Bitcoin Network Project

Comparing the Bitcoin Network activity from a day when the bitcoin price exploded with a day when the bitcoin price evolves normally

```
March 2021

The goal of this study is to analyse the activity on the
Blockchain with respect to the transactions involving Bitcoin on a
day on which a huge growth of the Bitcoin price was witnessed
compared to the activity on a day on which the price evolution is
more usual. We will therefore firstly identify the dates which
we'll be using from an analysis of the Bitcoin stock price. We
then carry out an analysis on these two days to identify the
differences in the Networks while paying particular attention to
the nodes with highest centrality and to the communities
existence.
```

In [312]:
```
#import librairies
import random
import networkx as nx
from statistics import mean
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import pandas as pd
import pandas_datareader as pdr
import numpy as np
import os
import pyarrow
from datetime import datetime
!pip install cdlib
import networkx.algorithms.community as nxcom
from cdlib import algorithms, viz, NodeClustering, evaluation
```

```
Requirement already satisfied: cdlib in c:\users\niksb\anaconda3\l
ib\site-packages (0.2.0)
Requirement already satisfied: nf1 in c:\users\niksb\anaconda3\lib
\site-packages (from cdlib) (0.0.3)
Requirement already satisfied: karateclub>=1.0.0 in c:\users\niksb
\anaconda3\lib\site-packages (from cdlib) (1.0.23)
Requirement already satisfied: tqdm>=4.20.* in c:\users\niksb\anac
onda3\lib\site-packages (from cdlib) (4.47.0)
Requirement already satisfied: seaborn>=0.9.* in c:\users\niksb\an
aconda3\lib\site-packages (from cdlib) (0.10.1)
Requirement already satisfied: networkx>=2.4 in c:\users\niksb\ana
conda3\lib\site-packages (from cdlib) (2.4)
Requirement already satisfied: bimlpa in c:\users\niksb\anaconda3\
lib\site-packages (from cdlib) (0.1.2)
Requirement already satisfied: pulp>=2.1 in c:\users\niksb\anacond
a3\lib\site-packages (from cdlib) (2.4)
```

```
Requirement already satisfied: numpy>=1.15.* in c:\users\niksb\ana
conda3\lib\site-packages (from cdlib) (1.18.5)
Requirement already satisfied: future>=0.16.* in c:\users\niksb\an
aconda3\lib\site-packages (from cdlib) (0.18.2)
Requirement already satisfied: pooch in c:\users\niksb\anaconda3\l
ib\site-packages (from cdlib) (1.3.0)
Requirement already satisfied: pandas>=0.24 in c:\users\niksb\anac
onda3\lib\site-packages (from cdlib) (1.0.5)
Requirement already satisfied: eva-lcd in c:\users\niksb\anaconda3
\lib\site-packages (from cdlib) (0.1.0)
Requirement already satisfied: markov-clustering in c:\users\niksb
\anaconda3\lib\site-packages (from cdlib) (0.0.6.dev0)
Requirement already satisfied: matplotlib>=3.0.* in c:\users\niksb
\anaconda3\lib\site-packages (from cdlib) (3.2.2)
Requirement already satisfied: scikit-learn<0.24,>=0.21.* in c:\us
ers\niksb\anaconda3\lib\site-packages (from cdlib) (0.23.1)
Requirement already satisfied: ASLPAw==2.0.0 in c:\users\niksb\ana
conda3\lib\site-packages (from cdlib) (2.0.0)
Requirement already satisfied: scipy>=1.3.* in c:\users\niksb\anac
onda3\lib\site-packages (from cdlib) (1.5.0)
Requirement already satisfied: omega-index-py3 in c:\users\niksb\a
naconda3\lib\site-packages (from cdlib) (0.3)
Requirement already satisfied: pquality==0.0.7 in c:\users\niksb\a
naconda3\lib\site-packages (from cdlib) (0.0.7)
Requirement already satisfied: shuffle-graph==1.1.1 in c:\users\ni
ksb\anaconda3\lib\site-packages (from cdlib) (1.1.1)
Requirement already satisfied: dynetx in c:\users\niksb\anaconda3\
lib\site-packages (from cdlib) (0.2.4)
Requirement already satisfied: demon>=2.0.5 in c:\users\niksb\anac
onda3\lib\site-packages (from cdlib) (2.0.5)
Requirement already satisfied: python-louvain==0.14 in c:\users\ni
ksb\anaconda3\lib\site-packages (from cdlib) (0.14)
Requirement already satisfied: chinese-whispers in c:\users\niksb\
anaconda3\lib\site-packages (from cdlib) (0.7.4)
Requirement already satisfied: gensim==3.8.3 in c:\users\niksb\ana
conda3\lib\site-packages (from karateclub>=1.0.0->cdlib) (3.8.3)
Requirement already satisfied: six in c:\users\niksb\anaconda3\lib
\site-packages (from karateclub>=1.0.0->cdlib) (1.15.0)
Requirement already satisfied: pygsp in c:\users\niksb\anaconda3\l
ib\site-packages (from karateclub>=1.0.0->cdlib) (0.5.1)
Requirement already satisfied: decorator>=4.3.0 in c:\users\niksb\
anaconda3\lib\site-packages (from networkx>=2.4->cdlib) (4.4.2)
Requirement already satisfied: amply>=0.1.2 in c:\users\niksb\anac
onda3\lib\site-packages (from pulp>=2.1->cdlib) (0.1.4)
Requirement already satisfied: packaging in c:\users\niksb\anacond
a3\lib\site-packages (from pooch->cdlib) (20.4)
Requirement already satisfied: requests in c:\users\niksb\anaconda
3\lib\site-packages (from pooch->cdlib) (2.24.0)
Requirement already satisfied: appdirs in c:\users\niksb\anaconda3
\lib\site-packages (from pooch->cdlib) (1.4.4)
Requirement already satisfied: pytz>=2017.2 in c:\users\niksb\anac
onda3\lib\site-packages (from pandas>=0.24->cdlib) (2020.1)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\
niksb\anaconda3\lib\site-packages (from pandas>=0.24->cdlib) (2.8.
1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\niksb
\anaconda3\lib\site-packages (from matplotlib>=3.0.*->cdlib) (1.2.
0)
Requirement already satisfied: cycler>=0.10 in c:\users\niksb\anac
onda3\lib\site-packages (from matplotlib>=3.0.*->cdlib) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=
2.0.1 in c:\users\niksb\anaconda3\lib\site-packages (from matplotl
ib>=3.0.*->cdlib) (2.4.7)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\ni
ksb\anaconda3\lib\site-packages (from scikit-learn<0.24,>=0.21.*->
cdlib) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\niksb\anac
onda3\lib\site-packages (from scikit-learn<0.24,>=0.21.*->cdlib) (
0.16.0)
Requirement already satisfied: multivalued-dict>=1.7.1 in c:\users
\niksb\anaconda3\lib\site-packages (from ASLPAw==2.0.0->cdlib) (2.
0.1)
Requirement already satisfied: count-dict>=1.0.1 in c:\users\niksb
\anaconda3\lib\site-packages (from ASLPAw==2.0.0->cdlib) (1.1.1)
Requirement already satisfied: Cython==0.29.14 in c:\users\niksb\a
naconda3\lib\site-packages (from gensim==3.8.3->karateclub>=1.0.0-
>cdlib) (0.29.14)
Requirement already satisfied: smart-open>=1.8.1 in c:\users\niksb
\anaconda3\lib\site-packages (from gensim==3.8.3->karateclub>=1.0.
0->cdlib) (4.2.0)
Requirement already satisfied: docutils>=0.3 in c:\users\niksb\ana
conda3\lib\site-packages (from amply>=0.1.2->pulp>=2.1->cdlib) (0.
16)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\niksb
\anaconda3\lib\site-packages (from requests->pooch->cdlib) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.
21.1 in c:\users\niksb\anaconda3\lib\site-packages (from requests-
>pooch->cdlib) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in c:\users\niksb\anac
onda3\lib\site-packages (from requests->pooch->cdlib) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\niks
b\anaconda3\lib\site-packages (from requests->pooch->cdlib) (2020.
6.20)
```

Selection of dates

In [314]:
```python
#Function to import BTC stock price from yahoo finance
def import_data(variables, name, start, end):
    # fetch a list of series
    indexlist = variables
    indexnames = name
    # define a start and end date
    starti = start
    endi = end
    dates = pd.date_range(starti,endi,freq='D')
    dfindex = pd.DataFrame(index=dates)
    for symboli in indexlist:
        stocki = pdr.get_data_yahoo(symboli, start=starti, end=endi
        stocki = pd.DataFrame(stocki['Adj Close'])  # take the clos
        stocki.columns = [symboli]
        dfindex = dfindex.join(stocki)
    dfindex.columns = name
    return dfindex
```

In [316]:
```python
#Importation of BTC stock price from yahoo finance from 28/04/2013
#And calculation of daily returns
btc=import_data(['BTC-USD'], ['BTC'], datetime(2013, 4, 28), dateti
btc['Return']=btc['BTC']-btc['BTC'].shift(1)
btc = btc.dropna(how='any')
btc=btc.astype(float)
btc
```

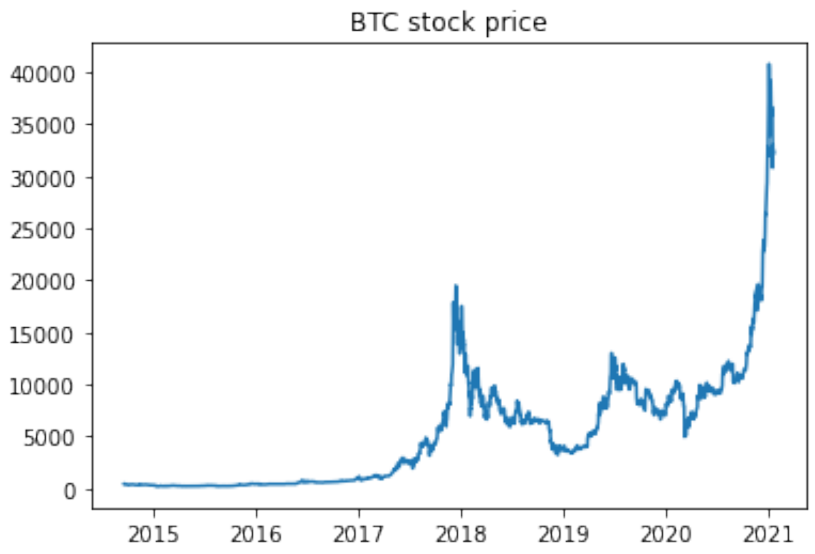
Out[316]:

	BTC	Return
2014-09-17	424.440002	-32.894012
2014-09-18	394.795990	-29.644012
2014-09-19	408.903992	14.108002
2014-09-20	398.821014	-10.082977
2014-09-21	402.152008	3.330994
...	...	...
2021-01-21	30825.699219	-4722.050781
2021-01-22	33005.761719	2180.062500
2021-01-23	32067.642578	-938.119141
2021-01-24	32289.378906	221.736328
2021-01-25	32366.392578	77.013672

2308 rows × 2 columns

```
In [317]: plt.title('BTC stock price')
          plt.plot(btc.index, btc.BTC)
```

```
Out[317]: [<matplotlib.lines.Line2D at 0x39b180ec2e0>]
```


BTC stock price

```
In [318]: #Sorting wrt daily return
          btc.sort_values('Return',ascending=False)
```

Out[318]:

	BTC	Return
2017-12-07	17899.699219	3608.199219
2021-01-13	37316.359375	3393.398438
2021-01-06	36824.363281	2831.933594
2021-01-02	32127.267578	2753.115234
2021-01-07	39371.042969	2546.679688
...	...	...
2018-01-16	11490.500000	-2329.299805
2021-01-15	36825.367188	-2361.960938
2021-01-11	35566.656250	-2789.785156
2020-03-12	4970.788086	-2940.642090
2021-01-21	30825.699219	-4722.050781

2308 rows × 2 columns

```
In [52]: max(btc['Return'])
```

```
Out[52]: 3608.19921875
```

```
In [53]: btc['Return'].idxmax()
```

```
Out[53]: Timestamp('2017-12-07 00:00:00')
```

The highest BTC stock price increase during the considered time period was observed from the 6th December 2020 to the 7th December 2020. We will therefore be studying the transactions occuring the the 6th which led to such an important increase.

In [56]:
```
#We now identify a date close to the above selection(from 01/11/201
#on which the BTC price evolution is the closest to the mean daily
start_date = "2017-11-1"

end_date = "2017-12-31"


after_start_date = btc.index >= start_date

before_end_date = btc.index <= end_date

between_two_dates = after_start_date & before_end_date
btc1= btc.loc[between_two_dates]
btc1["selection"]=btc1["Return"]-mean(btc["Return"])
print(btc1.to_string())
```

```
                     BTC      Return   selection
2017-11-01   6767.310059  298.910156  285.140789
2017-11-02   7078.500000  311.189941  297.420574
2017-11-03   7207.759766  129.259766  115.490399
2017-11-04   7379.950195  172.190430  158.421063
2017-11-05   7407.410156   27.459961   13.690594
2017-11-06   7022.759766 -384.650391 -398.419758
2017-11-07   7144.379883  121.620117  107.850750
2017-11-08   7459.689941  315.310059  301.540692
2017-11-09   7143.580078 -316.109863 -329.879230
2017-11-10   6618.140137 -525.439941 -539.209308
2017-11-11   6357.600098 -260.540039 -274.309406
2017-11-12   5950.069824 -407.530273 -421.299640
2017-11-13   6559.490234  609.420410  595.651043
2017-11-14   6635.750000   76.259766   62.490399
2017-11-15   7315.540039  679.790039  666.020672
2017-11-16   7871.689941  556.149902  542.380535
2017-11-17   7708.990234 -162.699707 -176.469074
2017-11-18   7790.149902   81.159668   67.390301
2017-11-19   8036.490234  246.340332  232.570965
2017-11-20   8200.639648  164.149414  150.380047
2017-11-21   8071.259766 -129.379883 -143.149250
2017-11-22   8253.549805  182.290039  168.520672
2017-11-23   8038.770020 -214.779785 -228.549152
2017-11-24   8253.690430  214.920410  201.151043
2017-11-25   8790.919922  537.229492  523.460125
2017-11-26   9330.549805  539.629883  525.860516
2017-11-27   9818.349609  487.799805  474.030438
2017-11-28  10058.799805  240.450195  226.680828
2017-11-29   9888.610352 -170.189453 -183.958820
2017-11-30  10233.599609  344.989258  331.219891
2017-12-01  10975.599609  742.000000  728.230633
2017-12-02  11074.599609   99.000000   85.230633
2017-12-03  11323.200195  248.600586  234.831219
2017-12-04  11657.200195  334.000000  320.230633
```

```
2017-12-05  11916.700195   259.500000   245.730633
2017-12-06  14291.500000  2374.799805  2361.030438
2017-12-07  17899.699219  3608.199219  3594.429852
2017-12-08  16569.400391 -1330.298828 -1344.068195
2017-12-09  15178.200195 -1391.200195 -1404.969562
2017-12-10  15455.400391   277.200195   263.430828
2017-12-11  16936.800781  1481.400391  1467.631024
2017-12-12  17415.400391   478.599609   464.830242
2017-12-13  16408.199219 -1007.201172 -1020.970539
2017-12-14  16564.000000   155.800781   142.031414
2017-12-15  17706.900391  1142.900391  1129.131024
2017-12-16  19497.400391  1790.500000  1776.730633
2017-12-17  19140.800781  -356.599609  -370.368976
2017-12-18  19114.199219   -26.601562   -40.370929
2017-12-19  17776.699219 -1337.500000 -1351.269367
2017-12-20  16624.599609 -1152.099609 -1165.868976
2017-12-21  15802.900391  -821.699219  -835.468586
2017-12-22  13831.799805 -1971.100586 -1984.869953
2017-12-23  14699.200195   867.400391   853.631024
2017-12-24  13925.799805  -773.400391  -787.169758
2017-12-25  14026.599609   100.799805    87.030438
2017-12-26  16099.799805  2073.200195  2059.430828
2017-12-27  15838.500000  -261.299805  -275.069172
2017-12-28  14606.500000 -1232.000000 -1245.769367
2017-12-29  14656.200195    49.700195    35.930828
2017-12-30  12952.200195 -1704.000000 -1717.769367
2017-12-31  14156.400391  1204.200195  1190.430828

<ipython-input-56-9f45d4c634a7>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  btc1["selection"]=btc1["Return"]-mean(btc["Return"])
```

In [40]:
```
mean(btc.Return)
```

Out[40]:
```
19.990898994606454
```

In [59]:
```
abs(btc1['selection']).idxmin()
```

Out[59]:
```
Timestamp('2017-11-05 00:00:00')
```

The BTC price increase which is closest to the average daily growth of 19.99 USD is observed from the 4th of November 2017 to the 5th November. We will therefore be studying the transactions occuring on the 4th of November.

Bitcoin Network Analysis

```
In [328]: #Import data
          df1=pd.read_parquet('C:/Users/niksb/Desktop/Bitcoin Network project
          df2=pd.read_parquet('C:/Users/niksb/Desktop/Bitcoin Network project
```

```
In [329]: df1['time'] = pd.to_datetime(df1['time'], unit='s')
          df2['time'] = pd.to_datetime(df2['time'], unit='s')
```

Type of senders and destinators

```
In [330]: #Filter out the self_spending users
          def filter_out_self_spending(df):
              return df[df["src_identity"]!=df["dst_identity"]]
          df1=filter_out_self_spending(df1)
          df2=filter_out_self_spending(df2)
```

```
In [331]: #Identifying the different type of users between : Ukwn_btc_add,"Uk
          conditions = [(df1["src_identity"].str.len()>33),(df1["src_identity
          values = ["Ukwn_btc_add", "Ukwn_wallet"]
          df1['sender_status'] = np.select(conditions, values)
          df1['sender_status']=df1['sender_status'].replace("0","kwn_wallet")

          conditions = [(df2["src_identity"].str.len()>33),(df2["src_identity
          df2['sender_status'] = np.select(conditions, values)
          df2['sender_status']=df2['sender_status'].replace("0","kwn_wallet")

          conditions = [
              (df1["dst_identity"].str.len()>33),
              (df1["dst_identity"].astype(str).str.isnumeric())]
          values = ["Ukwn_btc_add", "Ukwn_wallet"]
          df1['dest_status'] = np.select(conditions, values)
          df1['dest_status']=df1['dest_status'].replace("0","kwn_wallet")

          conditions = [
              (df2["dst_identity"].str.len()>33),
              (df2["dst_identity"].astype(str).str.isnumeric())]
          values = ["Ukwn_btc_add", "Ukwn_wallet"]
          df2['dest_status'] = np.select(conditions, values)
          df2['dest_status']=df2['dest_status'].replace("0","kwn_wallet")
```

Firstly, since we are interested in the transactions between different users, we filter out the self-spending users. Secondly, we identify the different types of senders and destinators amongst:
(i)A bitcoin address(if the user is unknown)-Ukwn_btc_add
(ii)An unknown wallet represented as an integer-Ukwn_wallet
(iii)A wallet corresponding to a known actor from walletexplorer.com-kwn_wallet

In [332]:
```
df1
```

Out[332]:

	value	time	src_identity	
0	176039	2017-11-04 12:33:16	17kzqdHjt3zKvZVDwxsh6WFPdg9vuu1MLh	
24	705855944	2017-11-04 15:57:56	68272571	
39	2328424	2017-11-04 11:44:41	1G6kncsk2CsTynQbWJX8QExfHL7LwYbLVy	
40	5185249	2017-11-04 01:47:15	1AA5gAknu9ePx3DCgtMQzCpjCz6Vb5Hbek	
41	2817172	2017-11-04 09:43:16	19YP1uNDphq3ymCwAG3G3jVzYxixEiDPJG	
...	...	...	...	
738769	0	2017-11-04 16:40:41	252	nonstandardcfcb1eae
738770	0	2017-11-04 21:24:55	8452317	nonstandarde115ad0
738771	0	2017-11-04 21:24:55	8452317	nonstandarde77e0a98
738772	0	2017-11-04 21:24:55	8452317	nonstandarde99d366
738773	0	2017-11-04 06:14:07	1NmiU6rTBhTojDjoMhUKshG2Dp4LF94e31	nonstandardf0f569d5

662789 rows × 7 columns

In [333]:
```
df2
```
Out[333]:

	value	time	src_identity	
0	253473	2017-12-06 03:42:49	1DWSL1jynT3TzXHTcaJaAmbp8YzQQpzvg9	
1	10871222	2017-12-06 16:49:19	16241514	
2	985000	2017-12-06 12:49:45	17YZiyEqZqFDjE2JfCUWYykiNHrnbPFeKC	
3	52351723	2017-12-06 22:13:42	Poloniex.com	
8	12285504	2017-12-06 06:38:52	41538099	
...	...	...	...	
1188925	0	2017-12-06 07:41:44	1DLBb5aPVmnrbUGTF52yWQHDvqYu3sTnWJ	nonstandardefa3c367
1188926	0	2017-12-06 12:39:49	8452317	nonstandardf74fad4
1188927	0	2017-12-06 12:39:49	8452317	nonstandardf7609f31
1188928	0	2017-12-06 08:45:52	8452317	nonstandardfdb34f1
1188929	0	2017-12-06 12:39:49	8452317	nonstandardfee22bb

1074306 rows × 7 columns

In [334]:
```
print("4 Nov destinator destription:\n" +str(df1["dest_status"].valu
print()
print("6 Dec destinator destription:\n" +str(df2["dest_status"].valu
```
```
4 Nov destinator destription:
Ukwn_wallet     422509
Ukwn_btc_add    181087
kwn_wallet       59193
Name: dest_status, dtype: int64

6 Dec destinator destription:
Ukwn_wallet     649639
Ukwn_btc_add    310027
kwn_wallet      114640
Name: dest_status, dtype: int64
```

In [335]:
```
print("4 Nov sender destription:\n" +str(df1["sender_status"].value
print()
print("6 Dec sender destription:\n" +str(df2["sender_status"].value
```

```
4 Nov sender destription:
Ukwn_btc_add    367308
Ukwn_wallet     247433
kwn_wallet       48048
Name: sender_status, dtype: int64

6 Dec sender destription:
Ukwn_btc_add    610715
Ukwn_wallet     362625
kwn_wallet      100966
Name: sender_status, dtype: int64
```

We can observe that there are more actors from all 3 fields on the 6th of December but this seems to be solely due to the increased amount of transactions. The proportion of different type of actors remain very similar.

Distribution of BTC amounts involved in transactions

In [336]:
```
#Calculating bitcoin value of transactions
bitcoinquantity1 = df1['value']*0.00000001
df1['bitcoinquantity'] = bitcoinquantity1

bitcoinquantity2 = df2['value']*0.00000001
df2['bitcoinquantity'] = bitcoinquantity2
```

```
In [337]: df1
```

```
Out[337]:
```

	value	time	src_identity	
0	176039	2017-11-04 12:33:16	17kzqdHjt3zKvZVDwxsh6WFPdg9vuu1MLh	
24	705855944	2017-11-04 15:57:56	68272571	
39	2328424	2017-11-04 11:44:41	1G6kncsk2CsTynQbWJX8QExfHL7LwYbLVy	
40	5185249	2017-11-04 01:47:15	1AA5gAknu9ePx3DCgtMQzCpjCz6Vb5Hbek	
41	2817172	2017-11-04 09:43:16	19YP1uNDphq3ymCwAG3G3jVzYxixEiDPJG	
...	...	...	...	
738769	0	2017-11-04 16:40:41	252	nonstandardcfcb1eae
738770	0	2017-11-04 21:24:55	8452317	nonstandarde115ad0
738771	0	2017-11-04 21:24:55	8452317	nonstandarde77e0a98
738772	0	2017-11-04 21:24:55	8452317	nonstandarde99d366
738773	0	2017-11-04 06:14:07	1NmiU6rTBhTojDjoMhUKshG2Dp4LF94e31	nonstandardf0f569d5

662789 rows × 8 columns

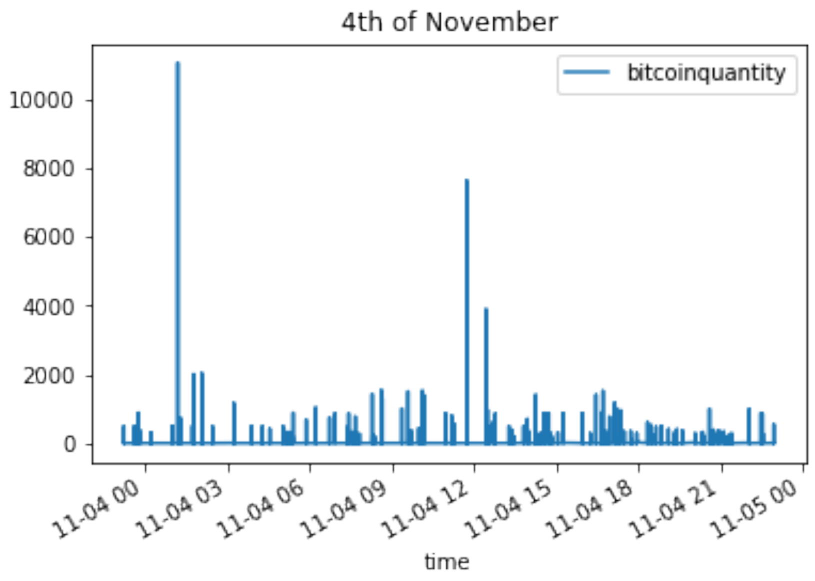
In [338]:
```
df2
```
Out[338]:

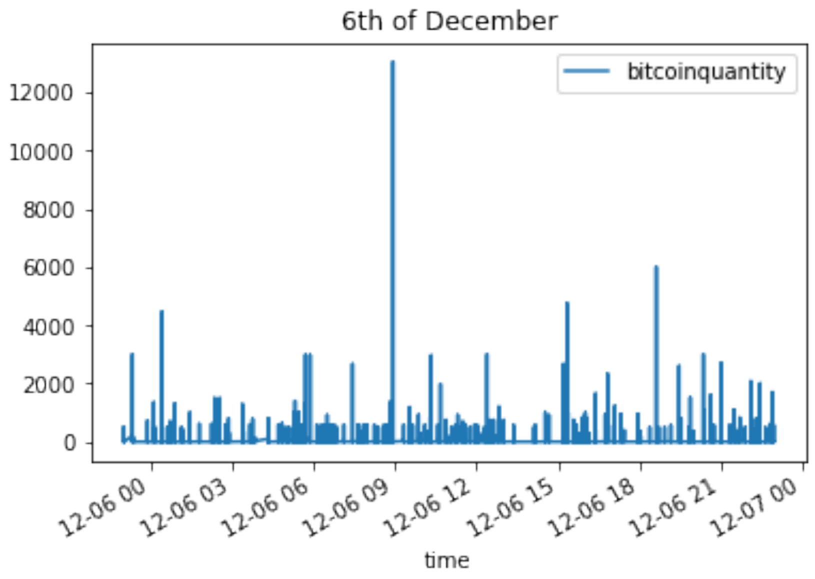
	value	time	src_identity	
0	253473	2017-12-06 03:42:49	1DWSL1jynT3TzXHTcaJaAmbp8YzQQpzvg9	
1	10871222	2017-12-06 16:49:19	16241514	
2	985000	2017-12-06 12:49:45	17YZiyEqZqFDjE2JfCUWYykiNHrnbPFeKC	
3	52351723	2017-12-06 22:13:42	Poloniex.com	
8	12285504	2017-12-06 06:38:52	41538099	
...	...	...	...	
1188925	0	2017-12-06 07:41:44	1DLBb5aPVmnrbUGTF52yWQHDvqYu3sTnWJ	nonstandardefa3c367
1188926	0	2017-12-06 12:39:49	8452317	nonstandardf74fad4
1188927	0	2017-12-06 12:39:49	8452317	nonstandardf7609f31
1188928	0	2017-12-06 08:45:52	8452317	nonstandardfdb34f1
1188929	0	2017-12-06 12:39:49	8452317	nonstandardfee22b

1074306 rows × 8 columns

```
In [343]: df1.plot(x="time", y = "bitcoinquantity",title='4th of November')
          df2.plot(x="time", y = "bitcoinquantity",title='6th of December')
```

```
Out[343]: <matplotlib.axes._subplots.AxesSubplot at 0x39b160eb280>
```





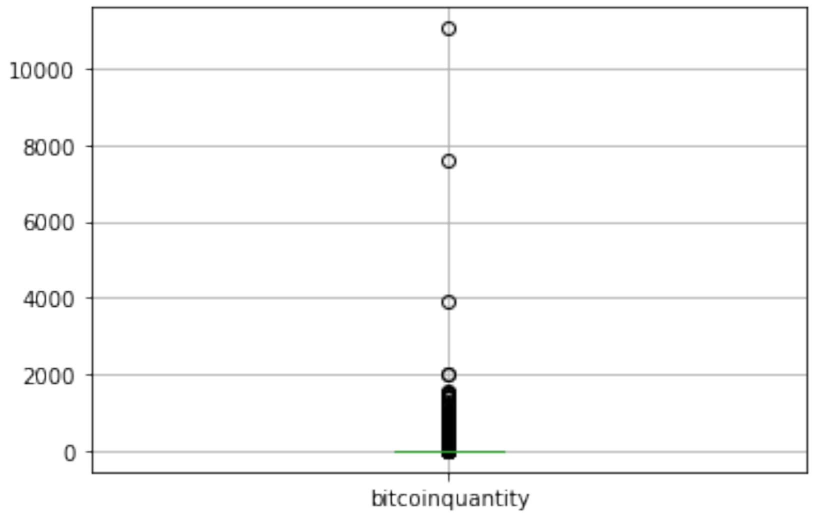
We can observe that on the 4th of November there are peaks in the amount of Bitcoins transferred at times close to midnight and noon while on the 6th of December, the 2 main peaks occur at 9 am and 6 pm.

```
In [344]: df1["bitcoinquantity"].describe()
```

```
Out[344]: count    662789.000000
          mean          1.608640
          std          25.305746
          min           0.000000
          25%           0.002674
          50%           0.013600
          75%           0.092560
          max       11049.827067
          Name: bitcoinquantity, dtype: float64
```

```
In [347]: df1.boxplot(column="bitcoinquantity")
```

```
Out[347]: <matplotlib.axes._subplots.AxesSubplot at 0x28df80a93d0>
```

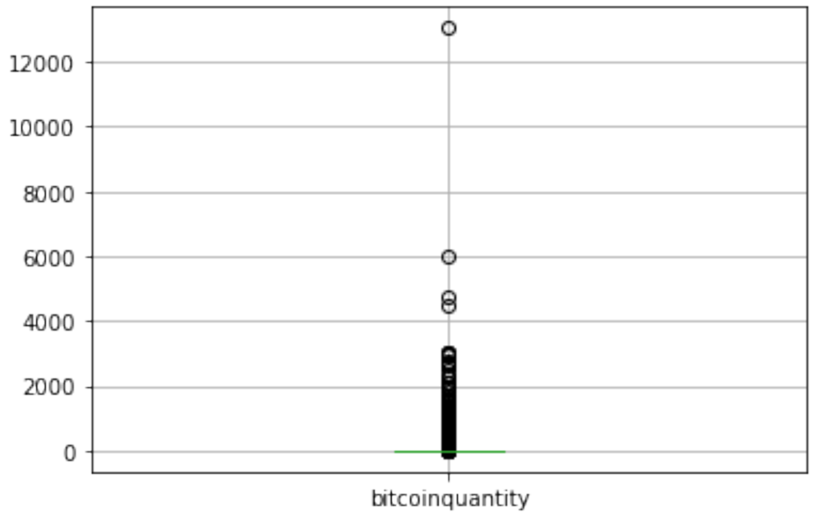


```
In [242]: df2["bitcoinquantity"].describe()
```

```
Out[242]: count    1.074306e+06
          mean     2.394909e+00
          std      2.845245e+01
          min      0.000000e+00
          25%      3.416800e-03
          50%      1.650000e-02
          75%      1.249998e-01
          max      1.304177e+04
          Name: bitcoinquantity, dtype: float64
```

```
In [348]: df2.boxplot(column="bitcoinquantity")
```

```
Out[348]: <matplotlib.axes._subplots.AxesSubplot at 0x39b160ea520>
```



From the above boxplots and statistics, we can observe that the big majority of the transactions are lower than 1 BTC with the third quartile for both days being lower than 1 BTC. The mean of the BTC transactions on the 6th of December(2.39 BTC) is though significantly higher than the one for the 4th of November(1.61 BTC). So is the case for the median, 3rd quartile and maximum BTC transaction value. We can therefore safely say that the amount of BTC involved in the transactions on the 6th of December were generally higher than that on the 4th of November.

In [349]:
```
dg1=df1[['src_identity','dst_identity']]
dg2=df2[['src_identity','dst_identity']]
```

In [350]:
```
#Directed and undirected graphs for the 2 dates
g10 = nx.from_pandas_edgelist(dg1, 'src_identity', 'dst_identity',c
g11 = nx.from_pandas_edgelist(dg1, 'src_identity', 'dst_identity',c
g20 = nx.from_pandas_edgelist(dg2, 'src_identity', 'dst_identity',c
g21 = nx.from_pandas_edgelist(dg2, 'src_identity', 'dst_identity',c
```

Network Statistics

In [351]:
```
print ("number of nodes for the 4th November 2017: "+str(len(g10.no
print ("number of edges for the 4th November 2017: "+str(len(g10.ed
print ("density for the 4th November 2017: "+str(nx.density(g10)))
print ("avg degree for the 4th November 2017: "+str(mean([n for n i
print ("clustering coefficient for the 4th November 2017: "+str(nx.
```

```
number of nodes for the 4th November 2017: 380061
number of edges for the 4th November 2017: 546729
density for the 4th November 2017: 3.785006589639088e-06
avg degree for the 4th November 2017: 2.8770592089164633
clustering coefficient for the 4th November 2017: 0.00252745199613
81078
```

In [352]:
```
print ("number of nodes for the 6th December 2017: "+str(len(g20.no
print ("number of edges for the 6th December 2017: "+str(len(g20.ed
print ("density for the 6th December 2017: "+str(nx.density(g20)))
print ("avg degree for the 6th December 2017: "+str(mean([n for n i
print ("clustering coefficient for the 6th December 2017: "+str(nx.
```

```
number of nodes for the 6th December 2017: 568451
number of edges for the 6th December 2017: 852206
density for the 6th December 2017: 2.6372984644654896e-06
avg degree for the 6th December 2017: 2.998344624250815
clustering coefficient for the 6th December 2017: 0.00145582359242
00323
```

From the above statistics, we can observe that there is a much higher number transactions(edges) on the 6th of December compared to the 4th of November(852 206 VS 546 729) and so is the case for the number of actors(nodes) involved in these transactions(568 451 VS 380 061). The average degree for the 6th of December is higher than that of 4 Nov which is usually synonymous with greater network size. The density is higher for 4 Nov and hence, these results are consistent with the findings of Leskovec(2006) since he observed that when a graph increases in size, then the average degree increases while density decreases.The higher global clustering coefficient for the 4th of November can also be explained by the higher density of this Network.

Identifying Nodes of Highest Centrality

In [357]:
```
leaderboard = {}
for x in g10.nodes:
    leaderboard[x] = len(g10[x])

s = pd.Series(leaderboard, name='edges')
dc1 = s.to_frame().sort_values('edges', ascending=False)
dc1
```

Out[357]:
```
                                                      edges
                                                  0    4400
                                        Bittrex.com    4191
                                       Poloniex.com    2652
                                            8452317    2270
                                                 20    2057
                                                ...     ...
                 19joXisRSGEA7au6KVQFtuPRBbJ3cdz1d1       0
                 13MTiLZQapJprUGVEgPN3CAXcKbGfuwFbF       0
                                           66895027       0
                 13N78oVQeTV8sVHhG6A3MbCzFWuRE4jT5h       0
nonstandardf0f569d5d62d680a6e245f4376d5723ccec6e9b9       0

380061 rows × 1 columns
```

In [360]:
```
#Since it is a directed network
print("nodes of highest degree 4th November 2017:")
sorted(nx.degree(g10), key=lambda x: x[1],  reverse=True)[:10]
```

```
nodes of highest degree 4th November 2017:
```

Out[360]:
```
[('0', 18714),
 ('Bittrex.com', 13216),
 ('141', 6538),
 ('66', 6346),
 ('Poloniex.com', 5201),
 ('4', 4678),
 ('CoinPayments.net', 4024),
 ('20', 3323),
 ('ePay.info', 3270),
 ('39', 3202)]
```

In [361]:
```
leaderboard = {}
for x in g20.nodes:
    leaderboard[x] = len(g20[x])

s = pd.Series(leaderboard, name='edges')
dc2 = s.to_frame().sort_values('edges', ascending=False)
dc2
```

Out[361]:

	edges
Bittrex.com	8952
8452317	8219
0	5118
Poloniex.com	5036
Bitstamp.net	4387
...	...
16tBu3gpVi9LrnckfLdYtU2mgr3nWbAJVf	0
23333431	0
16uh9vDVmyXYXgJ41DbKXwM83eXk46QeR7	0
16vAS7v6kHsdW4L77wbjT5rPuEmTArZHfU	0
nonstandardfee22bb1686fe7bd4cbeed3f4df4b6a9d610f165	0

568451 rows × 1 columns

In [362]:
```
print("nodes of highest degree 6th December 2017:")
sorted(nx.degree(g20), key=lambda x: x[1],  reverse=True)[:10]
```

```
nodes of highest degree 6th December 2017:
```

Out[362]:
```
[('Bittrex.com', 36898),
 ('39', 31497),
 ('0', 24113),
 ('8', 23126),
 ('Poloniex.com', 10453),
 ('296', 9676),
 ('8452317', 8219),
 ('198', 7728),
 ('Bitstamp.net', 6459),
 ('66', 5773)]
```

We can observe that there are a number of main actors which appear on both dates. These include: Bittrex, Poloniex and unknown wallets 0 and 8452317. The number of transactions Bittrex participated in significantly increased from 4191 to 8952. It is also generally observed for the other main actors that the number of transactions they were involved in on the 6 Dec is much higher than for the 4 Nov.

Focus on the 1000 nodes with highest degree

In [363]:
```
#Keeping the 1000 nodes with highest degree
smallestDegree1=[x[0] for x in sorted(g10.degree, key=lambda x: x[1
g10.remove_nodes_from(smallestDegree1)
smallestDegree2=[x[0] for x in sorted(g11.degree, key=lambda x: x[1
g11.remove_nodes_from(smallestDegree1)
smallestDegree3=[x[0] for x in sorted(g20.degree, key=lambda x: x[1
g20.remove_nodes_from(smallestDegree3)
smallestDegree4=[x[0] for x in sorted(g21.degree, key=lambda x: x[1
g21.remove_nodes_from(smallestDegree4)
```

In [311]:
```
#Exporting to gephi
#nx.write_gexf(g10, 'nikhi10.gexf')
#nx.write_gexf(g20, 'nikhi20.gexf')
```

In [364]:
```
#Directed network 4 Nov
print ("number of nodes for the 4th November 2017: "+str(len(g10.no
print ("number of edges for the 4th November 2017: "+str(len(g10.ed
print ("density for the 4th November 2017: "+str(nx.density(g10)))
print ("avg degree for the 4th November 2017: "+str(mean([n for n i
print ("clustering coefficient for the 4th November 2017: "+str(nx.
```

```
number of nodes for the 4th November 2017: 1000
number of edges for the 4th November 2017: 14137
density for the 4th November 2017: 0.014151151151151151
avg degree for the 4th November 2017: 28.274
clustering coefficient for the 4th November 2017: 0.13698311972112
6
```

In [372]:
```
#Undirected network 4 Nov
print ("number of nodes for the 4th November 2017: "+str(len(g11.no
print ("number of edges for the 4th November 2017: "+str(len(g11.ed
print ("density for the 4th November 2017: "+str(nx.density(g11)))
print ("avg degree for the 4th November 2017: "+str(mean([n for n i
print ("clustering coefficient for the 4th November 2017: "+str(nx.
```

```
number of nodes for the 4th November 2017: 1000
number of edges for the 4th November 2017: 13230
density for the 4th November 2017: 0.026486486486486487
avg degree for the 4th November 2017: 26.46
clustering coefficient for the 4th November 2017: 0.16110333551340
747
```

In [365]:
```
#Directed network 6 Dec
print ("number of nodes for the 6th December 2017: "+str(len(g20.no
print ("number of edges for the 6th December 2017: "+str(len(g20.ed
print ("density for the 6th December 2017: "+str(nx.density(g20)))
print ("avg degree for the 6th December 2017: "+str(mean([n for n i
print ("clustering coefficient for the 6th December 2017: "+str(nx.
```

```
number of nodes for the 6th December 2017: 1000
number of edges for the 6th December 2017: 19725
density for the 6th December 2017: 0.019744744744744743
avg degree for the 6th December 2017: 39.45
clustering coefficient for the 6th December 2017: 0.16143702060423
773
```

In [373]:
```
#Undirected Network 6 Dec
print ("number of nodes for the 6th December 2017: "+str(len(g21.no
print ("number of edges for the 6th December 2017: "+str(len(g21.ed
print ("density for the 6th December 2017: "+str(nx.density(g21)))
print ("avg degree for the 6th December 2017: "+str(mean([n for n i
print ("clustering coefficient for the 6th December 2017: "+str(nx.
```

```
number of nodes for the 6th December 2017: 1000
number of edges for the 6th December 2017: 18661
density for the 6th December 2017: 0.037359359359359356
avg degree for the 6th December 2017: 37.322
clustering coefficient for the 6th December 2017: 0.17632517122398
023
```

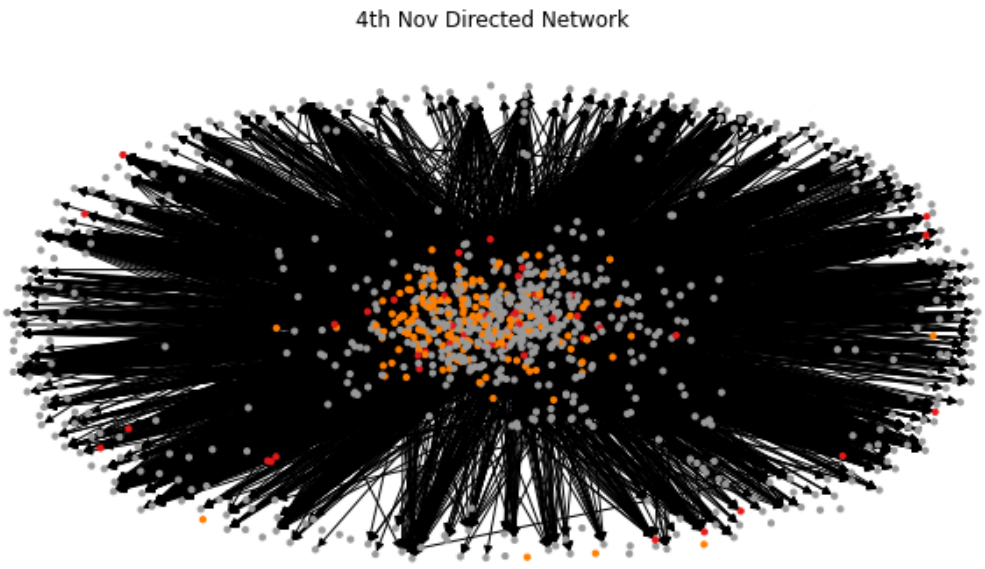
When focusing on the 1000 nodes of highest degree for the directed networks, we can see that the density as well as the global clustering coefficient for the 6 Dec is higher than that for the 4 Nov while this was not the case when we considered the complete networks. This indicates that for the 6 Dec, there is a higher concentration of connections between the main actors than it is the case for the 4 Nov.
If we compare the directed and undirected networks of the respective dates, we observe that the density and global clustering coefficient for the undirected graphs are higher than those of the directed networks.

In [366]:
```
cc1 = g11.subgraph(sorted(nx.connected_components(g11), key=len, re
print("average shortest path 4th November 2017: "+str(nx.average_sh

cc2 = g21.subgraph(sorted(nx.connected_components(g21), key=len, re
print("average shortest path 6th December 2017: "+str(nx.average_sh
```

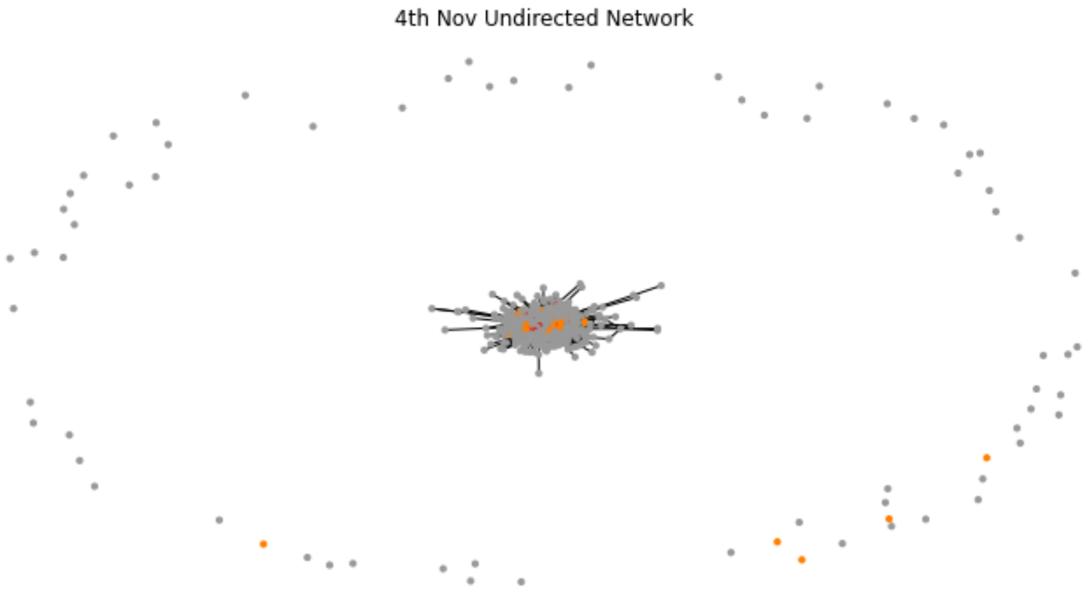
```
average shortest path 4th November 2017: 2.2654700245339594
average shortest path 6th December 2017: 2.1569291281421794
```

In [367]:
```
#Defining the user types so as to include graph colouring
cara10=pd.DataFrame({ 'id':g10.nodes})
conditions = [(cara10["id"].str.len()>33),(cara10["id"].astype(str)
values = [1,2]
cara10['node_status'] = np.select(conditions, values)
cara10['node_status']=cara10['node_status'].replace("0",3)
cara10= cara10.set_index('id')


cara11=pd.DataFrame({ 'id':g11.nodes})
conditions = [(cara11["id"].str.len()>33),(cara11["id"].astype(str)
values = [1,2]
cara11['node_status'] = np.select(conditions, values)
cara11['node_status']=cara11['node_status'].replace("0",3)
cara11= cara11.set_index('id')


cara20=pd.DataFrame({ 'id':g20.nodes})
conditions = [(cara20["id"].str.len()>33),(cara20["id"].astype(str)
values = [1,2]
cara20['node_status'] = np.select(conditions, values)
cara20['node_status']=cara20['node_status'].replace("0",3)
cara20= cara20.set_index('id')


cara21=pd.DataFrame({ 'id':g21.nodes})
conditions = [(cara21["id"].str.len()>33),(cara21["id"].astype(str)
values = [1,2]
cara21['node_status'] = np.select(conditions, values)
cara21['node_status']=cara21['node_status'].replace("0",3)
cara21= cara21.set_index('id')
```

```
In [374]: plt.figure(1,figsize=(12,6))
          plt.title('4th Nov Directed Network')
          nx.draw(g10, with_labels=False, node_color=cara10['node_status'], c
          #Color code kwn_wallet : Red, Ukwn_wallet : Grey, Ukwn_btc_add: Ora
```

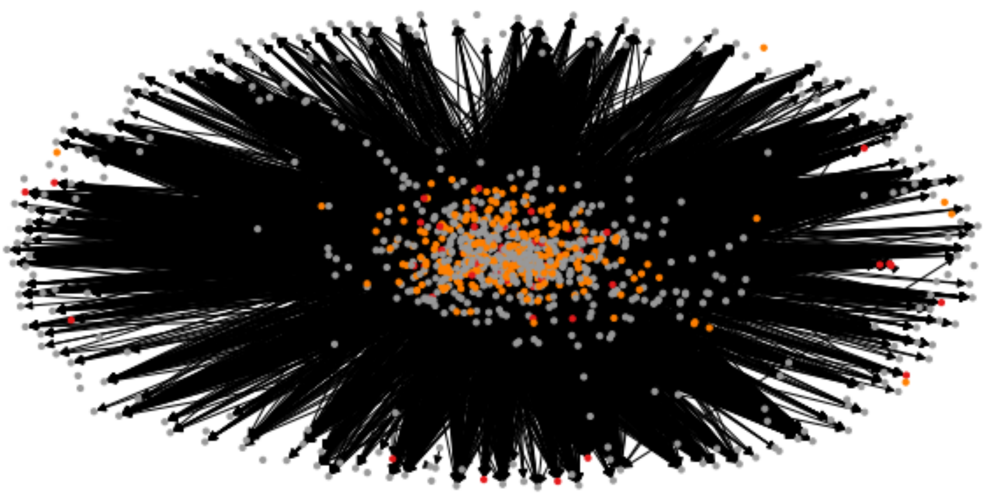


```
In [375]: plt.figure(1,figsize=(12,6))
          plt.title('4th Nov Undirected Network')
          nx.draw(g11, with_labels=False, node_color=cara11['node_status'], c
```



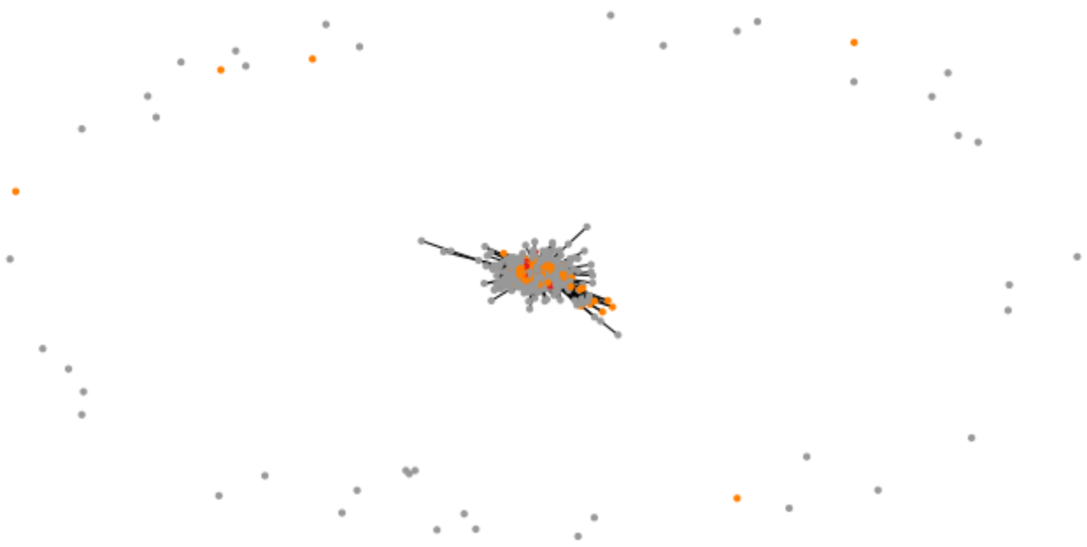
```
In [376]: plt.figure(1,figsize=(12,6))
          plt.title('6th Dec Directed Network')
          nx.draw(g20, with_labels=False, node_color=cara20['node_status'], cr
```

6th Dec Directed Network



```
In [377]: plt.figure(1,figsize=(12,6))
          plt.title('6th Dec Undirected Network')
          nx.draw(g21, with_labels=False, node_color=cara21['node_status'], cr
```

6th Dec Undirected Network



From the directed networks graphs, we can visually detect the higher concentration of main actors for the 6th of December. We can also observe a more frequent presence of the orange nodes which correspond to the unknown actors referred by BTC addresses for the 6 Dec than for 4 Nov.
For both dates, as we detected before, we can now easily observe that the undirected networks are indeed much more dense and clustered than their respective directed networks.

Community Detection from undirected networks

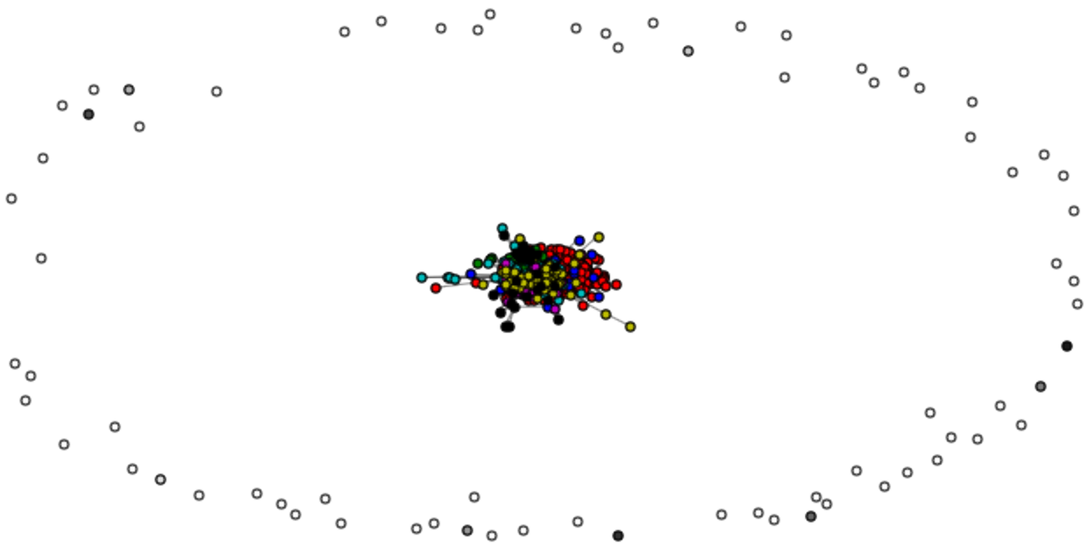
In [378]:
```
communities1 = algorithms.louvain(g11)
communities2 = algorithms.louvain(g21)
```

In [379]:
```
com1 = sorted(nxcom.greedy_modularity_communities(g11), key=len, re
com2 = sorted(nxcom.greedy_modularity_communities(g21), key=len, re
print(f"On the 4th November the network has {len(com1)} communities
print(f"On the 6th December the network has {len(com2)} communities
```

```
On the 4th November the network has 82 communities.
On the 6th December the network has 49 communities.
```

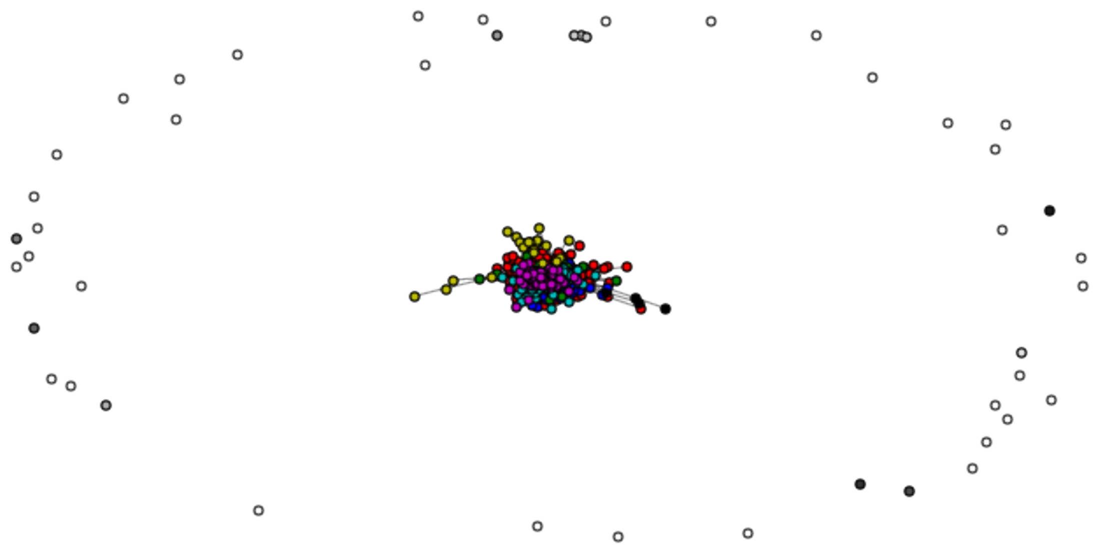
In [382]:
```
viz.plot_network_clusters(g11,partition=communities1,figsize=(12,6)
```

Out[382]:
```
<matplotlib.collections.PathCollection at 0x39b1d6b4730>
```



```
In [383]: viz.plot_network_clusters(g21,partition=communities2,figsize=(12,6)
```

```
Out[383]: <matplotlib.collections.PathCollection at 0x180e20b3460>
```



We are able to detect the presence of 33 more communities for 4 Nov than for 6 Dec. However, the density and global clustering coefficient for 6 Dec were both higher. Therefore, these results may suggest that there is the presence of larger communities i.e with more nodes on the 6 Dec.

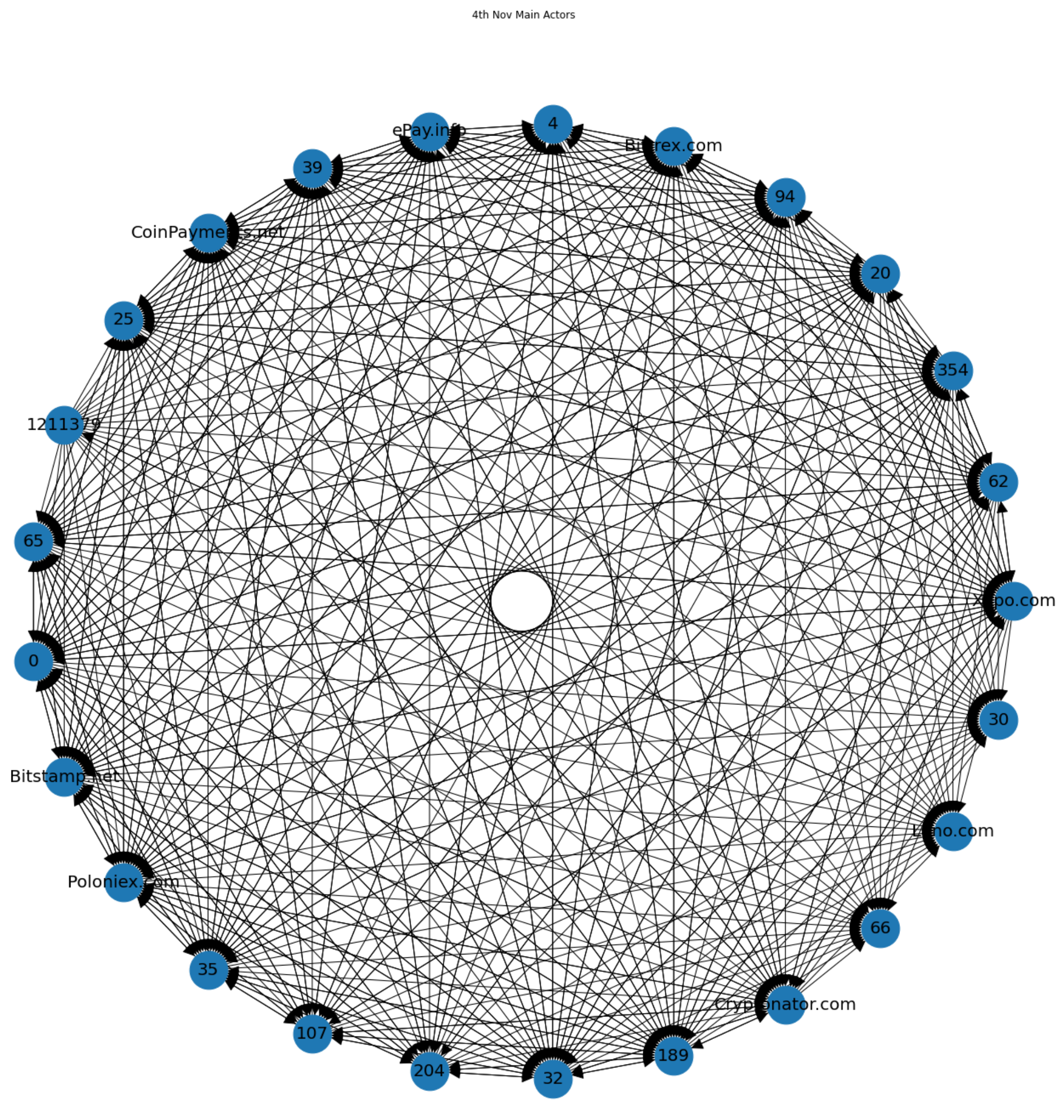
Focus on the main actors

```
In [384]: #Keeping the 25 nodes with highest degree
smallestDegree1=[x[0] for x in sorted(g10.degree, key=lambda x: x[1
g10.remove_nodes_from(smallestDegree1)
smallestDegree2=[x[0] for x in sorted(g11.degree, key=lambda x: x[1
g11.remove_nodes_from(smallestDegree1)
smallestDegree3=[x[0] for x in sorted(g20.degree, key=lambda x: x[1
g20.remove_nodes_from(smallestDegree3)
smallestDegree4=[x[0] for x in sorted(g21.degree, key=lambda x: x[1
g21.remove_nodes_from(smallestDegree4)
```

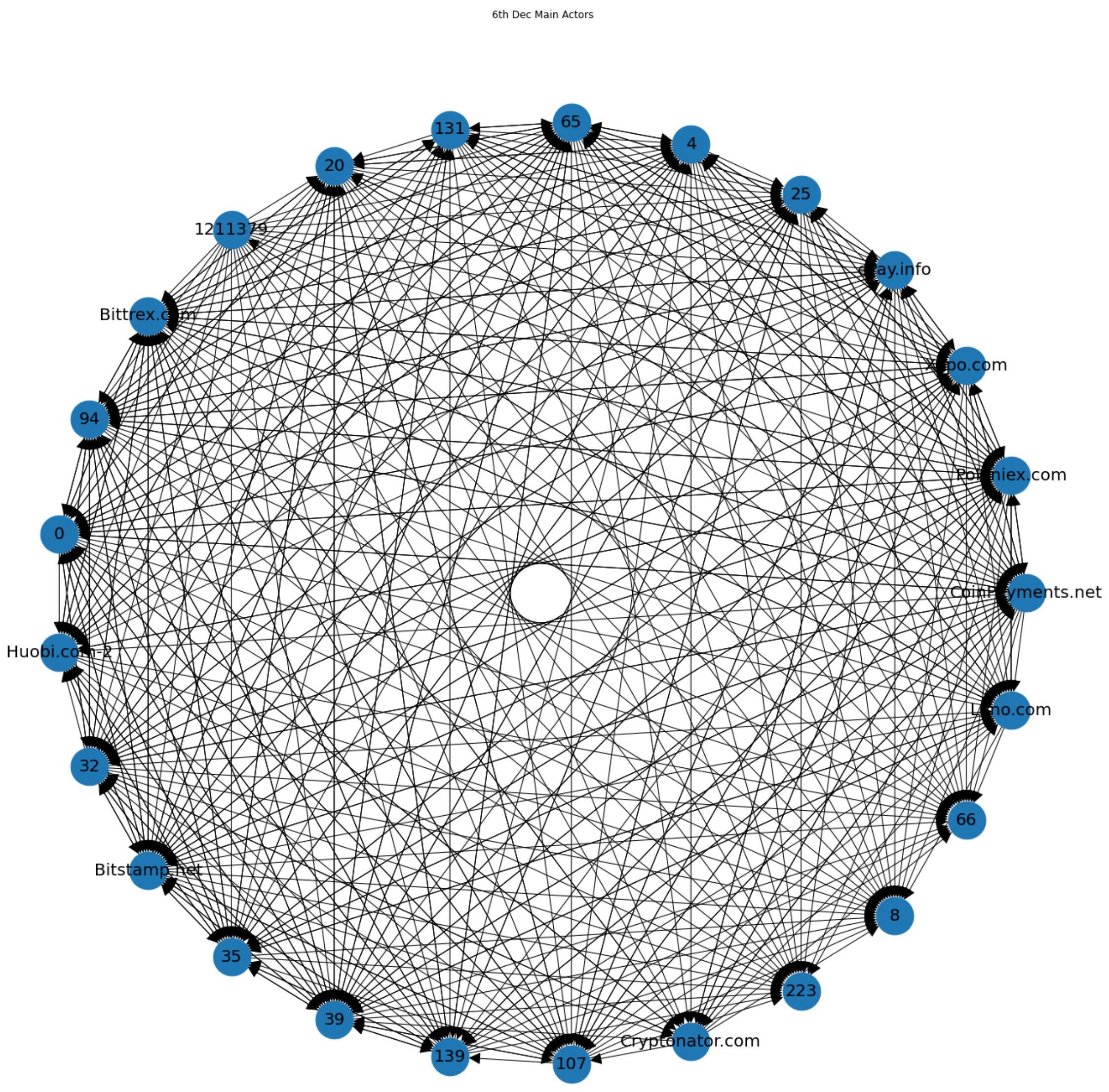
```
In [385]: print("nodes of highest degree 4th November 2017:")
          sorted(nx.degree(g10), key=lambda x: x[1],  reverse=True)[:25]

          nodes of highest degree 4th November 2017:

Out[385]: [('Bittrex.com', 44),
           ('CoinPayments.net', 44),
           ('Poloniex.com', 44),
           ('32', 44),
           ('354', 43),
           ('39', 43),
           ('25', 43),
           ('65', 43),
           ('0', 43),
           ('35', 43),
           ('Xapo.com', 42),
           ('Bitstamp.net', 42),
           ('189', 42),
           ('ePay.info', 41),
           ('94', 40),
           ('4', 40),
           ('62', 38),
           ('20', 37),
           ('Cryptonator.com', 37),
           ('107', 34),
           ('204', 32),
           ('1211379', 23),
           ('Luno.com', 23),
           ('30', 22),
           ('66', 21)]
```

```
In [386]: print("nodes of highest degree 6th December 2017:")
          sorted(nx.degree(g20), key=lambda x: x[1],  reverse=True)[:25]

          nodes of highest degree 6th December 2017:

Out[386]: [('Poloniex.com', 44),
           ('Bittrex.com', 44),
           ('Bitstamp.net', 42),
           ('39', 42),
           ('CoinPayments.net', 41),
           ('65', 41),
           ('94', 40),
           ('0', 40),
           ('32', 39),
           ('Xapo.com', 37),
           ('4', 37),
           ('20', 36),
           ('35', 36),
           ('25', 35),
           ('107', 35),
           ('ePay.info', 34),
           ('139', 33),
           ('131', 30),
           ('Huobi.com-2', 30),
           ('Luno.com', 28),
           ('1211379', 24),
           ('Cryptonator.com', 24),
           ('8', 22),
           ('66', 21),
           ('223', 19)]
```

```
In [400]: figure(figsize=(25, 25))
          plt.title('4th Nov Main Actors')
          nx.draw_shell(g10,node_size=2000 ,with_labels=True,arrows=True,arro
```



```
In [401]: figure(figsize=(25, 25*))
          plt.title('6th Dec Main Actors')
          nx.draw_shell(g20,node_size=2000 ,with_labels=True,arrows=True,arro
```



From these two graphs, we can confirm that there's a heavy concentration of transactions originating from the major actors and that there is bilateral connections between most of the nodes studied for both of the considered dates. Most of the major actors are also common to both the dates.