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 (Sifr Data, 2018) between Bitcoin, the first and most valued cryptocurrency, Ether and other currencies.

# Problem statement:

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

1. What is the potential of smart contract X hosted in the Ethereum blockchain?
2. Are cryptocurrencies consistently correlated to Bitcoin? What is the point of holding a portfolio of different currencies if that is the case?
3. Can we predict the price of Bitcoin?

# 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 definition and concepts used later in this report. For more information please refer to the Ethereum wiki[[1]](#footnote-1) and the Ethereum white paper[[2]](#footnote-2)

## 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.)

Below is an example contract that creates a new token:

1. pragma solidity ^ 0.4.20;
2. contract MyToken { /\* This creates an array with all balances \*/
3. mapping(address => uint256) public balanceOf; /\* Initializes contract with initial supply tokens to the creator of the contract \*/
4. function MyToken(uint256 initialSupply) public {
5. balanceOf[msg.sender] = initialSupply; // Give the creator all initial tokens
6. } /\* Send coins \*/
7. function transfer(address \_to, uint256 \_value) public {
8. require(balanceOf[msg.sender] >= \_value); // Check if the sender has enough
9. require(balanceOf[\_to] + \_value >= balanceOf[\_to]); // Check for overflows
10. balanceOf[msg.sender] -= \_value; // Subtract from the sender
11. balanceOf[\_to] += \_value; // Add the same to the recipient
12. }
13. }

## 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)

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 …

## 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)

## 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)

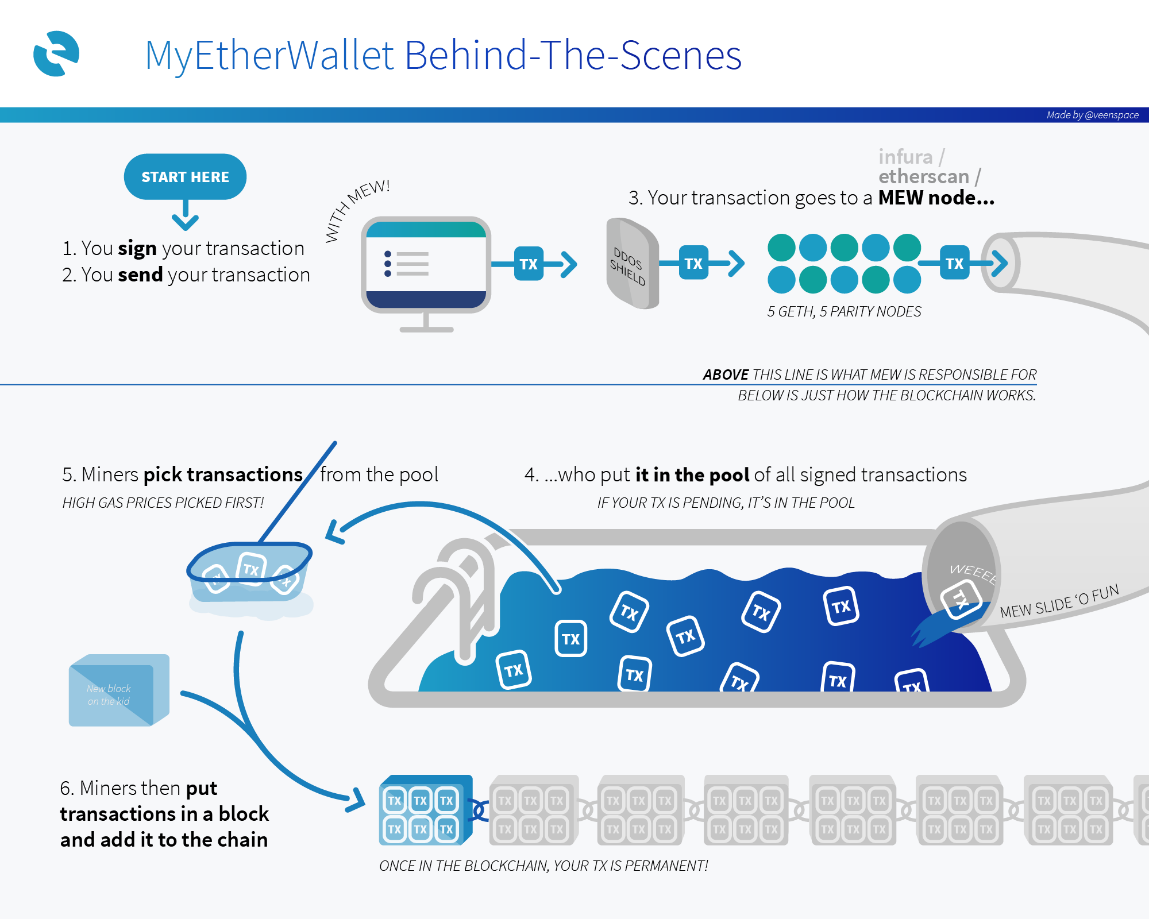


Figure 1 A typical Ethereum transaction flow. MyEtherWallet

## 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[[3]](#footnote-3) 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.

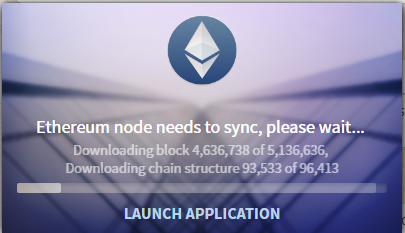


Figure 2 Synching a full Ethereum node

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[[4]](#footnote-4) 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[[5]](#footnote-5) 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!

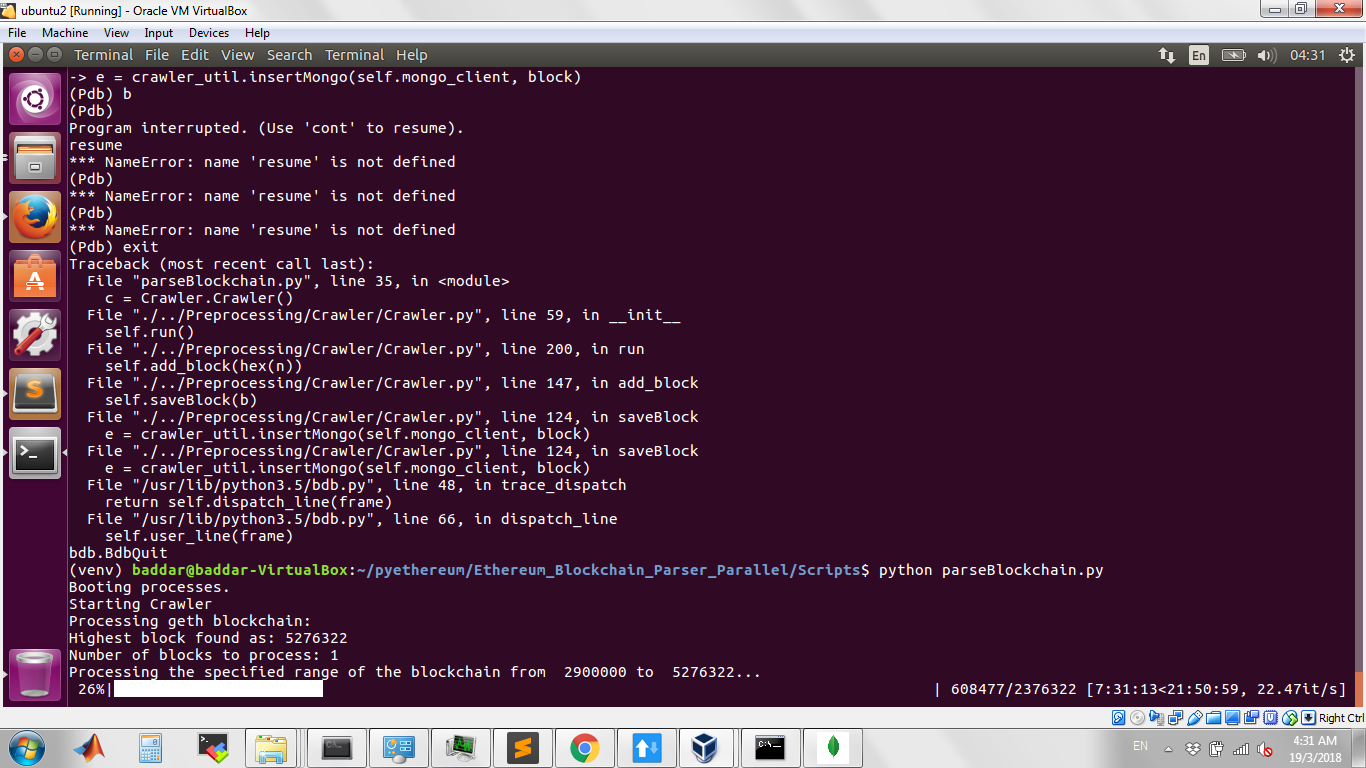


Figure 3 Crawling the blockchain. Early blocks had a bigger throughput (22.7 block/sec) as they contained smaller number of transactions. It then went down to about 6 blocks/sec

Data Wrangling accomplished!

Analysis: Price volatility, Centrality of wealth, smart contracts, Tokens, mining centrality

Data used: The blockchain has more than 5 million blocks with a new block added every 15 seconds. 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, …

Visualization: Piecharts, timeseries, histogram

References:

**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. Wikipedia.
3. Ethereum yellow paper. <https://github.com/ethereum/yellowpaper>. Latest pdf: <https://ethereum.github.io/yellowpaper/paper.pdf>
4. Ethereum Frontier Guide. <https://ethereum.gitbooks.io/frontier-guide/content/>
5. Ethereum JSON-RPC API. <https://github.com/ethereum/wiki/wiki/JSON-RPC>
6. Etherscan. Ethereum Contracts with verified source code. <https://etherscan.io/contractsVerified>
7. Jason Brownlee. Gentle Introduction to Autocorrelation: <https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>
8. Jason Brownlee. How to Create an ARIMA Model for Time Series Forecasting with Python. <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>
9. Wikipedia. Autoregressive integrated moving average. <https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average>
10. ARIMA Models. <https://ec.europa.eu/eurostat/sa-elearning/arima-models>
11. <https://machinelearningmastery.com/time-series-data-stationary-python/>
12. <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

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

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/

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

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/

*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. Alex Miller Ethereum Blockchain Parser: <https://github.com/alex-miller-0/Ethereum_Blockchain_Parser> [↑](#footnote-ref-3)
4. Github. Pyethereum. <https://github.com/ethereum/pyethereum> [↑](#footnote-ref-4)
5. <https://github.com/alex-miller-0/Ethereum_Blockchain_Parser/pull/3> [↑](#footnote-ref-5)