|  |  |
| --- | --- |
| Auto Insurance Claims Analysis and Prediction | Abstract  EDA, Data preparation and model fitting using am auto claims dataset to predict total claim amounts.  Rebecca Leu  Northeastern University - ALY6040 Data Mining |

**Introduction to the dataset**

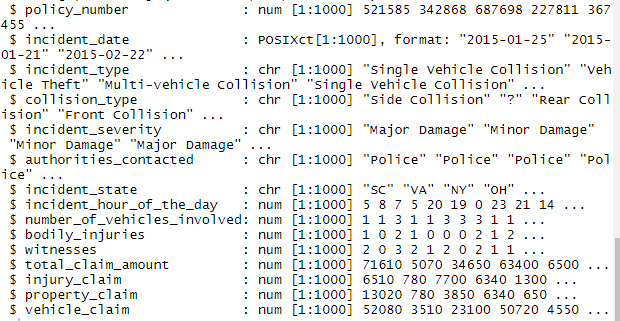
The dataset we are analyzing, and modeling is from Kaggle. The dataset originally consisted of 40 variables and 1000 observation. Each observation is a claim or incident with each column pertaining to the policy holder themselves or the claim data. Since this project is focusing on predicting total claims values based off other pieces of claims data, I initially removed a lot of the policyholder information as well as a few columns that didn’t add much value to the analysis. What was left was the 1000 observations and 12 variables.

**The business question**

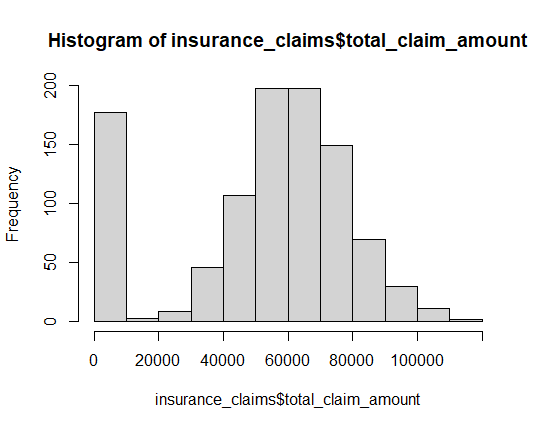
Using information gathered in the first notice of the accident to the insurance companies, can we predict what the total claim value will be? This model would be helpful for the reserving and accounting departments of the insurance company to better prepare for claims payout. Often times claims take months or even years to fully settle. If we are able to predict claim values based off of information gathered within the first few weeks after the claim was filed, our company would be better able to brace for the payments that will eventually need to be made.

**Exploratory data analysis**

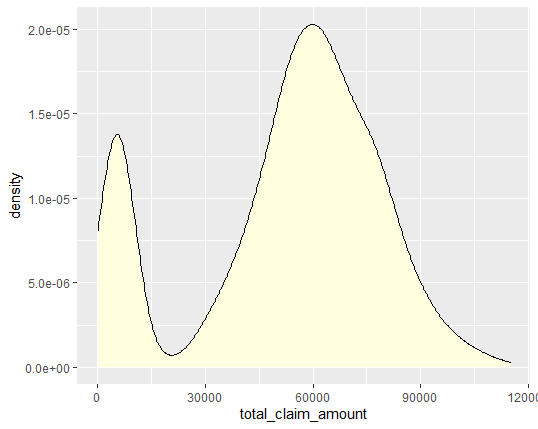
The initial exploration of the data is done to get a feel for what variables are available to us and what initial observations or questions we can answer about the dataset. The first thing I like to look at is the structure of each of the observations:



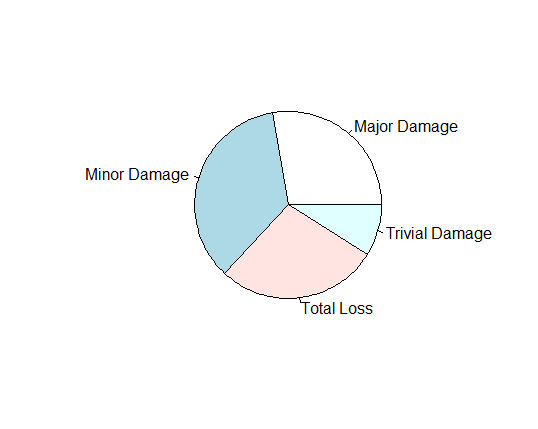
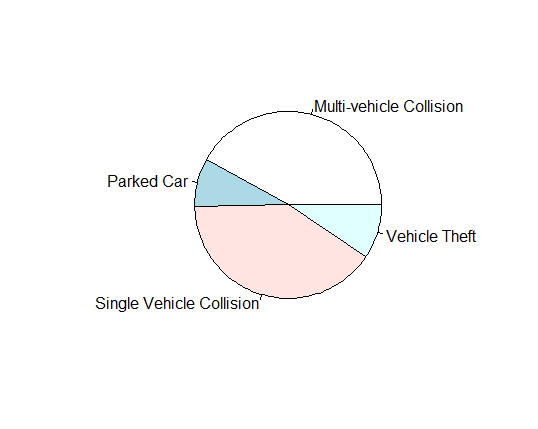
This table gives us a look at what each of the columns look like and what kind of structure of data we are working with. From here I would like to see what the distribution of our predictor variable is and do that by creating a histogram of that column of data.

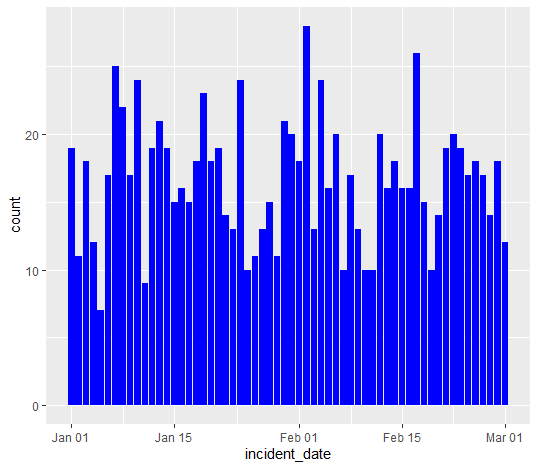


A density plot can give us a smoother look at this data

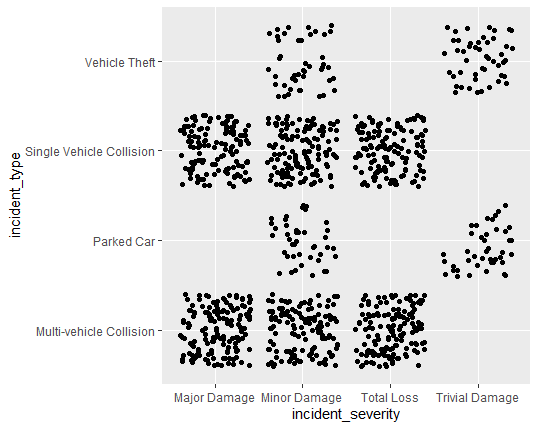


In addition, I would like to see more about the proportion of claims types in the dataset. Pie charts are a good visual for this.

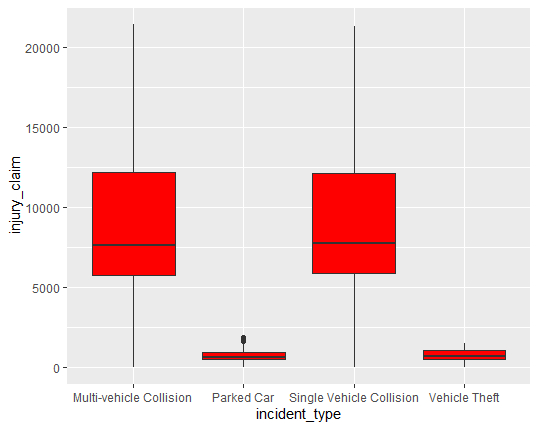




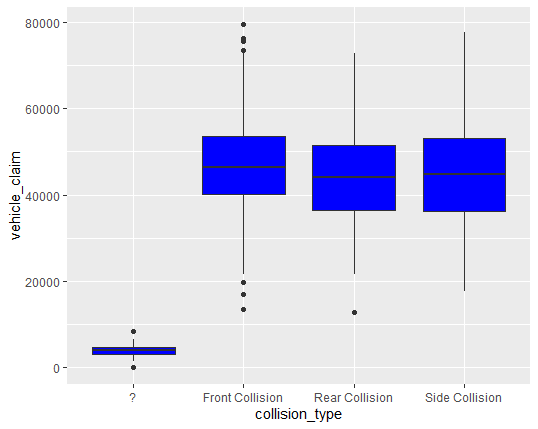
Continuing our initial analysis, having an idea of the range of when these accidents happened can be useful as well. This graph shows us the data was evenly accrued. It also tells us the dates ranged from January through March of that year.



This graph shows us the incident type matched with the incident severity. As suspected, multi-vehicle collisions usually don’t have trivial damages, and vehicle theft damages are mostly minor to trivial. With my background in insurance claims handling, these results confirm my understanding that most theft vehicles that are found have very minor damages, where as any claim that includes more than one other vehicle usually means its was a hard 3 car rear ender that pushed one vehicle into another, or an intersection accident with such speed that the two cars involved were spun or rolled into other vehicles as well.



A boxplot showing types of collisions and their injury claim values was also telling. Vehicle thefts rarely involve accidents with other vehicles causing injury. Additionally parked vehicle accidents don’t usually have injury claims because the parked vehicle doesn’t have anyone in it, or because these accidents are usually very low speed in parking lots.

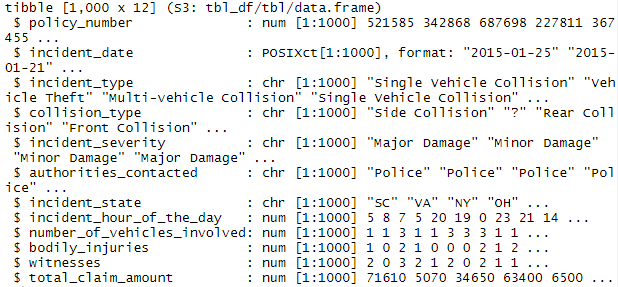


Additionally, a boxplot of the collision types and vehicle claim values does show some difference. Front collisions are usually pretty hard impacts resulting in more damage to the vehicle.

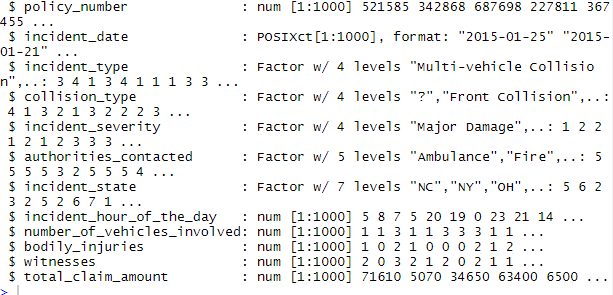
**Data preparation with dplyr**

Now that we have gotten a good understanding of the variables and how they interact with the total claim value, we can start preparing our data for modeling and predicting. The first thing to I wanted to do was remove the separate claims total columns for the injury, property, and vehicle portion of the claims. All these columns add up to our total claim amount. Having them in our prediction model doesn’t really help us gain any new insights to determine claim values from the first notice of the loss.

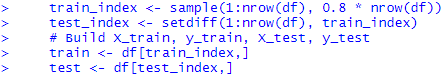
Next, we check the structure of the data from our new subset of data we will be using in our prediction model.



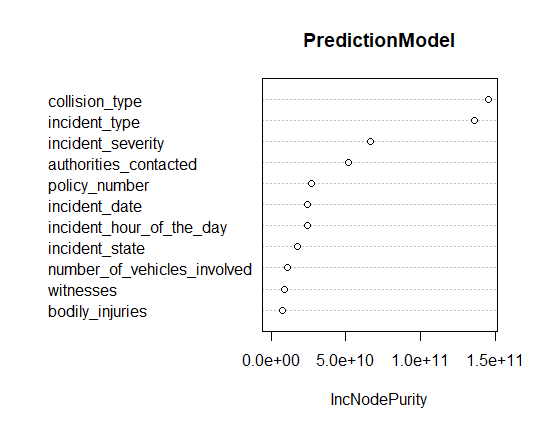
From here we can see that a few of our variable columns are characters instead of factors or numbers. For our prediction model to work with randomForest we will need to fix these columns and make them factors. We use the as.factor() function to do this for each of those columns and end up with this.



Next, we are ready to divide the data into a training and testing set. We do this so that we can train our model to predict the total claim amount with that data in the set and be able to apply that trained model to the portion of the data set that is missing those values. We can then see how our model preformed by comparing the predicted claims values against the actual claim’s values in that testing set. I split the data 80/20, giving our test data 200 of the observations and the training data 800.



From here we can get to the fun part, running our random forest model! The first thing we look at after running the model is which of the variables are the most important in predicting the total claim amounts. The best way to look at these results is in a plot.



This graph shows us that the most important variables are the collision type and incident type. Also, incident severity and if the authorities were contacted are decent predictors as well.

After looking at our important variables, we can run our prediction code and look at our results against our test dataset. Here I used root mean squared error using the rmse function. Our result is what we would use to compare this to new model development in the future.

**Final Thoughts**

Although the dataset has some pretty important variables that could help us predict claim values at an earlier time, overall the dataset is too small. 1000 observations is almost nothing in the world of claims. What would be better is to look at claims from over years of data. This would give our model more room to learn and give us a much better RMSE.

**References**

Arpan Gupta (Indian Institute of Technology, R. (n.d.). Random Forest in R. Retrieved December 04, 2020, from http://analyticsdataexploration.com/random-forest-for-data-analytics-in-r/

Data Wrangling with dplyr and tidyr Cheat Sheet. (n.d.). Retrieved from https://rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

Kabacoff, R. (2015). *R in action: Data analysis and graphics with R*. Shelter Island, NY: Manning.

Misraaakash1998Check out this Author's contributed articles., Misraaakash1998, & Check out this Author's contributed articles. (2020, July 22). Root-Mean-Square Error in R Programming. Retrieved December 04, 2020, from https://www.geeksforgeeks.org/root-mean-square-error-in-r-programming/

Tutorial: Building train and test sets with the same characteristics. (2020, November 13). Retrieved December 04, 2020, from https://cran.r-project.org/web/packages/dataPreparation/vignettes/train\_test\_prep.html