NATIONAL RESEARCH

UNIVERSITY HIGHER SCHOOL

OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme HSE University and University of London Double Degree Programme in "Data Science and Business Analytics"

Software Project Documentation

on the topic **Cross Exchange Liquidity Analysis**

Made by 2d year DSBA students
Anna Gertsog and Grigory Zhitniy
Supervisor name:
Lukianchenko P.P.
28/05/2022

Table of contents

Description

This project is created to calculate cross-liquidity coefficient based on input high frequency history trade data between two exchanges. The output coefficient has dimension of 1/sec and describes how powerful arbitrage are.

User experience

The code is separated by three library files: Utils -> consist of all utility functions that are used to render and process data, CryptoLib –> includes api for preparing data, UserLib -> includes api to observe final results.

Data rendering

CryptoLib provides two main functions to prepare data:

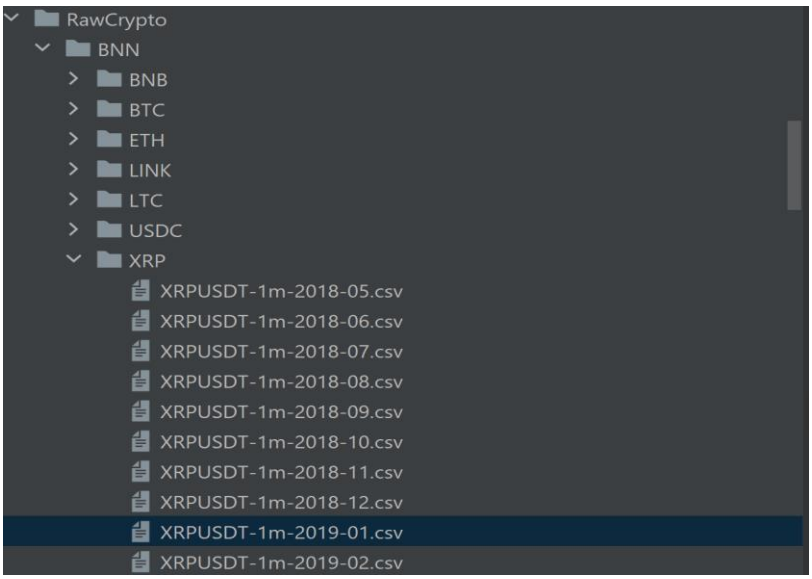
```
def prep_dif_every_exchange():
```
-> prepares intermediate data

```
def write_liquidity_all():
```
-> calculates final liquidity results and write them in CSVs.

Program automatically search for filenames, those functions prepare all possible comparisons between all possible pairs of pairs and maximum time intervals, if data was already prepared it would be skiped, so if input data has been changed -> delete all corresponding intermediate and final CSVs.

Data managing

Program search for files in RawCrypto directory, every subdirectory means exchange abbreviation, every sub-subdirectory means currency abbreviation and includes all history files.

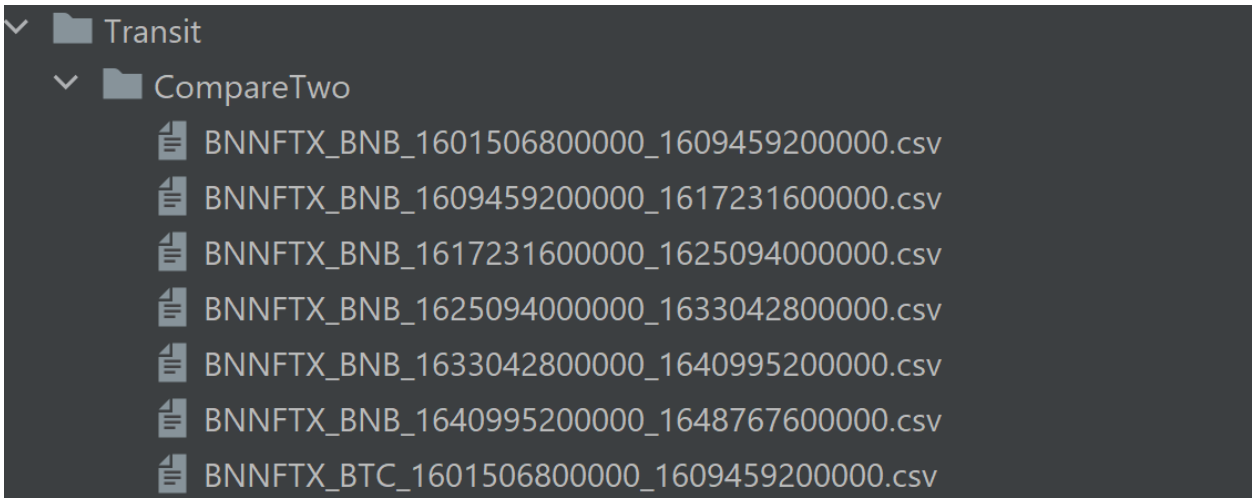


The data should be presented in following format, with the same time intervals form 1 min to 1 hour against the same currency in the same amount. The time should be presented in unix timestamp, the rate is considered to be average of open and close values.

```
,unix,date,symbol,open,high,low,close,Volume BNB,Volume USDT
0,1483228740000,2016-12-31 23:59:00,BTCUSD,974.55,974.55,974.55,974.55,0.02320545,
1,1483228680000,2016-12-31 23:58:00,BTCUSD,974.55,974.55,974.55,974.55,0.0,
2,1483228620000,2016-12-31 23:57:00,BTCUSD,974.55,974.55,974.55,974.55,0.04643379,
3,1483228560000,2016-12-31 23:56:00,BTCUSD,974.0,974.55,974.0,974.55,0.02784628,
4,1483228500000,2016-12-31 23:55:00,BTCUSD,974.0,974.0,974.0,974.0,0.0,
5,1483228440000,2016-12-31 23:54:00,BTCUSD,974.0,974.0,974.0,974.0,0.07423678,
6,1483228380000,2016-12-31 23:53:00,BTCUSD,974.55,974.55,974.0,974.0,0.01949537,
7,1483228320000,2016-12-31 23:52:00,BTCUSD,974.55,974.55,974.55,974.55,0.0,
8,1483228260000,2016-12-31 23:51:00,BTCUSD,974.55,974.55,974.55,974.55,0.0338961386,
9,1483228200000,2016-12-31 23:50:00,BTCUSD,974.55,974.55,974.55,974.55,0.0,
10,1483228140000,2016-12-31 23:49:00,BTCUSD,974.55,974.55,974.55,974.55,0.04549614,
```

Then data is processed by comparing rates of two exchanges into DifTwo dir

In format Exchange1Exchange2_Currency



Then each time data is borrowed for liquidity analysis it is rewritten divided by standard deviation in Transit/Compare Two folder.

Final liquidity results are written in Liquidity folder by every month.

Each file represents comparison between two exchanges of all possible currencies.

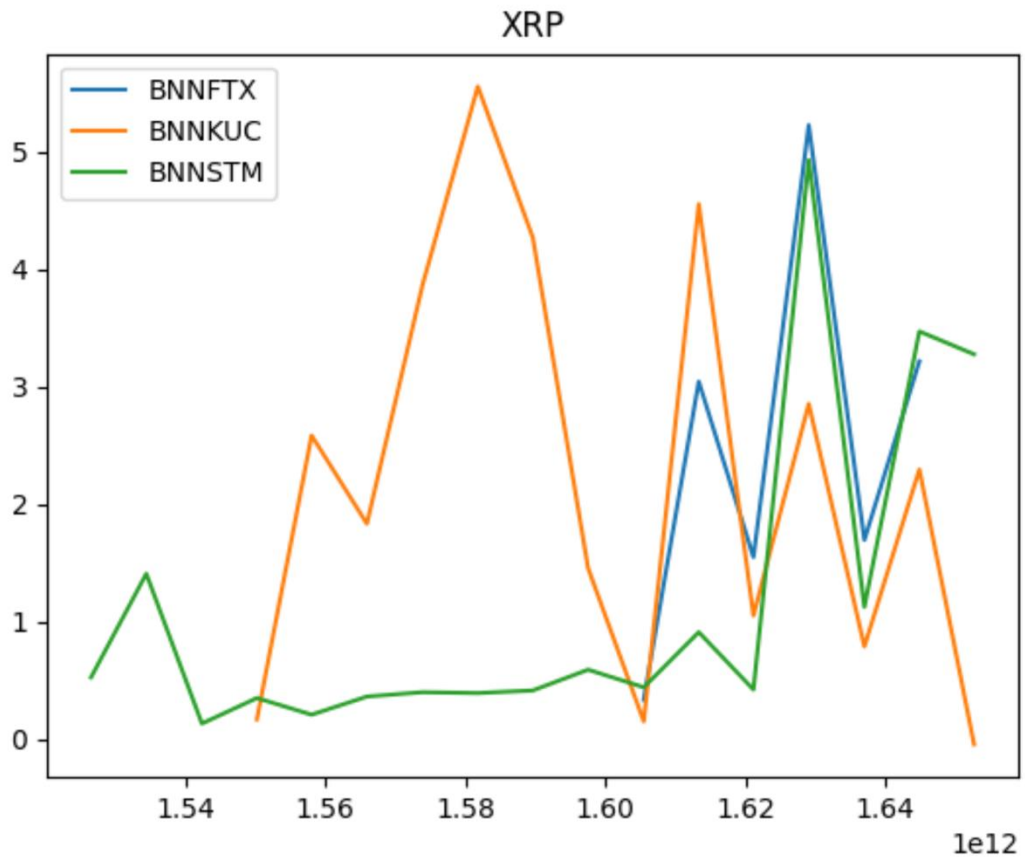
Result representation

UserLib provides function to represent intermediate and results:

```
def track_liquidity(pairs, token):
```

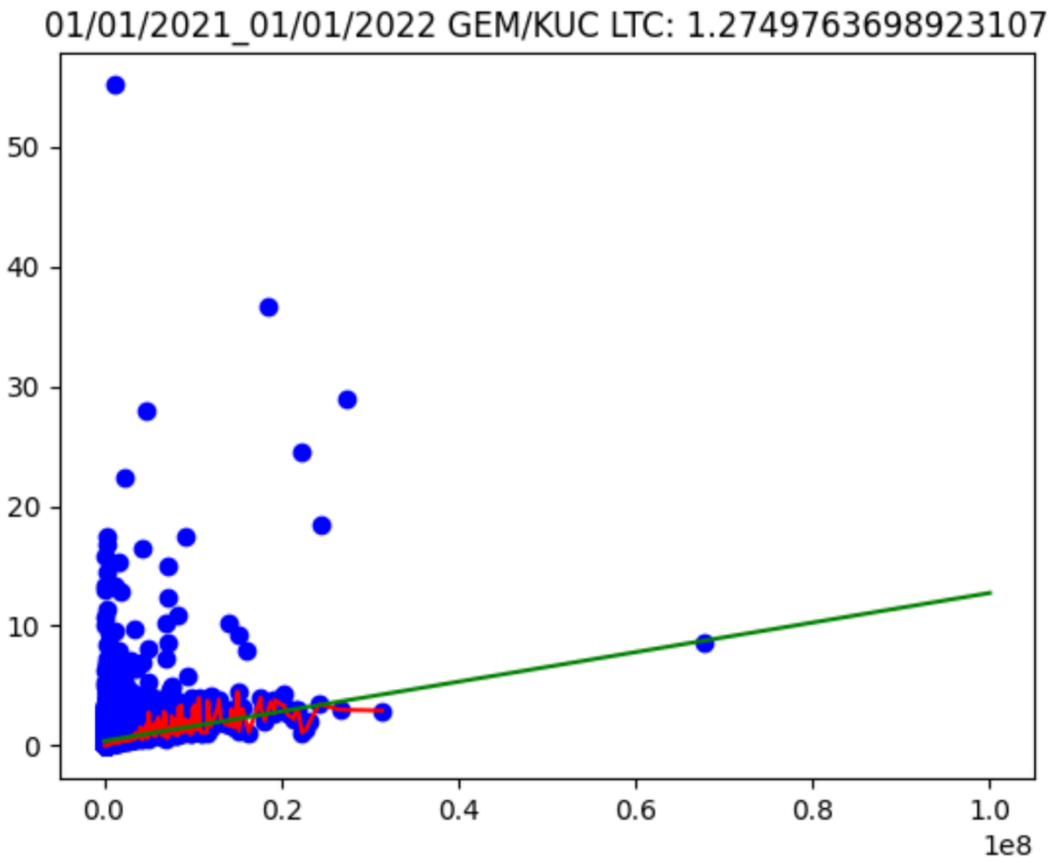
-> shows cross-liquidity fluctuations over the time for pairs of exchanges with particular token.

```
UserLib.track_liquidity([("BNN", "FTX"), ("BNN", "KUC"), ("BNN", "STM")], "XRP")
```



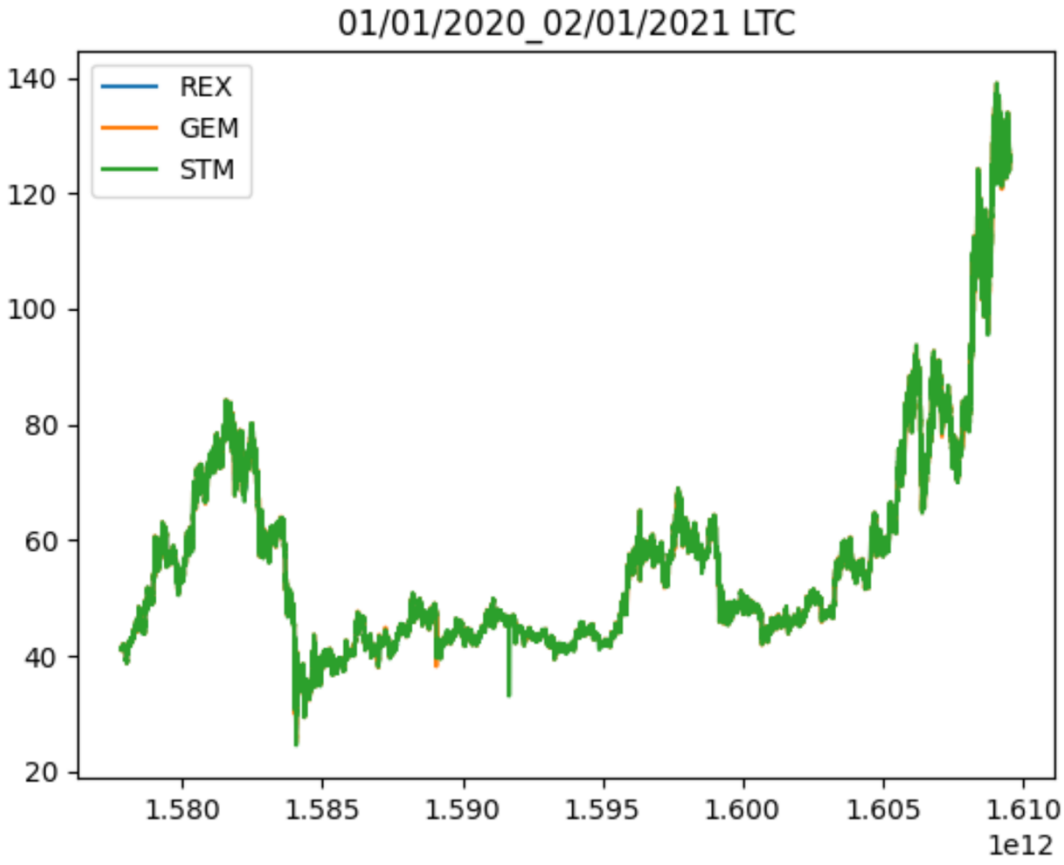

```
def show_liquidity(exchange1, exchange2, token, start_time, end_time):
```
-> shows how coefficient was approximated, between exchnage1 and exchange2 with token and in time interval of start_time and end_time.

```
UserLib.show_liquidity("GEM", "KUC", "LTC", (1,1,2021), (1,1,2022))
```



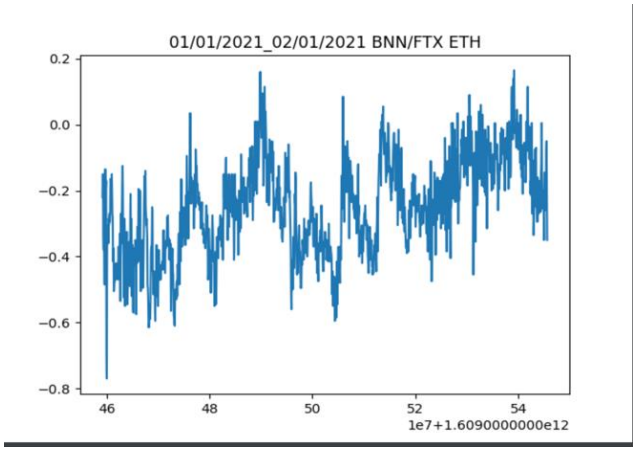

```
def show_rates(exchanges, token, start_time, end_time):
```
->presents rates fluctuations of specific time interval on a list of exchanges for a token.

```
UserLib.show_rates(("REX", "GEM", "STM"), "LTC", (1,1,2020), (2,1,2021))
```

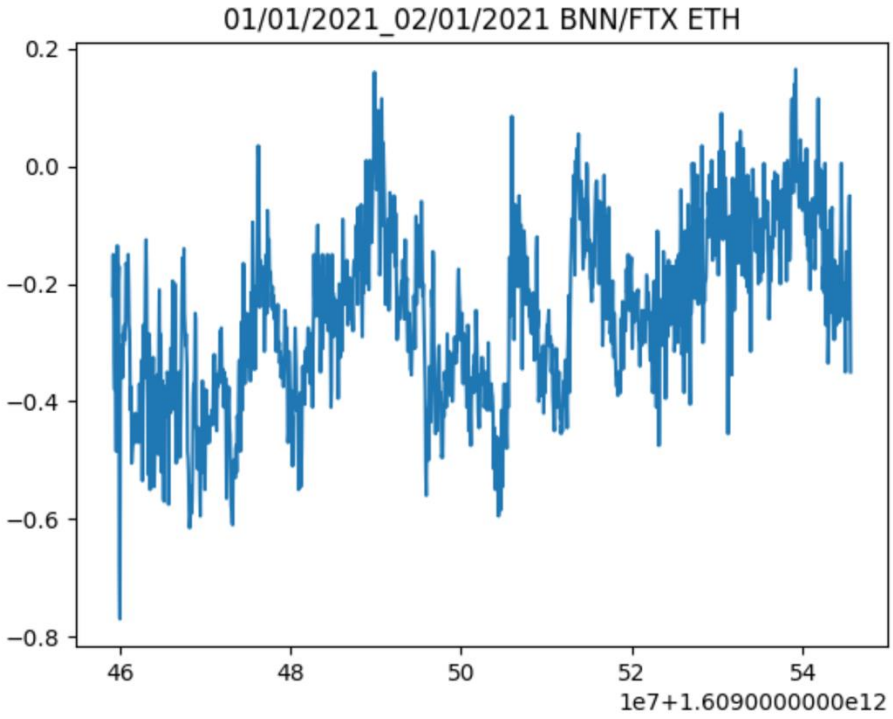


Algorithm

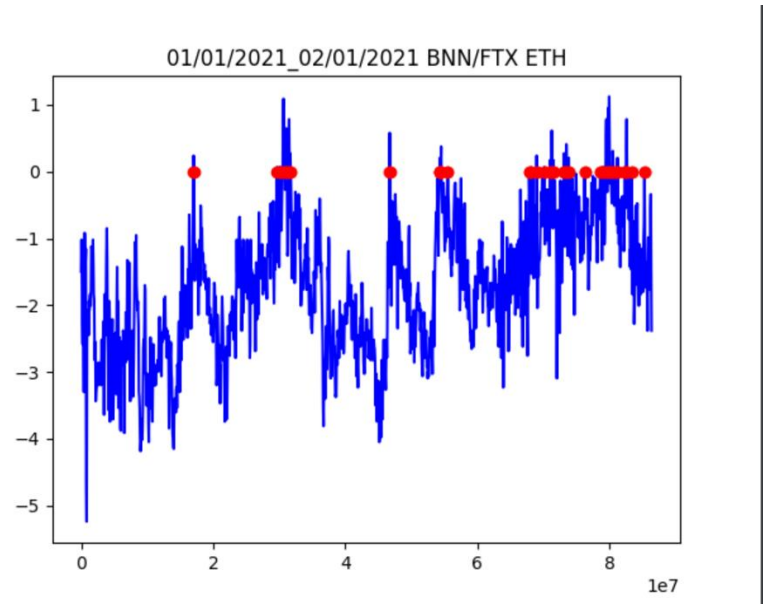
Data processing starts with preparation: synchronizing rates between two exchanges that it represents time, rate on first, on second and absolute difference between them:



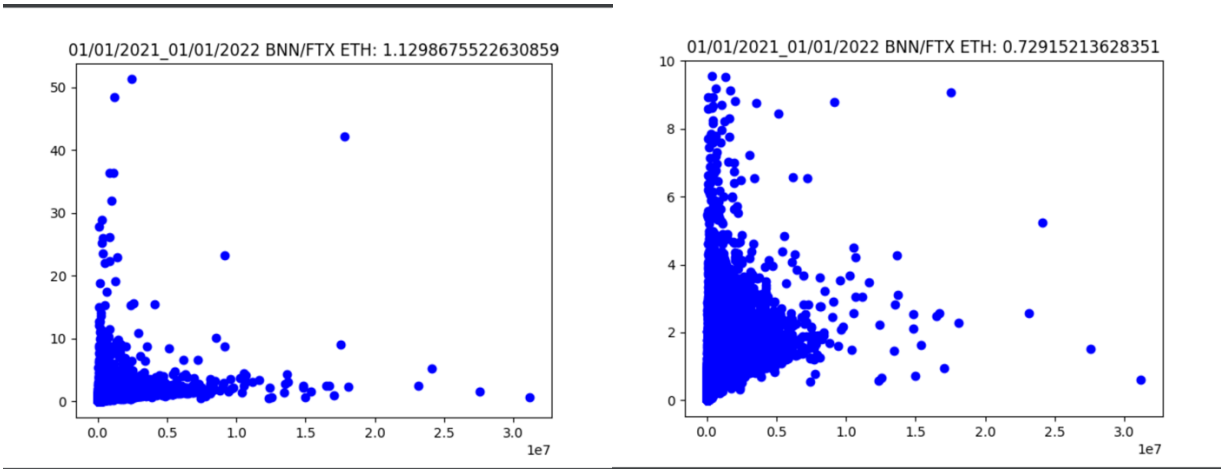
Then difference is divided by standard deviation of difference on observed period.



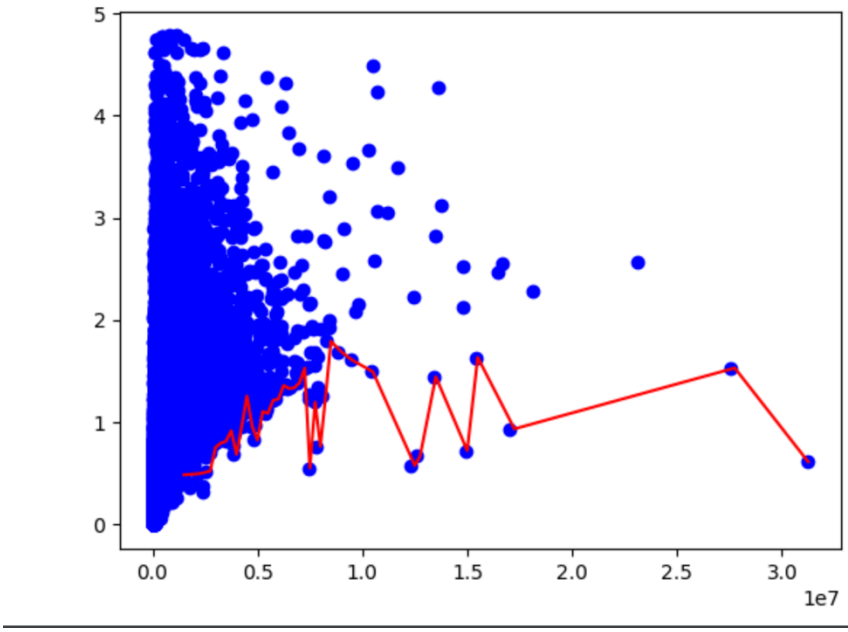
Then we add zeroes – intersections of rate differences and equilibrium (zero) points:



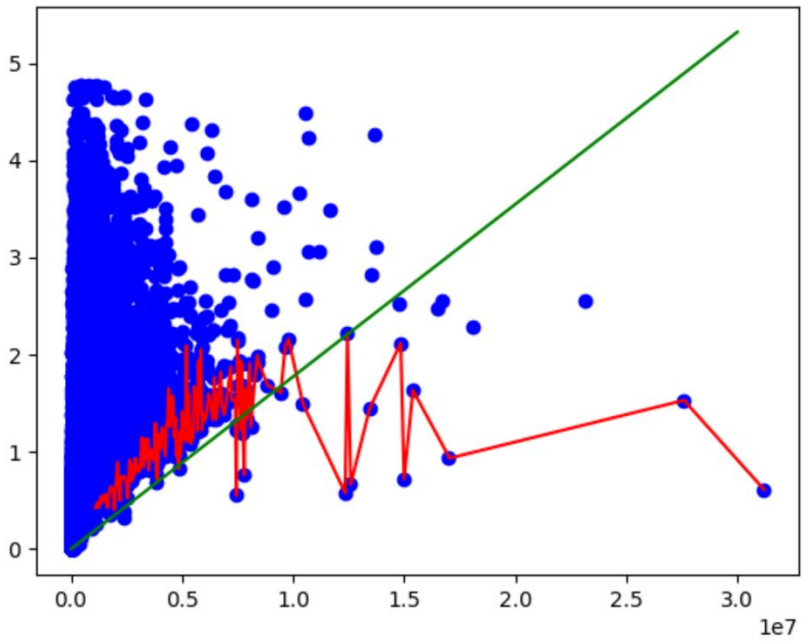
Then we scatter distribution of maximum deviation height among deviation (from zero to zero). Relatively to duration of deviation (time of zero to zero) and cut it by 100 height standard deviations.



Then we find a least points by dividing to 1000 segments among 5*1e7



Then we estimate bottom line by linear regression based on those least points: y = ax + b



The resulting a coefficient is desired cross-liquidity coefficient.