Describe the dataset and any issues with it.

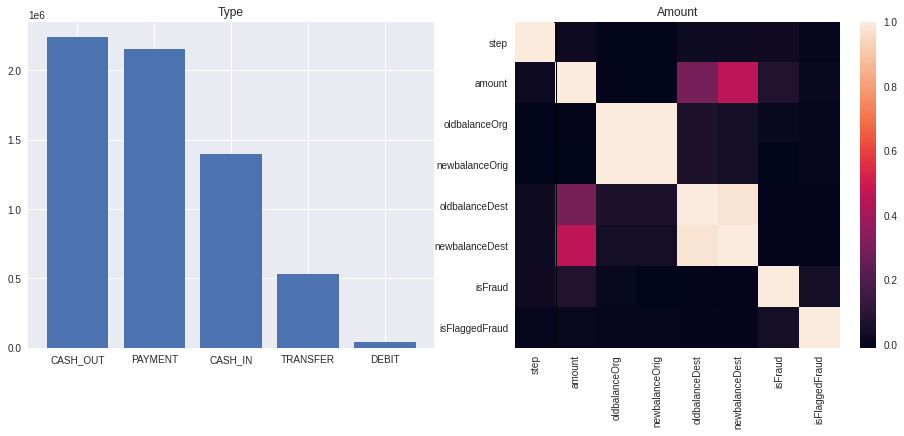
 Generate a minimum of 5 visualizations using the data and write a brief description of your

observations. Additionally, all attempts should be made to make the visualizations visually

appealing

**Data Set Issues:**

I think one of the key issues with this data set is how its imbalanced. Because were dealing with fraud data like this, over 99.8% of the transaction records are non-fraudulent. Because only a tiny fraction of the dataset represents fraud, fraudulent transactions are likely to be under-represented in our model. If unaddressed, data imbalance can cause issues, such as misleading accuracy metrics in the model. The model may attempt to label every transaction as non-fraudulent, and since any randomly given transaction is highly likely to be non-fraudulent, a high accuracy will be reported, despite incorrectly classifying all fraudulent transactions as non-fraudulent.



**Bar Graph Code**

paysim\_data = pd.read\_csv("../input/paysim1/PS\_20174392719\_1491204439457\_log.csv",encoding="ISO-8859-1",error\_bad\_lines=False)

paysim\_data.head(2)

x = paysim\_data['type'].value\_counts().index

y = paysim\_data['type'].value\_counts().values

f = plt.figure(1,figsize=(16,6))

ax1 = f.add\_subplot(1,2,1)

ax1.title.set\_text('Type')

\_ =ax1.bar(x,y)

**Heat Map Code**

corr = paysim\_data.corr()

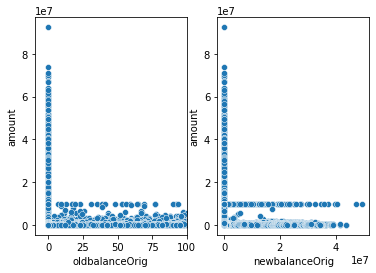
sns.heatmap(corr, xticklabels=corr.columns.values, yticklabels=corr.columns.values)

**Bar Graph Observations(Graph on the Left)**

**HeatMap Observations(Graph on the Right)**

If we observe the heatmap, you will notice that oldbalance\* and newbalance\* are highly correlated with each other. That is true because transaction amounts are typically a small proportion of account balances. In other words, account balances have high correlations with themselves.

What's odd is that while amount has some correlation with oldbalanceDest and newbalanceDest, it has very low correlations with oldbalanceOrig and newbalanceOrig. Let's check pairwise scatterplots for these variables, starting with amount vs. oldbalanceOrig and newbalanceOrig.



**Code:**

fig, ax = plt.subplots(1,2)

ax[0].set\_xlim([-10,100])

sns.scatterplot(data=df, x='oldbalanceOrig', y='amount', ax=ax[0])

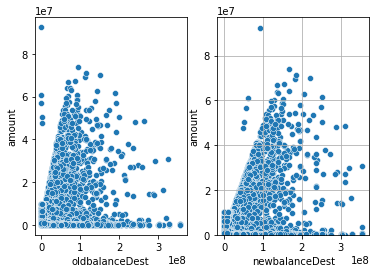
sns.scatterplot(data=df, x='newbalanceOrig', y='amount', ax=ax[1])

fig.show()

**Scatter 1(See Graph Above) Description**

Let's check pairwise scatterplots for these variables, starting with amount vs. oldbalanceOrig and newbalanceOrig.

There is a notably high frequency of transactions with the amount of exactly 10,000,000.



**Code:**

fig, ax = plt.subplots(1,2)

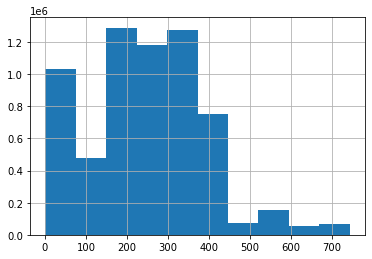
sns.scatterplot(data=df, x='oldbalanceDest', y='amount', ax=ax[0])

sns.scatterplot(data=df, x='newbalanceDest', y='amount', ax=ax[1])

fig.show()

**Scatter 2(Graph Above) Description**

We can see in these pairwise plots between amount and oldbalanceDest / newbalanceDest that they exhibit more evenly random distributions. There is a very weak positive relationship between amounts and destination account balances, which we observed earlier in the correlation matrix.



Code:

df = pd.read\_csv('../input/paysim1/PS\_20174392719\_1491204439457\_log.csv')

df['step'].hist()

**Histogram**

step is our time variable. Most transactions occur in the first half of our time sample range.

Amounts are skewed right—the vast majority of transactions are low amounts.

**Conclusion:**

We have established a basic process for exploratory data analysis and modeling that has resulted in an acceptable predictor for our transaction data. There are other modeling techniques that we could try, such as SVM and Logistic Regression. There are also additional techniques to improve model performance, as well as better ways to measure performance, such as hyper-parameter tuning and cross-fold validation. It would be prudent to use these methods if we were tasked with informing business decisions that could affect large amounts of revenue or costs.