Stori Data Science Challenge
performed by Jorge Luis Torres Moreno

Data preparation process:

The csv file had some missing values, they were filled as follows:

- Activated_date: Since the file is in ascending order with respect to this date, last valid date was assigned to the first null value found.
- Last_payment_date: The activated_date of the same row was used to fill this space only if a payment or a cash_advanced was done. There is one value which was left as null since neither a payment nor a cash_advanced was found.
- Balance, cash_advance, prc_full_payment, credit_limit and minimum_payments: Because of their values are very sparse, their missing values were filled with the corresponding medians. In the case of the cash_advance column, the median was assigned only in the case where the column cash_advance_frequency was greater than cero, otherwise 0.0 was assigned.
  My first insight with respect to these columns was to compute them, I am sure they can be calculated from the available information, however since the dataset only describes a period of six months, it is very likely that some information from the previous months is necessary for a proper computation, because of that I decided to use the medians.

No outliers were removed from the dataset. Although the values are so dispersed, it does not seem like there are clearly impossible or wrong values. I also had in mind the type of analysis for this decision, it was not a detailed analysis, if that had been the case, I would have removed outliers, or, at least, I would have segmented the data in a corresponding number of percentiles.

No duplicated rows were found after the preprocessing of the data. The process that has been described is performed by the code *preprocess.py*. The cleaned dataset was saved in another csv file called: *new_credit_card.csv*. this csv was used in the analysis.

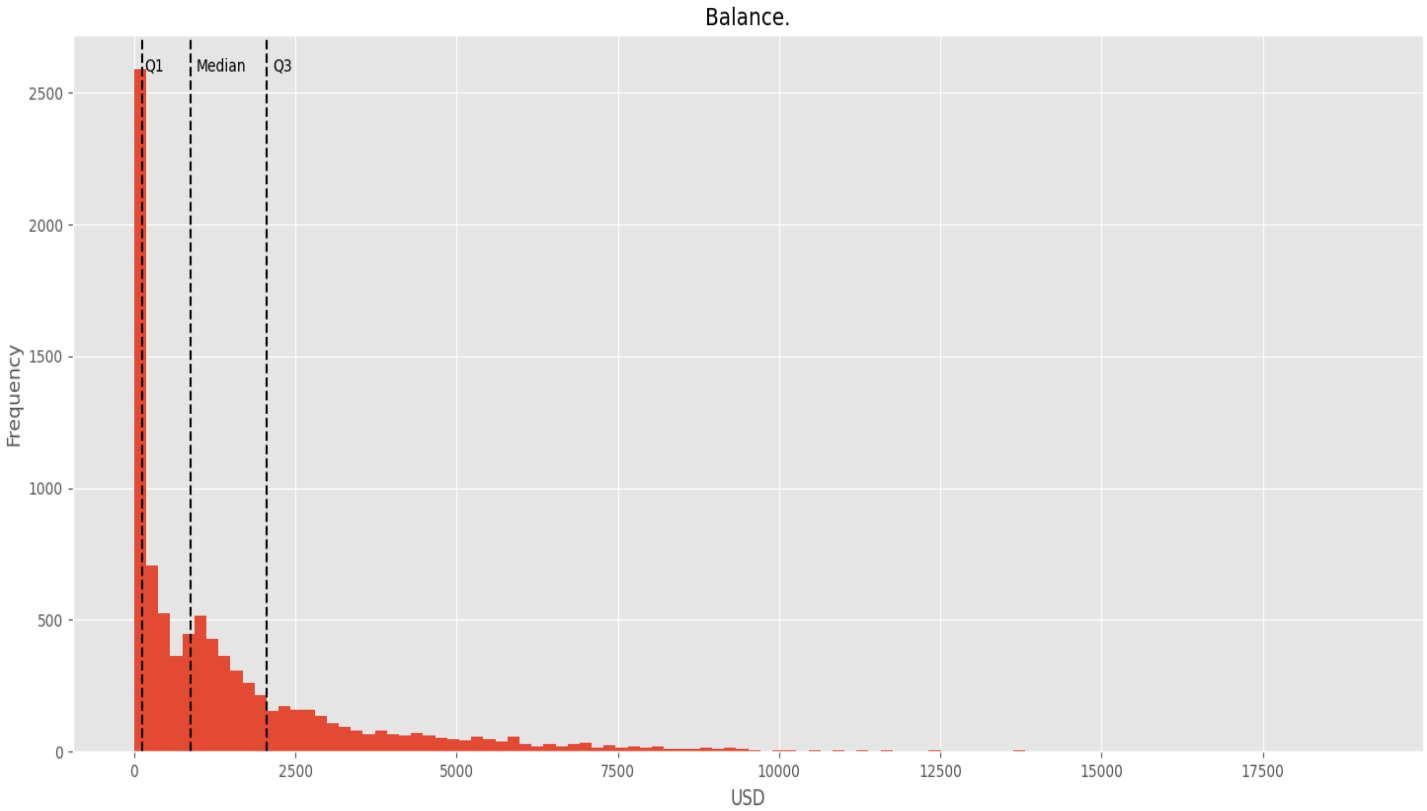
Question 1:

The histogram shows how dispersed is the data, it is remarkable the fact that just the first bin contains more than the 25% of all the data. I think we could say the curve is the half part of a highly leptokurtic gaussian curve, because of this dataset is not normally distributed I decided to use the Freedman-Diaconis rule to define the width of the bins. I assumed the figures were dollars.

Balance is described as the available amount of money to make purchases. This means that a considerable part of the credit card holders is in a compromised situation with their credit cards, however this could also mean that some of them are increasing their expenses because their income is also going up, therefore, an increment in their credit limit would be necessary.

On the other hand, there are some credit card holders who are not using their credit line, a special offer such as installments purchases, or cash disposal may stimulate the use of their cards.

In the following plot:
Q1 =128.2819155
Median = 873.385231
Q3 = 2054.1400355



The plot and the data showed before can be got using the program *histogram.py*.

The table containing the mean balance and the median balance grouped by year and month of activated_date is presented below:

Activated_date	Activated_date	Mean_balance	Median_balance
2019	10	2482.234166	1524.409377
	11	1846.847082	1081.065726
	12	2015.810109	1166.500558
2020	01	1852.434079	1173.662921
	02	1744.756667	994.841733
	03	1553.261113	828.954823
	04	1487.733269	910.141912
	05	1214.333732	734.557681
	06	939.789358	475.265493
	07	649.151531	221.203759

The mean_balance and the median_balance decrease as time advance. An explanation to this effect could be the following:
Credit cards approved near the holidays have a credit limit higher than those cards approved after the holidays because the expenses after those days used to be too smaller than those done during the end of the year. It is also very likely that less people is interested in a credit during those months because they do not have in mind several expenses.

The code used to get the last table is called: *statsbalance.py*.

Question 2.

The resultant table from this question is too large to be included in this document, therefore it was saved as a csv file named: *table_credit_card.csv.* To get rid from the letters in the cust_id column, regular expressions were used. Pandas' tools were used to fit dates in the requested format. The code which deals with this question is called: *table.py*.

Question 3.

Since fraud just have two possible outcomes (0 or 1) I decided to perform a logistic regression using python's Machine Learning library sklearn. This task was a bit complicated because the dataset is far from be balanced: there are just 70 fraudulent transactions whereas there are 8880 not fraudulent transactions. Because of this, it was necessary to build a sample which had the same amount of both kind of transactions. The 80% of that sample was used to train the model and the other 20% was used to test the model. After a careful inspection of the descriptive parameters of each of the columns I decided to build the model using the following columns:

- Purchases.
- Oneoff_purchases.
- Balance.
- Payments.
- Credit_limit.

This model reported an accuracy of around 90% with training data and more than 95% with test data. Although the combination of all those columns is important to achieve such accuracy, the column purchases showed a high influence in the model, using just that column the model yields an accuracy of around 85%. The reason to use these columns is because they show notably differences between fraudulent and honest transactions. Those columns which expresses values from 0 to 1 showed great differences as well, however, since those changes are no larger than 1, it is difficult to find a clear relationship between them and the fraudulent transaction. All this process was done by the program *model.py*.