Analysis and Prediction of Cryptocurrencies and Smart Contracts

Coursework for Data Analytics: ECS784P

Queen Mary University of London

April 2018

Mohamed Baddar Muhammad Omar Waqar Daniyal Asad Chughtai

[m.baddar@se17.qmu.ac.uk](mailto:m.baddar@se17.qmu.ac.uk) [m.waqar@se17.qmul.ac.uk](mailto:m.waqar@se17.qmul.ac.uk) [d.chughtai@se17.qmul.ac.uk](mailto:d.chughtai@se17.qmul.ac.uk)

# Abstract:

Last year witnessed the rise and decline of cryptocurrencies led by Bitcoin and Ethereum (Wikipedia, n.d.). Many believe the blockchain (Wikipedia, n.d.) and cryptocurrencies (Wikipedia, n.d.) as its primary use case will change the world. Whereas many others believe it is the new dot com bubble. In this coursework we have collected a huge dataset of the Ethereum blockchain, analyzed smart contracts on a subset of our collected data. And finally done more analysis on the correlation between Bitcoin, the first and most valued cryptocurrency, Ether and other currencies. We extended the concept of (Sifr Data, 2018) which publish the correlation on the last number of days to the weekly and monthly correlation. Which we believed making more sense due to the huge volatility of this market.

# Problem statement:

We wanted to build a proof-of-concept to answer the following questions:

1. Are cryptocurrencies consistently correlated to Bitcoin? What is the point of holding a portfolio of different currencies if that is the case?
2. Can we predict the price of Bitcoin?
3. What is the market share of smart contracts hosted in the Ethereum blockchain?
4. How is the trend of smart contracts changing with time compared to normal transactions in Ethereum blockchain?
5. What are some of the top smart contracts on Ethereum blockchain in terms of value?

# Background:

Many attempts have been made to create different forms of electronic currencies or generally means of exchange that do not necessarily involve banks or a trusted intermediary. David Chaum (Wikipedia, n.d.), an American cryptographer have founded DigiCash (Wikipedia, n.d.) in the Netherlands back in 1989. An innovation way ahead of its time (Amazon was founded in 1994. Ecommerce and online payment was on its infancy). Which later filed for bankruptcy and was sold for assets. Even Paypal (Throwback Thursday: PayPal’s Biggest Days In History, 2015) have first adopted peer-to-peer money before they discarded the idea later.

Many centralized peer-to-peer money innovations were founded and faded for either lack of adoption or regulatory busts followed by government closures.

Then comes Bitcoin.

Inspired by Nick Szabo’s Bit Gold. An idea that he never implemented. A person or a group go by the name Satoshi Nakamoto have invented Bitcoin in 2008. The first peer-to-peer currency that does not have a corporation for governments to target or a data center to seize. That was the digital anarchist dream come true.

Bitcoin was the first adopted implementation of a currency that:

1. Peer-to-Peer
2. Does not need a trusted intermediary to settle transactions
3. Decentralized
4. AND: Cannot be double-spent (Wikipedia, n.d.)

No one have achieved those properties combined before. And that is why Bitcoin is here to stay.

Fast forward to 2015. Ethereum was born.

Bitcoin is a cryptocurrency built on blockchain. Ethereum is a distributed computing open platform and a public blockchain. Where many other companies host their own cryptocurrencies (known as tokens) and smart contracts. It also has a native cryptocurrency known as Ether. This distinction between the currency and the network is sometimes overlooked

Ethereum have created a new era for the crypto world by enabling anyone to create their own currency. And more remarkably by inventing what is known as DAO: Decentralized Autonomous Organizations.

This did not go without pain: The first DAO managed to gather 12.7m Ether (worth around $150M at the time), making it the biggest crowdfund ever (Madeira, 2018). It had been later hacked by an unknown attacker who managed to drain 3.6m Ether (worth around $70M at the time).

## The ICO craze:

Initial Coin Offering (ICO) is an unregulated means by which funds are raised for a new cryptocurrency venture. An Initial Coin Offering (ICO) is used by startups to bypass the rigorous and regulated capital-raising process required by venture capitalists or banks. In an ICO campaign, a percentage of the cryptocurrency is sold to early backers of the project in exchange for legal tender or other cryptocurrencies, but usually for Bitcoin (Investopedia, n.d.).

In 2017. Startups and projects raised $5.6 billion through initial coin offerings (ICOs), according to a new report (Williams-Grut, 2018). That was the biggest catalyst to the hyperinflation of bitcoin and other cryptocurrencies.

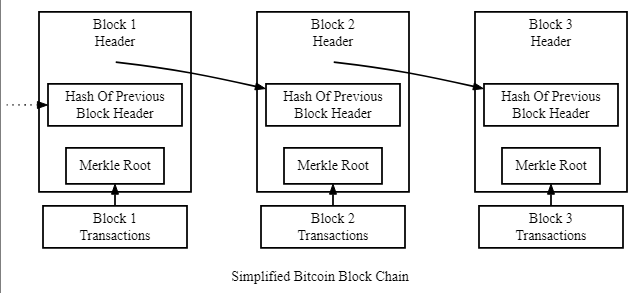
Hence, we chose this topic for our coursework!

# The blockchain:

This section is a brief technical overview on the Ethereum blockchain. We will only explain some definitions and concepts necessary for the reader to follow our analysis. For more information please refer to the Ethereum wiki[[1]](#footnote-1) and the Ethereum white paper[[2]](#footnote-2)

## What is a blockchain:

A blockchain is an ordered and timestamped record of transactions. This system is used to protect against double spending and modification of previous transaction records. (bitcoin.org, 2018)



Ethereum extends this concept (Buterin, 2014). More information on (Stack Exchange, 2017)

## Ethereum Accounts:

In Ethereum, the state is made up of objects called "accounts", with each account having a 20-byte address and state transitions being direct transfers of value and information between accounts. An Ethereum account contains four fields:

* The nonce, a counter used to make sure each transaction can only be processed once
* The account's current ether balance
* The account's contract code, if present
* The account's storage (empty by default) (Ray, 2018)

## Smart Contracts:

Smart contracts are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized blockchain network. Smart contracts permit trusted transactions and agreements to be carried out among disparate, anonymous parties without the need for a central authority, legal system, or external enforcement mechanism. They render transactions traceable, transparent, and irreversible. (Investopedia, n.d.)

## Decentralized Autonomous Organizations

An application of the Ethereum decentralized application platform. The general concept of a "decentralized autonomous organization" is that of a virtual entity that has a certain set of members or shareholders which, perhaps with a 67% majority, have the right to spend the entity's funds and modify its code. The members would collectively decide on how the organization should allocate its funds (Ray, 2018)

## Leveldb:

The database used to store the Ethereum blockchain. (Ghemawat & Dean, n.d.)

## Geth:

The official go implementation of the Ethereum protocol. One of many Ethereum client implementations. (Ethereum.org, 2018)

## Transaction:

The term "transaction" is used in Ethereum to refer to the signed data package that stores a message to be sent from an externally owned account. Transactions contain:

* The recipient of the message
* A signature identifying the sender
* The amount of ether to transfer from the sender to the recipient
* An optional data field
* A STARTGAS value, representing the maximum number of computational steps the transaction execution is allowed to take
* A GASPRICE value, representing the fee the sender pays per computational step (Ray, 2018)

Refer to Appendix D: Output overview for a sample JSON transaction

## EVM:

Ethereum Virtual Machine. the EVM can be thought of as a large decentralized computer containing millions of objects, called "accounts", which has the ability to maintain an internal database, execute code and talk to each other. (Liu, 2018)

## Solidity:

Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM). (The Ethereum Foundation, 2018). Appendix D: Output overview contains a sample solidity code

## Mining:

The word mining originates in the context of the gold analogy for crypto currencies. Gold or precious metals are scarce, so are digital tokens, and the only way to increase the total volume is through mining it. This is appropriate to the extent that in Ethereum too, the only mode of issuance post launch is via the mining. Unlike these examples however, mining is also the way to secure the network by creating, verifying, publishing and propagating blocks in the blockchain.

Mining Ether = Securing the network = verify computation

## Gas Limit/Gas price:

Ethereum is the network, also known as the blockchain. Ether (ETH) is the fuel for that network. When you send tokens, interact with a contract, send ETH, or do anything else on the blockchain, you must pay for that computation. That payment is calculated in Gas and gas is paid in ETH.

The total cost of a transaction (the "TX fee") is the Gas Limit \* Gas Price. (MyEtherWallet, 2018)

## ERC20 Token:

Similar to how the HTTP protocol defined the internet, ERC20 is a protocol that defines a set of commands that a token should implement. ERC20 is not a technology, software, or piece of code. It is a technical specification. If a token implements the spec, it is an ERC20 token (Siebel, 2017)

ERC20 tokens are most widely used for companies to issue their own cryptocurrencies and smart contract assets.

## JSON-RPC:

JSON-RPC is a stateless, light-weight remote procedure call (RPC) protocol. Geth also operates as a JSON-RPC server. (Ethereum.org, 2018)

# The Data:

We wanted to analyze Ethereum as it hosts 91% of token sales at the beginning of 2018 (AminCad, 2018). The next logic step is to find a dataset that contains proportionally sufficient transaction volume to do our analysis on.

Unfortunately, we could not find such a dataset from any online source. The available datasets only track the daily prices against USD.

That was a shocking finding!

We then reasoned that well, someone must have thought about it, so we should easily be able to find some code on github. Then all we need to do is just run it and get our dataset.

We found a github repository (Miller, 2017) that extracts the block information and transactions into a Mongo DB. We wanted to get a year’s worth of data. We first need to have a full node (The entire blockchain. More than 5 million blocks, more on that later). That step took a few days to complete. Downloading and verifying about 60 GB worth of data on a Windows 7 machine.

The first attempt to run the code failed!

The code was based on python 2.7. It also uses outdated python and Ethereum modules. After doing some research we managed to replace the outdated modules. One of the core dependencies was pyethereum[[3]](#footnote-3) and, well, it does not run on Windows! It has to be built from source and it has Linux dependencies. So, we ended up with a blockchain on Windows and a code that must run on Linux.

To export the blockchain to Linux we need a machine with at least 60GB of diskspace and a couple of days to re-verify transactions again. That was a time we did not have!

The only solution was to keep the blockchain on Windows and run the code on a Linux VM that connects to our Windows box.

Due to time constraint we ruled out using AWS or similar cloud VM. We used a local virtualbox Ubuntu 16.04 machine.

We wanted to get the data of around 1 million blocks. The code is designed to crawl from the very beginning. We customized it, and we finally managed to get the code running.

It was communicating with the blockchain via http requests. 1 at a time. The amount of data we needed (around 2.38 m blocks) would have taken more than 660 hours at 1 block/sec. speed! That was the current code capacity.

Again, a hurdle we needed to overcome.

We found a pull request[[4]](#footnote-4) on that repository (the repo owner does not maintain it any more) that claimed a higher throughput. It had a slightly different approach to crawl the chain however it had many defects we needed to fix (that pull request was never merged to the repo master branch.So, no one tested it apart from its creator).

We finally got our code running and progressing in a reasonable time. About 90% reduction from the original time estimation!

Data Wrangling accomplished!

# Data exploration and extraction:

The data wrangled in the previous part was stored as a MongoDB document database. To explore it, we had to use a MongoDB server, pymongo (python) package and Studio 3T for MongoDB. MongoDB server was installed and initialized on local machine for ease of use. Initially Studio 3T was used to get a visual representation of the documents in the database, their format and size. Due to the data being too large, a subset of the data was used to formulate the queries that would extract data.

Our first approach was to use pymongo and write code in python to query the required data from the database. This approach worked well on the subset of the data and we were able to extract and save query results in json files. These files were then used for plotting and visualizing the different metrics. However, on the complete dataset, this approach failed because the queries would take too long and either the operation would time out, or the system would run out of memory. Just as an example, a query to extract contract creation transactions from all the blocks in the dataset took about 9 hours to complete. Other queries either took longer or failed after 7-8 hours of operation.

After failed attempts to optimize the queries in various ways, we resorted to using Studio 3T software for data extraction. It was able to optimize the queries internally and gave out results in the form of MongoDB documents much quicker. These documents were used in the python code for plotting and visualization.

Following is some information about the dataset:

|  |  |
| --- | --- |
| Time span of the data set explore: | December 2016 – July 2017 |
| Total Number of blocks: | 1,169,618 |
| Total Number of Transactions: | 26,798,548 |

The scope of this report is restricted to analyzing smart contracts on the blockchain so our data extraction mainly revolves around them.

A brief explanation of the database queries and the plotting code refer to Appendix B: Data extraction and visualization

## Smart contract market share

The market share is determined by the amount of Ethereum traded in transactions involving contract addresses and those with normal addresses.

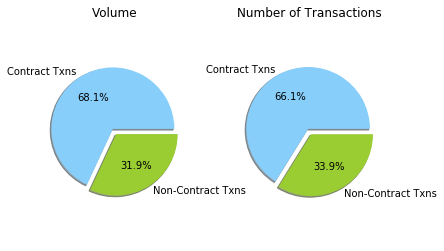


Figure : Market Capitalization as of July-2017

The figure 4, above shows that smart contracts hold a 68.1% in terms of Ethereum value and 66.1% in terms of number of transactions done through smart contracts. Considering Ethereum is the main platform for smart contracts, it is expected to hold a higher share.

## Smart contract creation trend with time:

Number of contracts created each month would tell us about how the idea of smart contracts is gaining or losing traction.

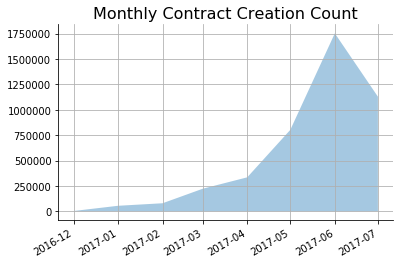


Figure : Contracts created each month

Figure above shows a growing trend in the contracts created each month. This growth shows an exponential trend, meaning the smart contracts are growing popular with time. The magnitude of this growth can be explained by the growing popularity of cryptocurrency in general as well as smart contracts. As our data starts from December 2016 and ends in July 2017, the values for these months are not accurate due to incomplete data for these months. We did not exclude these months from our plot for the sake of completeness.

## Smart contract transaction volume compared to normal transactions:

By comparing transaction volumes in monthly time interval, we can get some insights on how the smart contract fare against normal transactions and any changes in the trend if any.

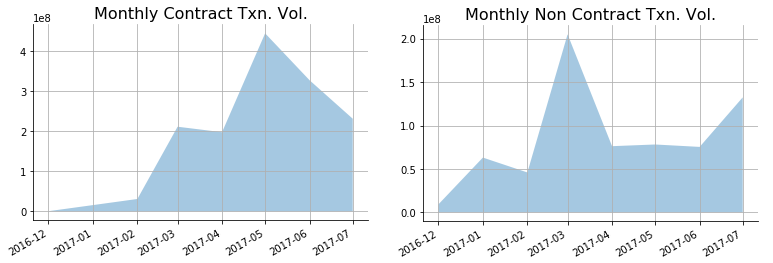


Figure : Comparison between smart contract and normal transaction volumes with time

From the y-axis scale it is clear that smart contracts have gone up considerably in comparison to normal transactions. Up until February 2017 normal transactions were higher in volume then smart contracts. In March 2017 smart contracts superseded normal transactions in terms of volume and since they have been on the rise. Another behavior shown in the graphs is that during the period March-July 2017, the smart contracts show a peak in the May 2017 while normal transactions dipped and remain low throughout this time. This shows that there was more activity in smart contracts then normal transactions although this was a period where cryptocurrency was very popular.

## 4 Notable smart contracts:

The following table shows the top smart contracts with high volume of transactions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rank | Address | Name/Tag | Volume | | No. of Transactions |
| 1 | 0x209c4784ab1e8183cf58ca33cb740efbf3fc18ef | Poloniex\_2 | 12362766.225237455 | 587990 | |
| 2 | 0x7727e5113d1d161373623e5f49fd568b4f543a9e | Bitfinex\_Wallet2 | 12072289.758089611 | 64061 | |
| 3 | 0xfa52274dd61e1643d2205169732f29114bc240b3 | Kraken split contract | 9981656.455467101 | 211251 | |
| 4 | 0x6fc82a5fe25a5cdb58bc74600a40a69c065263f8 | (Possible Spam) | 4213587.620218947 | 29604 | |
| 5 | 0xe94b04a0fed112f3664e45adb2b8915693dd5ff3 | Bittrex\_2 | 3680807.621031744 | | 276257 |

The above table show some of the top contract addresses as on July 2017. The names or tags have been taken from an online Ethereum explorer <https://etherscan.io/>.

As can be seen from the table, all the top 5 smart contracts, excluding one, belong to cryptocurrency exchanges. This could be because of high amount of trading in cryptocurrencies during this period, a lot of transactions have been happening through these cryptocurrencies.

## Challenges with data extraction:

Extracting data and exploring it has its challenges. In our case after weeks of effort to extract meaningful data fields from the dataset, we discovered that some of the dataset entries were not accurate. For example, some of the transactions were wrongly classified as smart contract transactions. Ideally, we were supposed to fix our crawler code and re-crawl the data to get the data right but due to time constraints and taxing process of data crawling we had to settle for filtering out the anomalies at the data extraction stage. Overall these anomalies have not much of an effect on our results as they result in the same conclusion.

# Further Analysis:

In our quest to understand the hugely volatile price movements of cryptocurrencies. We thought that understanding the correlation between Bitcoin and Ethereum and applying some timeseries analysis is the way to go. We had that impression that every cryptocurrency (There are more than a thousand of them!) is correlated with Bitcoin. And when Bitcoin sinks, every other currency follows.

We perceived Ethereum and Bitcoin should have different fundamental drivers and hence we did not buy into the above common presumption. Bitcoin is the most traded currency and the one with the most trading pairs on every crypto Exchange out there. Whereas Ethereum is an open platform and should be driven by its tokens. Whenever one interesting project succeeds. That adds to the entire network value and should reflect on price.

We also know that part of the correlation between Ether and Bitcoin (BTC) is that many tokens cannot be directly liquidated (sold for a fiat currency such as GBP). So if a token holder wants to cash out. They first need to sell it for Ether, the from Ether to either fiat or BTC depending on their location. In that context BTC is the most liquid crypto.

We looked at the cryptocurrency correlation matrix of (Sifr Data, 2018) and though that 90 days is too much time in the crypto world. We should do a finer analysis in order to get any insight out of correlation.

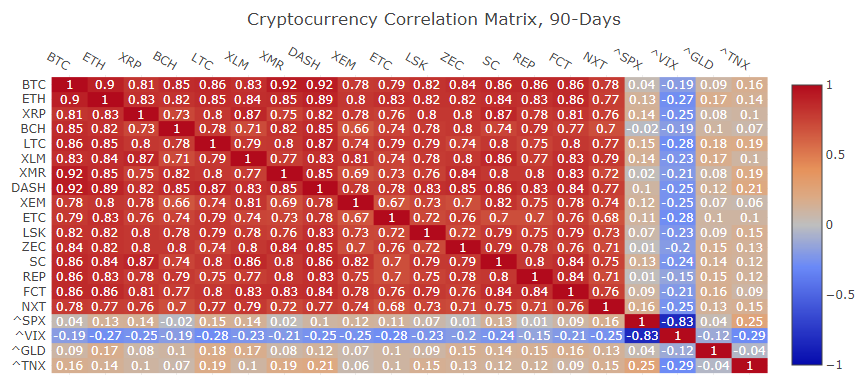


Figure Sifr Data 90 Days Crypto Correlation Matrix. As of 12/4/2018

We found a great Jupyter notebook by (Ma, 2018) that replicate the correlation matrix above on even a longer time interval. Along with other very useful tools that we based this part of our analysis on.

The code uses a Kaggle dataset (Vent, 2018). It tracks the daily price of 13 cryptocurrencies since 22/2/2018. it contains historic open, high, low, close, trading volume and market cap. We adopted the 7 currencies she picked, possibly by highest market cap at the time of coding. And we eventually focused on BTC and Ether.

Note: The analysis would have been more insightful should we had hourly prices.

Since this is a preprocessed data source. We thankfully did not have any issue loading the data. Refer to Appendix C: Analytics Jupyter Notebook Code Overview for a sample of the dataframe.

## Market Cap:

The market cap for Ethereum came second only to Bitcoin, going to third number for a short period of time in January 2018, then rapidly bouncing back. We can see the huge boom and bust of BTC followed by all others at the end of 2017.

It is worth noting that the majority of highly traded Ether token had a delayed trend of around one month since the downtrend of BTC started. Most have peaked in January and started falling.

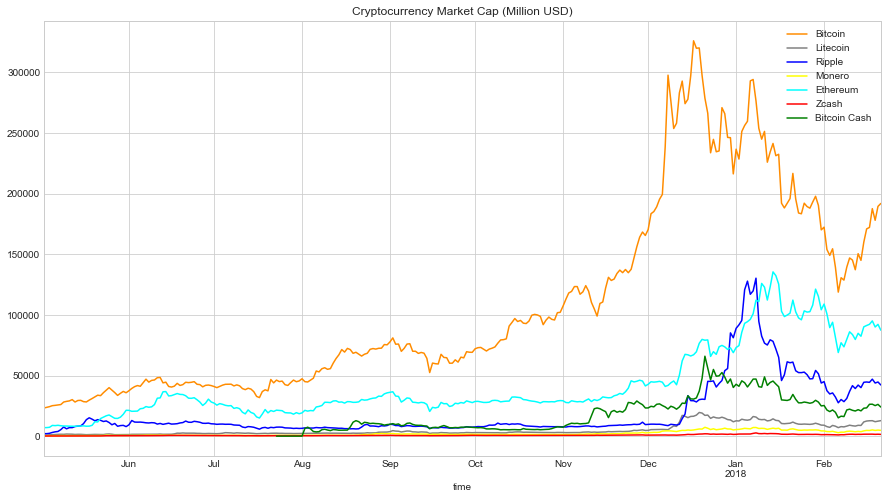


Figure Crypto Market Cap since May 2017

## Transaction Volume:

The transaction volume graph is denoted in million. The graph shows that Bitcoin is by far the leading cryptocurrency in terms of volume traded. Interestingly, there are some points, where Ethereum overthrew Bitcoin to be the leading cryptocurrency in terms of volume during the months of June till August 2017. We can see the spikes in the graph (denoting a rise in volume) for Bitcoin are way ahead of every other cryptocurrency. After December 2017, the gap between Bitcoin and other cryptocurrencies widened and no one could catch up in terms of volume.

Then came the bust. BTC is trading at less than $7000 at the time of writing.

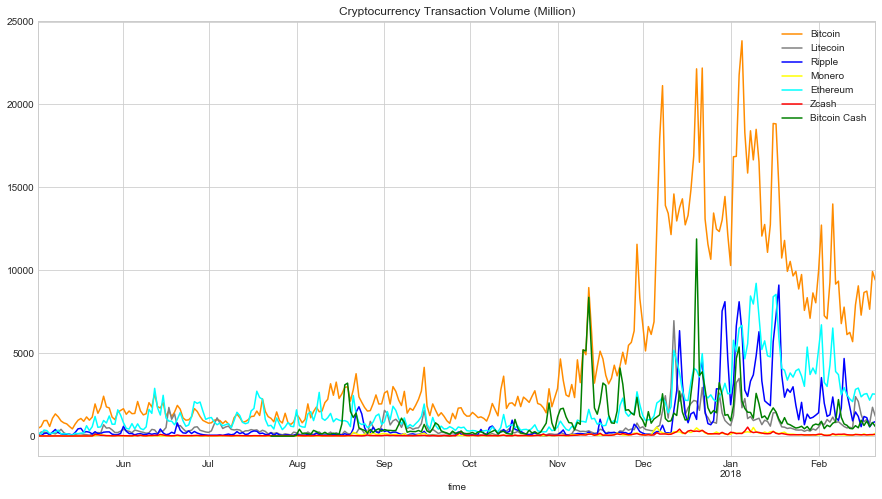


Figure Crypto Transaction Volume

Price Spread:

This chart tracks the differences of daily highest and lowest price of the top 7 currencies since 2017. Remarkably, Bitcoin had a price spread of 4000 USD, which could have given investors a chance to earn this amount in a single day for each BTC they trade. Just two out of the seven cryptocurrencies managed to touch 1000 USD price spread. Ethereum saw an increase in price spread after December 2017

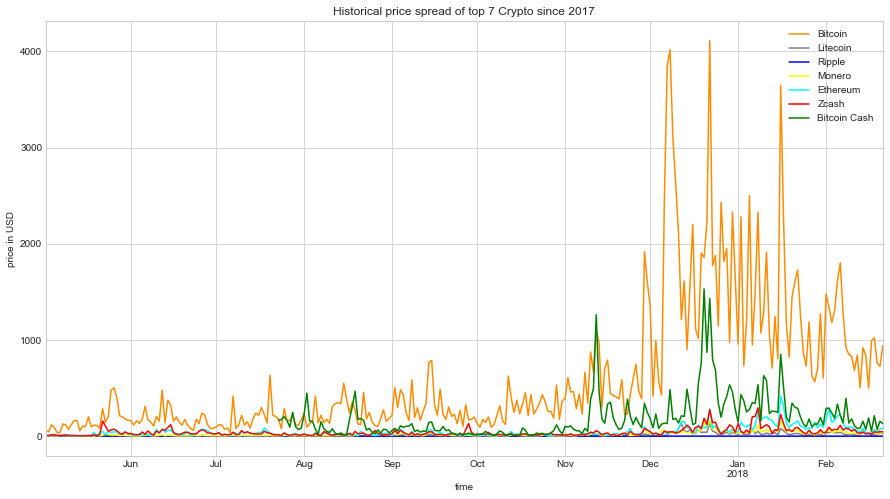


Figure Crypto Price Spread: Since January 2017

## Historical daily average prices:

The daily average price was calculated by adding up the highest price, lowest price, opening price and the closing price, then dividing the sum by 4. Fluctuation could be a measure of volatility as this averaging should work as a smoothing operator. The plot shows how Bitcoin is the highest in absolute value and volatility. Ether’s average peaked in January 2018. That was in line with the token peak mentioned earlier.

## 

Figure Historical daily average prices since 2017

More smoothing applied by plotting the 5 days moving average of BTC and ETH. We can see how Ether looks flat compared to BTC except in January.

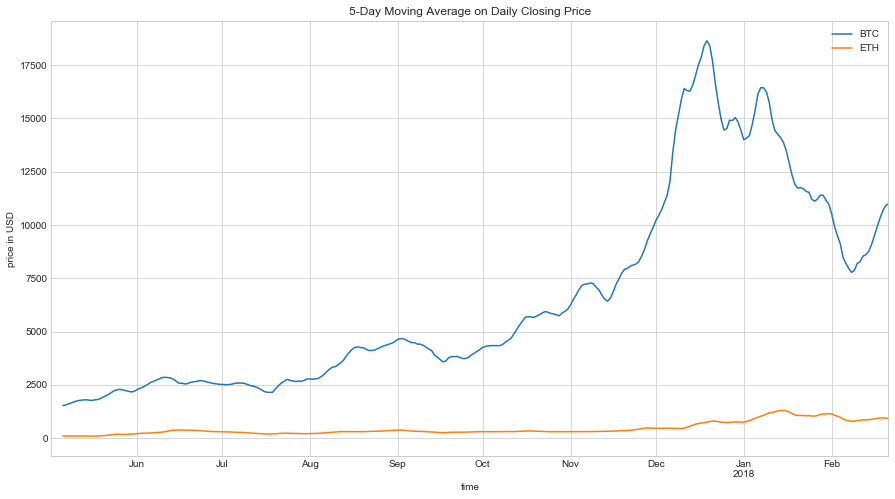


Figure BTC and Ether 5 days moving average

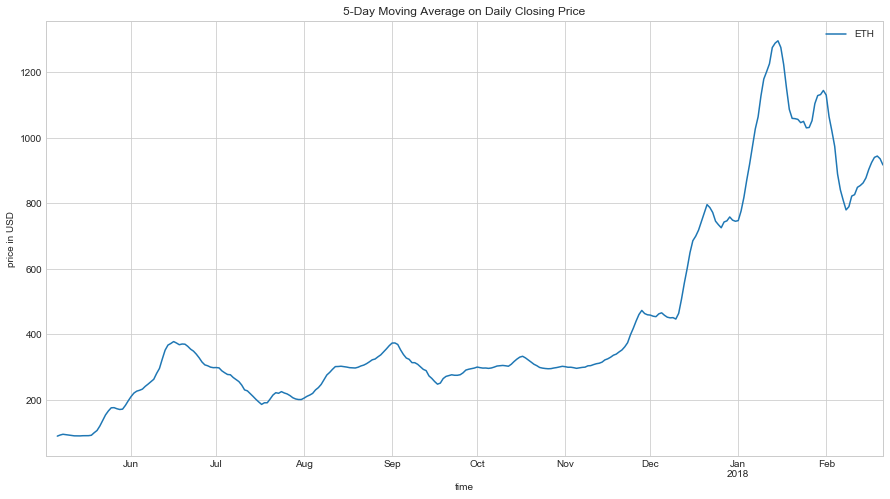
Ether alone follows a very similar trend.

Figure Figure 11 Ether 5 days moving average

## Correlation:

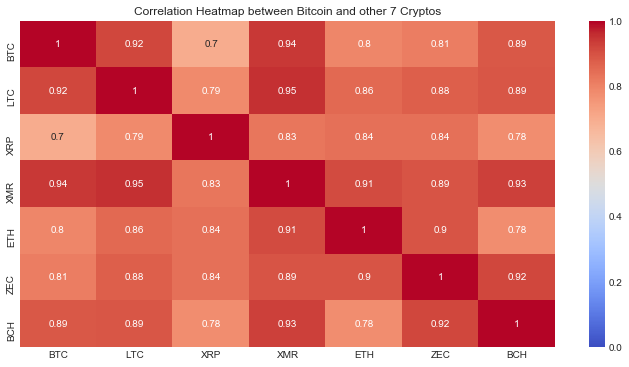
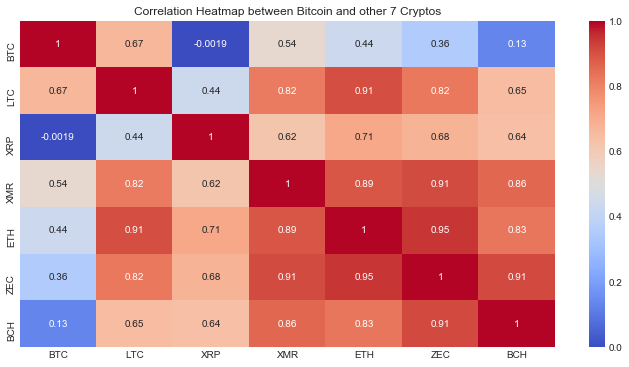
We started by looking at the Pearson correlation heatmap of the 7 currencies for the entire analysis period. We trimmed the data to start from May 2017. The long timespan tells us that the correlation between Ethereum and Bitcoin is quite high (0.8).

Figure Pearson correlation heatmap of the analyzed 7 cryptocurrencies. May 2017 to February 2018

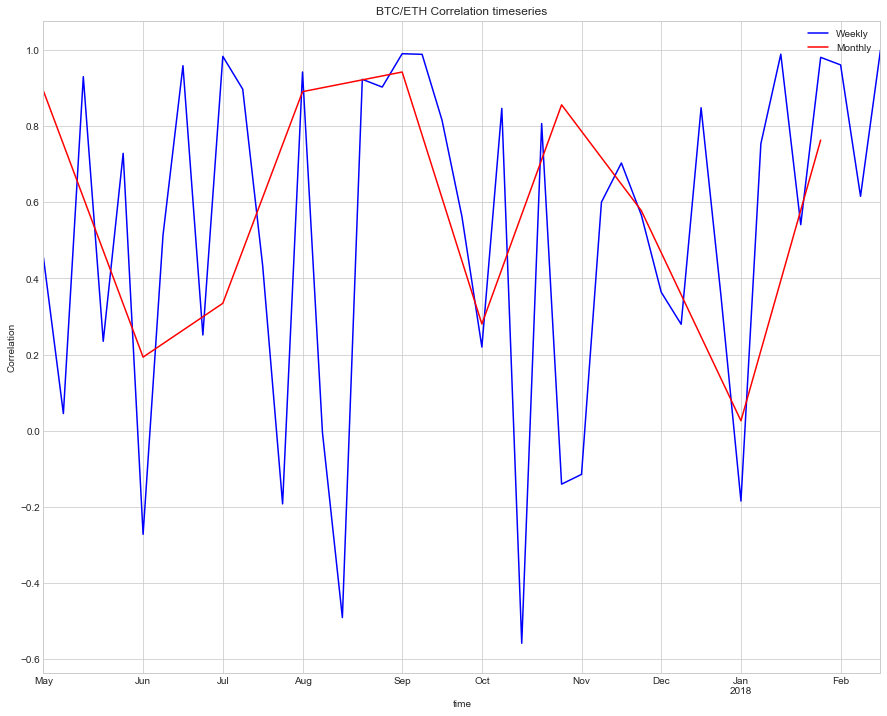
We did not believe the above was a true measure. That period was too high to represent market dynamics. We then computed the heatmap on a shorter timespan.

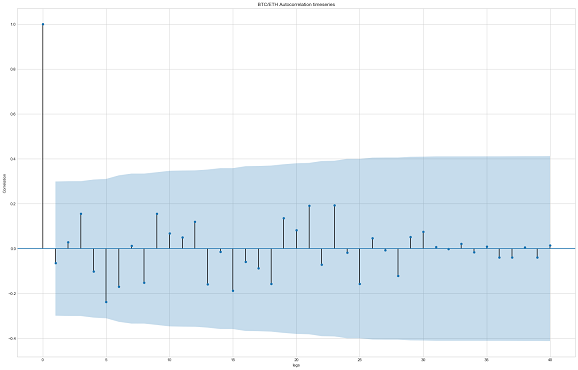
Figure Pearson correlation heatmap of the analyzed 7 cryptocurrencies. December 2017

Now we get a different story altogether. In December 2017. Ether lost its tight correlation with BTC.

We then thought of, what if BTC/ETH correlation varies over time? Can we model it as a timeseries?

We can see the strong seasonality in the figure below. The daily correlation (not shown) is heavily oscillating. Since the seasonality moves to the monthly correlation, this ruled out what we thought that it was only due to some lag effect. (BTC moves, some time lag takes place, then ETH moves at the same direction). In fact, there were two instances where there was almost zero correlation between BTC and ETH. Those were in August (a hard fork time for BTC (Rizzo, 2017)) and in January (BTC bust)

Figure BTC/ETH weekly and monthly correlation. Since May 2017

We then plotted the daily and weekly autocorrelation[[5]](#footnote-5). Time dependence is shown on the daily autocorrelation and then disappears on the weekly. Time dependence is confirmed if one of the below vertical lines (showing correlation between the data and its lagged version) falls above/below the blue confidence intervals.

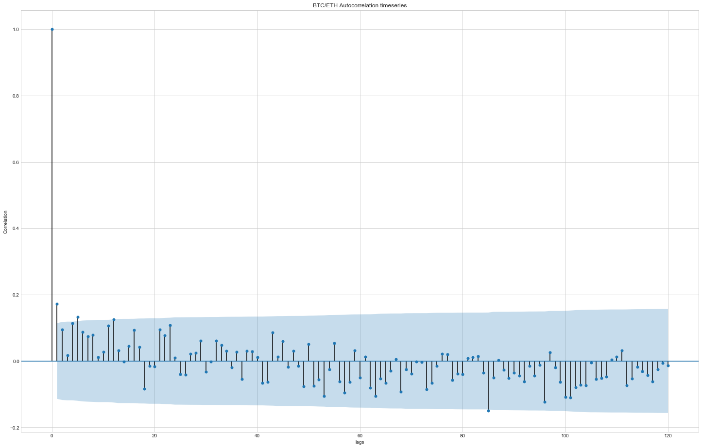


Figure Daily autocorrelation. BTC/ETH Figure Weekly autocorrelation. BTC/ETH

# Appendix A: Crawler Code Overview

Our code is hosted on the following github repository:

<https://github.com/mbaddar/Ethereum_Blockchain_Parser_Parallel>

Below is a list of the most relevant files:

## Scripts/parseBlockchain.py:

This is the entry point to call the Ethereum blockchain crawler. The crawling code is based on (Miller, 2017) repository (<https://github.com/alex-miller-0/Ethereum_Blockchain_Parser>)

## Preprocessing/Crawler/:

This is the crawler class called by the parser above. It sends JSON-RPC requests to a geth node and inserts the response into a mongo database. It first checks the highest block available on the database then iterates from the next block to the maximum block specified

We set min\_block\_geth = 2900000 that was mined on 29/12/2016 10:38 PM

The below is the crawler loop:

1. # Get all new blocks
2. **if** self.max\_block\_mongo > self.min\_block\_geth **and** self.max\_block\_mongo < self.max\_block\_geth:
3. #Crawl from the next block
4. **print**("Resuming crawling from block %8d" % ( self.max\_block\_mongo+1  ) )
5. self.min\_block\_geth = self.max\_block\_mongo+1
6. **try**:
7. self.min\_block\_geth = self.max\_block\_mongo
8. **except** Exception:
9. **print**("Error …")
10. **print**("Processing the specified range of the blockchain from %8d to %8d..." % (self.min\_block\_geth, self.max\_block\_geth))
11. **for** n **in** tqdm.tqdm(range(self.min\_block\_geth, self.max\_block\_geth)):
12. self.add\_block(hex(n))
13. **print**("Done!\n")

## Analysis Notebook/Analysis\_of\_Cryptocurrency\_Investments.ipynb:

This is the Jupyter notebook that contains the correlation and time series analytics we have done. It is based on (Ma, 2018) repository <https://github.com/jieyima/Cryptocurrency_Investment_Analysis_and_Modeling>

We have extended the correlation concept and attempted ARIMA analysis on bitcoin data.

Our contribution starts from plotting a correlation heatmap of a specific time period, then creating a correlation time series between bitcoin and Ethereum, then finally attempting to extend the regression models provided and apply ARIMA analysis on Bitcoin data.

# Appendix B: Data extraction and visualization

Code is hosted on the following github repository:

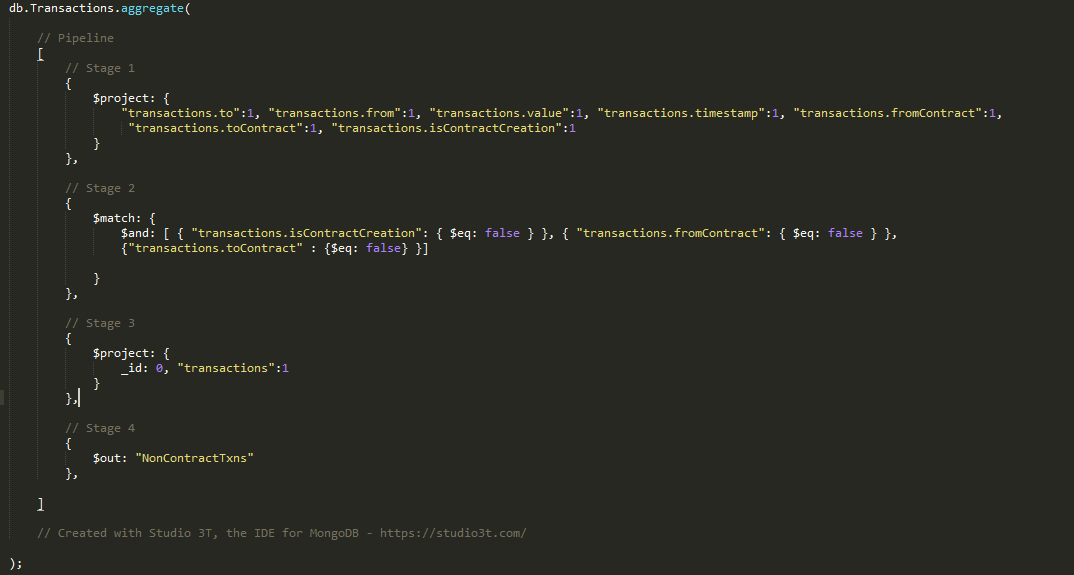
<https://github.com/mbaddar/Ethereum_Blockchain_Parser_Parallel>

Below is a list of the most relevant files:

## Data Extraction and Exploration/Analysis

This folder contains the MongoDB aggregation queries, each numbered by a step number to be run in that order one by one. These queries result in creating additional collections that are either consumed by subsequent queries or the visualization code directly.

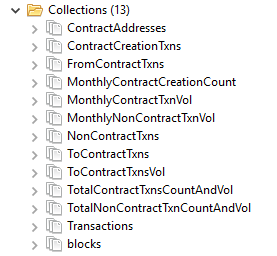
As an example, one of the queries is explained below:



The aggregation pipeline works in stages where the output of last stage is the output of the next.

In the code above, first stage extracts only the relevant data fields so we don’t have to pass unnecessary data in to later stages that would effect memory use. The next stage extract the documents that match our given criteria. The third stage extracts only the transaction data to be saved in a collection by the name of “NonContractTxns” in the fourth stage.

After running all the queries, the list of collections looks like this:



The original data had only ‘blocks’ collection.

The visualization code is written in python. The ‘MainScript.py’ connects to MongoDB and uses

‘VisualizationUtility.py’ to plot different graphs and save them in ‘.png’ format in the ‘Results’ folder.

For further information about the code, refer to the actual python code file.

# Appendix C: Analytics Jupyter Notebook Code Overview

## Loading the dataset

data = pd.read\_csv('crypto-markets.csv', parse\_dates = ['date'], index\_col = 'date')

The dataframe indexes the data by date. Here is a sample:

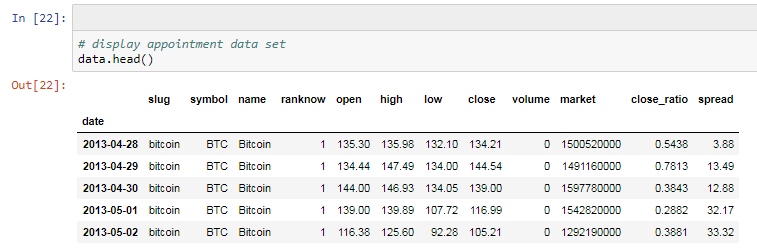


Figure A sample of the crypto daily prices dataset

## Correlation heatmap of a specific time period

1. **import** datetime as dt
3. plt.figure(figsize=(12,6))
4. sns.heatmap(close['30-11-2017':'01-01-2018'].corr(),vmin=0, vmax=1, cmap='coolwarm', annot=True)
5. plt.title('Correlation Heatmap between Bitcoin and other 7 Cryptos')
6. plt.show()

## This function models periodic correlation:

1. **def** calculate\_corr(data, date,timedelta, column\_name ):
2. #Weekly correlation
3. corr = []
4. date\_index = []
5. **while** date <  dt.datetime(2018,2,21,0,0,0):
6. date\_index.append(date)
7. correlation = close[date :date+timedelta][['BTC','ETH']].corr()
8. date = date+timedelta
9. corr.append(correlation['BTC']['ETH'])
10. **return** pd.DataFrame(corr, index=date\_index, columns = [column\_name])

It returns a dataframe with a periodic correlation from a start date with timedelta increments.

## Python Modules Used

On top of pandas, matplotlib, and scikit-learn. We used the following Python modules:

* **Pymongo:** To interface with mongodb
* **Requests**: an HTTP library to send get requests to the Geth node
* **Tqdm**: make loops show a smart progress meter
* **Statsmodels:** provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests.

# Appendix D: Output overview

## An example transaction extracted from the blockchain in json format:

1. "transactions": [{
2. "fromContract": false,
3. "gasPrice": NumberLong(20000000000),
4. "gas": NumberInt(134470),
5. "txNumber": NumberInt(1),
6. "to": "0xe94b04a0fed112f3664e45adb2b8915693dd5ff3",
7. "from": "0x2a73584d7a36f345524b5c74d9ee2a8f8d9ef158",
8. "timestamp": NumberInt(1483051090),
9. "toContract": false,
10. "isContractCreation": false,
11. "inputData": "0x0f2c9329000000000000000000000000fbb1b73c4f0bda4f67dca266ce6ef42f520fbb98000000000000000000000000e592b0d8baa2cb677034389b76a71b0d1823e0d1",
12. "number": NumberInt(2900000),
13. "value": 0.11236934
14. },
15. … More transactions …

## Running a full Ethereum node taking a few days to synchronize:

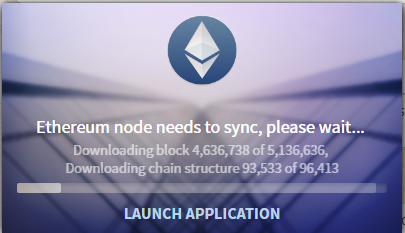


Figure Synching a full Ethereum node. A new block gets generated every 15 seconds

## Our crawling code running on a Virtual Box Linux machine:

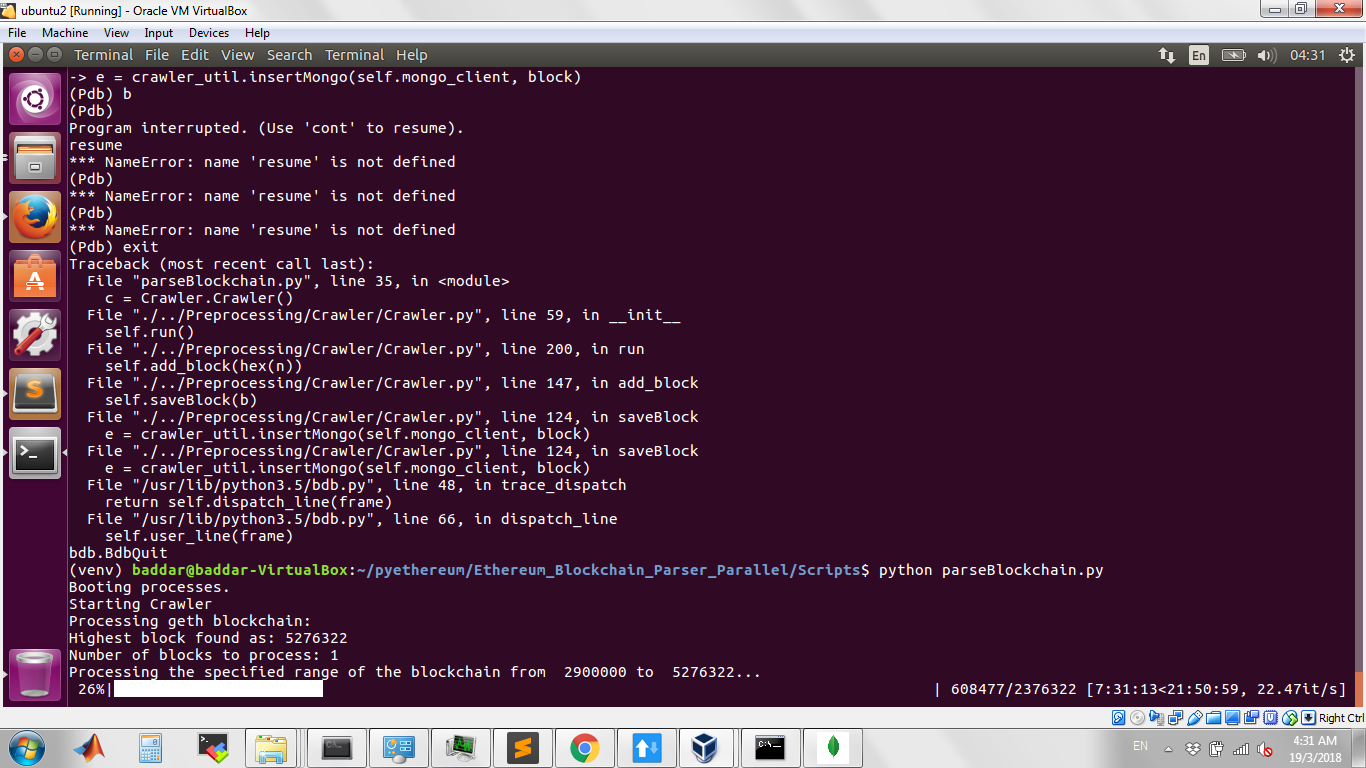


Figure Crawling the blockchain. Early blocks had a bigger throughput (22.7 block/sec) as they contained smaller number of transactions. It then went down to about 6 blocks/sec

## A typical Ether transaction flow:

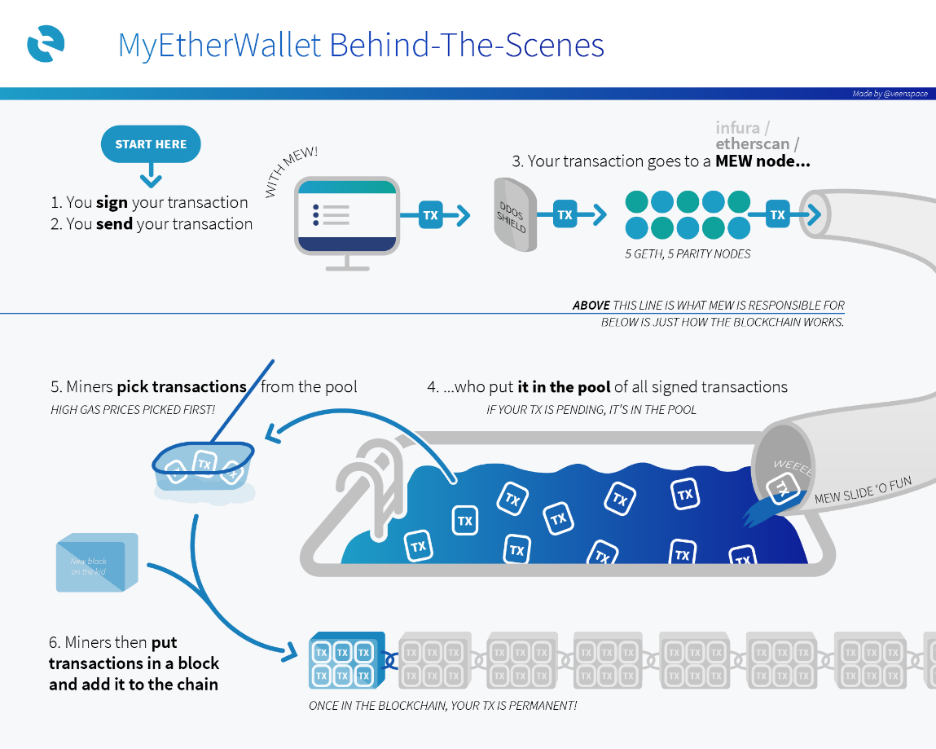


Figure A typical Ethereum transaction flow. MyEtherWallet

# Conclusion:

Analysis: Price volatility, Centrality of wealth

Each block permanently stores a list of confirmed transactions and it is linked to the next block. We have started our analysis from block 2,900,000 that was mined on the 29th of December 2016. The more recent in time the higher the no. of transactions stored in each block which makes the average crawling time to increase. We started by 22 iterations/second and as we moved along it went down to 6 iterations/second

External code

Libraries used: pyethereum, pymongo, tqdm, …

**Correlation and Autocorrelation**

Statistical correlation summarizes the strength of the relationship between two variables.

We can assume the distribution of each variable fits a Gaussian (bell curve) distribution. If this is the case, we can use the Pearson’s correlation coefficient to summarize the correlation between the variables.

The Pearson’s correlation coefficient is a number between -1 and 1 that describes a negative or positive correlation respectively. A value of zero indicates no correlation.

We can calculate the correlation for time series observations with observations with previous time steps, called lags. Because the correlation of the time series observations is calculated with values of the same series at previous times, this is called a serial correlation, or an autocorrelation.

A plot of the autocorrelation of a time series by lag is called the AutoCorrelation Function, or the acronym ACF. This plot is sometimes called a correlogram or an autocorrelation plot.

Partial Autocorrelation:

**ARIMA:**

A popular and widely used statistical method for time series forecasting is the ARIMA model.

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a class of model that captures a suite of different standard temporal structures in time series data.

**Future Work:**

Hourly data

Further expand time series

Bibliography:

1. Github. Maggie Ma. Cryptocurrency Investment Analysis and Modeling. <https://github.com/jieyima/Cryptocurrency_Investment_Analysis_and_Modeling>
2. Ethereum Frontier Guide. <https://ethereum.gitbooks.io/frontier-guide/content/>
3. Etherscan. Ethereum Contracts with verified source code. <https://etherscan.io/contractsVerified>
4. Jason Brownlee. Gentle Introduction to Autocorrelation: <https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>
5. Jason Brownlee. How to Create an ARIMA Model for Time Series Forecasting with Python. <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>
6. Wikipedia. Autoregressive integrated moving average. <https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average>
7. ARIMA Models. <https://ec.europa.eu/eurostat/sa-elearning/arima-models>
8. <https://machinelearningmastery.com/time-series-data-stationary-python/>
9. <https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test>

# Bibliography

AminCad. (2018, 1 12). *Market share of Ethereum-based tokens grows to 91%*. Retrieved from Medium: https://medium.com/@amincad/market-share-of-ethereum-based-tokens-grows-to-91-fdefadfd9f6e

Ethereum.org. (2018). *Go Ethereum. Official Go implementation of the Ethereum protocol*. Retrieved from geth.ethereum.org: https://geth.ethereum.org/

Ethereum.org. (2018). *JSON-RPC*. Retrieved from github.com: https://github.com/ethereum/wiki/wiki/JSON-RPC

Ghemawat, S., & Dean, J. (n.d.). *ethereum/leveldb*. Retrieved from Github: https://github.com/ethereum/leveldb

Griffith, K. (2014, 4 16). *A Quick History of Cryptocurrencies BBTC — Before Bitcoin*. Retrieved from Bitcoin Magazine: https://bitcoinmagazine.com/articles/quick-history-cryptocurrencies-bbtc-bitcoin-1397682630/

Investopedia. (n.d.). *Initial Coin Offering (ICO)*. Retrieved from Investopedia: https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp

Investopedia. (n.d.). *Smart Contracts*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/s/smart-contracts.asp

Liu, B. Y. (2018). *Ethereum Development Tutorial*. Retrieved from github.com: https://github.com/ethereum/wiki/wiki/Ethereum-Development-Tutorial

Madeira, A. (2018, 4 5). *The DAO, The Hack, The Soft Fork and The Hard Fork*. Retrieved from Cryptocompare.com: https://www.cryptocompare.com/coins/guides/the-dao-the-hack-the-soft-fork-and-the-hard-fork/

MyEtherWallet. (2018). *What is Gas?* Retrieved from myetherwallet: https://myetherwallet.github.io/knowledge-base/gas/what-is-gas-ethereum.html

Nakamoto, S. (2008, 10). Retrieved from Bitcoin.org: https://bitcoin.org/bitcoin.pdf

Ray, J. (2018, 4). *The Ethereum White Paper*. Retrieved from The Ethereum Wiki: https://github.com/ethereum/wiki/wiki/White-Paper#ethereum

Siebel, J. (2017, 7 21). *Ethereum ERC20 Tokens Explained*. Retrieved from Medium: https://medium.com/@james\_3093/ethereum-erc20-tokens-explained-9f7f304055df

Sifr Data. (2018, 4). *Cryptocurrency Correlation Matrix*. Retrieved from Sifr Data: https://www.sifrdata.com/cryptocurrency-correlation-matrix/

Szabo, N. (2008, 12 29). *Bit Gold*. Retrieved from Nakamoto Institute: http://nakamotoinstitute.org/bit-gold/

The Ethereum Foundation. (2018). *Solidity*. Retrieved from github.com: https://github.com/ethereum/solidity/blob/v0.4.21/docs/index.rst

*Throwback Thursday: PayPal’s Biggest Days In History*. (2015, 7 2). Retrieved from pymnts.com: https://www.pymnts.com/in-depth/2015/throwback-thursday-paypals-biggest-days-in-history/

Wikipedia. (n.d.). *Blockchain*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Blockchain

Wikipedia. (n.d.). *Cryptocurrency*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Cryptocurrency

Wikipedia. (n.d.). *David Chaum*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/David\_Chaum

Wikipedia. (n.d.). *DigiCash*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/DigiCash

Wikipedia. (n.d.). *Double Spending*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Double-spending

Wikipedia. (n.d.). *Ethereum*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Ethereum

Wikipedia. (n.d.). *Nick Szabo*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Nick\_Szabo

Wikipedia. (n.d.). *The DAO (organization)*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/The\_DAO\_(organization)

Williams-Grut, O. (2018, 1 31). *Only 48% of ICOs were successful last year — but startups still managed to raise $5.6 billion*. Retrieved from Business Insider: http://uk.businessinsider.com/how-much-raised-icos-2017-tokendata-2017-2018-1

1. <https://github.com/ethereum/wiki/wiki> [↑](#footnote-ref-1)
2. <https://github.com/ethereum/wiki/wiki/White-Paper> [↑](#footnote-ref-2)
3. Github. Pyethereum. <https://github.com/ethereum/pyethereum> [↑](#footnote-ref-3)
4. <https://github.com/alex-miller-0/Ethereum_Blockchain_Parser/pull/3> [↑](#footnote-ref-4)
5. More info: Time Series Basics. Penn State University. <https://onlinecourses.science.psu.edu/stat510/?q=book/export/html/41> [↑](#footnote-ref-5)