HW2: Exploring Bitcoin transactions - Report
Ramandeep Singh
8019-7991

Tool Used:-

Pandas - Pandas is a python library used for data analysis.
SQLite- An SQLite database can be read directly from pandas library and can be used to implement SQL concepts on the large CSV file for data analysis.

Installation Steps:-

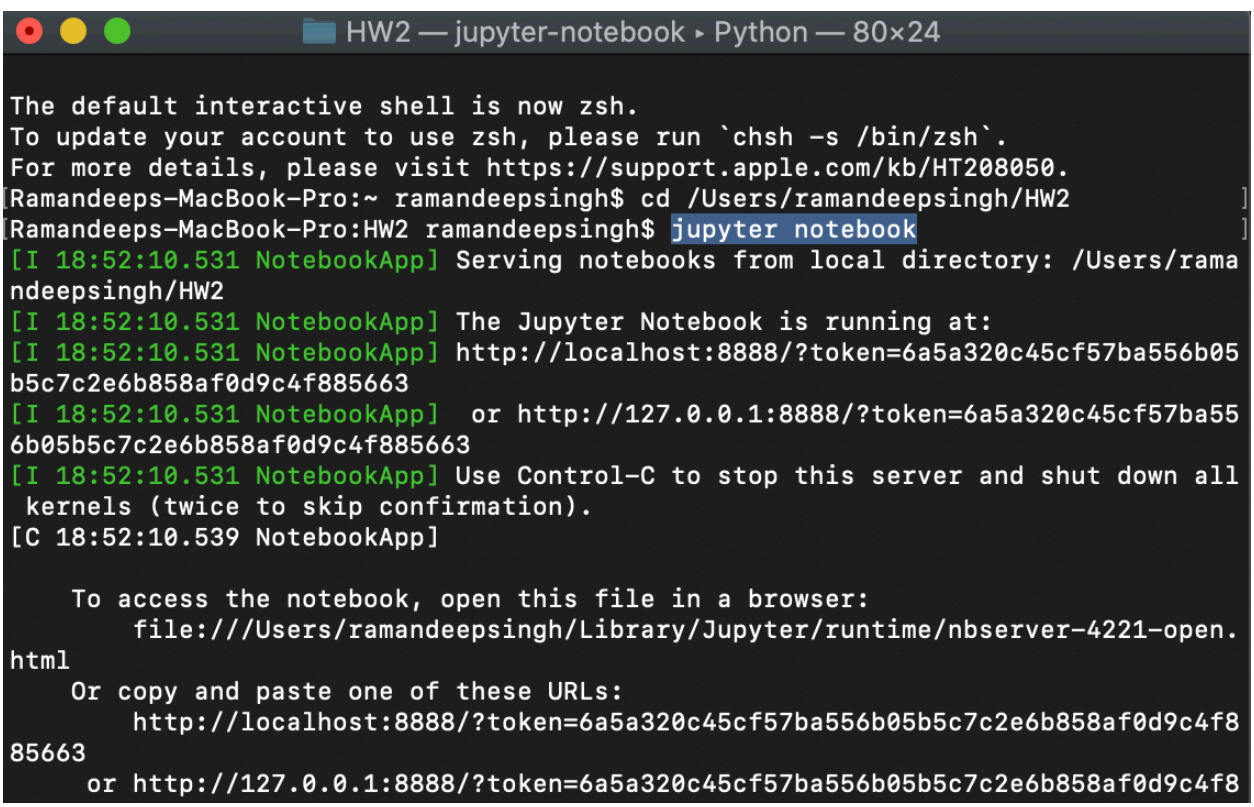
Run below commands in terminal to install pandas library, Jupyter notebook and sqlite in pandas

pip3 install pandas
pip3 install notebook
pip3 install -U pandasql

To run Jupyter notebook run below command

jupyter notebook

The above command should be run from the same path where csv files are placed.



Part 1: Transactions analysis

1. What is the number of transactions and addresses in the dataset?

Number Of Transactions - 10000055
Number of Addresses - 8385065

Code:-

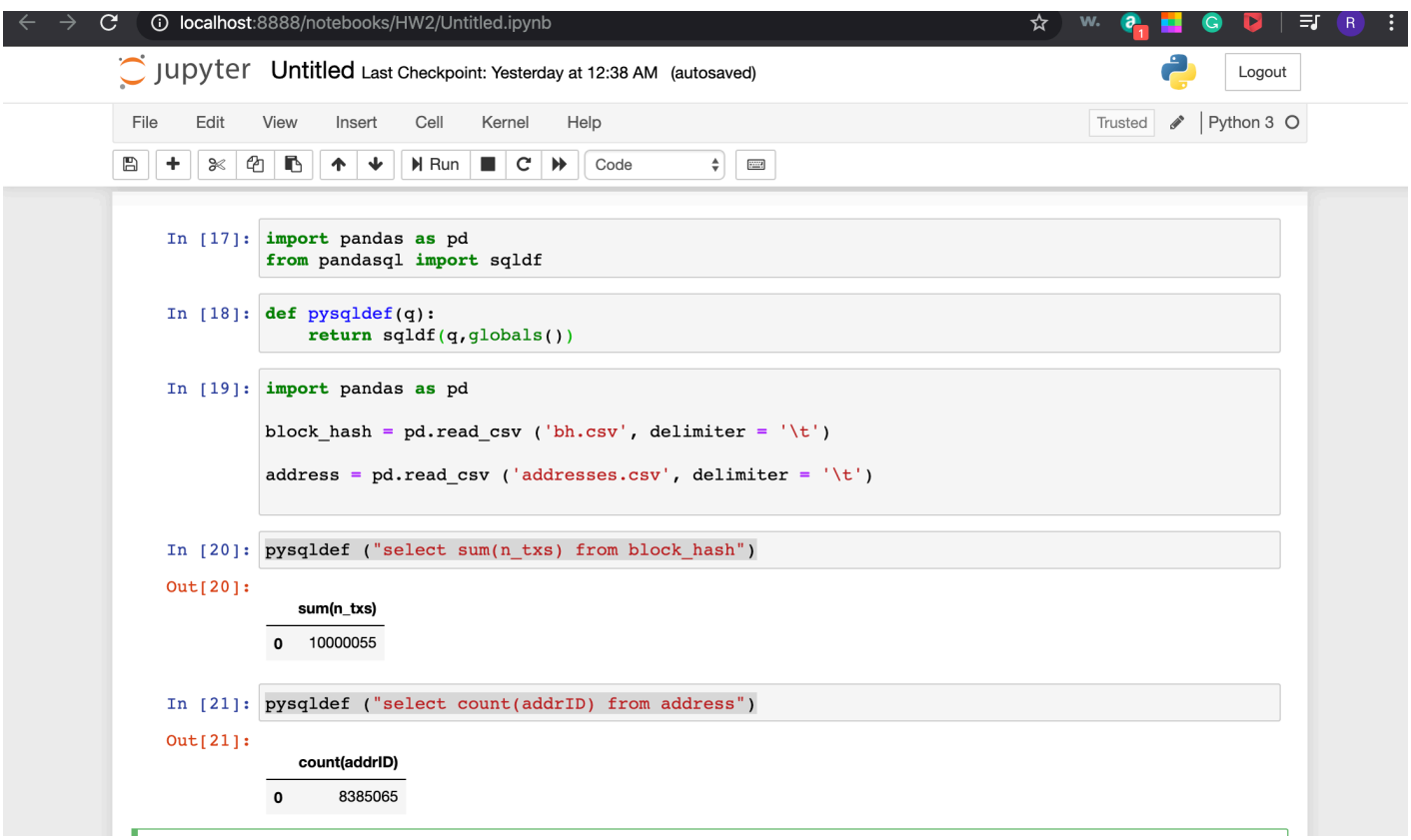
```
import pandas as pd
from pandasql import sqldf

def pysqldef(q):
    return sqldf(q,globals())

import pandas as pd
block_hash = pd.read_csv ('bh.csv', delimiter = '\t')
address = pd.read_csv ('addresses.csv', delimiter = '\t')

pysqldef ("select sum(n_txs) from block_hash")

pysqldef ("select count(addrID) from address")
```

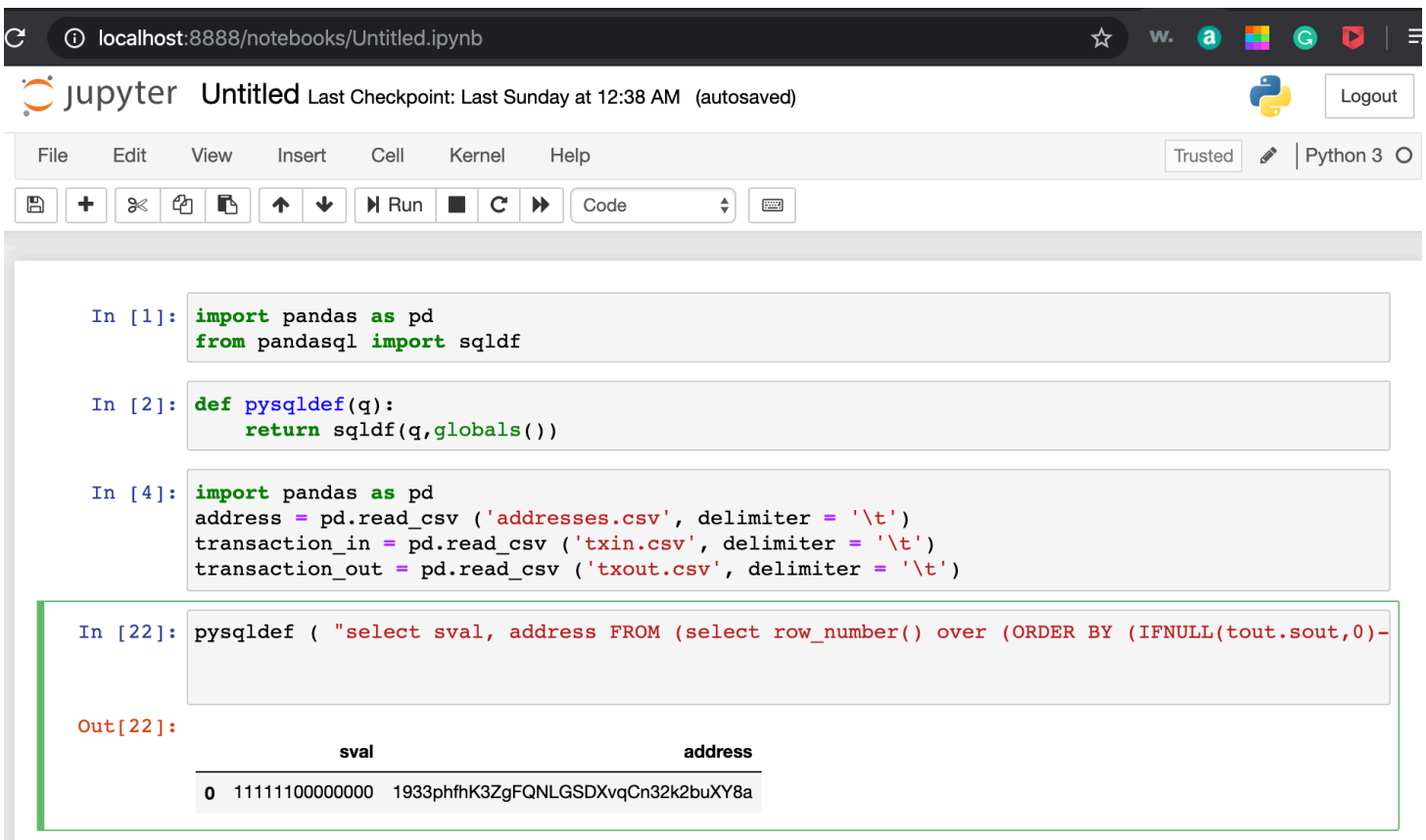


2. What is the Bitcoin address that is holding the greatest amount of bitcoins? How much is that exactly? Note that the address here must be a valid Bitcoin address string. To answer this, you need to calculate the balance of each address. The balance here is the total amount of bitcoins in the UTXOs of an address.

Address - 1933phfhK3ZgFQNLGSDXvqCn32k2buXY8a
Amount - 11111100000000

```
import pandas as pd
address = pd.read_csv ('addresses.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef ( "select sval, address FROM (select row_number() over (ORDER BY (IFNULL(tout.sout,0)-IFNULL(tin.sin,0)) DESC)  row_num, (IFNULL(tout.sout,0)-IFNULL(tin.sin,0)) sval, ad.address, ad.addrid from address ad, (select IFNULL(sum(sum),0) sout, addrID from transaction_out group by addrID) tout NATURAL LEFT OUTER JOIN (select IFNULL(sum(sum),0) sin, addrID from transaction_in group by addrID) tin where tout.addrID = ad.addrID) where row_num = 1" )
```

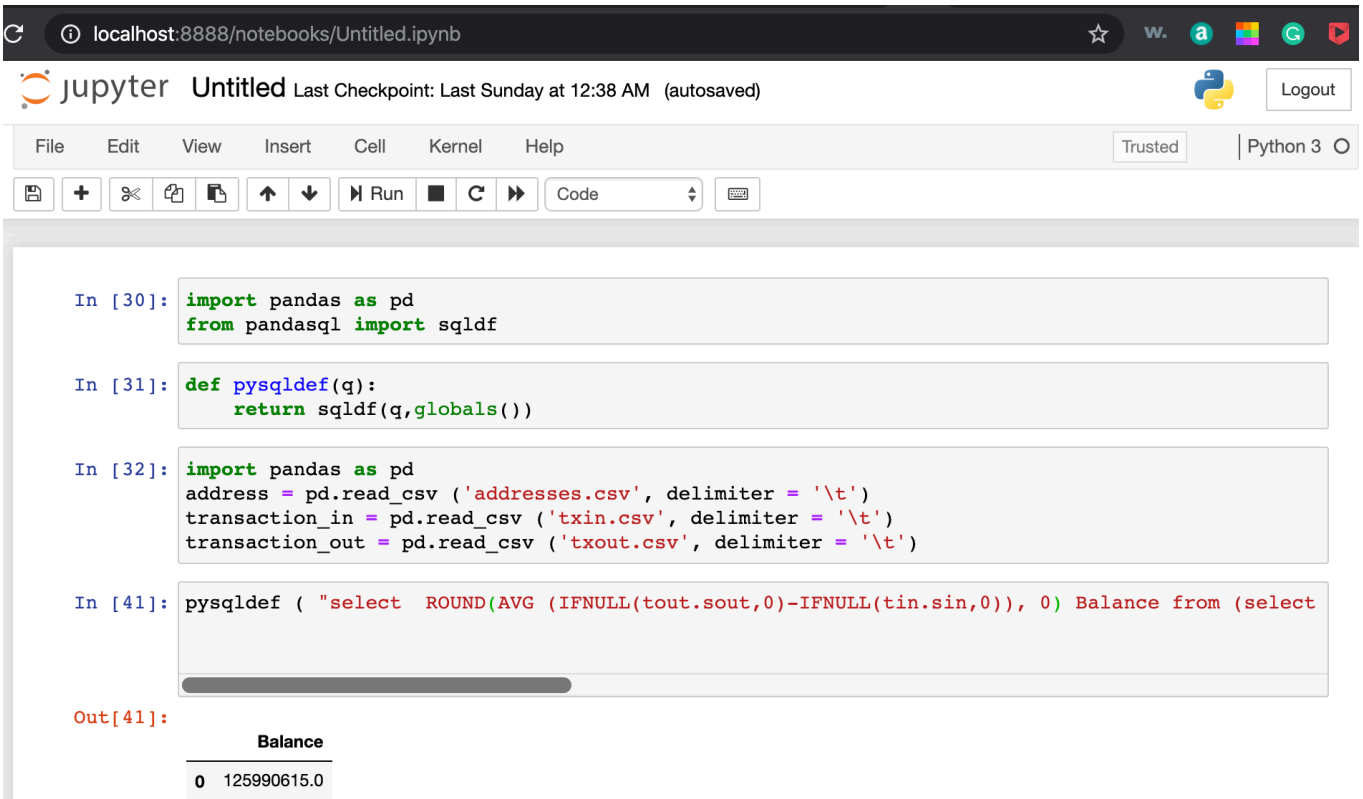


3. What is the average balance per address?

Average Balance per address - 125990615.0

```
import pandas as pd
address = pd.read_csv ('addresses.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef ( "select  ROUND(AVG (IFNULL(tout.sout,0)-IFNULL(tin.sin,0)), 0) Balance from (select
IFNULL(sum(sum),0) sout, addrID from transaction_out group by addrID) tout NATURAL LEFT
OUTER JOIN (select IFNULL(sum(sum),0) sin, addrID from transaction_in group by addrID)
tin" )
```



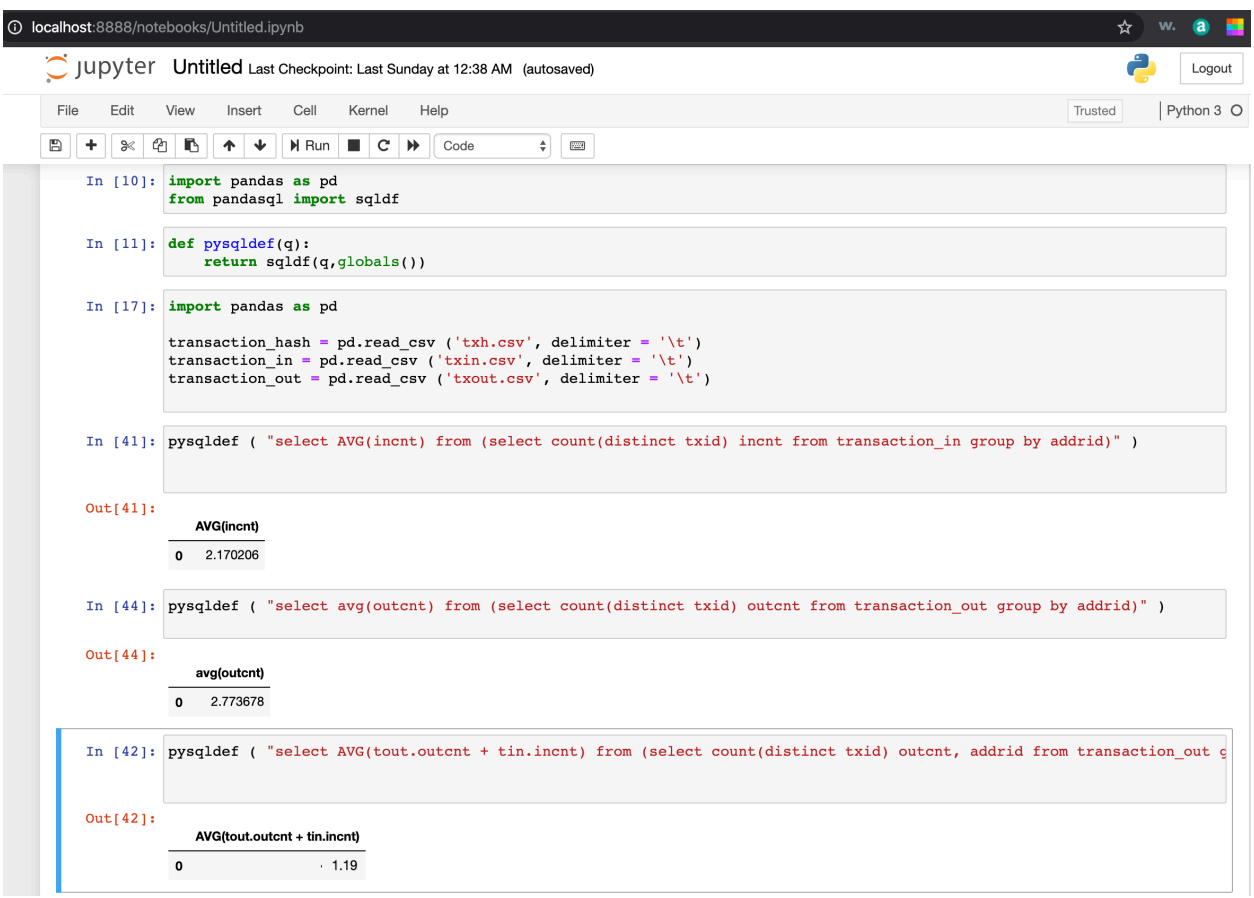
4. What is the average number of input and output transactions per address? What is the average number of transactions per address (including both inputs and outputs)? An output transaction of an address is the transaction that is originated from that address. Likewise, an input transaction of an address is the transaction that sends bitcoins to that address.

Average In - 2.17
Average Out - 2.773678
Average Total - 1.19

pysqldef ("select AVG(incnt) from (select count(distinct txid) incnt from transaction_in group by addrid)")

pysqldef ("select avg(outcnt) from (select count(distinct txid) outcnt from transaction_out group by addrid)")

pysqldef ("select AVG(tout.outcnt + tin.incnt) from (select count(distinct txid) outcnt, addrid from transaction_out group by addrid) tout, (select count(distinct txid) incnt , addrid from transaction_in group by addrid) tin where tout.addrID = tin.addrID")



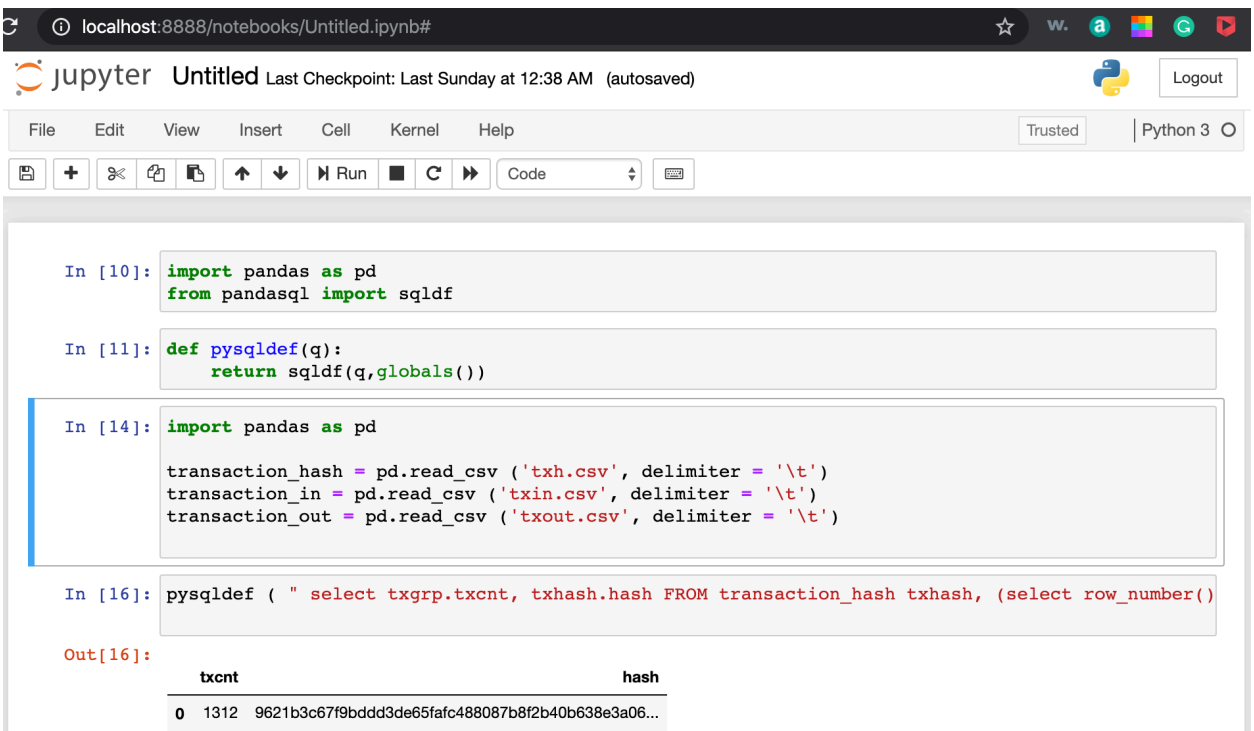
5. What is the transaction that has the greatest number of inputs? How many inputs exactly? Show the hash of that transaction. If there are multiple transactions that have the same greatest number of inputs, show all of them.

Transaction Hash - 9621b3c67f9bddd3de65fafc488087b8f2b40b638e3a06209a904c66c0b32982
Number of transactions - 1312

import pandas as pd

transaction_hash = pd.read_csv ('txh.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef (" select txgrp.txcnt, txhash.hash FROM transaction_hash txhash, (select row_number() over (ORDER BY count(txid) DESC) row_num, count(txid) txcnt, txid from transaction_in group by txid) txgrp where txgrp.txid = txhash.txid and row_num = 1 ")



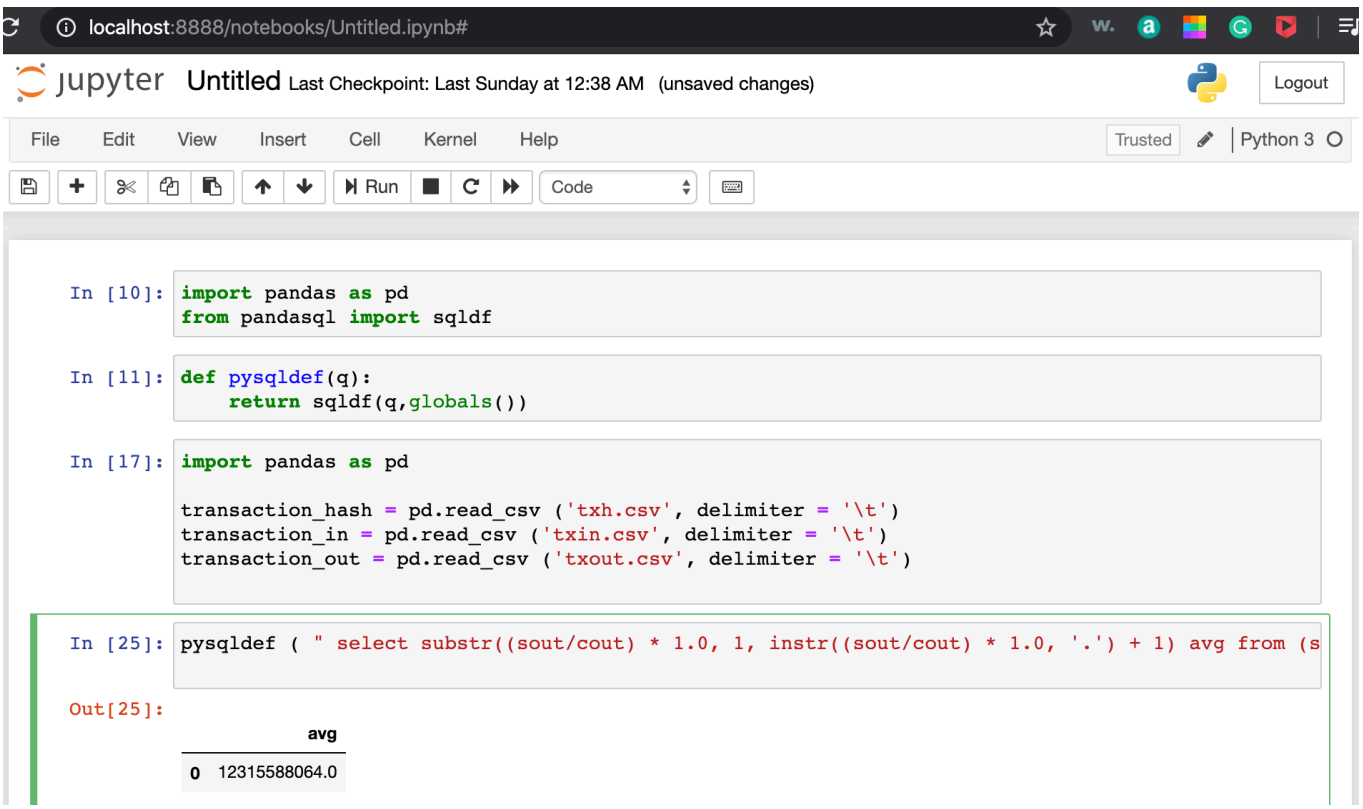
6. What is the average transaction value? Transaction value is the sum of all outputs' value.

Average Transaction Value - 12315588064.0

import pandas as pd

transaction_hash = pd.read_csv ('txh.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef (" select substr((sout/cout) * 1.0, 1, instr((sout/cout) * 1.0, '.') + 1) avg from (select sum(sum) sout,count(distinct txid) cout from transaction_out) ")



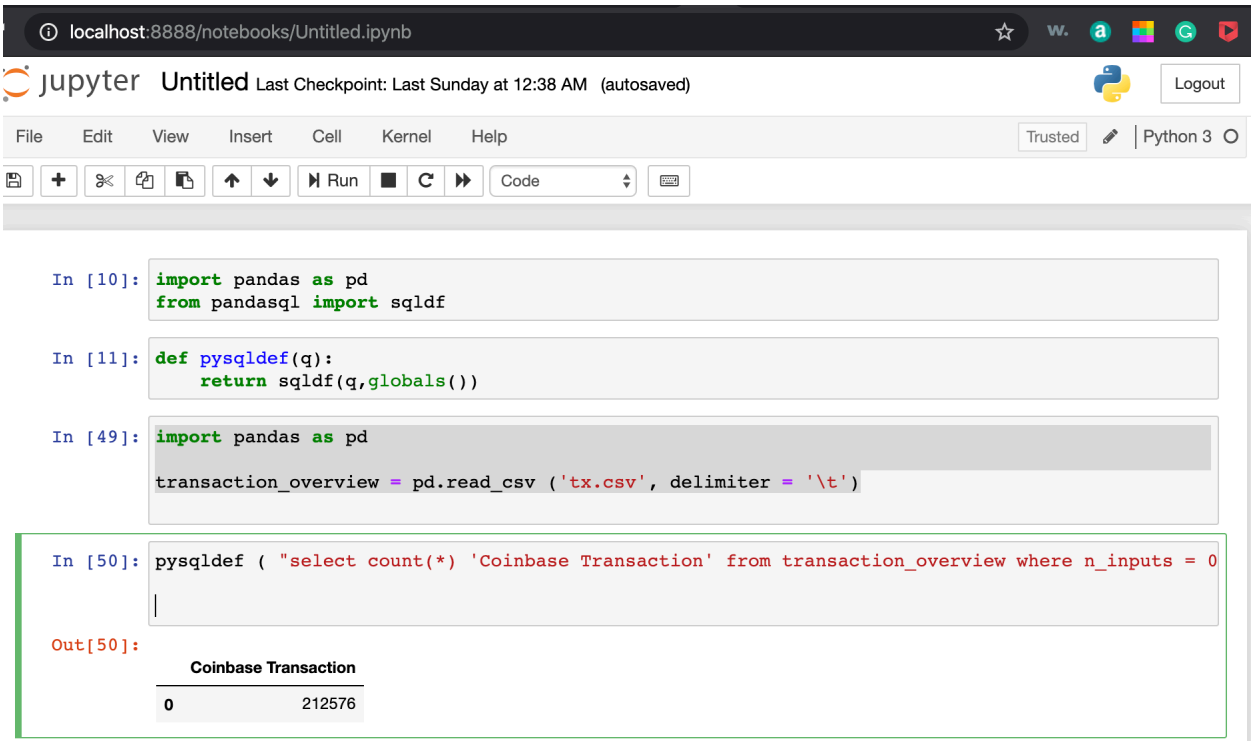
7. How many coinbase transactions are there in the dataset?

Number of Coinbase Transaction - 212576

import pandas as pd

transaction_overview = pd.read_csv ('tx.csv', delimiter = '\t')

pysqldef ("select count(*) 'Coinbase Transaction' from transaction_overview where n_inputs = 0")



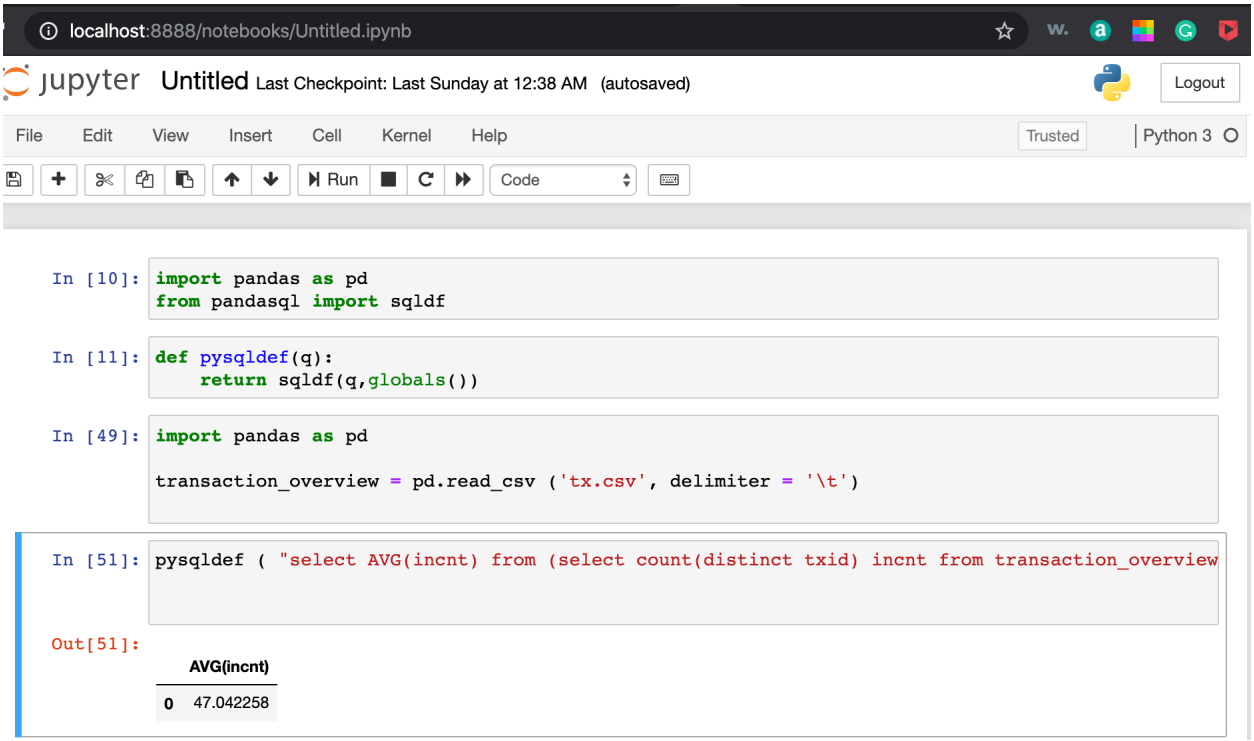
8. What is the average number of transactions per block?

Average Number of transactions per block - 47.042258

import pandas as pd

transaction_overview = pd.read_csv ('tx.csv', delimiter = '\t')

pysqldef ("select AVG(incnt) from (select count(distinct txid) incnt from transaction_overview group by blockid)")



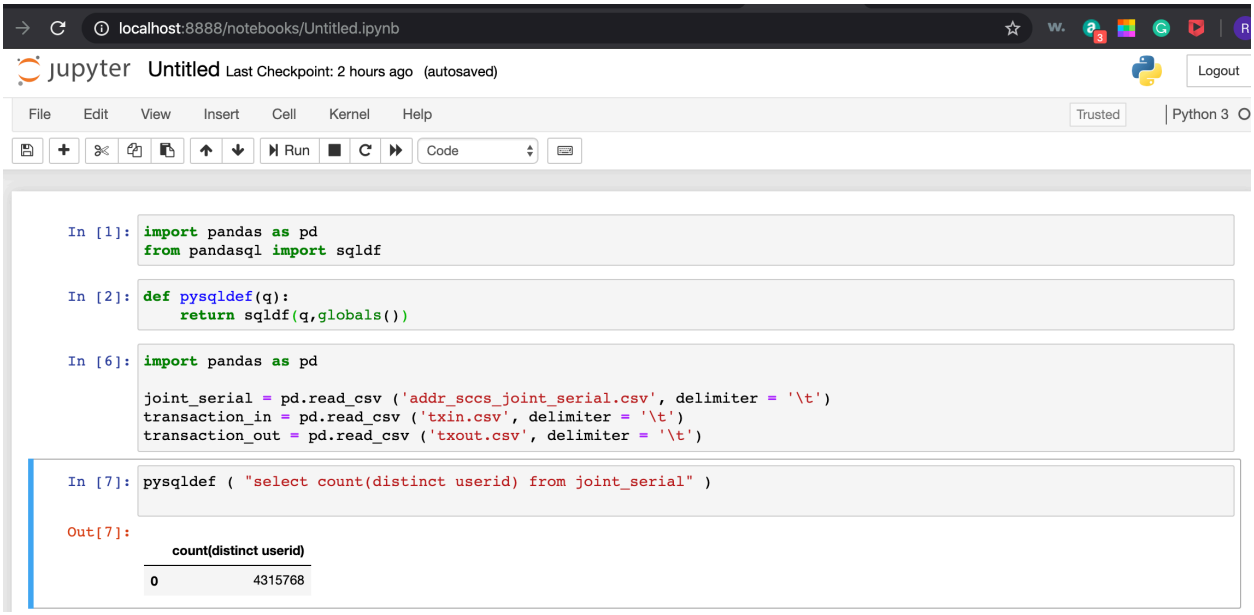
Part 2: Address de-anonymization

1. How many users are there in the dataset?

Number of distinct Users - 4315768

import pandas as pd
joint_serial = pd.read_csv ('addr_sccs_joint_serial.csv', delimiter = '\t')

pysqldef ("select count(distinct userid) from joint_serial")



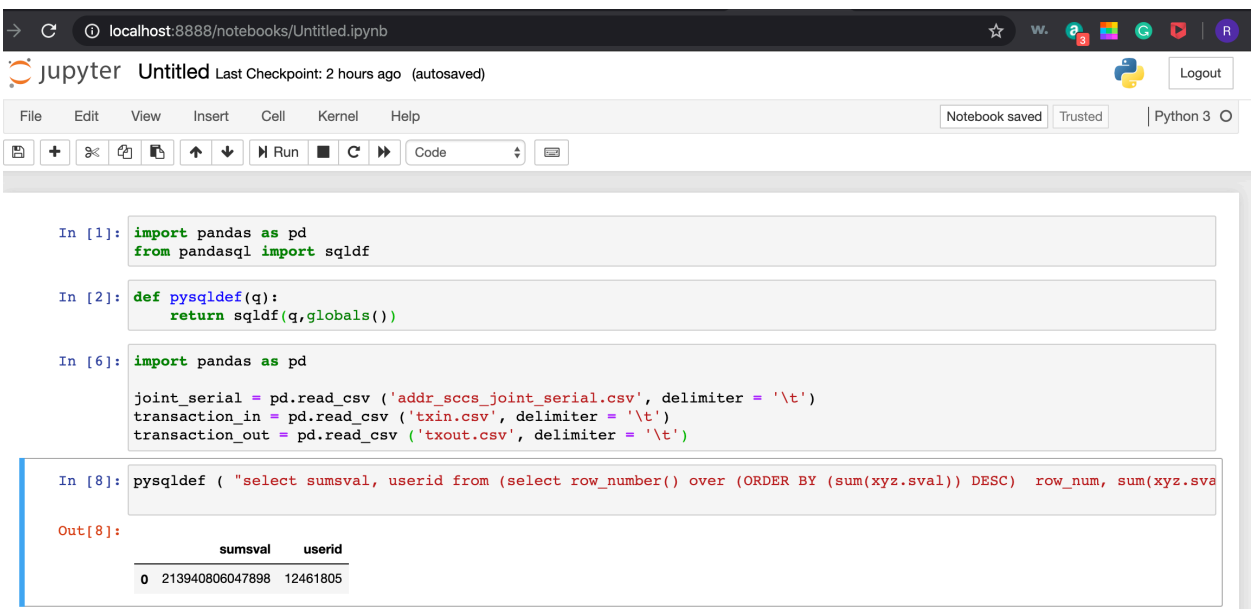
2.1 What is the Bitcoin user that is holding the greatest amount of bit coins? How much is that exactly? Note that the address here must be a valid Bitcoin address string. To answer this, you need to calculate the balance of each address. The balance here is the total amount of bitcoins in the UTXOs of an address.

UserID - 12461805
Amount - 213940806047898

```
import pandas as pd

joint_serial = pd.read_csv ('addr_sccs_joint_serial.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef ( "select sumsval, userid from (select row_number() over (ORDER BY (sum(xyz.sval)) DESC)  row_num, sum(xyz.sval) sumsval, xyz.userid from (select addr.addrid, addr.userid, abc.sval from joint_serial addr, (select (IFNULL(tout.sout,0)-IFNULL(tin.sin,0)) sval, tout.addrID addrid from (select IFNULL(sum(sum),0) sout, addrID from transaction_out group by addrID) tout NATURAL LEFT OUTER JOIN (select IFNULL(sum(sum),0) sin, addrID from transaction_in group by addrID) tin ) abc where addr.addrid = abc.addrid) xyz group by xyz.userid) where row_num = 1" )
```

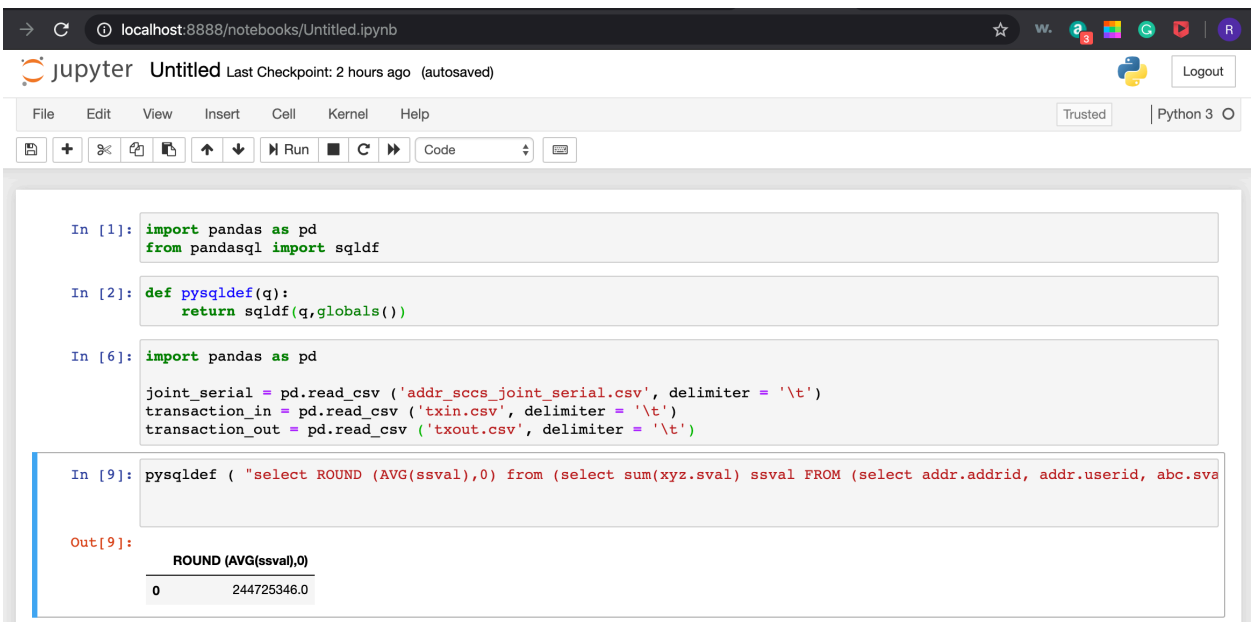


2.2 What is the average balance per user?

Average Balance per user - 244725346.0

```
import pandas as pd
joint_serial = pd.read_csv ('joint_serial.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef ( "select ROUND (AVG(ssval),0) from (select sum(xyz.sval) ssval FROM (select addr.addrid, addr.userid, abc.sval from joint_serial addr, (select (IFNULL(tout.sout,0)-IFNULL(tin.sin,0)) sval, tout.addrID addrid from (select IFNULL(sum(sum),0) sout, addrID from transaction_out group by addrID) tout NATURAL LEFT OUTER JOIN (select IFNULL(sum(sum),0) sin, addrID from transaction_in group by addrID) tin) abc where addr.addrid = abc.addrid) xyz group by xyz.userid)" )
```



2.3 What is the average number of input and output transactions per user? What is the average number of transactions per user (including both inputs and outputs)? An output transaction of user is the transaction that is originated from that user. Likewise, an input transaction of user is the transaction that sends bitcoins to that user.

Average In - 2.170198
Average Out - 2.773547
Average Total - 4.979047

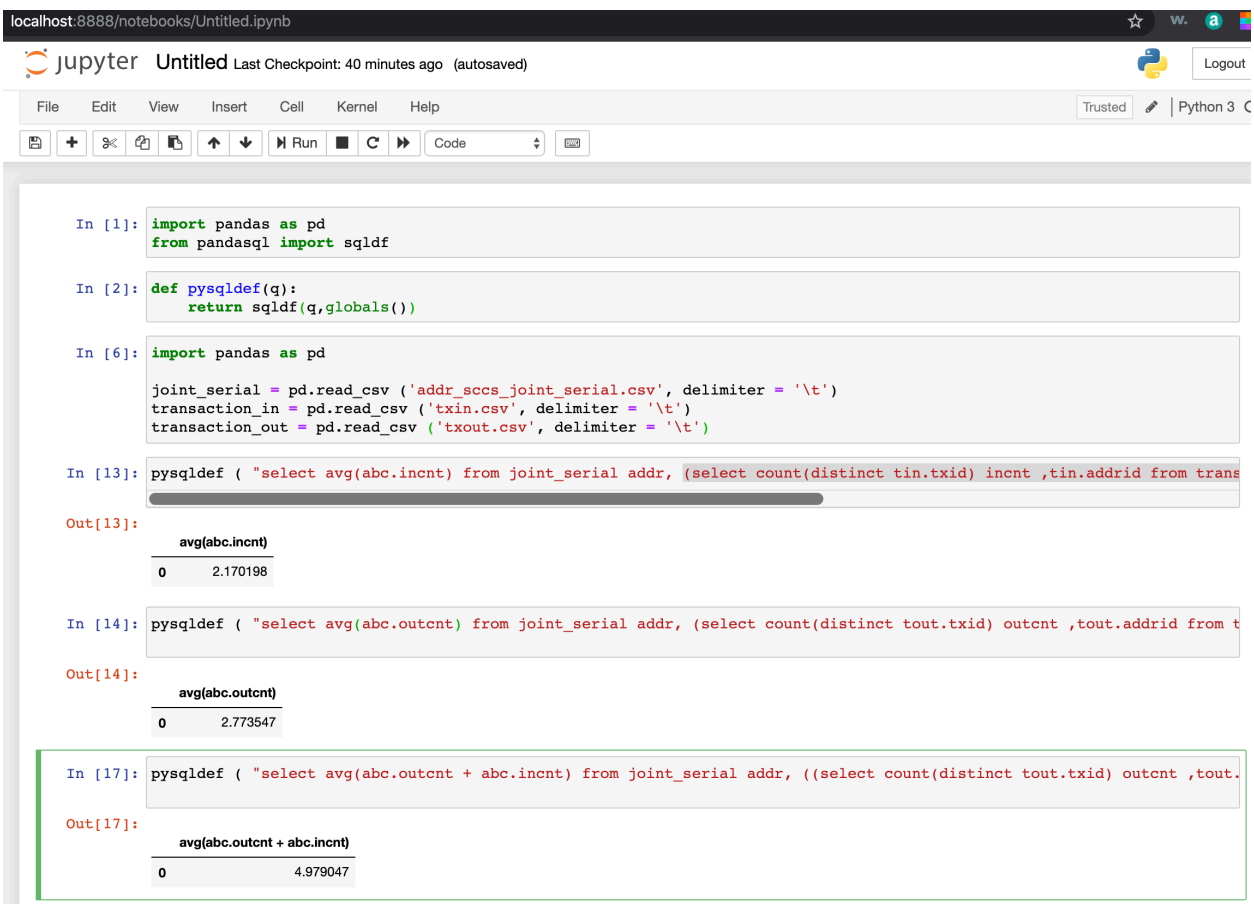
import pandas as pd

joint_serial = pd.read_csv ('joint_serial.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')

pysqldef ("select avg(abc.incnt) from joint_serial addr, (select count(distinct tin.txid) incnt ,tin.addrid from transaction_in tin group by tin.addrid) abc where abc.addrid = addr.addrid")

pysqldef ("select avg(abc.outcnt) from joint_serial addr, (select count(distinct tout.txid) outcnt ,tout.addrid from transaction_out tout group by tout.addrid) abc where abc.addrid = addr.addrid")

pysqldef ("select avg(abc.outcnt + abc.incnt) from joint_serial addr, ((select count(distinct tout.txid) outcnt ,tout.addrid from transaction_out tout group by tout.addrid) NATURAL LEFT OUTER JOIN (select count(distinct tin.txid) incnt ,tin.addrid from transaction_in tin group by tin.addrid)) abc where abc.addrid = addr.addrid")



3. Give the hash of the transaction sending the greatest number of bitcoins to the user who is holding the greatest balance.

Transaction Hash - c246c27e7bacc667d27ace253abf2bba82aa1e5fcd1d73e1b85863f6b890e1bf

import pandas as pd

joint_serial = pd.read_csv ('addr_sccs_joint_serial.csv', delimiter = '\t')
transaction_in = pd.read_csv ('txin.csv', delimiter = '\t')
transaction_out = pd.read_csv ('txout.csv', delimiter = '\t')
transaction_hash = pd.read_csv ('txh.csv', delimiter = '\t')

pysqldef ("select hash from transaction_hash where txid = (select txid from (select row_number() over (ORDER BY (sum) DESC) row_num, txid, sum, addrid from transaction_in where addrid IN (select addrid from joint_serial where userid = 12461805)) where row_num = 1)")