Data

I report the result of the first required question, namely ***Load***, in this section. I will first describe the structure of the data, and then provide some additional basic summary statistics for each field.

Structure of the data: There were 786,363 records. For each record, there were 29 fields/variables. Among the 29 variables, four were numerical. They were *"creditLimit", "availableMoney", "transactionAmount", and "currentBalance"*. The rest 25 variables were categorical.

Descriptive statistics: The mean, standard deviation (SD), median, minimum and maximum for the four numerical variables can be found in Table 1. There were no missing (Null) values in these four variables. The distribution of these four variables can be found in figure 1.

Table 1.
Descriptive Statistics for Numerical Variables

Var	Mean	SD	Median	Minimum	Maximum	Count of Null
creditLimit	10,759.460	11,636.170	7,500	250	50,000	0
availableMoney	6,250.725	8,880.784	3,184.860	-1,005.630	50,000	0
transactionAmount	136.986	147.726	87.900	0	2,011.540	0
currentBalance	4,508.739	6,457.442	2,451.760	0	47,498.810	0

The count of categories (unique values), and missing (Null) values for categorical variables can be found in table 2. Six variables *("echoBuffer", "merchantCity", "merchantState", "merchantZip", "posOnPremises", "recurringAuthInd")* were completely missing. Therefore, I excluded them from all analyses presented in this report.

For the categorical variables which had more than 20 categories. They were identity information, such as, *"accountNumber", "customerId", "merchantName", "cardLast4Digits"* and *"cardCVV"*. For the categorical variables which had less than 20 categories, the corresponding frequency table and histogram were provided (See Table 3 to Table 11 and Figure 3 To Figure 11).

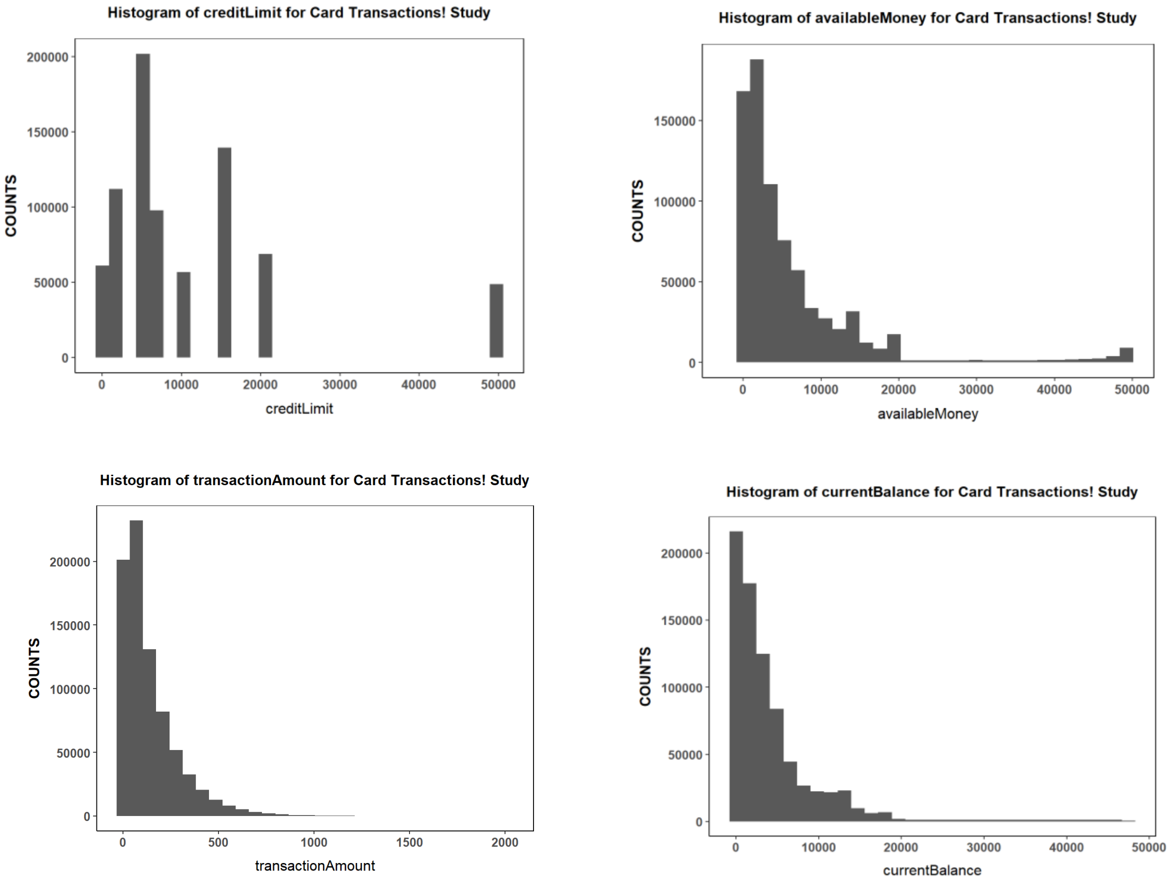

Figure 1. Histogram of numerical variables

Table 2.
Descriptive Statistics for Categorical Variables

Variable	Count of Unique Values	Count of Null
accountNumber	5,000	0
customerId	5,000	0
transactionDateTime	776,637	0
merchantName	2,490	0
acqCountry	5	4,562
merchantCountryCode	5	724
posEntryMode	6	4,054
posConditionCode	4	409
merchantCategoryCode	19	0
currentExpDate	165	0
accountOpenDate	1,820	0
dateOfLastAddressChange	2,184	0
cardCVV	899	0
enteredCVV	976	0
cardLast4Digits	5,246	0
transactionType	4	698
echoBuffer	1	786,363

merchantCity	1	786,363
merchantState	1	786,363
merchantZip	1	786,363
posOnPremises	1	786,363
recurringAuthInd	1	786,363

Table 3.
Frequency Table for acqCountry

	Count	Percentage
Null	4,562	1
CAN	2,424	0
MEX	3,130	0
PR	1,538	0
US	774,709	99

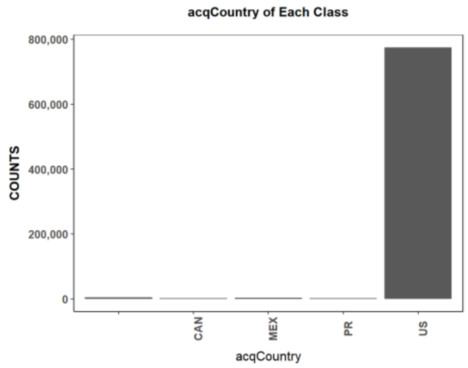

Figure 3. Histogram of acqCountry

Table 4.
Frequency Table for merchant Country Code

	Count	Percentage
Null	724	0
CAN	2,426	0
MEX	3,143	0
PR	1,559	0
US	778,511	99

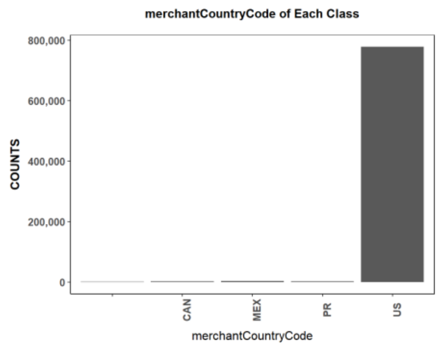

Figure 4. Histogram of merchant Country Code

Table 5.
Frequency Table for posEntryMode

	Count	Percentage
Null	4,054	1
02	195,934	25
05	315,035	40
09	236,481	30
80	15,283	2
90	19,576	2

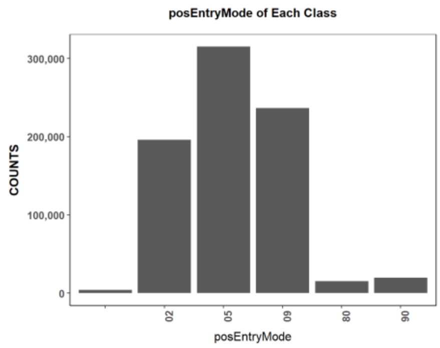

Figure 5. Histogram of posEntryMode

Table 6.
Frequency Table for posConditionCode

	Count	Percentage
Null	409	0
02	628,787	80
08	149,634	19
99	7,533	1

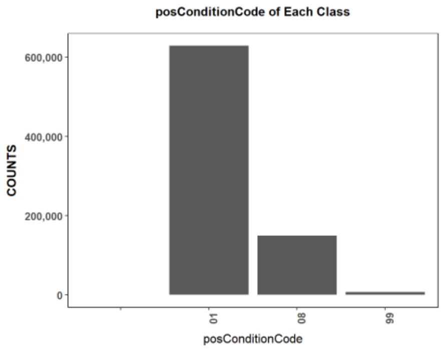

Figure 6. Histogram of posConditionCode

Table 7.
Frequency Table for merchantCategoryCode

	Count	Percentage
airline	15,412	2
auto	21,651	3
cable/phone	1,382	0
entertainment	80,098	10
fastfood	112,138	14
food	75,490	10
food_delivery	6,000	1
fuel	23,910	3
furniture	7,432	1
gym	2,209	0
health	19,092	2
hotels	34,097	4
mobileapps	14,990	2
online_gifts	66,238	8
online_retail	202,156	26
online_subscriptions	11,067	1
personal care	18,964	2
rideshare	51,136	7
subscriptions	22,901	3

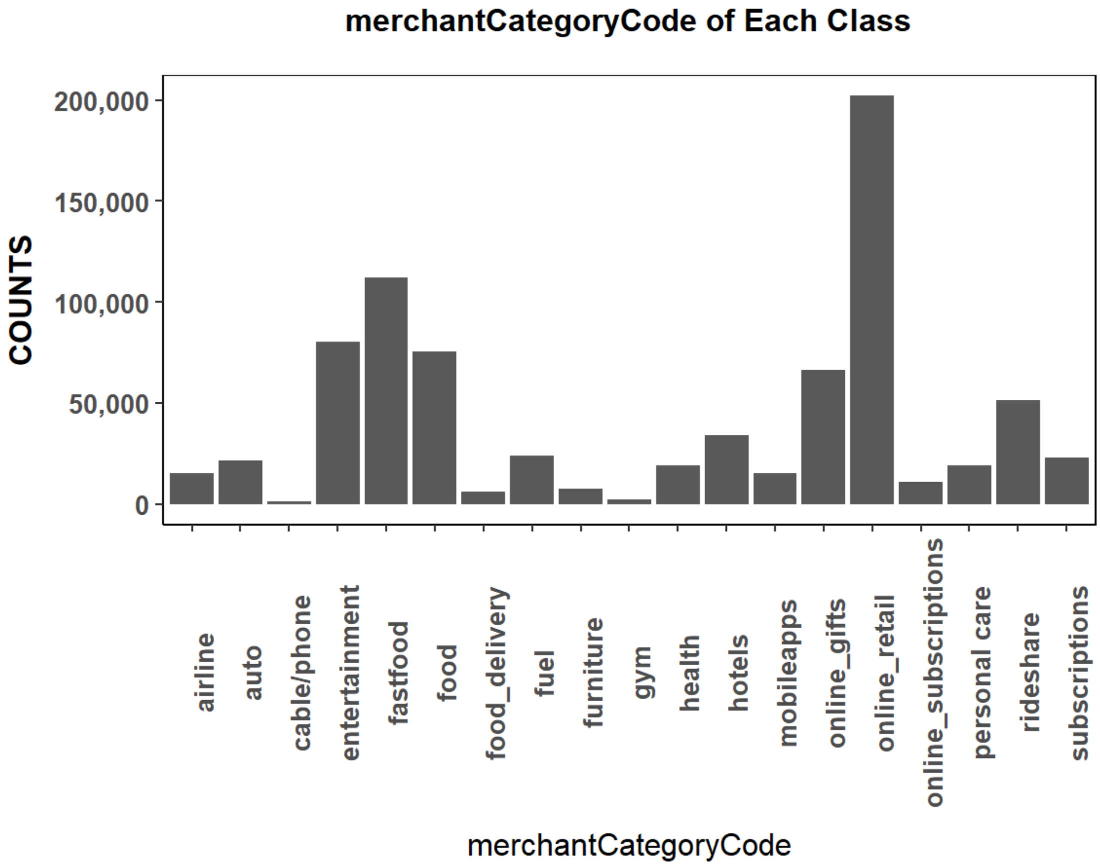

Figure 7. Histogram of merchantCategoryCode

Table 8.
Frequency Table for transactionType

	Count	Percentage
Null	4,562	1
ADDRESS_VERIFICATION	2,424	0
PURCHASE	3,130	0
REVERSAL	774,709	99

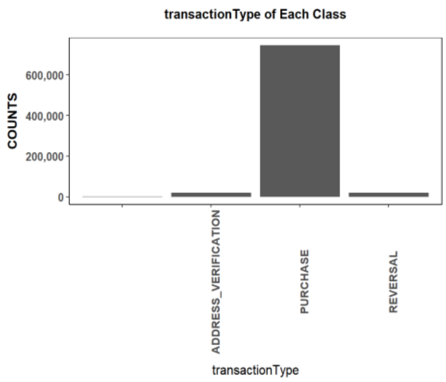

Figure 8. Histogram of transactionType

Table 9.
Frequency Table for cardPresent

	Count	Percentage
FALSE	433,495	55
TRUE	352,868	45

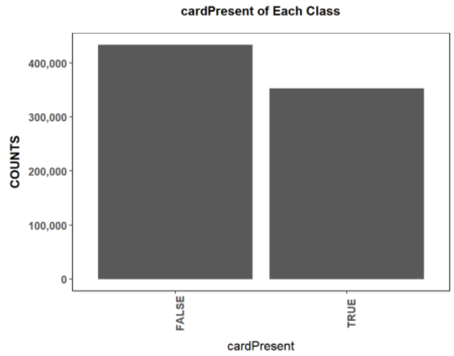

Figure 9. Histogram of cardPresent

Table 10.
Frequency Table for expirationDateKeyInMatch

	Count	Percentage
FALSE	785,320	55
TRUE	1,043	45

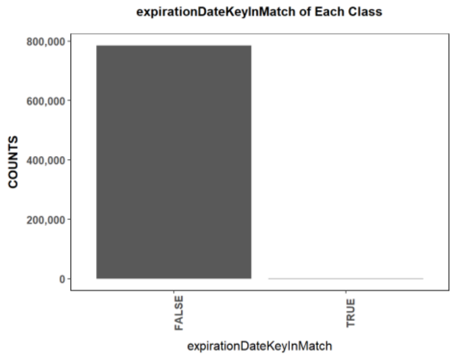

Figure 10. Histogram of expirationDateKeyInMatch

Table 11.
Frequency Table for isFraud

	Count	Percentage
FALSE	773,946	98
TRUE	12,417	2

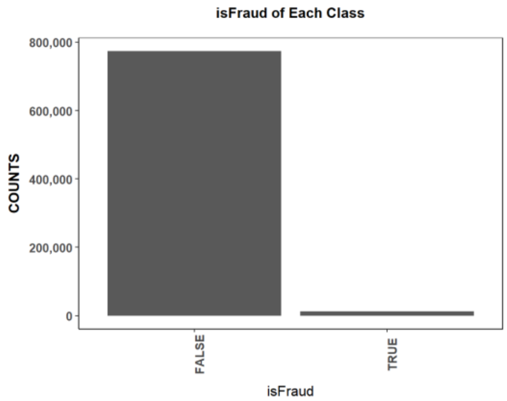

Figure 11. Histogram of isFraud

There were four special categorical variables, *"transactionDateTime"*, *"accountOpenDate"*, *"currentExpDate"*, and *"dateOfLastAddressChange"*. They were based on timestamps. I grouped the data by month, since *"transactionDateTime"* only contains the transaction time in 2016. Please find the frequency table and pie chart in Table 12 and Figure 12.

Table 12.
Frequency Table for transactionDateTime

Month	Count	Percentage
01	61,572	8
02	59,042	8
03	63,927	8
04	62,633	8
05	65,689	8
06	64,735	8
07	67,159	9
08	68,129	9
09	66,777	8
10	69,627	9
11	68,097	9
12	68,976	9

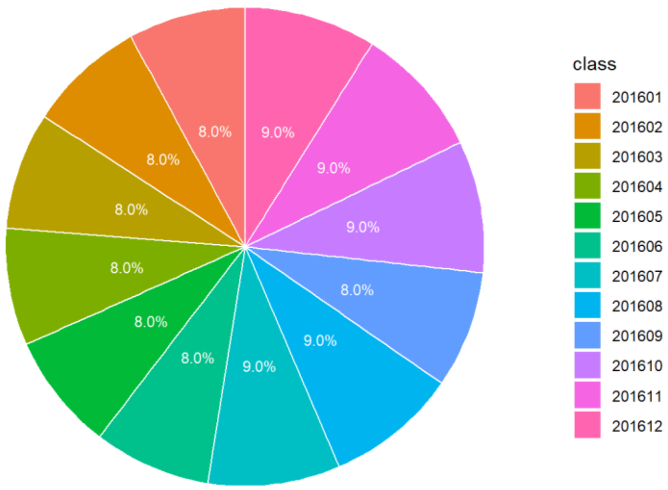

Figure 12. Pie chart of transactionDateTime

I grouped *"accountOpenDate", "currentExpDate",* and *"dateOfLastAddressChange"* by year, since contained data across years. Please find the frequency table (Table 13 to Table 15) and pie chart (Figure 13 to Figure 15).

Table 13.
Frequency Table for dateOfLastAddressChange

Year	Count	Percentage
1989	174	0
1997	5	0
1999	61	0
2001	24	0
2003	305	0
2004	654	0
2005	1,325	0
2006	625	0
2007	4,113	1
2008	6,505	1
2009	5,684	1

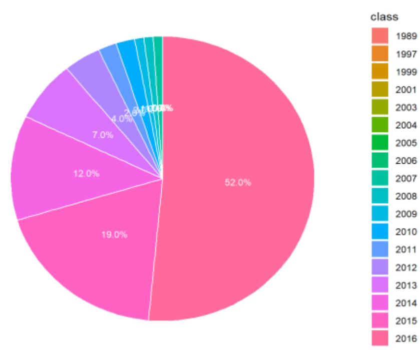

Figure 13. Pie chart of dateOfLastAddressChange

2010	14,398	2
2011	13,962	2
2012	31,670	4
2013	52,619	7
2014	90,459	12
2015	151,096	19
2016	412,684	52

Table 14.
Frequency Table for accountOpenDate

Year	Count	Percentage
1989	174	0
1997	5	0
1999	61	0
2001	24	0
2003	1,324	0
2004	883	0
2005	1,578	0
2006	1,179	0
2007	6,400	1
2008	9,017	1
2009	10,492	1
2010	35,450	5
2011	25,508	3
2012	64,997	8
2013	91,205	12
2014	249,441	32
2015	288,625	37

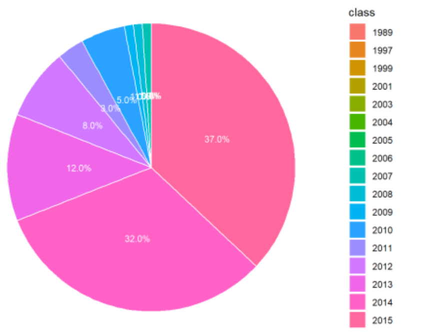

Figure 14. Pie chart of accountOpenDate

Table 15.
Frequency Table for currentExpDate

Year	Count	Percentage
2019	4,028	1
2020	57,454	7
2021	57,490	7
2022	57,073	7
2023	57,778	7
2024	56,928	7
2025	57,419	7
2026	57,402	7
2027	57,983	7
2028	57,679	7
2029	57,451	7
2030	57,264	7
2031	57,445	7
2032	57,664	7
2033	35,305	4

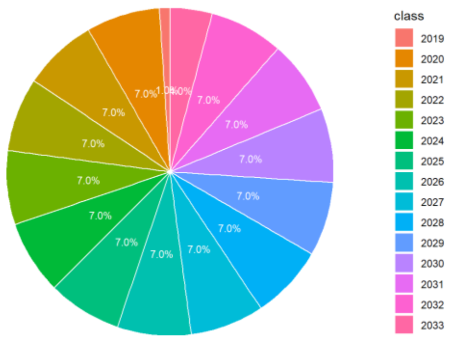

Figure 15. Pie chart of currentExpDate

Transaction Amount

I report the result of the second required question, namely ***Plot***, in this section. I will first describe the histogram of the processed amounts of each transaction, and then discuss the hypotheses I had.

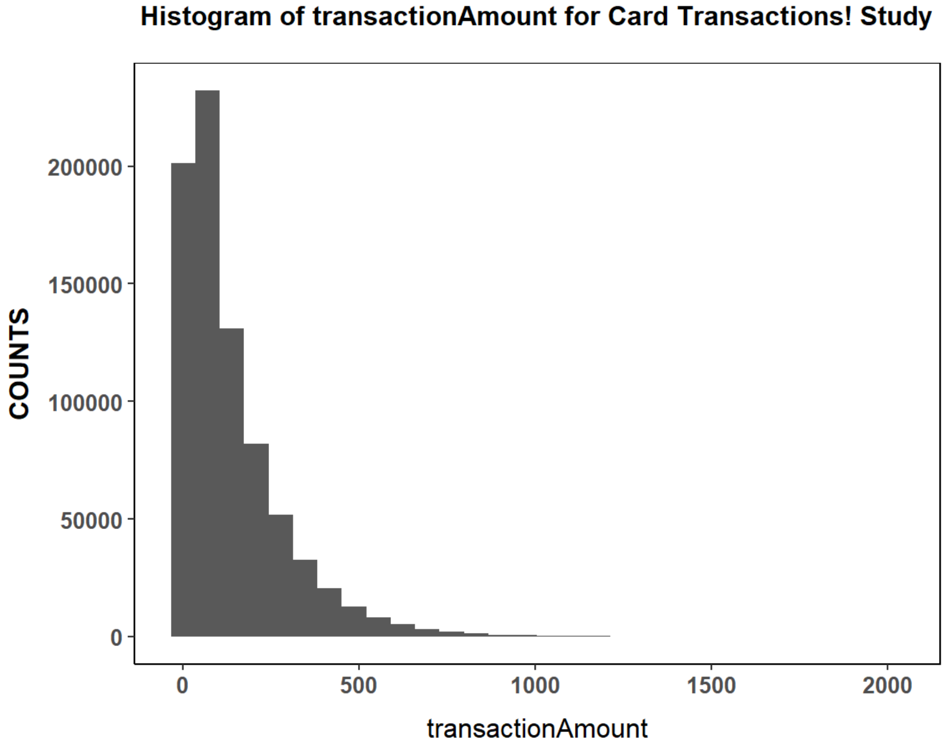

Figure 16. Histogram of transactionAmount

It can be seen from Table 1, there was no missing value of *"transactionAmount"*. The mean is 136.986 (SD = 147.726). The histogram of *"transactionAmount"* in Figure 16 showed that it was a right-skewed distribution. The majority of transactions were less than $1,000. By checking the skewness (2.09) and kurtosis (9.45), I found this distribution was leptokurtic and slightly right-skewed. For univariate distribution, the values for skewness and kurtosis between -2 and +2 are considered acceptable to prove normal.

The first hypothesis was: *"transactionAmount"* could approximate standard normal distribution after conducting logarithmic transformation. I compute "*log_transactionAmount"*, which is the natural logarithmic transformation of *"transactionAmount"*. The skewness (-1) and kurtosis (3.95) did not fully support my first hypothesis. The distribution was slightly left-skewed and less leptokurtic after logarithmic transformation. Please find the histogram of "*log_transactionAmount*" in Figure 17.

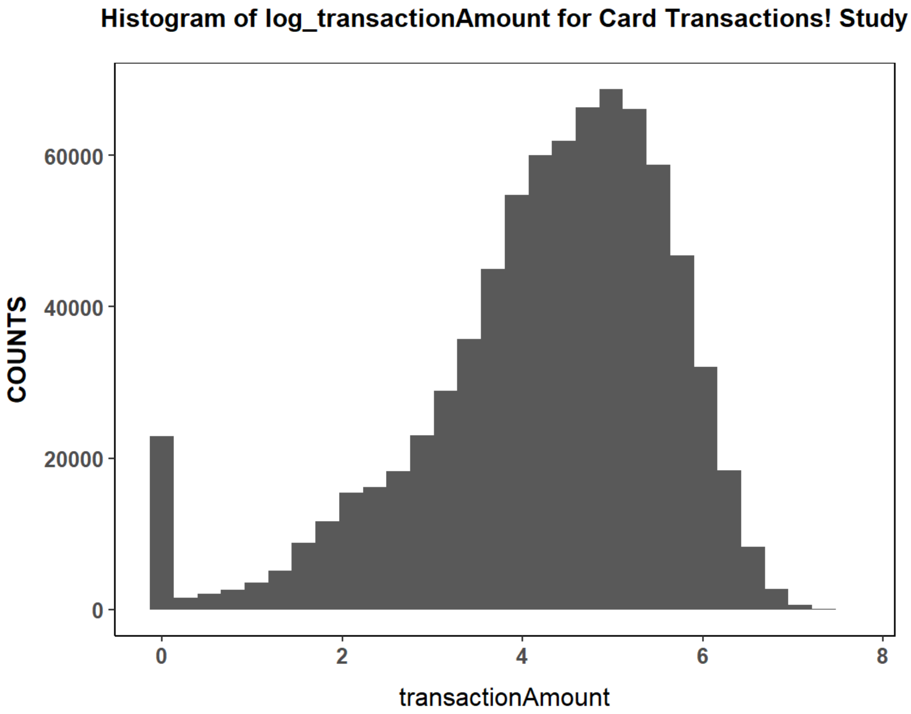

Figure 17. Histogram of log_transactionAmount

The second hypothesis I had was that the distribution of transaction amount was different based on the type of transaction. I assume by removing the transaction type of "ADDRESS_VERIFICATION", the distribution of *"transactionAmount"* will change. Table 16 indicated my second hypothesis hold. All "ADDRESS_VERIFICATION" transactions had the amount of zero. After removing the recode which the transaction type was "ADDRESS_VERIFICATION", both skewness and kurtosis decreased a bit. The skewness became 2.08 and the kurtosis became 9.41. It is still leptokurtic and slightly right-skewed.

Reversed Transaction and Multi-swipe Transactions

I report the result of the third required question, namely ***Data Wrangling - Duplicate Transactions,*** in this section. I will first describe how I detect duplicated transactions, and then present the Comparison of reversed transactions and the multi-swipe transactions in terms of the total number of transactions and total dollar amount. I will also discuss the investigation of comparing the reversed transactions and the multi-swipe transactions based on *"avaiableMoney"* and "*isFraud".*

Based on my understanding, I grouped the variables into three subgroups. Specifically, account holder based (2 variables), card-based (6 variables), and transaction-based (15 variables) (See Table 17). These variables will help me to define the duplicated transactions. As far as I see, the account holder based and card-based variables should be the same for duplicated transactions. Among the transaction-based variables, *"availableMoney", "transactionDateTime", "transactionType",* and *"currentBalance"* could be different for the reversed transaction or multi-swipe transactions. The rest 11 transaction-based variables should be the same for the duplicated transaction, especially, *"transactionAmount", "merchantName"*, and *"merchantCategoryCode"*. A new variable "*transactionDateTime_ymd*" was derived from *"transactionDateTime"* which corresponding to the year-month-date of the *"transactionDateTime"* (without hour-minute-second). The purpose of creating this new variable was that I assume the duplicated transaction should happen on the same day.

Table 17
Subgroups of the variables

Groups	Variables
account holder based	accountNumber, customerId
card-based	creditLimit, currentExpDate, accountOpenDate, dateOfLastAddressChange, cardCVV, cardLast4Digits
transaction-based	availableMoney, transactionDateTime, transactionAmount, merchantName, acqCountry, merchantCountryCode, posEntryMode, posConditionCode, merchantCategoryCode, enteredCVV, transactionType, currentBalance, cardPresent, expirationDateKeyInMatch, isFraud

Then, I captured the duplicated transactions by forcing the account holder based, card-based, 11 transaction-based, and “*transactionDateTime_ymd*” to be the same. After removing the first transaction (considered to be "normal"), I grouped the abnormal/duplicated transactions based on the transaction type. If the associated transaction type was “REVERSAL”, I treated it as reversed transaction. Controversially, I treated a transaction as multi-swipe in the dataset of duplicated transactions, if the associated transaction type was “PURCHASE”. Please find the total number of transactions and total dollar amount for the reversed transactions and the multi-swipe transactions in Table 18.

Table 18
Comparison of reversed transactions and the multi-swipe transactions

Transaction Type	Total Number of Transactions	Total Dollar Amount
Reversed	5,519	825,455.58
Multi-swipe	7,396	1,093,981.23

Regarding the interesting facts about the two types of duplicated transactions. I first assumed that the *“availableMoney”* will increase after reversed transactions but decrease after multi-swipe transactions Likewise, I assumed reversed transactions were less likely to be labeled as a fraud than multi-swipe transactions. However, the results did not support my assumptions. 6.4% (n = 476) of the multi-swipe transactions associated with increased *“availableMoney”*, while only 5.8% (n = 319) of the reversed transactions had increased *“availableMoney”*. 1.7% (n = 125) of the multi-swipe transactions were labeled as a fraud, while 1.8% (n = 102) of the reversed transactions were a fraud.

Predict Fraud

I report the result of the fourth required question, namely ***Model***, in this section. I will first describe the model I used, and then present the evaluation result. I will discuss the questions I had, and the thoughts for improvement at the end of this section.

First of all, I created a benchmark set for testing purposes. This test set was not used in the process of training parameter meters, so it is reasonable to test the performance of a trained model with it. 50% of the records were randomly selected as training data, and the rest 50% were used as the test set.

From table 11 and Figure 11, I noticed that the fraudulent transactions were very few among all transactions. There were about 2% of the transactions were a fraud. Such imbalanced records were a challenge for prediction. Therefore, I computed the weights based on the proportion of the fraud transactions. Essentially, I specified weights as the inverse of label proportion. For the non-fraud class, I used a weight of 2. And for the fraud class, will use a weight of 98. After weighting the records, the penalty of incorrect predicting fraud class would be 98 times more severe than incorrect predicting non-fraud class.

Within the training set, I selected **weighted logistic regression** as the training model. The reason for applying logistic regression was as following. Firstly, the label was binary for which logistic regression was an appropriate model. Secondly, logistic regression had fewer parameters to train other machine learning models, such as, Convolutional Neural Network (CNN). It allowed me to finish the data challenge within the time limit (4-6 hours for the whole project). Thirdly, the potential predictors were not in a high dimension. Logistic regression could perform as well as other advanced models with a small set of predictors. Lastly, the predictors were meaningful, and the logistic regression allowed researchers to interpret the predictors and labels straightforwardly.

Regarding select the appropriate predictors, I referred to the subgroups I created in Table 17. First of all, the 6 variables which were completely missing were excluded from the analysis. Secondly, the account holder based variables were excluded. They were categorical IDs, which did not commonly use as predictors. Likewise, the *"cardCVV"* and *"cardLast4Digits"* were also categorical IDs. I excluded *"accountOpenDate", "dateOfLastAddressChange"* as well, since they were categorical variables with many categories. It was meaningless to create dummy code (one-hot encoding) for these variables.

Only *"creditLimit"* was included as predictors among the card-based variables. I believe the transaction-based variables were the key predictors to predict fraud transactions, therefore, I included most of them. Only three variables were excluded. Specifically, the reason for removing *"transactionDateTime"*, *"merchantName"* and *"enteredCVV"* were the same as before. They were categorical variables with too many categories.

Fit the trained weighted logistic regression model with the test set, the predicted values were the predicted probabilities for each record to be a fraud. The cutoff of the probability was set as 0.5. If the associated probability was higher than 0.5, the record was labeled as fraud. Likewise, if the associated probability was less than 0.5, the record was labeled as non-fraud.

The trained weighted logistic regression achieved a model accuracy of 74.7% on the test set. Please see the confusion matrix in Table 19. The true positive rate was 61.2%, and the true negative rate was 74.9%. Please find the ROC curve in Figure 18.

Table 18.
Confusion matrix on test set

		Actual		
Prediction			TRUE	FALSE
		TRUE	3797	97109
		FALSE	2405	289871

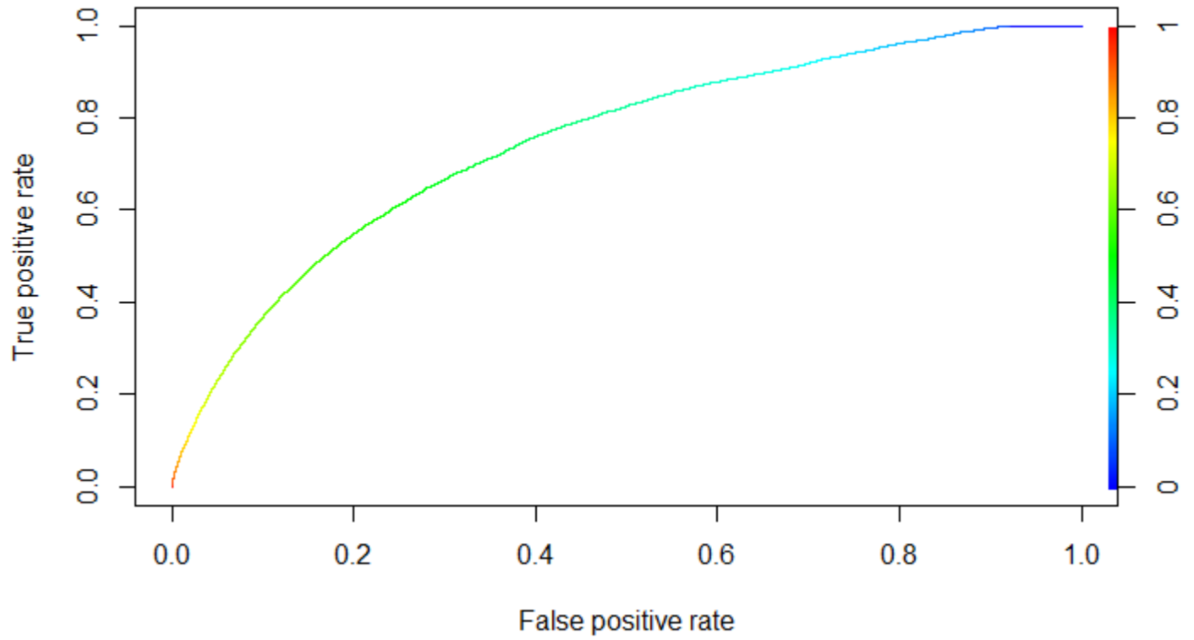

Figure 18. ROC curve on test set

The first problem I had for this prediction task was balancing the true positive rate and true negative rate. I am not sure to what extent, the false positive can be accepted. As far as I see, fraud transactions should not be missed. However, in this study, the high true positive rate was associated with a low true negative rate. For example, if I changed the weights for fraud transactions from 98 (a rough proportion) to 98.5 (more accurate proportion of non-fraud transition), and also changed the cutoff value of predicted probability from 0.5 to 0.4. The true positive rate increased from 61.2% to 84.5% (See Confusion matrix in Table 19). However, the true negative rate decreased from 74.9% to 53.3%. When the true negative rate decreases, the false positive rate increases. More human resources and capital resources may need for further investigation and classification.

Table 19.
Confusion matrix with changing weights and cutoff value of predicted probability

		Actual		
Prediction			TRUE	FALSE
		TRUE	5244	180380
		FALSE	958	206600

The second problem I had was related to the time frame. Due to the limited time, I did not implement cross-validation. The sample size was large enough for conducting cross-validation. If I have more time next time, I will conduct cross-validation to improve the model. Moreover, I could also derive more variables as predictors. I did not include *"cardCVV"* and *"enteredCVV"* in my current model. I wish to derive an additional predictor next time to represent the difference/inconsitance between *"cardCVV"* and *"enteredCVV"*. Likewise, I wish to explore the potential use of *"transactionDateTime"* by extracting the year, month, hour, and so on as predictors.