Avery Fulton
Zain Elsell
STAT 3400
Project Paper I
3/19/2023

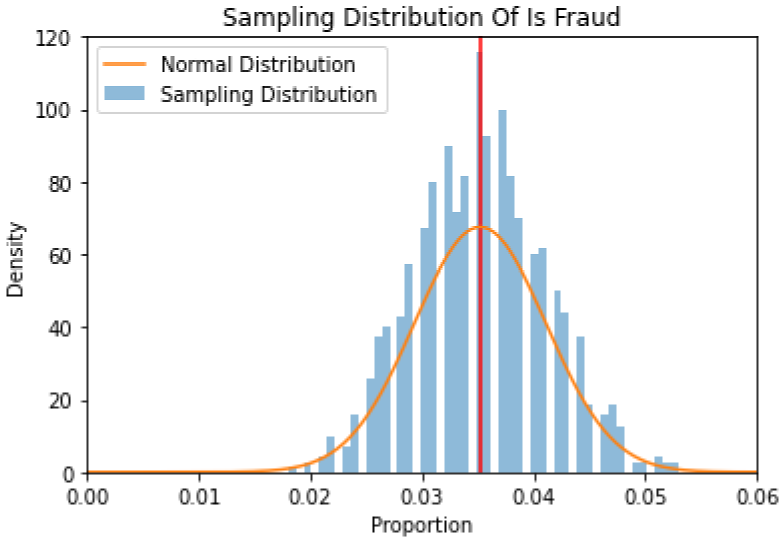
Overview:

As stated in our project proposal our goal is to try to create a logistic regression fraud detection algorithm from the IEEE fraud-detection data set located on Kagel. The primary objectives of this paper are to analyze the response variable in both problem and mathematical contexts, evaluate potential predictor variables, identify and address inconsistencies or outliers, establish confidence intervals for some predictor variables, and determine high correlation coefficients between them.

The initial step was importing the required dependencies that we are going through and then performing some rudimentary data wrangling. Initially, what we are doing is importing the data for the train and test sets that were given to us then merging the identity data with the transaction data for both our train and test sets on the TransactionID column so we can use only two .csv documents as opposed to the given four. The next step in the wrangling process is taking a consistent sample of 10% of the given data; the primary motivation behind this choice was to reduce the compute time for a select few manipulations and operations, as the initial data set contains 590,540 data points and 434 possible features. Note that we will run the created model on the overall data set to consistently represent the performance in the context of a larger sample.

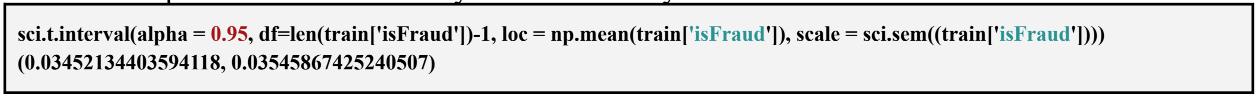
Response Variable:

To find the distribution of our response variable "isFraud", we calculated the proportion of fraudulent transactions (approx. 3.499%) and used it to construct a binomial sampling distribution. By comparing the generated samples to a normal distribution centered at the mean of sampled proportions, we obtained the sampling distribution shown below.



$$\text{Let } F := \text{A transaction is fraudulent}$$
$$F \sim \text{Normal}(\mu = 0.35213, \sigma^2 = 3.4923631 \cdot 10^{-5})$$

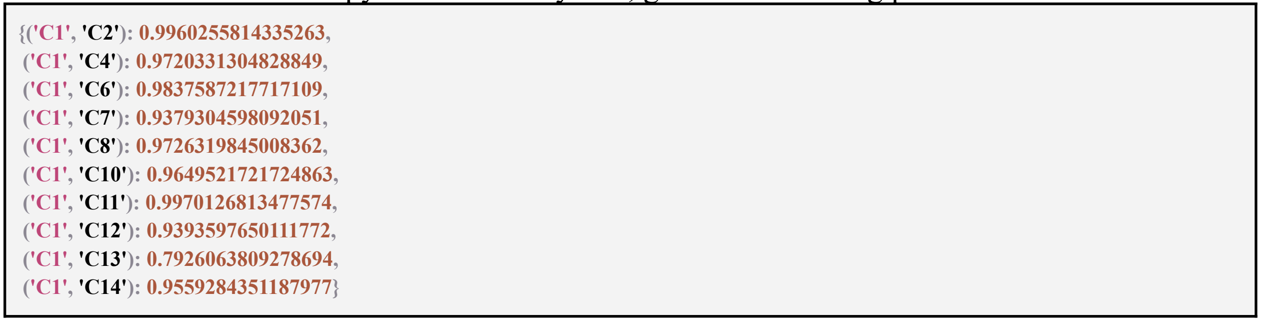
By applying the central limit theorem to the larger training dataset, we can draw the following conclusions. The initially calculated proportion reveals that the majority of transactions are non-fraudulent and normally distributed, which aligns with what we would expect to see in reality. Finally to determine the true proportion of fraudulent transactions we constructed a 95% confidence interval, to construct this confidence interval we used the sci-py library on our response variable giving us the result that with 95% confidence that the true proportion of fraudulent transactions is somewhere between 3.45% and 3.54%. Additionally, since our response variable is binary there aren't any outliers.



```
sci.t.interval(alpha = 0.95, df=len(train['isFraud'])-1, loc = np.mean(train['isFraud']), scale = sci.sem((train['isFraud'])))
(0.03452134403594118, 0.03545867425240507)
```

Multicollinearity:

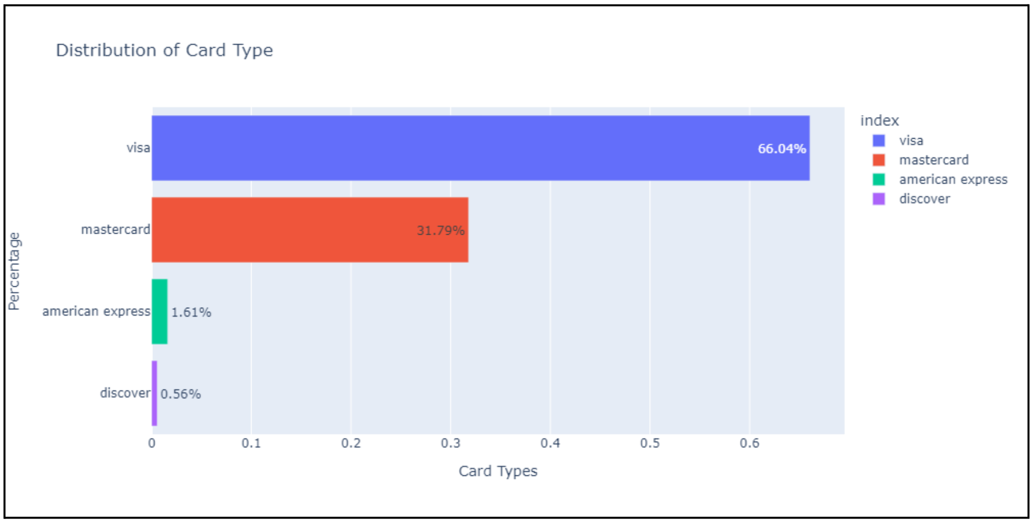
To preface, we had 433 possible predictor variables, and keeping that in mind we had to change some of the methods used to identify collinearity between our variables. Initially, we tried to plot a typical heat map to show the values but this visualization proved to be ineffective given the number of variables. Alternatively, the process used to find multicollinearity within our predictor variables was first computing a correlation matrix using the smaller 10% subset of the initial data. We then took the list of correlation coefficients and of each pair of possible predictor variables from the matrix, dropped duplicates so we were using a lower triangular matrix, took the absolute value of every element in that list, and finally, we filtered the variables on the conditions that strong correlation exists or if our correlation coefficient was greater than or equal to 0.70. This resulted in a python dictionary that, gave the following pairs:



```
{('C1', 'C2'): 0.9960255814335263,
 ('C1', 'C4'): 0.9720331304828849,
 ('C1', 'C6'): 0.9837587217717109,
 ('C1', 'C7'): 0.9379304598092051,
 ('C1', 'C8'): 0.9726319845008362,
 ('C1', 'C10'): 0.9649521721724863,
 ('C1', 'C11'): 0.9970126813477574,
 ('C1', 'C12'): 0.9393597650111772,
 ('C1', 'C13'): 0.7926063809278694,
 ('C1', 'C14'): 0.9559284351187977}
```

The output shows both high positive and negative correlations between variables. However, the variables themselves are masked for privacy reasons. As outlined in the competition guidelines each variable (C1-C15) represents counts of different criteria, for example, one may show the number of linked addresses to a payment card. Now even though the true meaning of the predictors is redacted it's clear that the counts would be dependent on each other, since they are parallel functions of one another. Due to the collinearity between these variables, we would likely remove them from our final model to reduce instability. Whether or not the variables have an effect on the model generated is going to be dependent on the model we pick in the future.

Predictor Variables:

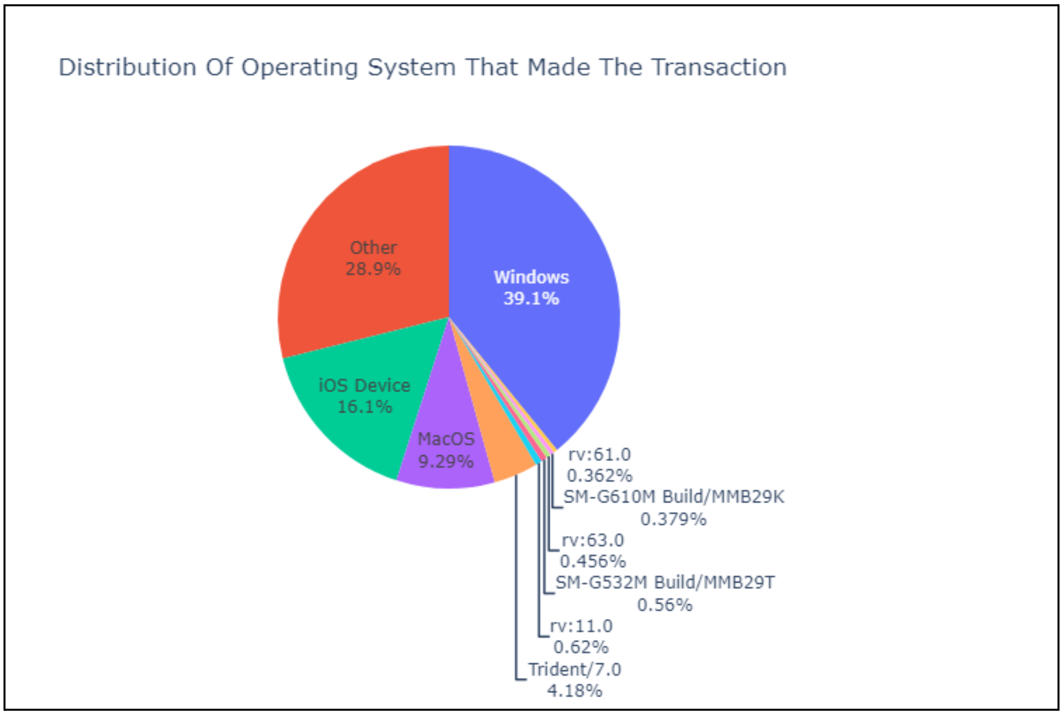


We picked a handful of variables to find distributions for. Some of these variables include: Card Type, Operating system used, Transaction Amount, and a few other variables regarding the infrastructure used to make the transaction. As can be seen in the graph, the majority of cardholders in our data use Visa followed by MasterCard, American Express, and finally Discover. This show's the distribution skewed towards Visa and because this is a categorical variable, the only outliers identified were Discover cardholders at 0.56% of the transactions. Additionally, we constructed 95% confidence intervals for each proportion of these variables resulting in the below values.

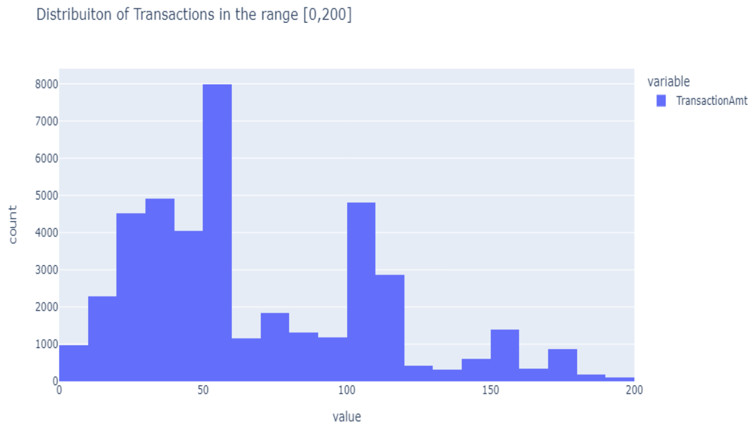
```
=============================================
visa proportion: 0.6649695694045928
Sample Proportion: 0.6649695694045928
Confidence Interval (95%): (0.6636659610605307, 0.666273177748655)
=============================================
mastercard proportion: 0.31407352984978304
Confidence Interval (95%): (0.31279161680394424, 0.31535544289562184)
=============================================
american express proportion: 0.015252032843200524
Confidence Interval (95%): (0.014913554777454948, 0.015590510908946099)
=============================================
discover proportion: 0.005704867902423526
Confidence Interval (95%): (0.005496857731396278,0.005912878073450775)
=============================================
```

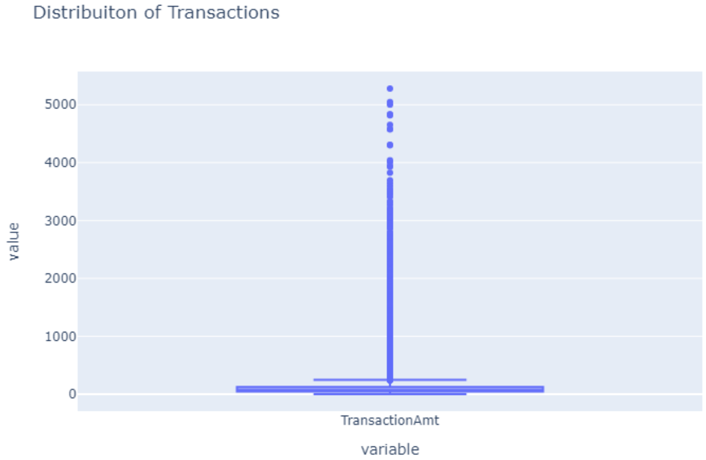
Confidence intervals at 95% show that the true proportion of each card type falls within the range of the respective sample proportion.

We considered the different types of operating systems used to make a transaction. As seen below, we can see the general distribution of the different types of OSs. This makes sense in the real world as there are a variety of operating systems and software that people use to manage and conduct their transactions. A lot of the data collected contains outliers meaning that out of the 50,000 rows sampled, only one had a unique OS. During our analysis, we must be careful to distinguish between suspicious and well-known operating systems.

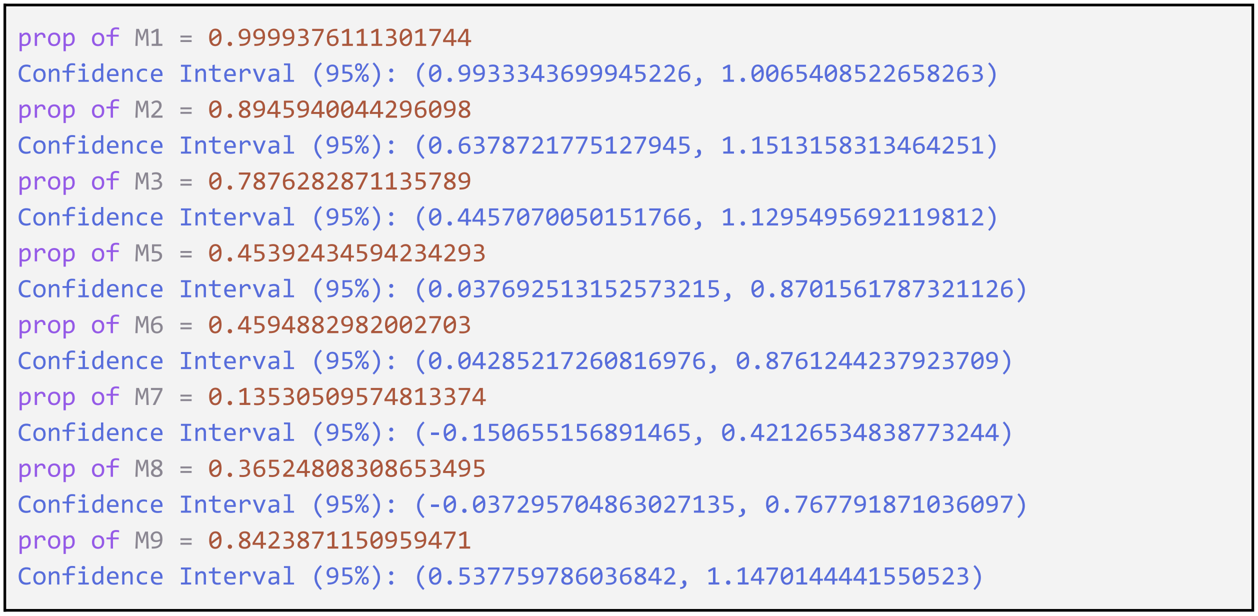


One of the most significant variables in the dataset is the amount spent on each transaction. This will have significant leverage in determining whether or not a data point is fraud or not based on previous trends. The graphs below are a scaled distribution of the transaction amount between 0 and $200 since the majority of the data was contained within that range. In this column, the data is very skewed to the right as there are many outliers between $200 and $10,000. Additionally, since we are working with an international data set the amounts given were in USD however a lot of the amounts were converted from a foreign currency resulting in fractional amounts giving the data significantly more variation than a typical predictor variable.





The columns M1-M9 are a variety of variables that check if center elements of the transaction information match the corresponding cardholder's information. For example, they may check if the given billing address is the same as the one connected to the card, or that the cardholder's name matches the corresponding name on file. The only feature in this set that we didn’t construct the confidence interval for was M4 as it used to flag the M1, M2, and M3, columns for lacking a match and likely was used to help construct the data set, and if it was left in, this would cause collinearity with the other features in this subset.



```
prop of M1 = 0.9999376111301744
Confidence Interval (95%): (0.9933343699945226, 1.0065408522658263)
prop of M2 = 0.8945940044296098
Confidence Interval (95%): (0.6378721775127945, 1.1513158313464251)
prop of M3 = 0.7876282871135789
Confidence Interval (95%): (0.4457070050151766, 1.1295495692119812)
prop of M5 = 0.45392434594234293
Confidence Interval (95%): (0.037692513152573215, 0.8701561787321126)
prop of M6 = 0.4594882982002703
Confidence Interval (95%): (0.04285217260816976, 0.8761244237923709)
prop of M7 = 0.13530509574813374
Confidence Interval (95%): (-0.150655156891465, 0.42126534838773244)
prop of M8 = 0.36524808308653495
Confidence Interval (95%): (-0.037295704863027135, 0.767791871036097)
prop of M9 = 0.8423871150959471
Confidence Interval (95%): (0.537759786036842, 1.1470144441550523)
```