LUISS

Department
of Economics and Finance

Course of Empirical Finance

Forecasting bitcoin volatility: Does GARCH provide extra information once VIX is included as a regressor?

Prof. Paolo Santucci De Magistris

SUPERVISOR

Prof. Proietti Tommaso

CO-SUPERVISOR

716311

CANDIDATE

Academic Year 2019/2020

Table of contents

Introduction

Inflation, inflation and again inflation.

Injecting more money into the economy causes prices to rise. As we all know from macroeconomics classes the optimal inflation rate is around two percent, which means a yearly increase of goods and services prices by two percent every year. The aim of increasing money supply is to boost the economic growth, but at the same time more money in the economy means also that the buying power of one cash dollar in our pocket today will be, ideally, 98 cents next year, and less nearly every year to come. Since 1913, when the Federal Reserve took over the United States Dollar, we have seen that the United States Dollar has decreased in value by almost 98%. As we can imagine from these few numbers, inflation represents a much higher tax burden on our net worth that hurts our net worth held in a currency every single year. For instance, in 1913 with one dollar we could buy 16 loaves of bread, but today with a dollar we can barely buy one loaves of bread. This is a clear and evident proof that the value of our cash is slowly fading away as the years go by. At the same time, by taking that same dollar and investing it at 2% in 1913 would now be worth $7.24, more than 600% return, versus a near total loss (-98%). Now you may ask what all of these has to do with Bitcoins; well, the answer is simple. Bitcoin represents a technological innovation able to tackle the above described issue, but that is not all. In fact, it is far more than a digital currency. It is a system based on mathematical truths which allows everyone in the world to be their own bank, creates a currency free from taxes and banking fees and provides a solution to the double spending problem. In fact, Bitcoin offers an alternative to trust-based model, composed by the two parties willing to make a financial transaction and by a “middleman” (or financial institution) who represent a trusted third party. The current financial system is based on this trust-based model, which presents several issues and costs for the parties involved in the transaction. Basically, we are paying a “trust premium” to the middleman for being our counterpart. Can we avoid paying “trust premium”? Yes, we can! Bitcoin can tackle this big dilemma of trust by using an electronic payment system based on cryptographic proof instead of trust, allowing a peer to peer transaction without the need of a trusted third party. Moreover, transactions that are computationally impractical to reverse would guarantee a protection to the sellers from fraud and, by introducing an escrow mechanism, we can easily guarantee the same level of protection for the buyers. The purpose of Bitcoin is to create a system that substitute the trust-based model and therefore removes all the weaknesses that the trust-based model brings. In fact, a financial system that cuts out these middlemen could be faster, cheaper and more secure.

This paper is structured as follows. The first chapter provides an overview of what Bitcoins are and examines the underlying technology behind them; furthermore, I will discuss the advantages of this technological innovation and the potential downside it implies. In the second chapter, I will present two different methods used to forecast future volatility: GARCH and VIX. Firstly, I will provide an overview of each methodology and then I will apply the models to our dataset. All of this to provide an answer to the question of this paper: “Does GARCH provides extra information about future realized volatility, once VIX is included as a regressor?”.

Chapter I: Overview of Bitcoin and Blockchain

1.1 Introduction

In a very naïve and perhaps lazy way, Bitcoin is defined as a digital currency that allows peer to peer payments. Bitcoin, and in particular the technology behind it, represents a technological innovation able to tackle many problems, not only in the financial industry, but also in many other fields. The story of Bitcoins begins on October 31 of 2008, during the Great Financial Crisis. They have been created by someone with a pseudonymous of Satoshi Nakamoto, whose identity is unknown, and it doesn't matter; the only thing that matters is the big innovation it brought to the world. Satoshi Nakamoto provides us a definition for Bitcoins, which is: "…a new electronic cash system that's fully peer to peer, with no trusted third party" [1]. Following this announcement, on January 3 of 2009 the first Bitcoin block, block 0, was mined. The Block 0, or the first transaction over, was carrying the following text with it: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks". The reference was to an article[2] written in the newspaper "The Times" about the possibility of a second wave of bailouts. However, the real innovation that Satoshi Nakamoto brought to us is represented by the technology underlying the Bitcoins, the Blockchain. The Blockchain is a term that has different meanings based on the kind of sector we are in; in particular, for what concern the financial sector, it refers to a distributed ledger which represents a way to store, organize and modify data contained within it. This information is immutable and highly secured; for this reason, it can be used in many sectors. For instance, in the health sector, the Blockchain technology can be implemented to guarantee a highly efficient, secure and fast way for the exchange of health information. In what follows, I will provide an overview of how the Blockchain technology underlying Bitcoins works.

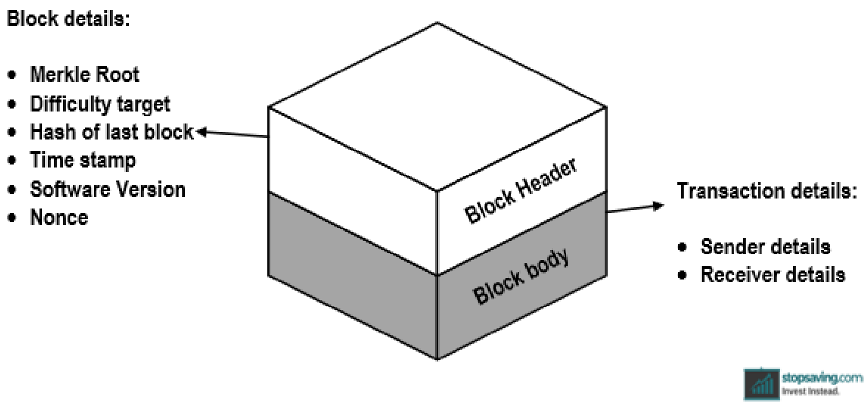
1.2 Blockchain

Bitcoin is implemented through the Blockchain framework and it represents the first application of Blockchain technology. As underlined before, the Blockchain refers to a distributed ledger which represent a way to store, organize and modify data contained within it; this information is immutable and highly secured. In the Blockchain all data are separated into blocks, where each block contains several elements and is linked to the previous one, thus creating a chain. In each block, elements are separated into the block header and its transactions. Block header contains the

[1] Original source is Cryptography Mailing list at metzdowd.com
[2] See: https://www.thetimes.co.uk/article/chancellor-alistair-darling-on-brink-of-second-bailout-for-banks-n9l382mn62h

essential metadata about each block (e.g. timestamp, block height, etc..), while the transactions account for most of the data. Figure 1.1 illustrates an example of a block.

Figure 1.1 – *Block breakdown*



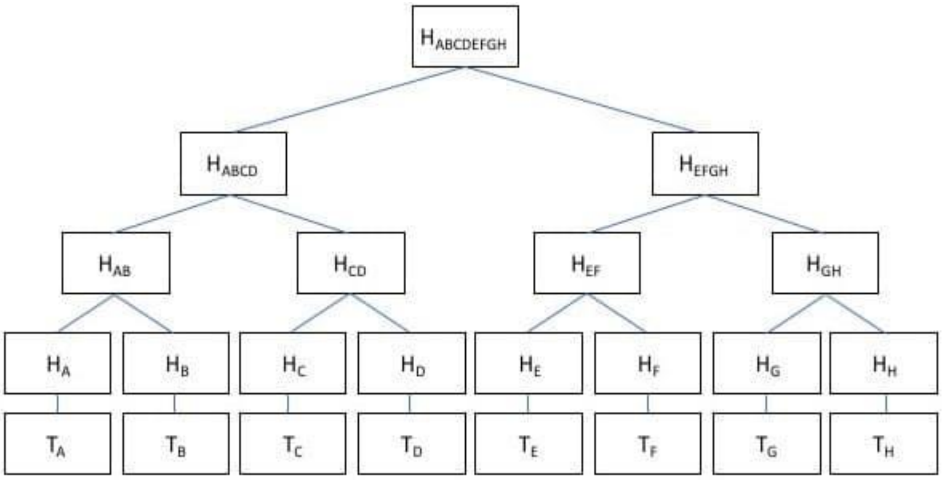
Source: stopsaving.com

The block header contains a series of important data, which are:

1. *Version*, which points out what block validation rules were followed. This is very important, since the blocks must have the same version number in order to be part of the same blockchain.
2. *Previous block hash*, which is a 32-byte field that contains the hash[3] of the previous block header. This contains a pointer to the previous block, an important feature that makes sure that data in a block cannot be changed without modifying all the previous blocks. The block header uses the SHA256 hashing algorithm.
3. *Merkle Root*, which is a data structure used within blocks, where the leaves of the Merkle Tree represent the transactions included in the block, and makes sure that none of those transactions can me modified without modifying the entire header. The Merkle Tree consists of all TXIDs (Transactions ID) of transactions in the block. The TXIDS are placed in the order established by the consensus rules: at the top we find the coinbase transaction (rewarding the miners with newly issued coins), the following transactions follow the rule, according to which a transaction output must be placed prior to a transaction input. The Merkle Root can always be traced back to bottom line of the data tree structure and it represents an efficient wat to verify the data.

[3] A hash is a digital fingerprint, that is unique

Figure 1.2 – *Merkle Tree*



Source: btc-investor.net

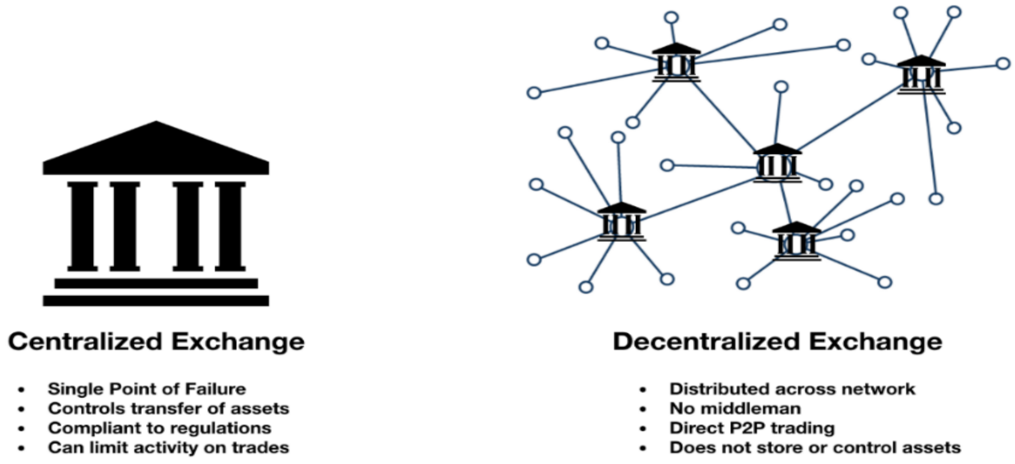
As we can see in the Figure 1.2, the root of the tree is H(ABCDEFGH), which is the hash of all transactions from T(A) to T(H). The hash H() of the root is a concatenation of all the transactions in the blockchain.

4. *Timestamp*, which is the time when the miner started hashing the header, measured in Unix epoch time. According to the Bitcoin protocols, the block propagation time must be of 10 minutes on average (but it can be much higher, it depends on the difficulty target).
5. *Difficulty Target*, which represents the difficulty of finding a new block. If in the network there are too many miners, the difficulty increases to control the supply of Bitcoins. When the difficulty is too low, the timestamp falls below 10 minutes, and this could exhaust the supply of Bitcoin very quickly; so, in order to avoid this, the protocol increases the difficulty.
6. *Nonce*, which represents the variable that miners change to modify the block header hash. It corresponds to the number that miners have to discover to become the block validator and so to earn the incentive (a newly created bitcoin).

The Blockchain works in a very similar way to a linked list, but with a huge difference: the references in a blockchain are cryptographically secured and therefore cannot be changed without affecting the integrity of the data, while this is not true in case of a linked list. The cryptographically secured references are constantly checked for validity to avoid any attempt from an attacker to change data in an existing block. This means that the Blockchain technology can be defined as a chain of blocks, where each block contains a set of information about previous block in the chain. As underlined before, the data on the blockchain are immutable, but how? It is thanks to a strong consensus mechanism and a large number of nodes (miners) in the network, which make sure that the majority of participants follow the protocol and reject invalid blocks, since they have incentives to do so.

Blockchain technology was implemented for the first time with Bitcoins as an alternative to the existing centralized banking system (or the trust-based model), with a decentralized one.

Figure 1.3 – *Centralized and Decentralized System*

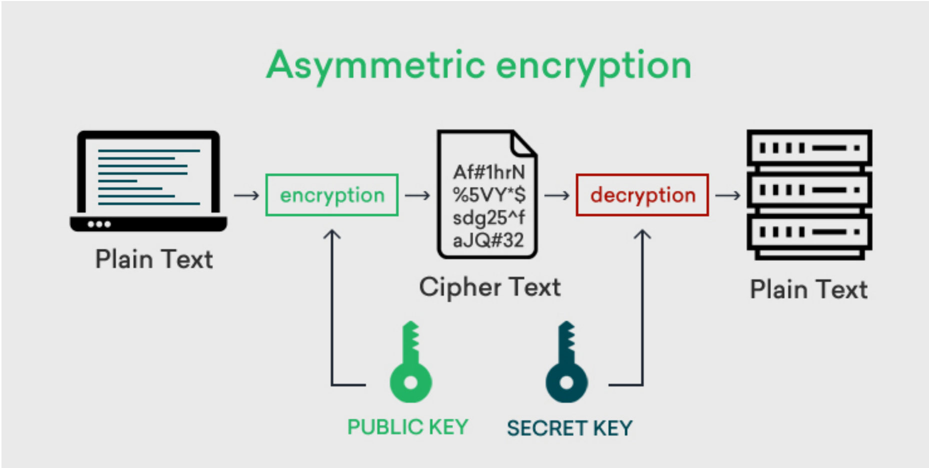


Source: hackernoon.com

The decentralized system brings many new features and, in particular, several advantages; some of the main ones are the following: the absence of a "middleman", which translates into lower transaction costs; the authority of control is no longer given to a single authority, but it is being distributed across a network; the risk that your data may be stolen is close to zero.

Basically, Bitcoins are a peer to peer version of electronic cash built upon a distributed peer to peer network. But, what is a peer to peer network (P2P)? A practical example may be the Torrent network: files are not downloaded from a single source but from multiple sources where data are already downloaded; these multiple sources can be considered as nodes, where each single node contains a copy of the file we are trying to download. Bitcoin works in the same way as Torrent in the example above, where in each node the bitcoin's transactions are recorded and each node in the network has its own copy of transactions done in the entire network. Moreover, every node in the network gets updated whenever occurs a new transaction. But does it mean that each node in the network may see my transaction details? Well, while some data, like sender and recipient of a transaction, are visible to all the nodes in a blockchain, sensitive information is encrypted using a public and private key, meaning that only the owner of a transaction can decrypt the information with his private key. The following figure illustrates the process.

Figure 1.4 – *Asymmetric Encryption*



Source: clickssl.net

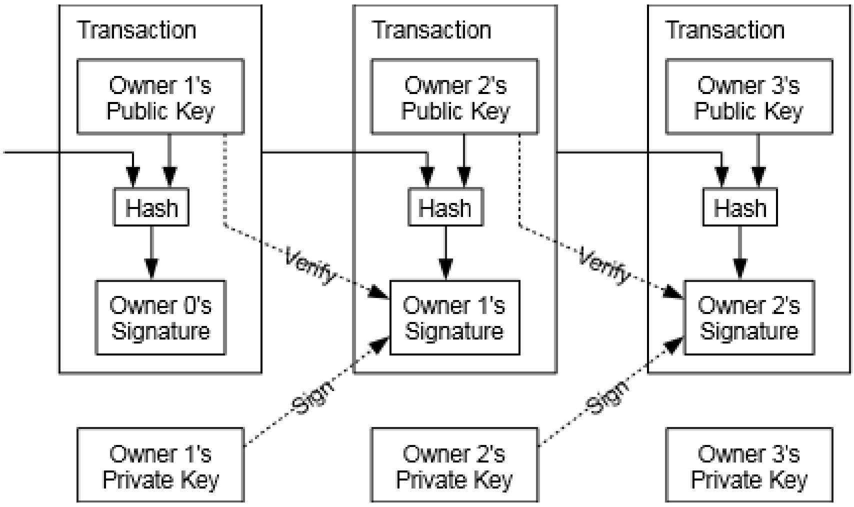
In fact, one of the core concepts that make Blockchain work is the concept of asymmetric cryptography, also known as public-key cryptography. With symmetric cryptography, a message is encrypted and decrypted using the same key; with the asymmetric one, instead, two different keys must be used: the public key and the private key. The keys always come in pair, meaning that the message is encrypted with a public key and decrypted with the corresponding private key, and vice versa. This also means that your key pair is your identity on the blockchain. In particular, you receive funds with your public key and spend them with your private key. You can share your public key with anybody, but you should not do the same with your private key, because it allows to spend your money. If somebody gets it, they can access and steal your funds. In case you lose your keys, you cannot recover them, and your funds gets lost. However, to tackle this possibility there is a sort of recovery mechanism with many wallets called mnemonic phrase (usually of 12 or 24 words) that allow you to recover your keys.

1.3 Transactions

The idea underlying Bitcoins is to have a completely nonreversible transactions system, but, how can this be achieved? Let's look at how transactions should work in such system.

Today banks record each transaction as a plus and minus in their ledgers (databases), while bitcoin's ledger is the blockchain (a publicly distributed database) that record every Bitcoin in existence and every bitcoin transaction ever made. This distributed database is always balanced, because no one can take out Bitcoins from the system and put them under the mattress. In fact, Bitcoin is a digital coin that can be represented as a chain of digital signatures whose aim is to prove who owns what (in this case who is the owner of the coin). The process is very simple, every coin's owner has two keys: private key and public key. The private key, known only by the coin's owner, represents a proof of the ownership and also the key to access the wallet[4]; on the other hand, the public key is the proof of the ownership that can be disclosed to the public since it does not give access to the owner's wallet. Figure 1.5 shows how transactions works.

Figure 1.5 – *how transactions work*



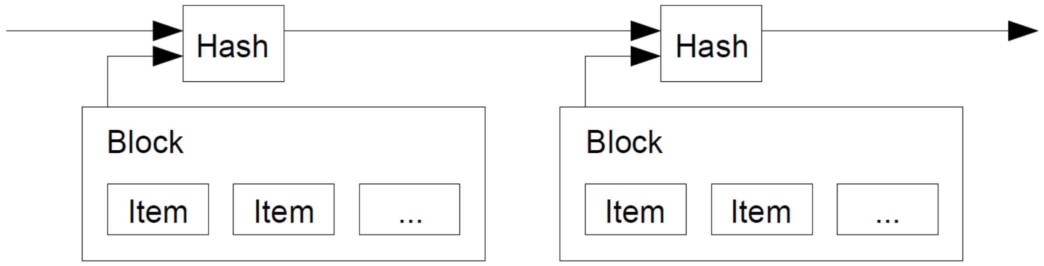
Source: "Bitcoin: A peer-to-peer electronic cash system"

Assume that the Owner 1 wants to transfer one Bitcoin to the Owner 2: he can do it by digitally signing a hash of the previous transaction and the public key of Owner 2 by using his own private key. This means that Owner 2's public key has been signed by the private key of the Owner 1 and we are able to verify it, since we have Owner 1's public key. The hash of the block two in the Figure 1.5 is the hash of the previous transaction. This information is then added to the end of the coin and the payee can verify the chain of ownership by checking the public key of the previous owner. In plain English, when a Bitcoin is sent from one digital wallet to another, what is actually being sent is the control over that part of the database. However, this is not the true innovation of bitcoin, since digital signatures were born years ago. The real innovation of bitcoin is the solution provided to the double spending problem, through the creation of a system where each transaction is public and the network (blockchain) collectively has to agree on what is true and what is not. In order to accomplish this without a trusted party, all miners should cooperate to have a single version of truth, a single record of transactions and a single distributed ledger. If most of mining nodes agreed that a specific transaction was the first received, then the payee has the proof against the double spending.

[4] A wallet in the cryptocurrency world is a software or a physical devise where our public and private keys are stored.

To get the single version of truth, a timestamp server is needed, where each timestamp contains the previous timestamp in its hash; this represents a proof of history that makes sure that there exists one single truth in the system. The following figure shows how it works.

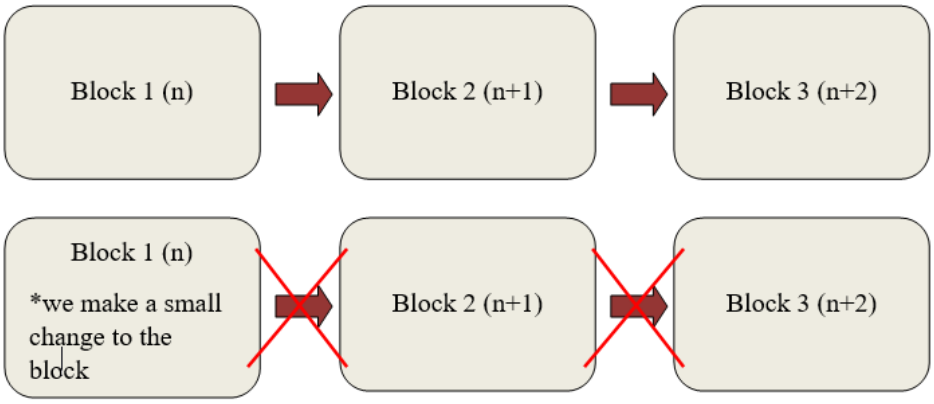
Figure 1.6 – *Timestamp server*



Source: "Bitcoin: A peer-to-peer electronic cash system"

All blocks in the Blockchain are cryptographically linked together and the whole identity of the next block (n+1) is determined by this linkage. This means that if, for example, the starting block is modified, the link to the second block will break and the whole identity of the second block will change, and also the link to the third block will break and its whole identity will change (this means you have to remind everything); Figure 1.7 illustrates this process.

Figure 1.7 – *Attempt to modify a block*

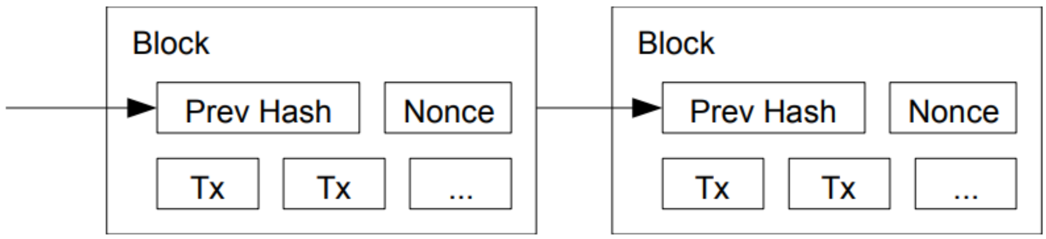


Source: Created on Word

If we try to change something in the Block 1, even a single bit, the whole identity of the block will be completely different, meaning that the identity of the next block will be completely different too, since it is based on the identity of the previous one. This means that, if we change something in the past, the chain will break after that point and this is a part of the whole security mechanism, because to change something in the past you have to remind everything. Moreover, the hash of the previous block is used to create the hash of the next block (and so on).

However, implementing a timestamp server is not easy. To implement it, we need to use a proof of work system, which looks for a value such that, when hashed (e.g. SHA-256[5]), the hash starts with several zero bits; its aim is to make the cryptographic connection, necessary to create this chain of blocks, hard to make. To do this, it requires a lot of hash power (electricity and time). Once the hash power used satisfies the proof-of-work, the block cannot be changed without redoing the work (if we break the chain it will be hard to put everything together). Moreover, as later blocks are chained after it, the work to change the block would include redoing all the blocks after it. Figure 1.8 provides a representation of how this system works.

Figure 1.8 – *Proof of Work*

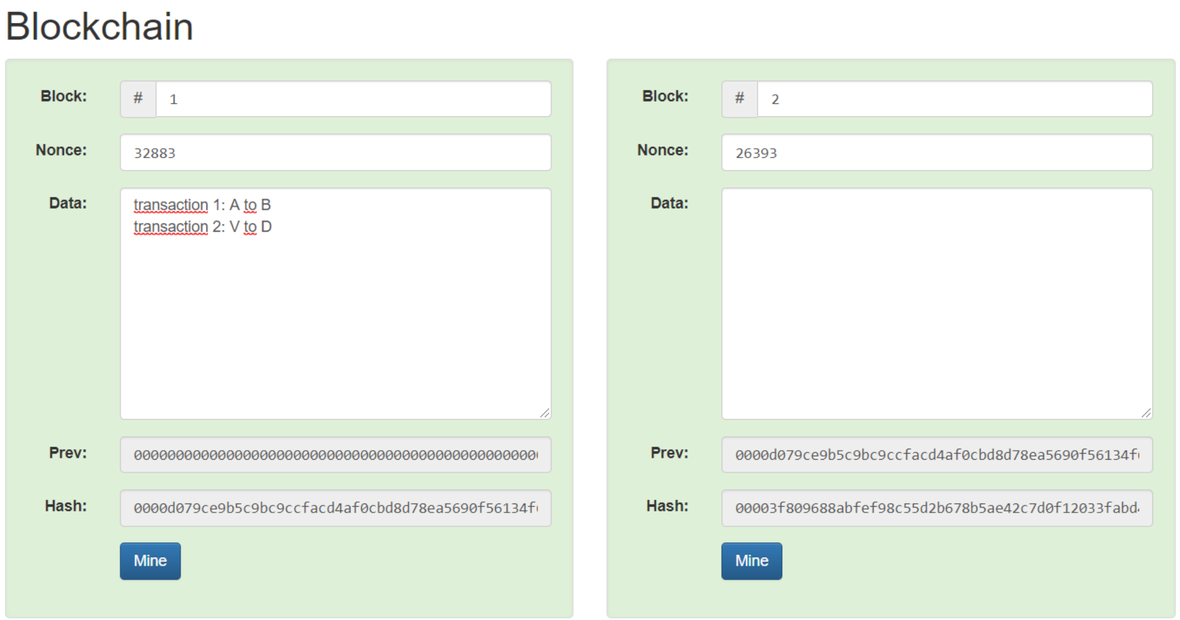


Source: "Bitcoin: A peer-to-peer electronic cash system"

The proof of work has also another important function, which is helping to determine the single version of truth. In the proof of work system, the truth is represented by the longest chain, since it has the biggest proof of work effort invested in it. For instance, assume we have two chains: one made of 4 blocks and the other one made of 3 blocks, both claiming to be the real chain. The network will accept the 4-block chain; if a majority (51%) of hash power is controlled by honest nodes, the honest chain will grow faster and outpace any competing chains. As stated before, proof of work system makes modifying a past block a high burden to carry, because an attacker willing to modify a past block would have to do from the beginning the proof of work of the block he or she wants to modify and also of all blocks after it and then surpass the work of the honest nodes (the probability for an attacker of catching up diminishes exponentially as subsequent blocks are added). Figure 1.9 illustrates the process.

[5] SHA-256 stands for Secure Hash Algorithm 256-bit and it is used for cryptographic security

Figure 1.9 – *Proof of Work applied*



Source: andersbrownworth.com

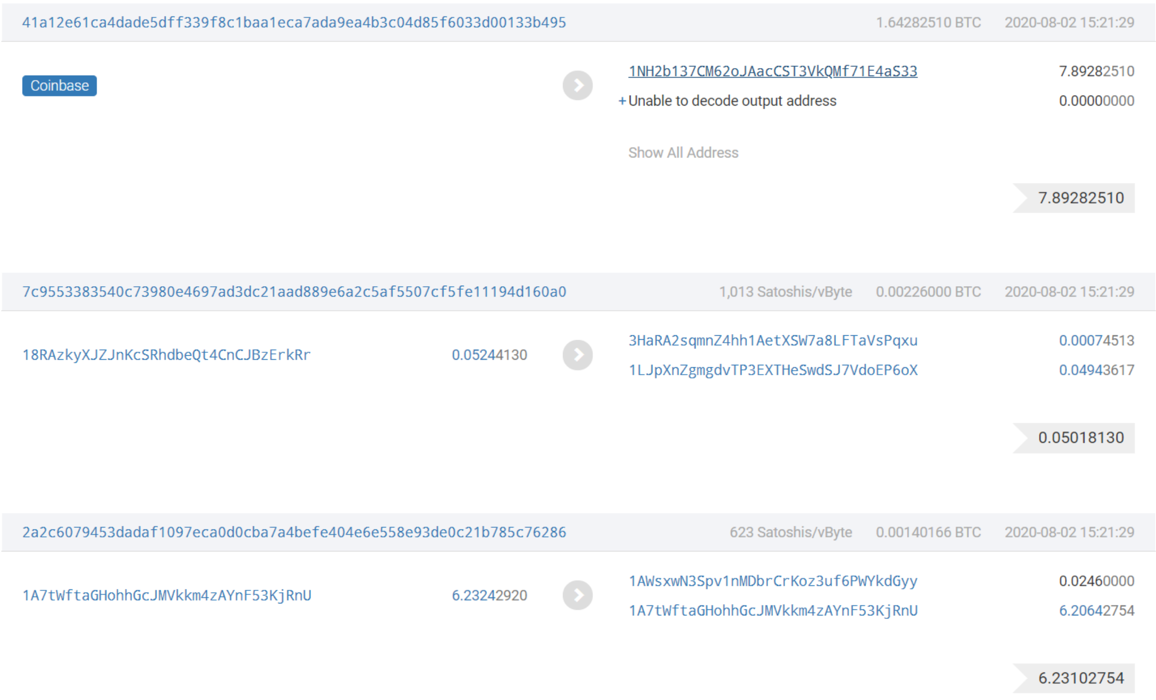
When we press "Mine" (Figure 1.9), the proof of work algorithm will start, and it will produce a hash such that it starts with 4 zeroes. Block 1 represent the concept of zero bites. The only way to find this hash is by guessing the nonce of the Block 1 itself. So, assume that we try to guess the nonce; unless we guess correctly all the numbers, the blockchain will not work and it is almost impossible to do it. When we mine, we basically guess until we find the correct nonce. In this way we can mine this block and thus we are able to validate all these transactions; this brings us to the fact that we need to spend hash power to do these links. Now, if for instance we change something in the block 1, even a single number or a letter, we need to remine the whole chain.

The beauty of proof of work comes into the fact that it is hard to change previous information. As underlined before, miners always consider the longest chain to be the correct one and will keep working on extending it. However, sometimes we may have a situation in which two miners are able to come up with the next block at approximately the same time and so competing with each other. This happens because bitcoin's network is unstructured. But which block will win? Some miners will take the Block 1 and start mining upon it the next block, and others will start mining on the other block; it is all about who will be able to mine the next block faster. Therefore, if Block 1 is the fastest one, then the miners who picked the second block will just discard the second chain and will pick the chain 1, since it is the longest and will be the main chain. This summarize the way in which miners reach a consensus when there are two conflicting versions of true.

1.4 Incentive

By keeping the blockchain secure and functional, miners are rewarded; for instance, the first transaction in a block is a special transaction that reward with a newly created coin the creator of the block. The following figure illustrates the process.

Figure 1.10 – *Miner's incentive*



Source: https://btc.com/

In Figure 1.10, if we look at a random block, we can see that the first transaction always comes from nowhere and goes to the miner; the other transactions comes from other wallets. These first transactions are called "coin-based transaction"[6]. In this case, the miner is rewarded with 7.9 bitcoins; this reward not only support the network, but also provides a way to put newly issued coins into the system, since there is no central authority that can decide how and when to issue them. Moreover, there is another incentive that miners receive; it is represented by transaction fees. In such system transaction fees are not fixed, but time varying. There are two main factors in the determination of transactions fees: network congestion and transaction size (these two factors determine also the time taken for a transaction to be confirmed). For instance, if we decide to send a transaction with very low fees attached to it and the Bitcoin Mempool[7] is full, then the miners

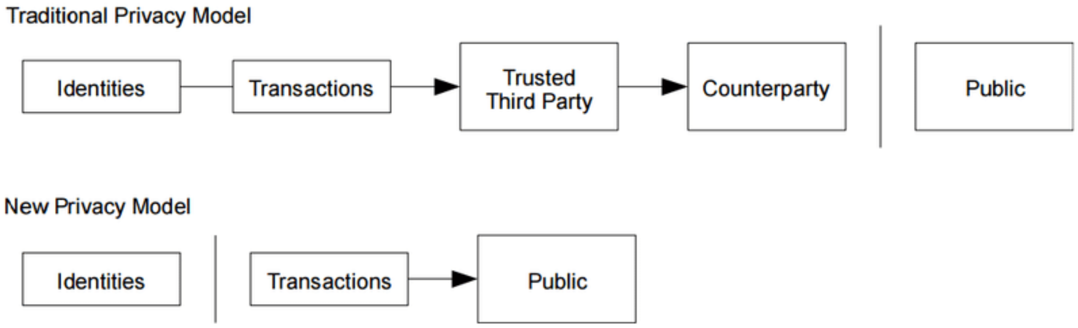
[6] Technical term of the first transaction in the block that goes to the miner
[7] A Bitcoin Mempool is a collection of all Bitcoin transaction awaiting verifications and confirmation

will take a lot of time to confirm our transaction. In fact, miners, just like enterprises, aim to maximize their profit and, for this reason, they first prefer to execute those transactions where they can maximize their profits (or transaction fee). This means that it could take hours for the transaction to be confirmed. However, if we are willing to pay a higher transaction fee, then we can get the confirmation of our transaction in 10 minutes, which is the time necessary to mine a block. Moreover, it is also worth to note that for our transaction to be completely validated (bulletproof), the Bitcoin community requires six such confirmations. This means that, if there is any network congestion and the fee attached is high, the transaction can be successfully processed in an hour. On the other side, we should not forget about the transaction size; transactions that occupy more space need more work for validation, which means that they will carry a higher fee to be included in the next block.

1.5 Privacy

In the traditional privacy model, privacy is represented by an access to information limited to the parties involved in the transaction and the trusted third party. Bitcoins have introduced a new model, which ensures a much higher level of privacy. Figure 1.11 illustrates the two models.

Figure 1.11 – *Privacy models*



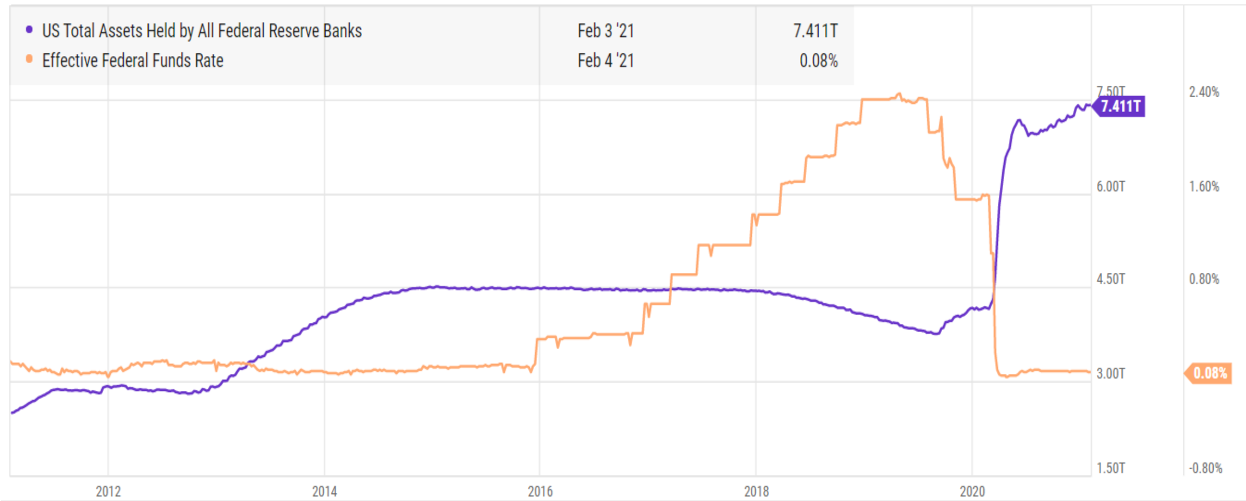
Source: "Bitcoin: A peer-to-peer electronic cash system"

In this new model, when someone sends an amount of money to someone else, the public can only see the amount sent and the address from where it comes from, but without linking the transaction to any physical person. Addresses play a key role in this process; these are produced by hashing the public key twice. You start with a private key, multiply it by the generator points and produce a public key; you cannot go back from the public key to the private key. This mechanism of privacy is not something new to us, since stock exchanges operate in the same manner, where only time, price and size of individual trades are made public, without linking it to the specific parties involved in the trade. However, it is not enough, since it is still possible to link that specific address to a real person. The big problem here is that if the public key of a person is revealed, then it is possible to link all other transaction made by the same owner. As someone may think, why do not we just make some reversal engineering and try to retrieve the public key from the address? The answer is simple: it is impossible to do; we cannot retrieve the public key from an address, since the address is computed from the hash of the public key.

1.6 Bitcoin as inflationary hedge

One of the greatest advantages of bitcoins, but also cryptocurrencies in general, is represented by their characteristic of being inflation free assets, since their supply is limited. The cost of the 2020 global pandemic is estimated to be worth trillions of dollars, which will represent a high burden for the taxpayers in the next years. But how governments will face such high costs? The answer is very easy: by borrowing more from their central banks, which will lower interest rates and will inject more money into the economy. Figure 1.12 illustrates the total assets held by all federal reserve banks in the US and the effective federal fund rate.

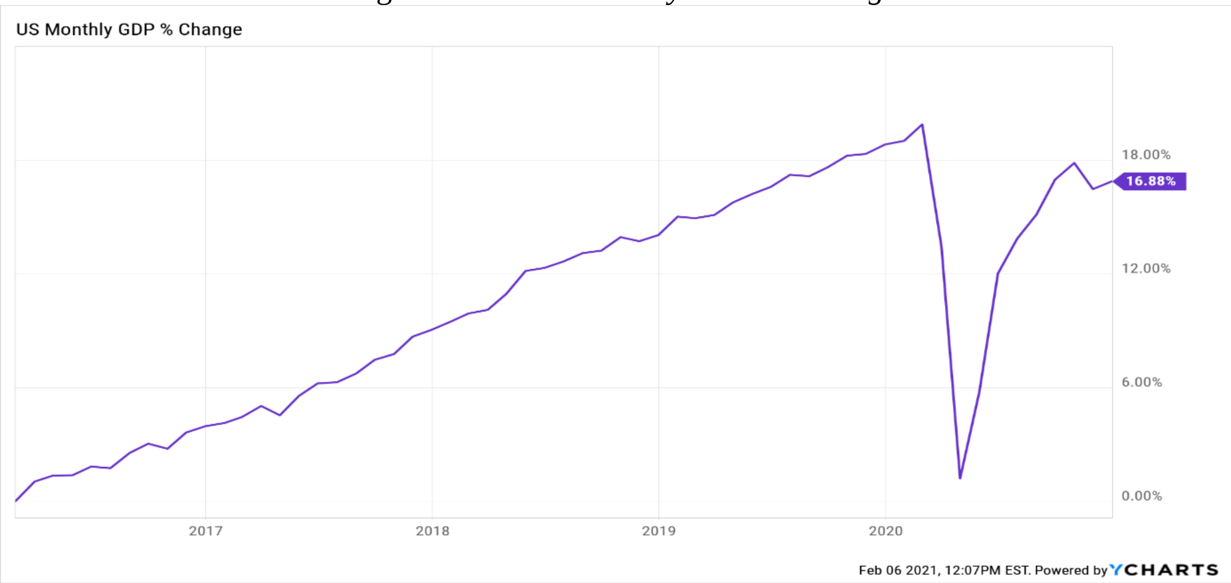
Figure 1.12 – *US total assets held by all federal reserve banks and effective federal fund rate*



Source: ycharts.com

However, injecting new money into the economy is not bad if it is supported by the GDP growth. The following figure displays the US monthly GDP % change.

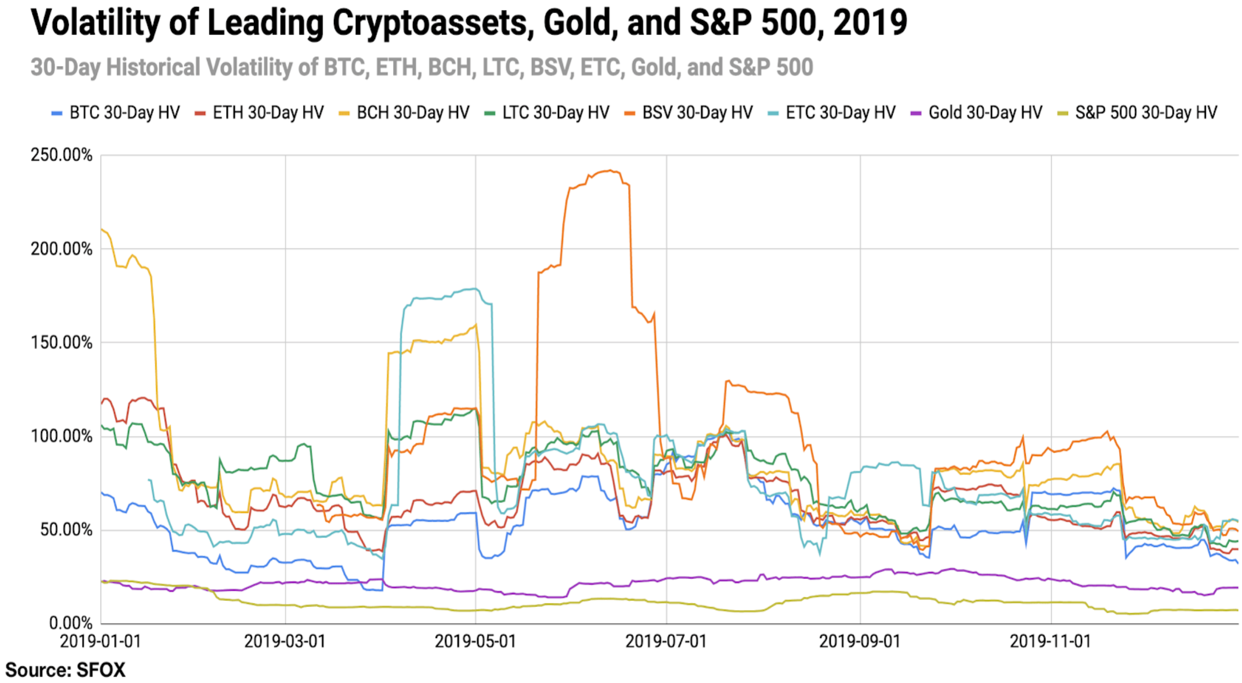
Figure 1.13 – *US Monthly GDP % Change*



Source: ycharts.com

As we can see from Figure 1.13, the newly injected money into the economy is not supported by the GDP growth. But what if we inject more money into the economy without getting the required GDP growth to balance it? Is it bad? The answer is yes. Nowadays, even if the debt is cheap, the debt is still debt and carries certain requirements; in particular, it must be repaid. There are two main ways in which governments can repay their debt: either by increasing taxes and perhaps cut spending or by inflating the debt away. The latter is more likely than the former. This all means that the major currencies are about to get devaluated. Ray Dalio, a well-known hedge fund manager, has made an analysis of debt cycles where he explains that roughly 80% of all currencies that have been created since 1700 are now worth zero and the other 20% have been devalued. The reason for this devaluation is unchecked spending by governments which leads to their currency's demise. In times like these, people are used to buy "safe-haven assets"; for instance, gold is likely to continue to increase in value and stocks are likely to perform well too. And in these times, Bitcoin may be the best store of value. But how? Bitcoin is extremely volatile; for instance, from the 19th February to the 12rd March it lost roughly two thirds of its value, bottoming on the 12th of March at $3858. In the same timeframe, the other traditional "safe-haven assets" such as gold dropped roughly 3%. So, is Bitcoin really a safe-haven asset? Well, by looking at these numbers, it doesn't seem so. Indeed, it is important to remember that Bitcoin is a risky investment due to its high volatility. Figure 1.14 illustrates the volatility of Bitcoin and other crypto assets versus gold and SP500.

Figure 1.14 – *Volatility of BTC vs Other assets*

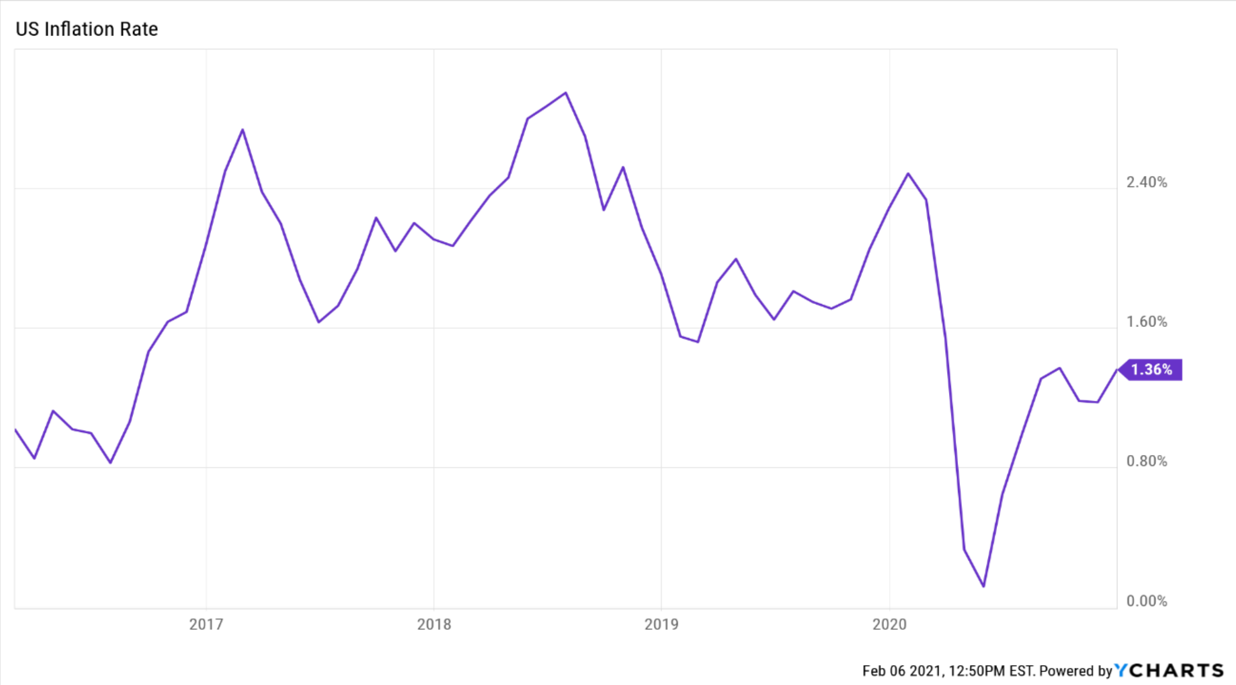


Source: sfox.com

However, even if now Bitcoin is an extremely volatility asset, in the future it will no longer be like that, and its limited supply makes him the best candidate as store of value or as hedge against the inflation. We know that there are only ever going to be 21 million bitcoins (an immutable fact) secured by the mining network. This scarcity means that Bitcoin is a deflationary asset, even more than gold, since the gold supply could increase almost infinitely if we consider the presence of gold in space. For instance, the latest halving, happening this May, reduced the inflation rate to 2.42% per annum[8]. This is a very important fact to account for, especially in periods like this, when the inflation is extremely high. You may be wondering to know how it is possible that the inflation is high when it is only at 1.36%[9]. Figure 1.15 shows the current US inflation rate.

[8] See: https://charts.bitcoin.com/btc/chart/inflation?ref=hackernoon.com#5moc
[9] US inflation rate

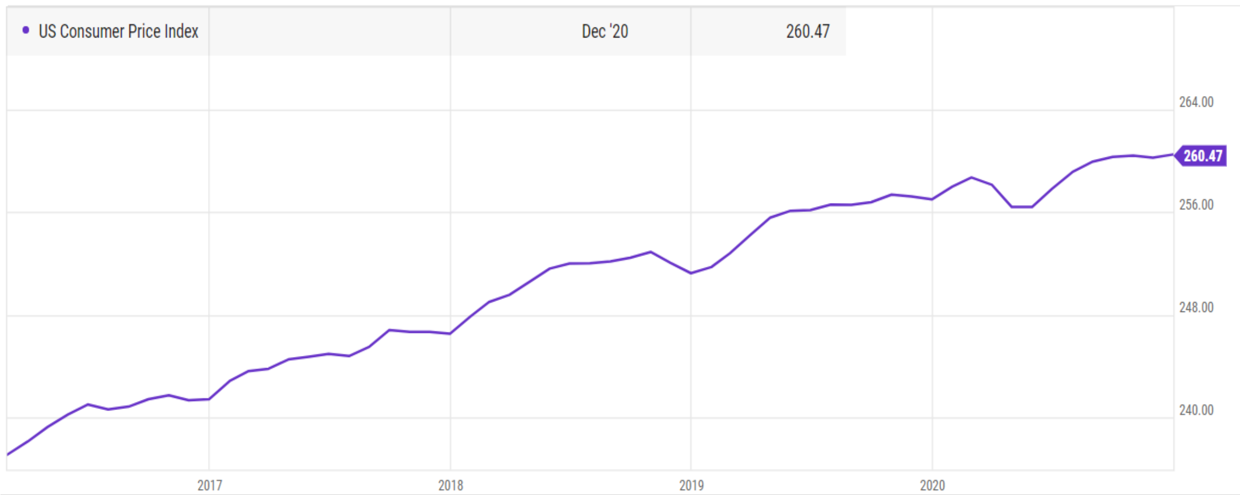
Figure 1.15 – *US inflation rate*



Source: ycharts.com

By looking at the figure above, it seems that trillions of dollars injected into the economy are just free money that do not carry any inflationary burden. But is this really the truth? The truth is that we are not looking in the right place; to find the inflationary burden carried by all the newly injected money into the economy we should look, for instance, at the consumer price index. Figure 1.16 illustrates the US Consumer Price Index.

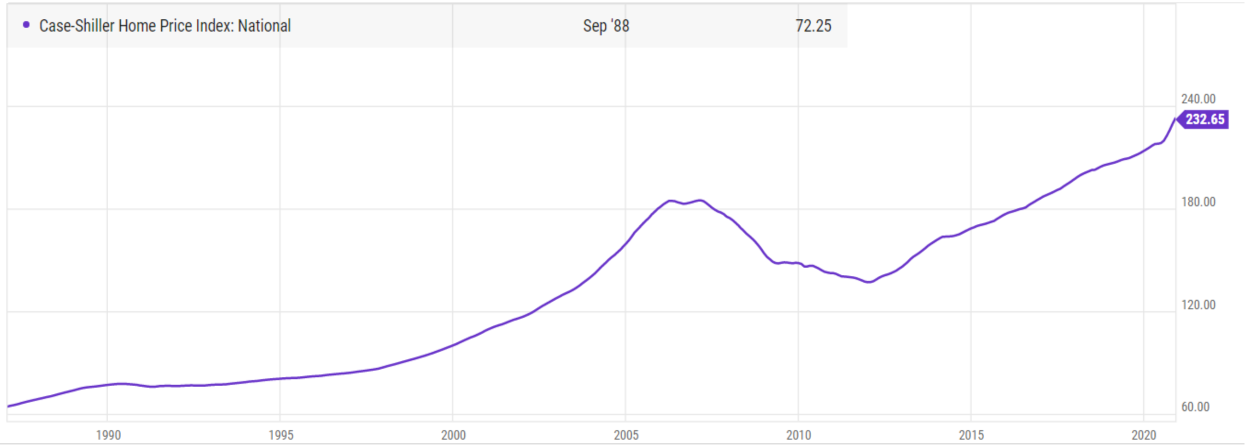
Figure 1.16 – *US consumer price index*



Source: ycharts.com

But how is it possible that US Inflation rate does not rise, while consumer price index keeps rising? To understand the reason, we must decompose the consumer price index into its components to understand the drivers. The components of the consumer price index are food, housing, transportation, etc.; for example, the "food and beverage" category increased by almost 4%, nearly double the inflation target rate (or close to it, if we consider the new revisited target set by the FED). The other big category in which we can find inflationary sins is the "housing" category. Figure 1.17 illustrates the changes in the US housing prices.

Figure 1.17 – *Case-Shiller Home Price Index*



Source: ycharts.com

Finally, you may be wondering if bitcoins may be a good long-term bet. Well, someone believes that. For example, the CEO of MicroStrategy Inc, Michael Saylor thinks that Bitcoins is much safer asset than cash; in fact, on July of 2020 he converted the entire cash position of the company into Bitcoins and he is advising firms to do the same. On February 03 of 2021, MicroStrategy hosted the event "Bitcoins for Corporations", whose aim was to promote Bitcoins as a mean of additional value to companies' shareholders; over one thousand firms took part in this event. On February 08 of 2021, also Tesla announced that it would convert $1.5 billion of its cash balance in Bitcoins and with a future possibility of Bitcoins being accepted as a payment for the purchase of Tesla's cars.

Paul Tudor, a well-known fund manager, in a recent letter to investors published an analysis of bitcoin as an inflationary hedge. In this analysis, he compares Bitcoins with other stores of wealth, like gold and TIPS and he shows how bitcoin outperforms its competitors almost on any front. For instance, bitcoins are much more liquid asset (24/7 trading), they have higher portability (we can send it anywhere in the world within an hour) and higher purchasing power (due to the capped supply). However, it presents a very big downside, the trust. The trust is caused not only by the underlying complexity, but also because most of the times it is used for illegal means.

In fact, during these years we have seen a lot of opposition to bitcoins, that mainly comes from countries with a well-established financial and payment system, where most of the population has a bank account. But what if we take a country with a poor financial and payment system? All the people in the world have one common factor, a mobile phone. In 2019, according to a survey conducted by Facebook, it has been found that 31% of adults around the world do not have a bank account but roughly 65% of them has a mobile phone. We can understand from these few numbers the big innovation that bitcoin can bring in these countries. According to grandviewresearch.com the global market size for the blockchain technology was valued at $1,590.9 million in 2018 and this number is expected to grow at a CAGR of 69.4% from 2019 to 2025. To conclude, Bitcoins have a very bright future ahead of them, which will be even brighter once it will earn investor's trust.

Chapter 2: Volatility prediction with GARCH

2.1 Literature overview

Volatility seeks to capture the strength of the return variation over a given period and, as we know, it is not directly observable. For this reason, we need to rely on different approaches to estimate it. There are mainly two approaches we can pursue to empirically quantify volatility: the first based on parametric models, where the focus is on ex ante expected returns; the second is based on non-parametric models, where the focus is on ex-post realized volatility. The family of GARCH models belongs to the former category. With parametric models like GARCH, we want to parametrize the expected volatility $\sigma^2(t,h)$ over the period $[t-h;t]$ as a function of the past information set I_{t-h}:

$$\sigma^2(t,h) = E[[r(t,h) - E(r(t,h)|I_{t-h}]^2 I_{t-h}]$$

where $r(t,h) = p(t) - p(t-h)$.

The idea behind ARCH-GARCH models is to capture the features of the financial log returns. These family of models assume that the past information set I_{t-h}, underlying the parameterized conditional expectations of volatility, depends solely on observable variables. In the ARCH(1) model, introduced by Engle (1982), the expected volatility is modeled as:

$$\sigma_t^2 = \omega + \alpha(r_{t-1}u_{t-1})^2$$

where $\varepsilon_{t-1} = r_{t-1} - u_{t-1}$ is the past return innovation.

But what does ARCH stand for? ARCH is the acronym for “Auto Regressive Conditional Heteroscedasticity”. “Autoregressive” means that volatility, at a certain point in time, will be a function of the volatility at the prior timestamps; “Conditional” means that volatility of our time series is not fixed over time, but it is based on where we are in time; “Heteroscedasticity” means that residuals are time variant. The downside of ARCH models, however, it is that these models typically require at least 5-8 lags of the squared shock to model the conditional volatility in a proper way. To tackle this issue, it has been introduced by Bollerslev (1986) the new family of GARCH models. The “G” in GARCH stands for “Generalized”, which means that we are now considering also the lagged conditional volatility, and this improves the ARCH(P) original specification because a low order GARCH(P,Q) model fits as well as a high order ARCH(P). The GARCH(P,Q) process is defined as:

$$r_t = u_t + \varepsilon_t$$

$$\sigma_t^2 = \omega + \sum_{p=1}^{P} \alpha_p \, \varepsilon_{t-p}^2 + \sum_{q=1}^{Q} \beta_q \, \sigma_{t-q}^2$$

$$\varepsilon_t = \sigma_t \cdot z_t$$

$$z_t \sim^{iid} \mathrm{N}(0,1)$$

with the constraints: $\omega > 0, \alpha_i, \beta_i \geq 0$ (to ensure that the conditional variances are uniformly positive) and $\alpha(1) + \beta(1) < 1$ (the necessary and sufficient condition for covariance stationarity). Moreover, $\alpha + \beta$ measure the persistence of the process: if $\alpha + \beta \approx 0$, we have a fast reversion to the unconditional mean; if $\alpha + \beta \approx 0.99$, we have a slow reversion to the unconditional mean, and if $\alpha + \beta = 1$, we have that the conditional variance k steps in the future is $\sigma_{T+K|T}^2 = \sigma_{T+1}^2$. Finally, a GARCH(P,Q) model has an equivalent ARMA(max(P,Q),Q) representation:

$$\varepsilon_t^2 \ = \ \omega \ + \sum_{i=1}^{max(P,Q)} (\alpha_i + \beta_i) \, \varepsilon_{t-i}^2 - \sum_{q=1}^{Q} \beta_i v_{t-q} \ + \ v_t$$

However, rather than focusing on GARCH(P,Q), it is better to focus on the GARCH(1,1), which is the most popular of the GARCH family, since it is simple but at the same time very powerful. In fact, it is often used as a benchmark, because only few other configurations can beat it, as showed in the work of Hansen and Lunde (2005). In the GARCH(1,1) model the conditional mean is assumed to be zero[10] and the process is defined as:

$$r_t = \varepsilon_t$$

$$\sigma_t^2 = \omega + \ \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

$$\varepsilon_t \ = \ \sigma_t z_t$$

$$z_t \sim^{iid} N(0,1)$$

with the constraints $\alpha + \beta \ < 1$ and $\omega > 0, \alpha_1 \geq 0 \ and \ \beta_1 \geq 0$.

To compute the unconditional variance of r_t, $Var(r_t) \ = E(\varepsilon_t^2) = \ E(\sigma_t^2)$, it is enough to take the expectation on both sides to get:

$$E(\sigma_t^2) \ = \ \omega + \ \alpha_1 E(\varepsilon_{t-1}^2) + \ \beta_1 E(\sigma_{t-1}^2)$$

and since by stationarity we have that $E(\sigma_{t-1}^2) = \ E(\varepsilon_{t-1}^2) = \ E(\sigma_t^2)$, then we get that:

$$Var(r_t) \ = \ \frac{\omega}{1 - \alpha_1 - \beta_1}$$

[10] It is reasonable to assume that, for many risky assets, the unconditional long run mean is zero.

The above unconditional variance can be generalized for a GARCH(P,Q) model as:

$$Var(r_t) = \frac{\omega}{1 - \sum_{p=1}^{P} \alpha_p - \sum_{q=1}^{Q} \beta_q}$$

Finally, I would like also to present a well know extension to the ARCH and GARCH family of models, which is the GJR-GARCH. Introduced in 1993 by Glosten, Jagannathan and Runkle, it adds asymmetric term to the standard GARCH(P,Q) model. The aim of this additional asymmetric term is to capture a well-known phenomenon in the conditional variance of stocks, the so called "leverage effect". The leverage effect refers to the tendency of the volatility to rise more in response to a large negative shock rather than to a large positive shock, which causes an asymmetric distribution.

A GJR-GARCH(P,O,Q) process is defined as:

$$r_t = \mu_t + \varepsilon_t$$
$$\sigma_t^2 = \omega + \sum_{p=1}^{P} \alpha_p\, \varepsilon_{t-p}^2 + \sum_{o=1}^{O} \gamma_o\, \varepsilon_{t-o}^2 I_{|\varepsilon_{t-o}<0|} + \sum_{q=1}^{Q} \beta_q\, \sigma_{t-q}^2$$
$$\varepsilon_t = \sigma_t \cdot z_t$$
$$z_t \sim^{iid} \mathrm{N}(0,1)$$

where $I_{|\varepsilon_{t-o}<0|}$ is a dummy variable which takes the values of 1 if $\varepsilon_{t-o} < 0$ and 0 otherwise.
As before, rather than focusing on GJR-GARCH(P,O,Q) is better to focus on GJR-GARCH(1,1,1), whose process is defined as:

$$\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \gamma_1\varepsilon_{t-1}^2 I_{|\varepsilon_{t-1}<0|} + \beta_1\sigma_{t-1}^2$$

with the constraints that $\omega > 0, \alpha_1 \geq 0, \beta_1 \geq 0, \gamma_1 \geq 0\ and\ \alpha_1 + \gamma_1 \geq 0$. Let's decompose the above expression for the GFJ-GARCH(1,1,1):

$$\sigma_t^2 = \omega + \alpha_1^*\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2 \text{ if } \varepsilon_{t-1} \geq 0$$
$$\sigma_t^2 = \omega + (\alpha_1 + \gamma_1)\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2 \text{ if } \varepsilon_{t-1} < 0$$

When shocks are positive then the effect of a past good news on the volatility will be loaded by a coefficient α_1^*, however, when the shock is negative the effect of a past bad news will be loaded by $(\alpha_1 + \gamma_1)$. The condition for positivity imposes that $\gamma_1 \geq 0$. However, if $\gamma_1 = 0$ then there is no asymmetric effect and we are back in GARCH(1,1) specification; if $\gamma_1 > 0$, then past bad news loads on the volatility more than the bad one. This means that, if γ_1 represents the amount of asymmetry in the distribution, the unconditional variance of r_t can be computed as:

$$Var(r_t) = \frac{\omega}{1 - \alpha_1 - \frac{\gamma_1}{2} - \beta_1}$$

with $\alpha_1 + \frac{\gamma_1}{2} + \beta_1 < 1$.

2.2 Applying GARCH

Using the BTC daily log-returns, the goal is to predict the future realized volatility 1-day ahead. The raw dataset provides BTC daily prices and it contains 1370 trades from 01/01/2017 to 02/10/2020; the data has been gathered from Yahoo-Finance. I am going to split our forecasting problem into 3 main steps:

1. *Initial Analysis:* using the BTC daily prices, I will provide an initial analysis to better understand the data, in particular, summary statistics and visual representations. This initial analysis will help us not only to understand better the underlying data but also to get an insight about anomalies like missing data, the presence of outliners, the trends seasonality, etc.
2. *Choosing and Fitting the model:* first, I will split the dataset, according to the 70/30 rule, into a train and a test dataset. Then, I will try to understand which is the best configuration for our GARCH model. Once found the best configuration for the GARCH model, I will fit the GARCH model using such configuration.
3. *Forecasting and Evaluating the Forecasting model:* in this step, I will use the GARCH model configured in the previous step to forecast the realized volatility 1-day ahead, through the rolling forecasting window approach. I will also present the GJR-GARCH model with the same configuration. I will use BTC 1-min prices, from 01/01/2017 to 02/10/2020, to compute daily realized volatility. Finally, I will review the residuals and use the mean squared error to assess the performance of the models.

2.2.1 Initial Analysis

I will start by looking at the summary statistics of the whole series, which is a very useful exercise to get an insight about the spread and the distribution of our values. Table 2.1 provides the summary statistics for the BTC daily prices.

Table 2.1 – *Summary Statistics*

count	1371.00
mean	6906.84
std	3436.97
min	777.76
25%	4024.60
50%	7176.41
75%	9315.12
max	19497.40

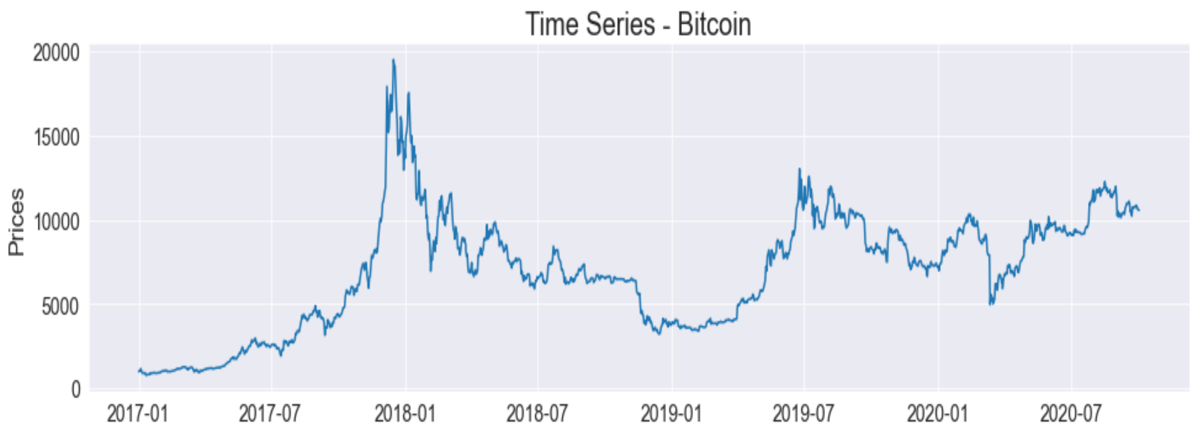
Source: Created with Python

As we can see, the mean of our series is equal to $6906.84, which can be considered the level of our series; the standard deviation is relatively large at $3436.97 and, along with the percentiles, suggest a large spread in the data. A large spread in the data can reduce drastically the accurateness of our predictions, especially if it is not a systematic one.

Now, I am going to visualize the time series by using different plotting approaches.

1. *Line Plot:*

Figure 2.1 – *Line Plot-BTC Time Series*

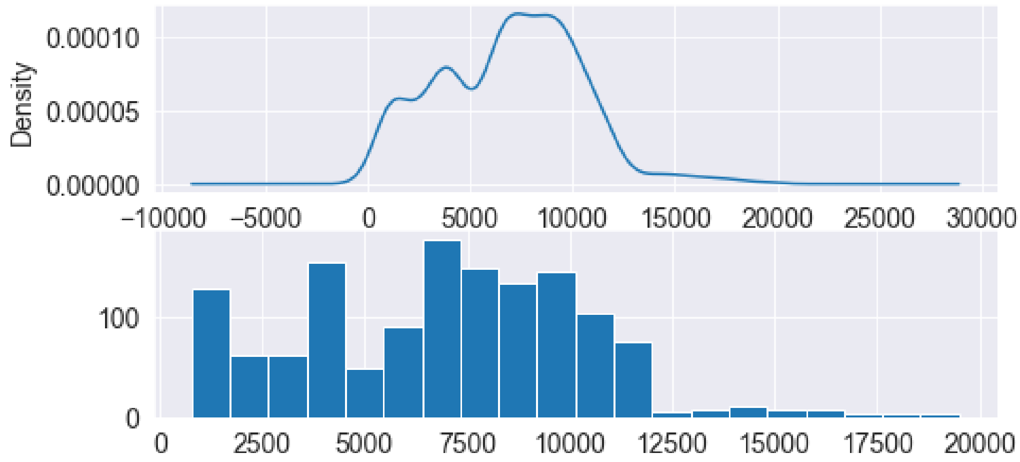


Source: Created with Python

A line plot representation of our time series can provide useful information. By looking at the line plot of our time series, we can clearly identify large fluctuations from year to year, with a possible outliner at the beginning of 2018. It doesn’t seem to be any clear pattern in the prices; the presence of a possible outliner may suggest that the distribution of returns is asymmetric, perhaps with large kurtosis and a skewness different from zero.

2. *Density Plot:*

Figure 2.2 – *Density Plot- BTC-Time Series*



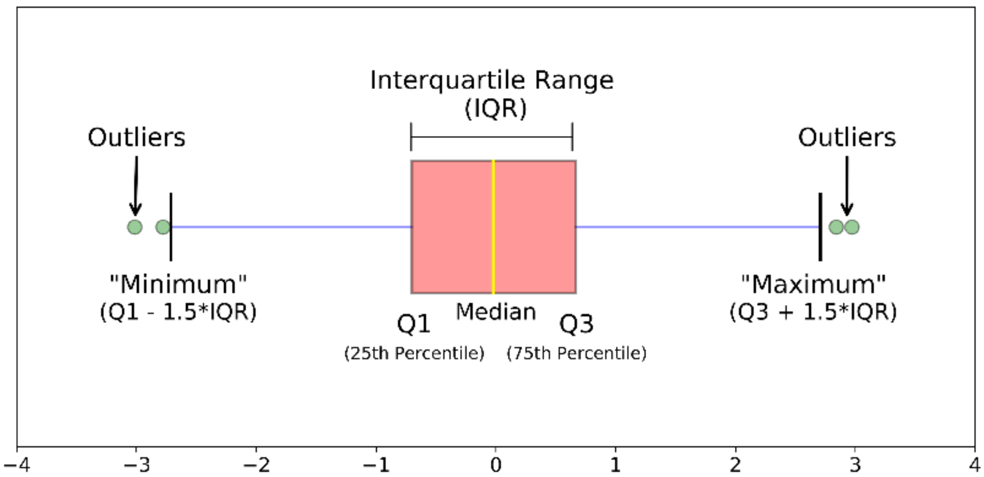
Source: Created with Python

As we can see from the density plot, the distribution is clearly not Gaussian. It is right shifted, which means a skewness greater than 0 (0.16), and it has a kurtosis of -0.02. A line plot suggested a kurtosis greater than zero, however, it does not seem to be the case. Moreover, the second plot suggests that the distribution may be exponential.

3. *Box and Whisker Plots:*

A Box and Whisker plot is a very useful plot that can provide us major insights about the distribution of values in different time intervals. The Figure 2.3 represents an example of a box and whisker plot decomposed in its components.

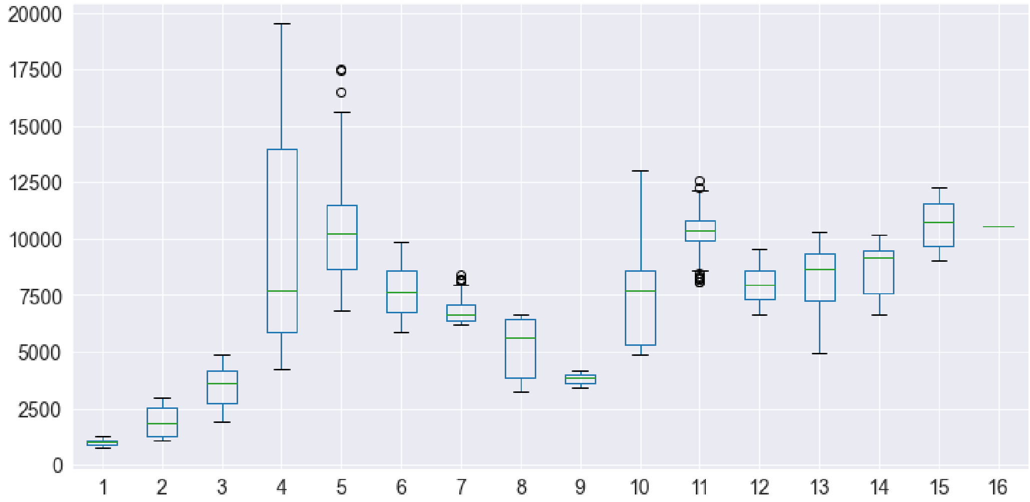
Figure 2.3 – *Box and Whisker Components*



Source: https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

This plot draws a box that captures the 50% of the data around the 25th percentile and the 75th percentile. The yellow line represents the median or the 50th percentile, while the whiskers represents the extents of the observations. In this representation, dots represent the outliners. Before I draw the Box and Whisker plots, I group the Bitcoin daily prices into quarters; by doing this, I want to see the distribution of the values in different quarters, spot possible outliners and look for possible year trends or seasonality.

Figure 2.4 – *Box and Whisker Plot-BTC Time Series*



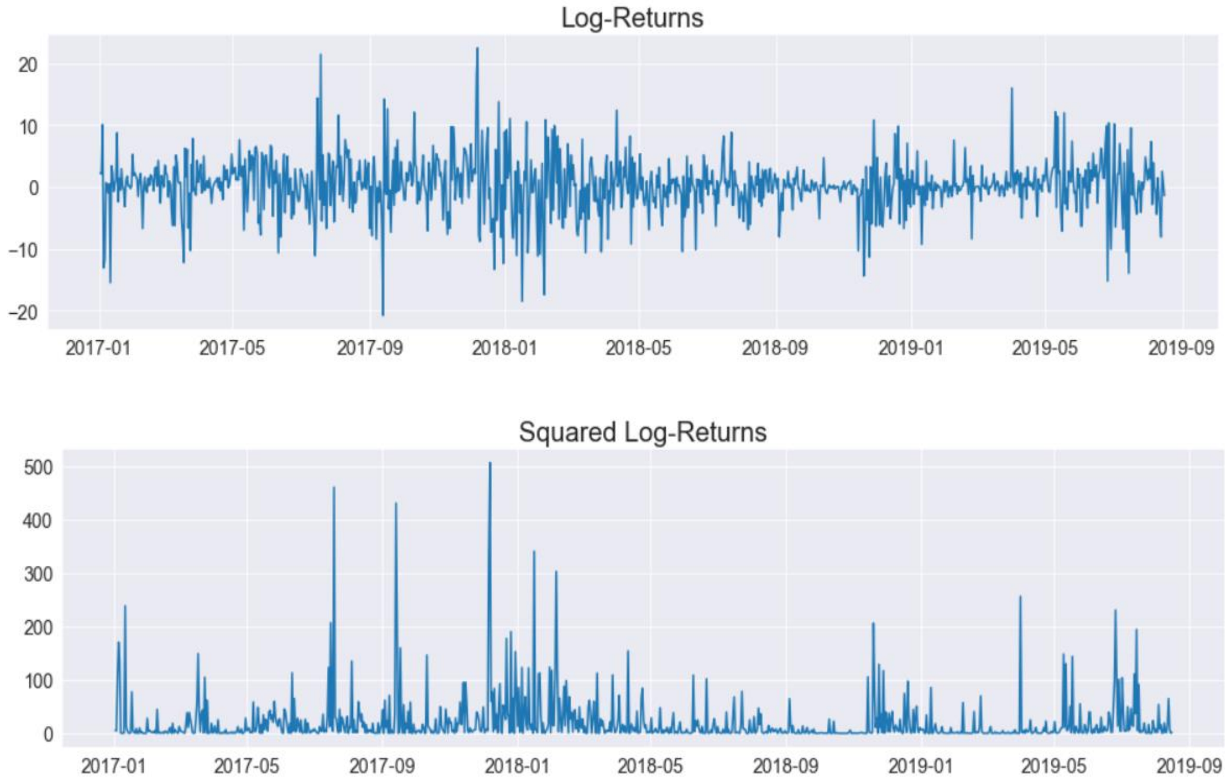
Source: Created with Python

By looking at Figure 2.4, it does not seem to be any clearly trend across quarters. However, the box which contains the 50% of the data, change over time. Finally, we can clearly identify the presence of outliners in the fifth, seventh and eleventh quarter.

2.2.2 Choosing and Fitting the model

After computing the BTC daily-log returns, I split the dataset into a train and a test set according to the 70/30 rule. The train dataset contains 959 trades and it goes from 02/01/2017 to 17/08/2019, while the dataset for the validation part contains 411 trades and goes from 18/08/2019 to 02/10/2020. The whole analysis will be applied on the train dataset. Figure 2.5 represents the daily log returns and the daily squared log returns.

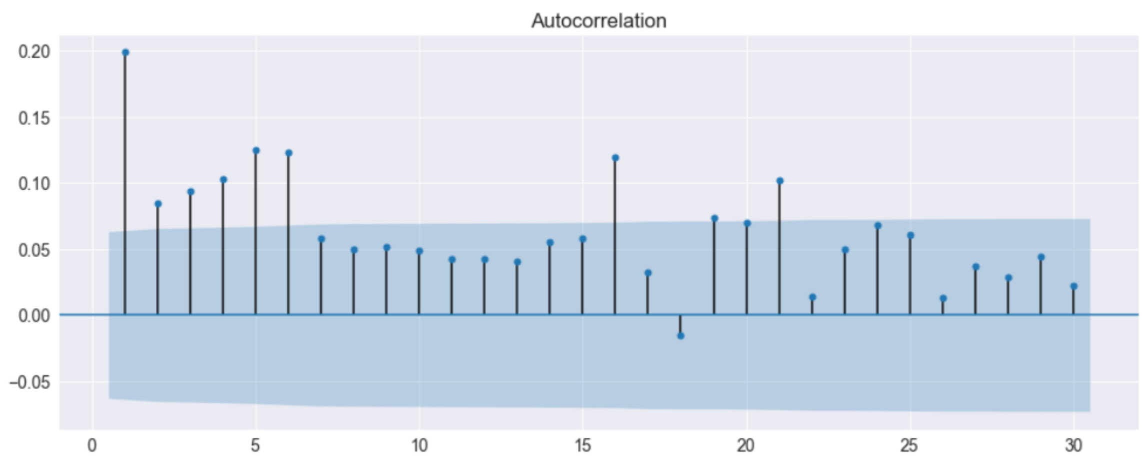
Figure 2.5 – *BTC Log-Returns Plot*

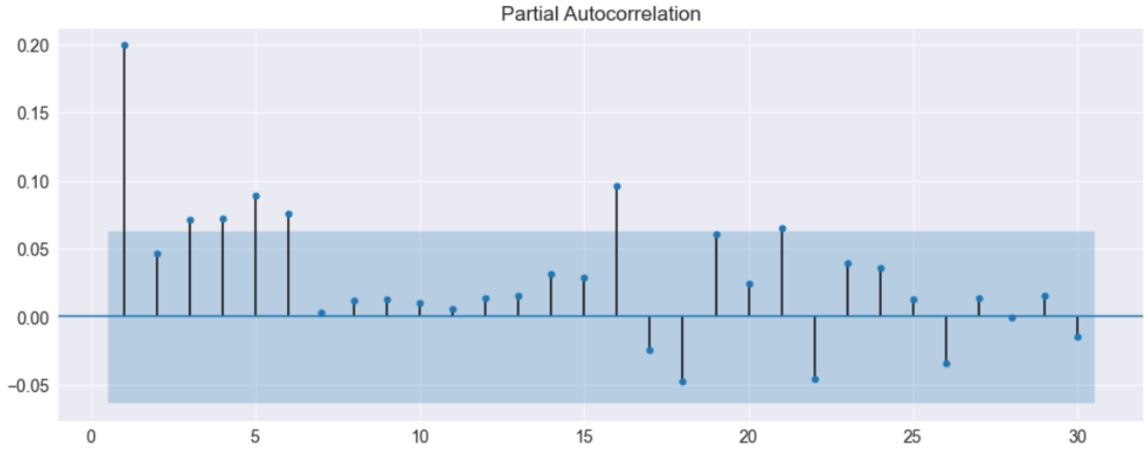


Source: Created with Python

The plot of returns is a graphical evidence of time-varying volatility, which means that this is a good candidate for being modelled by GARCH family of models. Now we will see which is the best configuration for our GARCH model. We can start by looking at the autocorrelation function (ACF) and at the partial autocorrelation function (PACF) of a squared series of returns to get an insight about which P and Q to use in our GARCH(P,Q) model.

Figure 2.6 – *ACF and PACF*





Source: Created with Python

The ACF shows us a significant positive autocorrelation in the variance up to the twentieth lag; however, the autocorrelations beyond the second lag may be due to the propagation of the autocorrelation at lag 1. The PACF shows us that maybe we could try with P=1 and Q=1. However, to be precise in my choice and to avoid the "doing by trial", I rely on the Bayesian Information Criterion (BIC) for choosing the best P and Q in our (P,Q) model. The BIC can be represented as:

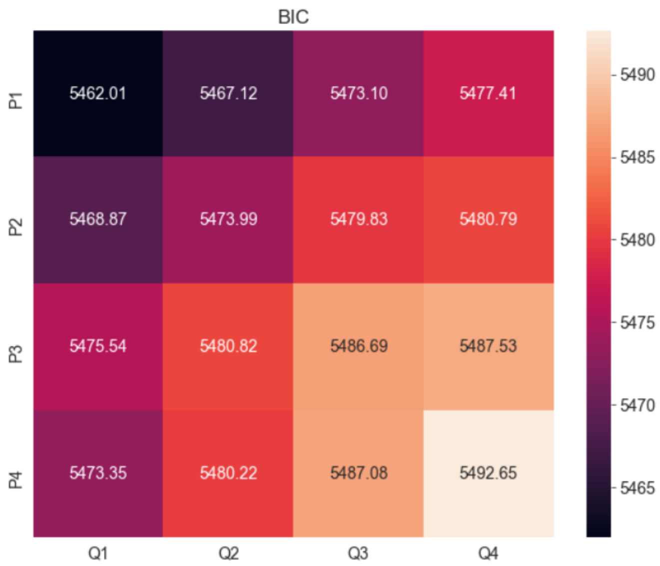
$$BIC = k\ln(n) - 2\ln(\hat{L})$$

where k is the number of parameters estimated by the model, n is the number of observations and $\hat{L}$ the maximized value of the loglikelihood function of the model[11]. The best model is the one associated with the lowest value of BIC.

So, in order to choose the right model, we have to look at the BIC of different GARCH models and, to do so, we perform a for loop in which we look at different combination of P and Q, where both P and Q cannot exceed 4 (to keep the model simple). The results are displayed in the Figure 2.7.

[11] https://en.wikipedia.org/wiki/Bayesian_information_criterion

Figure 2.7 – *BIC scores*



Source: Created with Python

The results suggest that the best configuration we can choose is the GARCH(1,1) model. This result is something that we could expect, because it is very difficult to beat the GARCH(1,1) model. Finally, I fit the GARCH(1,1) model to the data; the results are displayed in Table 2.2.

Table 2.2 – *GARCH(1,1) fitting results*

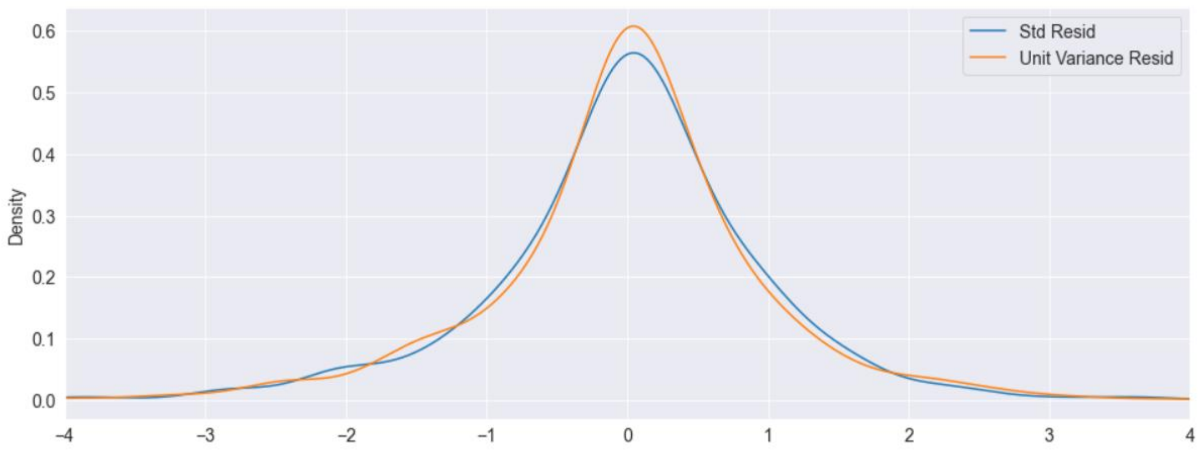
```
                     Constant Mean - GARCH Model Results
==============================================================================
Dep. Variable:                Adj Close   R-squared:                      -0.000
Mean Model:               Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                        GARCH   Log-Likelihood:               -2717.27
Distribution:                    Normal   AIC:                           5442.55
Method:              Maximum Likelihood   BIC:                           5462.01
                                          No. Observations:                  958
Date:                  Sun, Feb 07 2021   Df Residuals:                      954
Time:                          00:48:15   Df Model:                            4
                                 Mean Model
===========================================================================
                 coef    std err          t      P>|t|      95.0% Conf. Int.
---------------------------------------------------------------------------
mu             0.2270      0.122      1.868  6.179e-02 [-1.120e-02,  0.465]
                              Volatility Model
==========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
--------------------------------------------------------------------------
omega          1.1413      0.533      2.141  3.224e-02 [9.669e-02,  2.186]
alpha[1]       0.1230  3.241e-02      3.794  1.481e-04 [5.946e-02,  0.187]
beta[1]        0.8214  4.246e-02     19.346  2.212e-83   [  0.738,  0.905]
==========================================================================

Covariance estimator: robust
```

Source: Created with Python

From the table above, we can see that all the coefficients are statistically significant at the 1% level and the series is stationary, since $\alpha_1 + \beta_1 < 1$ (0.944). The condition for covariance positivity is also satisfied because all the coefficients are greater than zero. Now we can look also at standardized residuals, comparing them with non-standardized one; results are plotted in the following figure.

Figure 2.9 – *Residuals Comparison*



Source: Created with Python

Figure 2.9 shows us that the non-standardized residuals are more peaked, indicating that the distributions is heavier tailed than that of the standardized residuals. For this reason, we assume a different distribution for our standardized residuals, the t-distribution. The t-distribution allows us to account for heavier tails. Table 2.3 shows the results.

Table 2.3 – *GARCH(1,1) t-distribution*

```
                        Constant Mean - GARCH Model Results
====================================================================================
Dep. Variable:                    Adj Close   R-squared:                      -0.000
Mean Model:                   Constant Mean   Adj. R-squared:                 -0.000
Vol Model:                            GARCH   Log-Likelihood:               -2631.25
Distribution:      Standardized Student's t   AIC:                           5272.50
Method:                  Maximum Likelihood   BIC:                           5296.83
                                              No. Observations:                  958
Date:                      Fri, Feb 12 2021   Df Residuals:                      953
Time:                              11:54:36   Df Model:                            5
                                Mean Model
==========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
--------------------------------------------------------------------------
mu             0.2903      0.101      2.877  4.017e-03 [9.251e-02,  0.488]
                              Volatility Model
==========================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
--------------------------------------------------------------------------
omega          0.2191      0.405      0.541      0.588   [ -0.574,  1.012]
alpha[1]       0.1269  3.690e-02      3.440  5.817e-04 [5.462e-02,  0.199]
beta[1]        0.8731  5.990e-02     14.575  4.028e-48   [  0.756,  0.990]
                               Distribution
========================================================================
                 coef    std err          t      P>|t|  95.0% Conf. Int.
------------------------------------------------------------------------
nu             3.5622      0.344     10.360  3.784e-25 [  2.888,  4.236]
========================================================================

Covariance estimator: robust
```

Source: Created with Python

From these results we can see that the model better fits the data (higher log-likelihood); however, the condition for stationarity is not satisfied, since $\alpha_1 + \beta_1 = 1$, and we can also see that the constant is not significant. Next, I implement the GJR-GARCH(1,1,1) model and represent the results in Table 2.4.

Table 2.4 – *GJR-GARCH(1,1,1)*

```
                     Constant Mean - GJR-GARCH Model Results                      
==================================================================================
Dep. Variable:                Adj Close   R-squared:                        -0.000
Mean Model:               Constant Mean   Adj. R-squared:                   -0.000
Vol Model:                    GJR-GARCH   Log-Likelihood:                 -2717.16
Distribution:                    Normal   AIC:                             5444.31
Method:          Maximum Likelihood   BIC:                             5468.64
                                          No. Observations:                    958
Date:                  Fri, Feb 12 2021   Df Residuals:                        953
Time:                          15:08:15   Df Model:                              5
                                 Mean Model                                 
============================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
----------------------------------------------------------------------------
mu             0.2181      0.120      1.816  6.944e-02 [-1.735e-02,  0.454]
                              Volatility Model                              
============================================================================
                 coef    std err          t      P>|t|     95.0% Conf. Int.
----------------------------------------------------------------------------
omega          1.1708      0.580      2.019  4.344e-02  [3.447e-02,  2.307]
alpha[1]       0.1161  3.665e-02      3.169  1.529e-03  [4.431e-02,  0.188]
gamma[1]       0.0151  5.055e-02      0.298      0.766 [-8.401e-02,  0.114]
beta[1]        0.8188  4.621e-02     17.719  3.007e-70    [  0.728,  0.909]
============================================================================
```

Source: Created with Python

As we can see from the results, the stationarity condition, $\alpha_1 + \frac{\gamma_1}{2} + \beta_1 < 1$, is satisfied (0.942). The covariance positivity condition is also satisfied; however, γ_1 is not statistically significant at any significance level. This suggest that the "leverage effect" is not present in Bitcoins.

2.2.3 Forecasting and Evaluating the Forecasting model

Finally, we can make our prediction about the future volatility. The GARCH(1,1) exploits the relationship $E_t[\varepsilon_{t+1}^2] = \sigma_{t+1}^2$ to make forecasts, and we have that:

$$\begin{aligned} \sigma_{t+1}^2 &= \omega + \alpha\varepsilon_t^2 + \beta\sigma_{t-1}^2 \\ \sigma_{t+k}^2 &= \omega + \alpha E_t[\varepsilon_{t+k-1}^2] + \beta E_t[\sigma_{t+k-1}^2]\ for\ k \geq 2 \end{aligned}$$

I will produce the 1-day ahead forecast using the rolling forecasting window approach. With the rolling forecasting window approach, we keep the window length fixed and then produce a one-step ahead forecast from the final observation. Moreover, I will use 1-min BTC prices[12] to compute daily realized variance as:

$$r_t = ln\left(\frac{p_t}{p_{t-1}}\right)$$

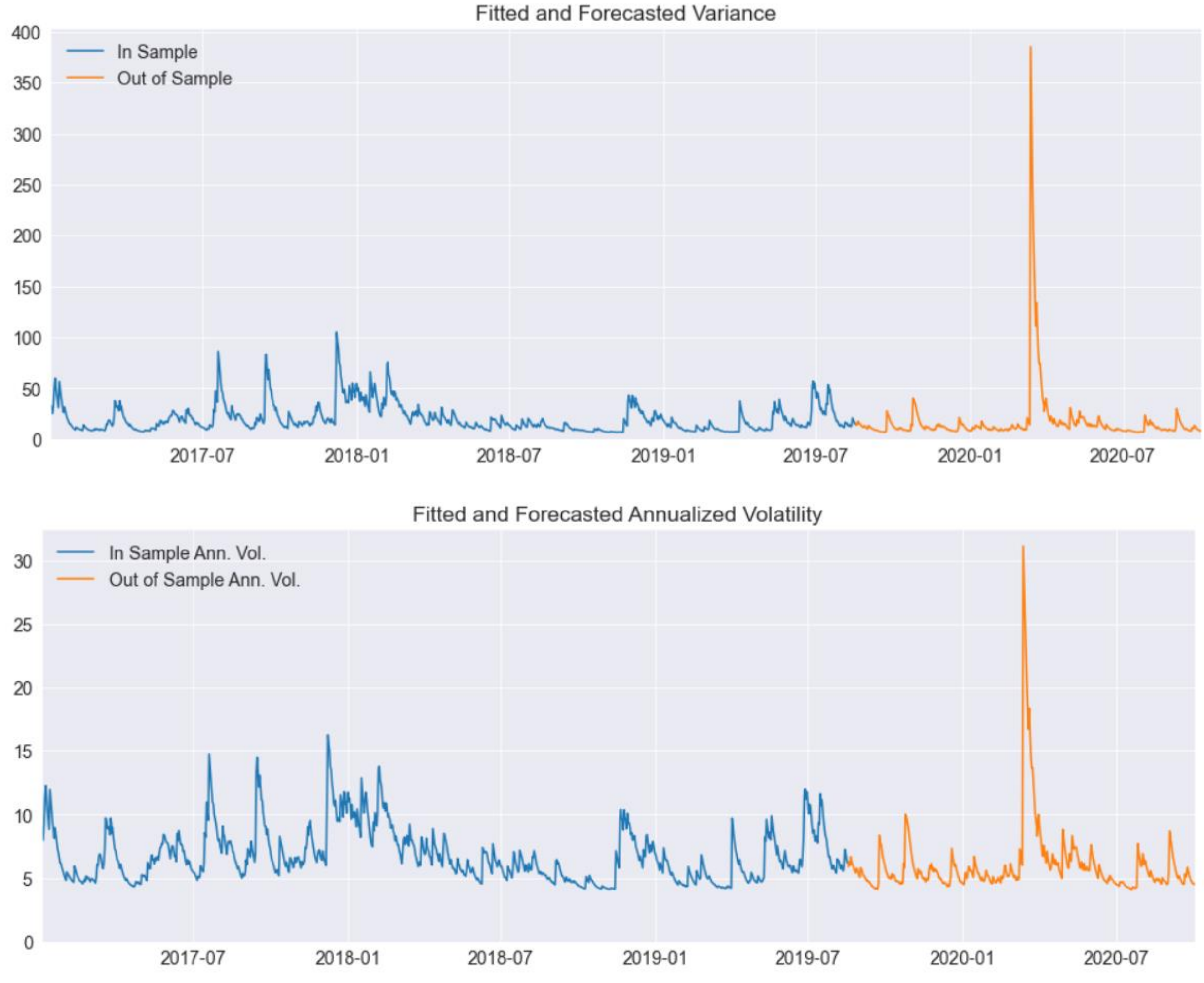
[12] See: https://www.cryptodatadownload.com/data/gemini/

$$RV_t^{AC1(1440)} = \sqrt{\sum_{i=1}^{1440} r_{i,t}^2 + 2 * \sum_{i=1}^{1440} r_{i,t} r_{i-1,t}}$$

where r_t is the measure of BTC 1-min log returns and $RV_t^{AC1(1440)}$ is the measure of BTC daily realized volatility adjusted for bid-ask bounce[13]. In fact, the high frequency data are contaminated by noise and, if we don't account for it, the data will be upward biased. To remove the bias, we rely on $RV_t^{AC1(1440)}$.

Finally, I perform the forecast and plot the results, showed in the following figure.

Figure 2.10 – *Fitted and Forecasted measures*

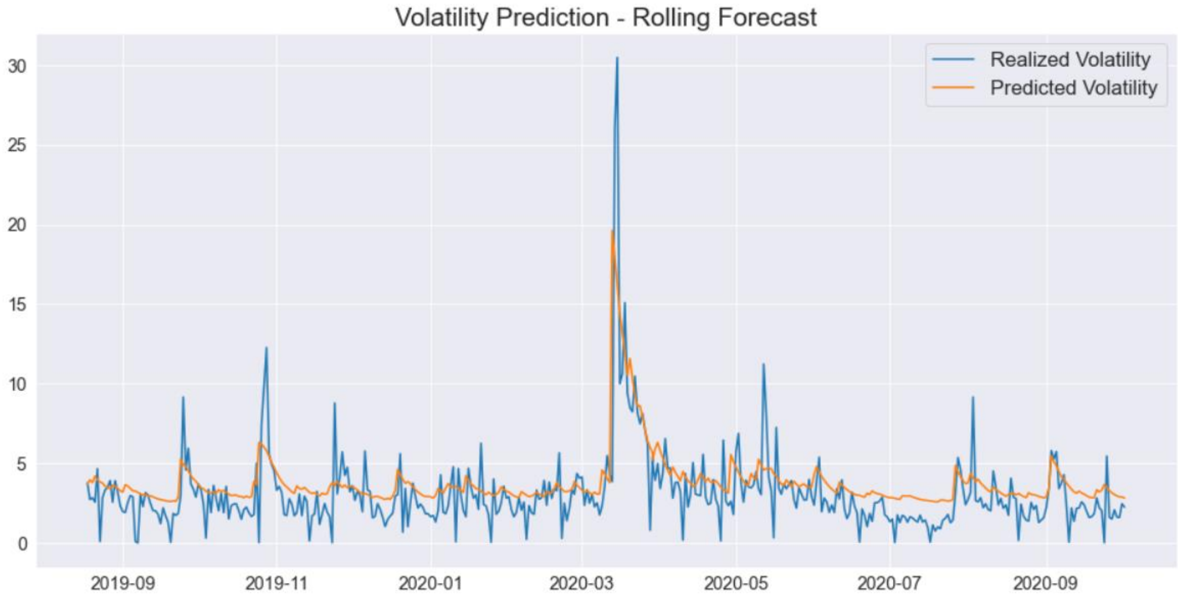


Source: Created with Python

Then, I plot the predicted volatility against the realized volatility, the Figure X shows the results.

[13] Bid-ask bounce refers to the fact that with high-frequency data we don't observe a single price and the prices are observed only at the bid and the ask, which makes the consecutive prices to oscillate between the two.

Figure 2.11 – *RV vs GARCH(1,1) PV*



Source: Created with Python

As we can see, the realized volatility predicted by the GARCH(1,1) model seems to be a good approximation of the true realized volatility. However, to understand better the accuracy of the forecasting model, I am going to check first the mean squared error, and then the distribution of residuals. The mean squared error is computed as follows:

$$MSE = \sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}$$

The smaller is the value of the mean squared error, the better is our model; for instance, a model with a mean squared error of zero indicates a perfect model with no error. The MSE of our model is equal to 3.705, which means that our model can be improved. However, we must consider another factor that may affect the high level of our MSE: the daily realized variance has been estimated by using intraday prices from Gemini Exchange and the low trading volume on this exchange is likely to affect our estimation.

As a final check, I would like to assess the accuracy of the forecast; for this purpose, I perform the Mincer-Zarnowitz regression of the forecast error, which is defined as $\lfloor r_{t+1} - \hat{u} \rfloor^2 - \hat{\sigma}^2_{t+1}$, on a constant and the forecasted variance. The regression is used to test the key property of a well specified model: $E_t \lfloor {r_{t+1}}^2 - \hat{\sigma}^2_{t+1} \rfloor = 0$; namely we test that $H_0: \alpha = \beta = 0$. The results are displayed in the Table 2.5.

Table 2.5 – *Mincer-Zarnowitz regression results*

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.052
Model:                            OLS   Adj. R-squared:                  0.050
Method:                 Least Squares   F-statistic:                     112.8
Date:                Fri, 12 Feb 2021   Prob (F-statistic):           1.96e-23
Time:                        23:11:10   Log-Likelihood:                -2513.6
No. Observations:                 411   AIC:                             5031.
Df Residuals:                     409   BIC:                             5039.
Df Model:                           1
Covariance Type:                  HC0
=================================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const          11.4425      5.734      1.996      0.046       0.205      22.680
roll_variance  -0.7938      0.075    -10.621      0.000      -0.940      -0.647
==============================================================================
Omnibus:                      956.348   Durbin-Watson:                   1.984
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          2346548.449
Skew:                          18.832   Prob(JB):                         0.00
Kurtosis:                     371.247   Cond. No.                         42.0
==============================================================================
```

Source: *Created with Python*

The Wald test returns the t-statistic of 116.42, which means that the results are statistically significant and we cannot accept the alternative hypothesis.

Chapter 3: VIX Estimation

3.1. Literature overview

In Chapter 2, we looked at volatility in its two shapes, namely the conditional volatility and the realized volatility. Here I will treat the third shape of volatility: implied volatility. The implied volatility is the volatility which solves the Black-Scholes equation. The Black-Scholes formula is derived by assuming that stock prices follow a geometric Brownian motion with drift:

$$dS_t = uS_t dt + \sigma S_t dW_t$$

where S_t is the stock price at time t, u is the drift, σ is the constant volatility and dW_t is a Wiener process. Under assumptions of no arbitrage, the formula of a call option price can be proved to be:

$$C_t(T,K) = S_t\phi(d_1) + Ke^{-rT}\phi(d_2)$$

$$d_1 = \frac{ln(S_t/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{ln(S_t/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}$$

where T is the time to maturity (measured in years), r is the risk-free rate, K is the strike price and $\phi(\cdot)$ is the normal CDF. The implied volatility is the expected volatility between t and T under the risk-neutral measure (which is the same as under the physical one, when volatility is constant), namely:

$$\sigma_t^{implied} = g(C_t(T,K), S_t, K, T, r)$$

However, Black-Scholes presents different issues, in particular, among others:

1. It is derived under constant volatility, which is an unrealistic assumption, since returns on many financial instruments exhibit conditional heteroskedasticity and this generates heavy tails.
2. It does not account for the leverage effect, or negative correlation between the price and the volatility of a financial instrument, which is very important, since it increases the likelihood of extreme negative returns relative to the log-normal price process assumed in the Black-Scholes option pricing formula.

As an alternative to the Black-Scholes implied volatility, to tackle the above described issues, the model-free implied volatility has been developed. This new measure is not only a forward-looking measure, but also a market based one. The VIX is an example of this alternative measure of volatility.

The volatility index (VIX), also known as a "Fear Index", is a financial instrument which measures, under risk neutrality, the market expectations of one month ahead volatility on the S&P500 index. It is constructed using monthly and weekly options on S&P500; the series of VIX is an annualized and percentage measure of standard deviation. Moreover, VIX is an implied volatility measure used not only as an indication of the market stress or fear, but also as a forecast of a future realized volatility. But why do we define VIX as a fear gauger? It is because stock market returns and volatility are negatively correlated, which means that VIX is a financial instrument that provides diversification benefits once included in an investment portfolio.

The volatility index was introduced for the first time in 1993 by the Chicago Board of options Exchange (CBOE). The initial version of the VIX was based on options data on the S&P100 index (OEX), where the implied volatility used for the construction of the index was the average of the Black and Scholes option implied volatility, with the strike prices closed to the current index level and maturities interpolated at about one month. However, the old version of VIX has been highly criticized due to the so-called "trading-day conversion", whose introduction lead to an upward bias. In fact, since the Black and Scholes implied volatility is already an annualized measure, where the annualization is based on actual/365 days-counting conversion, the trading-day conversion leads to an upward bias in the level of the VIX and makes it no longer comparable to annualized realized volatilities computed from stock index returns.

In 2003, the CBOE introduced a new way to measure the implied volatility used in the VIX construction. In the new version of the VIX, OEX is replaced by S&P500 as underlying stock index and the implied volatility is measured as a weighted average of option prices of S&P500 puts and calls over a wide range of strike prices at two nearby maturities (near term and next term). In more concrete terms, the VIX is constructed using S&P500 options with more than 23 days and less than 37 days to expiration. Moreover, the implied volatility is no longer upward biased due to the trading-day conversion. In fact, the VIX represents an annualized volatility using the actual/365 days-counting conversion. Since the VIX is no longer upward biased, it can be used to compare to annualized realized volatilities computed from stock index returns. In particular, the VIX squared approximates the conditional risk-neutral expectation of the annualized return variance from time t to 30 calendar days later ($RV_{t,t+30}$):

$$VIX_t^2 \cong E_t^Q\,[RV_{t,t+30}]$$

3.2 VIX Index Calculation

In this section, I will compute the VIX index using options on Bitcoins. The reason for using options in the implied volatility computation is that the price of each option reflects the market's expectation of future volatility.

The options used for the computation of the BTC-VIX Index have been obtained from Deribit Exchange. Deribit is an exchange founded in 2016, based in Netherland, which is not only the first in the world offering options, but also the largest one in terms of daily trading volume. Deribit offers European style cash-settled options. "European style" means that they are options that can be exercised only at expiration, while "cash settled" means that, at expiry, there will be only the exchange of cashflows, rather than the transfer of any assets. The options are priced in Bitcoins, where a call option represents the right to buy 1 Bitcoin at a specific price K (strike price) and at a given maturity T. Deribit exchange presents different interesting features; among others, we have the following ones:

1. It operates continuously, 24/7.
2. Fiat currencies are not accepted and only Bitcoins or Ethereum can be used, and this provide high degree of anonymity.
3. Options expire every Friday at 8:00 UTC, with the settlement value calculated by using the average of the Deribit Bitcoin Index over the last 30 minutes before the expiry. Deribit Bitcoin Index is an index computed by using the price of Bitcoins from eight Bitcoin Exchanges, including Bitfinex, Bitstamp, Bittrex, Coinbase, Gemini, Itbit, Kraken, LMAX Digital.[14]
4. The minimum tick size is set to 0.0005 Bitcoins, minimum options order size is set to 0.1 option contract and the strike price interval is not fixed but depends on the Bitcoin Price (it can vary between 250 USD and 5000 USD).

The data I was able to retrieve from Deribit exchange do not provide bid and ask quotes, but they give the "mark price". As defined by the Deribit Exchange, the mark price of an option contract is the value spot of the option calculated as the average of the best bid and the best ask price. The raw dataset contains 1.416.460 trades. In the first cleaning step, I exclude all trades with missing mark price values and split the dataset in two groups, one containing all the call options trades and the other one containing the put options trades. Then I group each dataset according to the strike price and the expiration date. Finally, I merge the two datasets together. Now the dataset is composed by 370 different DataFrame objects, sorted according to strike price and expiration date, where the mark price is computed as the simple mean of all trades with that specific expiration

[14] https://www.deribit.com/prinx_chart

date, strike price and type of the option. The first DataFrame, or the first option chain, is represented in Table 3.1.

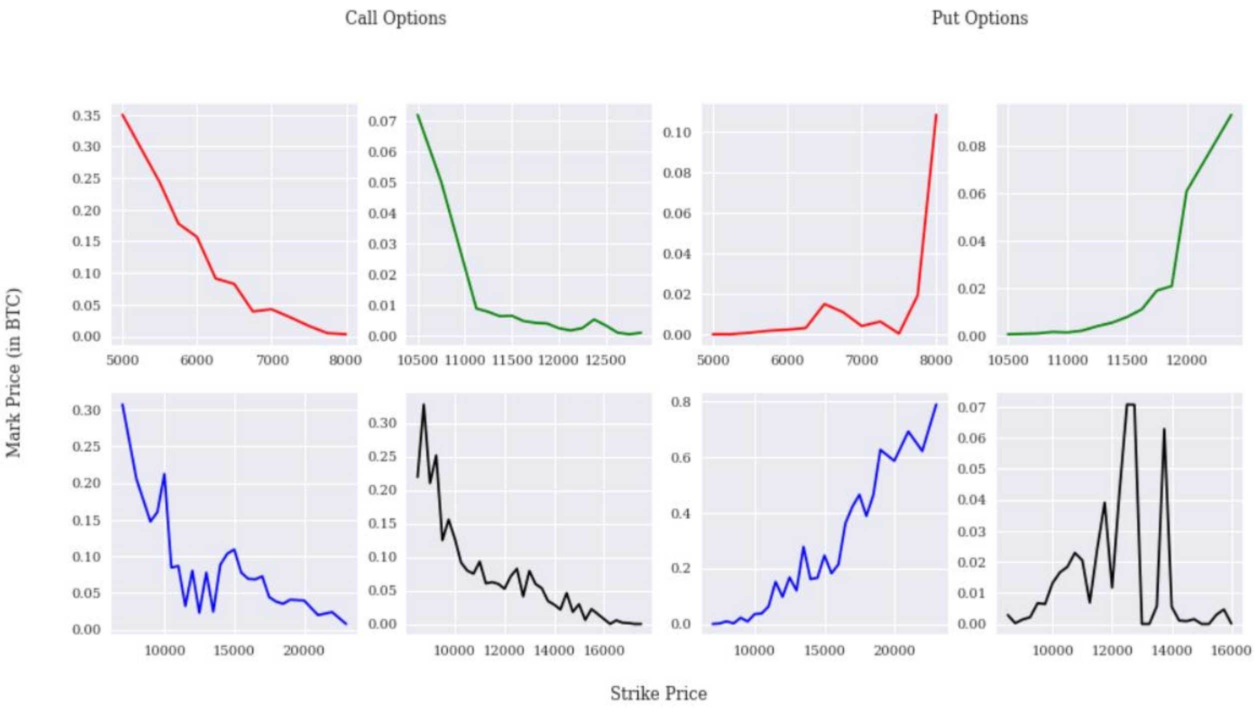
Table 3.1 – *1/370 Option Chain*

strike	call_mid	put_mid	expiration_date
5000	0.349921	0.000004	2017-11-17 08:00:00
5500	0.243198	0.000736	2017-11-17 08:00:00
5750	0.178023	0.001708	2017-11-17 08:00:00
6000	0.156843	0.002225	2017-11-17 08:00:00
6250	0.091295	0.003087	2017-11-17 08:00:00
6500	0.082608	0.014952	2017-11-17 08:00:00
6750	0.039251	0.010860	2017-11-17 08:00:00
7000	0.042716	0.004064	2017-11-17 08:00:00
7250	0.029983	0.006251	2017-11-17 08:00:00
7500	0.016419	0.000308	2017-11-17 08:00:00
7750	0.004961	0.019134	2017-11-17 08:00:00
8000	0.003257	0.108428	2017-11-17 08:00:00

Source: Created with Python

Now I perform the second cleaning step, in which I check whether the no-arbitrage condition holds for BTC-options. Figure 3.1 displays the results of the randomly chosen put option and call option chains:

Figure 3.1 – *Randomly chosen option chains*



Source: Created with Python

As we can see from the above graphs, the arbitrage condition does not hold, especially if we look at the second plot line. To overcome this problem, it is possible to use the curve fitting technique. Curve fitting is an optimization method that, given a map function, finds the optimal set of parameters for that specific function that best fits the underlying data. The underlying function can be of any type, such as, among others, a straight line (in this case we are considering a linear regression) or a curved lined (polynomial regression). For our problem, a good mapping function could be a simple linear regression function, because it is the one that would make the no-arbitrage condition hold:

$$Price^{USD} = a * Strike\ Price^{USD} + b$$

However, I decided to compute the VIX index as it is, to avoid any possible bias in the estimation, but being aware of the above described issue.
So, after the initial setting, we can finally start with the computations of the VIX index. In what follows I will review each step followed to compute the VIX index and, for this purpose, I will split the VIX computation in different steps:

- *Step 1: Selecting the options to be used in the BTC-VIX index computation.*

As stated before, the original VIX index measures one month ahead volatility on the S&P500 index. The S&P500 options used in its computation are near term and next term put and call options with more than 23 days to expiration but less than 37 days to expiration. These include S&P500 options with third Friday expiration dates and weekly options that expire every Friday, except the weekly ones that expire the third Friday, in this case the former is used instead of the latter ones. Once each week, the S&P500 options used in the computation of the VIX Index are rolled to new contract maturities within the above boundaries. Only non-zero bid prices options are used in the computation and the selected options are out of the money S&P500 calls and S&P500 puts which are centered around an at the money strike price (K_0).
To find the at the money strike price (K_0), for each contract month, we need to determine first the forward S&P500 level (F). The latter is determined by identifying the strike price such that the absolute difference between the call and put option prices is the smallest one, according to the below formula:

$$F = Strike\ Price + \mathrm{e}^{RT} \cdot (Call\ Price - Put\ Price)$$

Where T is the time to expiration and R is the risk-free rate based on U.S. Treasury yield curve rates, to which a cubic spline is applied to interpolate yields on the expiration dates of the S&P500 options used in the computation. The time to expiration, T, is defined in calendar days, where each calendar day is divided into minutes in order to guarantee the higher level of precision. The time to expiration is computed according to the following expression:

$$T = \frac{\left[M_{current\ day} + \ M_{settlement\ day} + \ M_{other\ days}\right]}{Minutes\ in\ a\ year}$$

where $M_{current\ day}$ represents the minutes remaining until the midnight of the current day; $M_{settlement\ day}$ are the minutes from the midnight until the expiration of the S&P500 options (weekly or the monthly one); $M_{other\ days}$ are the total minutes in the days between current day and expiration day. Since the formula used in the determination of the forward level is applied to both near and next term options, we will find two different forward levels, F_1 (for near term options) and F_2 (for next term options), two different risk-free rates, R_1 and R_2, and two different time to expiration measures, T_1 and T_2. Once computed the forward levels, it is possible to determine the at the money strike price K_0, which is nothing else that the strike price equal to or immediately below the forward level F. In this way we obtain $K_{0,1}$ and $K_{0,2}$.

Finally, to select the out of the money options to be used in the VIX computation, we should distinguish between put and call options. For the put options, we select all the out of the money put options with strike price K<K_0, having a non-zero bid price. Once we find two subsequent put options with bid price equal to zero, then any puts with lower strike prices are considered for the inclusion in the computation of the VIX index. The same rules, but in the opposite way (K>K_0), applies to the call options. Finally, we include both the puts and calls with strike price equal to K_0. Once we have selected our put and call options, for each selected option we compute the midpoint price, which is the average between the bid and ask price. This mid-price will be then used for the computation of the VIX index. Now that we have defined the procedure to follow, we can apply it to our dataset. As stated before, we don't have bid and ask prices quotes which means that we cannot follow the standard VIX procedure. By using a mark price directly, which is always different from zero, we will include all put options with a strike price K<K_0 and all the call options with the strike price K>K_0. The first available expiration of our options is on 17/11/2017 and the last is on 02/10/2020. I set as the initial date for the computational reasons at 24/10/2017 (Tuesday), which is 24 days prior to the first expiration date available in our dataset; once per week, on Tuesday, I will roll to new contract maturities. I start by selecting the first two options whose maturities are greater than 23 days from the initial date but less 37 days. The first near term option is the one that expires on 17/11/2017 (24 days to expiration), while the next term option expires on 24/11/2017 (31 days to expiration). The option chain for the near term option is represented in Table 3.2.

Table 3.2 – *1st option chain for near term options*

strike	call_mid	put_mid	diff	expiration_date
5000	0.349921	0.000004	0.349917	2017-11-17 08:00:00
5500	0.243198	0.000736	0.242462	2017-11-17 08:00:00
5750	0.178023	0.001708	0.176315	2017-11-17 08:00:00
6000	0.156843	0.002225	0.154618	2017-11-17 08:00:00
6250	0.091295	0.003087	0.088208	2017-11-17 08:00:00
6500	0.082608	0.014952	0.067657	2017-11-17 08:00:00
6750	0.039251	0.010860	0.028391	2017-11-17 08:00:00
7000	0.042716	0.004064	0.038652	2017-11-17 08:00:00
7250	0.029983	0.006251	0.023732	2017-11-17 08:00:00
7500	0.016419	0.000308	0.016111	2017-11-17 08:00:00
7750	0.004961	0.019134	0.014172	2017-11-17 08:00:00
8000	0.003257	0.108428	0.105171	2017-11-17 08:00:00

Source: Created with Python

Once identified the near and the next term options, we can determine the forward BTC level, F, by finding the strike price with smallest absolute difference between the call and put prices.
Doing the computations, the forward level for the near term options, F_1, turns out to be equal to 7749.98. This means that $K_{0,1}$ for the near term options is equal to 7500, which is immediately below the forward index level F_1. The same procedure can be applied to the next term options, giving the result F_2 equal to 8249,98, with the $K_{0,2}$ equal to 8000.
To compute the forward level, I use the 10 year Treasury bond with weekly timeframe.

$$F_1 = 7750 + e^{(0.02354*0.06575)} * (0.004961 - 0.019134) = 7749.98$$
$$F_2 = 8250 + e^{(0.02322*0.08493)} * (0.020181 - 0.031139) = 8249.98$$

- *Step 2: Calculating volatility for near term and next term options.*

Once we have identified the at the money strike prices, $K_{0,1}$ and $K_{0,2}$, we should determine the strike price boundaries. These boundaries make sure that only calls and puts options with non-zero bid price are included. We can start considering the near-term options. For the out of the money put options, with strike price lower than $K_{0,1}$, the lower price boundary is identified by that price which is followed by two consecutive zero bid prices. Similarly, for the out of the money call options, with strike price greater than $K_{0,1}$, the upper price boundary is identified by that price which is followed by two consecutive zero bid prices. Finally, we can apply the same logic to the next-term options. After boundaries have been identified, we can start computing the contribution of each option to the VIX. This amount can be computed by using the following formula:

$$\frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i)$$

where ΔK_i is the half difference between the immediately below and immediately above strike price on either side of K_i (with K_i being either the lower strike price bound or the upper strike price bound); and $Q(K_i)$ is the midpoint of bid-ask spread at K_i. Now, we have all the ingredients necessary to compute the volatility for both the near term and the next term options.

The formula to compute the volatility is given by the expression:

$$\sigma_1^2 = \frac{2}{T_1} \sum_I \frac{\Delta K_i}{K_i^2} e^{R_1 T_1} Q(K_i) - \frac{1}{T_1} \left[\frac{F_1}{K_0} - 1\right]^2$$

$$\sigma_2^2 = \frac{2}{T_2} \sum_I \frac{\Delta K_i}{K_i^2} e^{R_2 T_2} Q(K_i) - \frac{1}{T_2} \left[\frac{F_2}{K_0} - 1\right]^2$$

where σ_1^2 and σ_2^2 are respectively the volatility of the near term and next term options.

Now that we have reviewed the theory, we can apply it to our problem. Let's start by computing the boundaries for the near term and the next term options. Since we use mark prices, we cannot apply the standard approach to identify the boundaries. For this reason, I assumed as the lower boundary the second last put option with strike price K = 5500; the reason for doing this is that we need a value immediately above and below such value. Similarly, the upper boundary is identified by the call option with the strike price K = 7750.

The next step consists in computing the contribution of each option to the VIX, but before starting to compute the contribution, an intermediate step has been performed, namely the transformation of our option prices expressed in Bitcoins into the ones expressed in USD. The transformation consists in the multiplication between the price of the options expressed in Bitcoins and the value of Deribit Bitcoin Index at that specific time, which is expressed in USD. This step is necessary to get the correct results. Table 3.3 displays the results.

Table 3.3 – *Contribution of the 1st near-term option*

	Option Type	mid	contrib
5000	Put	0.026443	3.972586868301833657363564470E-7
5500	Put	4.86882	0.00006045079472681481089978835627
5750	Put	11.2714	0.0001280399854636102443793605200
6000	Put	15.2487	0.0001590867343019707452606683365
6250	Put	21.8273	0.0002098669588425670596687331935
6500	Put	103.931	0.0009238948702786100514718025144
6750	Put	77.2377	0.0006366855759931738612706635676
7000	Put	30.5212	0.0002339421080567975756925955167
7250	Put	46.9928	0.0003357829589222469384104192353
7500	Call/Put Avg	60.9697	0.0002741019942214790200844187502
7750	Call	37.6365	0.0001568982717817530183240293619
8000	Call	25.2289	0.00009870306777631337244940332467

Source: Created with Python

Analogously, the same procedure is applied to the next-term options. Once computed the contribution of each option to the VIX, we can now compute the volatility of both near-term and next-term options. The results are the following:

$$\sigma_1^2 = 0.09253440936$$
$$\sigma_2^2 = 0.16626152264$$

- *Step 3: Calculating the VIX Index*

To compute the VIX index we take the square root of the weighted average of σ_1^2 and σ_2^2 and multiply it by 100 to get the VIX value; in formula it can be expressed as:

$$VIX = 100 * \sqrt[2]{\frac{N_{365}}{N_{30}}\left[T_1\sigma_1^2\left(\frac{N_{T_2} - N_{30}}{N_{T_2} - N_{T_1}}\right) + \ T_2\sigma_2^2\left(\frac{N_{30} - \ N_{T_1}}{N_{T_2} - N_{T_1}}\right)\right]}$$

where:

N_{T_1} = number of minutes to settlement of the near-term options (34.560);

N_{T_2} = number of minutes to settlement of the next-term options (44.640);

N_{30} = number of minutes in 30 days (30*1.440 = 43.200);

N_{365} = number of minutes in a 365-day year (365*1.440 = 525.600).

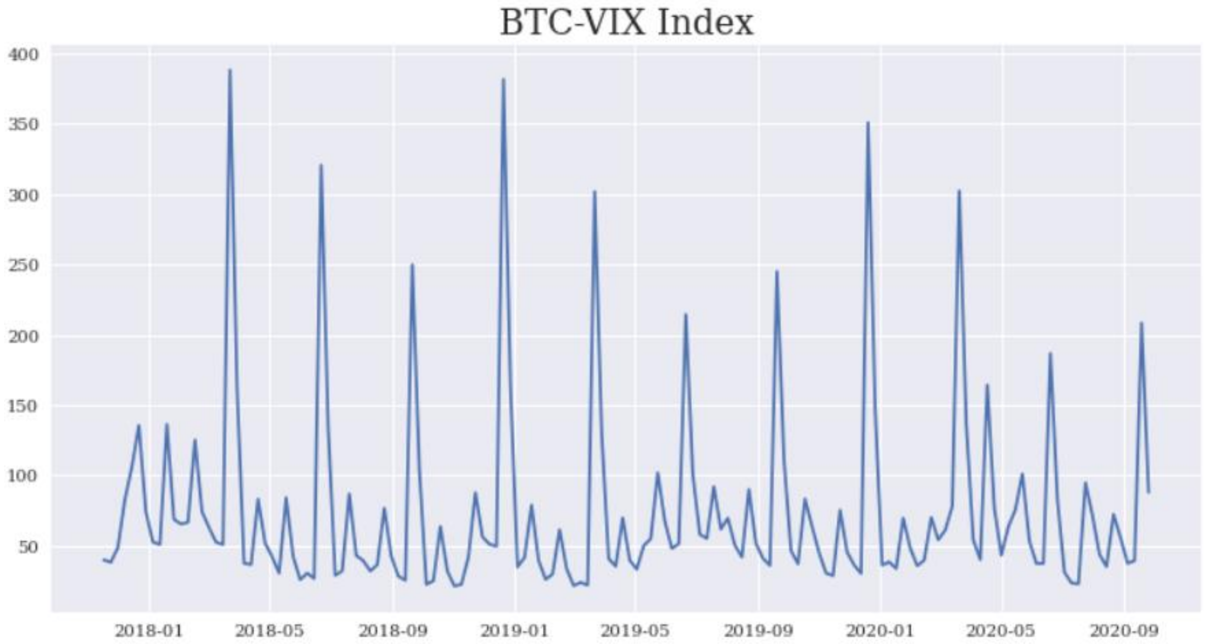
Finally, applying the formula, we find that on 24/10/2017 the VIX index is equal to:

VIX = 100 * 0.39728 = 39.728

- *Step 4: VIX index representation*

I have applied the above described steps to all our options, by rolling to new maturities each week. Now, after the VIX value has been computed for each contract maturity, we can represent it graphically (Figure 3.2).

Figure 3.2 – *BTC-VIX index*



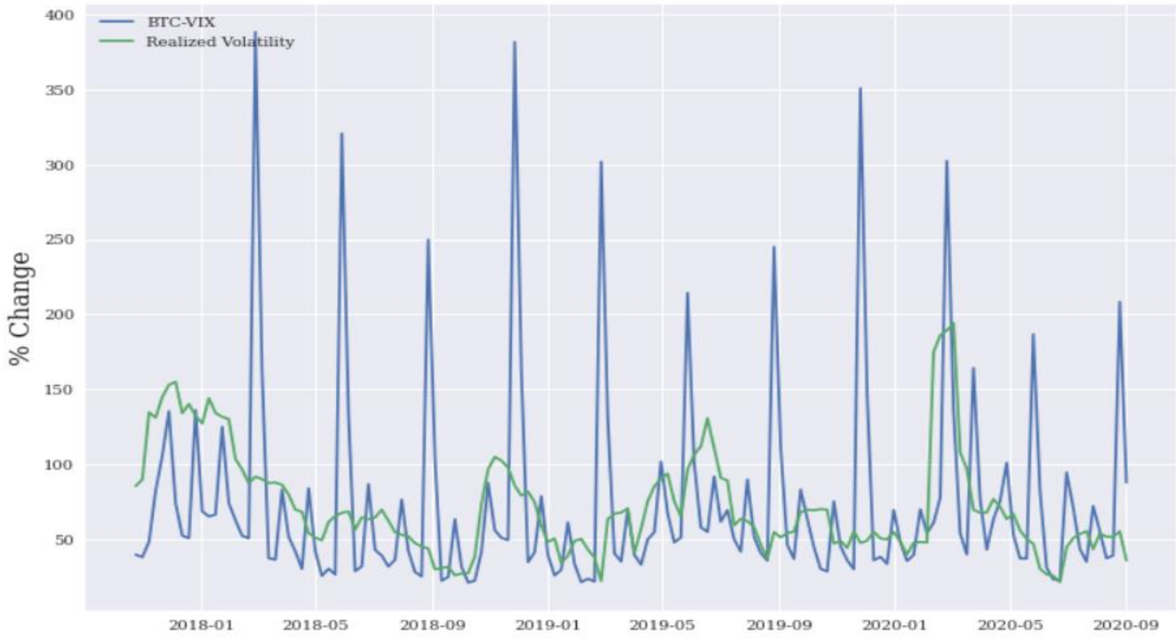
Source: created with Python

The initial visual inspection suggests that there are too many, and perhaps unusual, spikes. To check which are the unusual spikes, we compare the VIX level, which is the expectation of the annualized realized volatility over the next 30 days, to the realized volatility itself. The annualized realized volatility is computed using daily BTC prices as log price changes over the next 30 calendar days[15], according to the following formula:

$$Realized\ Volatility = \sqrt{12 * \sum_{i=1}^{T} \left(log\left(\frac{P_t}{P_{t-1}}\right)\right)^2}$$

[15] Meaning 30 days after the day t in which the VIX values was computed

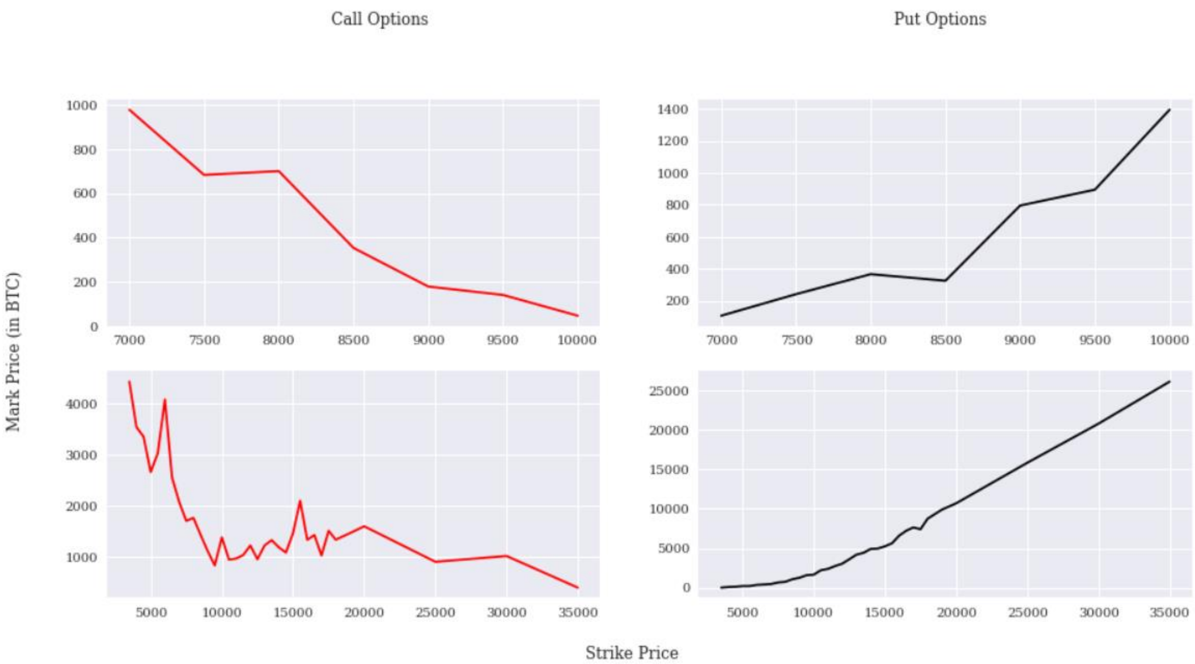
Figure 3.3 – *VIX vs Realized Volatility*



Source: created with Python

As underlined before, only the spike which occurs in March is correct, while all other spikes are unusual. At the beginning of the analysis we saw that the no-arbitrage condition does not hold for BTC options; for this reason, I would like to analyze the options used in the computation of the VIX value which corresponds to the first big spike in 23/03/2018 which carries a VIX value of 388.27. Figure 3.4 displays the results for the near-term options (first plot line) and the next-term options used in the computation of that specific VIX value.

Figure 3.4 – *Option chain with expiration 23/03/2018 and 30/03/2018*



Source: created with Python

As we can see, the no-arbitrage condition does not hold, especially for the call options with expiration date on 30/03/2018 (position [1,0] in the matrix plot). This of course is one of the reasons which affect the accuracy of our estimations. For this reason, I would like to go deeper in the analysis and perform a sort of inverse engineering process. I start by looking at the spread between the volatilities of near and the next term options and find that to each unusual peak corresponds a very large spread between the two volatilities, which is the reason of peaks. For instance, the absolute value of the difference between the near and next term option volatilities with expiration date 23/03/2018 is equal to 16.65 (|0.32-16.97|) versus an average value of 0.2948. In this case, the reason of the peak seems to be the next-term volatility. By decomposing the next-term volatility, I find that the reason of such large spread can be linked to a different size of the option chains used in the computations of that specific values. In fact, the near-term option chain with expiration on 23/03/2018 is made of 7 elements, where the sum of the contributions of the call options is equal to 0.002 and the sum of the contributions of the put options is equal to 0.006. At the same time, the next-term option chain with expiration on 30/03/2018 contains 35 elements, where the sum of contributions of the call and the put options is respectively 0.63 and 0.056. I believe that this, along with the fact that the no-arbitrage condition is violated, may be responsible of the additional spikes that we can clearly see from the visual inspection of the VIX index.

Nevertheless, it should also be considered that usually the implied volatility is much higher than the subsequent realized volatility, because Deribit Bitcoin Index, as well as any stock index, do not follow the lognormal distribution assumed by Black and Scholes model.

Conclusion

In this paper, I initially presented an overview of what Bitcoins and the underlying Blockchains are. Then I started the computational part by estimating the conditional volatility using the GARCH model, where I showed that the best configuration of the GARCH is the GARCH(1,1). The implementation of the GJR-GARCH(1,1) showed that the "leverage effect" is not a feature of Bitcoins and for this reason I performed the forecasting exercise by relying on GARCH(1,1). The results showed that GARCH(1,1) perform well in estimating the subsequent realized volatility. The comparison has been made using the realized volatility computed from 1-min BTC prices. To account for bid-ask bound, a feature present in high frequency, I relied on an alternative way of estimating realized volatility, the one similar to a Newey-West Estimator. The mean squared error of my estimation is surely not low, 3.705; however, I believe it is a good result accounting for the Bitcoin illiquidity of the initial years and possible data corruption.
After that, I started the estimation of the VIX index, where interesting features has been identified. One of this is the fact that no-arbitrage condition does not hold, for some option chains more than for others. In fact, some option chains present a very unusual behavior, where in the money options were priced as deep out of the money options. However, I also believe that this discovery may be biased by the fact that I did not have the access to the bid-ask quotes. As we can observe just by looking at option chains of some biotech stocks, we can clearly identify a similar behavior if we look at options mid-price without knowing the bid-ask quotes. The illiquidity is one of the drivers that may have produced the conclusion that the 'no-arbitrage' condition does not for Bitcoin markets. After the initial cleaning, I started with the computational part of the VIX index. The final value I computed for the VIX index showed some unusual behavior or spikes. In fact, I have identified spikes which were not something natural, since the only correct "big" spike was the one related to the shock happened from February to March of 2020. To understand the causes of these spikes, I decided to do an inverse engineering process to spot possible errors along the way. The process showed that the spikes were due to the big spread, in absolute values, between the near-term options volatilities and the next-term ones. To explain the reason of this big spread, the volatility has been decomposed in its components. Here I identified that possible reasons for the spikes may be, among others, the mismatch of the options chains and the large spread of strikes for that specific chains. With mismatch of the options I am referring to the fact that, for instance, the first big spike in the VIX index was computed by using near-term options with strikes in it from \$7000 to \$10.000, while near term options with a strike range from \$3500 to \$35000. Another reason may be the fact that the no-arbitrage condition does not hold. This is something I could obviously get rid of, perhaps by using the curve fitting technique; however, I preferred not to apply such technique and to proceed forward in the computations of the VIX index. In fact, I believe that

applying the curve fitting technique would make the options satisfy the no-arbitrage condition, but in doing so I would have biased computations. For this reason, I decided not to proceed in such way and go forward but taking into account the fact that I was assuming that no-arbitrage condition holds.

Finally, it is time to answer to the initial question of this paper: does GARCH provide extra information in forecasting the subsequent realized volatility once VIX is included as a regressor? To answer the question, I ran the following regression:

$$RVol_{t,t+30} = a + bVIX_t + cGARCH_t + e_{t,t+30}$$

Well, even if it is not really powerful, due to the low R^2 (3.6%), the answer is in line with our initial expectations that GARCH is unable to provide additional information once VIX is included as a regressor. In fact, the GARCH is not statistically significant at any level (p-value = 0.992), while VIX is statistically significant at $\alpha = 10\%$ (p-value = 0.091).

Bibliography

BOLLERSLEV, TIM and TAUCHEN, GEORGE E. and ZHOU, HAO, *Expected Stock Returns and Variance Risk Premia* (July 1, 2008). AFA 2008 New Orleans Meetings Paper, Review of Financial Studies, Forthcoming, Duke Department of Economics Research Paper No. 5, CREATES Research Paper No. 2008-48: https://ssrn.com/abstract=948309

CARR, PETER P. and WU, LIUREN, *A Tale of Two Indices* (March 2004). NYU Working Paper No. SC-CFE-04-01: https://ssrn.com/abstract=1297081

CHAN F.M and LAZZARA C.J., *The valuation of low volatility*, February 2020

CBOE, Cboe VIX: *White Paper Cboe Volatility Index*, August 2014

EDWARDS T. and PRESTON H., *Reading VIX: Does VIX predict future volatility?*, November 2017

EDWARDS T. and PRESTON H., *A Practioner's guide to reading VIX?,* December 2017

HAMILTON J.D, *Time series analysis,* 1994 Princeton University Press

HOANG LT, BAUR DG: *Forecasting bitcoin volatility: Evidence from the options market*, J Futures Markets. 2020;1-19: https://doi.org/10.1002/fut.22144

HUANG, D., SCHLAG, C., SHALIASTOVICH, I., & THIMME, J., *Volatility-of-Volatility Risk. Journal of Financial and Quantitative Analysis,* 54(6), 2423-2452. doi:10.1017/S0022109018001436

HILPISCH Y., *Python for Finance: Analyze Big Financial Data,* December 30 2014

MORAN, MATTHEW T. and LIU, BERLINDA, *The VIX Index and Volatility-Based Global Indexes and Trading Instruments - A Guide to Investment and Trading Features* (August 7, 2020). CFA Institute Research Foundation Briefs, April 2020, ISBN 978-1-944960-95-7: http://dx.doi.org/10.2139/ssrn.3668983

MRAN M., *Thirty Volatility Indexes: Worldwide Tools to Gauge Sentiment and Diversity Portfolios,* The Journal of Index Investing Feb 2014, 4 (4) 69-87; DOI: 10.3905/jii.2014.4.4.069

MCKINNEY W., *Python for Data Analysis: Data Wrangling with Pandas, Numpy and IPYTHON,* November 3, 2017

NAKAMOTO S., *Bitcoin: A peer to peer electronic cash system*, March 2009: https://bitcoin.org/bitcoin.pdf

NUGROHO ET AL 2019, *Empirical performance of GARCH, GARCH-M, GJR-GARCH and log-GARCH models for returns volatility,* J. Phys.: Conf. Ser. 1307 012003; DOI: 10.1088/1742-6596/1307/1/012003

POSEDEL P., *Properties and Estimation of GARCH(1,1) model*

REINHARD HANSEN P. AND LUNDE A., *Does anything beat a GARCH(1,1)? A comparison based on test for superior predictive ability, 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.*, Hong Kong, China, 2003, pp. 301-307, doi: 10.1109/CIFER.2003.1196275.

Webography

BBW: https://www.buybitcoinworldwide.com/fee-calculator/

BROWNLEE J.: https://machinelearningmastery.com/category/time-series/

BUILTIN: *Blockchain: what is blockchain technology? How does it work?*, *https://builtin.com/blockchain*

CORBELLINI: *Elliptic curve cryptography: a gentle introduction*, May 17, 2015: https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/

DERIBIT: https://www.deribit.com/pages/docs/options

ELIOTT: *Chancellor Alistair Darling on brink of second bailout for banks*, January 03, 2009: https://www.thetimes.co.uk/article/chancellor-alistair-darling-on-brink-of-second-bailout-for-banks-n9l382mn62h

FRANKENFIELD: *What is it Bitcoin?*, February 18, 2021: https://www.investopedia.com/terms/b/bitcoin.asp

GENESIS: *What are Blockchain Protocols and how do they work?*, December 21, 2018: https://medium.com/@genesishack/draft-what-are-blockchain-protocols-and-how-do-they-work-94815be5efa7

MEDIUM: *Is Bitcoins a safe haven asset?*, May 29, 2020: https://medium.com/listing-help/is-bitcoin-a-safe-haven-asset-4ccbd26d15eb

PCMAG: https://www.pcmag.com/encyclopedia/term/bitcoin-wallet

SHEPPARD: https://www.kevinsheppard.com/teaching/mfe/notes/

TUDOR: *Letter to investors: The great monetary inflation*, May 09, 2020: https://seekingalpha.com/article/4345426-paul-tudor-jones-compelling-case-for-owning-bitcoin

WHITE: *What is a digital wallet and how does it work?*, October 5, 2020 https://www.thestreet.com/technology/what-is-a-digital-wallet-15100753