0.212 | -1.071 | 0.284 |
| EDUCATIONPhD | 0.035 | 0.249 | 0.142 | 0.887 |
| EDUCATIONHighSchool | -0.046 | 0.112 | -0.411 | 0.681 |
| JOBClerical | 0.492 | 0.232 | 2.120 | 0.034 |
| JOBDoctor | -0.343 | 0.317 | -1.083 | 0.279 |
| JOBHomeMaker | 0.564 | 0.250 | 2.257 | 0.024 |
| JOBLawyer | 0.235 | 0.200 | 1.176 | 0.240 |
| JOBManager | -0.295 | 0.199 | -1.480 | 0.139 |
| JOBProfessional | 0.355 | 0.209 | 1.700 | 0.089 |
| JOBStudent | 0.309 | 0.265 | 1.165 | 0.244 |
| JOBBlueCollar | 0.432 | 0.218 | 1.976 | 0.048 |
| TRAVTIME | 0.015 | 0.002 | 6.845 | 0.000 |
| CAR_USECommercial | 0.803 | 0.109 | 7.345 | 0.000 |
| BLUEBOOK | 0.000 | 0.000 | -3.007 | 0.003 |
| TIF | -0.053 | 0.009 | -6.123 | 0.000 |
| CAR_TYPEPanelTruck | 0.550 | 0.191 | 2.875 | 0.004 |
| CAR_TYPEPickup | 0.511 | 0.119 | 4.301 | 0.000 |
| CAR_TYPESportsCar | 1.049 | 0.154 | 6.803 | 0.000 |
| CAR_TYPEVan | 0.546 | 0.151 | 3.622 | 0.000 |
| CAR_TYPESUV | 0.828 | 0.132 | 6.263 | 0.000 |
| RED_CARyes | 0.004 | 0.102 | 0.038 | 0.969 |
| OLDCLAIM | 0.000 | 0.000 | -2.855 | 0.004 |
| CLM_FREQ | 0.196 | 0.034 | 5.830 | 0.000 |
| REVOKEDYes | 0.849 | 0.110 | 7.733 | 0.000 |
| MVR_PTS | 0.105 | 0.016 | 6.445 | 0.000 |
| CAR_AGE | -0.005 | 0.009 | -0.508 | 0.611 |
| URBANICITYUrban | 2.372 | 0.134 | 17.671 | 0.000 |
| OWNYes | -0.202 | 0.187 | -1.081 | 0.279 |

We can immediately see that a few variables *exceed* the 0.05 p-value threshold for significance.

**Enhanced Frequency Model** Backwards stepwise regression is performed and the result is a model with the following variables *removed*: `AGE`, `HOMEKIDS`, `YOJ`, `RED_CAR`, and `CAR_AGE`.

Table 10: Model 3 - Final Logistic Regression Output

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -3.874 | 0.316 | -12.270 | 0.000 |
| KIDSDRIV | 0.388 | 0.065 | 5.947 | 0.000 |
| INCOME | 0.000 | 0.000 | -2.773 | 0.006 |
| PARENT1Yes | 0.482 | 0.113 | 4.283 | 0.000 |
| HOME_VAL | 0.000 | 0.000 | -3.147 | 0.002 |
| MSTATUSYes | -0.392 | 0.095 | -4.119 | 0.000 |
| SEXF | -0.205 | 0.118 | -1.738 | 0.082 |

| | Estimate | Std. Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| EDUCATIONBachelors | -0.441 | 0.129 | -3.416 | 0.001 |
| EDUCATIONMasters | -0.284 | 0.190 | -1.497 | 0.134 |
| EDUCATIONPhD | -0.024 | 0.232 | -0.105 | 0.917 |
| EDUCATIONHighSchool | -0.054 | 0.112 | -0.482 | 0.630 |
| JOBClerical | 0.512 | 0.232 | 2.207 | 0.027 |
| JOBDoctor | -0.365 | 0.317 | -1.152 | 0.249 |
| JOBHomeMaker | 0.585 | 0.244 | 2.394 | 0.017 |
| JOBLawyer | 0.228 | 0.200 | 1.141 | 0.254 |
| JOBManager | -0.300 | 0.199 | -1.505 | 0.132 |
| JOBProfessional | 0.358 | 0.209 | 1.712 | 0.087 |
| JOBStudent | 0.436 | 0.250 | 1.743 | 0.081 |
| JOBBlueCollar | 0.441 | 0.219 | 2.019 | 0.043 |
| TRAVTIME | 0.015 | 0.002 | 6.811 | 0.000 |
| CAR_USECommercial | 0.804 | 0.109 | 7.371 | 0.000 |
| BLUEBOOK | 0.000 | 0.000 | -3.270 | 0.001 |
| TIF | -0.053 | 0.009 | -6.118 | 0.000 |
| CAR_TYPEPanelTruck | 0.569 | 0.191 | 2.983 | 0.003 |
| CAR_TYPEPickup | 0.503 | 0.119 | 4.234 | 0.000 |
| CAR_TYPESportsCar | 1.031 | 0.153 | 6.757 | 0.000 |
| CAR_TYPEVan | 0.557 | 0.151 | 3.701 | 0.000 |
| CAR_TYPESUV | 0.815 | 0.131 | 6.211 | 0.000 |
| OLDCLAIM | 0.000 | 0.000 | -2.923 | 0.003 |
| CLM_FREQ | 0.195 | 0.034 | 5.815 | 0.000 |
| REVOKEDYes | 0.858 | 0.110 | 7.829 | 0.000 |
| MVR_PTS | 0.106 | 0.016 | 6.559 | 0.000 |
| URBANICITYUrban | 2.372 | 0.134 | 17.691 | 0.000 |

**Coefficient Discussion** Some interesting observations from the final results:

- Living in an urban area has the greatest impact on an individual's likelihood of getting into a crash. This makes sense - urban areas are more crowded than rural areas, so the odds of crashing are higher.
- If your car is used for commercial purposes you have a higher probability of getting into a crash.
- As expected, if you've had your license revoked within the past 7 years, you're more likely to get into a crash.
- Being a female makes you less likely to get into a crash.
- Having a sports car makes you more likely to get into a crash than having either an SUV, Van, Pickup, or Panel Truck.
- Being a manager or doctor make you less likely to get into a car crash. Conversely, you're more likely to crash if you're a student, blue collar worker, clerk, homemaker, professional, or lawyer.

### Severity Model

The severity model is used to determine how much the crash will cost the insurance company. Linear regression will be used to estimate the total value of the cost for cars that are known to have been in an accident and accrued damage costs.

**Base Model** We will start with a simple logistic regression model to serve as a baseline. This includes all variables in the dataset, but only for the records that have a `TARGET_AMT` > $0.

Table 11: Model 3 - Base Linear Regression Output

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 2885.164 | 2498.088 | 1.155 | 0.248 |
| KIDSDRIV | -123.500 | 398.297 | -0.310 | 0.757 |
| AGE | 3.153 | 26.674 | 0.118 | 0.906 |
| HOMEKIDS | 212.115 | 256.666 | 0.826 | 0.409 |
| YOJ | 20.596 | 61.273 | 0.336 | 0.737 |
| INCOME | 0.002 | 0.010 | 0.178 | 0.859 |
| PARENT1Yes | -35.612 | 733.876 | -0.049 | 0.961 |
| HOME_VAL | -0.001 | 0.005 | -0.277 | 0.782 |
| MSTATUSYes | -945.325 | 637.896 | -1.482 | 0.139 |
| SEXF | -1303.756 | 827.104 | -1.576 | 0.115 |
| EDUCATIONBachelors | -305.035 | 805.073 | -0.379 | 0.705 |
| EDUCATIONMasters | 438.284 | 1348.151 | 0.325 | 0.745 |
| EDUCATIONPhD | 820.411 | 1600.127 | 0.513 | 0.608 |
| EDUCATIONHighSchool | -567.674 | 643.692 | -0.882 | 0.378 |
| JOBClerical | 1470.537 | 1488.727 | 0.988 | 0.323 |
| JOBDoctor | -901.870 | 2209.619 | -0.408 | 0.683 |
| JOBHomeMaker | 1316.332 | 1574.971 | 0.836 | 0.403 |
| JOBLawyer | 1104.038 | 1284.285 | 0.860 | 0.390 |
| JOBManager | 98.354 | 1291.902 | 0.076 | 0.939 |
| JOBProfessional | 1956.964 | 1392.498 | 1.405 | 0.160 |
| JOBStudent | 1736.779 | 1672.995 | 1.038 | 0.299 |
| JOBBlueCollar | 1160.888 | 1411.553 | 0.822 | 0.411 |
| TRAVTIME | -5.392 | 13.708 | -0.393 | 0.694 |
| CAR_USECommercial | 1108.711 | 660.901 | 1.678 | 0.094 |
| BLUEBOOK | 0.119 | 0.039 | 3.065 | 0.002 |
| TIF | 4.192 | 53.102 | 0.079 | 0.937 |
| CAR_TYPEPanelTruck | -1371.386 | 1190.925 | -1.152 | 0.250 |
| CAR_TYPEPickup | -700.665 | 746.693 | -0.938 | 0.348 |
| CAR_TYPESportsCar | 890.509 | 951.589 | 0.936 | 0.350 |
| CAR_TYPEVan | -69.021 | 971.582 | -0.071 | 0.943 |
| CAR_TYPESUV | 810.156 | 841.857 | 0.962 | 0.336 |
| RED_CARyes | 100.879 | 616.157 | 0.164 | 0.870 |
| OLDCLAIM | 0.025 | 0.028 | 0.892 | 0.372 |
| CLM_FREQ | -26.739 | 197.739 | -0.135 | 0.892 |
| REVOKEDYes | -993.726 | 670.606 | -1.482 | 0.139 |
| MVR_PTS | 65.942 | 88.033 | 0.749 | 0.454 |
| CAR_AGE | -51.702 | 55.217 | -0.936 | 0.349 |
| URBANICITYUrban | 8.575 | 955.200 | 0.009 | 0.993 |
| OWNYes | 1075.297 | 1067.391 | 1.007 | 0.314 |

A look at the final output shows us that only 1 variable is deemed important (`BLUEBOOK`).

**Enhanced Model** In this case, if we were to use backwards stepwise regression, we would be left with a model that is determined by only 1 variable. However, intuitively we know that some of the variables are likely to effect the final payout. For this reason, we will choose to include: `BLUEBOOK`, `CAR_AGE`, and `CAR_TYPE`.

Table 12: Model 3 - Final Linear Regression Output

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 4920.371 | 698.601 | 7.043 | 0.000 |
| BLUEBOOK | 0.092 | 0.033 | 2.828 | 0.005 |
| CAR_AGE | -50.778 | 38.134 | -1.332 | 0.183 |
| CAR_TYPEPanelTruck | -66.406 | 996.646 | -0.067 | 0.947 |
| CAR_TYPEPickup | -321.613 | 681.440 | -0.472 | 0.637 |
| CAR_TYPESportsCar | -246.987 | 758.456 | -0.326 | 0.745 |
| CAR_TYPEVan | 739.408 | 878.674 | 0.842 | 0.400 |
| CAR_TYPESUV | -291.204 | 632.772 | -0.460 | 0.645 |

**Coefficient Disucssion** The intercept of the final model tells us that the base cost of a crash is a little under $5000. Some results:

- The positive coefficient for the `BLUEBOOK` variable means that the higher the bluebook value for the vehicle, the more the payout will be. This makes sense.
- The negative coefficient for the `CAR_AGE` variable means that the older the vehicle is, the less the payout is. Once again, this makes sense.
- Non-intuitively, the positive coefficient on the Van option for the `CAR_TYPE` variable means that van collisions have a higher payout. All other options for this variable (including a sports car) result in a lower payout.

# SELECT MODELS

## Model Selection Criteria

For that *classification* algorithm, we will look at accuracy, AUC, and F1, and select the model which performs best two out of three in those metrics. For the *regression* algorithm, we will look at RMSE and MAE (since the values must be positive) and select the model which performs best on both, Ties will be broken by $R^2$.

## Model 1

### Frequency Model

```
##           Reference
## Prediction NoClaim Claim
##    NoClaim    1653   353
##    Claim       147   295
```

Frequency Model 1 returns an accuracy of 0.7958, an F1 score of 0.8686, and an AUC of 0.8273.

### Severity Model

Severity Model 1 returns an RMSE of 6896.5, an MAE of 3558.9, and an $R^2$ of 0.0324.

## Model 2

### Frequency Model

```
##           Reference
## Prediction NoClaim Claim
##    NoClaim    1654   358
##    Claim       146   290
```

Frequency Model 2 returns an accuracy of 0.7941, an F1 score of 0.8678, and an AUC of 0.8286.

### Severity Model

Severity Model 2 returns an RMSE of 6948.6, an MAE of 3495.6, and an $R^2$ of 0.0264.

## Model 3

### Frequency Model

```
##            Reference
## Prediction NoClaim Claim
##    NoClaim    1663   359
##    Claim       137   289
```

Frequency Model 3 returns an accuracy of 0.7974, an F1 score of 0.8702, and an AUC of 0.8271.

### Severity Model

Severity Model 3 returns an RMSE of 6923.8, an MAE of 3557.1, and an $R^2$ of 0.0239.

## Model Selection

### Frequency Model

Table 13: Frequency Model Comparison

| Models | ACC | F1 | AUC |
|---|---|---|---|
| Model 1 | 0.7958 | 0.8686 | 0.8273 |
| Model 2 | 0.7941 | 0.8678 | 0.8286 |
| Model 3 | 0.7974 | 0.8702 | 0.8271 |

**Frequency Model 3**'s performance on the holdout set was the best in two of the three metrics—accuracy and F1—and will be used to predict the frequency of a claim on the evaluation set.

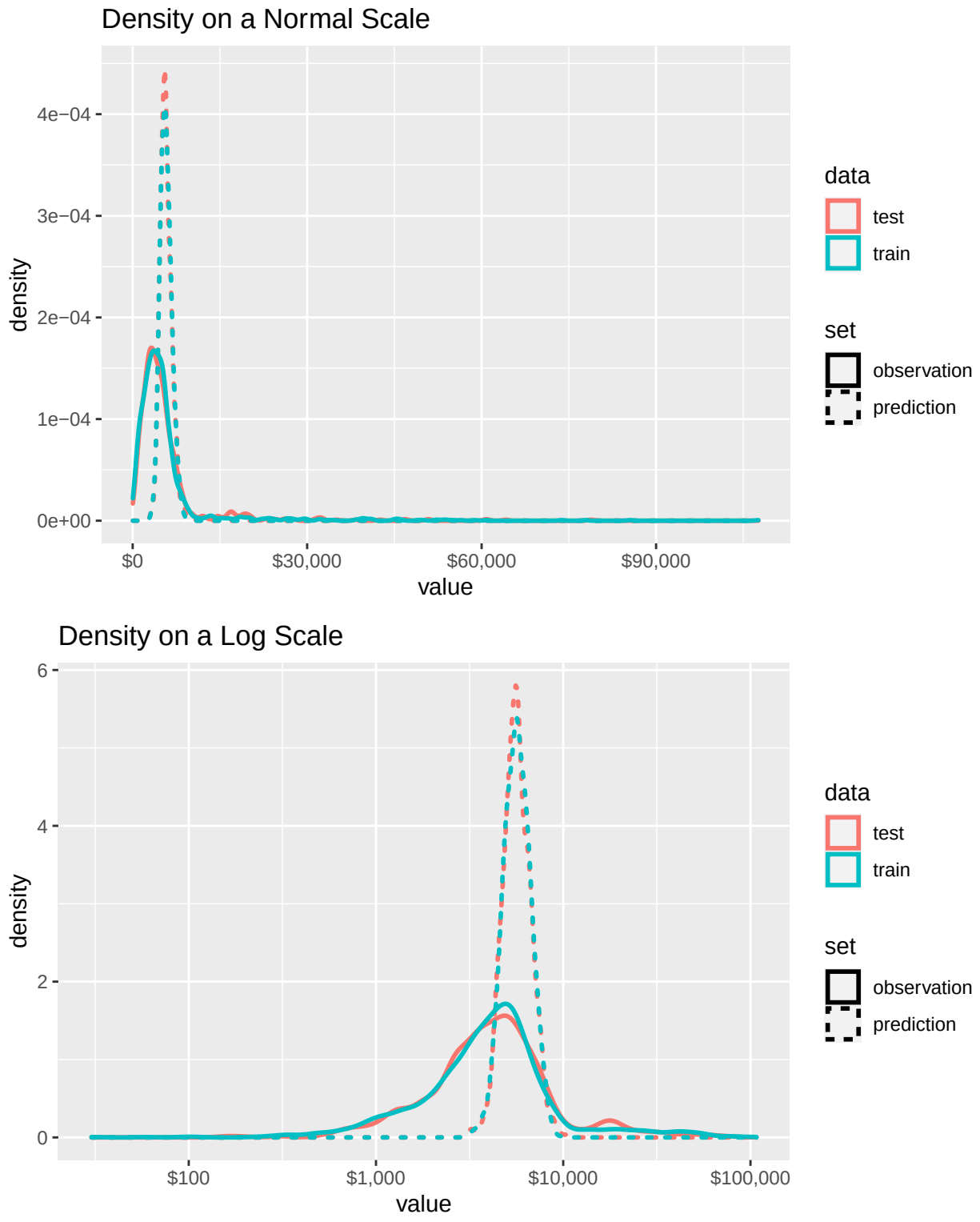
### Severity Model

Table 14: Severity Model Comparison

| Models | RMSE | MAE | R2 |
|---|---|---|---|
| Model 1 | 6896.535 | 3558.922 | 0.0324 |
| Model 2 | 6948.579 | 3495.614 | 0.0264 |
| Model 3 | 6923.837 | 3557.095 | 0.0239 |

**Severity Model 1**'s performance on the holdout set was the best in two of the three metrics—RMSE and $R^2$—and will be used to predict the severity for claims in the evaluation set for which Model 3 predicts a claim.

However, the performance of this model, or any of the three, is not expected to be good, as a linear regression with a normal assumption is most often a poor fit to severity data. Compare the distributions of both training and testing results for Model 1, ostensibly the best model, on both a real and a log-scale.




It is clear that the normal distribution assumption underlying the linear model is too restrictive. The variance is too low and the tails are too thin. The log-density plot implies a thicker tailed distribution such as a lognormal or a Pareto. Nevertheless, we will use Model 3 to predict the losses, understanding that it is the best option under the artificial constraints of the problem.[1]

[1] Apologies; one of us is an actuary and this is painful 8-) .

# PREDICTIONS

For our group, different model sets had the best performance for frequency and severity. These models processed the data differently. Therefore, we cannot simply use the output from Model 3's frequency predictions as the inputs to Model 1's severity.

What we will do is use Model 3 to determine *which* evaluation observations are predicted to have claims and then use those observations to predict severity. In other words, the evaluation data will be processed in accordance with Frequency Model 3 and the incidents of a claim will be recorded. For those evaluation observations which return a claim, their raw data will be processed according to Severity Model 1 to return predicted values.

As there are 2141 observations, we will not printing them in this document. Rather, we will show an estimated frequency rate. The actual observations for which claims are predicted may be obtained from the appropriate data object in the code, as found in the CODE APPENDIX.

## Evaluation Data Processing and Cleaning

In this section, the evaluation set will be processed and cleaned as the per the training set. The model-specific imputations, variable creations, and predictions will be done in their own sections.

## Frequency

Missing values will be imputed, with `INDEX` and the two prediction targets ignored, of course. Once done, the claim incidents will be predicted.

The model predicts 352 claims out of 2141 observations for a claim rate of 16.4%. The actual outcomes are stored in the `evalFreqFit` object in the code.

## Severity

The individual 352 observations for which Model 3 predicted a claim will be extracted from the state of the evaluation set prior to Model 3-specific adjustments, processed per Severity Model 1, and then the claim values will be predicted.

Once again, with so many observations, we will not simply print all 352. Rather, we will provide some summary statistics and graphs of the distribution of the claims. The actual claim values can be extracted from the appropriate data structure in the code, of course.

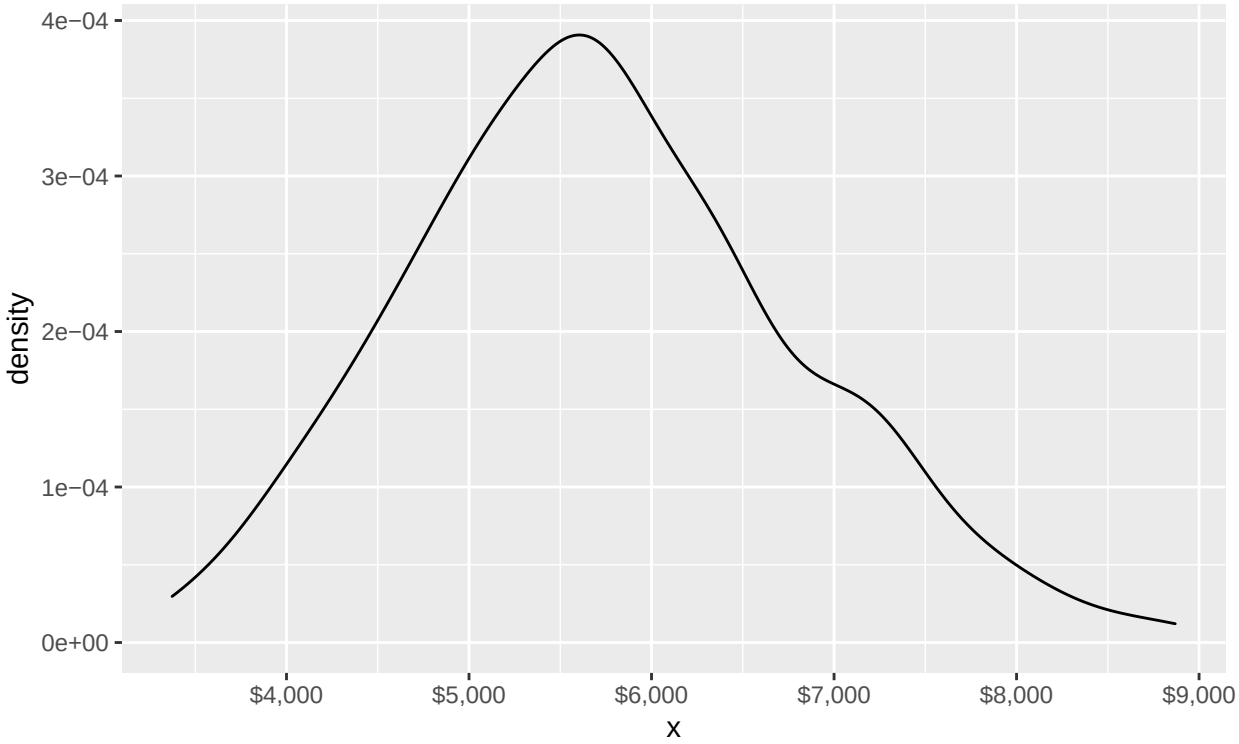



Table 15: Severity Statistics Comparison

| Set | Min | Median | Max | Mean | SD | CV |
|---|---|---|---|---|---|---|
| Training | 30.277 | 4130.000 | 107586.136 | 5728.867 | 8039.972 | 1.403 |
| Testing | 159.151 | 4061.500 | 78874.191 | 5640.198 | 7011.070 | 1.243 |
| PredictedEval | 3373.371 | 5659.734 | 8869.407 | 5731.738 | 1057.148 | 0.184 |

The actual outcomes are stored in the `evalSevFit` object in the code. The table and the graph show how the predictions are ***much*** more constrained than either the training or testing sets. The means are relatively close, which is to be expected when trying to shoehorn non-normal data into a Gaussian framework.

# CODE APPENDIX

```
# Load necessary libraries
library(arm)
library(dplyr)
library(ggplot2)
library(scales)
library(knitr)
library(caret)
library(RANN)
library(pROC)
library(mice)
library(corrplot)
library(data.table)

# Set master seed
```

```r
set.seed(71554)

# Set filepaths for data ingestion
urlRemote  = "https://raw.githubusercontent.com/"
pathGithub = "aadler/DT621_Fall2020_Group2/master/HW4/data/"
fileTrain = "insurance_training_data.csv"
fileTest = "insurance-evaluation-data.csv"

# Read training file
DT <- fread(paste0(urlRemote, pathGithub, fileTrain),
            colClasses = c(rep('integer', 2L), 'double', rep('integer', 4L),
                           'double', 'factor', 'double', rep('factor', 4L),
                           'integer', 'factor', 'double', 'integer',
                           rep('factor', 2L), 'double', 'integer', 'factor',
                           rep('integer', 2L), 'factor'))

# Convert character currencies to doubles
DT[, `:=`(INCOME = as.double(gsub('[$,]', '', INCOME)),
          HOME_VAL = as.double(gsub('[$,]', '', HOME_VAL)),
          BLUEBOOK = as.double(gsub('[$,]', '', BLUEBOOK)),
          OLDCLAIM = as.double(gsub('[$,]', '', OLDCLAIM)))]

# Number of training observations
ntrobs <- dim(DT)[[1]]

# Get the names of the variables and which ones are numeric
nmtrn <- names(DT)
nmtrnNUM <- names(DT[, .SD, .SDcols = sapply(DT, is.numeric)])
nmtrnFAC <- setdiff(nmtrn, nmtrnNUM)
# Drop INDEX and target variables from name list
nmtrnNUM <- nmtrnNUM[-(1:3)]
```

```r
# Check for NAs in all variables
missingV <- DT[, lapply(.SD, function(x) sum(is.na(x)))]
# Only keep those whose count is > 0
missingV <- missingV[, .SD, .SDcols = sapply(missingV, function (x) x > 0)]
kable(missingV, caption = "Variables with Count of Missing Values > 0")
```

```r
# Check for 0 value for the above variables
zeroV <- DT[, lapply(.SD, function(x) sum(x == 0, na.rm = TRUE)),
            .SDcols = c('AGE', 'YOJ', 'INCOME', 'HOME_VAL', 'TRAVTIME',
                        'BLUEBOOK', 'TIF', 'CAR_AGE')]
# Only keep those whose count of zero values is > 0
zeroV <- zeroV[, .SD, .SDcols = sapply(zeroV, function (x) x > 0)]
kable(zeroV, caption = "Variables with Count of Missing Values > 0")
```

```r
# Isolate numeric only predictors
numDT <- DT[, .SD, .SDcols = nmtrnNUM]

# Melt them from wide to long format
numDTM <- melt(numDT, variable.name = 'metric', value.name = 'value')

# Calculate summary statistics
statsN <- numDTM[, .(Mean = mean(value, na.rm = TRUE),
```

```
                    SD = sd(value, na.rm = TRUE),
                    Min = min(value, na.rm = TRUE),
                    Q1 = quantile(value, prob = 0.25, na.rm = TRUE),
                    Median = median(value, na.rm = TRUE),
                    Q3 = quantile(value, prob = 0.75, na.rm = TRUE),
                    Max = max(value, na.rm = TRUE),
                    IQR = IQR(value, na.rm = TRUE)), keyby = 'metric']

# Print the table
kable(statsN, digits = 3L, align = 'r',
      caption = "Summary Statistitics for Numeric Variables")
```

```
# Freedman-Diaconis rule for bin widths
FDbin <- function(x) {
  result <- 2 * IQR(x, na.rm = TRUE) / (length(x) ^ (1 / 3))
  return(ifelse(result == 0, 0.5, result))
}
ggplot(numDTM[!is.na(metric)], aes(x = value)) +
  geom_histogram(binwidth = FDbin, fill = 'lightblue3') +
  facet_wrap( ~ metric, scales = 'free')

ggplot(numDTM, aes(x = value)) +
  geom_histogram(binwidth = FDbin, fill = 'indianred4') +
  facet_wrap( ~ metric, scales = 'free') + scale_x_log10()
```

```
# Isolate factor only predictors
facDT <- DT[, .SD, .SDcols = nmtrnFAC]

# Melt them from wide to long format (need to explicitly call measure.vars since
# these are all factors)
facDTM <- melt(facDT, measure.vars = nmtrnFAC, variable.name = 'metric',
               value.name = 'value')
ggplot(facDTM, aes(x = value)) + geom_bar(fill = 'darkolivegreen') +
  facet_wrap( ~ metric, nrow = 5L, scales = 'free') + coord_flip()
```

```
corrplot::corrplot(cor(DT[, ..nmtrnNUM], use = 'complete.obs'),
         method = 'ellipse', type = 'lower', order = 'hclust',
         hclust.method = 'ward.D2')
```

```
# Clean up data values. Using setattr for fast data.table setting. Equivalent
# to levels(DT$X) <- c(a, b) but faster since changes by reference.
# Remove spaces where possible too
setattr(DT$MSTATUS, 'levels', c('Yes', 'No'))
setattr(DT$SEX, 'levels', c('M', 'F'))
setattr(DT$EDUCATION, 'levels', c('<HighSchool', 'Bachelors', 'Masters', 'PhD',
                                  'HighSchool'))
setattr(DT$JOB, 'levels', c('Unknown', 'Clerical', 'Doctor', 'HomeMaker',
                            'Lawyer', 'Manager', 'Professional', 'Student',
                            'BlueCollar'))
setattr(DT$CAR_TYPE, 'levels', c('Minivan', 'PanelTruck', 'Pickup',
                                 'SportsCar', 'Van', 'SUV'))

setattr(DT$URBANICITY, 'levels', c('Urban', 'Rural'))

# Relevel some of the variables to have the default make more sense
```

```r
# Set default marital status to unmarried
set(DT, NULL, 'MSTATUS', relevel(DT$MSTATUS, 'No'))
# Set default car usage to private
set(DT, NULL, 'CAR_USE', relevel(DT$CAR_USE, 'Private'))
# Set default are to rural, making urban a positive? risk factor
set(DT, NULL, 'URBANICITY', relevel(DT$URBANICITY, 'Rural'))
```

```r
# Create training and testing split
set.seed(642)
trnIDX <- createDataPartition(DT$TARGET_FLAG, p = 0.7)
trnX <- DT[trnIDX$Resample1, ]
tstX <- DT[!trnIDX$Resample1, ]
```

```r
# Model Set 1: Add OWN variable
trnX[, OWN := factor(ifelse(HOME_VAL > 0, 1, 0),
                     levels = c(0, 1), labels = c('No', 'Yes'))]
tstX[, OWN := factor(ifelse(HOME_VAL > 0, 1, 0),
                     levels = c(0, 1), labels = c('No', 'Yes'))]
```

```r
# Model 2: Clean Training set
m2.train.X <- trnX
m2.train.X$HOME_VAL[is.na(m2.train.X$HOME_VAL)] <- 0
m2.train.X$OWN[is.na(m2.train.X$OWN)] <- "No"
m2.train.X$CAR_AGE = abs(m2.train.X$CAR_AGE)
m2.train.r <- factor(m2.train.X$TARGET_FLAG, levels = c(0, 1),
                     labels = c("NoClaim", "Claim"))
m2.train.X.impute <- mice(m2.train.X, method = 'pmm', print = FALSE)
m2.train.X <- complete(m2.train.X.impute)
m2.train.X$CAR_AGE.f <- factor(ifelse(m2.train.X$CAR_AGE > 1, 1, 0),
                               levels = c(0, 1),
                               labels = c('1 Year or less', 'More than a Year'))
m2.train.X$HOMEKIDS.f <- factor(ifelse(m2.train.X$HOMEKIDS >= 1, 1, 0),
                               levels = c(0, 1),
                               labels = c('None', 'At least 1'))
m2.train.X$KIDSDRIV.f <- factor(ifelse(m2.train.X$KIDSDRIV >= 1, 1, 0),
                               levels = c(0, 1),
                               labels = c('None', 'At least 1'))
m2.train.p <- m2.train.X[, -c(1:3)]

# Clean Test Set
m2.test.X <- tstX
m2.test.X$HOME_VAL[is.na(m2.test.X$HOME_VAL)] <- 0
m2.test.X$OWN[is.na(m2.test.X$OWN)] <- "No"
m2.test.X$CAR_AGE = abs(m2.test.X$CAR_AGE)
m2.test.r <- factor(m2.test.X$TARGET_FLAG, levels = c(0, 1),
                     labels = c("NoClaim", "Claim"))
m2.test.X.impute <- mice(m2.test.X, method = 'pmm', print = FALSE)
m2.test.X <- complete(m2.test.X.impute)
m2.test.X$CAR_AGE.f <- factor(ifelse(m2.test.X$CAR_AGE > 1, 1, 0),
                               levels = c(0, 1),
                               labels = c('1 Year or less', 'More than a Year'))
m2.test.X$HOMEKIDS.f <- factor(ifelse(m2.test.X$HOMEKIDS >= 1, 1, 0),
                               levels = c(0, 1),
                               labels = c('None', 'At least 1'))
```

```
m2.test.X$KIDSDRIV.f <- factor(ifelse(m2.test.X$KIDSDRIV >= 1, 1, 0),
                               levels = c(0, 1),
                               labels = c('None', 'At least 1'))
m2.test.p <- m2.test.X[, -c(1:3)]
```

```
# Model 3 - data transformation
# MICE imputation on train and test set
m3train <- complete(mice(data = trnX %>% select(-INDEX),
                         method = "pmm", print = FALSE), 3)
m3train$TARGET_FLAG <- factor(m3train$TARGET_FLAG, levels = c(0, 1),
                      labels = c("NoClaim", "Claim"))

m3test <- complete(mice(data = tstX %>% select(-INDEX),
                        method = "pmm", print = FALSE), 3)
m3test$TARGET_FLAG <- factor(m3test$TARGET_FLAG, levels = c(0, 1),
                      labels = c("NoClaim", "Claim"))
```

```
# Model Set 1: Cast target as factor for caret purposes
trnFreq <- factor(trnX$TARGET_FLAG, levels = c(0, 1),
                  labels = c("NoClaim", "Claim"))
# Set up dummy variables
trnDumFreq <- dummyVars(TARGET_FLAG ~ . +
                          OWN * HOME_VAL * INCOME +
                          CLM_FREQ * OLDCLAIM * MVR_PTS +
                          CAR_USE * URBANICITY -
                         INDEX - TARGET_AMT,
                        fullRank = TRUE, data = trnX)
# Create matrix of dummy values
trnXFD <- predict(trnDumFreq, newdata = trnX)
```

```
# Model Set 1: preProcess with near-zero value and kNN imputation
trnXFDpp <- preProcess(trnXFD, method = c('nzv', 'knnImpute'),
                       k = 3, knnSummary = median)
# Create processed predictors
trnXFDI <- predict(trnXFDpp, newdata = trnXFD)
trC <- trainControl(method = 'none',            # Using StepAIC
                    classProbs = TRUE,          # Classification
                    summaryFunction = prSummary # Precision/Recall
)
set.seed(487)
# Model Set 1: Train Model
m1FreqFit <- train(x = trnXFDI, y = trnFreq, trControl = trC,
                   method = 'glmStepAIC', family = binomial(link = 'logit'),
                   direction = 'both', trace = 0)
# Print results
kable(summary(m1FreqFit$finalModel)$coefficients, digits = 3L,
      caption = "Model 1 Frequency Regression Output")
```

```
# Model Set 1: Extract those observations with values > 0
trnXS <- trnX[TARGET_AMT > 0,
              ][, `:=`(INDEX = NULL, TARGET_FLAG = NULL)]
trnYS <- trnXS[, TARGET_AMT]

# Set up dummy variables
trnXDum <- dummyVars(TARGET_AMT ~ ., fullRank = TRUE, data = trnXS)
```

```r
# Create matrix of dummy values
trnXSD <- predict(trnXDum, newdata = trnXS)

# preProcess with near-zero value and kNN imputation using median
trnXDumPP <- preProcess(trnXSD, method = c('nzv', 'knnImpute'), k = 3,
                        knnSummary = median)

# Create processed predictors
trnXSDImp <- predict(trnXDumPP, newdata = trnXSD)
```

```r
# Model Set 2: Train Frequency Model
set.seed(525)
m2FreqFit <- train(x = m2.train.p, y = m2.train.r,
                   trControl = trainControl(method = 'repeatedcv',
                                            classProbs = TRUE,
                                            number = 10),
                   method = 'bayesglm', family = 'binomial')
# Results
kable(summary(m2FreqFit$finalModel)$coefficients, digits = 3L,
      caption = "Model 2 Frequency Regression Output")
```

```r
# Model Set 2: Train Severity Model
temp <- split(m2.train.X, m2.train.X$TARGET_AMT > 0)[[2]]
m2.train.r <- temp$TARGET_AMT
m2.train.p <- model.matrix(temp$TARGET_AMT~., temp[, -c(1:3)])[,-1]
set.seed(525)
m2SevFit <- train(x = m2.train.p, y = m2.train.r,
                   trControl = trainControl(method = 'repeatedcv',
                                            number = 10),
                   method = 'glmnet',
                   tuneLength = 10,
                   preProcess = c("center", "scale"))
# Results
coeffs = coef(m2SevFit$finalModel, m2SevFit$bestTune$lambda)
kable(data.frame(cbind(variables = coeffs@Dimnames[[1]][coeffs@i+1],
                       coef = round(coeffs@x,3))),
      caption = "Model 2 Severity Regression Output")
```

```r
# Base log model 3
m3BaseLog <- glm(TARGET_FLAG ~ ., family = binomial,
                 data = m3train %>% select(-TARGET_AMT))
kable(summary(m3BaseLog)$coefficients, digits = 3L,
      caption = 'Model 3 - Base Logistic Regression Output')
```

```r
# Final log model 3
train_control <- trainControl(method = "cv", number = 10)
m3FreqFit <- train(TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 + HOME_VAL +
                     MSTATUS + SEX + EDUCATION + JOB + TRAVTIME + CAR_USE +
                     BLUEBOOK + TIF + CAR_TYPE + OLDCLAIM + CLM_FREQ + REVOKED +
                     MVR_PTS + URBANICITY,
              data = m3train %>% select(-TARGET_AMT),
              trControl = train_control,
              method = "glm",
              family=binomial())
```

```
kable(summary(m3FreqFit)$coefficients, digits = 3L,
      caption = 'Model 3 - Final Logistic Regression Output')
```

```
m3BaseLin <- lm(TARGET_AMT ~ .,
                data = m3train %>%
                  filter(TARGET_AMT > 0) %>%
                  select(-TARGET_FLAG))
knitr::kable(summary(m3BaseLin)$coefficients, digits = 3L,
             caption = 'Model 3 - Base Linear Regression Output')
```

```
m3SevFit <- lm(TARGET_AMT ~ BLUEBOOK + CAR_AGE + CAR_TYPE,
               data = m3train %>%
                 filter(TARGET_AMT > 0) %>%
                 select(-TARGET_FLAG))

knitr::kable(summary(m3SevFit)$coefficients, digits = 3L,
             caption = 'Model 3 - Final Linear Regression Output')
```

```
# Create comparison tables
freqCompTable <- data.frame(Models = c('Model 1', 'Model 2', 'Model 3'),
                            ACC = double(3),
                            F1 = double(3),
                            AUC = double(3))
sevCompTable <- data.frame(Models = c('Model 1', 'Model 2', 'Model 3'),
                           RMSE = double(3),
                           MAE = double(3),
                           R2 = double(3))
```

```
# Model 1 Frequency:: process validation data similar to training data
tstFreq <- factor(tstX$TARGET_FLAG, levels = c(0, 1),
                  labels = c("NoClaim", "Claim"))
tstDumFreq <- dummyVars(TARGET_FLAG ~ . +
                          OWN * HOME_VAL * INCOME +
                          CLM_FREQ * OLDCLAIM * MVR_PTS +
                          CAR_USE * URBANICITY -
                         INDEX - TARGET_AMT,
                        fullRank = TRUE, data = tstX)
tstXFD <- predict(tstDumFreq, newdata = tstX)
tstXFDpp <- preProcess(tstXFD, method = c('nzv', 'knnImpute'),
                       k = 3, knnSummary = median)
tstXFDI <- predict(tstXFDpp, newdata = tstXFD)

# Model 1 Frequency: predict on validation data
m1Fpred <- predict(m1FreqFit, newdata = tstXFDI)
m1FpredProb <- predict(m1FreqFit, newdata = tstXFDI, type = 'prob')

# Model 1 Frequency: Calculate accuracy statistics
m1FCM <- confusionMatrix(m1Fpred, tstFreq, mode = 'everything')
freqCompTable[1, 2] <- m1FACC <- sum(diag(m1FCM$table)) / sum(m1FCM$table)
freqCompTable[1, 3] <- m1FF1 <- m1FCM$byClass[7]
freqCompTable[1, 4] <- m1FAUC <- auc(response = tstFreq,
                                     predictor = m1FpredProb[, 1])
m1FCM$table
```

```
# Model 1 Severity: process validation data similar to training data
tstXS <- tstX[TARGET_AMT > 0
              ][, `:=`(INDEX = NULL, TARGET_FLAG = NULL)]
tstYS <- tstXS[, TARGET_AMT]
tstXDum <- dummyVars(TARGET_AMT ~ ., data = tstXS, fullRank = TRUE)
tstXSD <- predict(tstXDum, newdata = tstXS)
tstXDumPP <- preProcess(tstXSD, method = c('nzv', 'knnImpute'), k = 3,
                        knnSummary = median)
tstXSDImp <- predict(tstXDumPP, newdata = tstXSD)

# Model 1 Severity: predict on validation data
m1Spred <- predict(m1SevFit, newdata = tstXSDImp)
# Model 1 Severity: Calculate accuracy statistics
m1Ssum <- postResample(m1Spred, tstYS)
sevCompTable[1, 2] <- m1SRMSE <- m1Ssum[1]
sevCompTable[1, 3] <- m1SMAE <- m1Ssum[3]
sevCompTable[1, 4] <- m1SR2 <- m1Ssum[2]
```

```
# Model 2 Frequency
# Prediction
m2Fpred <- predict(m2FreqFit, newdata = m2.test.X)
m2FpredProb <- predict(m2FreqFit, newdata = m2.test.X, type = 'prob')

# Accuracy Statistics
m2FCM <- confusionMatrix(m2Fpred, m2.test.r, mode = 'everything')
freqCompTable[2, 2] <- m2FACC <- sum(diag(m2FCM$table)) / sum(m2FCM$table)
freqCompTable[2, 3] <- m2FF1 <- m2FCM$byClass[7]
freqCompTable[2, 4] <- m2FAUC <- auc(response = m2.test.r,
                                     predictor = m2FpredProb[, 1])
m2FCM$table
```

```
# Model 2 Severity
temp <- split(m2.test.X, m2.test.X$TARGET_AMT > 0)[[2]]
m2.test.r <- temp$TARGET_AMT
m2.test.p <- model.matrix(temp$TARGET_AMT~., temp[, -c(1:3)])[,-1]

# Prediction
m2Spred <- predict(m2SevFit, newdata = m2.test.p)

# Accuracy Statistics
m2Ssum <- postResample(m2Spred, tstYS)
sevCompTable[2, 2] <- m2SRMSE <- RMSE(m2Spred, m2.test.r)
sevCompTable[2, 3] <- m2SMAE <- MAE(m2Spred, m2.test.r)
sevCompTable[2, 4] <- m2SR2 <- R2(m2Spred, m2.test.r)
```

```
# Model 3 Frequency: predict on validation data
m3Fpred <- predict(m3FreqFit, m3test %>% select(-TARGET_AMT))
m3FpredProb <- predict(m3FreqFit, m3test %>% select(-TARGET_AMT), type = 'prob')

# Model 3 Frequency: Calculate accuracy statistics
m3FCM <- confusionMatrix(m3Fpred, m3test$TARGET_FLAG, mode = 'everything')
freqCompTable[3, 2] <- m3FACC <- sum(diag(m3FCM$table)) / sum(m3FCM$table)
freqCompTable[3, 3] <- m3FF1 <- m3FCM$byClass[7]
freqCompTable[3, 4] <- m3FAUC <- auc(response = m3test$TARGET_FLAG,
```

```
                                    predictor = m3FpredProb[, 1])
m3FCM$table
```

```
# Model 3 Severity: subset data
m3testFinal <- m3test %>%
  select(-TARGET_FLAG) %>%
  filter(TARGET_AMT > 0)

# Model 3 Severity: predict on validation data
m3Spred <- predict(m3SevFit, newdata = m3testFinal)

# Model 3 Severity: Calculate accuracy statistics
m3Ssum <- postResample(m3Spred, tstYS)
sevCompTable[3, 2] <- m3SRMSE <- m3Ssum[1]
sevCompTable[3, 3] <- m3SMAE <- m3Ssum[3]
sevCompTable[3, 4] <- m3SR2 <- m3Ssum[2]
```

```
kable(freqCompTable, digits = 4L, caption = "Frequency Model Comparison")
```

```
kable(sevCompTable, digits = 4L, caption = "Severity Model Comparison")
```

```
m1trnPred <- predict(m1SevFit, trnXSDImp)
densDT <- data.table(trainObs = trnYS,
                     trainPred = m1trnPred,
                     testObs = tstYS,
                     testPred = m1Spred)
densDT <- melt(densDT, variable.factor = FALSE)
densDT[, `:=`(data = ifelse(substr(variable, 1, 5) == 'train', 'train', 'test'),
              len =  lapply(variable, nchar))
              ][, set := ifelse(substr(variable, len, len) == 's',
                                'observation', 'prediction')
                ][, `:=`(variable = NULL, len = NULL)]
ggplot(densDT, aes(x = value, col = data,  linetype = set)) +
  geom_density(size = 1) + scale_x_continuous(labels = dollar) +
  ggtitle("Density on a Normal Scale")
ggplot(densDT, aes(x = value, col = data, linetype = set)) +
  geom_density(size = 1) + scale_x_log10(labels = dollar) +
  ggtitle("Density on a Log Scale")
```

```
# Read and process the test/evaluation file
# Can re-use the name DT for convenience now
DT <- fread(paste0(urlRemote, pathGithub, fileTest),
              colClasses = c(rep('integer', 2L), 'double', rep('integer', 4L),
                             'double', 'factor', 'double', rep('factor', 4L),
                             'integer', 'factor', 'double', 'integer',
                             rep('factor', 2L), 'double', 'integer', 'factor',
                             rep('integer', 2L), 'factor'))
neval <- dim(DT)[[1]]
DT[, `:=`(INCOME = as.double(gsub('[$,]', '', INCOME)),
          HOME_VAL = as.double(gsub('[$,]', '', HOME_VAL)),
          BLUEBOOK = as.double(gsub('[$,]', '', BLUEBOOK)),
          OLDCLAIM = as.double(gsub('[$,]', '', OLDCLAIM)))]
setattr(DT$MSTATUS, 'levels', c('Yes', 'No'))
setattr(DT$SEX, 'levels', c('M', 'F'))
setattr(DT$EDUCATION, 'levels', c('<HighSchool', 'Bachelors', 'Masters', 'PhD',
```

```
                                    'HighSchool'))
setattr(DT$JOB, 'levels', c('Unknown', 'Clerical', 'Doctor', 'HomeMaker',
                            'Lawyer', 'Manager', 'Professional', 'Student',
                            'BlueCollar'))
setattr(DT$CAR_TYPE, 'levels', c('Minivan', 'PanelTruck', 'Pickup',
                                 'SportsCar', 'Van', 'SUV'))

setattr(DT$URBANICITY, 'levels', c('Urban', 'Rural'))

# Relevel some of the variables to have the default make more sense
# Set default marital status to unmarried
set(DT, NULL, 'MSTATUS', relevel(DT$MSTATUS, 'No'))
# Set default car usage to private
set(DT, NULL, 'CAR_USE', relevel(DT$CAR_USE, 'Private'))
# Set default are to rural, making urban a positive? risk factor
set(DT, NULL, 'URBANICITY', relevel(DT$URBANICITY, 'Rural'))
```

```
# Impute the missing values but not INDEX or the values we will be predicting!
set.seed(94)
m3eval <- complete(mice(data = DT %>%
                          select(-c(INDEX, TARGET_FLAG, TARGET_AMT)),
                        method = "pmm", print = FALSE), 3)
evalFreqFit <- predict(m3FreqFit, m3eval)
numPred <- sum(evalFreqFit == "Claim")
claimRate <- numPred / neval
claimIDX <- which(evalFreqFit == "Claim")
```

```
# Extract the "claim" observations from the data
evalObs <- DT[claimIDX, ]

# At this point the unneeded columns of INDEX and the 2 targets will be removed
evalObs[, `:=`(INDEX = NULL, TARGET_FLAG = NULL, TARGET_AMT = NULL)]

# Create the OWN variable
evalObs[, OWN := factor(ifelse(HOME_VAL > 0, 1, 0), levels = c(0, 1),
                        labels = c('No', 'Yes'))]

# Set up dummy variables
evalDumSev <- dummyVars( ~ ., fullRank = TRUE, data = evalObs)
# Create matrix of dummy values
evalDumSevMatrix <- predict(evalDumSev, newdata = evalObs)
# preProcess with near-zero value and kNN imputation
evalDumSevPP <- preProcess(evalDumSevMatrix, method = c('nzv', 'knnImpute'),
                           k = 3, knnSummary = median)
# Create matrix of processed and imputed predictors
evalDumSevPPI <- predict(evalDumSevPP, newdata = evalDumSevMatrix)
evalSevFit <- predict(m1SevFit, evalDumSevPPI)

# Kernel-smoothed density
ggplot(data.frame(x = evalSevFit), aes(x = x)) + geom_density() +
  scale_x_continuous(labels = dollar)

# Summary statistics comparison
tabularComp <- data.table(Set = c("Training", "Testing", "PredictedEval"),
```

```
                          Min = c(min(trnYS), min(tstYS), min(evalSevFit)),
                          Median = c(median(trnYS), median(tstYS),
                                     median(evalSevFit)),
                          Max = c(max(trnYS), max(tstYS), max(evalSevFit)),
                          Mean = c(mean(trnYS), mean(tstYS), mean(evalSevFit)),
                          SD = c(sd(trnYS), sd(tstYS), sd(evalSevFit)))
tabularComp[, CV := SD / Mean, by = Set]
knitr::kable(tabularComp, digits = 3L,
             caption = "Severity Statistics Comparison")
```