UNIVERSITÀ DEGLI STUDI DI TRENTO

Department of Information Engineering And Computer Science

Bachelor degree in
Computer Science

Final Dissertation

Analysis of *Blockchains* and *Bitcoin* with a focus on the integration of *Bitcoin Cash* in R

Supervisor

Agostinelli Claudio

Candidate/Student

Vasile Adrian Bogdan Pop

Accademic Year 2018/2019

I would like to thank my family and all those who supported me during this step of my life

Contents

Abstract

The thesis attempts to explain what a blockchain is, how it works and why it seems to have so much potential. In particular, this debate focuses on the different purposes of this technology with its functionalities and its implementations by analysing two specific blockchains: Bitcoin (BTC)[**?**] and Bitcoin Cash (BCH)[**?**]. The thesis also exposes which are the requirements a blockchain should have when it is used as a coin, in order to be globally handled as the leading currency. Hence, it goes deeper into the differences between the BTC and BCH analysing the distinct approaches to satisfy the demands of a cryptocurrency. Especially, it illustrates the main reasons why the Bitcoin hard fork was reached on August 1st, 2017, giving life to the Bitcoin Cash blockchain. Once the behaviour of this new technology is clear, the thesis goes further with the integration of BCH in the programming language *R* (see section **??** for more details). In this way, it provides a means to work with the data stored into the Bitcoin Cash blockchain and retrieve interesting statistics regarding it.

In other words, the steps to achieve the goal for this thesis will be:

- Understanding and analysing blockchain technologies.
- Studying the two most important Bitcoin blockchains: BTC - Bitcoin and BCH - Bitcoin Cash.
- Designing a new Package in R for data reading and manipulation of the Bitcoin Cash blockchain.
- Implementation of the Package in R with a description of all the possible commands developed.
- Example of the Package functionalities and its potential.

1 Blockchain

The application of the blockchain in multiple fields is gaining more and more traction today to the point of which some are even considering it the new Internet[?]. But actually, the main concept behind this revolutionary idea was described as early as 1991 in an academic paper[?] published by Stuart Haver and W. Scott Stornetta. With their solution, the two scientists aimed to Time-Stamp documents, in order to certify when a document was created or last changed. The system used a cryptographically secured chain of blocks to store the time-stamped documents and in 1992 Merkle trees[?] were incorporated into the design, making it more efficient by allowing several documents to be collected into one block.[?] However, this was not enough. Indeed, in order to have a fully working blockchain more components are needed.

1.1 Distributed ledger

As said in a report from the UK Government, "A distributed ledger is essentially an asset database that can be shared across a network of multiple sites, geographies or institutions. All participants within a network can have their own identical copy of the ledger"[?]. Due to this technology, there are neither central administrators nor centralized data storages. To have this mechanism working, both a Peer-to-Peer Network (the peers are computer systems connected to each other via the Internet, the data can be shared directly between systems on the network) and a Consensus Algorithm are needed(See section ?? for more details).

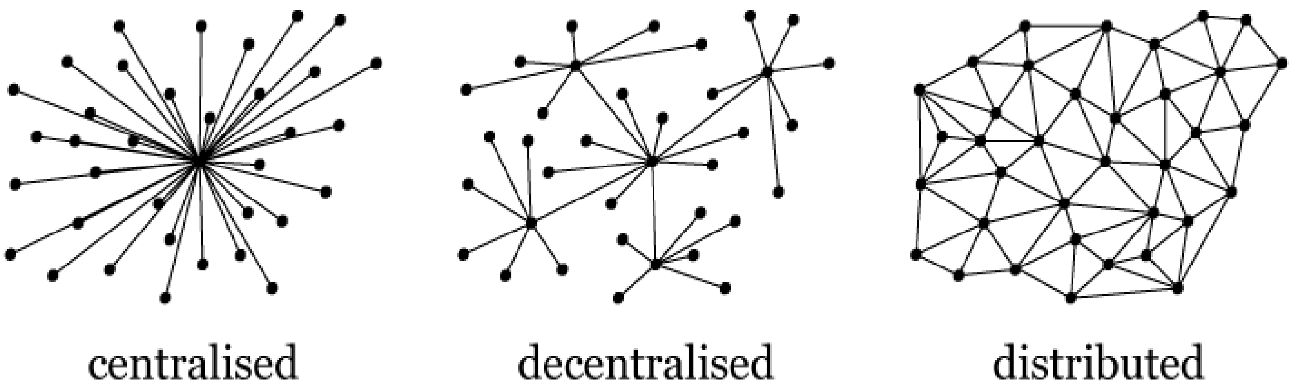

Figure 1.1: Representation of the different types of network architecture.[?]

1.2 Consensus Algorithms

To clarify the benefits of this type of algorithm we must start with a dilemma known as "The Byzantine Generals Problem"[?], in which, each general has its own army and each group is situated in different locations around the city they intend to attack. The generals need to agree on either attacking or retreating. It does not matter whether they attack or retreat, as long as all generals reach consensus, i.e., agree on a common decision in order to execute it in coordination. Therefore, the following requirements are to be considered:

- Each general has to decide: attack or retreat (yes or no).
- After the decision is made, it cannot be changed.
- All generals have to agree on the same decision and execute it in a synchronized manner.

The aforementioned communication problems are related to the fact that one general is only able to communicate with another through messages, which are forwarded by a courier. Consequently, the central challenge of the Byzantine Generals' Problem is that the messages can get somehow delayed, destroyed or lost.

Further, even if a message is successfully delivered, one or more generals may choose (for whatever reason) to act maliciously and send a fraudulent message to confuse the other generals, leading to a total failure.

If we apply the dilemma to the context of blockchains, each general represents a network node, and the nodes need to reach consensus on the current state of the system. To clarify the concept, the majority of participants within a distributed network have to agree and execute the same action to avoid complete failure.[?][?] There are various implementations of the mechanism through which a blockchain network may reach consensus.

1.2.1 Proof of Work

Due to Bitcoin (see ?? for more details), the most known consensus algorithm is the Proof of Work (PoW), in which is employed for block generation. The process of generating correct proofs to add a block to the blockchain is known as "mining" and the individuals that participate in the mining process are known as "miners".[?]

PoW mining involves numerous hashing attempts, so more computational power means more trials per second. In other words, miners with a high hash rate have better chances to find a valid solution for the next block. The PoW consensus algorithm makes sure that miners are only able to validate a new block and add it to the blockchain as long as the distributed nodes of the network reach consensus and agree that the block hash provided by the miner is a valid proof of work.[?]

1.2.2 Proof of Stake

The Proof of Stake (PoS) consensus algorithm is relatively different and a new way to generate and append blocks in a blockchain. It was developed in 2011 as an alternative to PoW, which requires a massive amount of energy to make it work. In fact, according to the bitcoin energy consumption tracker at Digiconomist[?], bitcoin currently consumes 66.7 terawatt-hours per year. That is comparable to the total energy consumption of the Czech Republic, a country of 10.6 million people.

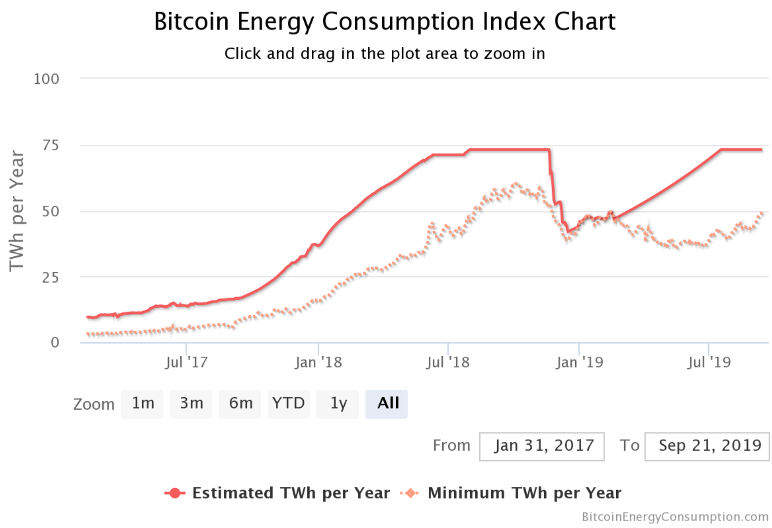

Figure 1.2: Bitcoin energy consumption in the mining process.[?]

Basically, the mining process is replaced with a mechanism where blocks are validated according to the stake of the participants, also known as "validators". The criteria used to choose the validator depends on the proof of stake system but mainly, the choice is based on the economic stake in the network of each node.[?][?]

1.2.3 Proof of Burn

The Proof of Burn (PoB) algorithm is an approach to avoid the massive waste of energy used for hashing. Indeed, with this technique, the mining process is replaced with a greener one. Where, the "miners" of the PoB coins will send coins to an unspendable address, known as "eater address". Even these transactions are recorded on the blockchain ensuring that the coin cannot be spent again. The principle behind this consensus algorithm is that the user burning the cryptocurrency is showing long-term commitment to the coin by burning it. This is because they are taking a short-term loss in exchange for a long-term gain.[?]

1.3 Advantages and Disadvantages

As a result of its complexity, blockchain's potential as a distributed form of record-keeping is almost without limit. But like every other technology, even the blockchain has its pros and cons.

1.3.1 Advantages

Trustless system and small fees. In most traditional systems, a consumer pays a bank to verify a transaction, a notary to sign a document, or a minister to perform a marriage. Using blockchain technologies, this is no longer necessary because the distributed nodes verify the transactions through consensus algorithms.
Therefore, blockchain systems negate the risk of trusting a single person or organization and reduces the overall costs and transaction fees by cutting third-parties. For this reason, blockchain is often referred to as a "trustless" system.[?][?]

Transactions are secure, private and efficient. By eliminating the human factor, we obtain a 24 hour a day working system, instead of institutions operating only during business hours. As a result, transactions can be completed in about ten minutes and can be considered secure after just a few hours. Actually, once a transaction is recorded, its authenticity must be verified by the blockchain network where millions of nodes rush to confirm that the details of the transaction are correct.

Moreover, every block of the chain has the hash of the block before it along with its own hash, which is computed with the data contained in the block. So, when a piece of information in a block is edited in any way, its own hash will change, but the hash in the block after it will not. This contrast makes it extremely difficult to change data inside the blockchain without being noticed.

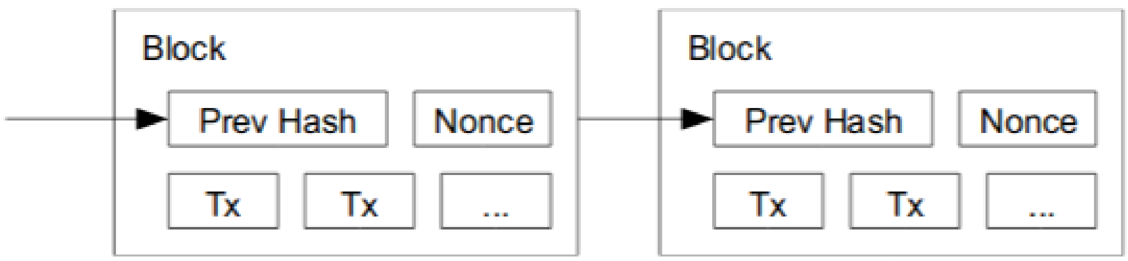

Figure 1.3: A sample of the chain of blocks.[?]

Considering that the majority of blockchains are public, means that anyone can access the blockchain and read the data inside it with all the transaction history. So, if anyone can read all the transactions, why do people say that blockchain transactions are anonymous?
This is because anonymity is confused with confidentiality. Indeed, every user makes public transactions with their unique code called "public key", which is recorded on the blockchain, rather than their personal information.[?] "Public key cryptography" is a cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are known only to the owner.

Transparency. A big benefit to this technology is given by its open-source nature. This means that the users of the blockchain network can modify the code as they consider is the most fitting for their purposes, so long as they have the approval from the majority of the network's nodes.

Distributed. Blockchain does not store any of its information in a unique node or central location. Instead, data is stored in thousands of devices on a distributed network. Whenever a block is added to the chain, every node updates its blockchain to reflect the change. Because of this, there is no single point of failure and so it becomes more difficult to tamper with.[?][?]

Stability. Once a block is confirmed, it is extremely difficult to remove or change the data inside it. This makes blockchain a great technology for storing financial records or any other data where an audit trail is required because every change is tracked and permanently recorded on a distributed and public ledger.[?]

1.3.2 Disadvantages

While there are significant benefits to the blockchain, there are also serious challenges to its adoption. The most signigivatives ones are political and regulatory[?], but there are also some technical challenges to take into account.

Private keys. Blockchain uses public key cryptography to give users ownership over their blockchain data. Each public key corresponds to a private key which should be kept secret. Users need their private key to access their data. If a user loses its private key, the money is effectively lost, and there is nothing they can do about it.[?]

`key_generation.png`

Figure 1.4: Private key, public key, and address generation flow.[?]

Energy inefficiency. Some blockchains, especially those using Proof of Work consensus algorithms **??**, are highly inefficient. Since mining is highly competitive and there is only one winner every ten minutes, the work of every other miner is wasted. As miners are continously trying to increase their computational power to have a greater chance of finding a valid block hash, the resources used by the Bitcoin network increased significantly in the last few years, and it currently consumes more energy than that of many countries, such as Denmark, Ireland, and Nigeria.[?]

Scalability. The Bitcoin is a perfect case study for the low scalability of blockchains. In fact, to add a new block on the blockchain, it takes about ten minutes and the maximum size of each block is of 1 MegaBytes. This entails that there is a limited number of transactions per block. It is estimated that Bitcoin's blockchain can manage only seven transactions per second (TPS). Meanwhile, the Visa's actual infrastructure can process 24.000 TPS.[?]

51% Attacks. The consensus algorithms that protect the blockchain have proven to be very efficient over the years (thanks to Bitcoin). However, there are a few potential attacks that can be performed against blockchain networks. The most discussed is the 51% attack. Such an attack may happen if one entity manages to control more than 50% of the network, which would allow them to edit or modify the order of the transactions. Although it is theoretically possible, there has never been a successful 51% attack on the Bitcoin blockchain, additionally, as the network gets larger, the security increases.[?]

Storage. Distributed ledgers can increase significantly over time. For example, the Bitcoin blockchain currently requires 239 GigaByte of storage (Last update: 10/9/2019). Even if in the Bitcoin paper it is said that thanks to "Moore's Law"[?] the storage should not be a problem, the current growth in blockchain size appears to be outstripping the growth in hard drives and at the same time the network risks losing nodes if the ledger becomes too large for individuals to download and store.[?]

1.4 Use Cases

The merit to the great attention given to the blockchain technology is to be attributed to Bitcoin. Indeed, thanks to these digital currencies (cryptocurrencies) the blockchain-based solutions have grown. So learning how this innovative technology can be applied to different scenarios is very important.

1.4.1 Digital Identity

The identity component in a blockchain is fulfilled through the use of cryptographic keys. Combining a public and private key creates a strong digital identity reference based on possession of pieces of information or certifications. The public key is distributed publicly and it is what identifies the identity, whereas the private key is kept secret by the identity and it is used to express consent to digital interactions and to sign the data to record in the block.[?]

To make an example, it could be useful when one has to prove some information about themselves. Suppose someone has a digital copy of the driver's license and then that person must prove to be old enough to drink. The bartender could verify the proof using the public key of the issuer. But the bartender never learns his actual birth date, that is what it is called "Zero Knowledge Proof".[?]

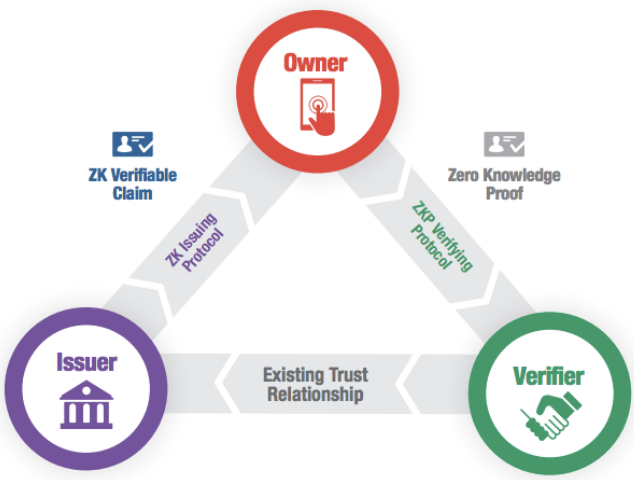

Figure 1.5: Zero Knowledge Verifiable Claims.[?]

1.4.2 Charity

The number of charitable organizations which are adopting cryptocurrencies as a donation method is increasing more and more. The reason is due to crypto's transparency, in fact, thanks to this characteristic, every donation can be tracked and verified. So, the higher is the level of transparency and public accountability, the easier will be for donors to be encouraged to make an offer and as a consequence the charity's reputation for integrity will be reinforced too. Moreover, blockchain technology reduces fees and taxes to manage every donation by reducing the number of required intermediaries.[?]

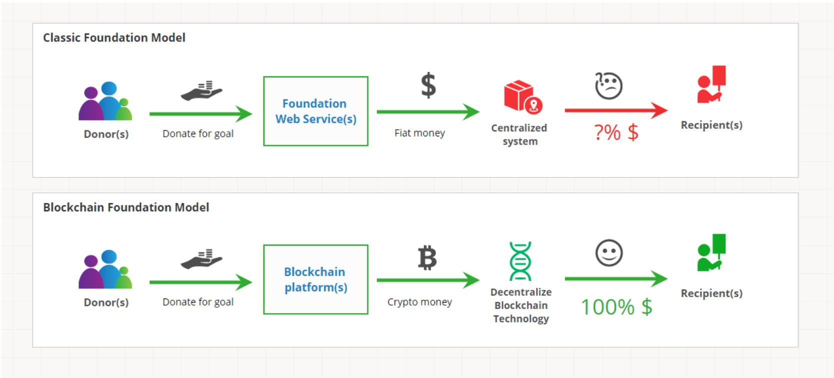

Figure 1.6: Charity donation schema.[?]

1.4.3 Healthcare

Blockchain may offer significant benefits to hospitals in terms of security, interoperability, and transparency. Unlike traditional databases that rely on a centralized server, the use of a distributed system allows for data exchange with higher levels of security. In addition, distributed ledgers increase the interoperability among clinics, hospitals, and other health service providers by allowing all of these parties to interface with a unique storage system.

Furthermore, blockchain systems may also give higher levels of accessibility and transparency over their own health information. But also, allow pharmaceutical organizations to increase the efficiency of their infrastructure by cutting down on the widespread problem of drug counterfeiting.[?]

1.4.4 Governance

Governance can be greatly improved in various sectors by blockchain technologies. A short example can be the efficiency of taxes, given that governments rely on taxpayer funds, they must use their budgets wisely. Blockchain systems and smart contracts can be employed to automate tasks and workflows, which would greatly reduce time and money spent on bureaucratic processes.[?]

Another application can be the election process. With the support of a working digital identity system and the high level of immutability given by the blockchain, this technology could be an excellent solution for ensuring that votes cannot be tampered with. Creating the possibility to transform the secure online voting into a reality.

1.4.5 Payments

Thanks to Bitcoin, this is the most widespread application of the blockchain. It started in 2008 when Satoshi Nakamoto published the Bitcoin's white paper, where, for the first time, all the above components of a blockchain were used to solve the double-spending problem.

2 Bitcoin

In a few words, Bitcoin is a digital form of actual cash. But in unlike the traditional fiat currency, which relies on a central bank controlling it, this new form of currency relies on blockchain technology and cryptography. In computer science, "cryptography" refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms to transform messages in ways that are hard to decipher. As a result, this new digital cash is mainly referred to as "cryptocurrency".

2.1 Infrastructure

Bitcoin, as the majority of cryptos, is based on a public permissionless blockchain architecture. Which means that anyone can join, read, write and commit, plus it is hosted on public servers, is anonymous and is highly resilient.
The Bitcoin's mechanism to reach consensus is the Proof of Work algorithm **??**. Due to the implementation described by Satoshi Nakamoto in the white paper "The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash. For our timestamp network, we implement the proof-of-work by incrementing a *nonce* (a 32-bit value of the block) in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it"[**?**]. It follows that the difficulty required to close a block is flexible. In fact, it changes every 2016 blocks suiting the computational power of the network. Considering that, at the desired rate of one block every 10 minutes, 2016 blocks mean that the difficulty is adjusted every two weeks.

2.1.1 Incentive

To have a working PoW, miners have to spend their CPU time and electricity. So, why should they allocate their resources in Bitcoin?

The answer is the incentive. Indeed, the first transaction of every block is a special transaction which gives a new amount of currency to the creator of the block. This, not only is a good incentive to support the network but also provides a way to initially distribute coins into circulation, since there is no central authority to issue them. However, this process is destinated to end, because there is a

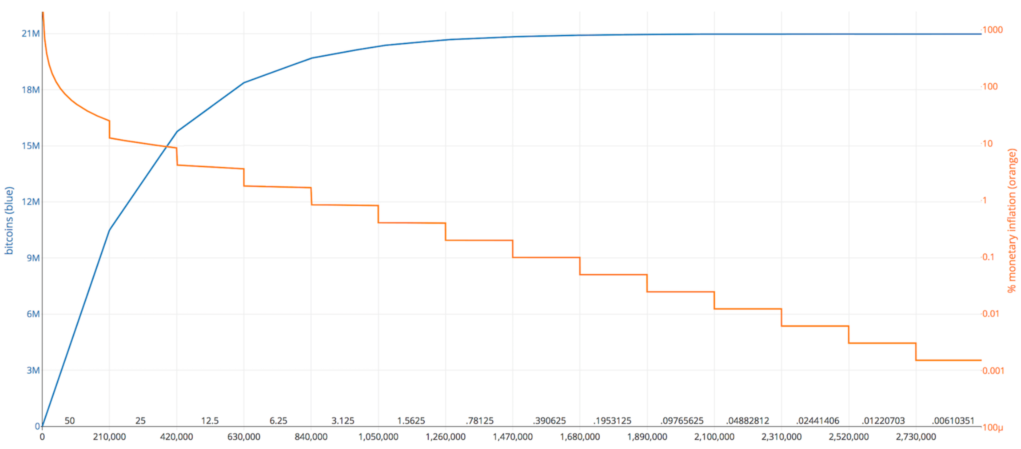

Figure 2.1: How mining rewarding is planned over time.[**?**]

predetermined number of coins (21 million) which can spread. To ensure that, every 210.000 blocks (corresponding to 4 years), the reward given to the creator of the block is halved. This leads to the second type of incentive in the Bitcoin network: transaction fees. In fact, for every transaction, there is also a fee amount, which is given to the creator of the block as a reward for its effort.

2.2 Transactions

To figure out how a transaction works, it is necessary to first understand the concept of "owning a bitcoin". Because, unlike the fiat currency, where you own a physical coin to spend, owning a bitcoin means that you own the key to unlock the coins from a past transaction and use them in the new transaction. It is possible to get a better sense of the transaction mechanism, from a brief extract of the Bitcoin paper:

"We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership"[?]

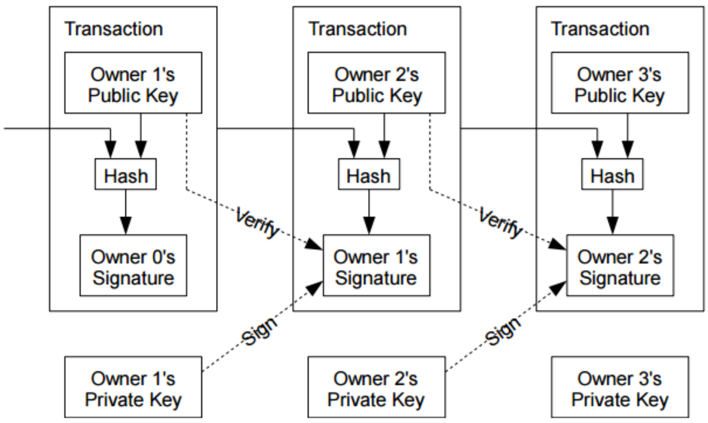

Figure 2.2: Transaction schema. From [?]

To go deeper into the design of a transaction, it is useful to imagine a double-entry ledger. In a few words, every transaction has one or more "inputs" and one or more "outputs" which are respectively debits and credits against a bitcoin address. Usually, outputs add up to slightly less than inputs, because the difference represents an implicit "transaction fee" **??**. The higher the fee, the higher the probability for the transaction to be included in the next block to mine.

Each output, before it is spent as an input in a new transaction, waits as an "Unspent Transaction Output" (UTXO). The sum of all the owned UTXOs, compose the total balance of a wallet **??**.
But, not every transaction in the blockchain produces an UTXO.
It depends on the *scriptPubKey* used in the specific transaction. A scriptPubKey is ascript included in outputs which set the conditions that must be fulfilled for those satoshis to be spent. Data for fulfilling the conditions can be provided in a signature script (another type of data generated by the spender).

Pay To Public Key Hash (P2PKH) Is the most common form of scriptPubKey used to send a transaction to one or multiple addresses. In this type of scriptPubKey, the coins sent can be unlocked by a single private key corresponding to the public key hashed in the transaction itself. It is possible to recognize it by looking at the first character in the address used for the output. In fact, if it starts with the number "1" it is P2PKH.[?]

Pay To Script Hash (P2SH) Was implemented in 2012 to add more flexibility to transactions by adding a hash of a second script, the redeem script. That is similar in function to a scriptPubKey. Where, one copy of it is hashed to create a P2SH address (used in an actual scriptPubKey) and another copy is placed in the spending signature script to enforce its conditions. It can be used to process multisignature transactions, Open Assets Protocol[1] and to store textual data on the blockchain. The P2SH can be recognized by the number "3" at the begin of the addresses.[?]

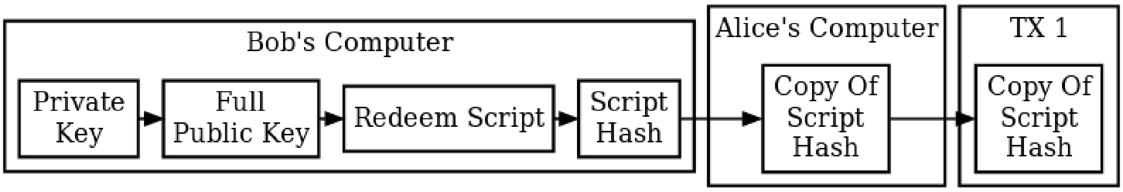

Figure 2.3: creating A P2SH Redeem Script Hash To Receive Payments [?]

Multisig Are special addresses where to spend a transaction, the signature of multiple private keys are required. Although P2SH multisig is now generally used for multisig transactions, this base script can be used to require multiple signatures before a UTXO can be spent. In multisig pubkey scripts, called m-of-n, m is the minimum number of signatures which must match a public key; n is the number of public keys being provided.[?]

Null Data This type adds arbitrary data to a probably unspendable scritptPubKey that nodes don't have to store in their UTXOs database. Consensus rules allow null data outputs up to the maximum allowed scriptPubKey size of 10.000 bytes provided.[?]

2.3 Wallets

In simple words, a wallet is the means through which is possible to manage all the owned cryptocurrency. But it can be split into two components, programs, and files. Wallet programs create the public keys to receive the Bitcoin and use the corresponding private key to spend them.
Wallet files, instead, store the private keys in a file or physically on pieces of paper. Furthermore, it also provides both process used for the public-private key creation and usage, and approach to a deterministic hierarchy key creation process in an unlinkable key manner.

[1] Used to create new currencies upon the Bitcoin blockchain. Called colored coins

3 Bitcoin Cash

"Imagine you have a dollar in your wallet, and you put it aside for a while and then realize that one coin has split in two. Sounds weird? Not in the cryptocurrency world." That is an analogy with the fiat currency reported in an article of South China Morning Post[?]. Bitcoin is like a software, but, due to its distributed nature, unlike all the software we know, there is not a single entity which determines how it should be updated. As a result, in order to upgrade the blockchain it is necessary to be in agreement with the major part of the community. That is what happened on July 20^{th}, 2017, when the 97% of the Bitcoin network voted to activate the "Segregated Witness" (SegWit) algorithm **??** to improve the scalability of Bitcoin. Although almost all the community agreed, some members believed that adopting SegWit without increasing the block size would simply postpone the scalability problem of Bitcoin. As a result, on August 1^{st}, 2017 there were two forks of the Bitcoin network. One for adopting SegWit and one incrementing the size of the blocks giving birth to the Bitcoin Cash blockchain.[?][?]

3.1 Fork

As said before, to have an upgrade in a blockchain we need the agreement of all the network, otherwise those updates generates what is called a "fork". Basically, there are two types of fork:

Soft Fork means that change in the blockchain protocol is backward-compatible. That means, that although some nodes of the network are outdated, they are still able to process a transaction on the network, as long as they do not break the new protocol rules. That is what happened with SegWit in August 2017.

Hard Fork is the opposite. An obsolete node cannot perform any transaction on the new protocol. According to the situation, a hard fork can either be planned or controversial. In a planned fork, participants are going to upgrade their software voluntarily, leaving the old version behind. The non-updated nodes are going to mine on the old blockchain, which will be used by very few participants. In a controversial fork instead, there is a disagreement within the community about the upgrade, so, the old blockchain is split into two new incompatible ones.
Both will have their own network, and as it happened to Bitcoin Cash on August 2017, will be developed in the way the participants believe is the best. This is where the analogy made at the beginning comes to help because, all the coins owned before the fork, will be split. So, the same amount of coins as before will be owned in both the blockchains.[?]

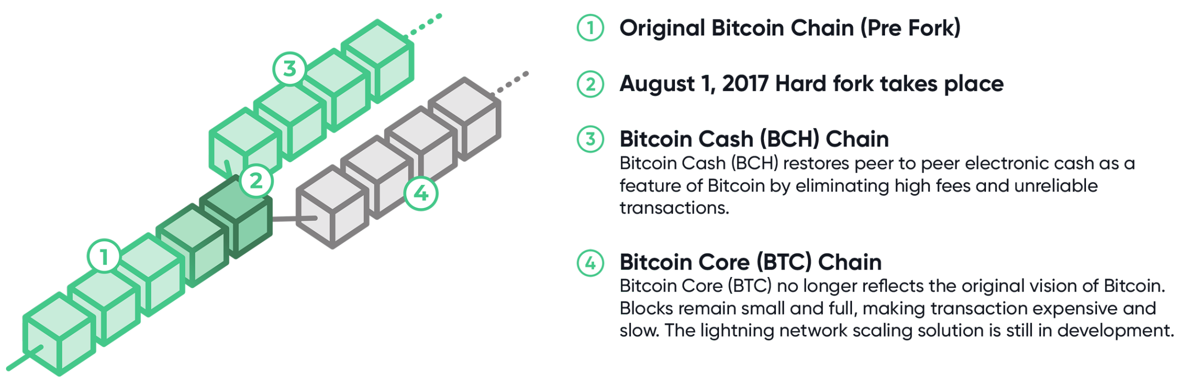

Figure 3.1: A simple schema of the controversial hard fork of Bitcoin Cash. From [?]

3.1.1 SegWit

Segregated Witness is the major discrepancy between Bitcoin and Bitcoin Cash. In fact, unlike BCH, where, to increase the number of transactions per second (TPS) the block size was only incremented to 8 MegaByte, with SegWit the upgrade was more complicated. The transaction is split in two segments, the first, which is saved on the blockchain, contains the information about the sender and the receiver. Meanwhile, the second part containing the scripts and the signature of the sender remains at the bottom in a separate structure. As a result, the amount of information per transaction is less and makes it possible to add more transactions in each block.
This is only one of the multiple benefits from the adoption of SegWit algorithm. To learn all about the benefits of this algorithm please read the official documentation on the BitcoinCore website[?].

3.1.2 Addresses

Due to the nature of Bitcoin Cash, its addresses are generated in the exact same manner as the Bitcoin ones, so it is very difficult to recognize the source of a specific address. To facilitate its use and decrease the probability of being confused when someone tries to read it, the new Cashaddr Address Format was introduced. In a few words, it is only a new type of encoding which displays the address with a different pattern. Besides, every new Cashaddr corresponds to an old address, so when necessary, the outdated addresses can also be used.

3.1.3 Block mining

With the fork on August 1^{st}, 2017, Bitcoin Cash not only changed the block size but implemented also the "Emergency Difficulty Adjustment" (EDA) algorithm. The motivation was to have a method to convince miners to expend their resources in BCH. At the beginning, even the price of Bitcoin Cash was significantly lower than the price of Bitcoin, due to the fork process, the difficulty for mining was the same. So, it was less convenient to mine in the BCH blockchain. EDA resolved this problem by reducing the difficulty for block number t by 20% only when the time difference between the $(t-6)^{th}$ block and $(t-12)^{th}$ block had been greater than 12 hours. This system had been active from the 1^{st} of August 2017, to the 12^{th} of November 2017. Indeed, the next day, a hard fork was made to implement the new "Difficulty Adjustment Algorithm" (DAA), which seeks to accomplish the following objectives:

- Adjust difficulty to hash rate to target a mean block interval of 600 seconds.
- Avoid sudden changes in difficulty when the hash rate is fairly stable.
- Adjust difficulty rapidly when the hash rate changes rapidly.
- Avoid oscillations from feedback between hash rate and difficulty.
- Be resilient to attacks such as timestamp manipulation.

To fulfill all these requirements, the new algorithm adjusts the difficulty with each block, taking into account the amount of work done and the elapsed time of the previous 144 blocks.[?][?]

4 Package implementation in R

Once understood what Bitcoin Cash is, how the technology works behind it and the reasons why it was created. We are going to describe how we can extract information from the blockchain using the R programming environment. We will describe how we can read blocks and all the data inside them, allowing us to make some useful analysis on the Bitcoin Cash behaviour.

4.1 R

R[**?**] is an Open Source software for statistical computing (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering) and graphics. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed.
R is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R which makes it easy for users to follow the algorithmic choices made. But, for computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Furthermore, thanks to its Open Source nature, it can be easily extended via "Packages" even written by the user.[**?**]

4.2 Feasibility study

The goal for this project is to extend the R language with a package which allows us to read and analyse the information contained into the Bitcoin Cash blockchain.
To make it in the best way, firstly, a research for the existing implementations has been done, but unfortunately, without any result. Except for two completely different ways to build the package.
The first one was to turn the used device into a node of the Bitcoin Cash network. This lead the user to download all the blockchain and make all the queries locally. As a result, the queries turned out to be very fast and all the data of the blockchain manipulated as one may desire. But, on the other hand, storing all the blockchain requires a lot of space on the device (around 143 GigaByte on 13/9/2019). So making a package of that size does not seem to be the best option.
For these reasons, the second way was chosen. Although it is the slowest implementation, it avoided the need to become a node of the network, and thus making the installation of the package very simple and fast. That was obtained by relying on a service provider (SP).

4.3 Package integration

For the second option, it was rather difficult to find a SP which not only implemented an "Application Programming Interface" (API)[**?**] where one could make queries and have the wanted data as a response. But also made it public, so anyone can use it. Additionally, the service providers were chosen also according to the amount of data provided, the format of the response and the number of queries one was allowed to make. At the end, the picks were:

- Blockdozer Explorer.
- Blockchair.

The motivation for these two options was the fact that they were complementary. The first one allowed an unlimited number of queries returning essential information. The second one instead, had a limit of 30 queries per minute but returned more and better-formatted data.

Let us begin with the technical implementation by saying that in order to maintain all the R project, was used the RStudio editor[**?**] has been used.

First of all, it is necessary to create and manage a new package in R. For this purpose, some other packages are required :

- *devetools* : to manage the project and the source code of the new package.
- *roxygenize2* : to handle all the documentation of the package.

Then it is necessary to install the other two packages to handle the data and the connection with the service providers.

- *httr* :to perform correctly the Http requests to the service providers and receive the linked response
- *jsonlite* :to manipulate and store the received data, in order to have a good dataset for the future analysis

With these tools is possible to proceed to the performing of a simple query to an API. Suppose one might to know some information about a specific Bitcoin Cash address. After choosing the appropriate SP, their documentation must be readed and seen how to correctly build the Http request. In this case, the pattern is:

GET /api/addr/{paymentAddress}[?noTxList&from&to&returnLegacyAddresses]

```r
bch_address <- "qrn0adtaah922mwz3vems7jqf6hpemdppu2mvm75xe"
base <- "https://blockdozer.com/insight-api/"
endpoint <- "addr/"
txs <- "1"

call <- paste(base,endpoint,bch_address,"/?noTxList=",txs, sep = "")
```

Figure 4.1: Example of the building of an Http request

With a few lines code, we can then construct a possible request for the desired information. As a result one can have:

https://blockdozer.com/insight-api/addr/qrn0adtaah922mwz3vems7jqf6hpemdppu2mvm75xe/?noTxList=1

As said before, the response will be in a *json* format and will be handled thanks to the *jsonlite* package. But due to the multiple ways to manipulate the data, it is suggested to look the source code on the GitHub repository: `github.com/PopBogdan97/bCashReader`


```
addrStr:                   "qrn0adtaah922mwz3vems7jqf6hpemdppu2mvm75xe"
balance:                   0
balanceSat:                0
totalReceived:             0.34323506
totalReceivedSat:          34323506
totalSent:                 0.34323506
totalSentSat:              34323506
unconfirmedBalance:        0
unconfirmedBalanceSat:     0
unconfirmedTxApperances:   0
txApperances:              9504
```


Figure 4.2: Example *json* response

4.3.1 Commands

Due to the multiple data formats provided by the Service Providers and the different information that can be retrieved from the Bitcoin Cash blockchain, one can subdivide all the implemented commands in four macro categories.

Address Commands. All the next commands give the possibility to make different type of queries on the address details by giving only address identifier as the argument of the function. It could be in both LegacyAddress (those similar to Bitcoin) and Cashaddr (the new format eplained here **??**).

Function	Description
addr_balance(hash string)	Get the balance of a Bitcoin Cash address.
addr_history(hash string)	Get the general data (history) of a Bitcoin Cash address.
addr_totalReceived(hash string)	Get the total amount of Bitcoin Cash received by the address.
addr_totalSent(hash string)	Get the total amount of Bitcoin Cash sent by the address.
addr_txApperances(hash string)	Get the total amount of transactions made by the address.
addr_txs(hash string)	Get the all the transactions made by the address.
addr_unconfirmedBalance(hash string)	Get the unconfirmed balance of a Bitcoin Cash address.
addr_unconfirmedTransactions(hash string)	Get the total amount of unconfirmed transactions made by the address.
addr_utxo(hash string)	Get the unspent transactions (UTXO) of an address.

Block Commands. The underlying commands allow to retrieve all the information about the blocks by inserting as parameter of the function three types data: *Date, block's hash*, and *block's height* in the blockchain.

Function	Description
block_byDate("yyyy-MM-dd" string)	Get all the blocks in a specific Date with the format yyyy-MM-dd.
block_byDateCount("yyyy-MM-dd" string)	Get the block count in a specific Date with the format yyyy-MM-dd.
block_byHash(hash string)	Get the information of a specific Bitcoin Cash block (without the transactions).
block_byHashRaw(hash string)	Get the raw information of a specific Bitcoin Cash block.
block_byHashTxs(hash string)	Get the transaction hashes of a specific Bitcoin Cash block.
block_byHeight(height integer)	Get the information contained in a specific Bitcoin Cash block.

Transaction Commands. The later functions return the information of a specific transaction and its inputs and outputs by knowing the *transaction hash*.

Function	**Description**
txs_byHash(hash string)	Get the all the information of a specific transaction.
txs_byHashIn(hash string)	Get the all the information of a specific transaction inputs.
txs_byHashOut(hash string)	Get the all the information of a specific transaction outputs.

To see the description of the alle values retrieved by the functions visit the Blockchair documentation[**?**] in section *Dashboard/transaction*

Chain Commands. These last commands give live information and statistics on the behaviour of the Bitcoin Cash blockchain.

Function	**Description**
chain_currency()	Get the Bitcoin Cash current value..
chain_stats()	Get the Bitcoin Cash current blockchain stats.

To see the description of the alle values retrieved by the functions visit the Blockchair documentation[**?**] in section *Dashboard/stats*

5 Usage samples

At this point, we can start using the package functionalities to retrieve some data from the Service Providers we chose and make some analysis from which we shall extract useful information.

5.1 Block size

The Block size was the primary reason for the Hard Fork on August 1^{st}, 2017. Orginally, the dimension of Bitcoin's blocks was of 1 MegaByte and in order to increase the scalability of the network, with Bitcoin Cash, it has been raised to 8 MegaBytes. A relevant analysis could be done by inspecting all the blocks to get the real size of each mined block since the BCH Hark Fork.

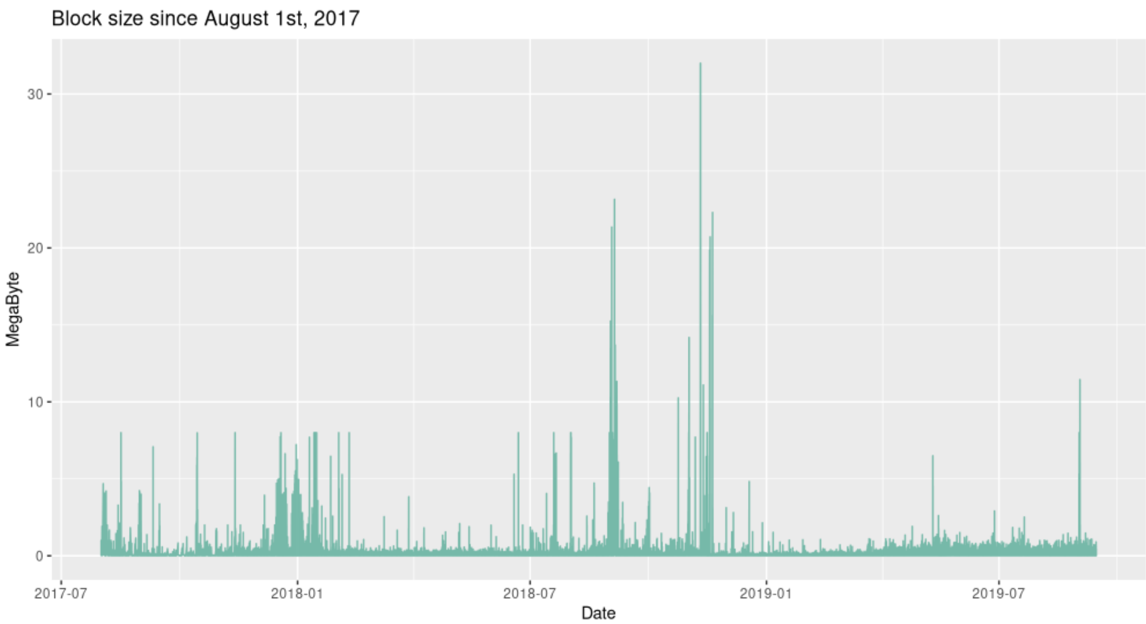

Figure 5.1: Graph of the dimension of each mined block since August 1^{st}, 2017

Thanks to this Graph, it is possible to determine that almost one year later, the block size has incremented again bringing the dimension of a block up to 32 MegaBytes. But, even if there is a lot of space in each block, it is not entirely used, actually for the most of the time is used just a small portion of its capacity.

5.2 Transactions

Growing the block size, was only a mean for increasing the number of the transactions in each block. Thus, not only the scalability has been improved, but also the fees for every transaction have been lowered.

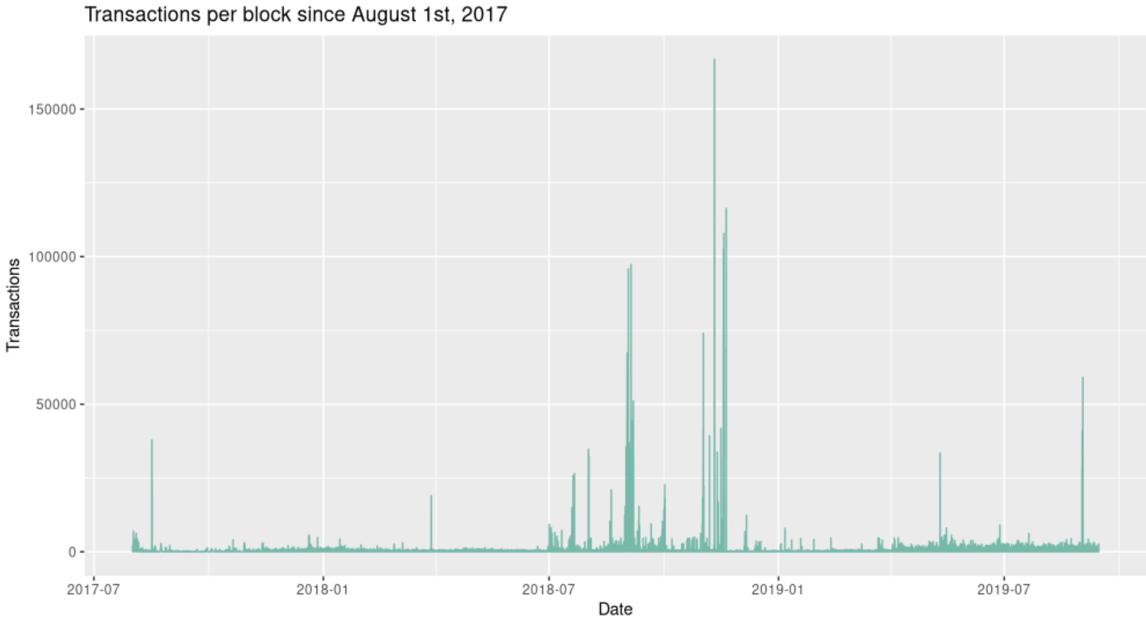

Figure 5.2: Graph of the number of transactions per block since August 1^{st}, 2017

By watching the above graphs it is possible to see and to analyse the behaviour of the transaction count related to the size of each block. Moreover, a comparison shall be made between Bitcoin and Bitcoin Cash throughput. The BTC network can handle a maximum of 7-8 TPS[**?**]. Meanwhile, by reading the graph and by making a little bit of math, the throughput of the Bitcoin Cash network can be estimated. Considering that a block is mined every 10 minutes, the formula to find the throughput is:

$$TPS = Transactiosperblock/minutesperblock/60seconds.$$

The above graph shows that the BCH throughput is more than: $150.000/10/60 = 250TPS$

5.3 Mining Pools

As already mentioned, the consensus algorithm adopted by Bitcoin Cash is the Proof of Work. With the growing of the Bitcoin Cash network, also the number of miners has grown. In fact, there is a massive electricity consumption for the mining process. That leads also to a massive computational power. Furthermore, trying to mine a block as an individual is a waste of resources, because, by comparing your computational power with the one of the entire network, your probability to close a block before the entire network is extremely low. For this reason mining pools have been created. In this way, if someone wants to participate at the mining process, he/she has just to join a mining pool where all the computational power of the participants is combined. At this point, the probability for a pool to mine a block is much more, and in case it succeeds in closing a block, the rewarding is distributed to all the participants proportionally to their computational power.

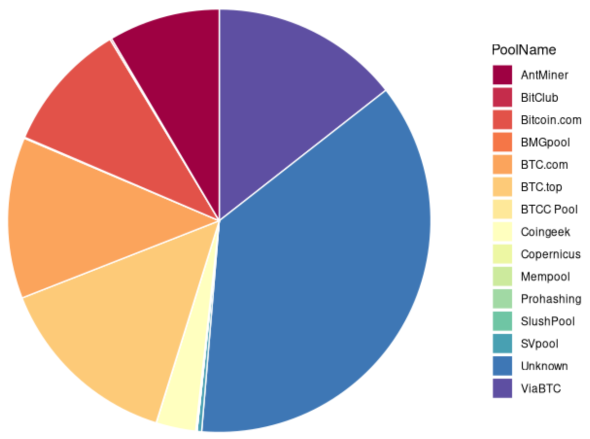

Figure 5.3: Proportion of the mined blocks since Auguts 1^{st}, 2017 by each mining pool

In the graph above, we can see in proportion how many blocks have mined every mining pool and how many pools joined in mining Bitcoin Cash ever since his birth.
With the next graph, we can extend the analysis of the mining pools by spreading them over the time. By doing so, we can also have an overall idea on how much computational power each pool spent in a specific period.

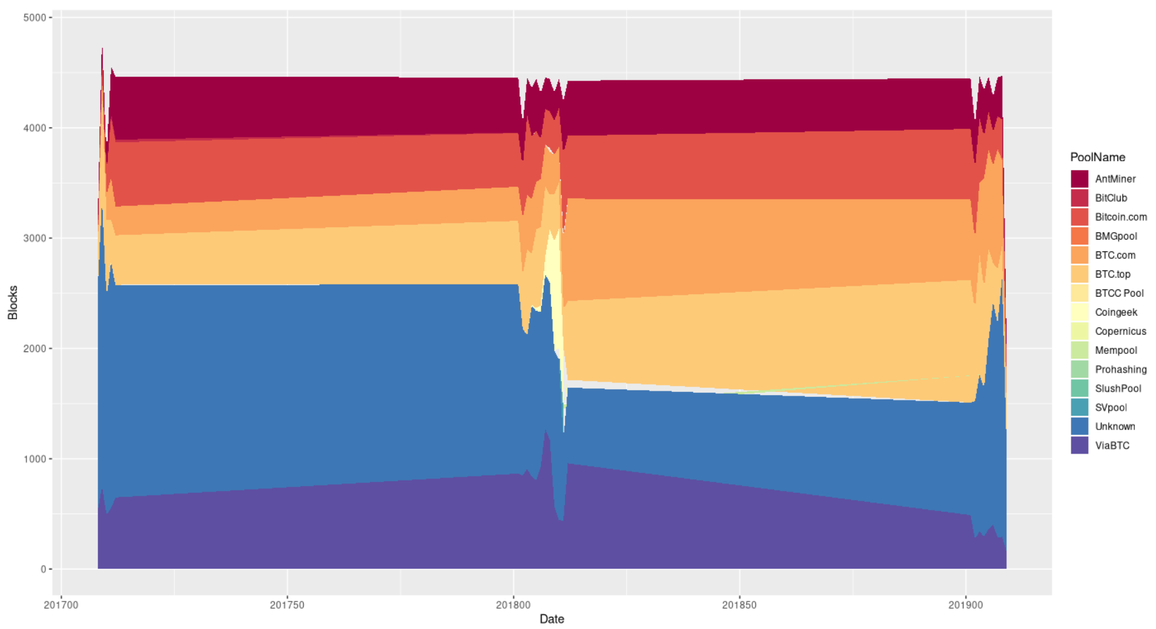

Figure 5.4: Number of blocks mined by each mining pool since August 1^{st}, 2017

5.4 Conclusion

I am satisfied with the possibilities of data analysis this implementation gives to anyone that wants to take a look at the Bitcoin Cash blockchain. The work done for this thesis gave me a chance to go deeper into the history and the working mechanism underlying the blockchain technologies. Furthermore, by going deeper into the study of this kind of technology, I feel I gained enough knowledge to be able in the future to develop new applications based on it.