Task 1: Compute statistics from a data stream

Report

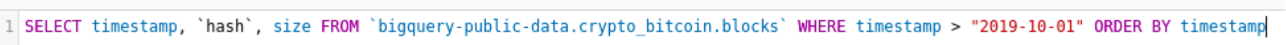
Igor Hamidović

1. Introduction

In the document is described results of working on task 1[1]. Complete source core used in this work is publicly available on GitHub[2].

2. Data

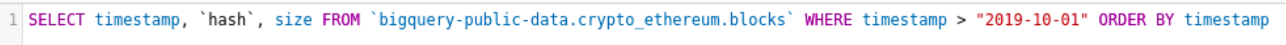
Data used here are downloaded form Google Cloud BigQuery public datasets. For the work is used two datasets Bitcon Cryptocurrency[3] dataset and Ethereum Cryptocurrencey[4] dataset. In both datasets are saved data about blocks and transactions of the cryptocurrences. In this work focus is analyzing size of blocks through time.



```
SELECT timestamp, `hash`, size FROM `bigquery-public-data.crypto_bitcoin.blocks` WHERE timestamp > "2019-10-01" ORDER BY timestamp
```

Figure 1. Code for quering data for Bitcoun

Query used for getting the data for Bitcoin and Etherium in presented in figure 1 and figure 2. So, in the data there are size, hash and timestamp of Etherium and Bitcoin blocks created from 2019-10-01. Also, the data are saved as CSV files in the GitHub repository in Datasets folder.



```
SELECT timestamp, `hash`, size FROM `bigquery-public-data.crypto_ethereum.blocks` WHERE timestamp > "2019-10-01" ORDER BY timestamp
```

Figure 2. Code for quering data for Etherium

Time between two blocks is different in both datasets and blocks in Etherium appear more frequently than blocks for Bitcoin because there may be problems with calculating colerrations between the datasets. The problem is solved hourly aggregating using arithmetic mean for both datasets. In this way is simulated two data streams with one hour as time interval (*timepoint*). So, after preprocessing the datasets contains 781 row in both datasets.

Code for loading, aggregating and with useful functions for data streams is in folder preprocessing in the GitHub repository.

3. Single Stream Statistics

Both streams are analyzed separately using arithmetic mean, standard deviation and autocorrelations. Analyzing is done using incremental statistics and statistics based on sliding windows. For sliding window statistics is use 50 as window size and 0.004 as delta.

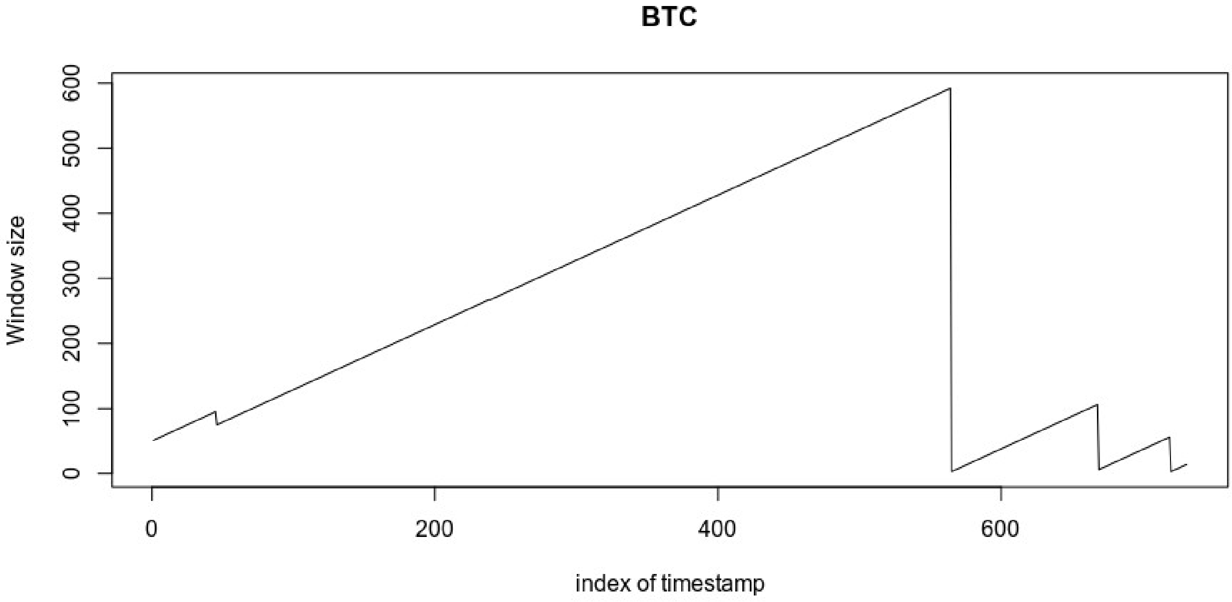
In some of next diagrams is used *index of timestamp* for clearly visualization. One value of index of timestamp is equal to one hour of the timestamp.

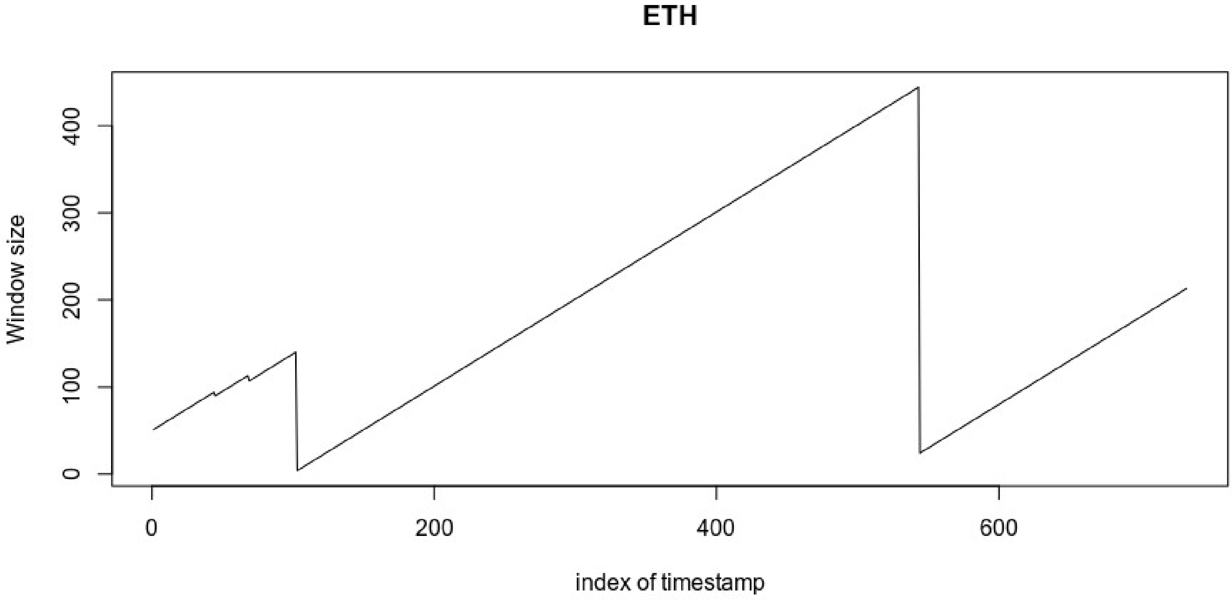
[1] https://ucilnica.fri.uni-lj.si/mod/assign/view.php?id=8603

[2] https://github.com/igorHamidovic/fri_ipui1

[3] https://console.cloud.google.com/marketplace/details/bitcoin/crypto-bitcoin?filter=solution-type:dataset&filter=category:finance&id=7fd60425-cb95-4a58-b59f-ab3789642844

[4] https://console.cloud.google.com/marketplace/details/ethereum/crypto-ethereum-blockchain?filter=solution-type:dataset&filter=category:finance&id=999d739c-de61-4550-8d9c-0753ca827cd3





3.1. Arithmetic mean

In figures 3 and 4 are presented arithmetic mean through time of BTC using incremental statistics and sliding windows.

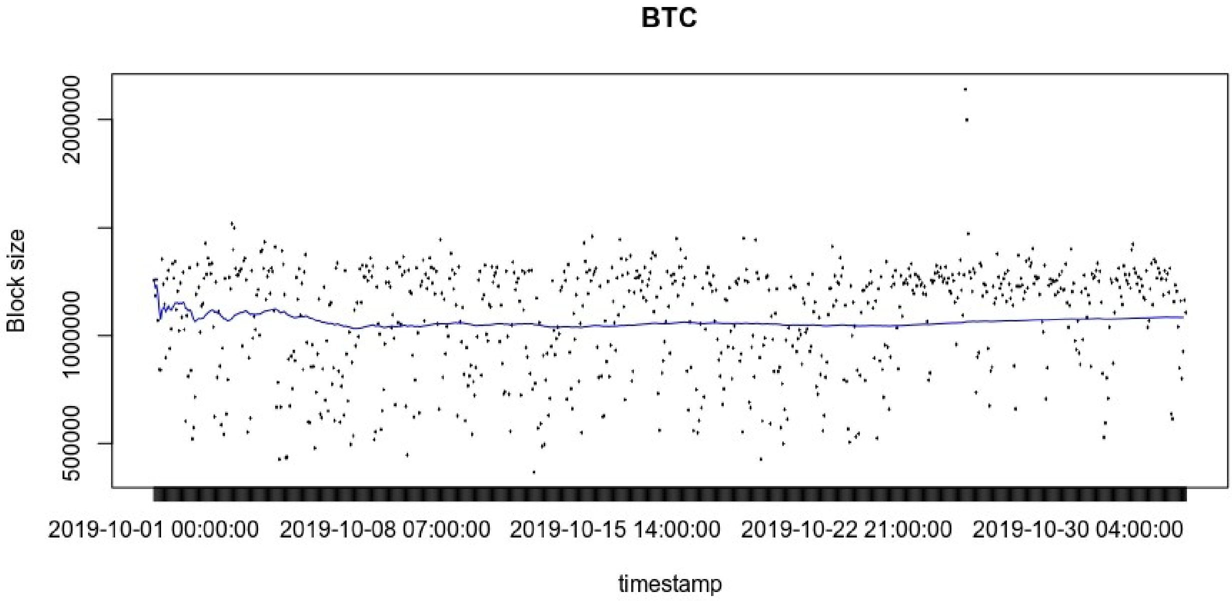


Figure 3. Arithmetic mean for size of blocks of BTC through time using incremental statistics

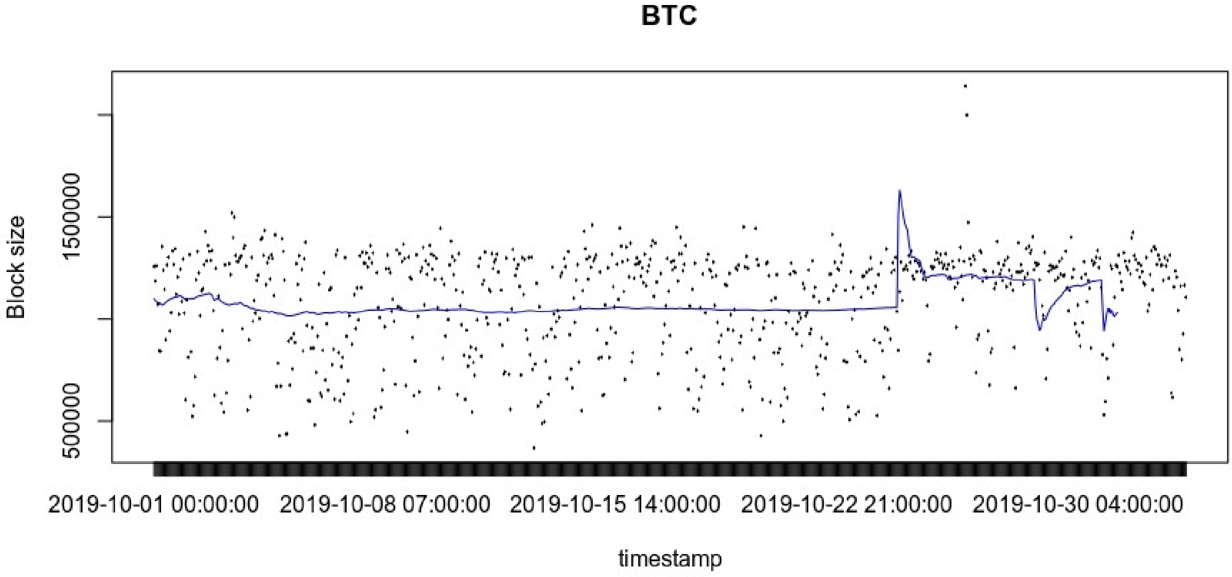


Figure 4. Arithmetic mean for size of blocks of BTC through time using sliding window

In figures 5 and 6 are presented arithmetic mean through time of ETH using incremental statistics and sliding windows.

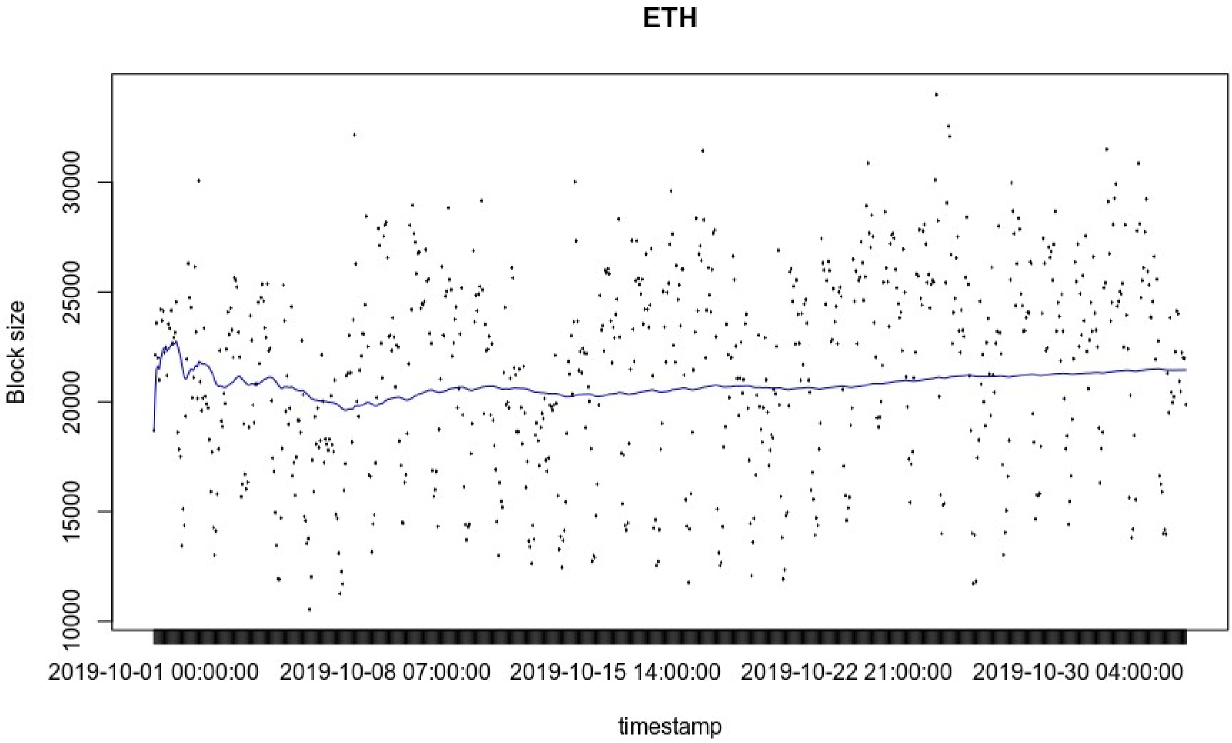


Figure 5. Arithmetic mean for size of blocks of ETH through time using incremental statistics

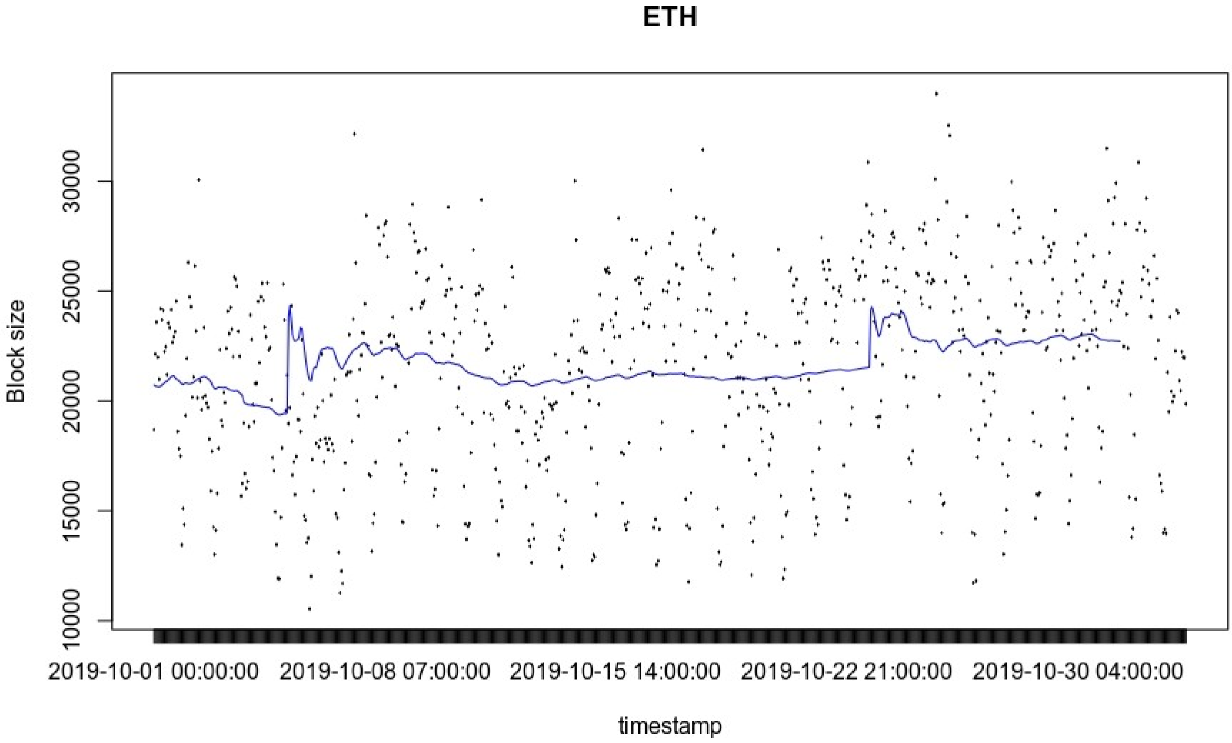


Figure 6. Arithmetic mean for size of blocks of ETH through time using sliding window

3.2. Standard deviation

In figures 7 and 8 are presented standard deviation through time of BTC using incremental statistics and sliding windows.

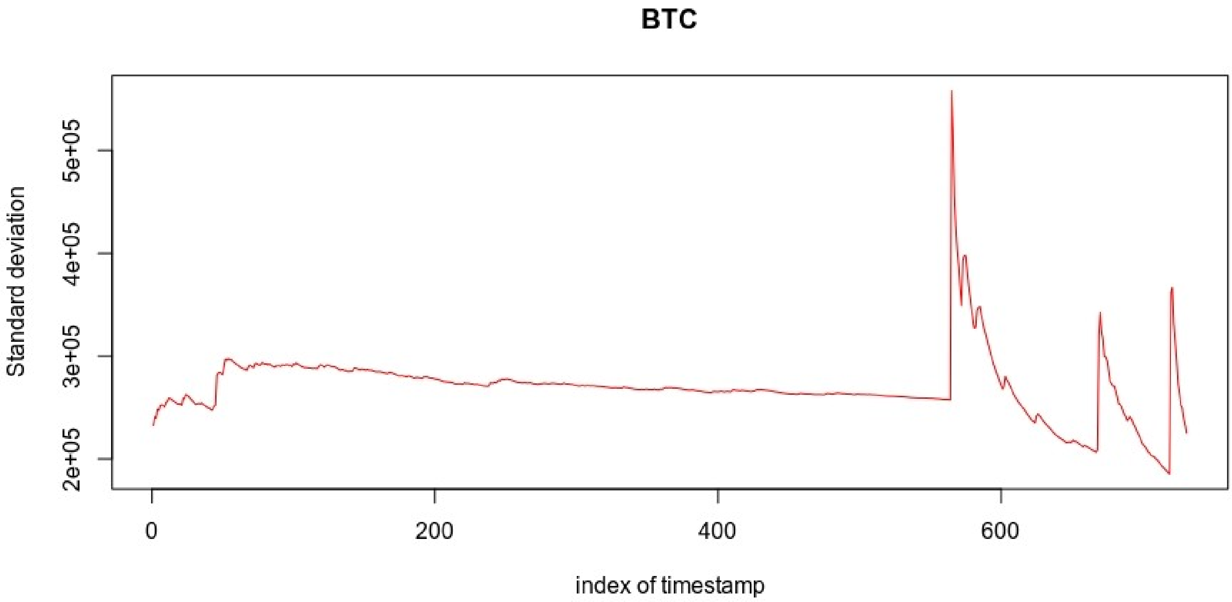


Figure 7. Standard deviation for size of blocks of BTC through time using incremental statistics

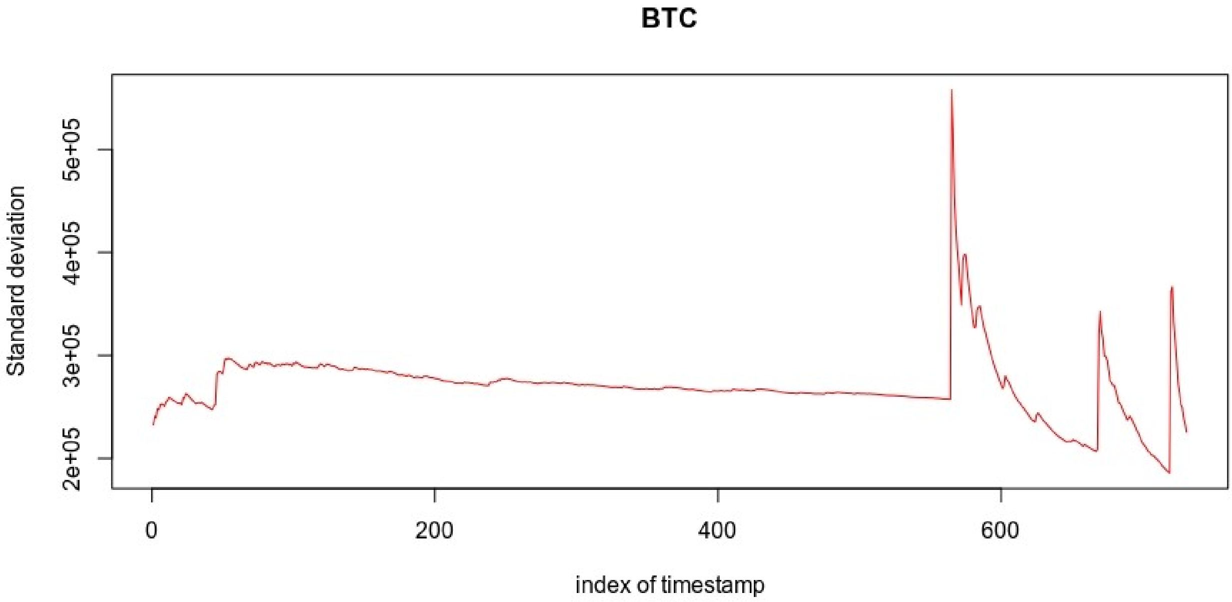


Figure 8. Standard deviation for size of blocks of BTC through time using sliding window

In figures 9 and 10 are presented standard deviation through time of ETH using incremental statistics and sliding windows.

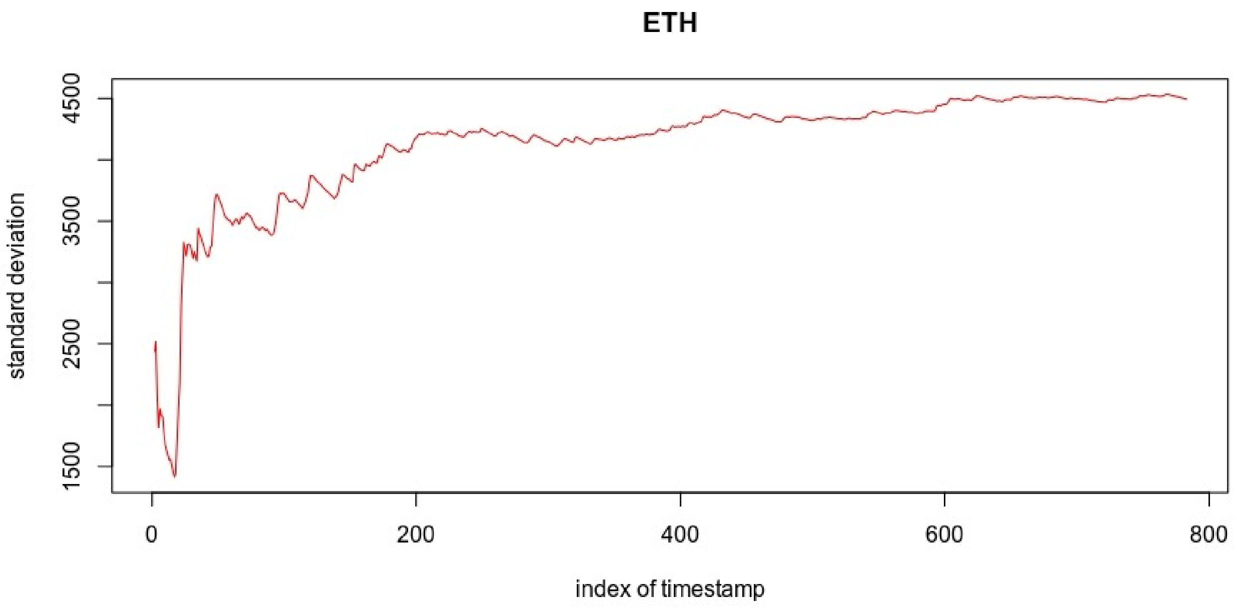


Figure 9. Standard deviation for size of blocks of ETH through time using incremental statistics

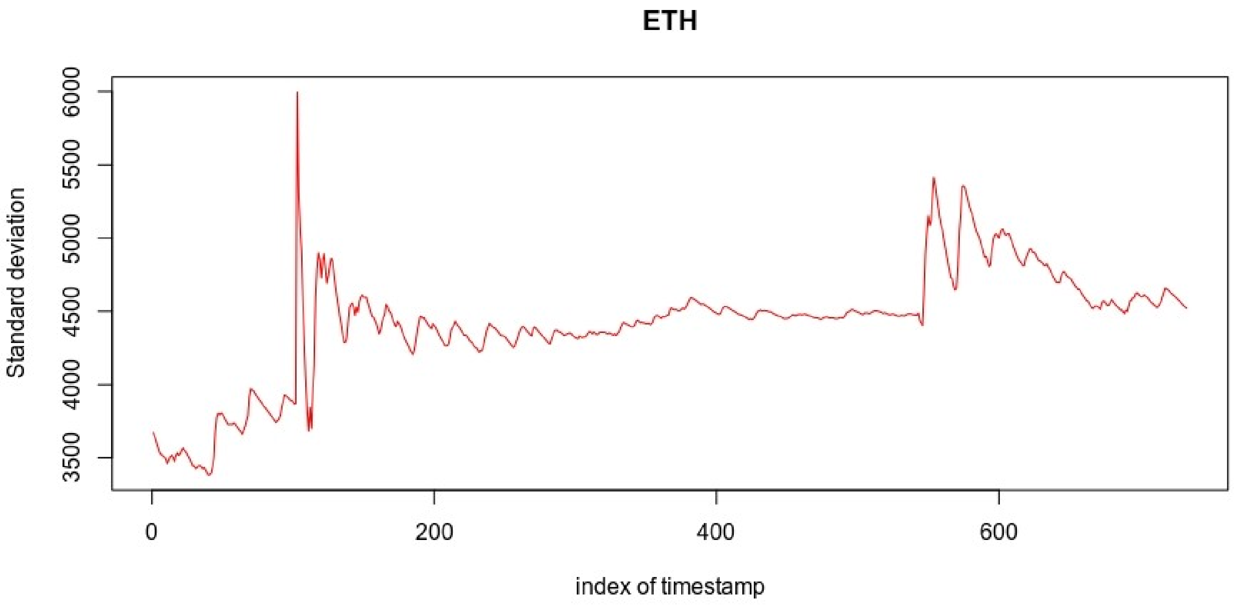


Figure 10. Standard deviation for size of blocks of ETH through time using sliding window

3.3. Autocorrelation

In figures 11 and 12 are presented autocorrelation through time of BTC using incremental statistics and sliding windows.

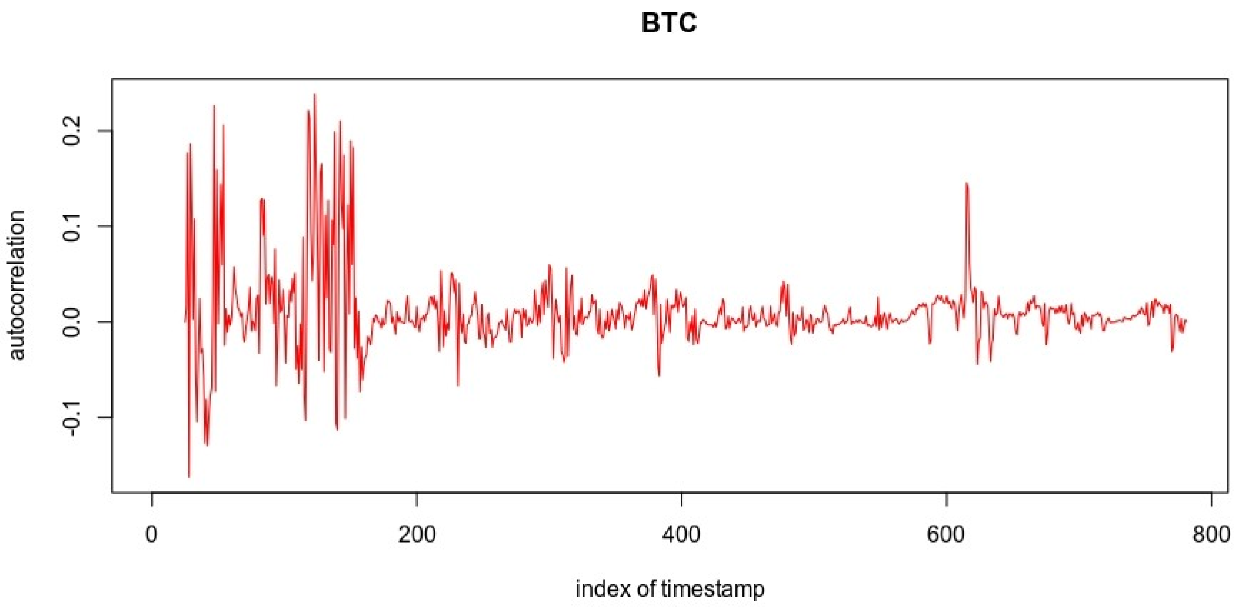


Figure 11. Autocorrelation for size of blocks of BTC through time using incremental statistics

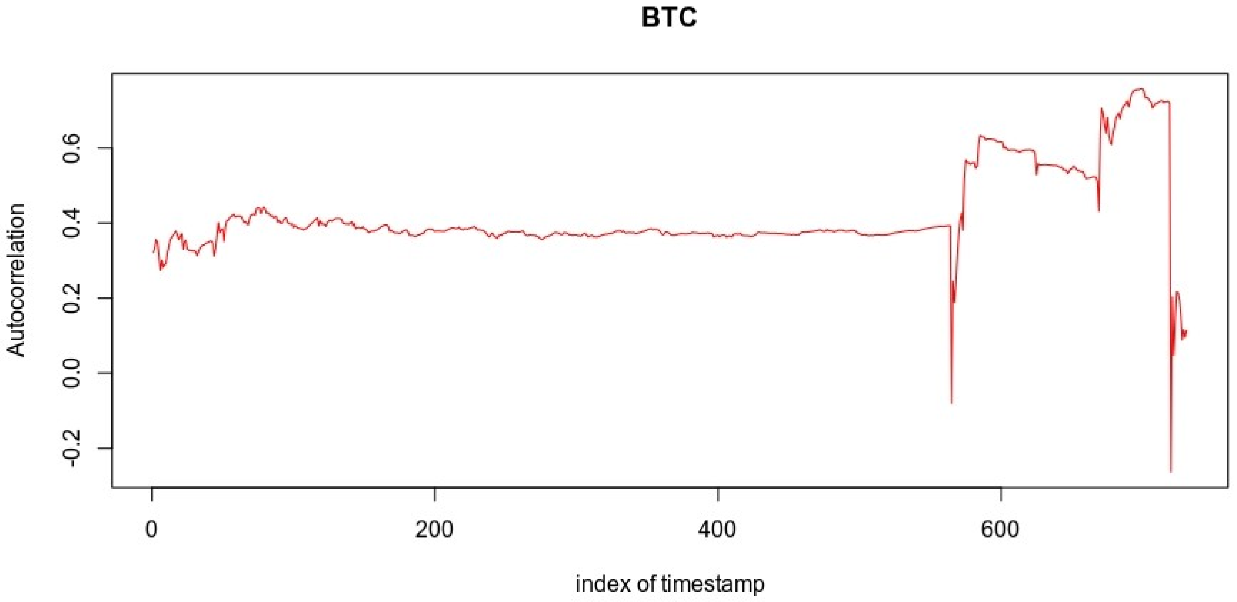


Figure 12. Autocorrelation for size of blocks of BTC through time using sliding window

In figures 13 and 14 are presented autocorrelation through time of ETH using incremental statistics and sliding windows.

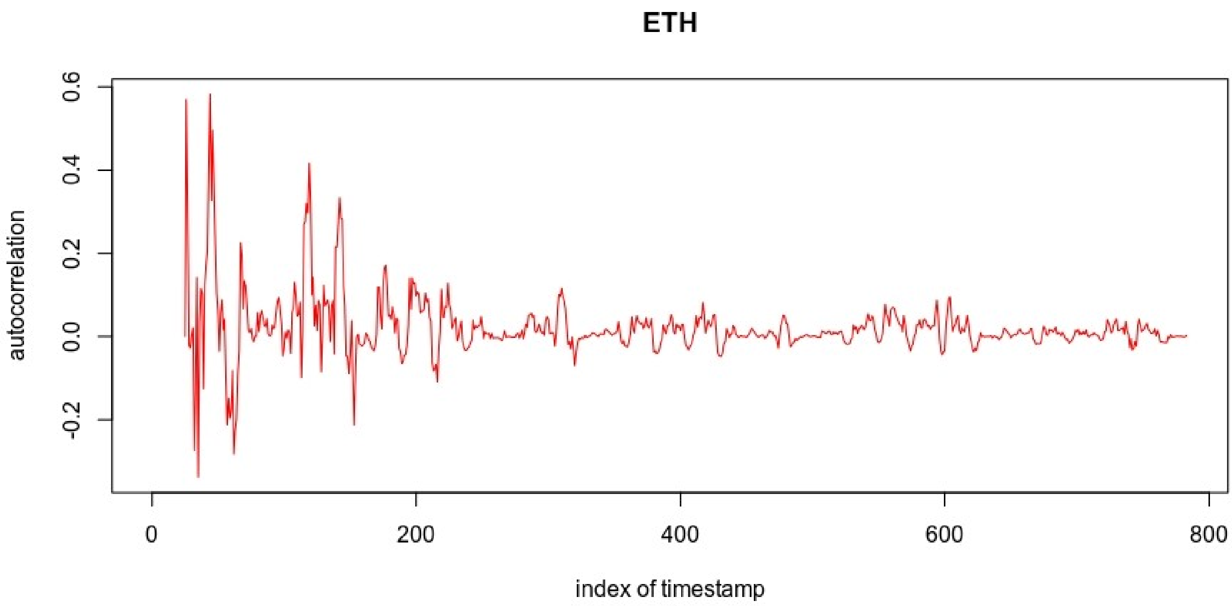


Figure 13. Autocorrelation for size of blocks of ETH through time using incremental statistics

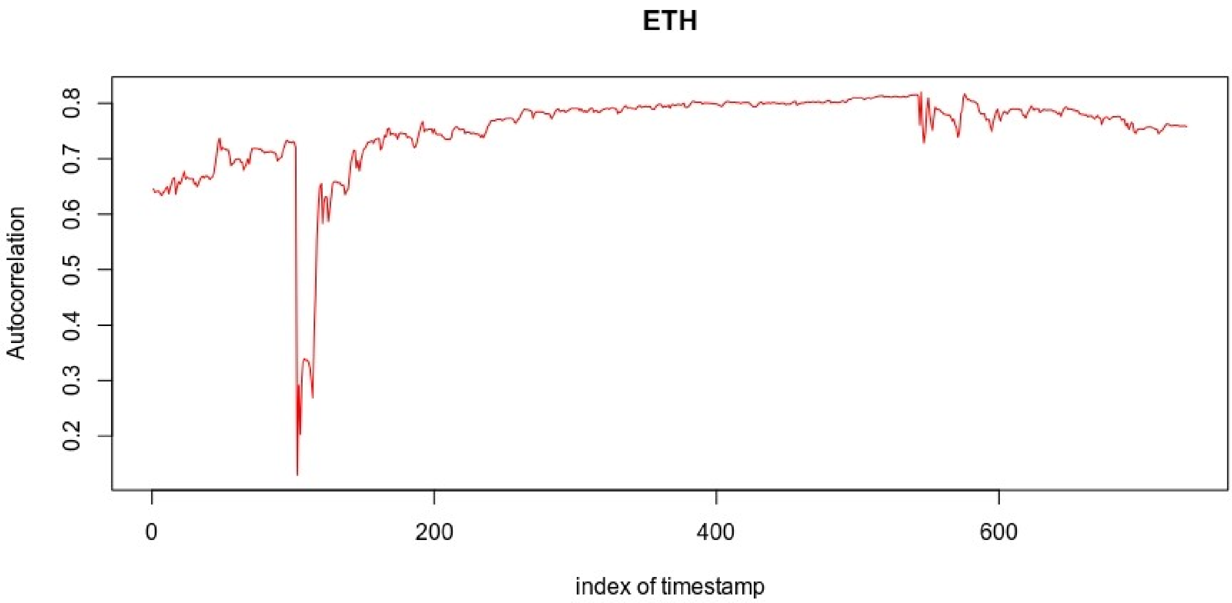


Figure 14. Autocorrelation for size of blocks of ETH through time using sliding window

4. Correlation between streams

In figure 15 is presented correlation between BTC data stream and ETH data stream. In figures 15 and 16 are presented sensitivity of values one to another data streams.

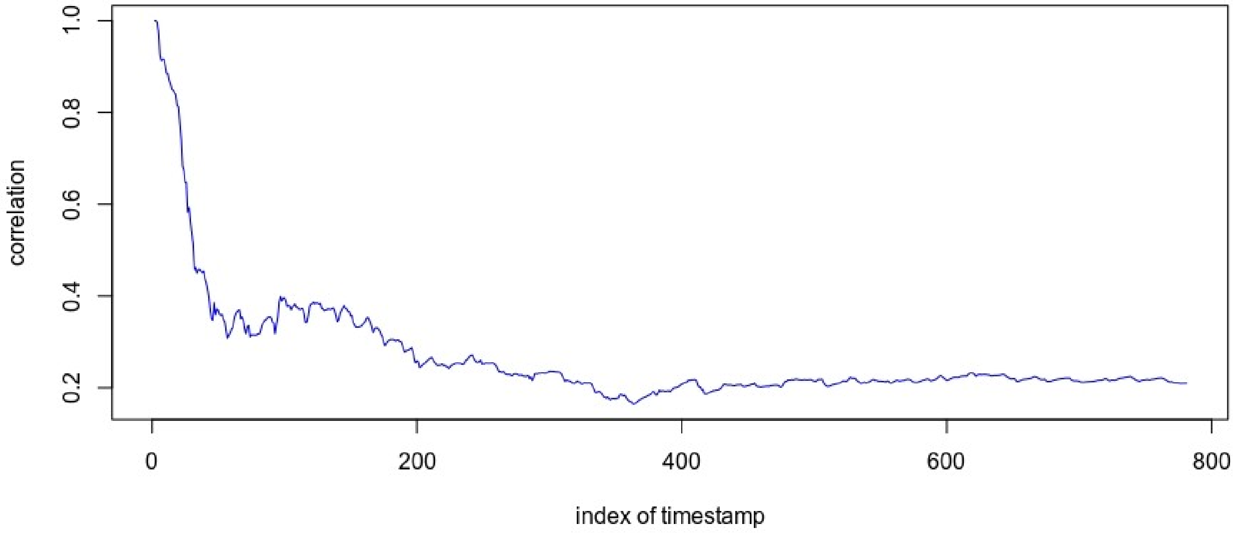


Figure 15. Correlation between BTC and ETH data streams in time

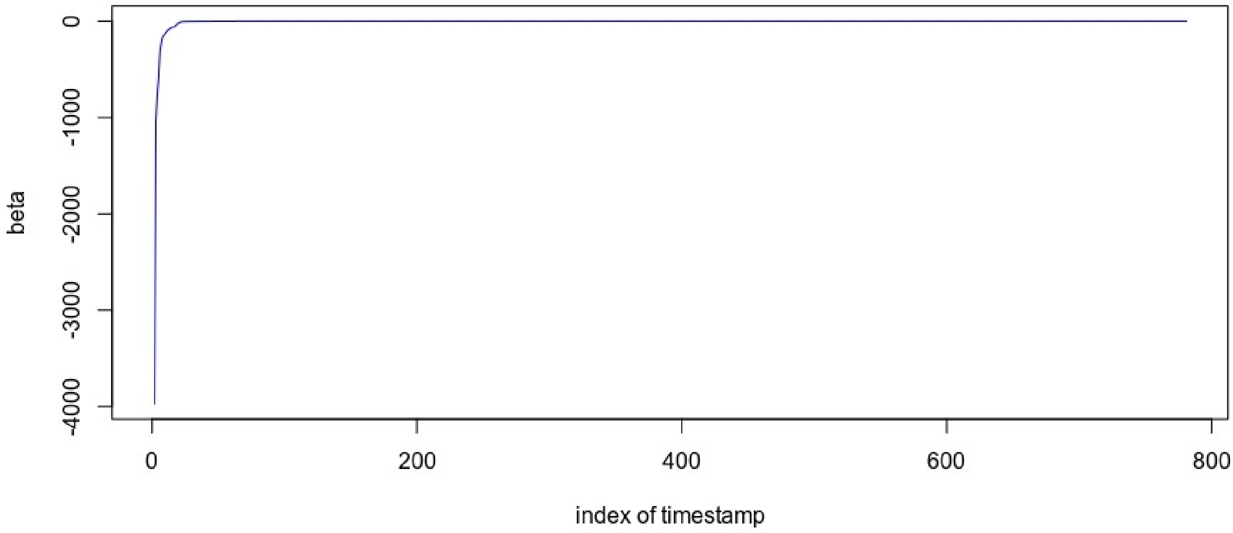


Figure 16. Beta BTC data stream to ETH data stream

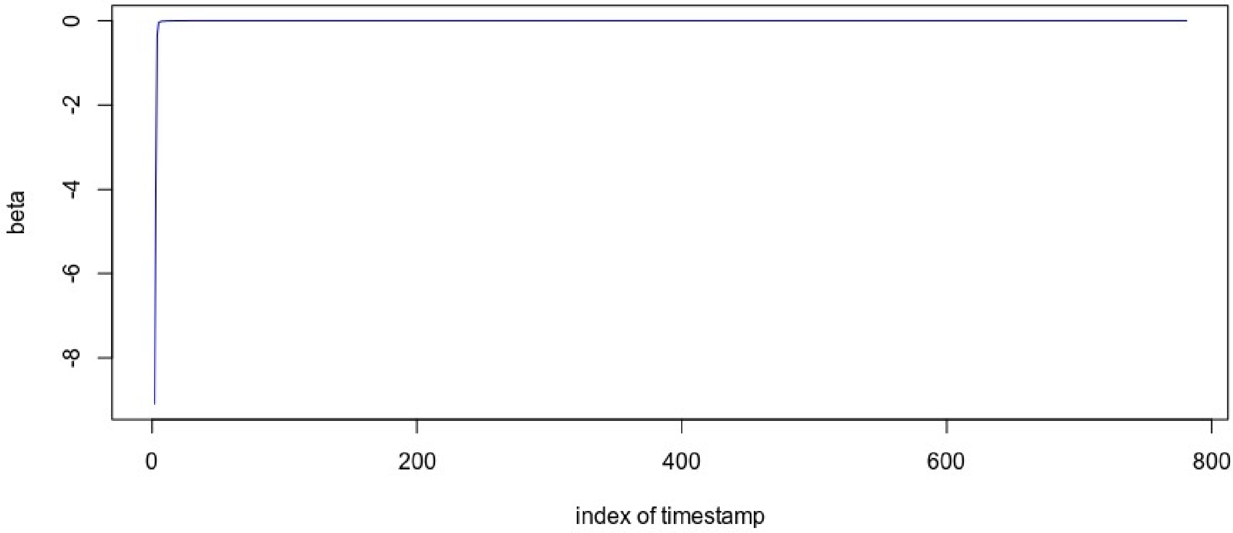


Figure 17. Beta ETH data stream to BTC data stream