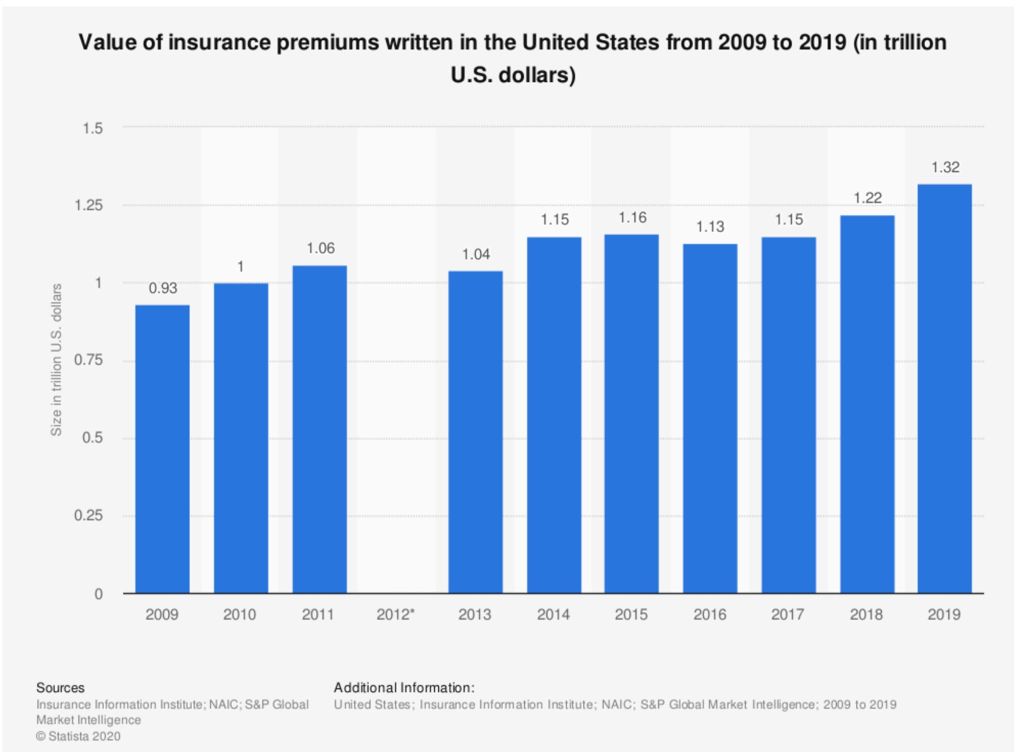
Ardalan Mahdavieh
Professor Christoph Riedl
MSM 6203
12/16/2020

Final Project: Auto Insurance Fraud

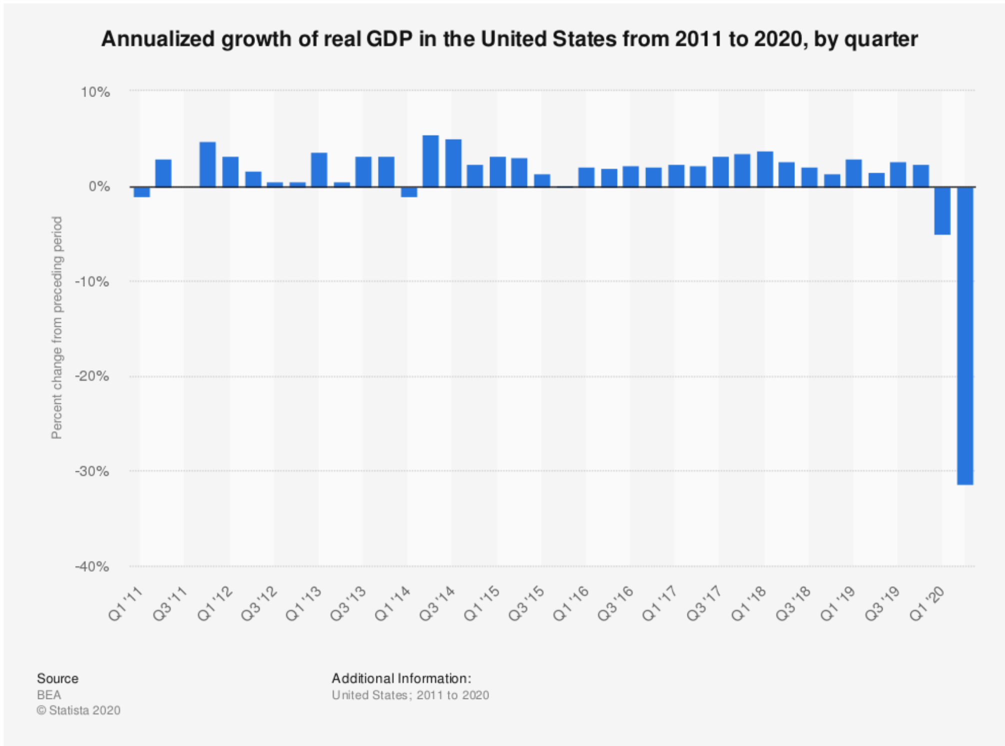
**Business Problem**

The insurance industry is one of the biggest industries in not only the United States, but the world. The global insurance premiums "increased by 2.9% in 2019 to $6.3 trillion dollars" (III, 2020). Meanwhile, the insurance industry in the United States consists of more than 7,000 companies with a combined premium that "reached $1.32 trillion dollars in 2019" (Shaulova,2019).




It is important to note that due to the COVID-19 pandemic, the Insurance Information Institute expects the "insurance industry in the United States to decrease by 6%" [1]. Since the pandemic has taken a toll on the insurance industry, companies are looking at ways to reduce operating costs in order to remain profitable. While many companies are looking to cut down on their workforce or their advertising budget, it is worth looking into insurance fraud to minimize losses.

According to the FBI, "the total cost of insurance fraud is estimated to be more than $40 billion a year" (FBI,2020). Moreover, "fraud frequencies tend to increase during economic distress" (Shaw,2020). A recent study was conducted by the University of Portsmouth, U.K, that "indicates towards a correlation between reduction in GDP and an overall increase in all forms of fraud" (Button, 2019). The study also found that "during the 1980 recession, a 3% fall in GDP resulted in an 5.6% increase in fraud. While fraud increased by 9.9% during the 1990 recession and similarly a 7.3% increase during the 2008 recession" (Button, 2019).




With the US GDP dropping by more than 30% in the second quarter of 2020, it is safe to assume that, unfortunately, the incidence of fraud is going to increase as people look for creative ways to create income during an underperforming economy.

Although there are many different types of insurance fraud some are more vulnerable to fraud than others. According to the Insurance Information Institute, "healthcare, workers compensation, and auto insurance are considered to be the sectors that are most affected" (III,2020). Therefore, we have decided to focus primarily on auto insurance fraud. According to the Insurance Fraud Prevention Authority, some of the most common auto insurance frauds are:

- staged auto accidents and false claims of injury
- false reports of stolen vehicles
- false claims that an accident happened after a policy or coverage was purchased
- false claims for damage that already existed
- claimants who concealed that a person excluded from coverage by their policy was driving at the time of the accident

The main objective of this project is to create a model for the selected dataset to flag suspected fraudulent auto insurance claims. In order to maintain a positive customer service experience, these claims are not rejected outright, but are marked for manual inspection to ensure that it can be properly scrutinized before actually passing or rejecting the claim.

**Data Understanding**

Our dataset, "Insurance Fraud" (Sharma,2019), was obtained through Kaggle and is said to be from a small, unnamed insurance company. The dataset consists of 1000 entries of auto insurance claim records from Ohio, Illinois, and Indiana between January 1, 2015 and March 1, 2015. Before making any adjustments and initiating the data wrangling process, the dataset consisted of 39 numerical and categorical variables. The main variable that our regression model is built on is called reported_fraud, which is labeled by Y or N depending on whether or not the insurance claim was fraudulent.

**Data Preparation**

Before we start working on our model, it is important that we familiarize ourselves with the dataset. The first step we took was to examine every column of our dataset and look for any missing or null values by utilizing the is.null() function; we didn't find any null values, however we found "?" symbols in three of our columns (police_reported_available, property_damage and collusion_type). We decided to keep those observations as they might provide us with some insights, therefore we switched the question mark symbols to "NA".

```
> is.null(inclaims)
[1] FALSE
```

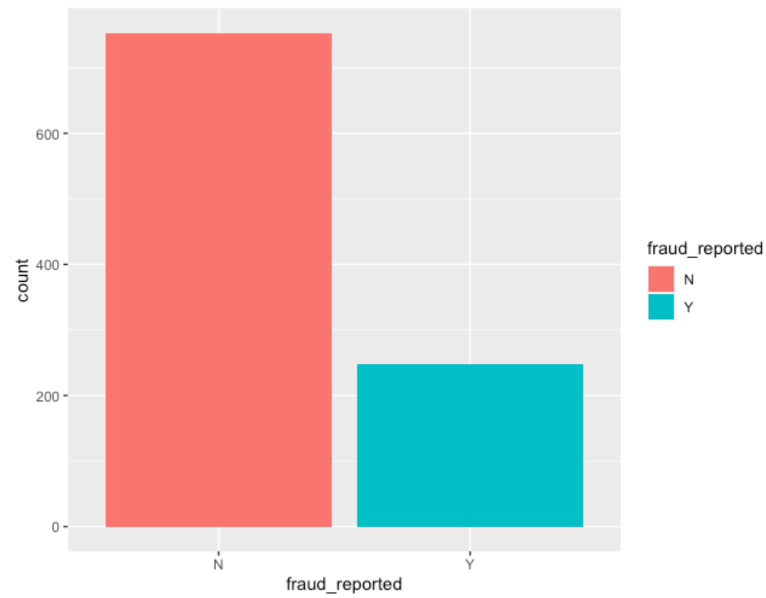
Once we ensured that there aren't any missing values or unknown symbols, we used the sapply() function to determine the classes of our variables. This is a very important step because it allows us to familiarize ourselves with the variable classes and, most importantly, it allows us to check if any changes need to be made to their classes. For example, insurance_zip and policy_number are listed as integers. Although these variables are integers, they need to be categorized as characters because they are simply numerical representations of locations and identifications, respectively. These values cannot be increased or decreased as actual numbers would be, and thus should be considered as characters in reference to this dataset.

```
> sapply(df, class)
         months_as_customer                         age               policy_number
                  "integer"                   "integer"                   "integer"
             policy_bind_dd              policy_bind_mm            policy_bind_yyyy
                  "integer"                   "integer"                   "integer"
               policy_state                  policy_csl           policy_deductable
                "character"                 "character"                   "integer"
      policy_annual_premium              umbrella_limit                 insured_zip
                  "numeric"                   "integer"                   "integer"
                insured_sex     insured_education_level          insured_occupation
                "character"                 "character"                 "character"
            insured_hobbies        insured_relationship               capital.gains
                "character"                 "character"                   "integer"
               capital.loss                 incident_dd                 incident_mm
                  "integer"                   "integer"                   "integer"
              incident_yyyy               incident_type              collision_type
                  "integer"                 "character"                 "character"
          incident_severity         authorities_contacted              incident_state
                "character"                 "character"                 "character"
              incident_city           incident_location    incident_hour_of_the_day
                "character"                 "character"                   "integer"
number_of_vehicles_involved             property_damage              bodily_injuries
                  "integer"                 "character"                   "integer"
                  witnesses      police_report_available          total_claim_amount
                  "integer"                 "character"                   "integer"
               injury_claim              property_claim               vehicle_claim
                  "integer"                   "integer"                   "integer"
                  auto_make                  auto_model                   auto_year
                "character"                 "character"                   "integer"
             fraud_reported            fraud_reported_n
                "character"                   "integer"
```

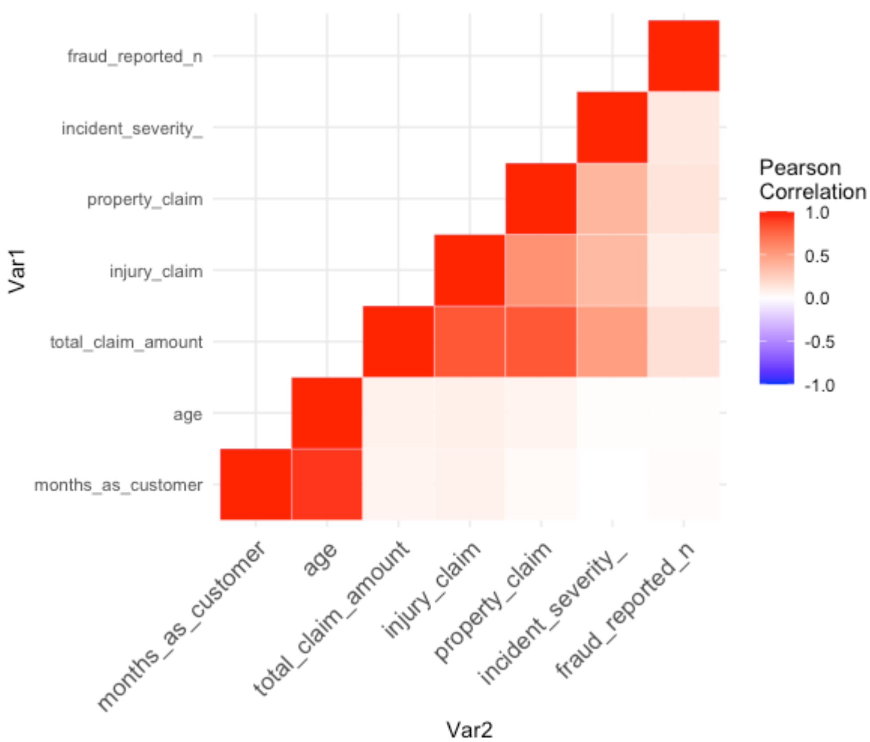
Once we fixed our variable classes, we decided to take a closer look at our most important variable, fraud_reported. This is going be our dependent variable as it tells us whether or not the claim was fraudulent. Since this is a qualitative variable that contains “Y” for yes and “N” for no we decided to create a new quantitative, binary variable called fraud_reported_n; which contains 1 for yes and 0 for no.

Next, we checked to see how many of our observations are fraudulent. Using the count() function we found that 24.7% of our claims are fraudulent. We also created a bar chart to visually inspect the data.

```
> count(inclaims, vars = "fraud_reported")
  fraud_reported freq
1              N  753
2              Y  247
```




After analyzing our dependent variable, we analyzed the correlation between our variables, including our dependent variable. We used variables that had at least a 0.3 Pearson's correlation coefficient to plot a heatmap. The strongest correlation we found was between age and month_as_a_customer, which makes sense; older customers are more likely to own a car and have had car insurance for a longer period of time. There seems to be a correlation between incident severity and claims as well, which also makes sense because as the severity of incidents increase, claim amounts tend to increase. Other than the two aforementioned examples, however, we were unable to find any other correlations and as a result, we did not encounter a multicollinearity problem.



**Modeling**

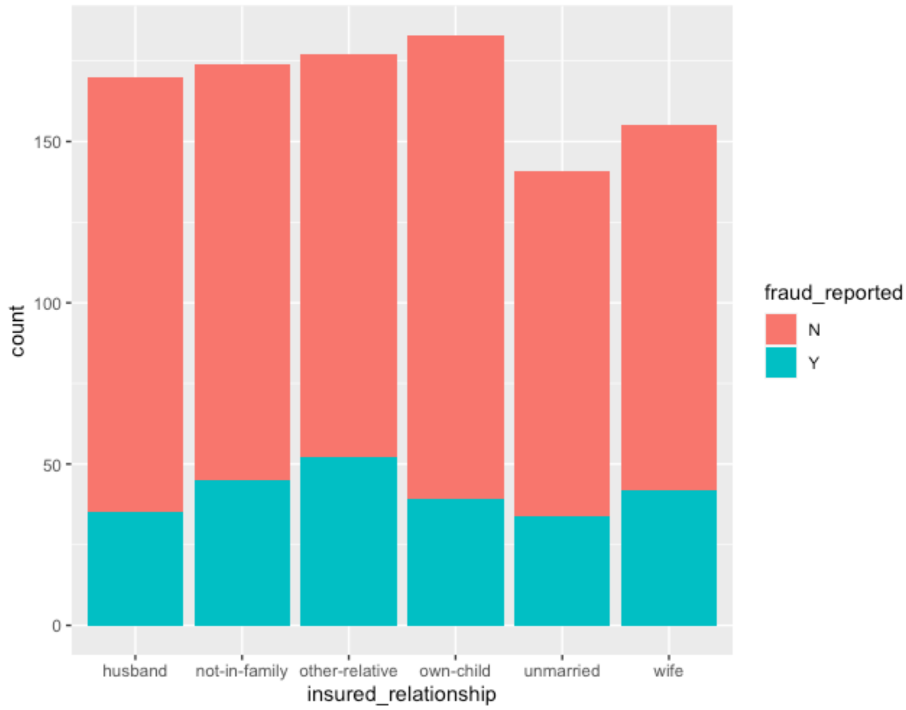
In order to start creating our model, we first have to split our data into training and test/validation sets. This is a crucial part of our project as it allows us to train our model using the training set and then compare our model's performance against a dataset that it has not been trained on. Our training set contains 80% of our data while the test set contains the remaining 20%, leaving us with 800 observations to train our model and 200 observations to test it.

```
#Splitting our data 80/20
split <- round(nrow(inclaims) * .80)

# Create train
train <- inclaims[1:split,]

# Create test
test <- inclaims[(split + 1):nrow(inclaims),]
```

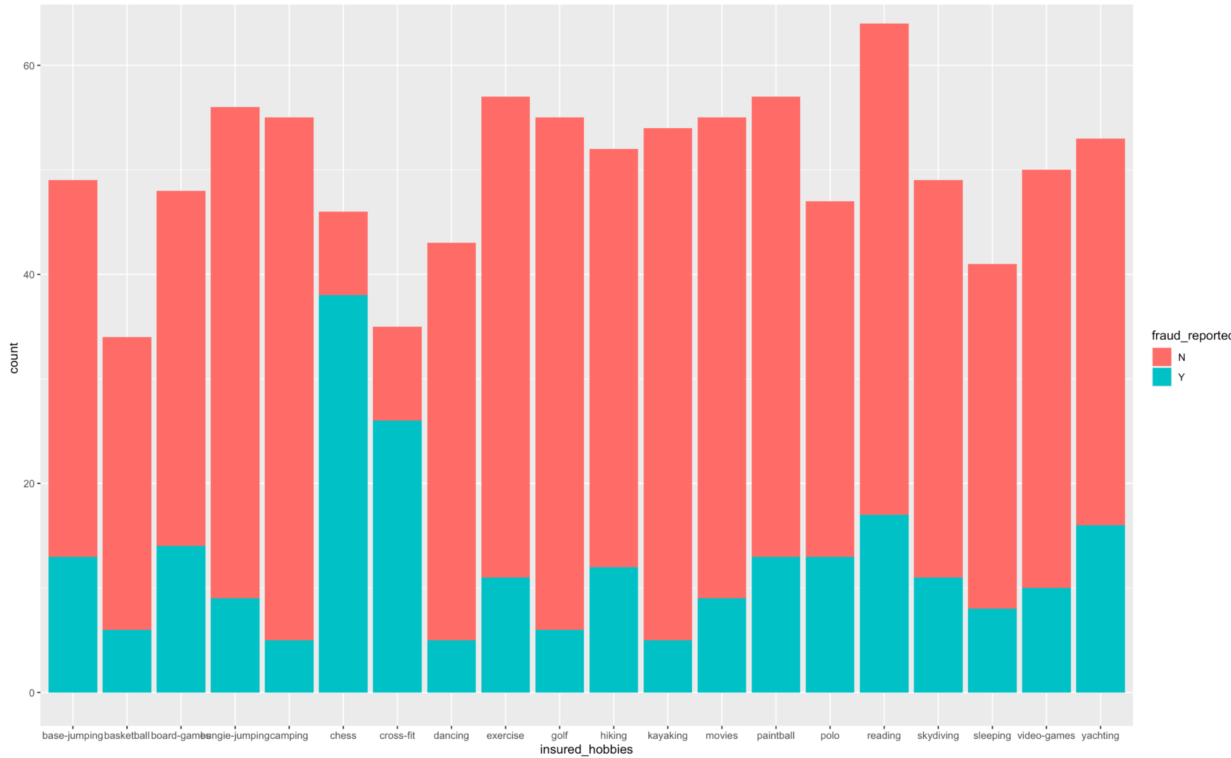
Our goal for the model was to find a model with the highest adjusted R-squared. As the adjusted R-squared increases, the proportion of the variation in our dependent variable (fraud_reported) that can be explained by the variation in our independent variable increases, which ultimately results in a more accurate model. In order to increase the adjusted R-squared, we need only to include independent variables that are significant ($\alpha = 0.05$). Additionally, we looked for a relatively high F1-score with a low p-value to ensure that our model is significant and has a linear relationship. After testing out numerous different models with different independent variables, we decided that the model should only include the following significant variables; insured_relationship, insured_hobbies, and insured_severity. The figure below shows the summary of our final model. It includes all of the coefficient estimates, their standard errors, t-values, and p-values, which are marked by "*" depending on how significant they are.

```
Call:
lm(formula = fraud_reported_n ~ insured_relationship + insured_hobbies +
    incident_severity, data = inclaims)

Residuals:
     Min       1Q   Median       3Q      Max
-1.23363 -0.12484 -0.04634  0.05959  1.02631

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                         0.556895   0.055551  10.025  < 2e-16 ***
insured_relationshipnot-in-family   0.076502   0.035731   2.141  0.03252 *
insured_relationshipother-relative  0.096895   0.035610   2.721  0.00662 **
insured_relationshipown-child       0.027005   0.035446   0.762  0.44632
insured_relationshipunmarried       0.070868   0.038030   1.863  0.06270 .
insured_relationshipwife            0.049889   0.036984   1.349  0.17767
insured_hobbiesbasketball          -0.079829   0.073680  -1.083  0.27888
insured_hobbiesboard-games          0.031384   0.067229   0.467  0.64073
insured_hobbiesbungie-jumping      -0.067245   0.064535  -1.042  0.29768
insured_hobbiescamping             -0.154154   0.064787  -2.379  0.01753 *
insured_hobbieschess                0.600229   0.067740   8.861  < 2e-16 ***
insured_hobbiescross-fit            0.504253   0.073009   6.907 8.95e-12 ***
insured_hobbiesdancing             -0.085833   0.069013  -1.244  0.21390
insured_hobbiesexercise            -0.085012   0.064266  -1.323  0.18621
insured_hobbiesgolf                -0.082750   0.064936  -1.274  0.20285
insured_hobbieshiking              -0.008552   0.065887  -0.130  0.89675
insured_hobbieskayaking            -0.096237   0.065164  -1.477  0.14005
insured_hobbiesmovies              -0.068862   0.064901  -1.061  0.28894
insured_hobbiespaintball           -0.069450   0.064379  -1.079  0.28096
insured_hobbiespolo                -0.001372   0.067285  -0.020  0.98373
insured_hobbiesreading              0.007080   0.062709   0.113  0.91013
insured_hobbiesskydiving           -0.017997   0.066754  -0.270  0.78753
insured_hobbiessleeping            -0.141441   0.069983  -2.021  0.04355 *
insured_hobbiesvideo-games         -0.004440   0.066291  -0.067  0.94661
insured_hobbiesyachting             0.069992   0.065347   1.071  0.28440
incident_severityMinor Damage      -0.502048   0.026815 -18.723  < 2e-16 ***
incident_severityTotal Loss        -0.499920   0.028298 -17.666  < 2e-16 ***
incident_severityTrivial Damage    -0.531658   0.040334 -13.181  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3291 on 972 degrees of freedom
Multiple R-squared:  0.4339,    Adjusted R-squared:  0.4182
F-statistic: 27.59 on 27 and 972 DF,  p-value: < 2.2e-16
```

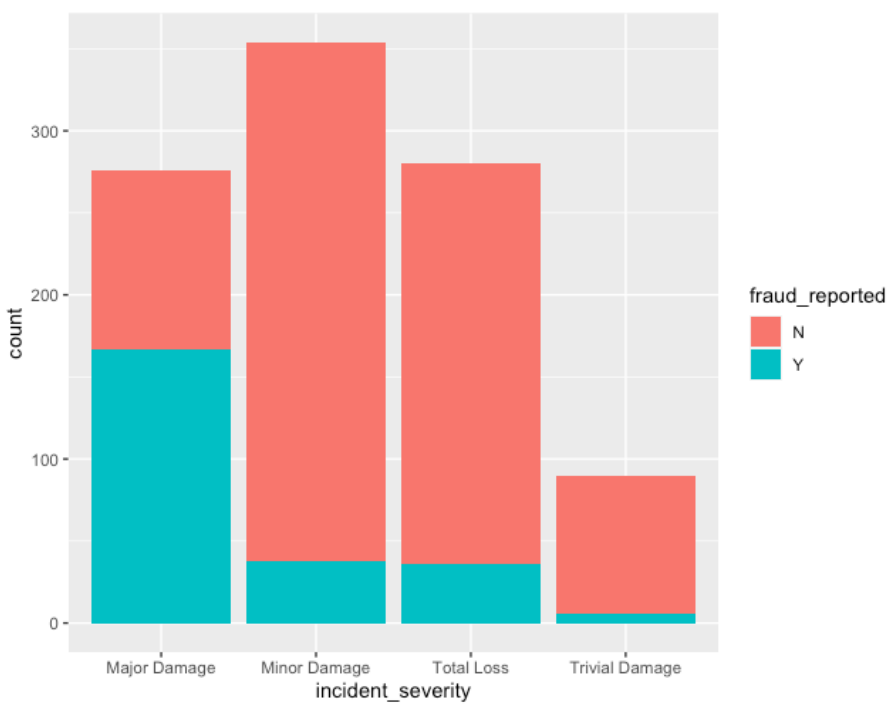
By analyzing the following bar chart, we can see that within the insured_relationship variable, people with other relatives and those who are not in a family have a slightly higher chance of committing insurance fraud. This was reflected in our final regression model, as “other-relative” is highly significant at $\alpha = 0.01$ while not in family is significant at $\alpha = 0.05$. Additionally, unmarried people are also significant at $\alpha = 0.1$. Based on the data, it can be concluded that those who are unmarried and are not part of a family are more likely to commit auto insurance fraud.




Initially, we did not predict that insured_hobbies would be part of our model, however, it appears that people whose hobbies included chess or CrossFit are more likely to commit auto insurance fraud, as shown by the chart below. Looking at our regression model, we can also see that they are both highly significant at $\alpha = 0.001$. We tried to discover a relationship between people who play chess and/or participate in CrossFit and people who are unmarried and not in a family, but there doesn't appear to be a clear correlation between these groups. Meanwhile, people who enjoy camping and sleeping are significant at $\alpha = 0.05$ with a negative coefficient, suggesting that people who enjoy camping and sleeping are less likely to commit insurance fraud.




Our final independent variable is incident_severity. Based on the bar chart below, we can see that incidents with major damage tend to have a higher rate of fraud, while trivial damage, minor damage, and total loss have a very low rate of fraud. Additionally, this is confirmed by our regression model where trivial damage, minor damage, and total loss are all highly significant at $\alpha = 0.001$ with negative coefficients.



**Evaluation**

Our model has an adjusted R-squared score of 0.4182, which means that for the insurance claims in the population that was sampled, 41.82% of the variation incorporating reported fraud, can be explained by variation in our independent variables. Although our model has a relatively low adjusted R-squared score, this was the highest score we could obtain based on the independent variables that were provided within the dataset.

The designed model can be considered a success if it is able to highlight potentially fraudulent claims in the database. This is measured by the F1 score, which can be defined as 2*((precision*recall)/(precision+recall)). The base F1 score of 0.397 is the benchmark that needed to be surpassed. Some of the previous methods and analysis reports on the same data have shown an F1 Score of approximately 0.70. Any significant improvement above this score can be classified as a more appropriate model for flagging fraudulent auto insurance claims.

| Values | |
|---|---|
| F1_Overfit | 0.0359420289855072 |
| f1_test | 0.884057971014493 |
| f1_train | 0.92 |
| Precision_Overfit | 0.00113472250876834 |
| Precision_test | 0.931297709923664 |
| Precision_train | 0.932432432432432 |
| Recall_Overfit | 0.0665154264972777 |
| Recall_test | 0.841379310344828 |
| Recall_train | 0.907894736842105 |

Fortunately, we were able to achieve a relatively high F1 score of 0.92 on our training set. The high F1 score is a result of our precision (0.93) and recall (0.91) scores, which were also relatively high. The precision score is important when we are trying to reduce the chances of a false positive. A high precision score means that our model will rarely falsely identify a non-fraudulent claim as a fraudulent claim. The recall score is even more important to us as we are trying to reduce the chances of a false negative. We do not want our model to falsely predict that a claim is not fraudulent when in reality, it is. Although we were aiming for a higher recall score, we believe that our recall score is still very good. We created a confusion matrix to showcase our predictions for our training dataset. We can see that our model was able to correctly identify 90.79% of non-fraudulent cases and 79.17% of fraudulent cases.

```
> train_matrix

    0   1
0 552  56
1  40 152
```

In order to ensure that our model wasn't overfitting, we used our model on our test set and we received a F1 score of 0.88, precision score of 0.93 and a recall score of 0.84. We were expecting to receive lower scores on our test set as our model hasn't been trained with those observations, but we believe that our test scores are still very close to our training scores.

Therefore, we don't suspect that our model has been overfitted. For further assessment of our test data predictions, we created another confusion matrix. Our model was able to correctly identify 84.13% of non-fraudulent cases and 83.64% of fraudulent cases. These results are very interesting because despite the fact that our model had a lower F1 score for the test data, it performed 4.47% better with the test data in regard to identifying fraudulent cases.

```
> test_matrix

     0   1
  0 122  23
  1   9  46
```

**Management Recommendations**

The goal of our project was to showcase that even smaller companies with limited access to data and resources still have the ability to utilize machine learning.

Our first recommendation for the insurance company is to improve their data maintenance. The dataset didn't have any missing values, but we found multiple cases where "?" were used. Additionally, the dates in the datasets had different formatting that made it almost impossible to use. Moreover, we found cases of negative values in the umbrella_limit column, which does not make sense because an umbrella insurance cannot be a negative value. By simply improving a few things, the company can utilize their dataset even more.

One area in which our model needs improvement is its adjusted R-squared. There is a large amount of random variation that is not captured by our model. By gradually collecting more data on fraudulent and non-fraudulent claims, the insurance company can improve the current model's accuracy levels and create new insights that can eventually increase the adjusted R-squared. The dataset that we were working on only included data for 3 months, which made it difficult to create any insights or spot any seasonality trends. We recommend that the insurance company retrains the model when they get access to new data to ensure that the model is updated and more accurate with current data.

Additionally, the dataset failed to address the different types of auto insurance fraud. We recommend that the insurance company includes this information so they can better understand their data and create more informative insights. For example, one of the most common types of auto insurance fraud is a staged auto accident and a false claim of injury; by including that information in the dataset, we have a good chance of finding a correlation between those specific fraudulent claims and the severity of accidents.

We understand that collecting and maintain datasets can be very expensive and time consuming, especially for smaller companies. However, with the rate of insurance fraud that is expected to increase due to an unstable economy, the cost of collecting and maintaining data can be justified solely based on the amount that they will be able to save. Based on the data, we calculate a total loss of $14,894,620 from the 247 fraudulent claims. That is an average of $60,302 per fraudulent claim. This means that, on average, our model will be able to save the company $12,482,514, assuming that the .084 test_recall score stands true.

## Works Cited

1. Button, Mark. "The Financial Cost of Fraud 2019." *The Latest Data from around the World*, 2019, researchportal.port.ac.uk/portal/files/18625704/The_Financial_Cost_of_Fraud_2019.pdf.
2. FBI. "Insurance Fraud." *FBI*, FBI, 17 Mar. 2020, www.fbi.gov/stats-services/publications/insurance-fraud.
3. III. "Background on: Insurance Fraud." *III*, Property Casualty Insurers Association of America, 2020, www.iii.org/article/background-on-insurance-fraud.
4. Sharma, Roshan. "Insurance Claim." *Kaggle*, 3 July 2019, www.kaggle.com/roshansharma/insurance-claim.
5. Shaulova, Esther, and Lodovica Biagi. "Insurance Industry in the United States." *Statista*, 2020, www.statista.com/study/37002/insurance-sector-in-the-us-statista-dossier/.
6. Shaw, Gary. "2021 Insurance Outlook." *Deloitte Insights*, 3 Dec. 2020, www2.deloitte.com/us/en/insights/industry/financial-services/financial-services-industry-outlooks/insurance-industry-outlook.html.
7. III."World Insurance Marketplace." *Insurance Information Institute* , 1 Apr. 2020, www.iii.org/publications/insurance-handbook/economic-and-financial-data/world-insurance-marketplace.

## Appendices

(1) how you did each of the above-mentioned steps

**The approach in this project consisted of the following steps:**

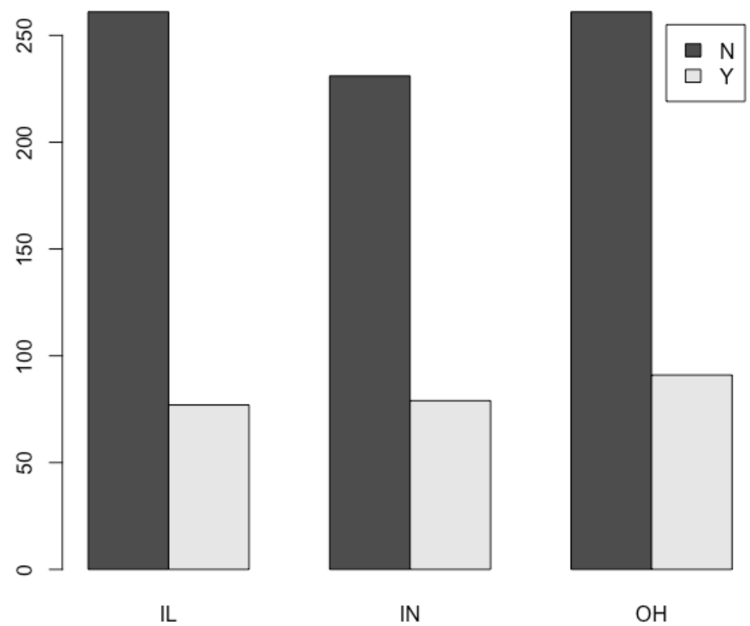
a. selection of database
b. cleaning of database
c. preprocessing of database to adjust the missing values
d. conversion of variables to processable data form
e. division of database into train and test database
f. evaluation of database to find out the significant model design
g. model making using the train database and
h. finally evaluating the performance of the model using its performance over the test dataset.

## (2) Any auxiliary analyses or visualizations that did not fit into the body of the report

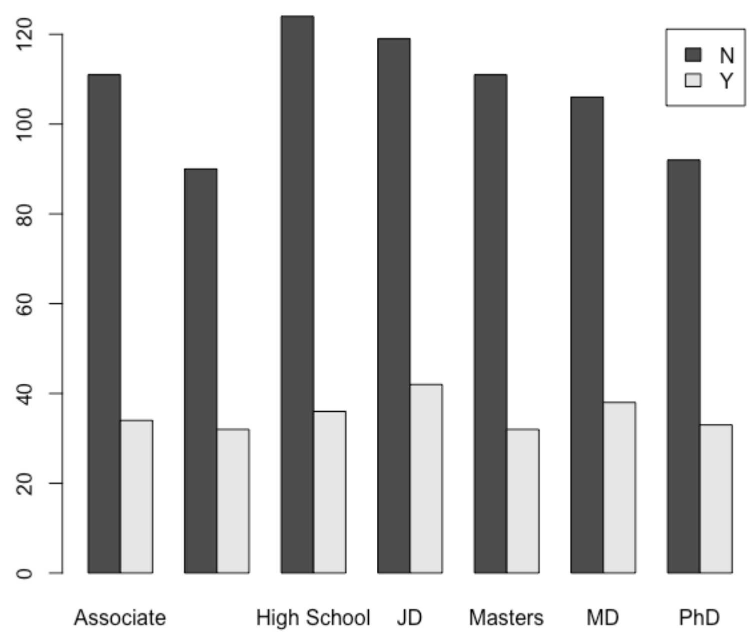
Summary of our dataframe:

```
> summary(df)
 months_as_customer      age         policy_number    policy_bind_dd  policy_bind_mm
 Min.   :  0.0      Min.   :19.00   Min.   :100804   Min.   : 1.00   Min.   : 1.000
 1st Qu.:115.8      1st Qu.:32.00   1st Qu.:335980   1st Qu.: 8.00   1st Qu.: 4.000
 Median :199.5      Median :38.00   Median :533135   Median :16.00   Median : 7.000
 Mean   :204.0      Mean   :38.95   Mean   :546239   Mean   :15.45   Mean   : 6.559
 3rd Qu.:276.2      3rd Qu.:44.00   3rd Qu.:759100   3rd Qu.:23.00   3rd Qu.: 9.000
 Max.   :479.0      Max.   :64.00   Max.   :999435   Max.   :31.00   Max.   :12.000
 policy_bind_yyyy policy_state        policy_csl        policy_deductable policy_annual_premium
 Min.   :1990     Length:1000        Length:1000        Min.   : 500      Min.   : 433.3
 1st Qu.:1995     Class :character   Class :character   1st Qu.: 500      1st Qu.:1089.6
 Median :2002     Mode  :character   Mode  :character   Median :1000      Median :1257.2
 Mean   :2002                                           Mean   :1136      Mean   :1256.4
 3rd Qu.:2008                                           3rd Qu.:2000      3rd Qu.:1415.7
 Max.   :2015                                           Max.   :2000      Max.   :2047.6
 umbrella_limit      insured_zip     insured_sex        insured_education_level insured_occupation
 Min.   :-1000000   Min.   :430104   Length:1000        Length:1000             Length:1000
 1st Qu.:       0   1st Qu.:448404   Class :character   Class :character        Class :character
 Median :       0   Median :466446   Mode  :character   Mode  :character        Mode  :character
 Mean   : 1101000   Mean   :501214
 3rd Qu.:       0   3rd Qu.:603251
 Max.   :10000000   Max.   :620962
 insured_hobbies    insured_relationship capital.gains     capital.loss      incident_dd
 Length:1000        Length:1000          Min.   :     0   Min.   :-111100   Min.   : 1.00
 Class :character   Class :character     1st Qu.:     0   1st Qu.: -51500   1st Qu.: 2.00
 Mode  :character   Mode  :character     Median :     0   Median : -23250   Median :15.00
                                         Mean   : 25126   Mean   : -26794   Mean   :13.08
                                         3rd Qu.: 51025   3rd Qu.:      0   3rd Qu.:22.00
                                         Max.   :100500   Max.   :      0   Max.   :31.00
  incident_mm     incident_yyyy  incident_type      collision_type     incident_severity
 Min.   : 1.000   Min.   :2015   Length:1000        Length:1000        Length:1000
 1st Qu.: 1.000   1st Qu.:2015   Class :character   Class :character   Class :character
 Median : 2.000   Median :2015   Mode  :character   Mode  :character   Mode  :character
 Mean   : 3.407   Mean   :2015
 3rd Qu.: 5.000   3rd Qu.:2015
 Max.   :12.000   Max.   :2015
 authorities_contacted incident_state     incident_city      incident_location
 Length:1000           Length:1000        Length:1000        Length:1000
 Class :character      Class :character   Class :character   Class :character
 Mode  :character      Mode  :character   Mode  :character   Mode  :character


 incident_hour_of_the_day number_of_vehicles_involved property_damage    bodily_injuries
 Min.   : 0.00            Min.   :1.000               Length:1000        Min.   :0.000
 1st Qu.: 6.00            1st Qu.:1.000               Class :character   1st Qu.:0.000
 Median :12.00            Median :1.000               Mode  :character   Median :1.000
 Mean   :11.64            Mean   :1.839                                  Mean   :0.992
 3rd Qu.:17.00            3rd Qu.:3.000                                  3rd Qu.:2.000
 Max.   :23.00            Max.   :4.000                                  Max.   :2.000
   witnesses     police_report_available total_claim_amount  injury_claim    property_claim
 Min.   :0.000   Length:1000             Min.   :   100     Min.   :    0   Min.   :    0
 1st Qu.:1.000   Class :character        1st Qu.: 41812     1st Qu.: 4295   1st Qu.: 4445
 Median :1.000   Mode  :character        Median : 58055     Median : 6775   Median : 6750
 Mean   :1.487                           Mean   : 52762     Mean   : 7433   Mean   : 7400
 3rd Qu.:2.000                           3rd Qu.: 70592     3rd Qu.:11305   3rd Qu.:10885
 Max.   :3.000                           Max.   :114920     Max.   :21450   Max.   :23670
 vehicle_claim    auto_make          auto_model          auto_year     fraud_reported
 Min.   :   70   Length:1000        Length:1000        Min.   :1995   Length:1000
 1st Qu.:30292   Class :character   Class :character   1st Qu.:2000   Class :character
 Median :42100   Mode  :character   Mode  :character   Median :2005   Mode  :character
 Mean   :37929                                         Mean   :2005
 3rd Qu.:50822                                         3rd Qu.:2010
 Max.   :79560                                         Max.   :2015
 fraud_reported_n
 Min.   :0.000
 1st Qu.:0.000
 Median :0.000
 Mean   :0.247
 3rd Qu.:0.000
 Max.   :1.000
```

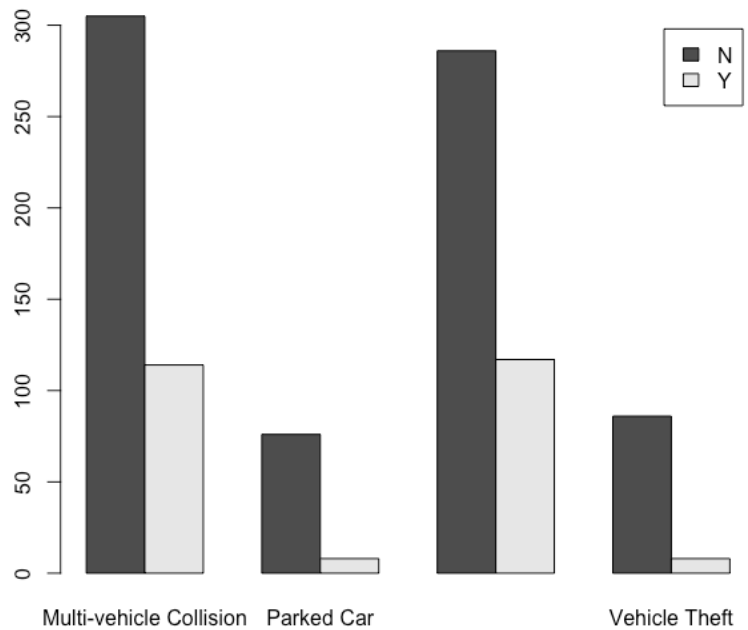
policy_state in relation to whether fraud was committed or not



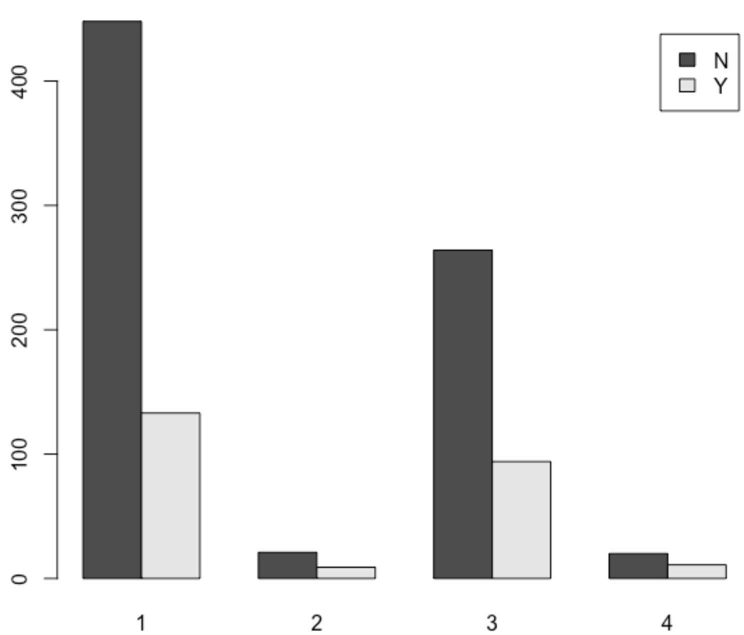

insured_education_level in relation to whether fraud was committed or not



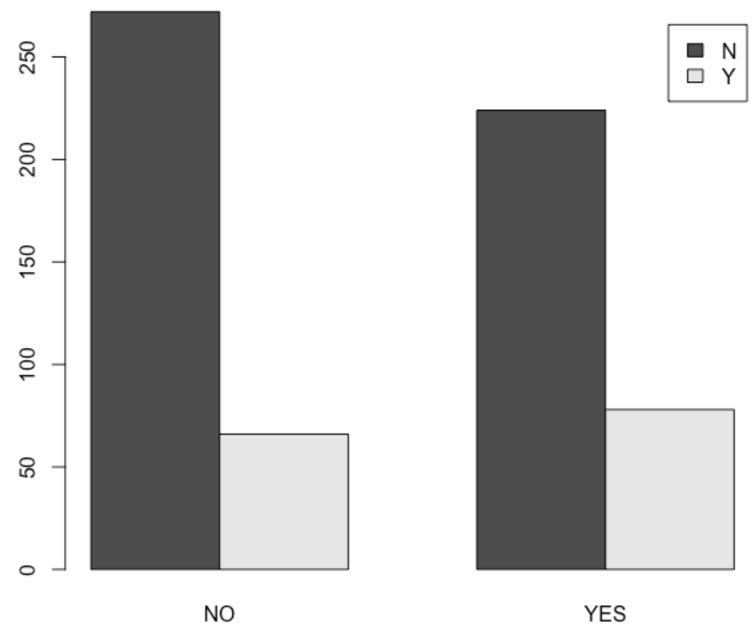
incident_type in relation to whether fraud was committed or not



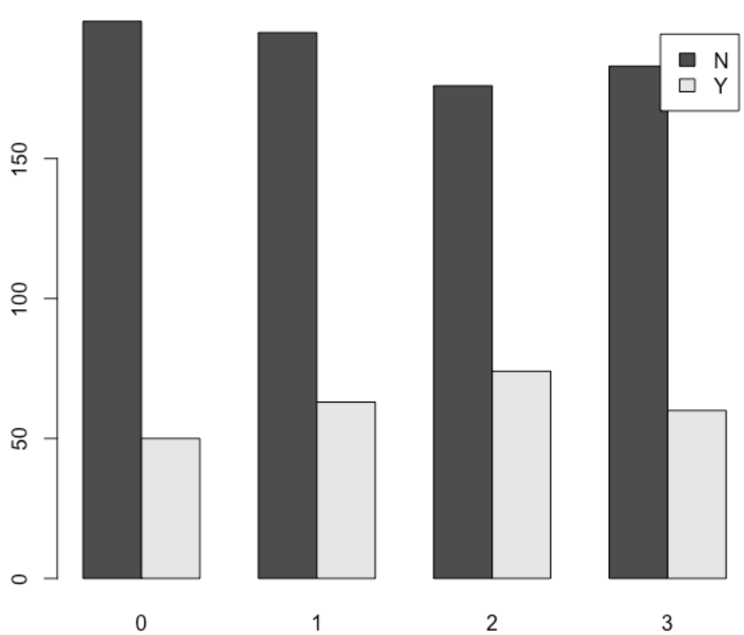

Number_of_vehicles_involved in relation to whether fraud was committed or not



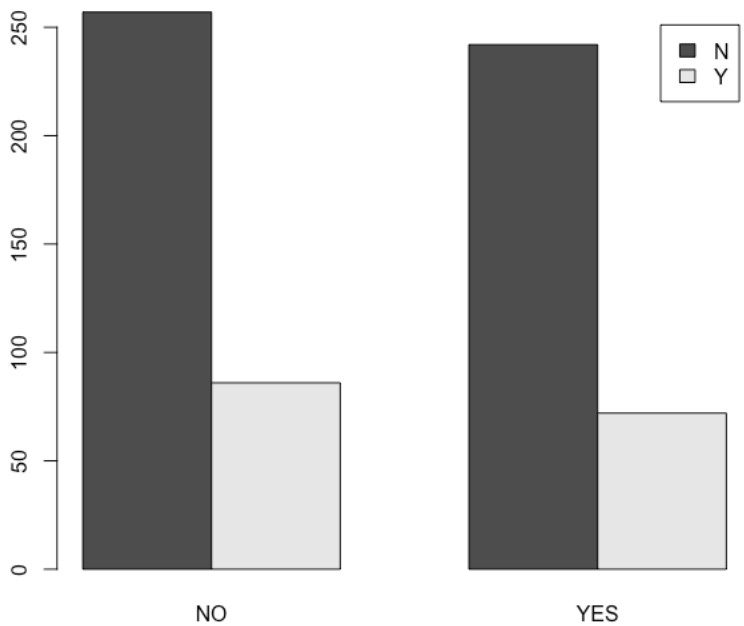
property_damage in relation to whether fraud was committed or not



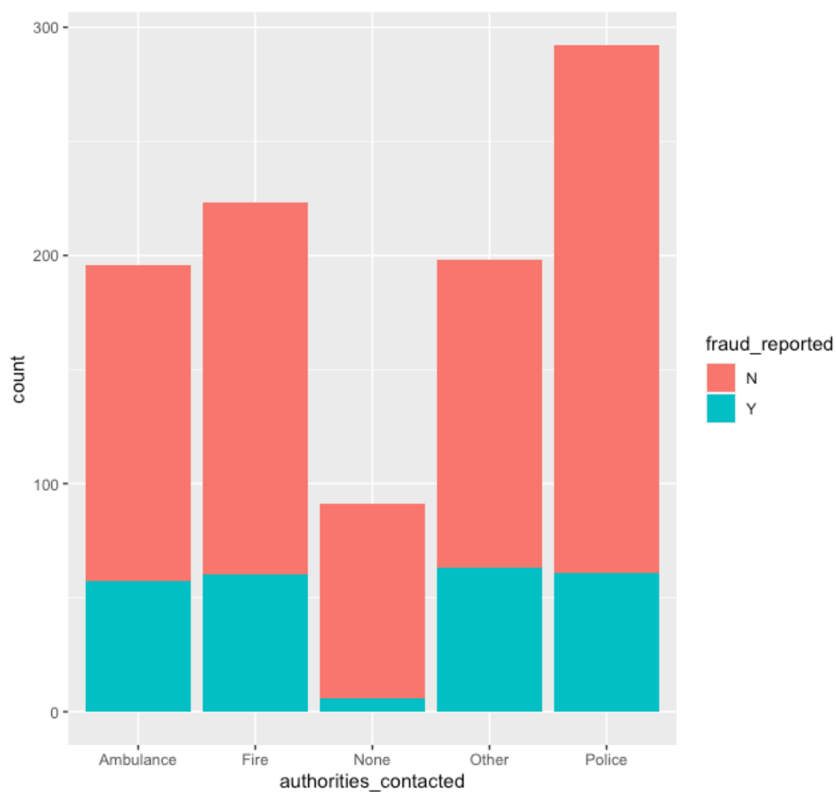

Number of witnesses in relation to whether fraud was committed or not



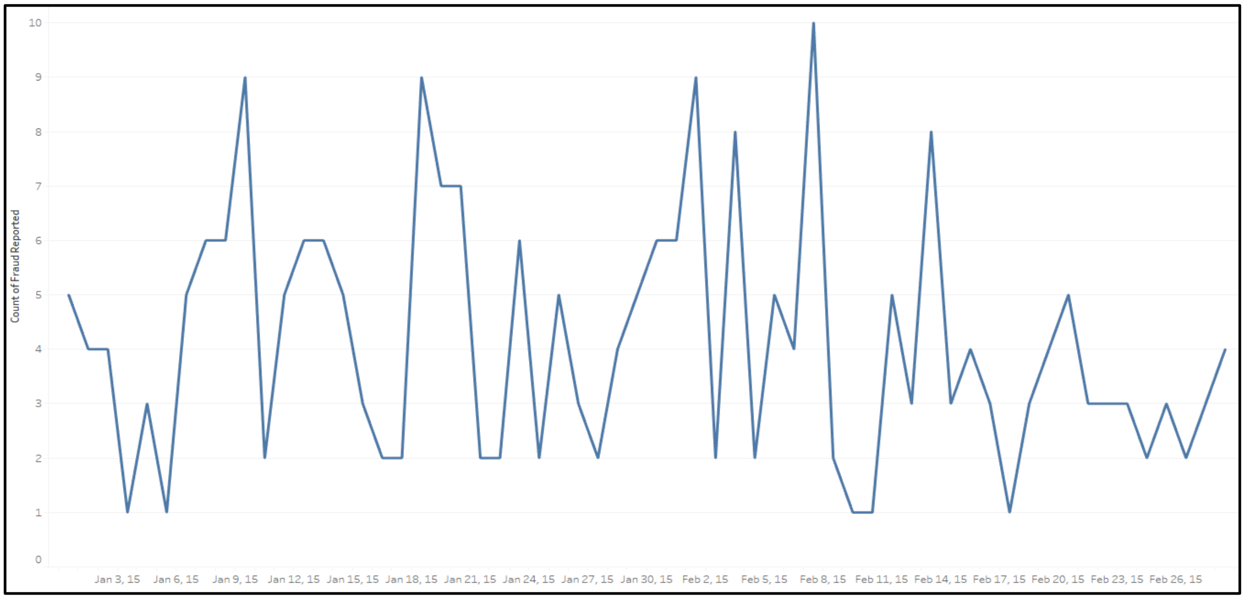
Police report available in relation to whether fraud was committed or not




Authorities contacted or not in relation to whether fraud was committed or not

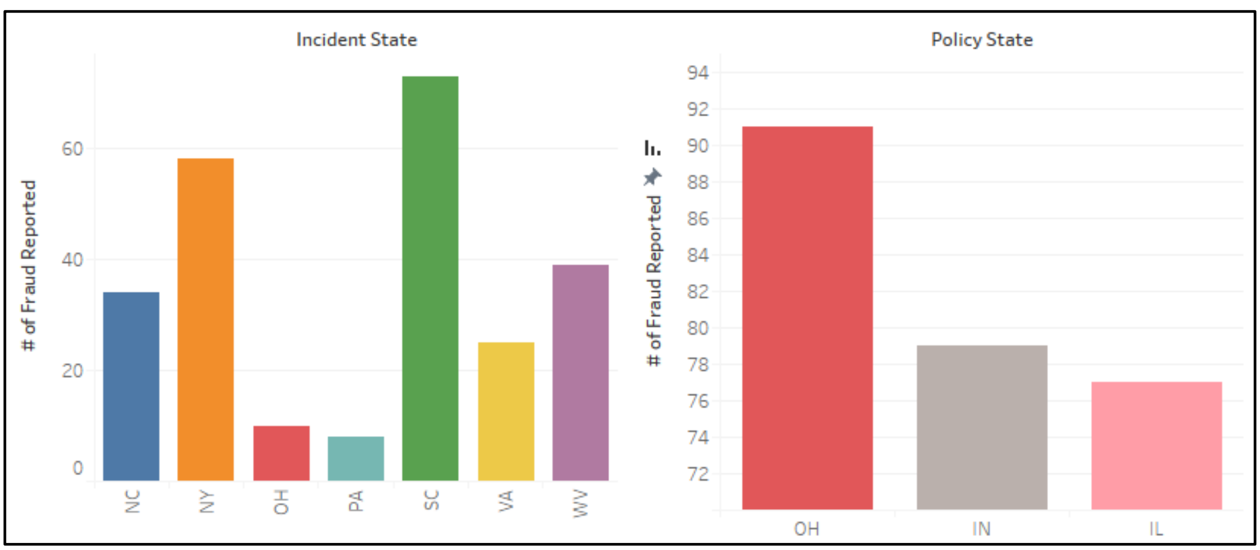


Fraudulent claims over the time period



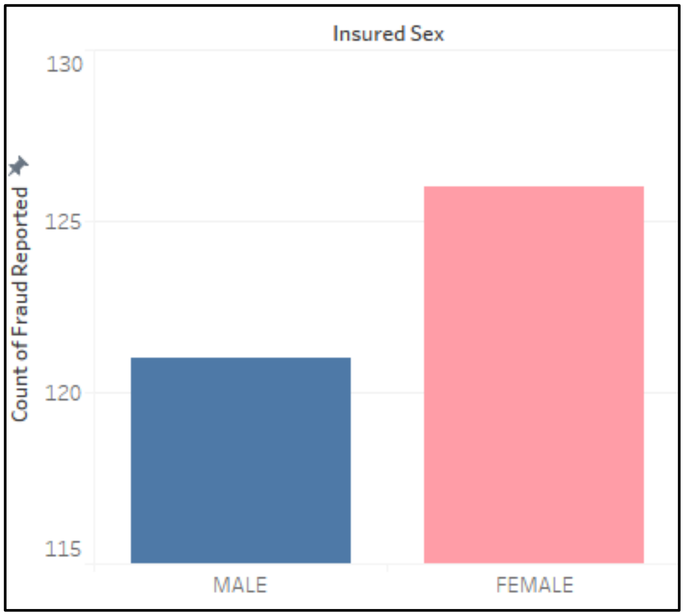

We can clearly see here that the fraudulent claims are being generated continuously over the period of time. There is no specific seasonality for the fraudulent claim generation.

Policy State/ Incident State wise Fraud Count




According to the visualization incident occurring in fraudulent claims are out of the policy state. Most of the claims marked as fraudulent except some from Ohio can be seen as out of state claims.

Gender wise fraudulent claims generation




It is important to note that a greater number of fraud cases being reported are made in the name of females than those being done in the name of males. It might not be a coincidence but the insurance premium of women is lesser due to lesser kms driven and the careful approach and hence increase on that might not be a significant impact.

## (3) All code written

## Initial data analyses

```r
df <-read.csv("~/Downloads/inclaims_updated.csv")
library(caret)
View(df)

# dimensions of dataset
dim(df)

# list types for each attribute
sapply(df, class)

# summarize attribute distributions
summary(df)

#check for null values
is.null(df)

#Assessing fraud rate
mean(df$fraud_reported)
#24.7% fraud
count <- table(df$fraud_reported)
barplot(count)

#looking at relationships
hobbies_reg <- lm(fraud_reported_n ~ insured_hobbies , data = df)
summary(hobbies_reg)

mc_reg <- lm(fraud_reported_n ~ months_as_customer , data = df)
summary(mc_reg)

is_reg <- lm(fraud_reported_n ~ incident_severity , data = df)
summary(is_reg)

#Bar chart for assessment
count <- table(df$fraud_reported, df$policy_state)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$insured_education_level)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$insured_relationship)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$incident_type)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$incident_severity)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$number_of_vehicles_involved)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$property_damage)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$witnesses)
barplot(count, beside = TRUE, legend = rownames(count))

count <- table(df$fraud_reported, df$police_report_available)
barplot(count, beside = TRUE, legend = rownames(count))

mean(df$total_claim_amount)
```

## More in depth data analyses

```r
inclaims <- read_csv("~/Downloads/claims.csv")
library(plyr)
library(MLmetrics)

#fraud report bar chart
ggplot(data = inclaims, aes(x=fraud_reported, fill =fraud_reported )) + geom_bar()
count <-table(inclaims$fraud_reported)

#Hobbies
hobbies_reg <- lm(fraud_reported_n ~ insured_hobbies, data = inclaims)
summary(hobbies_reg)
ggplot(data = inclaims, aes(x=insured_hobbies, fill =fraud_reported )) + geom_bar()

#Incident Severity
incident_reg <- lm(fraud_reported_n ~ incident_severity, data=inclaims)
summary(incident_reg)
ggplot(data = inclaims, aes(x=incident_severity, fill =fraud_reported )) + geom_bar()

#relationship
relationship_reg <- lm(fraud_reported_n ~ insured_relationship, data=inclaims)
summary(relationship_reg)
ggplot(data = inclaims, aes(x=insured_relationship, fill =fraud_reported )) + geom_bar()

#authorities_contacted
authorities_reg <- lm(fraud_reported_n ~ authorities_contacted, data=inclaims)
summary(authorities_reg)
ggplot(data = inclaims, aes(x=authorities_contacted, fill =fraud_reported )) + geom_bar()

#Correlation Matrix
mydata <- inclaims[, c(1,2,32,33,34,41,40)]
data.frame(colnames(inclaims$total_claim_amount))
cormat <- round(cor(mydata),2)

melted_cormat <- melt(cormat)
head(melted_cormat)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()

# Get lower triangle of the correlation matrix
get_lower_tri<-function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)}

upper_tri <- get_upper_tri(cormat)
upper_tri

# Melt the correlation matrix
library(reshape2)
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Heatmap
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()
```

**Machine Learning**

```
library(readr)
library(MLmetrics)
inclaims <- read_csv("~/Downloads/claims.csv")

#Splitting our data 80/20
split <- round(nrow(inclaims) * .80)

# Create train
train <- inclaims[1:split,]

# Create test
test <- inclaims[(split + 1):nrow(inclaims),]

#Model
reg <- lm(fraud_reported ~ ., data = train)
summary(reg)

#Prediction for Train data set
train$prediction <- round(predict(reg, train))

#Recall Score for Training Data
Recall_train <- Recall(train$fraud_reported, train$prediction, positive = NULL)

#Precision for Training data set
Precision_train <- Precision(train$fraud_reported, train$prediction, positive = NULL)

#F1 Score for Training data set
f1_train <- F1_Score(y_pred = train$prediction, y_true = train$fraud_reported)

#Prediction for Test data set
test$prediction <- round(predict(reg, test))

#Recall Score for Test Data
Recall_test <- Recall(test$fraud_reported, test$prediction, positive = NULL)

#Precision for Test data set
Precision_test <- Precision(test$fraud_reported, test$prediction, positive = NULL)

#F1 Score for Test data set
f1_test <- F1_Score(y_pred = test$prediction, y_true = test$fraud_reported)

#Checking for Overfitting
Recall_Overfit  <- Recall_train - Recall_test
Precision_Overfit <- Precision_train - Precision_test
F1_Overfit <- f1_train - f1_test

#confusion Matrix for train
train_matrix <- table(train$fraud_reported, train$prediction)
train_matrix

#confusion Matrix for Test
test_matrix <- table(test$fraud_reported, test$prediction)
test_matrix

```

(4) a detailed description of each group member's specific contributions to the project.

- Ardalan’s contribution:
    - Wrote the entirety of the paper besides the second paragraph on page 10 and appendix part 1.
    - Worked on all R visualization.
    - Worked on finding relationships and insights.
    - Contributed to data wrangling.
    - Contributed to creating the modeling.
    - Worked on assessing the model.
    - Worked on the business problem and solutions.
    - Co-presented on both of the presentations.