Bitcoin Wallet Analysis

By: Julia Farson

1. Introduction

In this notebook, I will be exploring Bitcoin transactions from a target BTC wallet. The target wallet I will be analzying is Wallet 18c52f4ad1 found on WalletExplorer.com.

WalletExplorer.com is a Bitcoin blockchain explorer that uses a basic algorithm to cluster BTC addresses into wallets.

The data set can be viewed at the following link: https://www.walletexplorer.com/wallet/18c52f4ad14b9997?from_address=12wWGGS74QpKdigWY4GWVxPivJ8pKNEgU2

Through my analysis I will answer the following questions:

- What suspicious transactions, if any, are occuring within the target wallet?
- Which wallets/entities sent the most virtual curency to the target wallet?
- Which wallets/entities recieved the most virtual currency from the target wallet?

2. Importing and Cleaning Data

To begin my exploratory data analysis process, I imported the required packages needed to load, clean, and later visualize the data.

In [64]:
```
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('seaborn')
import folium
import plotly as px
```

Next I uploaded the cryptocurrency wallet data set.

In [44]:
```
crypto_wallet = pd.read_csv('C:/Users/jfars/OneDrive/Desktop/crypto project/crypto_wallet_explorer.csv')
```

I began the data cleaning process by checking for duplicate values.

In [45]:
```
crypto_wallet.duplicated().sum()
```

Out[45]:
```
0
```

Next I reviewed the data types within the dataframe.

In [46]:
```
print(crypto_wallet.dtypes)
```

```
date               object
received from      object
received amount   float64
sent amount       float64
sent to            object
balance           float64
transaction        object
dtype: object
```

Since the 'date' column was an object, I converted the 'date' column to the proper date-time format and double checked that the changes were made.

In [47]:
```
crypto_wallet.date = pd.to_datetime(crypto_wallet.date, format='%m/%d/%Y %M:%S')
print(crypto_wallet.dtypes)
```

```
date               datetime64[ns]
received from              object
received amount           float64
sent amount               float64
sent to                    object
balance                   float64
transaction                object
dtype: object
```

3. Analysis and Visualizations

I began the analysis process by identifying any suspicious transactions. While viewing the data set, I found multiple transactions between the target wallet and DarkNet markets.

In [48]:
```
options = ['NucleusMarket (0001a5d354cf4f44)', 'AlphaBayMarket (000078c74c35206a)']

received_from_darknet_market= crypto_wallet[crypto_wallet['received from'].isin(options)]
```

In [49]:
```
received_from_darknet_market.head(15)
```

Out[49]:

	date	received from	received amount	sent amount	sent to	balance	transaction
30	2016-03-26 00:05:15	NucleusMarket (0001a5d354cf4f44)	0.018000	NaN	NaN	2.591976	0a8dff9f7508c739c9a67d673d5f5bf23b8ab5a8f9bfdc...
34	2016-03-23 00:23:04	NucleusMarket (0001a5d354cf4f44)	0.015394	NaN	NaN	2.129982	802a7e393e77f7d6e7ad4b904a7eb660a3fe117db29b61...
51	2016-03-03 00:03:41	NucleusMarket (0001a5d354cf4f44)	0.016000	NaN	NaN	4.256629	4e66d51b8e2f25b37bbe1fc527e809f5159a443bedfde7...
53	2016-02-28 00:19:52	NucleusMarket (0001a5d354cf4f44)	0.023833	NaN	NaN	4.235599	ab6bf6b056627eff084f2467a65d90a8f53b5089e57749...
57	2016-02-20 00:09:40	NucleusMarket (0001a5d354cf4f44)	0.007761	NaN	NaN	4.114073	395df7a337781a0f0df205271261e18e56522f0860d87c...
62	2016-02-14 00:01:40	NucleusMarket (0001a5d354cf4f44)	0.014000	NaN	NaN	8.598825	c93ce2863517e4d47154cb7f8ab1511722e4bf3b4c2d9a...
63	2016-02-13 00:15:24	AlphaBayMarket (000078c74c35206a)	0.062100	NaN	NaN	8.584825	1d0fd5167c89b92e8414adc4faf72b1358880cc1d25589...
65	2016-02-12 00:00:31	AlphaBayMarket (000078c74c35206a)	0.025238	NaN	NaN	8.496514	1cda47fe06506d6638f3676ce889d1d468a454718b91c4...
71	2016-02-04 00:22:25	AlphaBayMarket (000078c74c35206a)	0.028016	NaN	NaN	2.237033	35da1cb3d223aef08f2ddda3289d4e91acc0b4511b7590...
77	2016-01-31 00:18:41	AlphaBayMarket (000078c74c35206a)	0.096361	NaN	NaN	10.527109	5e0db0c91843ba9968e2728df46a5a0f6cdbfd8e9b1a63...
93	2016-01-22 00:09:08	NucleusMarket (0001a5d354cf4f44)	0.027062	NaN	NaN	3.975924	b331d13c79f10787789662024c10c72e6bb54190cba1b9...
95	2016-01-22 00:02:55	NucleusMarket (0001a5d354cf4f44)	0.007190	NaN	NaN	1.948862	4edafc219cb553a244199dee3c3a8202b09783696ddfb8...
96	2016-01-22 00:01:17	NucleusMarket (0001a5d354cf4f44)	0.011114	NaN	NaN	1.941671	dd5dbfe0829629a21d2028e8343f59b8e2810d50c1e3f8...

The above table shows transactions received from DarkNet Markets including Nuclues Market and Alpha Bay Market.

In [50]:
```
options = ['NucleusMarket (0001a5d354cf4f44)', 'AlphaBayMarket (000078c74c35206a)']

sent_to_darknet_market= crypto_wallet[crypto_wallet['sent to'].isin(options)]
```

In [51]:
```
print(sent_to_darknet_market)
```

```
Empty DataFrame
Columns: [date, received from, received amount, sent amount, sent to, balance, transaction]
Index: []
```

The code output above shows that the target wallet did not send any currency to wallets associated with DarkNet Markets.

In [52]:
```
received_from_darknet_market.count()
```

Out[52]:
```
date               13
received from      13
received amount    13
sent amount         0
sent to             0
balance            13
transaction        13
dtype: int64
```

The table above shows that 13 total transactions occured with wallets associated with DarkNet Markets. All of these transactions entailed the target wallet receiving virtual currency.

Next I analyzed what wallets sent the most virtual currency to the target wallet through a pivot table.

In [53]:
```
pivot1 = crypto_wallet.pivot_table(index=['received from'], values=['received amount'], aggfunc='sum').sort_values('received amount', ascending=False)
pivot1.reset_index
```

Out[53]:
```
<bound method DataFrame.reset_index of                   received amount
received from
fa4b2eaa533f1721        18.000000
2481ea7f4dd3c4f0        10.000000
e465a8e2dbf482f9         8.725491
e43080b947a787e5         5.725282
0046bd062801670f         2.000000
...                           ...
d22a425f65f61333         0.002600
5d0fcba41e5d01f6         0.002539
385f03232edf2f2b         0.002400
1c5ce502191e21a0         0.002300
f81881ac3abe276a         0.002277

[78 rows x 1 columns]>
```

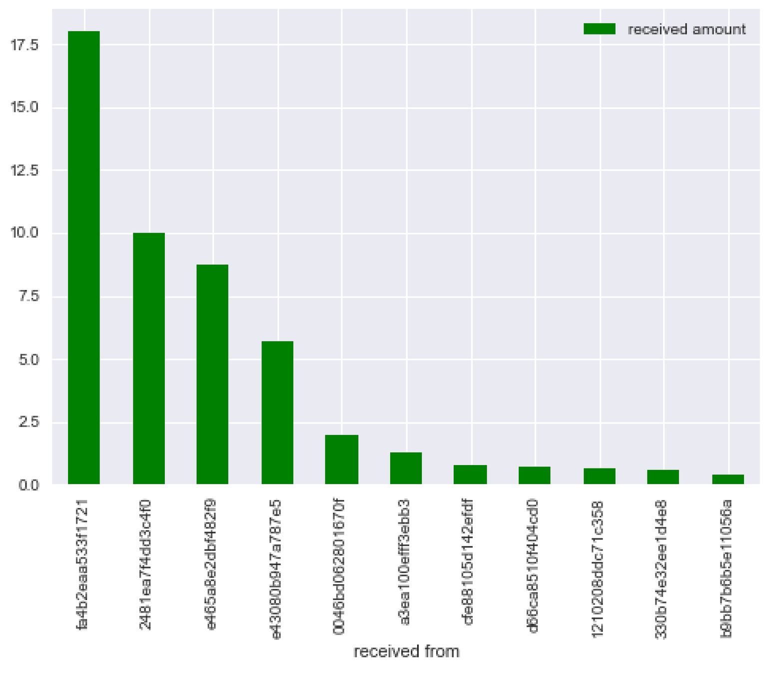
In [54]:
```
pivot1.head(10)
```

Out[54]:

received from	received amount
fa4b2eaa533f1721	18.000000
2481ea7f4dd3c4f0	10.000000
e465a8e2dbf482f9	8.725491
e43080b947a787e5	5.725282
0046bd062801670f	2.000000
a3ea100efff3ebb3	1.274307
cfe88105d142efdf	0.772670
d66ca8510f404cd0	0.709302
1210208ddc71c358	0.617136
330b74e32ee1d4e8	0.568462

The pivot table above shows the total amount of virtual currency our target wallet received from other wallets in descending order.

In [55]:
```
pivot1 = crypto_wallet.pivot_table(index=['received from'], values=['received amount'], aggfunc='sum').sort_values('received amount', ascending = False)[0:11].plot(kind= 'bar', colo
```



The bar graph above shows the top ten transactions with wallet fa4b2eaa533f1721 sending the most total virtual currency.

I also analyzed what wallets the target wallet sent virtual currency to.

In [56]:
```
pivot2 = crypto_wallet.pivot_table(index=['sent to'], values=['sent amount'], aggfunc='sum').sort_values('sent amount', ascending=False)
pivot2.reset_index
```

Out[56]:
```
<bound method DataFrame.reset_index of                   sent amount
sent to
2bd754d61ba9c764    17.909887
e465a8e2dbf482f9     8.725587
e43080b947a787e5     5.725378
ba76b5b717524a8c     4.530000
e868ab8ae74e2bd1     3.800000
7bc5125eea3e3506     3.300000
bbd14fd67a7cdcd9     3.000000
bfcf7fc21da02525     3.000000
30d69f9bab3f1f0c     1.274300
12a40d07d78860a8     0.340000
a233c7aa4da05e98     0.228349
0232534ab1671ecc     0.113450
ce6104e6701d8a87     0.090000
97e9e3496a1ecbaf     0.035747
(fee)                0.003485>
```

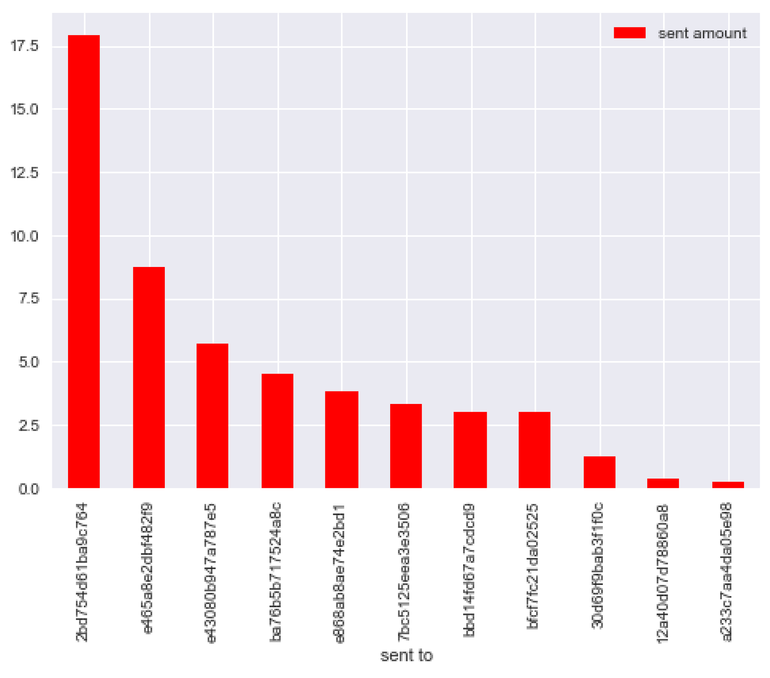
In [57]:
```
pivot2.head(10)
```

Out[57]:

sent to	sent amount
2bd754d61ba9c764	17.909887
e465a8e2dbf482f9	8.725587
e43080b947a787e5	5.725378
ba76b5b717524a8c	4.530000
e868ab8ae74e2bd1	3.800000
7bc5125eea3e3506	3.300000
bbd14fd67a7cdcd9	3.000000
bfcf7fc21da02525	3.000000
30d69f9bab3f1f0c	1.274300
12a40d07d78860a8	0.340000

The pivot table shows the amount of virtual currency sent to each wallet from the target wallet in descending order.

In [58]:
```
pivot2 = crypto_wallet.pivot_table(index=['sent to'], values=['sent amount'], aggfunc='sum').sort_values('sent amount', ascending = False)[0:11].plot(kind ='bar', color ='red')
```



The bar graph shows the top ten wallets that the target wallet sent virtual currency to. Wallet 2bd754d61ba9c764 received the most virtual currency from the target wallet.

4. Conclusion

Based on my analysis, I have found that the target wallet has had transactions with wallets associated with DarkNet Markets. The target wallet recived virtual currency from Nucleus Market or Alphay Bay Market within the thirteen transactions. The target wallet did not send virtual currency to wallets associated with DarkNet Markets.

The target wallet received the most virtual currency from wallet fa4b2eaa533f1721 and sent the most virtual currency to wallet 2bd754d61ba9c764.

Thank you for reading my notebook!!

Regards,

Julia Farson