CSE C10 Data Intensive Computing Project
Phase 1

Report done by:

Mali Pashupathi	20211A05F8
Kondakalla Bhoomika Reddy	20211A05D6
Kondaveeti Bhargav	20211A05D7

Motivation

Investors of all stripes have become interested in the cryptocurrency craze during the past few years. Additionally, it has drawn the interest of scammers. Most cryptocurrency scams try to deceive their victim into sending money to a hacked digital wallet. These targets are probably being persuaded by the enormous returns promised by the attackers through the use of specialized social engineering techniques like romance scams, email phishing, and even Ponzi schemes.
We plan to use the method of supervised and unsupervised learnings, in order to construct different types of classifiers to detect fraudulent Ethereum transactions, in order to put our security analyst talents to use in tackling real-world issues.

Dataset Overview

The Ethereum Fraud Detection Dataset, an open-source, labeled dataset from Kaggle, contains over 10,000 samples. The dataset is hugely imbalanced.

Link to data source: Ethereum Fraud Detection Dataset | Kaggle

Columns Review

We do have a total of 51 columns in the dataset, the detailed description of the columns is given below:

Index: the index number of a row

Address: the address of the ethereum account

FLAG: whether the transaction is fraud or not

Avg min between sent tnx: Average time between sent transactions for account in minutes

Avgminbetweenreceivedtnx: Average time between received transactions for account in minutes

TimeDiffbetweenfirstand_last(Mins): Time difference between the first and last transaction

Sent_tnx: Total number of sent normal transactions

Received_tnx: Total number of received normal transactions

NumberofCreated_Contracts: Total Number of created contract transactions

UniqueReceivedFrom_Addresses: Total Unique addresses from which account received transactions

UniqueSentTo_Addresses20: Total Unique addresses from which account sent transactions

MinValueReceived: Minimum value in Ether ever received

MaxValueReceived: Maximum value in Ether ever received

AvgValueReceived5Average value in Ether ever received

MinValSent: Minimum value of Ether ever sent

MaxValSent: Maximum value of Ether ever sent

AvgValSent: Average value of Ether ever sent

MinValueSentToContract: Minimum value of Ether sent to a contract

MaxValueSentToContract: Maximum value of Ether sent to a contract

AvgValueSentToContract: Average value of Ether sent to contracts

TotalTransactions(IncludingTnxtoCreate_Contract): Total number of transactions

TotalEtherSent:Total Ether sent for account address

TotalEtherReceived: Total Ether received for account address

TotalEtherSent_Contracts: Total Ether sent to Contract addresses

TotalEtherBalance: Total Ether Balance following enacted transactions

TotalERC20Tnxs: Total number of ERC20 token transfer transactions

ERC20TotalEther_Received: Total ERC20 token received transactions in Ether

ERC20TotalEther_Sent: Total ERC20token sent transactions in Ether

ERC20TotalEtherSentContract: Total ERC20 token transfer to other contracts in Ether

ERC20UniqSent_Addr: Number of ERC20 token transactions sent to Unique account addresses

ERC20UniqRec_Addr: Number of ERC20 token transactions received from Unique addresses

ERC20UniqRecContractAddr: Number of ERC20token transactions received from Unique contract addresses

ERC20AvgTimeBetweenSent_Tnx: Average time between ERC20 token sent transactions in minutes

ERC20AvgTimeBetweenRec_Tnx: Average time between ERC20 token received transactions in minutes

ERC20AvgTimeBetweenContract_Tnx: Average time ERC20 token between sent token transactions

ERC20MinVal_Rec: Minimum value in Ether received from ERC20 token transactions for account

ERC20MaxVal_Rec: Maximum value in Ether received from ERC20 token transactions for account

ERC20AvgVal_Rec: Average value in Ether received from ERC20 token transactions for account

ERC20MinVal_Sent: Minimum value in Ether sent from ERC20 token transactions for account

ERC20MaxVal_Sent: Maximum value in Ether sent from ERC20 token transactions for account

ERC20AvgVal_Sent: Average value in Ether sent from ERC20 token transactions for account

ERC20UniqSentTokenName: Number of Unique ERC20 tokens transferred

ERC20UniqRecTokenName: Number of Unique ERC20 tokens received

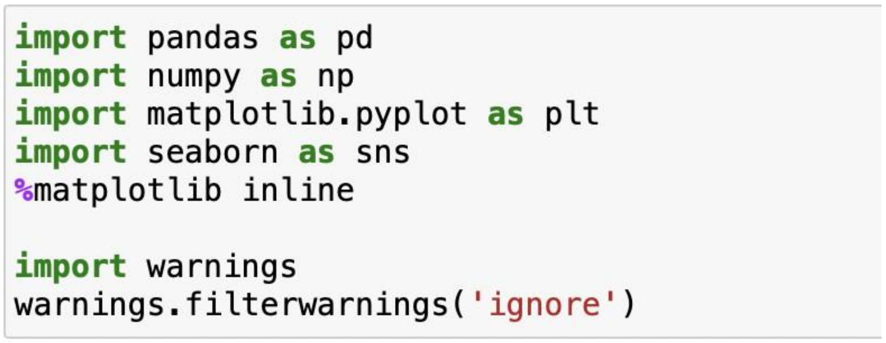
ERC20MostSentTokenType: Most sent token for account via ERC20 transaction

ERC20MostRecTokenType: Most received token for account via ERC20 transactions

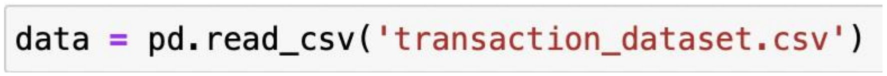
Questions

1. Finding the transactions which cause high disturbance and removing them.
2. Finding the duration of each transaction to find when exactly fraud happens!

Importing required libraries and dataset

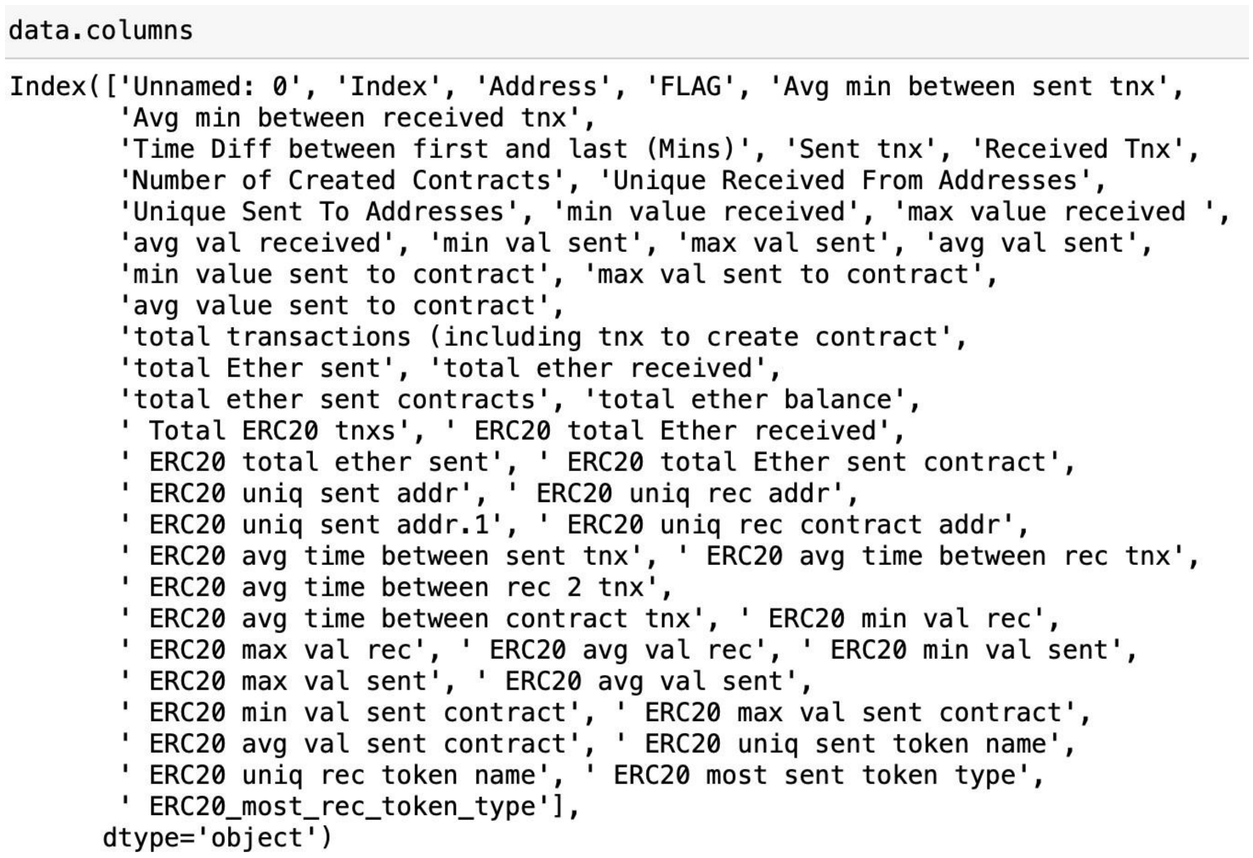


```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```



```python
data = pd.read_csv('transaction_dataset.csv')
```

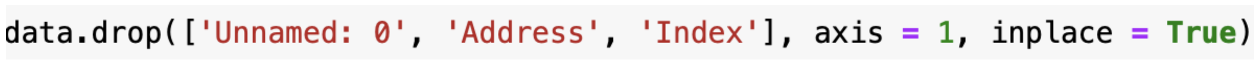
Columns of the dataset



```python
data.columns
```

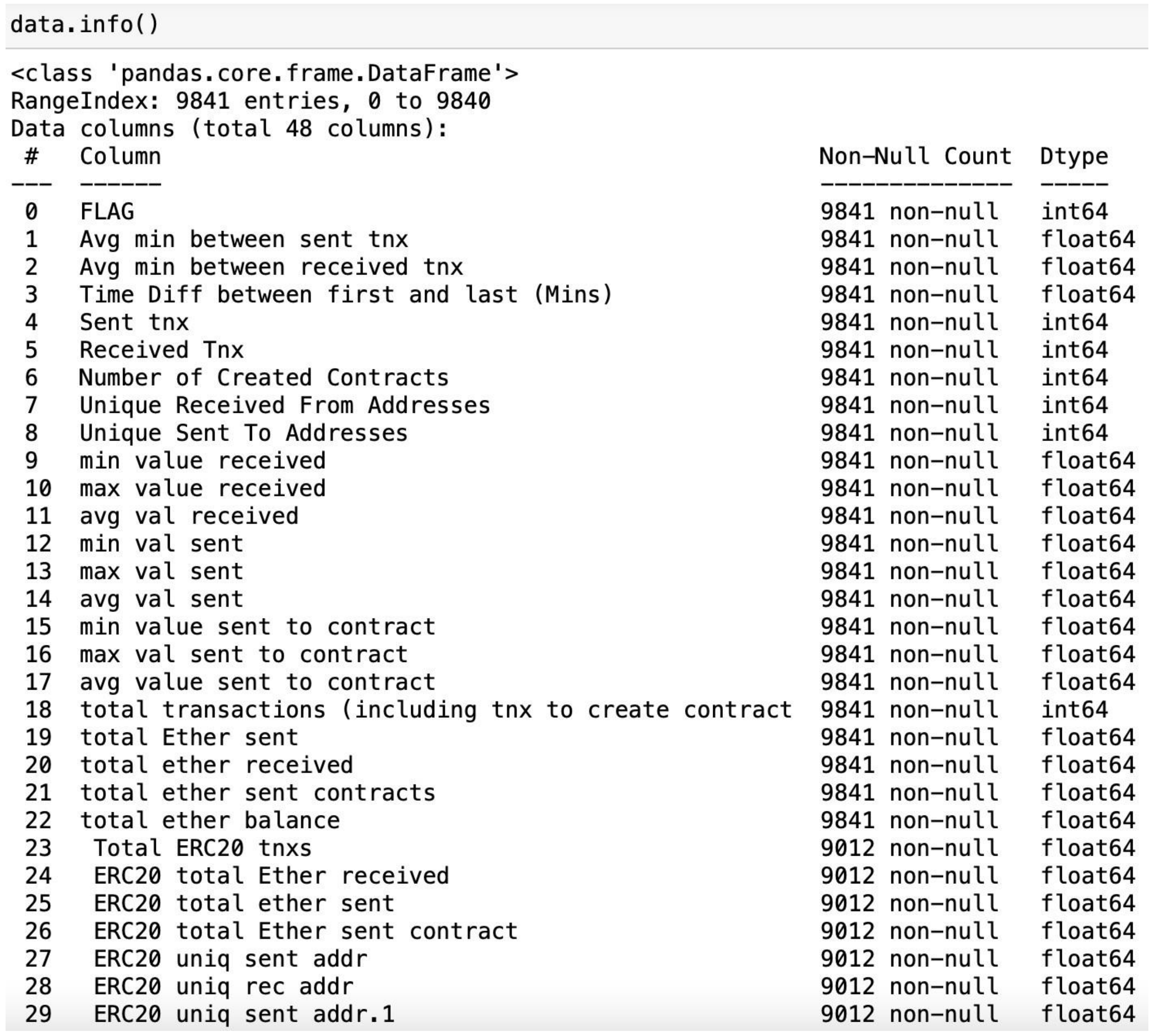
```
Index(['Unnamed: 0', 'Index', 'Address', 'FLAG', 'Avg min between sent tnx',
       'Avg min between received tnx',
       'Time Diff between first and last (Mins)', 'Sent tnx', 'Received Tnx',
       'Number of Created Contracts', 'Unique Received From Addresses',
       'Unique Sent To Addresses', 'min value received', 'max value received ',
       'avg val received', 'min val sent', 'max val sent', 'avg val sent',
       'min value sent to contract', 'max val sent to contract',
       'avg value sent to contract',
       'total transactions (including tnx to create contract',
       'total Ether sent', 'total ether received',
       'total ether sent contracts', 'total ether balance',
       ' Total ERC20 tnxs', ' ERC20 total Ether received',
       ' ERC20 total ether sent', ' ERC20 total Ether sent contract',
       ' ERC20 uniq sent addr', ' ERC20 uniq rec addr',
       ' ERC20 uniq sent addr.1', ' ERC20 uniq rec contract addr',
       ' ERC20 avg time between sent tnx', ' ERC20 avg time between rec tnx',
       ' ERC20 avg time between rec 2 tnx',
       ' ERC20 avg time between contract tnx', ' ERC20 min val rec',
       ' ERC20 max val rec', ' ERC20 avg val rec', ' ERC20 min val sent',
       ' ERC20 max val sent', ' ERC20 avg val sent',
       ' ERC20 min val sent contract', ' ERC20 max val sent contract',
       ' ERC20 avg val sent contract', ' ERC20 uniq sent token name',
       ' ERC20 uniq rec token name', ' ERC20 most sent token type',
       ' ERC20_most_rec_token_type'],
      dtype='object')
```

Dropping Address and Index as they are reputative



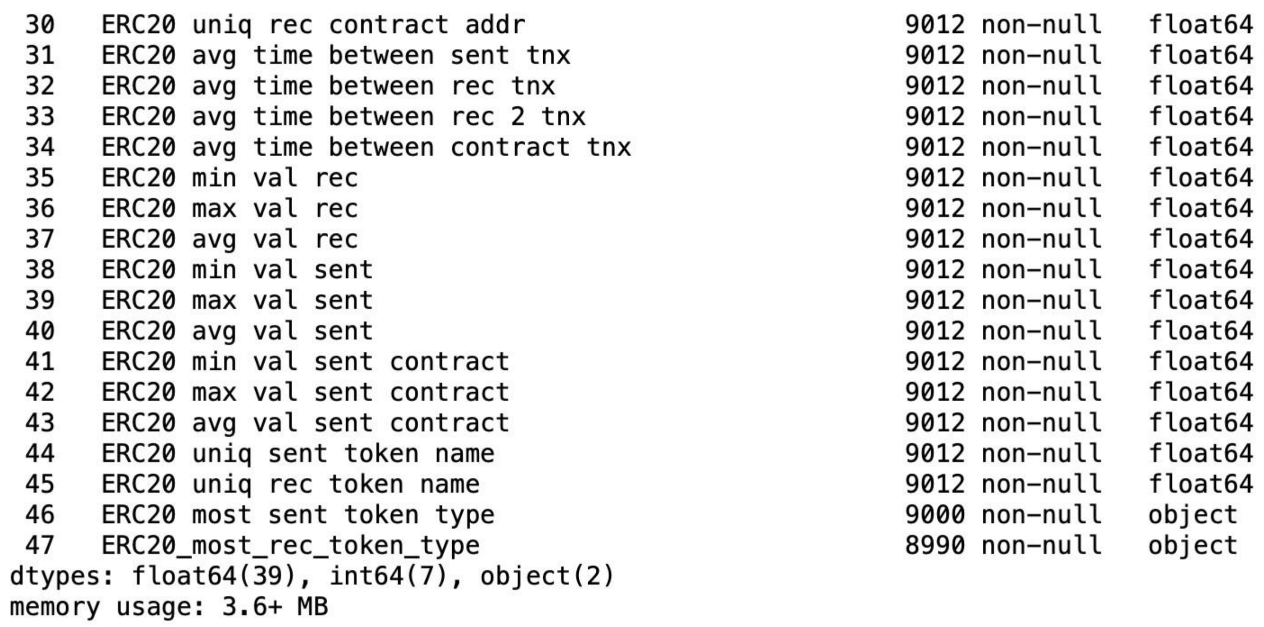
```
data.drop(['Unnamed: 0', 'Address', 'Index'], axis = 1, inplace = True)
```

Information about the columns



```
data.info()
```

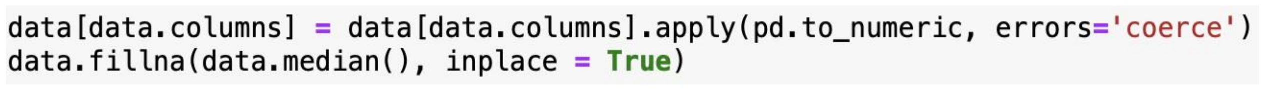
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9841 entries, 0 to 9840
Data columns (total 48 columns):
 #   Column                                               Non-Null Count  Dtype  
---  ------                                               --------------  -----  
 0   FLAG                                                 9841 non-null   int64  
 1   Avg min between sent tnx                             9841 non-null   float64
 2   Avg min between received tnx                         9841 non-null   float64
 3   Time Diff between first and last (Mins)              9841 non-null   float64
 4   Sent tnx                                             9841 non-null   int64  
 5   Received Tnx                                         9841 non-null   int64  
 6   Number of Created Contracts                          9841 non-null   int64  
 7   Unique Received From Addresses                       9841 non-null   int64  
 8   Unique Sent To Addresses                             9841 non-null   int64  
 9   min value received                                   9841 non-null   float64
 10  max value received                                   9841 non-null   float64
 11  avg val received                                     9841 non-null   float64
 12  min val sent                                         9841 non-null   float64
 13  max val sent                                         9841 non-null   float64
 14  avg val sent                                         9841 non-null   float64
 15  min value sent to contract                           9841 non-null   float64
 16  max val sent to contract                             9841 non-null   float64
 17  avg value sent to contract                           9841 non-null   float64
 18  total transactions (including tnx to create contract 9841 non-null   int64  
 19  total Ether sent                                     9841 non-null   float64
 20  total ether received                                 9841 non-null   float64
 21  total ether sent contracts                           9841 non-null   float64
 22  total ether balance                                  9841 non-null   float64
 23   Total ERC20 tnxs                                    9012 non-null   float64
 24   ERC20 total Ether received                          9012 non-null   float64
 25   ERC20 total ether sent                              9012 non-null   float64
 26   ERC20 total Ether sent contract                     9012 non-null   float64
 27   ERC20 uniq sent addr                                9012 non-null   float64
 28   ERC20 uniq rec addr                                 9012 non-null   float64
 29   ERC20 uniq sent addr.1                              9012 non-null   float64
```



```
 30   ERC20 uniq rec contract addr                          9012 non-null   float64
 31   ERC20 avg time between sent tnx                       9012 non-null   float64
 32   ERC20 avg time between rec tnx                        9012 non-null   float64
 33   ERC20 avg time between rec 2 tnx                      9012 non-null   float64
 34   ERC20 avg time between contract tnx                   9012 non-null   float64
 35   ERC20 min val rec                                     9012 non-null   float64
 36   ERC20 max val rec                                     9012 non-null   float64
 37   ERC20 avg val rec                                     9012 non-null   float64
 38   ERC20 min val sent                                    9012 non-null   float64
 39   ERC20 max val sent                                    9012 non-null   float64
 40   ERC20 avg val sent                                    9012 non-null   float64
 41   ERC20 min val sent contract                           9012 non-null   float64
 42   ERC20 max val sent contract                           9012 non-null   float64
 43   ERC20 avg val sent contract                           9012 non-null   float64
 44   ERC20 uniq sent token name                            9012 non-null   float64
 45   ERC20 uniq rec token name                             9012 non-null   float64
 46   ERC20 most sent token type                            9000 non-null   object
 47   ERC20_most_rec_token_type                             8990 non-null   object
dtypes: float64(39), int64(7), object(2)
memory usage: 3.6+ MB
```

Inference: It is clearly evident that there are some missing values in the dataset.

Applying median to fill the null values



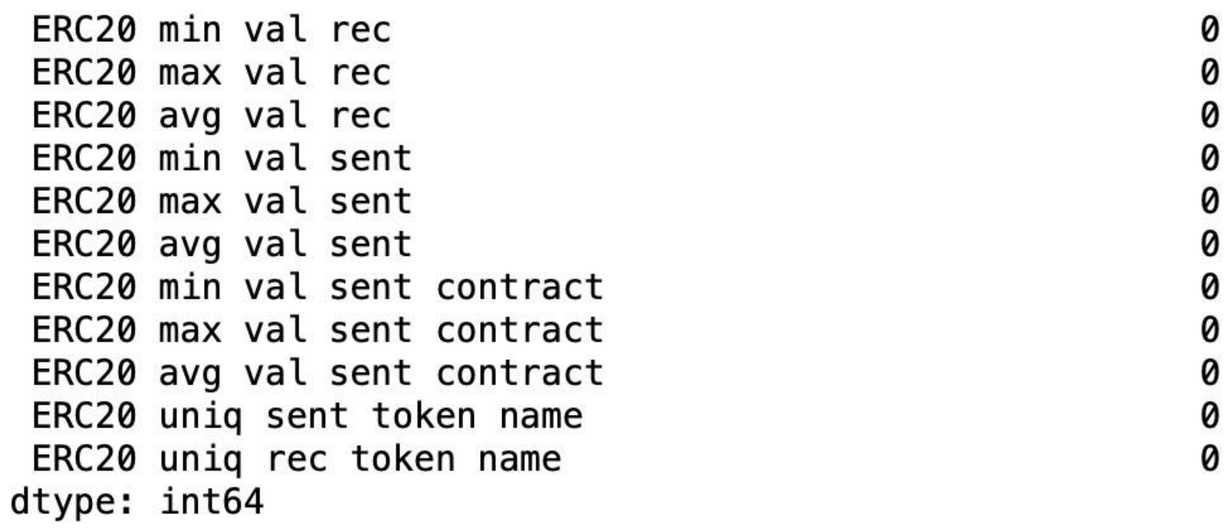
```
data[data.columns] = data[data.columns].apply(pd.to_numeric, errors='coerce')
data.fillna(data.median(), inplace = True)
```

Inference: We can replace the missing values using mean and median. Here I have chosen median because it sorts all the values and replace the missing data with the mid value, while coming to the mean, it is vastly affected by outliers!! So, to replace missing values, median is the preferable method.

Crosschecking the missing data

```
data.isna().sum()
```

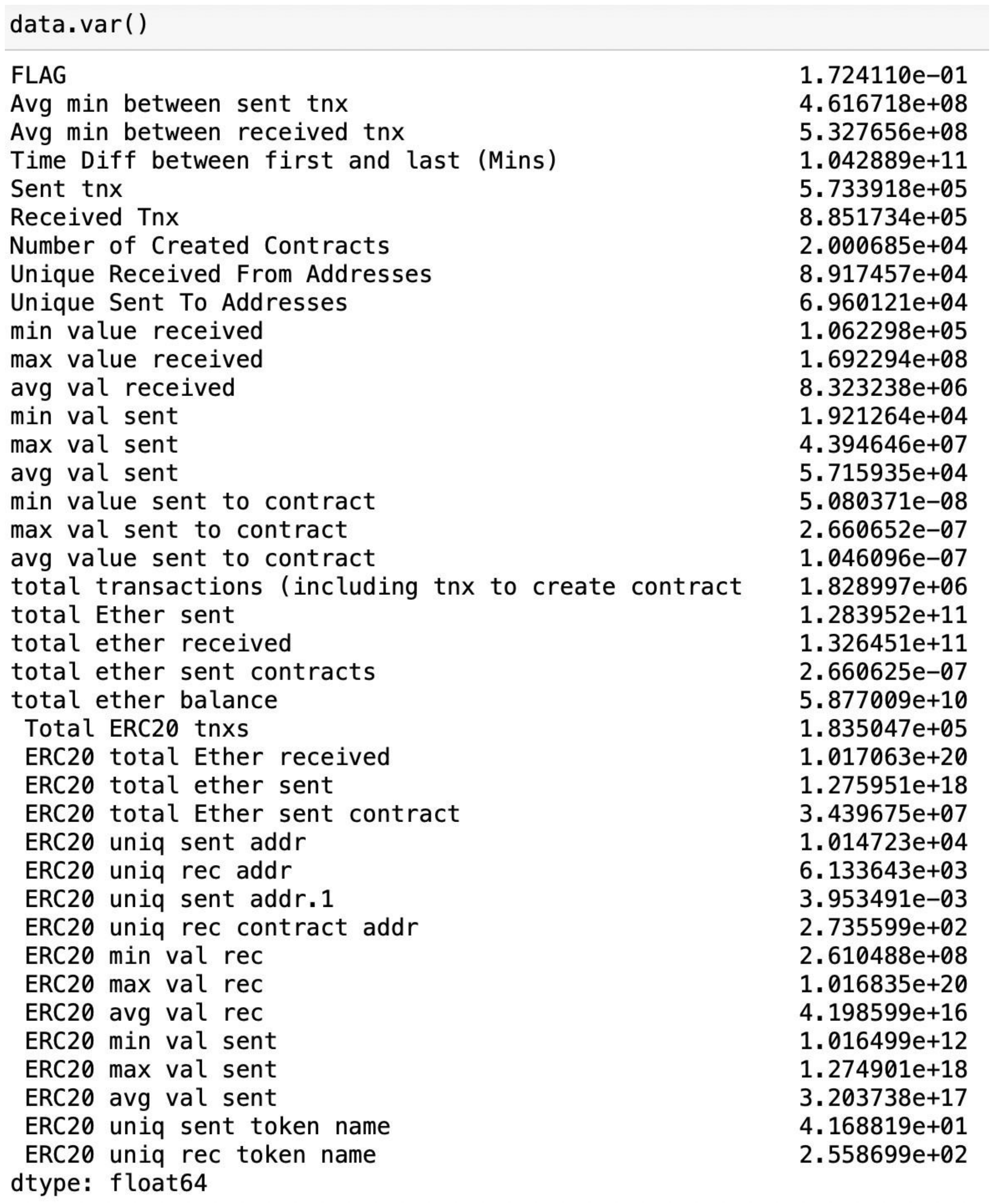
```
FLAG                                                      0
Avg min between sent tnx                                  0
Avg min between received tnx                              0
Time Diff between first and last (Mins)                   0
Sent tnx                                                  0
Received Tnx                                              0
Number of Created Contracts                               0
Unique Received From Addresses                            0
Unique Sent To Addresses                                  0
min value received                                        0
max value received                                        0
avg val received                                          0
min val sent                                              0
max val sent                                              0
avg val sent                                              0
min value sent to contract                                0
max val sent to contract                                  0
avg value sent to contract                                0
total transactions (including tnx to create contract      0
total Ether sent                                          0
total ether received                                      0
total ether sent contracts                                0
total ether balance                                       0
 Total ERC20 tnxs                                         0
 ERC20 total Ether received                               0
 ERC20 total ether sent                                   0
 ERC20 total Ether sent contract                          0
 ERC20 uniq sent addr                                     0
 ERC20 uniq rec addr                                      0
 ERC20 uniq sent addr.1                                   0
 ERC20 uniq rec contract addr                             0
 ERC20 avg time between sent tn                           0
 ERC20 avg time between rec tnx                           0
 ERC20 avg time between rec 2 t x                         0
 ERC20 avg time between contract tnx                      0
```



```
 ERC20 min val rec                                         0
 ERC20 max val rec                                         0
 ERC20 avg val rec                                         0
 ERC20 min val sent                                        0
 ERC20 max val sent                                        0
 ERC20 avg val sent                                        0
 ERC20 min val sent contract                               0
 ERC20 max val sent contract                               0
 ERC20 avg val sent contract                               0
 ERC20 uniq sent token name                                0
 ERC20 uniq rec token name                                 0
dtype: int64
```

Now, we can say that there are no more missing data in the dataset.

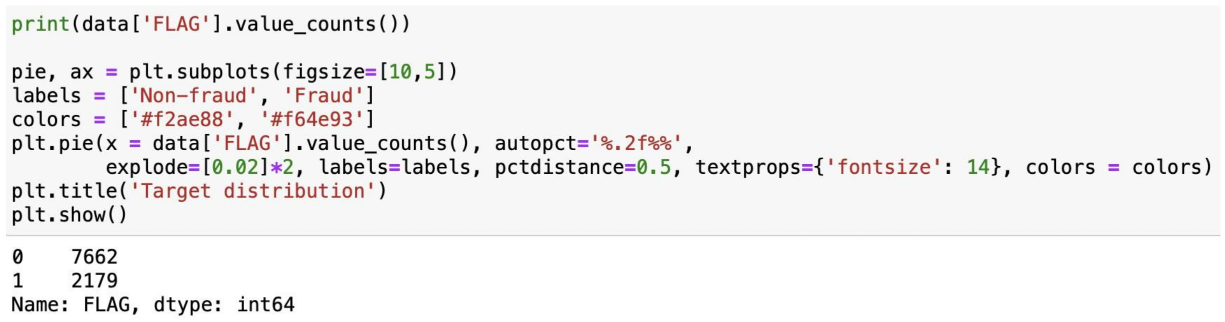
Calculating the variance to know the degree of spread in the dataset.



```
data.var()
```

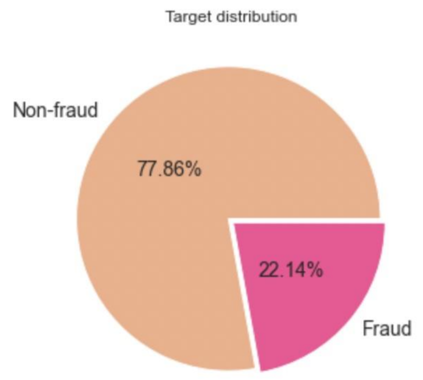
```
FLAG                                                        1.724110e-01
Avg min between sent tnx                                    4.616718e+08
Avg min between received tnx                                5.327656e+08
Time Diff between first and last (Mins)                     1.042889e+11
Sent tnx                                                    5.733918e+05
Received Tnx                                                8.851734e+05
Number of Created Contracts                                 2.000685e+04
Unique Received From Addresses                              8.917457e+04
Unique Sent To Addresses                                    6.960121e+04
min value received                                          1.062298e+05
max value received                                          1.692294e+08
avg val received                                            8.323238e+06
min val sent                                                1.921264e+04
max val sent                                                4.394646e+07
avg val sent                                                5.715935e+04
min value sent to contract                                  5.080371e-08
max val sent to contract                                    2.660652e-07
avg value sent to contract                                  1.046096e-07
total transactions (including tnx to create contract        1.828997e+06
total Ether sent                                            1.283952e+11
total ether received                                        1.326451e+11
total ether sent contracts                                  2.660625e-07
total ether balance                                         5.877009e+10
 Total ERC20 tnxs                                           1.835047e+05
 ERC20 total Ether received                                 1.017063e+20
 ERC20 total ether sent                                     1.275951e+18
 ERC20 total Ether sent contract                            3.439675e+07
 ERC20 uniq sent addr                                       1.014723e+04
 ERC20 uniq rec addr                                        6.133643e+03
 ERC20 uniq sent addr.1                                     3.953491e-03
 ERC20 uniq rec contract addr                               2.735599e+02
 ERC20 min val rec                                          2.610488e+08
 ERC20 max val rec                                          1.016835e+20
 ERC20 avg val rec                                          4.198599e+16
 ERC20 min val sent                                         1.016499e+12
 ERC20 max val sent                                         1.274901e+18
 ERC20 avg val sent                                         3.203738e+17
 ERC20 uniq sent token name                                 4.168819e+01
 ERC20 uniq rec token name                                  2.558699e+02
dtype: float64
```

Our entire dataset is dependent on the Flag column of the dataset to ensure the transaction is fraud or not. So, Here is the Pie chart showing the fraud and other transactions.



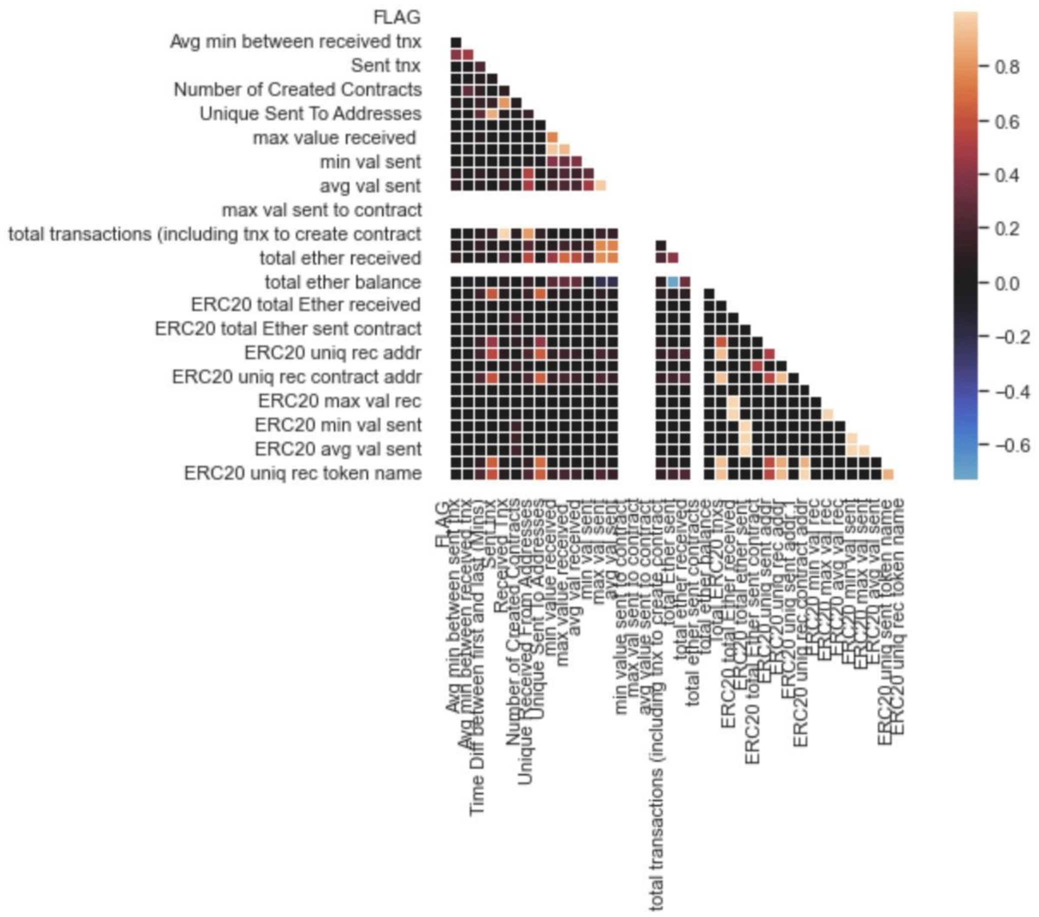
```
print(data['FLAG'].value_counts())

pie, ax = plt.subplots(figsize=[10,5])
labels = ['Non-fraud', 'Fraud']
colors = ['#f2ae88', '#f64e93']
plt.pie(x = data['FLAG'].value_counts(), autopct='%.2f%%',
        explode=[0.02]*2, labels=labels, pctdistance=0.5, textprops={'fontsize': 14}, colors = colors)
plt.title('Target distribution')
plt.show()
```

```
0    7662
1    2179
Name: FLAG, dtype: int64
```



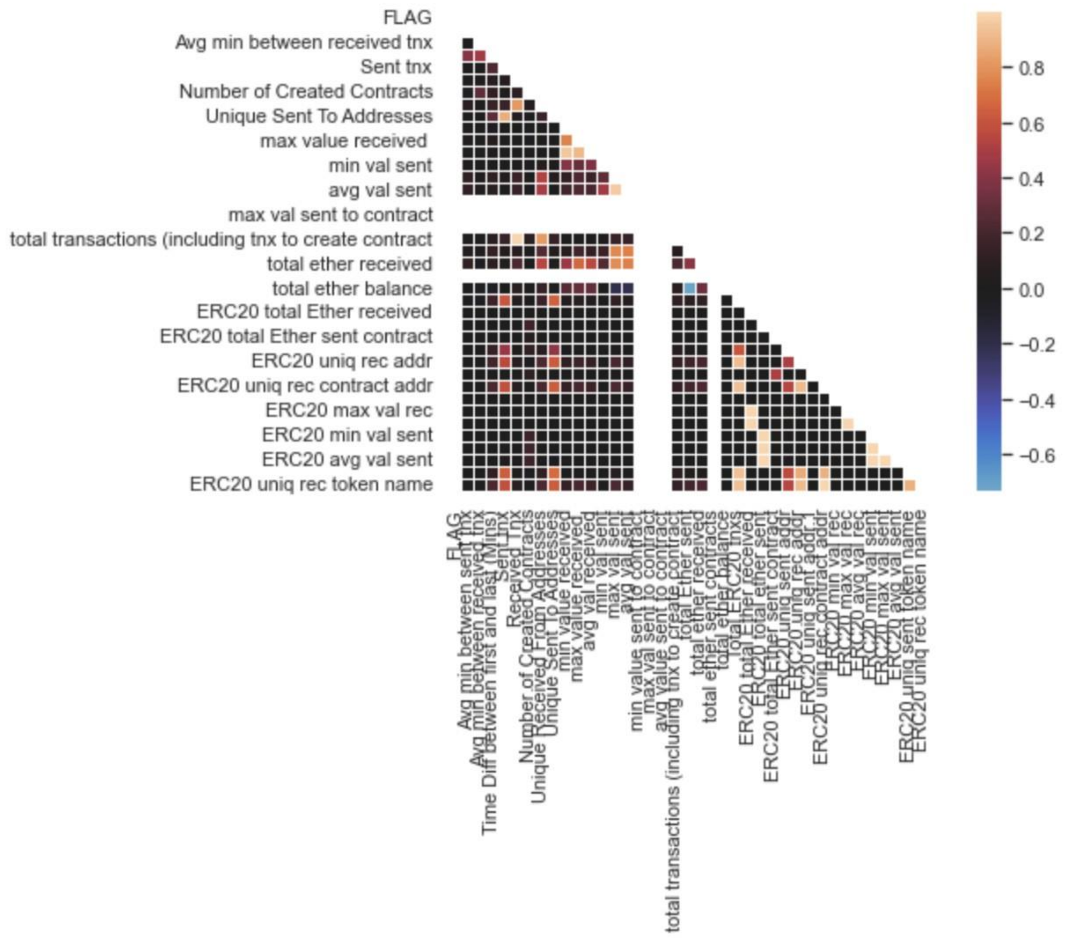
Finding the correlation of the fraudulent transactions!!

```
data_fraud = data[data['FLAG']==1]
corr = data_fraud.corr()
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)]=True
with sns.axes_style('white'):
    fig, ax = plt.subplots(figsize=(10,5))
    sns.heatmap(corr,  mask=mask, annot=False,center=0, linewidths=0.8, square=True)
```

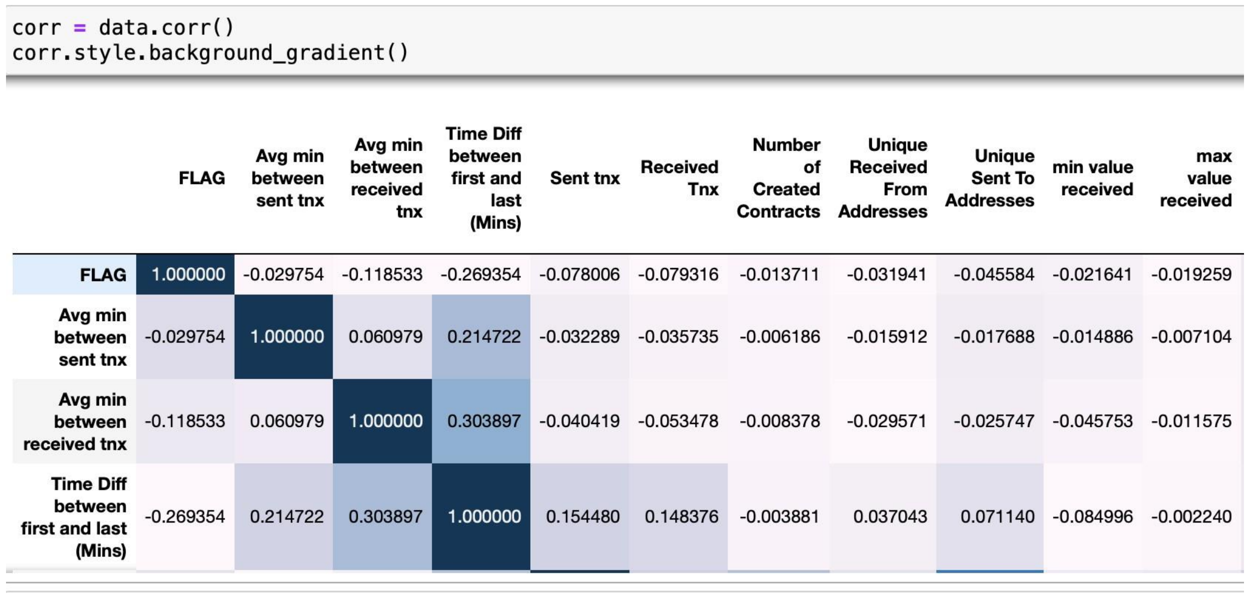


Finding the correlation of non-fraudlent transactions!!

```
data_not_fraud = data[data['FLAG']==0]
corr = data_fraud.corr()
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)]=True
with sns.axes_style('white'):
    fig, ax = plt.subplots(figsize=(10,5))
    sns.heatmap(corr, mask=mask, annot=False,center=0, linewidths=0.8, square=True)
```



Correlation of entire dataset (screenshot captured only 1/10th part)

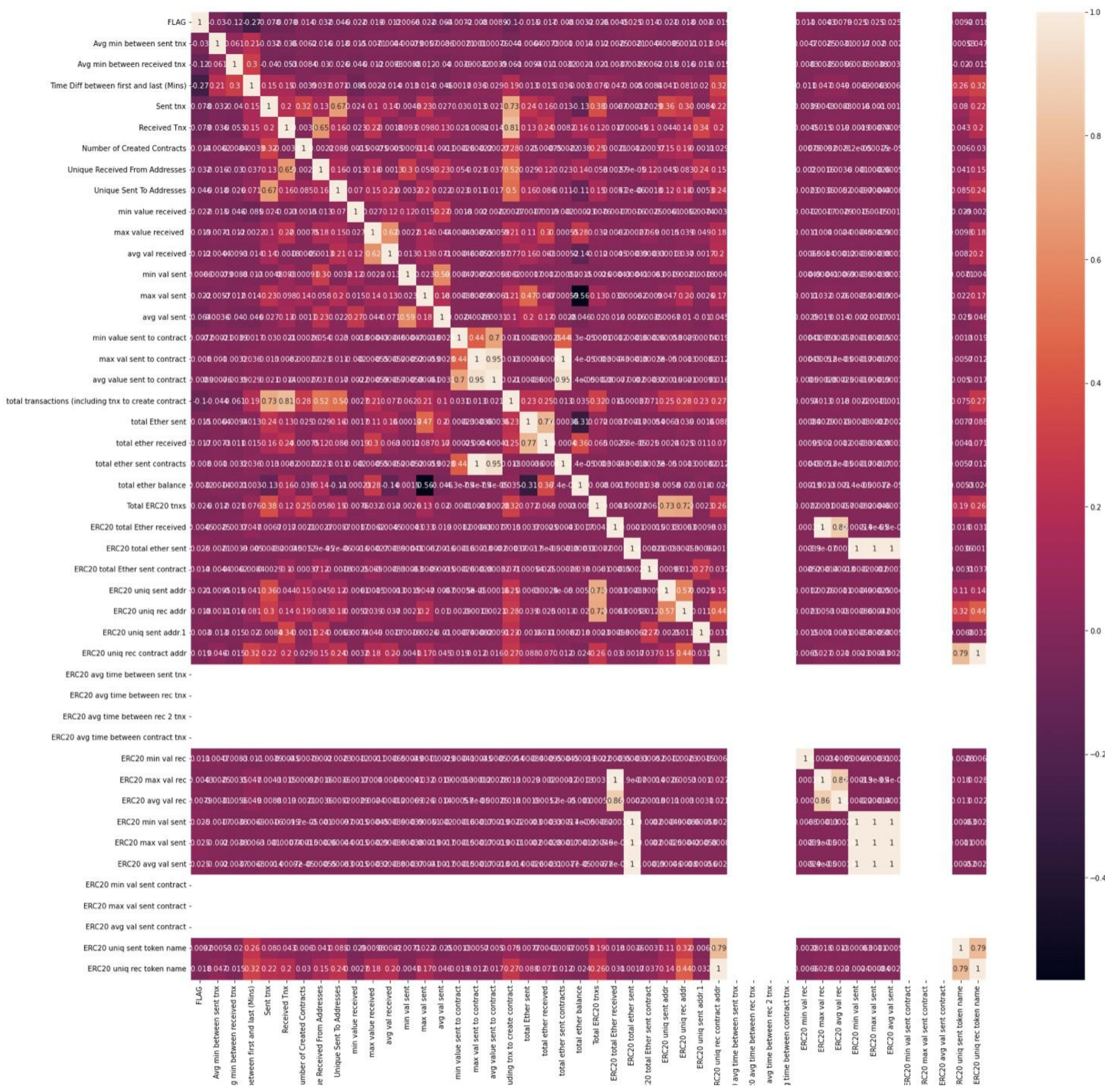


```
corr = data.corr()
corr.style.background_gradient()
```

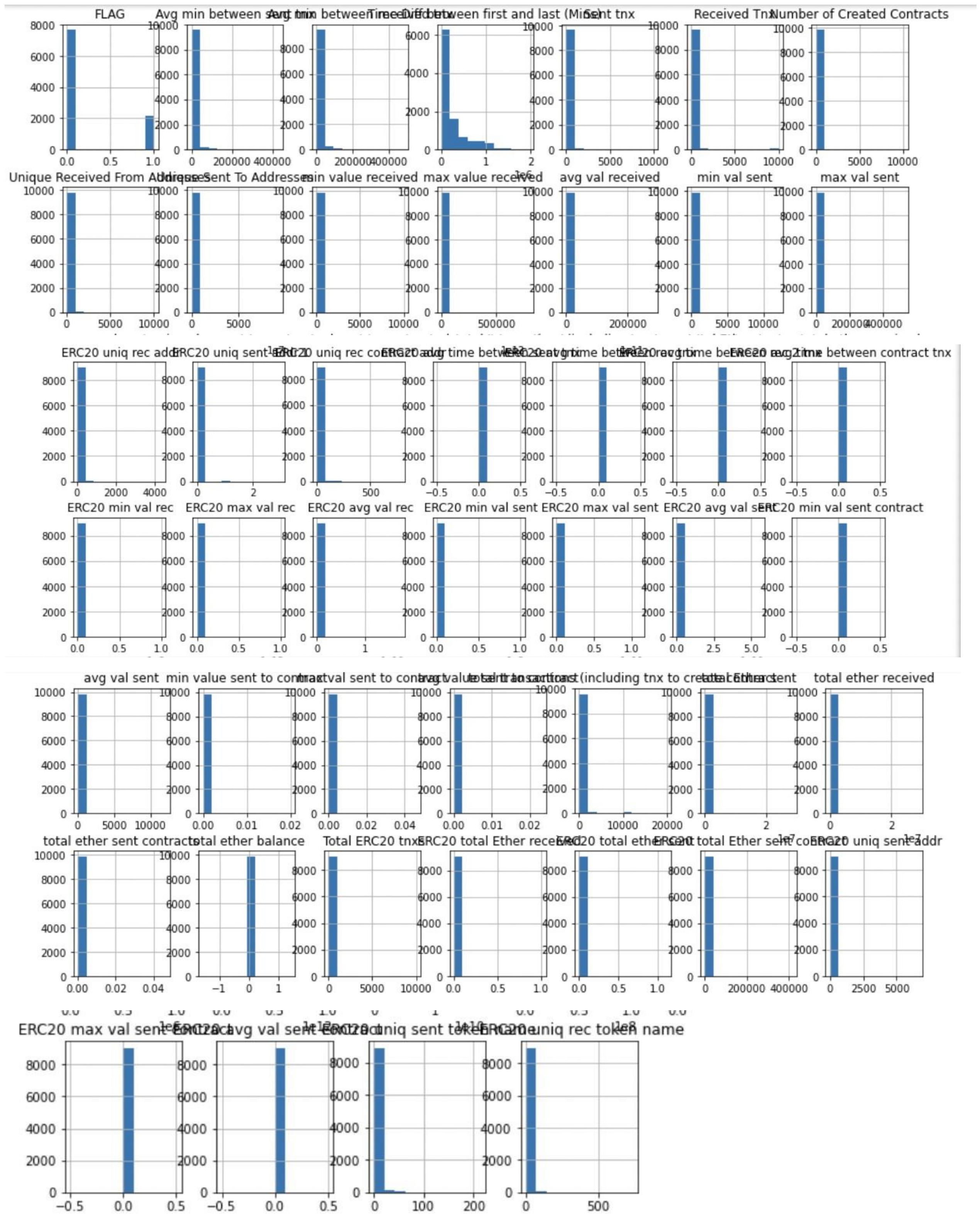
	FLAG	Avg min between sent tnx	Avg min between received tnx	Time Diff between first and last (Mins)	Sent tnx	Received Tnx	Number of Created Contracts	Unique Received From Addresses	Unique Sent To Addresses	min value received	max value received
FLAG	1.000000	-0.029754	-0.118533	-0.269354	-0.078006	-0.079316	-0.013711	-0.031941	-0.045584	-0.021641	-0.019259
Avg min between sent tnx	-0.029754	1.000000	0.060979	0.214722	-0.032289	-0.035735	-0.006186	-0.015912	-0.017688	-0.014886	-0.007104
Avg min between received tnx	-0.118533	0.060979	1.000000	0.303897	-0.040419	-0.053478	-0.008378	-0.029571	-0.025747	-0.045753	-0.011575
Time Diff between first and last (Mins)	-0.269354	0.214722	0.303897	1.000000	0.154480	0.148376	-0.003881	0.037043	0.071140	-0.084996	-0.002240

Heat map - to show the relationship between the variables

```
plt.figure(figsize=(25,25))
sns.heatmap(data.corr(), annot = True)
plt.show()
```



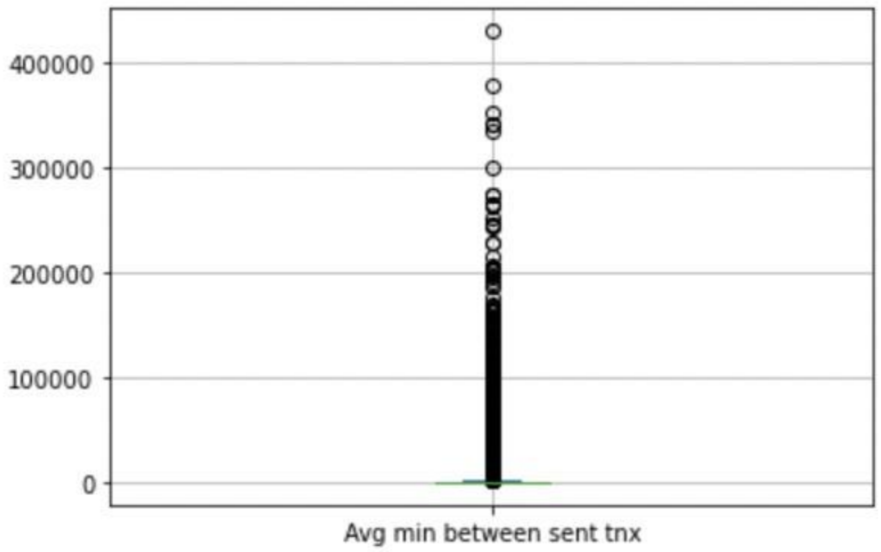
Histograms - to illustrate major features of the distribution



Box plot visulatization

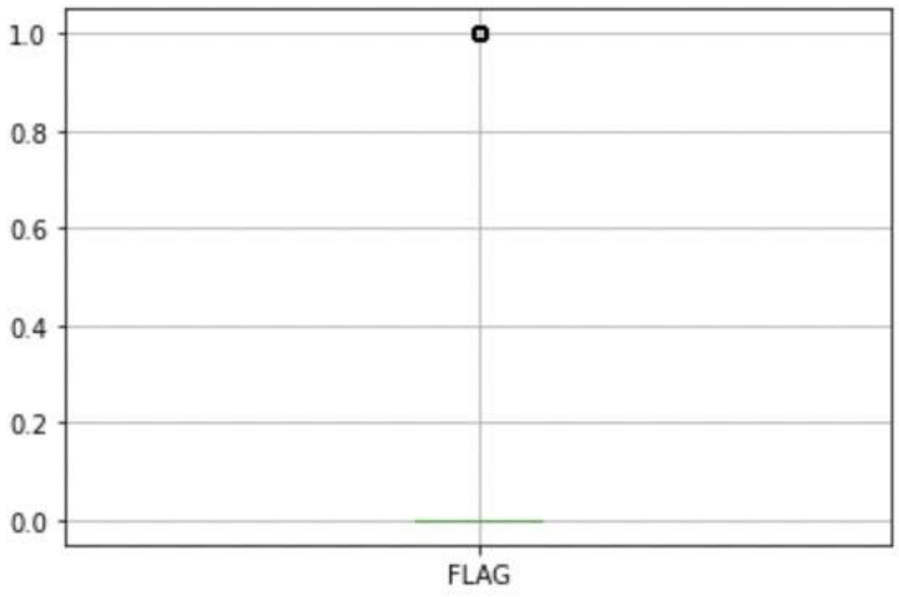
```
data['Avg min between sent tnx'].plot.box(grid = True)
```

```
<AxesSubplot:>
```



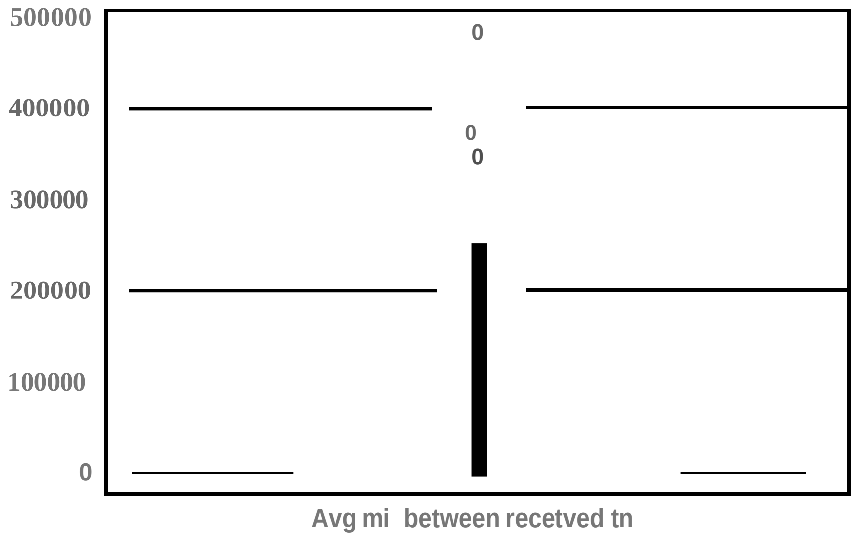
```
data['FLAG'].plot.box(grid = True)
```

```
<AxesSubplot:>
```



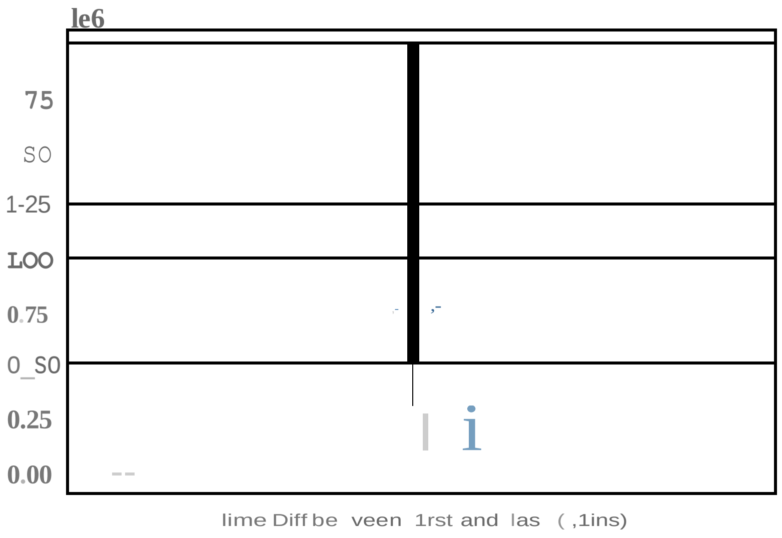
```
data['Avg min between received tnx'].plot.box(grid = True)
```

```
<AxesSubplot:>
```

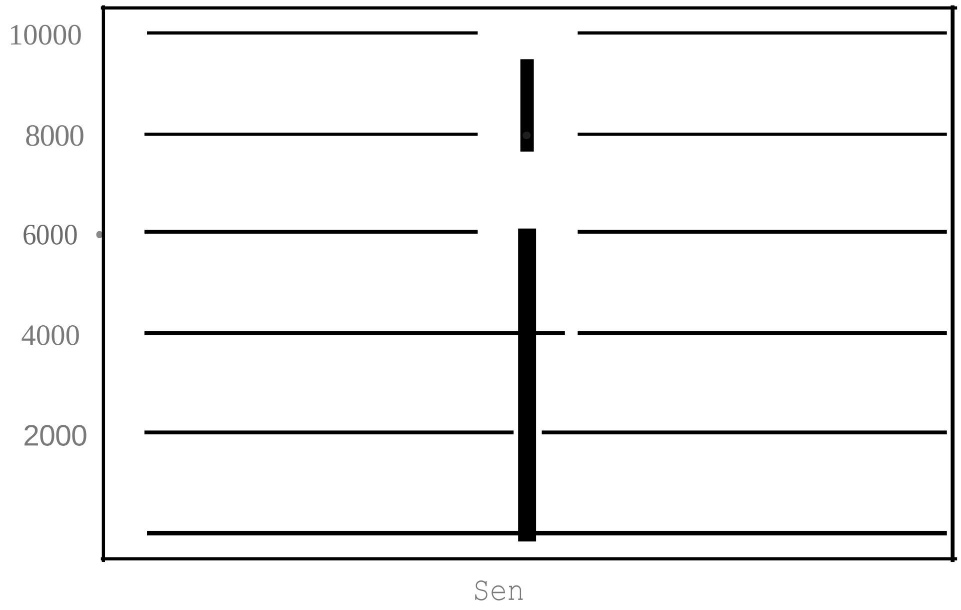


```
data['Time Diff between first and last (Mins)'].plot.box(grid = True)
```

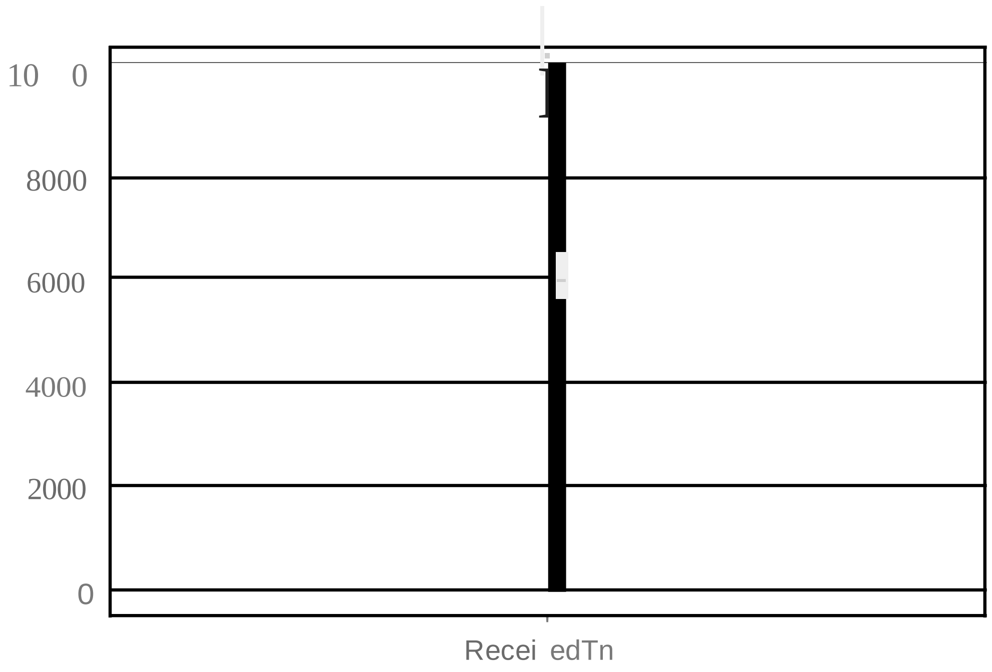
```
<AxesSubplot:>
```



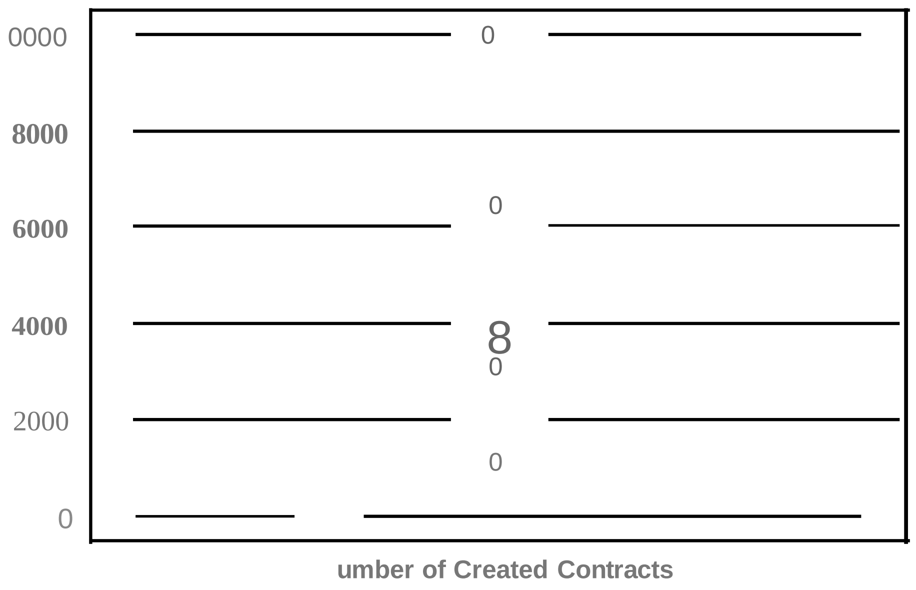
```
data['Sent tnx'].plot.box{grid = True)
<AxesSubplot:>
```



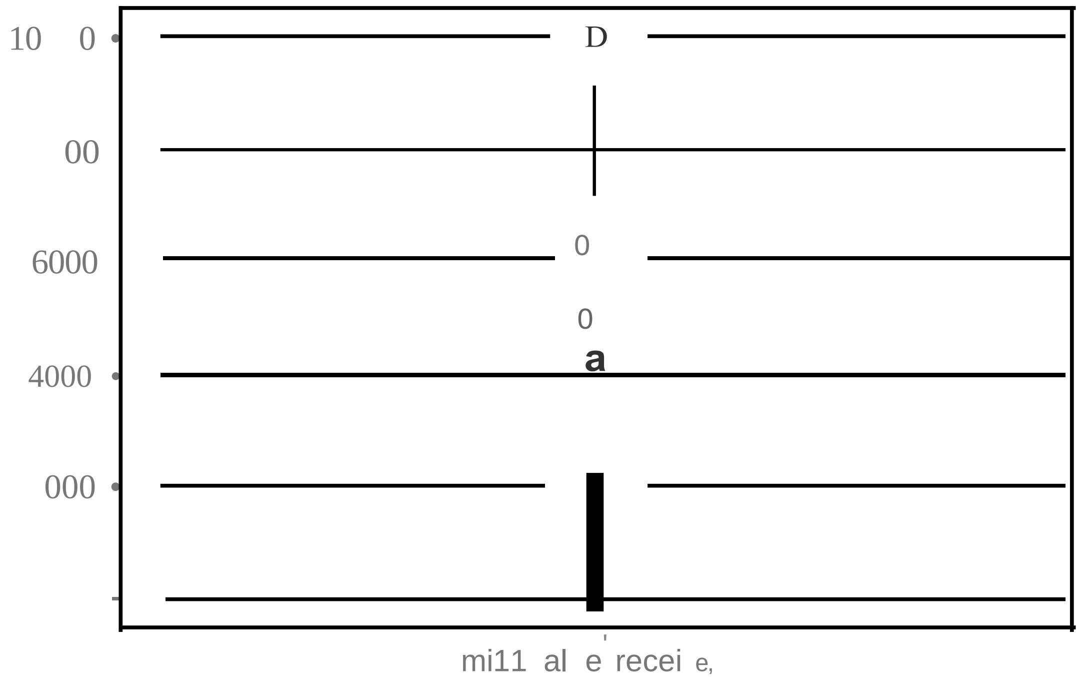
```
data['Received Tnx'].plot.box(grid = True)
<AxesSubplot:>
```



```
data['Number of Created Contracts'].plot.box(grid = True)
<AxesSubplot:>
```

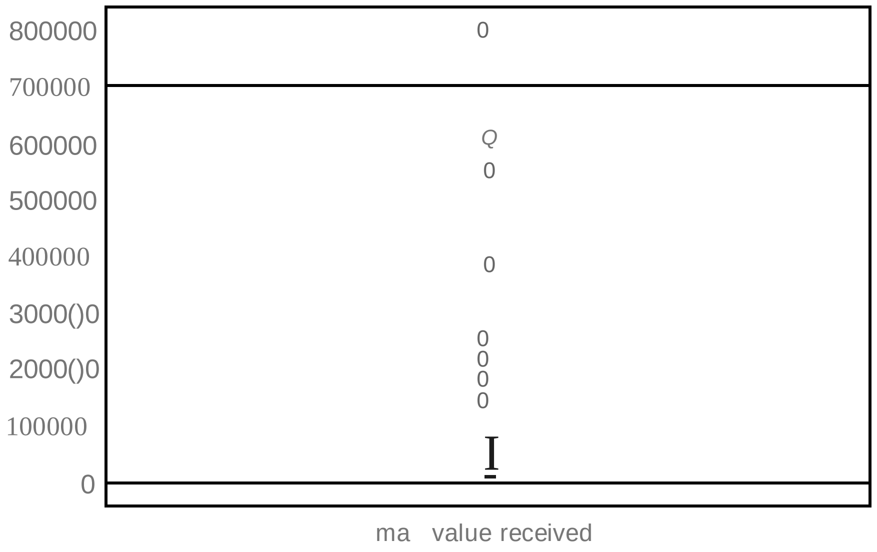


```
data['min value received'].plot.box(grid       True)
<AxesSubplot:>
```



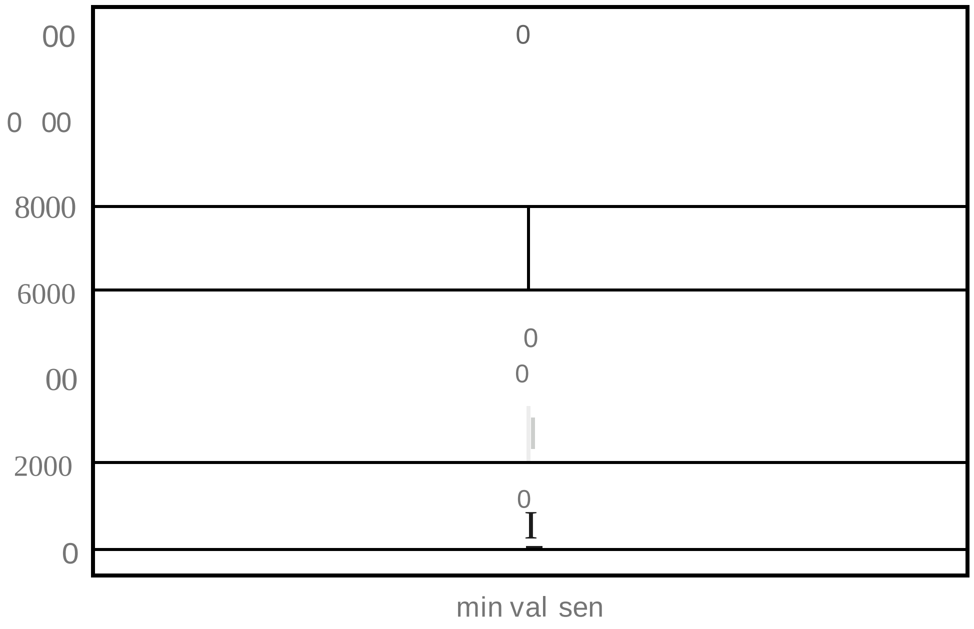
```
data['max value received '].plot.box(grid = True)
```

```
<AxesSubplot:>
```

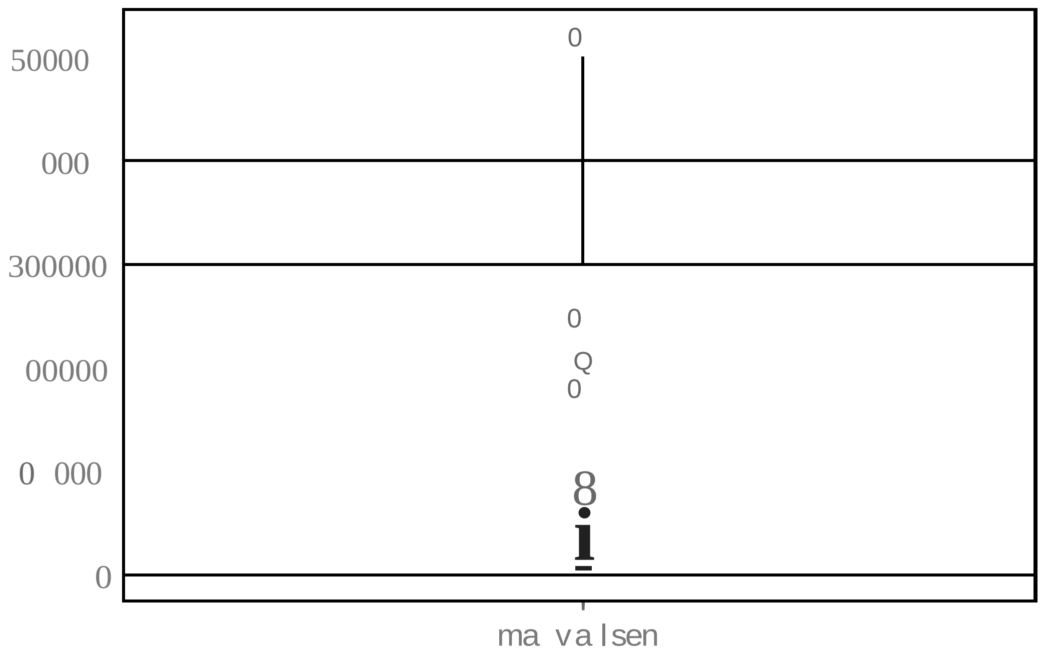


```
data['min val sent ']. plot.box(grid = True)
```

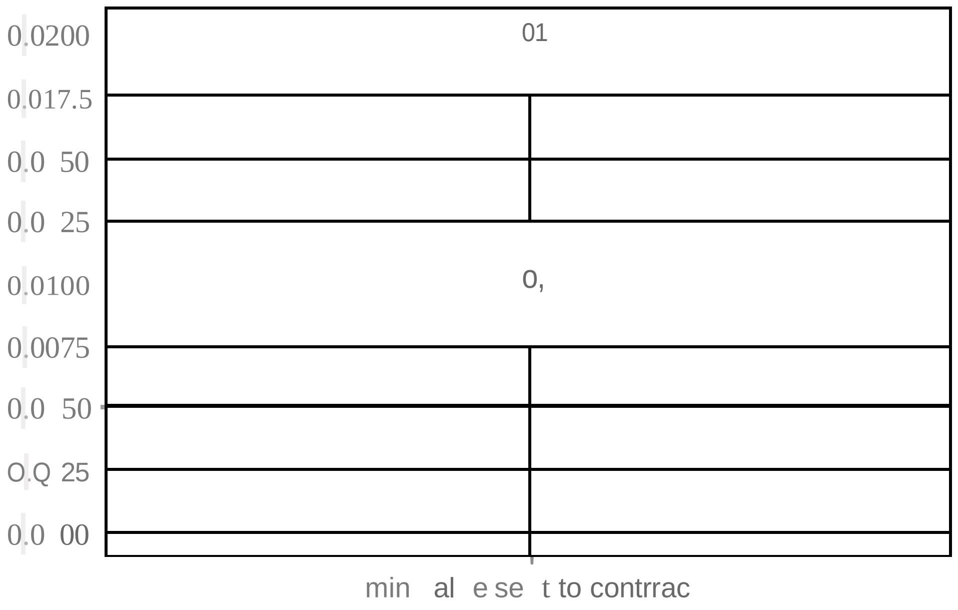
```
<AxesSubplot:>
```



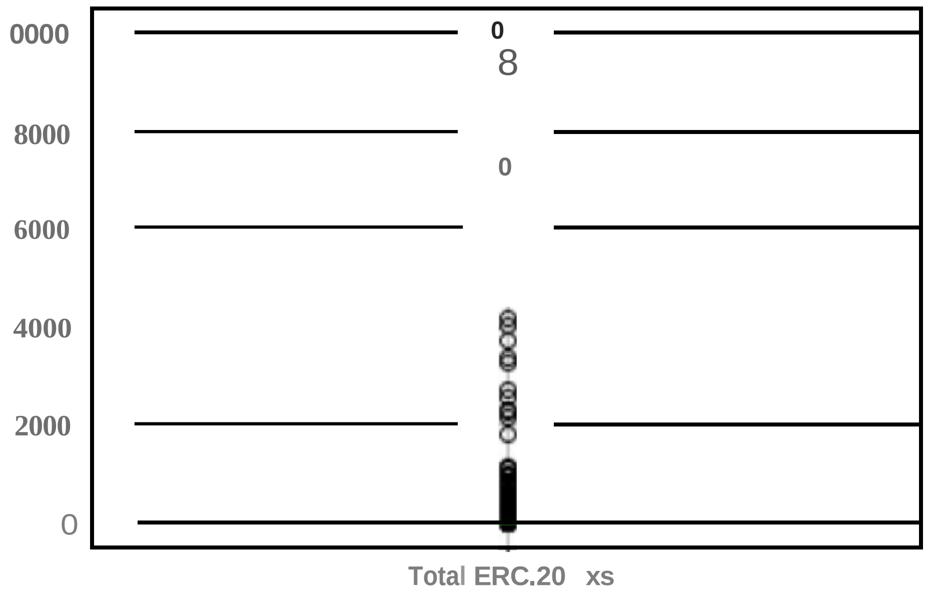
```
data['max val sent'].plot.box(grid = True)
<AxesSubplot:>
```



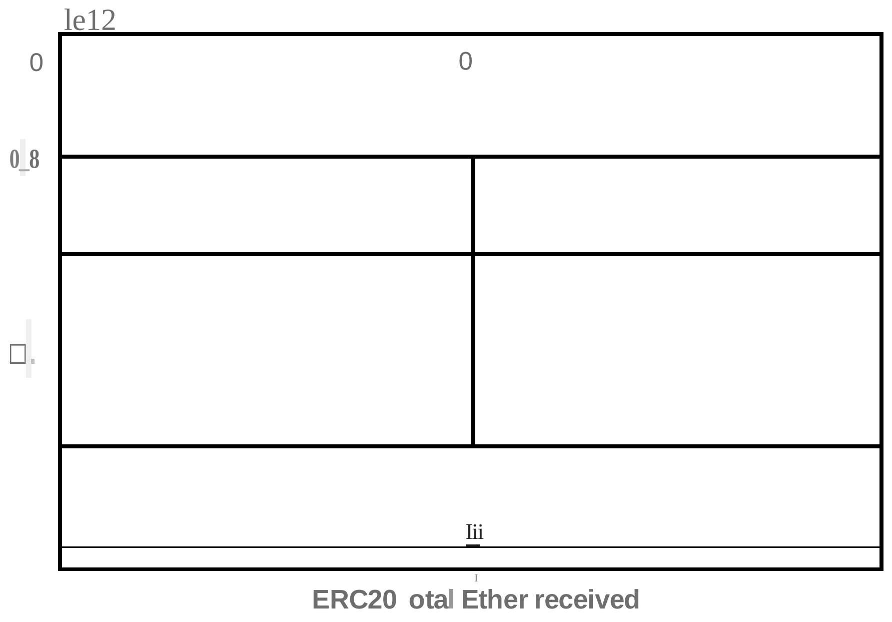
```
data['min value sent to contract'].plot.box(grid = True)
<AxesSubplot:>
```



```
data['Total ERC20 tnxs'].plot.box(grid = True)
<AxesSubplot:>
```

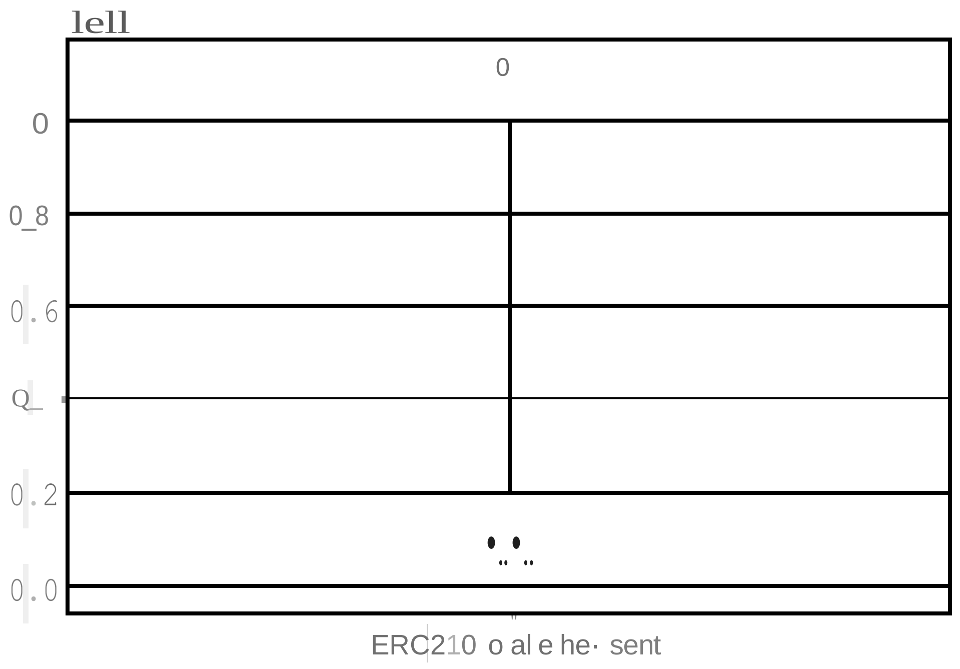


```
data['ERC20 total Ether received'].plot.box(grid = True)
<AxesSubplot:>
```



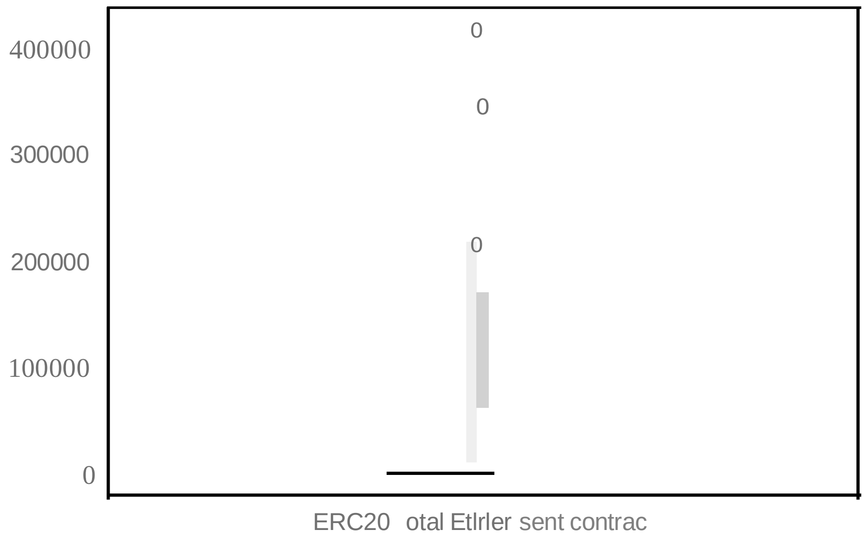
```
data['ERC20 total ether sent'].plot.box(grid = True)
```

```
<AxesSubplot:>
```



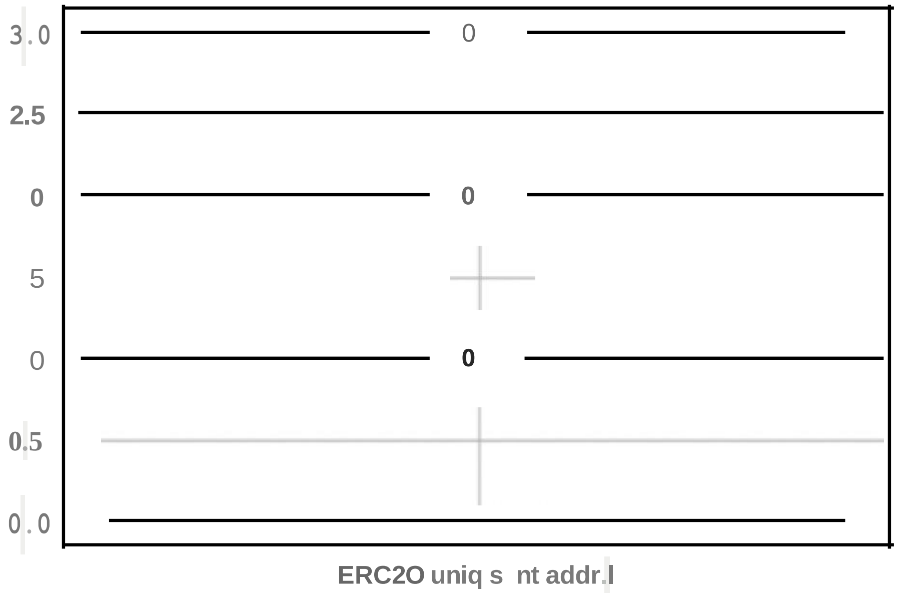
```
data['ERC20 total Ether sent contract'].plot.box(grid = True)
```

```
<AxesSubplot:>
```



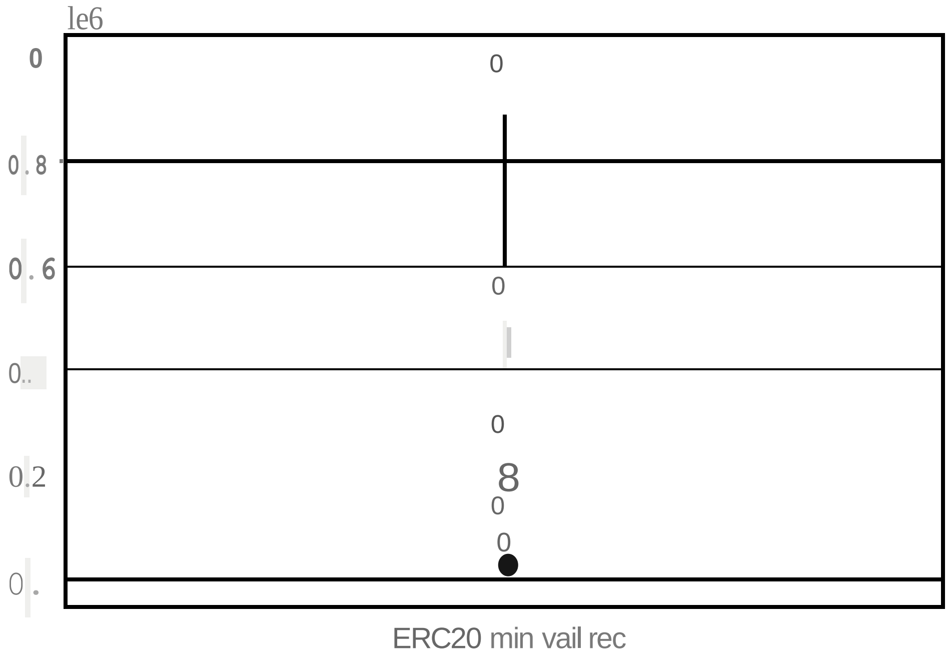
```
data['ERC20 uniq sent addr.1'].plot.box(grid = True)
```

```
<AxesSubplot:>
```



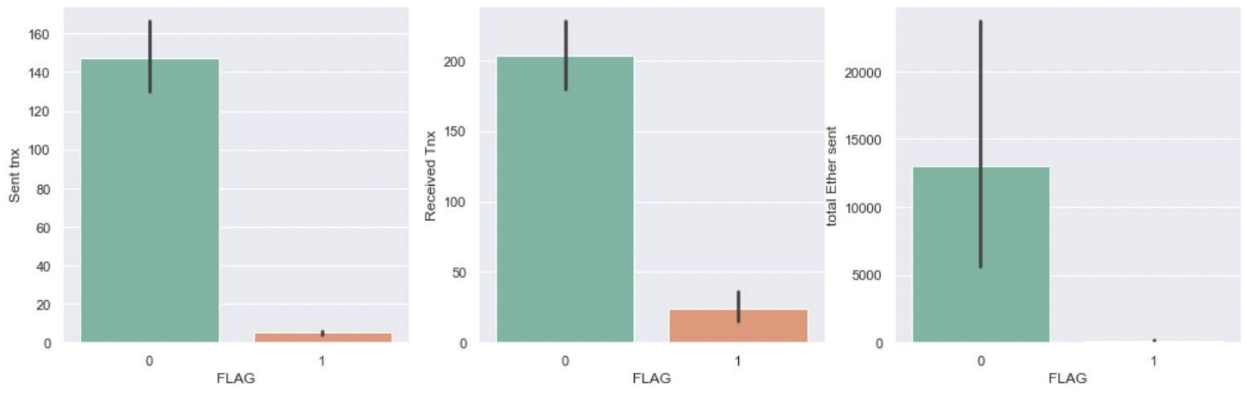
```
data['ERC20 min val rec'].plot.box(grid = True)
```

```
<AxesSubplot:>
```



Bar graph

```
plt.figure(figsize=(17,5))
plt.subplot(1,3,1)
sns.barplot(y = 'Sent tnx', x='FLAG', palette='Set2', data = data)
plt.subplot(1,3,2)
sns.barplot(y = 'Received Tnx', x='FLAG', palette='Set2', data = data)
plt.subplot(1,3,3)
sns.barplot(y = 'total Ether sent', x='FLAG', palette='Set2', data = data)
plt.show()
```



Inference: We know that maximum of the fraud happens only during the transactions. So, we can visualize the FLAG column with respect to transaction details.