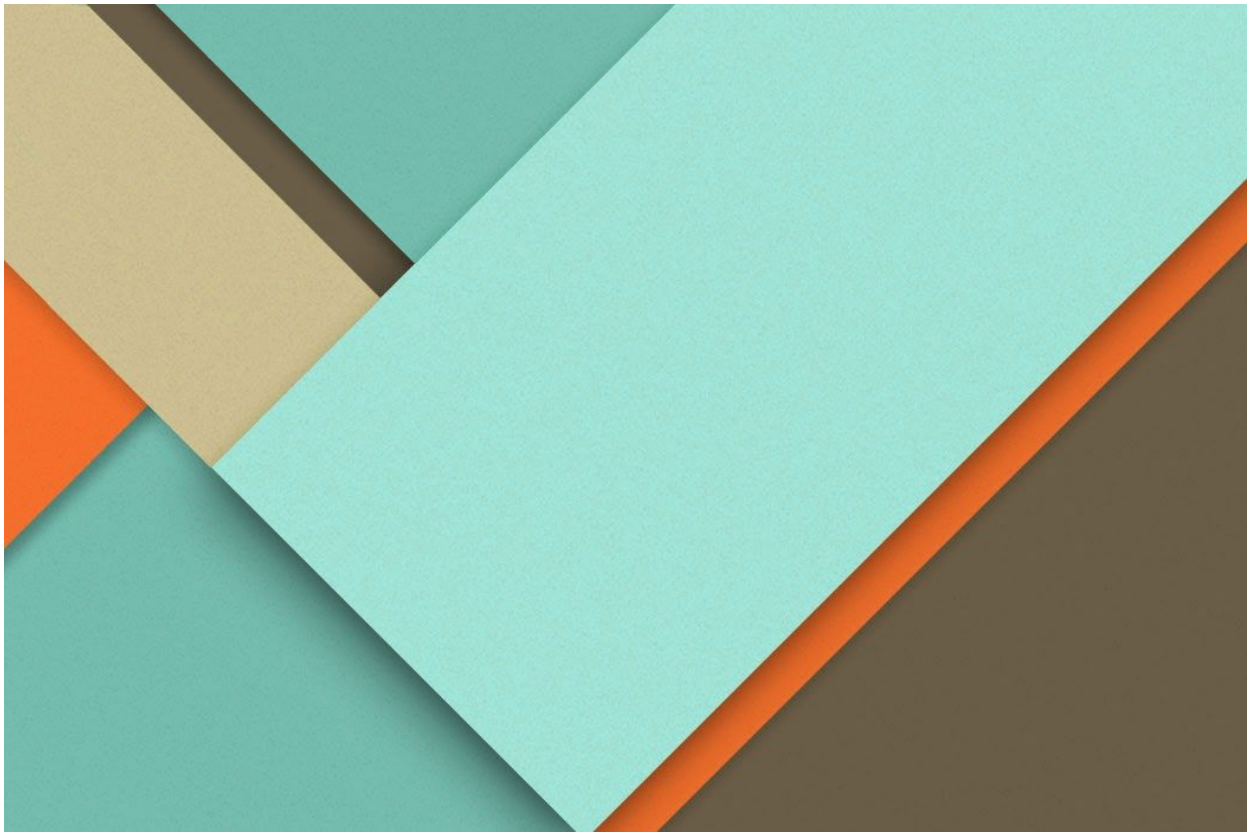


Report (Introduction to Coding)

18.01.2021

Javokhir Isomurodov
876613

Overview

The following is the report of my analysis of the Bitcoin block 657354. It provides information on statistical patterns of movements within this specific block. Such patterns include the expected value of bitcoins moved in the block, the maximum and minimum amounts of transactions, a few plots illustrating the distribution of transactions, and some information regarding the fees paid by the users to consolidate their transactions in the block.

Specifications

One of the important moments in the process of my analysis was the discovery of the following: it is almost impossible to identify the owners of wallets. In other words, every time one makes a transaction, he/she can generate new keys, which are all stored as a collection in their bitcoin wallet. For this reason, my findings can offer only the information about the movements/transactions between addresses, and not between wallets.

Parts of the code:

I divided my code into the following three parts:

I. Retrieving the data

I used this technique which was taught during the classes in order to get more complete data that would also include SegWit transactions as well as the missing addresses.

II. Processing the data

This is the part where actual analysis takes place. The following part includes one function (**minmax**), several variables, and a for loop comprising two other loops with if-else structures in each of them. After the execution of the loop, I display some data on the screen using the print function.

Important findings include:

- Mean value of bitcoins moved in the block is **65380381.929245286 S**[1]

[1] Stored in the dictionary **'dic'** - **keys** are the numbers of order of transactions; **values** are the outputs made in each transaction

- Maximum and minimum amounts of fees paid are **334212** and **178 S** respectively[2];
- Maximum/minimum output/input[3]:

	OUTPUT	**INPUT**
MAXIMUM	43119903055 S	43119903723 S
MINIMUM	0 S	546 S

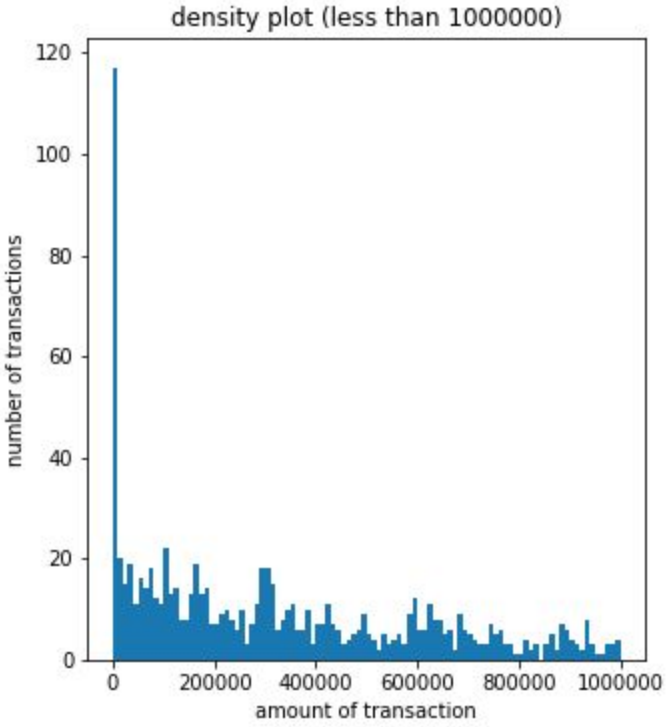
  - **Maximum** output is **43119903055 S** by address **3DRdYoNeVrAYVDcN7rTTekfwT8K644Uwj7;**
  - **Minimum** outputs are multiple, all equal to **0;**
  - **Maximum** input is **43119903723 S** by address **3BBcr1ps4h8yX7U6wdbkdnks3NPJoeeLq7;**
  - **Minimum** output is **546 S** by address **1HpZ5uLZLufGnyYGhoMqdqZVsQRFghvNMq**

III. Graphs

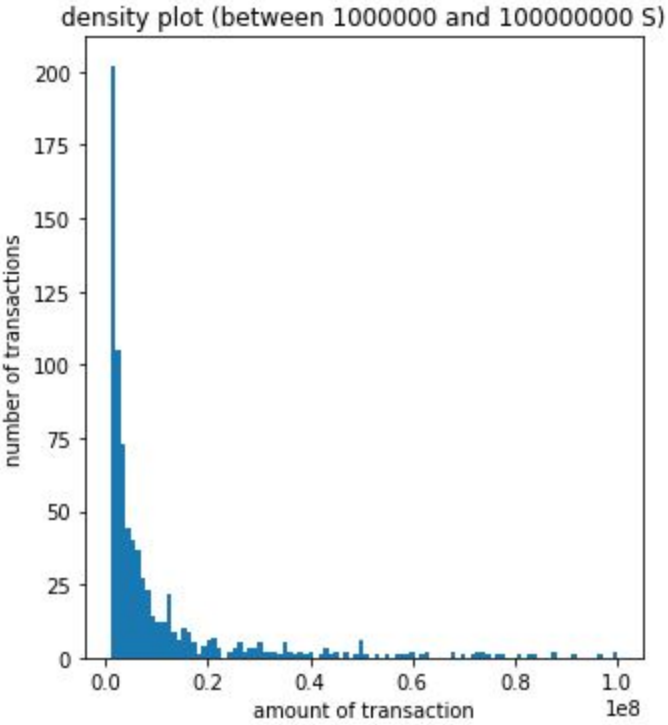
In the last part, I used the pyplot method of the matplotlib library to represent the density plots and graphic visualizations of variables. In overall, there are six graphs. Three of them illustrate the distribution of transaction according to their size:

- First graph (histogram) depicts the transactions whose values are **less** than **1000000 S**:

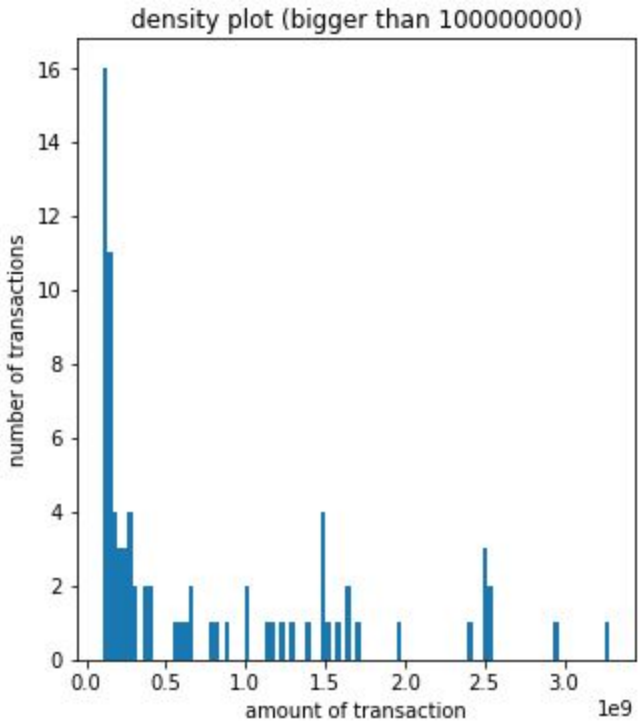
[2] Stored in the list **'fees'**; values are added into list in **line 57**, right after entering each transaction
[3] The values are extracted from **'dic_outs'** and **'dic_ins'** by using the **'minmax'** function



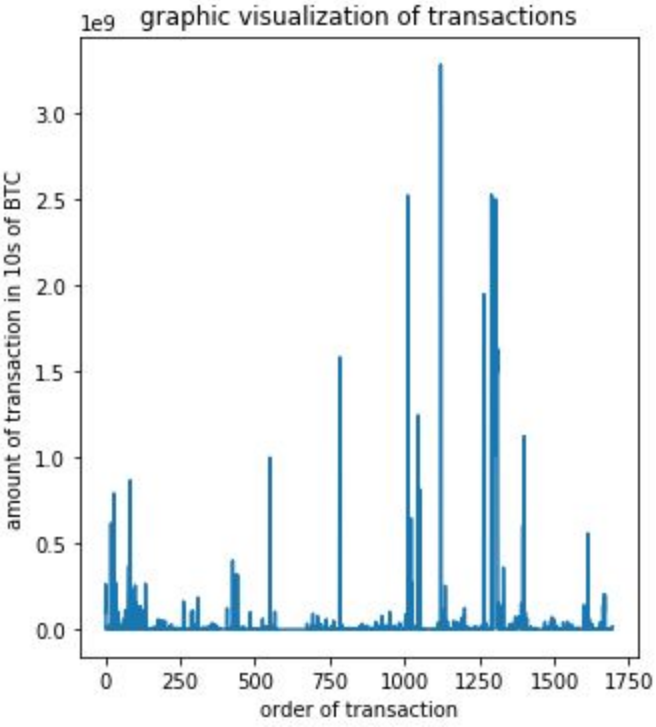
- Second graph (histogram) depicts the transactions whose values are **between 1000000** and **100000000 S**:



- Third graph (histogram) depicts the transactions whose values **exceed 100000000 S:**



The 4th plot visualizes all the transactions except for the transaction #1374 which contains abnormally big value:



While the 5th graph is similar to the previous one, the last, 6th, graph provides an interesting insight. It illustrates the fees as percentages of the respecting transactions. It can be seen from the graph that most of the users paid up to 0.02% of the amount of transactions: